

УТВЕРЖДЕНО
Первый Заместитель
Министра здравоохранения
Российской Федерации

_____ В.А. Зеленский

« » 2023 г.

Подлинник электронного документа, подписанного ЭП,
хранится в системе электронного документооборота
Министерства Здравоохранения
Российской Федерации.

СВЕДЕНИЯ О СЕРТИФИКАТЕ ЭП

Сертификат: 01D90BBE79A4FA0000B13A100060002
Кому выдан: Зеленский Владимир Анатольевич
Действителен: с 09.12.2022 до 09.12.2023

УТВЕРЖДЕНО
Заместитель
Министра цифрового развития, связи,
и массовых коммуникаций
Российской Федерации

_____ О.Ю. Качанов

« » 2023 г.



Минцифры
России

Документ подписан
электронной подписью

СВЕДЕНИЯ О СЕРТИФИКАТЕ ЭП

Сертификат: 7A4564704E80E7A321E1FC92A539897D
Владелец: Качанов Олег Юрьевич
Действителен с 16-03-2023 до 08-06-2024

**Методические рекомендации по обеспечению взаимодействия
государственных информационных систем в сфере
здравоохранения субъектов Российской Федерации
с программным обеспечением «Витрина данных»
и подсистемой «Федеральная электронная регистратура»
единой государственной информационной системы в сфере
здравоохранения с целью обеспечения единого порядка
предоставления услуг (сервисов) посредством Единого портала
государственных и муниципальных услуг (функций)**

На 383 листах

2023 год

СОДЕРЖАНИЕ

ТЕРМИНЫ И СОКРАЩЕНИЯ	9
1 ОБЩАЯ ИНФОРМАЦИЯ	15
1.1 Область применения документа.....	15
1.2 Правовое основание для оказания услуги.....	16
1.3 Сведения о текущем порядке информационного взаимодействия.....	16
2 ПОРЯДОК ИНФОРМАЦИОННОГО ВЗАИМОДЕЙСТВИЯ.....	19
2.1 Общие сведения об информационном взаимодействии	19
2.2 Сведения об участниках информационного взаимодействия	20
2.2.1 Сведения о ЕПГУ	20
2.2.2 Сведения о НСУД и ПОДД СМЭВ 4.....	20
2.2.3 Сведения о ГИСЗ субъекта РФ	22
2.2.4 Сведения о ЕГИСЗ.....	23
2.3 Обеспечение качества данных	25
3 ПОРЯДОК ОКАЗАНИЯ УСЛУГ И СЕРВИСОВ В СФЕРЕ ЗДРАВООХРАНЕНИЯ НА ЕПГУ	29
3.1 Запись на прием к врачу	29
3.1.1 Краткое описание	29
3.1.2 Клиентский путь «Запись на прием к врачу»	30
3.1.3 Сценарий «Запись на прием к врачу».....	30
3.2 Запись на прием по направлению	34
3.2.1 Краткое описание	34
3.2.2 Клиентский путь «Запись на прием по направлению».....	35
3.2.3 Сценарий «Запись на прием по направлению»	35
3.3 Запись на прием к врачам, осуществляющим диспансерное наблюдение	37
3.3.1 Краткое описание	37
3.3.2 Клиентский путь «Запись на прием к врачам, осуществляющим диспансерное наблюдение»	38
3.3.3 Сценарий «Запись на прием к врачам, осуществляющим диспансерное наблюдение»	38
3.4 Запись на вакцинацию.....	41
3.4.1 Краткое описание	41
3.4.2 Клиентский путь «Запись на вакцинацию»	42

3.4.3 Сценарий «Запись на вакцинацию».....	42
3.5 Запись на прием на углубленную диспансеризацию для лиц, переболевших COVID-19	44
3.5.1 Краткое описание	44
3.5.2 Клиентский путь «Запись на приём на углубленную диспансеризацию»..	45
3.5.3 Сценарий «Запись на приём на углубленную диспансеризацию»	45
3.6 Просмотр и отмена записей, совершенных через иные источники (не ЕПГУ). 47	
3.6.1 Краткое описание	47
3.6.2 Клиентский путь процесса «Изменение и отмена записей, совершенных через иные источники»	48
3.6.3 Сценарий процесса «Изменение и отмена записей, совершенных через иные источники»	49
3.7 Вызов врача на дом	51
3.7.1 Краткое описание	51
3.7.2 Клиентский путь «Вызов врача на дом»	52
3.7.3 Сценарий «Вызов врача на дом».....	52
3.8 Телемедицинская консультация.....	55
3.8.1 Краткое описание	55
3.8.2 Клиентский путь «Телемедицинская консультация»	56
3.8.3 Сценарий «Телемедицинская консультация».....	56
3.9 Запись на прием на плановую госпитализацию	59
3.9.1 Краткое описание	59
3.9.2 Клиентский путь «Запись на прием на плановую госпитализацию»	60
3.9.3 Сценарий «Запись на прием на плановую госпитализацию»	60
3.10 Запись на прием на медицинское освидетельствование.....	63
3.10.1 Краткое описание	63
3.10.2 Клиентский путь «Запись на прием на медицинское освидетельствование»	64
3.10.3 Сценарий «Запись на прием на медицинское освидетельствование».....	64
4 ПРЕДЛАГАЕМОЕ РЕШЕНИЕ «РЕГИОНАЛЬНАЯ ВИТРИНА РАСПИСАНИЙ» 67	
4.1 Общие сведения о Витрине данных.....	67
4.2 Рекомендованные системные требования.....	67
4.3 Требования к наполнению данными.....	71
4.4 Требования к составу и структуре региональных витрин расписаний	71

4.5 Требования к предоставлению данных на ЕПГУ из региональной витрины расписаний	76
4.6 Требования к предоставлению данных в витрину из ГИСЗ субъекта РФ	81
5 ОРГАНИЗАЦИОННЫЕ МЕРОПРИЯТИЯ.....	85
5.1 Порядок перехода ГИСЗ субъекта РФ на решение с использованием технологии витрин данных.....	85
5.2 Порядок подключения ГИСЗ субъекта РФ к СМЭВ 4.....	87
6 ПОРЯДОК РАЗВЁРТЫВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	88
6.1 Порядок установки программного обеспечения «Витрина данных Медиум» ..	88
6.1.1 Предварительные действия	88
6.1.2 Базовый план распределения серверов для конфигурации Медиум	94
6.1.3 Порядок установки конфигурации Медиум	94
6.1.4 Базовый порядок установки приложений конфигурации Медиум	94
6.1.5 Рекомендуемое распределение компонентов по серверам для конфигурации Медиум	95
6.1.6 Установка Kafka Zookeeper	97
6.1.7 Установка PostgreSQL	97
6.1.8 Установка ПО ProStore	97
6.1.9 Установка коннекторов к СУБД Postgres-13	99
6.1.10 Установка ПОДД-адаптера.....	100
6.1.11 Запуск и остановка программы	107
6.1.12 Проверка программы.....	107
6.1.13 Дополнительные возможности	112
6.1.14 Логирование.....	113
6.2 Порядок установки программного обеспечения «Агент ПОДД СМЭВ»	113
6.2.1 Условия работы с Агентом ПОДД СМЭВ	113
6.2.2 Подготовка к работе.....	114
ПРИЛОЖЕНИЕ А СВЕДЕНИЯ ДЛЯ НАПОЛНЕНИЯ РЕГИОНАЛЬНЫХ ВИТРИН РАСПИСАНИЙ	115
А.1 Сведения о расписании	115
А.1.1 Диаграмма связей и структуры объектов данных расписания	115
А.1.2 Описание логической структуры объектов данных расписания	116
А.2 Сведения о результатах контроля качества расписания.....	148
А.2.1 Описание логической структуры объектов данных результатов	

валидации	148
ПРИЛОЖЕНИЕ Б ОПИСАНИЕ ПРАВИЛ ПРОВЕРОК, ОСУЩЕСТВЛЯЕМЫХ В КОМПОНЕНТЕ «МОДУЛЬ УПРАВЛЕНИЯ БИЗНЕС-ПРАВИЛАМИ» ФЭР ЕГИСЗ.....	152
ПРИЛОЖЕНИЕ В ОБЩЕЕ ОПИСАНИЕ РЕШЕНИЯ «ЗАПИСЬ НА ПРИЕМ К ВРАЧУ».....	154
В.1 Бизнес-сценарий решения «Запись на прием к врачу».....	154
В.1.1 Архитектура решения	154
В.2 Перечень регламентированных запросов.....	155
В.3 Пример спецификации OpenAPI.....	164
В.3.1 Бронирование слота.....	171
В.3.2 Отмена бронирования выбранного слота.....	173
В.4 Коды ошибок.....	174
В.5 Примеры запросов/ответов.....	175
В.5.1 Получение данных по регламентированным запросам	175
В.5.2 Пример запроса/ответа на бронирование.....	177
В.5.3 Пример запрос/ответа на отмену бронирования	179
В.6 Тестовые сценарии	180
В.6.1 Проверка реализации сценария записи себя, ребенка или другого человека» по услуге «Запись на прием к врачу».....	180
Открылся экран «Укажите данные ребёнка».....	181
В.6.2 Проверка реализации сценария записи и отмены записи к врачу по направлению по услуге «Запись на прием к врачу»	182
В.6.3 Проверка реализации сценария записи и отмены записи в выбранную медицинскую организацию по услуге «Запись на прием к врачу»	183
В.6.4 Проверка реализации сценария записи и отмены записи к врачу выбранной специальности по услуге «Запись на прием к врачу».....	184
В.6.5 Проверка реализации сценария отображения информации по временно недоступным расписаниям врачей (отпуск, больничный, командировка, прочее)	185
В.6.6 Проверка реализации проектирования сценария записи и отмены записи по записи по диспансерному наблюдению	186
В.7 Проверки и фильтры на стороне ЕПГУ.....	187
В.7.1 Проверка доступности витрины и редирект на старую форму записи	187

В.7.2	Определение региона пользователя.....	187
В.7.3	Фильтр списка доступных специальностей	188
В.7.4	Фильтр списка доступных слотов.....	188
В.8	Дополнительные настройки	188
В.8.1	Рекомендации по быстродействию.....	189
В.8.2	Пример организации Витрины РМИС для выполнения сценария «Запись на прием к врачу»	189
В.8.3	Добавление Витрины-кэш	190
В.8.4	Пример исходного кода для организации Витрины-кэш	191
В.8.5	Оптимизация настроек Postgres	196
ПРИЛОЖЕНИЕ Г АНКЕТА «СВЕДЕНИЯ ОБ ИТ-ЛАНДШАФТЕ ВЕДОМСТВА»		199
ПРИЛОЖЕНИЕ Д ФОРМА ЗАПРОСА ОБЪЕМА УСЛУГ ГОСОБЛАКА (ОБЩЕСТВЕННОЕ ОБЛАКО)		202
ПРИЛОЖЕНИЕ Е ТИПОВАЯ МЕТОДИКА ОБСЛУЖИВАНИЯ ТИПОВОГО ПО «ВИТРИНА ДАННЫХ» 1035009567450.СМЭВ-ЕСНСИ-ЕИП НСУД-2023-1.М 01 ОП1		216
Е.1	Описание набора инструментов для работы с витриной данных НСУД...	217
Е.1.1	DBeaver. Описание инструмента и его функциональность.....	217
Е.1.2	Zoonavigator. Описание инструмента и его функциональность	224
Е.1.3	OffsetExplorer. Описание инструмента и его функциональность	232
Е.1.4	VisualVM. Описание инструмента и его функциональность	237
Е.2	POSTGRESQL. Настройка и обслуживание	242
Е.2.1	Периодическое обслуживание	242
Е.2.2	Планировщик запросов	244
Е.2.3	Перестройка Индексов	245
Е.2.4	Динамическое статистическое представление - pg_stat_activity	248
Е.2.5	Просмотр активности PostgreSQL	248
Е.3	Настройка и обслуживание Kafka и Zookeeper.....	248
Е.3.1	Описание инструментов, входящих в комплект поставки Kafka	248
Е.3.2	Описание zkCli, входящего в комплект поставки Zookeeper	250
Е.3.3	Схема данных для Prostore в Zookeeper	253
Е.3.4	Схема данных для ПОДД-Адаптеров в Zookeeper	254
Е.3.5	Схема данных для Брокеров в Zookeeper	255

Е.3.6	Описание топиков Kafka	256
Е.3.7	Скрипт для удаления топиков Kafka	257
Е.3.8	Контроль количества транзакционных логов и снимков в Zookeeper	259
Е.4	Дополнительные операции, резервное копирование, логирование, очистка данных, восстановление данных	259
Е.4.1	Восстановление витрины данных	259
Е.4.2	Резервное копирование и восстановление витрины при помощи DTM-Tools	264
Е.4.3	Развертывание логической базы данных из бэкапа	266
Е.4.4	Логирование витрины	269
Е.4.5	Очистка данных при помощи TRUNCATE HISTORY	276
Е.5	Мониторинг витрины данных	278
Е.5.1	Базовые настройки Grafana для витрины данных	278
Е.5.2	Дашборд Lite	279
Е.5.3	Дашборд Node Exporter Full	280
Е.5.4	Загрузка дашбордов	283
ПРИЛОЖЕНИЕ Ж МЕТОДИКА ОЦЕНКИ КАЧЕСТВА РАБОТЫ УСЛУГИ «ЗАПИСЬ НА ПРИЕМ К ВРАЧУ» С ИСПОЛЬЗОВАНИЕМ ЕДИНОГО ПОРТАЛА ГОСУДАРСТВЕННЫХ И МУНИЦИПАЛЬНЫХ УСЛУГ (ФУНКЦИЙ)		285
Ж.1	Назначение методики	285
Ж.2	Перечень отчетов и источников данных	285
Ж.2.1	Перечень отчетов услуги «Запись на прием к врачу (на витринах данных)»	285
Ж.2.2	Перечень отчетов услуги «Запись на прием к врачу (КУ ФЭР)»	286
Ж.2.3	Перечень общих отчетов	286
Ж.3	Перечень отслеживаемых показателей	286
Ж.3.1	Перечень показателей услуги «Запись на прием к врачу (на витринах данных)»	286
Ж.3.2	Перечень показателей услуги «Запись на прием к врачу (КУ ФЭР)»	288
Ж.4	Методика расчета показателей	289
Ж.4.1	Передача данных по событиям услуги «Запись на прием к врачу»	289
Ж.4.2	Расчет показателей по результатам прохождения услуги «Запись на прием к врачу»	299

Ж.4.3	Расчет длительности недоступности услуги в регионе	300
Ж.4.4	Расчет конверсии	301
Ж.4.5	Методика оценки качества данных, размещаемых в витринах субъектов РФ	302
Ж.5	Мониторинг показателей и реагирование на их отклонение от нормативных значений для услуги «запись на прием к врачу (на витринах данных)»	311
Ж.5.1	Соглашение о коммуникациях	311
Ж.5.2	Мониторинг качества функционирования услуги	311
Ж.5.3	Мониторинг качества данных, размещаемых в витринах субъекта РФ	329
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....		380

ТЕРМИНЫ И СОКРАЩЕНИЯ

В документе используются обозначения и сокращения, представленные в следующей таблице:

Таблица 1 – Перечень терминов и сокращений

Термин	Содержание
Apache Kafka	Распределённый программный брокер сообщений, проект с открытым исходным кодом, разрабатываемый в рамках фонда Apache
API	Программный интерфейс приложения (англ. Application Programming Interface)
Java DataBase Connectivity (JDBC)	Стандарт взаимодействия Java-приложений с различными СУБД
JDBC-драйвер	Библиотека классов, реализующая стандарт JDBC и подключения к источнику данных с использованием специализированного протокола, поддерживаемого источником данных
OpenAPI, спецификация OpenAPI	Формализованная спецификация и экосистема множества инструментов, предоставляющая интерфейс между front-end системами, кодом библиотек низкого уровня и коммерческими решениями в виде API
Representational State Transfer (REST)	Архитектурный стиль взаимодействия компонентов распределённого приложения в сети
Агент (ПОДД СМЭВ)	Типовое программное обеспечение, устанавливаемое в контуре ИС УВ и обеспечивающее сопряжение Витрин данных ИС УВ и ИС УВ с ПОДД СМЭВ
БД	База данных
Витрина (данных)	Комплекс программных и технических средств в составе информационно-телекоммуникационной инфраструктуры Участника взаимодействия, обеспечивающий хранение и предоставление данных другим Участникам взаимодействия с использованием ПОДД СМЭВ
Внешняя ИС	Информационная система, зарегистрированная в компоненте «Ядро ПОДД» и участвующая в оказании услуг (ИС Пользователя или ИС Поставщика)

ГИСЗ субъекта РФ	Государственная информационная система в сфере здравоохранения субъекта Российской Федерации
ЕГИСЗ	Единая государственная информационная система в сфере здравоохранения
ЕПГУ, ФГИС ЕПГУ	Единый портал государственных и муниципальных услуг (функций)
ЕСИА	Единая система идентификации и аутентификации в инфраструктуре, обеспечивающей информационно-технологическое взаимодействие информационных систем, используемых для предоставления государственных и муниципальных услуг в электронной форме
ЕСКС	Единая система контекстных справок ЕПГУ
ЗНП	Запись на приём (к врачу; по специальности, по направлению)
ИА	Интеграционная подсистема «Единая система идентификации, аутентификации и авторизации» ЕГИСЗ
Инструкция по работе в ЕИП НСУД	Инструкция по работе в федеральной государственной информационной системе «Единая информационная платформа Национальной системы управления данными» (ЕИП НСУД) в части описания информационных ресурсов, информационных систем, наборов данных, модели витрины данных, а также формирования регламентированных запросов. Актуальная версия размещена на веб-портале НСУД и доступна по ссылке: https://nsud.gosuslugi.ru/ifp/portals/documents/54
ИПС	Сервис ИПС, Единая интеграционная шина ЕГИСЗ
ИПМИ	Информационная подсистема мониторинга инфраструктуры
ИС	Информационная система
ИС Пользователя	Информационная система, предоставляющая гражданам возможность записи на прием к врачу
ИС Поставщика	Информационная система, предоставляющая информацию в «Витрину данных»
ИС УВ	Информационная система организации, участвующей в информационном обмене через СМЭВ
ИЭП	Инфраструктура электронного правительства
КУ ФЭР	Компонент «Концентратор услуг ФЭР» подсистемы «Федеральная электронная регистратура» ЕГИСЗ

ЛК	Личный кабинет
МО	Медицинская организация
Минздрав России	Министерство здравоохранения Российской Федерации
Минцифры России	Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации
Направление	Документ "Направление на госпитализацию, восстановительное лечение, обследование, консультацию" (далее – "Направление...") выдается медицинским учреждением, осуществляющим наблюдение или прием больных и несет в себе информацию, необходимую для оказания пациенту консультативно-диагностических услуг, специализированного лечения в стационаре или медицинской реабилитации. Документ может использоваться как при направлении пациента в стороннее медицинское учреждение, так и между разными структурными подразделениями и филиалами одной организации
Направление на телемедицинскую консультацию	Процесс направления на телемедицинскую консультацию с учётом наличия у пациента направления учётной формы
ОГРН	Основной государственный регистрационный номер
ОКАТО	Общероссийский классификатор объектов административно-территориального деления
ОМС	Обязательное медицинское страхование
Открытый случай	Медицинский случай, открытый на очном приеме, дающий возможность пациенту получить телемедицинскую консультацию для коррекции лечения или закрытия случая
ПО	Программное обеспечение
Подсистема обеспечения доступа к данным СМЭВ, ПОДД СМЭВ	Часть транспортной подсистемы СМЭВ, обеспечивающая доступ к данным, размещённым на Витринах данных или в ИС Поставщика данных
Пользователь	Гражданин, обратившийся за получением услуг в сфере здравоохранения на ЕПГУ
ПОС	Подсистема обратной связи ЕПГУ
Поставщик данных	Участник взаимодействия, являющийся источником данных для других участников и использующий ПОДД СМЭВ для передачи данных

Потребитель данных	Участник взаимодействия, получающий данные от Поставщиков данных для дальнейшей их обработки и использующий для передачи запросов и получения данных ПОДД СМЭВ
НСУД, ФГИС НСУД, ЕИП НСУД	Единая информационная платформа Национальной системы управления данными
ОИВ	Орган исполнительной власти
Оператор ИЭП	Оператор эксплуатации инфраструктуры электронного правительства, определенный в соответствии с Распоряжением Правительства РФ от 15.10.2009 № 1475-р
Регламентированный запрос (РЗ)	SQL-запрос, выраженный в терминах Модели данных, загруженной в ПОДД, и зарегистрированный в Ядре ПОДД СМЭВ под символической мнемоникой, используемой ИС Потребителя ПОДД СМЭВ для выполнения регламентированного запроса. Может иметь параметры, значения которых задаются Потребителем данных ПОДД СМЭВ при выполнении регламентированного запроса
РМИС	Региональная медицинская информационная система
РП ВС «Загрузка регламентированного запроса»	Руководство пользователя Вида сведений в единой системе межведомственного электронного взаимодействия. Документ размещён на Технологическом портале СМЭВ.
РЭМД	Подсистема ЕГИСЗ «Реестр электронных медицинских документов»
Сведения о расписании	Сведения о расписании и ресурсах медицинских организаций, содержащие также сведения о пациентах, направлениях, записях на приём и прочие сведения, необходимые для Совокупность бизнес-функций государственных услуг и оказания гражданам сервисов в сфере здравоохранения, посредством ФГИС «Единый портал государственных и муниципальных услуг (функций)»
Сведения о валидации	Сведения о результатах контроля качества расписания (сведений о расписании)
СИП ЕГИСЗ	Подсистема автоматизированного сбора информации о показателях системы здравоохранения из различных источников и предоставления отчетности ЕГИСЗ (также ФЭР.ВІ)
СМЭВ	Система межведомственного электронного взаимодействия
СМЭВ 3	Единая система межведомственного электронного взаимодействия, функционирующая в соответствии с Методическими рекомендациям по работе со СМЭВ версии 3.x

СМЭВ 4	Единая система межведомственного электронного взаимодействия нового поколения, основанная на использовании технологий Витрин данных
СНИЛС	Страховой номер индивидуального лицевого счёта
СП МО	Структурное подразделение медицинской организации
СТП	Служба технической поддержки
СЦ МЦ	Федеральный ситуационный центр электронного правительства
ТСМЭВ	Тестовая среда СМЭВ
ТФОМС	Территориальный фонд обязательного медицинского страхования
УКЭП	Усиленная квалифицированная электронная подпись
Услуги	Совокупность бизнес-функций государственных услуг и сервисов в сфере здравоохранения, предоставляемых гражданам посредством ФГИС «Единый портал государственных и муниципальных услуг (функций)»: – Запись на прием к врачу; – Запись на прием по направлению; – Запись на прием к врачам, осуществляющим диспансерное наблюдение; – Запись на вакцинацию; – Запись на прием на углубленную диспансеризацию для лиц, переболевших COVID-19; – Отмена записей, совершенных через иные источники (не ЕПГУ); – Вызов врача на дом.
Участник взаимодействия, УВ	Орган или организация, участвующая в информационном обмене через СМЭВ
ФЗ	Федеральный закон
ФИО	Фамилия, Имя, Отчество (при наличии)
ФМБА	Федеральное медико-биологическое агентство
ФНСИ	Федеральный реестр нормативно-справочной информации в сфере здравоохранения
ФОИВ	Федеральный орган исполнительной власти
ФОМС	Фонд обязательного медицинского страхования
ФРМО	Подсистема ЕГИСЗ «Федеральный реестр медицинских организаций»

ФРМР	Подсистема ЕГИСЗ «Федеральный регистр медицинских работников»
ФСТЭК	Федеральная служба по техническому и экспортному контролю
Ядро ПОДД СМЭВ	Часть ПОДД СМЭВ, предназначенная для проверки и маршрутизации запросов между участниками взаимодействия

1 ОБЩАЯ ИНФОРМАЦИЯ

1.1 Область применения документа

Настоящий документ представляет собой методические рекомендации для государственных информационных систем в сфере здравоохранения субъектов Российской Федерации (далее – ГИСЗ субъекта РФ) по использованию программного обеспечения «Витрина данных» в рамках модернизации информационного взаимодействия при управлении потоками пациентов и ресурсами медицинских организаций.

Модернизация информационного взаимодействия осуществляется в части оказании следующих услуг и сервисов в сфере здравоохранения на Едином портале государственных услуг и функций (далее – Услуги):

- Запись на прием к врачу;
- Запись на прием по направлению;
- Запись на прием к врачам, осуществляющим диспансерное наблюдение;
- Запись на вакцинацию;
- Запись на прием на углубленную диспансеризацию для лиц, переболевших COVID-19;
- Отмена записей, совершенных через иные источники (не ЕПГУ);
- Вызов врача на дом;
- Телемедицинская консультация;
- Запись на прием на плановую госпитализацию;
- Запись на прием на медицинское освидетельствование.

Материал документа описывает порядок модернизированного информационного взаимодействия (п.2), порядок оказания отдельных Услуг (п. 3), описание решения «Региональная витрина расписаний» (п.4), а также порядок действий по модернизации информационного взаимодействия, включая: порядок перехода на решение с использованием «Витрины данных» (п. 5), порядок установки программного обеспечения (п.6), состав сведений для наполнения «Региональной витрины расписаний субъекта РФ» данными, необходимыми для надлежащего оказания Услуг (Приложение А), описание

правил проверок, осуществляемых в рамках контроля за качеством сведений о расписании медицинских организаций Приложение А(Приложение Б).

1.2 Правовое основание для оказания услуги

Правовым основанием оказания медицинскими организациями услуг в области здравоохранения на ЕПГУ является распоряжение Правительства Российской Федерации от 25 апреля 2011 г. № 729-р «О перечне услуг, оказываемых государственными и муниципальными учреждениями и другими организациями, в которых размещается государственное задание (заказ) или муниципальное задание (заказ), подлежащих включению в реестры государственных или муниципальных услуг и предоставляемых в электронной форме».

Основанием для предоставления Услуг с использованием ЕГИСЗ являются распоряжение Правительства Российской Федерации от 15.11.2017 №2521-р «О Перечне услуг в сфере здравоохранения, возможность предоставления которых гражданам в электронной форме посредством единого портала государственных и муниципальных услуг обеспечивает единая государственная информационная система в сфере здравоохранения», а также постановление Правительства Российской Федерации от 9 февраля 2022 г. № 140 «О единой государственной информационной системе в сфере здравоохранения».

Основанием для предоставления Услуг с использованием НСУД и программного обеспечения «Витрина данных» является Указ Президента Российской Федерации от 21 июля 2020 г. № 474 «О национальных целях развития Российской Федерации на период до 2030 года», а также поручение Заместителя Председателя Правительства Российской Федерации Т.А. Голиковой от 29 июня 2021 года № ТГ-П12-8627.

1.3 Сведения о текущем порядке информационного взаимодействия

В настоящий момент в субъектах Российской Федерации для предоставления услуги «Запись на прием к врачу» на ЕПГУ используется компонент «Концентратор услуг ФЭР» подсистемы ФЭР ЕГИСЗ (далее – КУ ФЭР).

КУ ФЭР обеспечивает:

- получение заявления на оказание услуг, поступающих с ИС Пользователя;

- двухстороннюю трансформацию протоколов обмена данными ИС Пользователя и ИС Поставщика, зарегистрированными в компоненте «Концентратор услуг ФЭР»;
- маршрутизацию заявлений в ИС Поставщиков, зарегистрированные в компоненте «Концентратор услуг ФЭР»;
- обеспечение данными, необходимыми для оказания услуг посредством обращения непосредственно к сервисам ИС Поставщиков;
- регистрацию/запись в журнал результатов взаимодействия со внешними ИС.

Базовый сценарий оказания услуги «Запись на прием к врачу» на ЕГИС в рамках текущего порядка информационного взаимодействия приведен на рисунке ниже.

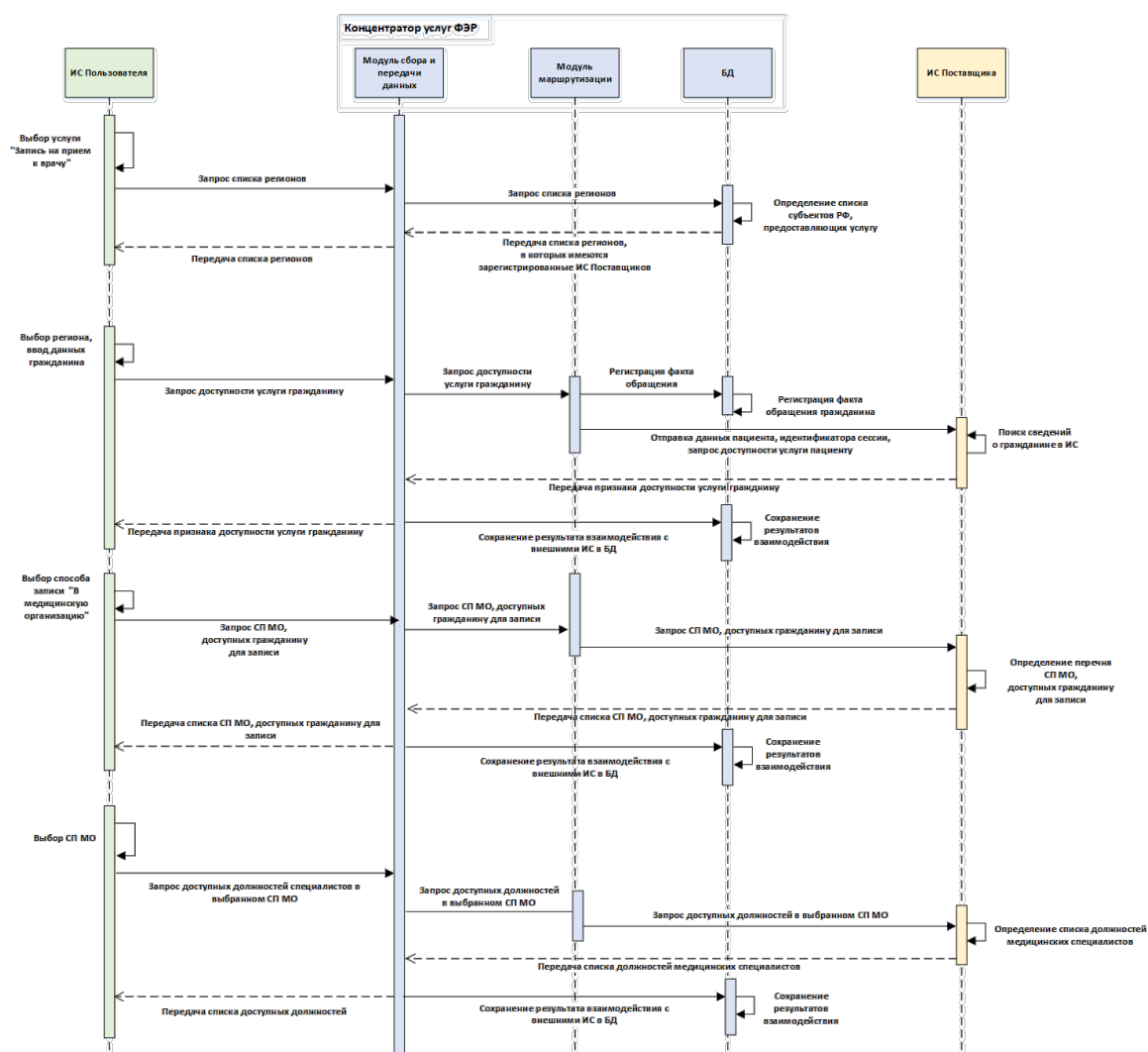


Рисунок 1 –Процесс предоставления Пользователю услуги «Запись на прием к врачу» посредством компонента «Концентратор услуг ФЭР» подсистемы ФЭР ЕГИСЗ

Подключение ИС Поставщика к компоненту КУ ФЭР и порядок информационного взаимодействия ГИСЗ субъекта Российской Федерации, КУ ФЭР и ЕПГУ определяется

документом «Методические рекомендации по обеспечению взаимодействия информационных систем в сфере здравоохранения субъектов Российской Федерации с Федеральной электронной регистратурой для подключения к услугам «Запись на прием к врачу» и «Вызов врача на дом» » через компонент «Концентратор услуг ФЭР»». Документ разработан Минздравом России, размещен на Портале оперативного взаимодействия участников ЕГИСЗ и доступен по ссылке: <https://portal.egisz.rosminzdrav.ru/materials/541>.

2 ПОРЯДОК ИНФОРМАЦИОННОГО ВЗАИМОДЕЙСТВИЯ

2.1 Общие сведения об информационном взаимодействии

В рамках модернизации информационного взаимодействия определена новая схема информационного взаимодействия при оказании Услуг на ЕПГУ с использованием НСУД и типового решения «Витрина данных».

Состав подсистем, модулей и компонентов предлагаемого решения оказания Услуг на ЕПГУ, а также общая схема взаимодействия приведена на рисунке ниже.

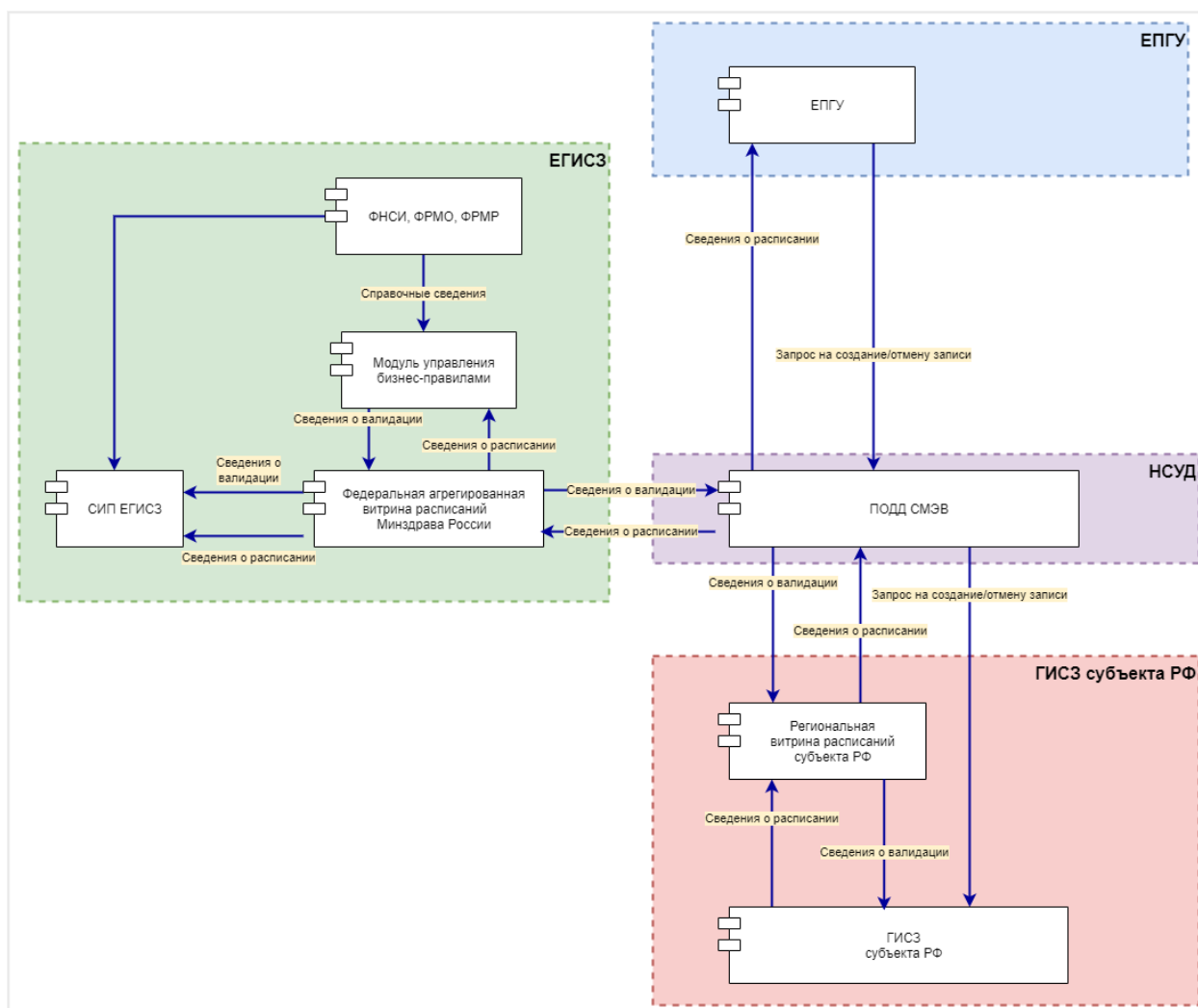


Рисунок 2 – Общая схема информационного взаимодействия

2.2 Сведения об участниках информационного взаимодействия

2.2.1 Сведения о ЕПГУ

Единый портал государственных и муниципальных услуг предназначен для обеспечения гражданам доступа к Услугам посредством интерактивных форм.

ЕПГУ взаимодействует с Региональной витриной расписаний субъекта РФ для получения сведений о расписании, взаимодействует с пользователем ЕПГУ- отображает указанные сведения на интерактивной форме, а также взаимодействует непосредственно с ГИСЗ субъекта РФ с целью создания записи на приём.

Сведения мониторинга информационного взаимодействия ЕПГУ с Региональной витриной расписаний субъекта РФ и ГИСЗ субъекта РФ передаются в подсистему СИП ЕГИСЗ.

2.2.2 Сведения о НСУД и ПОДД СМЭВ 4

НСУД представляет собой систему, состоящую из взаимосвязанных элементов информационно-технологического, организационного, методологического, кадрового и нормативно-правового характера и обеспечивающую достижение целей и выполнение задач, обозначенных в Концепции Национальной системы управления данными, утвержденной распоряжением Правительства Российской Федерации от 3 июня 2019 года № 1189-р.

НСУД предназначена для управления информацией, содержащейся в информационных системах органов и организаций государственного сектора, а также в информационных ресурсах, созданных в целях реализации полномочий органов и организаций государственного сектора и для осуществления информационного обмена между Поставщиками и Потребителями данных, присоединившимися к НСУД.

В границах процесса предоставления услуг в сфере здравоохранения на ЕПГУ, Поставщиками данных являются:

- ГИСЗ субъекта РФ и иные ИС, подключенные к сервису ФЭР и предоставляющие сведения о расписании, а также сведения о фактах создания, изменения и отмены записей;

- ЕГИСЗ, предоставляющая сведения о результатах контроля качества расписания, а также предоставляющая сведения федеральных реестров и справочников (ФНСИ, ФРМО, ФРМР);

Потребителями данных являются:

- ЕПГУ и иные ИС, подключенные к сервису ФЭР и получающие сведения о расписании;
- ГИСЗ субъекта РФ и иные ИС, подключенные к сервису ФЭР и получающие сведения о результатах контроля качества расписания;
- ЕГИСЗ, получающая сведения о расписании, а также сведения о фактах создания, изменения и отмены записей.

Управление процессами информационного обмена между Поставщиками и Потребителями данных осуществляется средствами федеральной государственной информационной системы «Единая информационная платформа Национальной системы управления данными» (ФГИС «ЕИП НСУД»).

Для информационного обмена между Поставщиками и Потребителями данных используется среда взаимодействия НСУД, состоящая из:

- Подсистемы обеспечения доступа к данным СМЭВ (ПОДД СМЭВ, СМЭВ 4),
- Агентов ПОДД СМЭВ, устанавливаемых на стороне Поставщиков и Потребителей данных.

ПОДД СМЭВ (СМЭВ 4) – подсистема обеспечения доступа к данным СМЭВ, направленная на автоматизацию процесса передачи данных и уведомлений об изменении данных между организациями или органами власти, ответственными за формирование и ведение информационных ресурсов, зарегистрированных в НСУД.

ПОДД СМЭВ обеспечивает:

- предоставление для Потребителей механизмов унифицированного доступа к данным, размещённым у Поставщиков, с использованием языка запросов, основанного на стандарте языка SQL;
- предоставление для Потребителей возможности получения данных, связанных по ключевым полям или вычисляемому критерию и размещённых у разных Поставщиков;
- контроль полномочий Потребителей по доступу к данным, размещённым у Поставщиков, и поддержку гибкой настройки полномочий с точностью до атрибута и критерия отбора строк.

ПОДД СМЭВ поддерживает следующие способы организации информационного обмена:

1. обмен с использованием Регламентированных запросов;
2. подписка Потребителя
 - на Репликацию данных по событию об изменении данных;
 - на Репликацию данных по расписанию;
 - на уведомления об изменении данных.

Запросы и ответы направляются в ПОДД СМЭВ от имени организаций и подписываются электронными подписями органов власти (ЭП-ОВ). Идентификация и аутентификация информационных систем в ПОДД СМЭВ выполняется по сертификату ЭП-ОВ, загруженному в ПОДД СМЭВ при регистрации ИС Участников взаимодействия.

ПОДД СМЭВ состоит из следующих компонентов:

- Ядро ПОДД СМЭВ – центральный компонент ПОДД СМЭВ, размещенный в ИЭП;
- Агент ПОДД СМЭВ – компонент ПОДД СМЭВ, типовое программное обеспечение, устанавливаемое на стороне Поставщика и Получателя данных.

Агент ПОДД СМЭВ обеспечивает сопряжение Витрин данных, хранилищ реплик, а также ИС Поставщиков и Получателей данных с ПОДД СМЭВ.

2.2.3 Сведения о ГИСЗ субъекта РФ

2.2.3.1 Региональная витрина расписаний субъекта РФ

Региональная витрина расписаний субъекта РФ является частью ГИСЗ субъекта РФ и предназначена для загрузки, хранения и предоставления данных, необходимых для оказания Услуг на ЕПГУ.

Программное обеспечение «Витрина данных», необходимое для Региональной витрины расписаний субъекта РФ, предоставляется Минцифры России по соглашению с ОИВ субъекта РФ, являющимся оператором ГИСЗ субъекта РФ, на безвозмездной основе.

Региональная витрина расписаний субъекта РФ взаимодействует со следующими участниками информационного взаимодействия:

- ЕПГУ - с целью передачи сведений о расписании и ресурсах медицинских организаций, а также с целью обработки запросов на создание и отмену записей при оказании Услуг гражданам;

- Федеральная агрегированная витрина расписаний Минздрава России - с целью передачи сведений о расписании, ресурсах медицинских организаций и записях при получении Услуг гражданами; а также с целью получения результатов проверки качества (валидации) данных;
- ГИСЗ в сфере здравоохранения субъекта РФ - с целью получения и обновления сведений о расписании, записях и ресурсах медицинских организаций.

2.2.4 Сведения о ЕГИСЗ

2.2.4.1 Федеральная агрегированная витрина расписаний Минздрава России

Федеральная агрегированная витрина расписаний Минздрава России предназначена для:

- сбора сведений о расписании из Региональных витрин расписаний субъектов РФ;
- сбора сведений о результатах контроля качества расписания из Модуля управления бизнес-правилами;
- предоставления сведений о результатах контроля качества расписания в Региональные витрины расписаний субъектов РФ;
- предоставления сведений о расписании, а также сведений о результатах контроля качества расписания, в подсистему СИП ЕГИСЗ.

Федеральная агрегированная витрина расписаний Минздрава России взаимодействует со следующими участниками информационного взаимодействия:

- Региональная витрина расписаний субъекта РФ посредством использования НСУД в качестве средства обеспечения доступа к данным;
- Модуль управления бизнес-правилами подсистемы ФЭР ЕГИСЗ - с целью получения результатов проверки качества сведений о расписании;
- подсистема СИП ЕГИСЗ с целью передачи сведений о расписании, а также сведений о результатах контроля качества расписания.

2.2.4.2 Модуль управления бизнес-правилами подсистемы ФЭР ЕГИСЗ

Модуль управления бизнес-правилами подсистемы ФЭР ЕГИСЗ (далее - Модуль бизнес-правил) предназначен для контроля качества сведений о расписании посредством выполнения проверки данных, которые хранятся в Федеральной агрегированной витрине расписаний Минздрава России подсистемы ФЭР ЕГИСЗ.

В Модуле бизнес-правил выполняются различные проверки качества сведений о расписании, включая:

- проверка соответствия сведений о медицинской организации сведениям, размещённым в ФРМО,
- проверка соответствия сведений о медицинском работнике сведениям, размещённым в ФРМР,
- проверка соответствия сведений о медицинских услугах сведениям, размещённым в ФНСИ,

Более подробный перечень проверок, выполняемых на федеральном уровне приведен в Приложение Б настоящего документа.

Модуль бизнес-правил взаимодействует со следующими участниками информационного взаимодействия:

- Федеральная агрегированная витрина расписаний Минздрава России в части получения сведений о расписании, а также в части передачи сведений о результатах контроля качества сведений о расписании;
- подсистемы ФНСИ, ФРМО, ФРМР ЕГИСЗ части получения справочных сведений, необходимых для выполнения действия по контролю качества сведений о расписании.

2.2.4.3 Подсистема СИП ЕГИСЗ

Подсистема СИП ЕГИСЗ в части процесса управления потоками пациентов и ресурсами медицинских организаций предназначена для обеспечения возможности оперативного мониторинга доступности Услуг для граждан на ЕПГУ, контроля качества данных и принятия своевременных управленческих решений ФОИВ и ОИВ субъектов РФ.

Подсистема СИП ЕГИСЗ взаимодействует со следующими участниками информационного взаимодействия:

- Федеральная агрегированная витрина расписаний Минздрава России подсистемы «Федеральная электронная регистратура» ЕГИСЗ – с целью получения сведений о расписании, а также сведений о результатах контроля качества сведений о расписании;
- подсистемы ФНСИ, ФРМО, ФРМР ЕГИСЗ – с целью получения справочных сведений.

2.3 Обеспечение качества данных

Ответственность за качество данных, размещаемых на Региональной витрине расписаний субъекта РФ, несет соответствующий субъект РФ.

Методика оценки качества определена в документе «Методика сквозной оценки мониторинга качества оказания гражданам Российской Федерации услуги «Запись на прием к врачу» с использованием Единого портала государственных и муниципальных услуг (функций)». Указанный документ доступен на технологическом портале ЕГИСЗ по ссылке: <https://portal.egisz.rosminzdrav.ru/materials/3975>.

Просмотр и анализ показателей, характеризующих качество оказания услуги «Запись на прием к врачу» с использованием Единого портала государственных и муниципальных услуг (функций), а также просмотр пороговых значений осуществляется в единой аналитической системе центра управления качеством услуги.

Получение доступа в аналитическую систему центра управления качеством услуги осуществляется посредством подачи заявки ответственному лицу команды ЕПГУ.



Шаблон_заявки_на_д
оступ_VPN_v3_6_ЦУП.

Шаблон заявки

Основным каналом для взаимодействия между участниками является Портал федерального ситуационного центра электронного правительства (СЦ МЦ). Это интернет-портал <https://sc.digital.gov.ru/>, предназначенный для обеспечения коммуникации по вопросам получения и предоставления информационно-методической поддержки Участникам. Операторам СЦ является Минцифры России (соглашение о коммуникации представлено в документе «Методика оценки качества работы услуги «Запись на приём к врачу» с использованием Единого портала государственных и муниципальных услуг (функций)»).

Выполнять роль регионального заказчика со стороны СЦ МЦ может в том числе представитель, выполняющий функции Вендора МИС, который в рамках работ по внедрению и эксплуатации услуги запись на приём к врачу (10700) на технологию витрины данных.

Данный представитель может выполнять все работы, касаемые эксплуатации данной услуги со стороны заказчика, в том числе открывать, закрывать, изменять заявки, созданные региональным заказчиком в СЦ МЦ.

В случае необходимости Участник может самостоятельно завести Инцидент при возникновении аварийной ситуации на своей стороне или обнаружении нестабильной работы ПОДД путем направления запроса с использованием Личного кабинета СЦ <https://sc.digital.gov.ru/>.

В случае возникновения ошибок на ПО «Витрины данных» необходимо оформить запрос в СЦ следующего типа:

- соглашение/услуга – Инциденты ИС ИЭП;
- тип запроса – Запрос/инцидент;
- система ИЭП – ПО «Витрина данных».

В случае возникновения ошибок в процессе работы Агента ПОДД СМЭВ/ПОДД СМЭВ необходимо оформить запрос в СЦ следующего типа:

- соглашение/услуга – Инциденты ИС ИЭП;
- тип запроса – Запрос/инцидент;
- система ИЭП – ПОДД.

В случае возникновения ошибок в процессе работы ЕПГУ необходимо оформить запрос в СЦ следующего типа:

- соглашение/услуга – Инциденты ИС ИЭП;
- тип запроса – Запрос/инцидент;
- система ИЭП – ЕПГУ.

В теме запроса указать возникшую проблему. В описании подробно описать проблему. Любые виды ошибок со стороны регионального заказчика, а также их представителя, должны присылаться только с вложением документа "План реакции", с обязательным заполнением всех пунктов.

В сообщении должно быть предоставлено для анализа (за период исследуемого времени ошибки):

1. Конфигурация витрины данных (Лайт/Медиум/Стандарт);
2. Системные ресурсы сервера (CPU, RAM, HDD);

3. Утилизация сервера (результат команды top или htop);
4. Результат проверки сетевой связанности компонентов;
5. Результат проверки портов;
6. Название и версия компонентов витрины с ошибкой;
7. Конфигурационные файлы затронутых компонентов (application.yml и/или содержимое ENV компонента из Portainer);
8. Логи компонента с ошибкой;
9. Логи всех компонент витрины данных, агента ПОДД;
10. Описание проблемы, при которой возникает ошибка, с обязательным указанием даты и времени ошибки.

В период эксплуатации услуги Запись на приём к врачу должно обеспечиваться соблюдение уровня качества (SLA) по следующим показателям:

- наличие технических ошибок не более 1%;
- наличие организационных ошибок не более 5%;
- доступность за месяц, не менее 99% времени работы услуги на технологии витрин данных;
- периодичность обновления данных «Региональной витрины расписаний субъекта РФ»:
 - данные о расписании МО (сведения о свободных слотах, таблица slot, см. п.А.1.2) – не реже чем раз в 30 сек,
 - прочие данные (сведения о ресурсах МО, пациентах, направлениях и прочее, см. п. А.1.2) – не реже чем раз в 1 час;
- согласованность всех передаваемых данных между «Региональной витрины расписаний субъекта РФ» и РМИС не более 2%.

На основании показателей качества (SLA) услуги Запись на приём к врачу передаваемых из единой аналитической системы центра управления качеством услуги, каждый месяц формируется региональный рейтинг функционирования услуги, со следующими показателями:

- наличие развернутой и выведенной услуги Запись на приём к врачу на технологии витрин данных – 1 балл;
- отсутствие развернутой и выведенной услуги Запись на приём к врачу на технологии витрин данных – 0 баллов;

- наличие технических ошибок не более 1% – 1 балл;
- наличие технических ошибок более 1% – 0 баллов;
- наличие организационных ошибок не более 5% – 10 баллов;
- наличие организационных ошибок более 5% – 0 баллов;
- доступность за месяц, не менее 99% времени работы услуги на технологии витрин данных – 5 баллов;
- доступность за месяц, менее 99% времени работы услуги на технологии витрин данных – 0 баллов;
- наличие механизма согласованности всех передаваемых данных между «Региональной витрины расписаний субъекта РФ» и РМИС не более 2% – 10 баллов;
- отсутствие механизма согласованности всех передаваемых данных между «Региональной витрины расписаний субъекта РФ» и РМИС не более 2% – 0 баллов;
- наличие механизма периодичности обновления «Региональной витрины расписаний субъекта РФ» (данные о расписании МО – не реже чем раз в 30 сек, прочие данные – не реже чем раз в 1 час) – 10 баллов
- отсутствие механизма периодичности обновления «Региональной витрины расписаний субъекта РФ» (данные о расписании МО – не реже чем раз в 30 сек, прочие данные – не реже чем раз в 1 час) – 0 баллов.

3 ПОРЯДОК ОКАЗАНИЯ УСЛУГ И СЕРВИСОВ В СФЕРЕ ЗДРАВООХРАНЕНИЯ НА ЕПГУ

3.1 Запись на прием к врачу

3.1.1 Краткое описание

Услуга предназначен для записи пациента на приём к врачам первичного звена следующих специальностей:

1. врач-терапевт участковый;
2. врач-педиатр участковый;
3. врач общей практики (семейный врач);
4. врач-хирург;
5. врач-акушер-гинеколог;
6. врач-офтальмолог;
7. врач-оториноларинголог;
8. врач-стоматолог;
9. врач-стоматолог-терапевт;
10. врач-стоматолог детский;
11. врач-психиатр детский;
12. врач-фтизиатр;
13. врач-психиатр-нарколог;
14. врач-детский хирург.

В рамках услуги «Запись на приём к врачу» обеспечиваются следующие функциональные возможности:

- подача заявки на запись пациента на прием к врачам первичного звена в медицинскую организацию;
- возможность выбора в качестве пациента себя, своего ребенка или другого человека;
- автоматическое определение региона получения услуги на основании сведений о прикреплении пациента;
- автоматическое определение МО получения услуги на основании сведений о прикреплении пациента;

- выбор для записи МО, подведомственной ФОИВ;
- выбор доступной для записи специальности врача;
- выбор доступного для записи времени приёма;
- на основании заявки на запись создание записи в выбранную МО, с учетом принятых в субъекте РФ правил предоставления первичной медико-санитарной амбулаторной помощи;
- обновления статуса записи со стороны ГИСЗ субъекта РФ;
- отмена записи по инициативе пользователя ЕПГУ;
- отмена записи также по инициативе МО;
- подача жалобы на отсутствие доступных МО, специалистов или слотов времени.

3.1.2 Клиентский путь «Запись на прием к врачу»

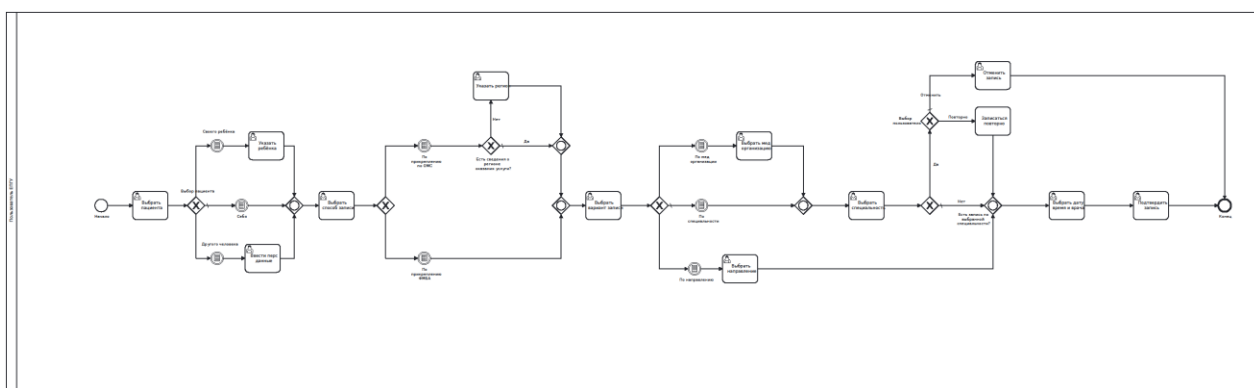


Рисунок 3 – Клиентский путь «Запись на прием к врачу» (объект PDF)

3.1.3 Сценарий «Запись на прием к врачу»

Цель	Записать пациента на приём к врачу
Участники	1. Пользователь ЕПГУ (Пользователь) 2. ЕПГУ 3. СМЭВ4 4. Федеральная витрина расписаний ЕГИСЗ (Федеральная витрина) 5. Региональная витрина расписаний (Региональная витрина) 6. ГИСЗ субъекта РФ (ГИСЗ)

Входная информация	1. Сведения о расписании 2. Сведения о пациенте
Выходная информация	1. Сведения о записи на приём к врачу 2. Сведения о расписании (обновлённые)

Основной сценарий. Записать пациента на приём к врачу выбранной специальности	
Предусловия	1. Пользователь авторизован на ЕПГУ 2. Пользователю доступна услуга «Запись на приём к врачу» 3. Информационная связность ЕПГУ и Региональной витрины функционируют в штатном режиме эксплуатации
Триггер	Прецедент начинается, когда Пользователь выбирает на ЕПГУ услугу «Запись на приём к врачу»
Последовательность (поток) действий	1. ЕПГУ запрашивает у пользователя данные о пациенте (записать себя, своего ребёнка или другого человека) 2. Пользователь указывает пациента: себя, своего ребёнка или другого человека – в случае выбора «себя» ЕПГУ использует персональные данные Пользователя как пациента – в случае выбора «своего ребёнка» Пользователь выбирает ребёнка из списка зарегистрированных на ЕПГУ своих детей или вводит вручную сведения о ребёнке, ЕПГУ использует персональные данные ребёнка как пациента – в случае выбора «другого человека» Пользователь вводит персональные данные пациента в составе: СНИЛС, ФИО, пол, дата рождения, полис ОМС; 3. Пользователь указывает вариант взаимодействия «По прикреплению по ОМС» 4. Пользователь вариант взаимодействия «Запись по специальности» 5. ЕПГУ выполняет следующие действия:

	– определяет регион оказания услуги (на основании сведений о пациенте или запросом к пользователю) – посредством СМЭВ4 считывает расписание с Региональной витрины; 6. ЕПГУ выводит пользователю список доступных для записи специальностей врачей (например, «врач-невролог») 7. Пользователь выбирает специальность врача 8. ЕПГУ проверяет наличие у пациента записей на выбранную специальность и далее, в случае отсутствия у пациента записей на выбранную специальность выводит пользователю список слотов (дата, время, врач) 9. Пользователь выбирает слот 10. ЕПГУ посредством СМЭВ4 передаёт в ГИСЗ информацию о запросе на создании записи в выбранный пользователем слот 11. ГИСЗ получает запрос на создании записи в слот, проверяет возможность создания записи и далее, в случае успеха – создаёт запись – передаёт на ЕПГУ сведения об успешном создании записи – вносит изменения в расписание – передаёт обновлённые сведения о расписании в Региональную витрину 12. Региональная витрина получает обновлённые сведения о расписании и далее – обновляет хранимые данные с учётом новых сведений – передаёт посредством СМЭВ4 обновлённые сведения о расписании в Федеральную витрину
Результат (постусловия)	Создана запись на приём к врачу

Альтернативный сценарий. Отменить запись пациента	
Предусловия	1. Пользователь авторизован на ЕПГУ 2. Пользователю доступна услуга «Запись на приём к врачу» 3. Информационная связность ЕПГУ и Региональной витрины функционируют в штатном режиме эксплуатации
Триггер	Прецедент начинается на шаге 8 основного потока
Последовательность (поток) действий	1. ЕПГУ проверяет наличие у пациента записей на выбранную специальность и далее, в случае выявления у пациента записей на выбранную специальность выводит пользователю запрос на отмену записи 2. Пользователь подтверждает отмену записи 3. ЕПГУ посредством СМЭВ4 передаёт в ГИСЗ информацию о запросе на отмену записи 4. ГИСЗ получает запрос на отмену записи и далее – отменяет запись – вносит изменения в расписание – передаёт обновлённые сведения о расписании в Региональную витрину 5. Региональная витрина получает обновлённые сведения о расписании и далее – обновляет хранимые данные с учётом новых сведений – передаёт посредством СМЭВ4 обновлённые сведения о расписании в Федеральную витрину
Результат (постусловия)	Отменена запись на приём к врачу

3.2 Запись на прием по направлению

3.2.1 Краткое описание

Услуга обеспечивает возможность записи пациента на приём к врачу по ранее созданному направлению.

В рамках услуги «Запись на прием по направлению» обеспечиваются следующие функциональные возможности:

- подача заявки на запись пациента на прием к врачам по ранее созданным направлениям в МО;
- возможность выбора в качестве пациента себя или своего ребенка;
- просмотр списка имеющихся направлений;
- автоматическое определение МО и специальности врача для получения услуги в соответствии со сведениями, содержащимися в направлении;
- выбор врача (кабинета, оборудования) для получения медицинской услуги, указанной в направлении, с учетом принятых в субъекте РФ правил предоставления первичной медико-санитарной амбулаторной помощи;
- выбор доступного для записи времени приёма;
- на основании заявки на запись создание записи в выбранную МО;
- отображение в ленте ЕПГУ факта записи, включая статус записи;
- обновления статуса со стороны ГИСЗ субъекта РФ;
- отмена записи по инициативе пользователя ЕПГУ;
- отмена записи также по инициативе МО;
- подача жалобы на отсутствие доступных МО, специалистов или слотов времени.

3.2.2 Клиентский путь «Запись на прием по направлению»

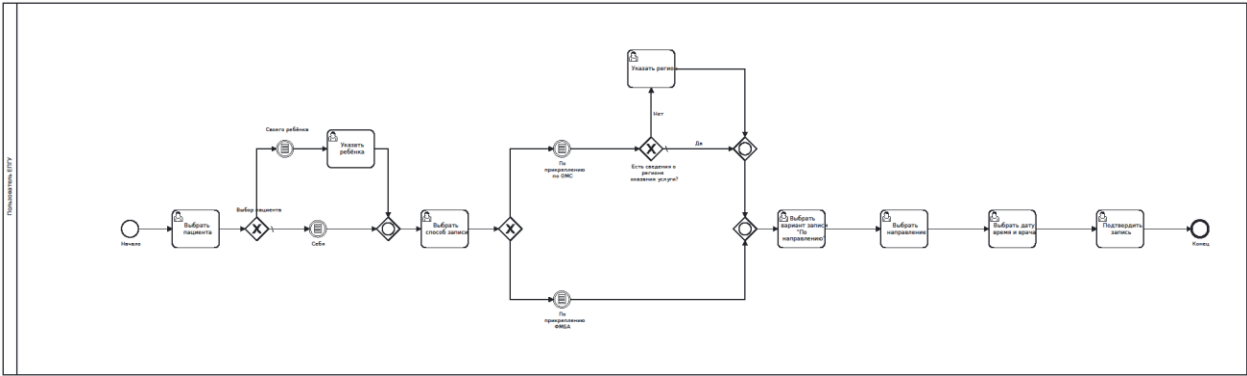


Рисунок 4 – Клиентский путь «Запись на прием по направлению» (объект PDF)

3.2.3 Сценарий «Запись на прием по направлению»

Цель	Записать пациента на приём к врачу в соответствии с направлением
Участники	1. Пользователь ЕПГУ (Пользователь) 2. ЕПГУ 3. СМЭВ4 4. Федеральная витрина расписаний ЕГИСЗ (Федеральная витрина) 5. Региональная витрина расписаний (Региональная витрина) 6. ГИСЗ субъекта РФ (ГИСЗ)
Входная информация	1. Сведения о расписании 2. Сведения о пациенте 3. Сведения о направлении
Выходная информация	1. Сведения о записи на приём к врачу 2. Сведения о расписании (обновлённые) 3. Сведения о направлении (обновлённые)

Основной сценарий. Записать пациента на приём к врачу по направлению	
Предусловия	1. Пользователь авторизован на ЕПГУ 2. Пользователю доступна услуга «Запись на приём к врачу»

	3. Информационная связность ЕПГУ и Региональной витрины функционируют в штатном режиме эксплуатации
Триггер	Прецедент начинается, когда Пользователь выбирает на ЕПГУ услугу «Запись на приём к врачу»
Последовательность (поток) действий	1. ЕПГУ запрашивает у пользователя данные о пациенте (записать себя, своего ребёнка или другого человека) 2. Пользователь указывает пациента: себя или своего ребёнка – в случае выбора «себя» ЕПГУ использует персональные данные Пользователя как пациента – в случае выбора «своего ребёнка» Пользователь выбирает ребёнка из списка зарегистрированных на ЕПГУ своих детей или вводит ручную сведения о ребёнке, ЕПГУ использует персональные данные ребёнка как пациента; 3. Пользователь указывает вариант взаимодействия «По прикреплению по ОМС» 4. Пользователь вариант взаимодействия «Запись по направлению» 5. ЕПГУ выполняет следующие действия: – определяет регион оказания услуги – посредством СМЭВ4 считывает расписание с Региональной витрины; 6. ЕПГУ выводит пользователю список направлений 7. Пользователь выбирает направление из списка 8. ЕПГУ выводит пользователю список слотов (дата, время, врач) 9. Пользователь выбирает слот 10. ЕПГУ посредством СМЭВ4 передаёт в ГИСЗ информацию о запросе на создании записи в выбранный пользователем слот

	11. ГИСЗ получает запрос на создании записи в слот, проверяет возможность создания записи и далее, в случае успеха – создаёт запись – передаёт на ЕПГУ сведения об успешном создании записи – вносит изменения в расписание – передаёт обновлённые сведения о расписании в Региональную витрину 12. Региональная витрина получает обновлённые сведения о расписании и далее – обновляет хранимые данные с учётом новых сведений – передаёт посредством СМЭВ4 обновлённые сведения о расписании в Федеральную витрину
Результат (постусловия)	Для пациента создана запись на приём к врачу

3.3 Запись на прием к врачам, осуществляющим диспансерное наблюдение

3.3.1 Краткое описание

Сервис предназначен для записи пациента на приём к врачу, осуществляющему диспансерное наблюдение.

В рамках сервиса «Запись на прием к врачам, осуществляющим диспансерное наблюдение» обеспечиваются следующие функциональные возможности:

- подача заявки на запись пациента на плановый осмотр к врачу, осуществляющему диспансерное наблюдение гражданина;
- возможность выбора в качестве пациента себя или своего ребенка;
- автоматическое определение региона получения услуги;
- просмотр списка лечащих врачей, у которых пациент состоит на диспансерном учете;
- выбор доступного для записи времени приёма;

- на основании заявки на запись создание записи в выбранную медицинскую организацию;
- отображение в ленте ЕПГУ факта записи, включая статус записи;
- обновление статуса со стороны ГИСЗ субъекта РФ;
- отмена записи по инициативе пользователя ЕПГУ;
- отмена записи также по инициативе МО;
- подача жалобы на отсутствие доступных МО, специалистов или слотов времени.

3.3.2 Клиентский путь «Запись на прием к врачам, осуществляющим диспансерное наблюдение»

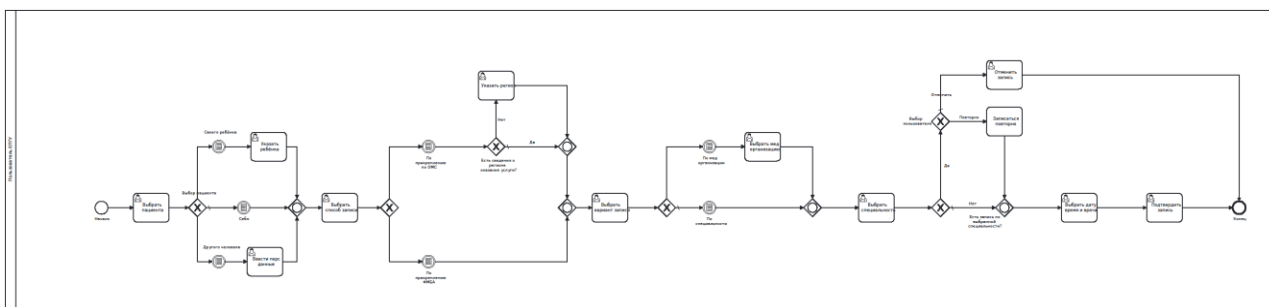


Рисунок 5 – Клиентский путь «Запись на прием к врачам, осуществляющим диспансерное наблюдение» (объект PDF)

3.3.3 Сценарий «Запись на прием к врачам, осуществляющим диспансерное наблюдение»

Цель	Записать пациента на приём к врачу, осуществляющему диспансерное наблюдение
Участники	1. Пользователь ЕПГУ (Пользователь) 2. ЕПГУ 3. СМЭВ4 4. Федеральная витрина расписаний ЕГИСЗ (Федеральная витрина) 5. Региональная витрина расписаний (Региональная витрина) 6. ГИСЗ субъекта РФ (ГИСЗ)

Входная информация	1. Сведения о расписании 2. Сведения о пациенте 3. Сведения о диспансерном наблюдении пациента
Выходная информация	1. Сведения о записи на приём к врачу 2. Сведения о расписании (обновлённые)

Основной сценарий. Записать пациента на приём к врачу, осуществляющим диспансерное наблюдение

Предусловия	1. Пользователь авторизован на ЕПГУ 2. Пользователю доступна услуга «Запись на прием к врачам, осуществляющим диспансерное наблюдение» 3. Информационная связность ЕПГУ и Региональной витрины функционируют в штатном режиме эксплуатации
Триггер	Прецедент начинается, когда Пользователь выбирает на ЕПГУ услугу «Запись на прием к врачам, осуществляющим диспансерное наблюдение»
Последовательность (поток) действий	1. ЕПГУ запрашивает у пользователя данные о пациенте (записать себя, своего ребёнка или другого человека) 2. Пользователь указывает пациента: себя, своего ребёнка или другого человека – в случае выбора «себя» ЕПГУ использует персональные данные Пользователя как пациента – в случае выбора «своего ребёнка» Пользователь выбирает ребёнка из списка зарегистрированных на ЕПГУ своих детей или вводит ручную сведения о ребёнке, ЕПГУ использует персональные данные ребёнка как пациента 3. Пользователь указывает вариант взаимодействия «По прикреплению по ОМС» 4. Пользователь вариант взаимодействия «Запись по специальности» 5. ЕПГУ выполняет следующие действия:

	– определяет регион оказания услуги (на основании сведений о пациенте или запросом к пользователю) – посредством СМЭВ4 считывает расписание с Региональной витрины; 6. ЕПГУ выводит пользователю список доступных для записи специальностей врачей, осуществляющих диспансерное наблюдение 7. Пользователь выбирает специальность врача 8. ЕПГУ проверяет наличие у пациента записей на выбранную специальность и далее, в случае отсутствия у пациента записей на выбранную специальность выводит пользователю список слотов (дата, время, врач) 9. Пользователь выбирает слот 10. ЕПГУ посредством СМЭВ4 передаёт в ГИСЗ информацию о запросе на создании записи в выбранный пользователем слот 11. ГИСЗ получает запрос на создании записи в слот, проверяет возможность создания записи и далее, в случае успеха – создаёт запись – передаёт на ЕПГУ сведения об успешном создании записи – вносит изменения в расписание – передаёт обновлённые сведения о расписании в Региональную витрину 12. Региональная витрина получает обновлённые сведения о расписании и далее – обновляет хранимые данные с учётом новых сведений – передаёт посредством СМЭВ4 обновлённые сведения о расписании в Федеральную витрину
Результат (постусловия)	Для пациента создана запись на приём к врачу

3.4 Запись на вакцинацию

3.4.1 Краткое описание

Сервис предназначен для записи пациента на приём к врачу перед вакцинацией.

В рамках сервиса «Запись на вакцинацию» обеспечиваются следующие функциональные возможности:

- подача заявки на запись пациента в кабинет на прием перед вакцинацией в медицинскую организацию по месту фактического нахождения гражданина, вне зависимости от прикрепления к медицинской организации;
- возможность выбора в качестве пациента себя, своего ребенка или другого человека;
- автоматическое определение региона получения услуги;
- выбор для записи из доступных мероприятий по вакцинопрофилактике по национальному календарю вакцинации, а также по эпидемиологическим показателям;
- выбор для записи медицинской организации и кабинета вакцинации из списка доступных для данного пациента в выбранном регионе получения услуги, с учетом наличия доступных слотов записи;
- выбор доступного для записи времени приёма;
- на основании заявки на запись создание записи в выбранную медицинскую организацию;
- отображение в ленте ЕПГУ факта записи, включая статус записи;
- обновление статуса со стороны ГИСЗ субъекта РФ;
- отмена записи по инициативе пользователя ЕПГУ;
- отмена записи также по инициативе МО;
- подача жалобы на отсутствие доступных МО, специалистов или слотов времени.

3.4.2 Клиентский путь «Запись на вакцинацию»

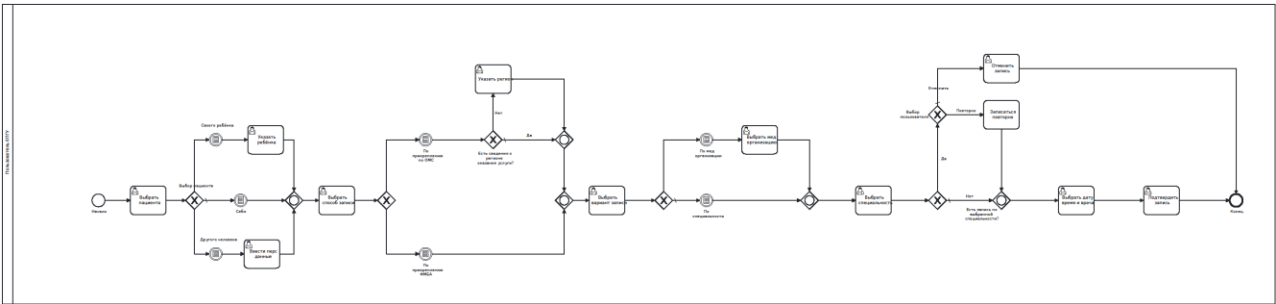


Рисунок 6 – Клиентский путь «Запись на вакцинацию» (объект PDF)

3.4.3 Сценарий «Запись на вакцинацию»

Цель	Записать пациента на приём к врачу перед вакцинацией
Участники	1. Пользователь ЕПГУ (Пользователь) 2. ЕПГУ 3. СМЭВ4 4. Федеральная витрина расписаний ЕГИСЗ (Федеральная витрина) 5. Региональная витрина расписаний (Региональная витрина) 6. ГИСЗ субъекта РФ (ГИСЗ)
Входная информация	1. Сведения о расписании 2. Сведения о пациенте
Выходная информация	1. Сведения о записи на приём к врачу 2. Сведения о расписании (обновлённые)

Основной сценарий. Записать пациента на приём к врачу перед вакцинацией	
Предусловия	1. Пользователь авторизован на ЕПГУ 2. Пользователю доступна услуга «Запись на вакцинацию» 3. Информационная связность ЕПГУ и Региональной витрины функционируют в штатном режиме эксплуатации
Триггер	Прецедент начинается, когда Пользователь выбирает на ЕПГУ услугу «Запись на вакцинацию»

Последовательность (поток) действий	1. ЕПГУ запрашивает у пользователя данные о пациенте (записать себя, своего ребёнка или другого человека) 2. Пользователь указывает пациента: себя, своего ребёнка или другого человека – в случае выбора «себя» ЕПГУ использует персональные данные Пользователя как пациента – в случае выбора «своего ребёнка» Пользователь выбирает ребёнка из списка зарегистрированных на ЕПГУ своих детей или вводит вручную сведения о ребёнке, ЕПГУ использует персональные данные ребёнка как пациента 3. Пользователь указывает вариант взаимодействия «По прикреплению по ОМС» 4. Пользователь указывает вариант взаимодействия «Запись по специальности» 5. ЕПГУ выполняет следующие действия: – определяет регион оказания услуги (на основании сведений о пациенте или запросом к пользователю) – посредством СМЭВ4 считывает расписание с Региональной витрины; 6. ЕПГУ выводит пользователю список доступных для записи специальностей врачей, осуществляющих осмотр перед вакцинацией 7. Пользователь выбирает специальность врача 8. ЕПГУ проверяет наличие у пациента записей на выбранную специальность и далее, в случае отсутствия у пациента записей на выбранную специальность выводит пользователю список слотов (дата, время, врач) 9. Пользователь выбирает слот 10. ЕПГУ посредством СМЭВ4 передаёт в ГИСЗ информацию о запросе на создании записи в выбранный пользователем слот
--	---

	11. ГИСЗ получает запрос на создании записи в слот, проверяет возможность создания записи и далее, в случае успеха – создаёт запись – передаёт на ЕПГУ сведения об успешном создании записи – вносит изменения в расписание – передаёт обновлённые сведения о расписании в Региональную витрину 12. Региональная витрина получает обновлённые сведения о расписании и далее – обновляет хранимые данные с учётом новых сведений – передаёт посредством СМЭВ4 обновлённые сведения о расписании в Федеральную витрину
Результат (постусловия)	Для пациента создана запись на приём к врачу

3.5 Запись на прием на углубленную диспансеризацию для лиц, переболевших COVID-19

3.5.1 Краткое описание

Сервис предназначен для записи пациента на приём к врачу, осуществляющему консультацию перед углубленной диспансеризации лиц, переболевших COVID-19.

В рамках сервиса «Запись на прием на углубленную диспансеризацию для лиц, переболевших COVID-19» обеспечиваются следующие функциональные возможности:

- подача заявки пациента на запись на углубленную диспансеризацию в медицинскую организацию по прикреплению гражданина;
- возможность выбора в качестве пациента себя;
- отображение в ленте ЕПГУ факта записи, включая статус записи;
- автоматическое определение региона получения услуги;
- выбор для записи медицинской организации из списка доступных для данного пациента в регионе прикрепления;

- выбор доступного для записи времени приёма;
- на основании заявки на запись создание записи в выбранную медицинскую организацию;
- обновление статуса со стороны ГИСЗ субъекта РФ;
- отмена записи по инициативе пользователя ЕПГУ;
- отмена записи также по инициативе МО;
- подача жалобы на отсутствие доступных МО, специалистов или слотов времени.

3.5.2 Клиентский путь «Запись на приём на углубленную диспансеризацию»

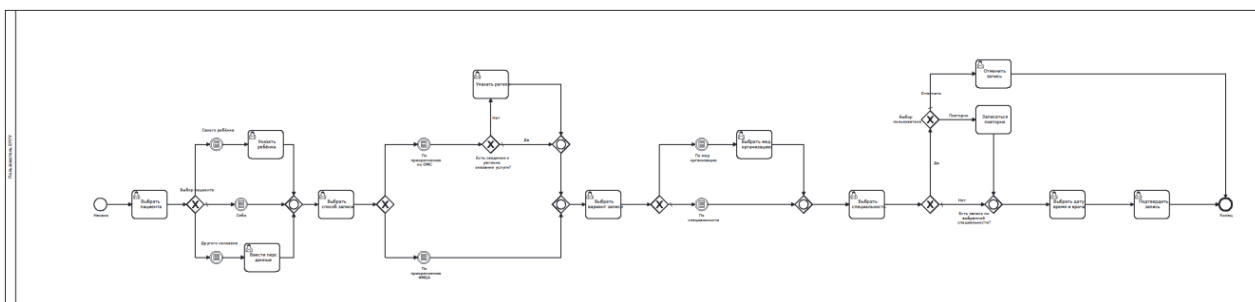


Рисунок 7 – Клиентский путь «Запись на приём на углубленную диспансеризацию» (объект PDF)

3.5.3 Сценарий «Запись на приём на углубленную диспансеризацию»

Цель	Записать пациента на приём к врачу перед углублённой диспансеризацией
Участники	1. Пользователь ЕПГУ (Пользователь) 2. ЕПГУ 3. СМЭВ4 4. Федеральная витрина расписаний ЕГИСЗ (Федеральная витрина) 5. Региональная витрина расписаний (Региональная витрина) 6. ГИСЗ субъекта РФ (ГИСЗ)

Входная информация	1. Сведения о расписании 2. Сведения о пациенте
Выходная информация	1. Сведения о записи на приём к врачу 2. Сведения о расписании (обновлённые)

Основной поток. Записать пациента на приём к врачу, осуществляющему консультацию перед углубленной диспансеризации лиц, переболевших COVID-19

Предусловия	1. Пользователь авторизован на ЕПГУ 2. Пользователю доступна услуга «Запись на приём на углубленную диспансеризацию» 3. Информационная связность ЕПГУ и Региональной витрины функционируют в штатном режиме эксплуатации
Триггер	Прецедент начинается, когда Пользователь выбирает на ЕПГУ услугу «Запись на приём на углубленную диспансеризацию»
Последовательность (поток) действий	1. ЕПГУ запрашивает у пользователя данные о пациенте (записать себя, своего ребёнка или другого человека) 2. Пользователь указывает вариант взаимодействия «По прикреплению по ОМС» 3. Пользователь вариант взаимодействия «Запись по специальности» 4. ЕПГУ выполняет следующие действия: – определяет регион оказания услуги (на основании сведений о пациенте или запросом к пользователю) – посредством СМЭВ4 считывает расписание с Региональной витрины; 5. ЕПГУ выводит пользователю список доступных для записи специальностей врачей, осуществляющих консультацию перед углубленной диспансеризации лиц, переболевших COVID-19 6. Пользователь выбирает специальность врача

	7. ЕПГУ проверяет наличие у пациента записей на выбранную специальность и далее, в случае отсутствия у пациента записей на выбранную специальность выводит пользователю список слотов (дата, время, врач) 8. Пользователь выбирает слот 9. ЕПГУ посредством СМЭВ4 передаёт в ГИСЗ информацию о запросе на создании записи в выбранный пользователем слот 10. ГИСЗ получает запрос на создании записи в слот, проверяет возможность создания записи и далее, в случае успеха – создаёт запись – передаёт на ЕПГУ сведения об успешном создании записи – вносит изменения в расписание – передаёт обновлённые сведения о расписании в Региональную витрину 11. Региональная витрина получает обновлённые сведения о расписании и далее – обновляет хранимые данные с учётом новых сведений – передаёт посредством СМЭВ4 обновлённые сведения о расписании в Федеральную витрину
Результат (постусловия)	Для пациента создана запись на приём к врачу

3.6 Просмотр и отмена записей, совершенных через иные источники (не ЕПГУ)

3.6.1 Краткое описание

Сервис предназначен для просмотра и отмены записей на приём, совершенных через иные источники (не ЕПГУ).

3.6.3 Сценарий процесса «Изменение и отмена записей, совершенных через иные источники»

Цель	Отменить запись (бронирование) пациента
Участники	1. Пользователь ЕПГУ (Пользователь) 2. ЕПГУ 3. СМЭВ4 4. Федеральная витрина расписаний ЕГИСЗ (Федеральная витрина) 5. Региональная витрина расписаний (Региональная витрина) 6. ГИСЗ субъекта РФ (ГИСЗ) 7. Подсистема СИП ЕГИСЗ (СИП) 8. Подсистема обратной связи ЕПГУ (ПОС)
Входная информация	1. Сведения о расписании 2. Сведения о пациенте 3. Сведения о записи
Выходная информация	1. Сведения о записи на приём к врачу (обновлённые) 2. Сведения о расписании (обновлённые)

Основной сценарий. Отмена записи пациента

Предусловия	1. Пользователь авторизован на ЕПГУ 2. Пользователю доступен сервис по просмотру сведений о записях на приём к врачу 3. Информационная связность ЕПГУ и Региональной витрины функционируют в штатном режиме эксплуатации
Триггер	Прецедент начинается, когда Пользователь на ЕПГУ переходит на страницу просмотра всех бронирований
Последовательность (поток) действий	13. ЕПГУ запрашивает у пользователя данные о пациенте (записать себя, своего ребёнка или другого человека) 14. Пользователь указывает пациента: себя, своего ребёнка или другого человека

	– в случае выбора «себя» ЕПГУ использует персональные данные Пользователя как пациента – в случае выбора «своего ребёнка» Пользователь выбирает ребёнка из списка зарегистрированных на ЕПГУ своих детей или вводит вручную сведения о ребёнке, ЕПГУ использует персональные данные ребёнка как пациента – в случае выбора «другого человека» Пользователь вводит персональные данные пациента в составе: СНИЛС, ФИО, пол, дата рождения, полис ОМС; 15. Пользователь указывает вариант взаимодействия «По прикреплению по ОМС» 16. Пользователь вариант взаимодействия «Запись по специальности» 17. ЕПГУ выполняет следующие действия: – определяет регион оказания услуги (на основании сведений о пациенте или запросом к пользователю) – посредством СМЭВ4 считывает расписание с Региональной витрины; 18. ЕПГУ выводит пользователю список доступных для записи специальностей врачей (например, «врач-невролог») 19. Пользователь выбирает специальность врача 20. ЕПГУ проверяет наличие у пациента записей на выбранную специальность и далее, в случае выявления у пациента записей на выбранную специальность выводит пользователю запрос на отмену записи 21. Пользователь подтверждает отмену записи 22. ЕПГУ посредством СМЭВ4 передаёт в ГИСЗ информацию о запросе на отмену записи 23. ГИСЗ получает запрос на отмену записи и далее – отменяет запись – вносит изменения в расписание
--	--

	– передаёт обновлённые сведения о расписании в Региональную витрину 24. Региональная витрина получает обновлённые сведения о расписании и далее – обновляет хранимые данные с учётом новых сведений – передаёт посредством СМЭВ4 обновлённые сведения о расписании в Федеральную витрину
Результат (постусловия)	Запись на приём к врачу отменена

3.7 Вызов врача на дом

3.7.1 Краткое описание

Услуга предназначена для подачи на заявки на вызов врача на дом для пациента.

В рамках услуги «Вызов врача на дом» обеспечиваются следующие функциональные возможности:

- подача заявки на вызов врача на дом в медицинскую организацию по месту фактического нахождения пациента;
- возможность выбора в качестве пациента себя, своего ребенка или другого человека;
- проверка доступности для пациента услуги вызова врача на дом по месту фактического нахождения гражданина;
- указание подъезда, этажа, кода домофона и другой дополнительной информации для получения доступа к помещению, в котором находится пациент;
- указание жалоб пациента и оснований для вызова врача на дом;
- получение данных о медицинской организации, в которую был направлен вызов, этапах обработки и подтверждения вызова, интервале времени прибытия врача, а также ФИО врача, его специальности;
- на основании заявки на запись создание записи для вызова врача на дом;
- отображение в ленте ЕПГУ факта записи, включая статус записи;
- обновление статуса со стороны ГИСЗ субъекта РФ;
- отмена записи по инициативе пользователя ЕПГУ;

- отмена записи также по инициативе МО;
- подача жалобы на отсутствие доступных МО, специалистов или слотов времени, а также подача жалобы «врач не пришел»;
- возможность приоритетного оказания услуги для отдельных групп пациентов, таких как инвалиды первой группы и дети-инвалиды, в соответствии с пунктом 4 раздела I протокола совещания у Заместителя Председателя Правительства Российской Федерации Т. А. Голиковой от 05.05.2022 № ТГ-П12-44пр.

3.7.2 Клиентский путь «Вызов врача на дом»

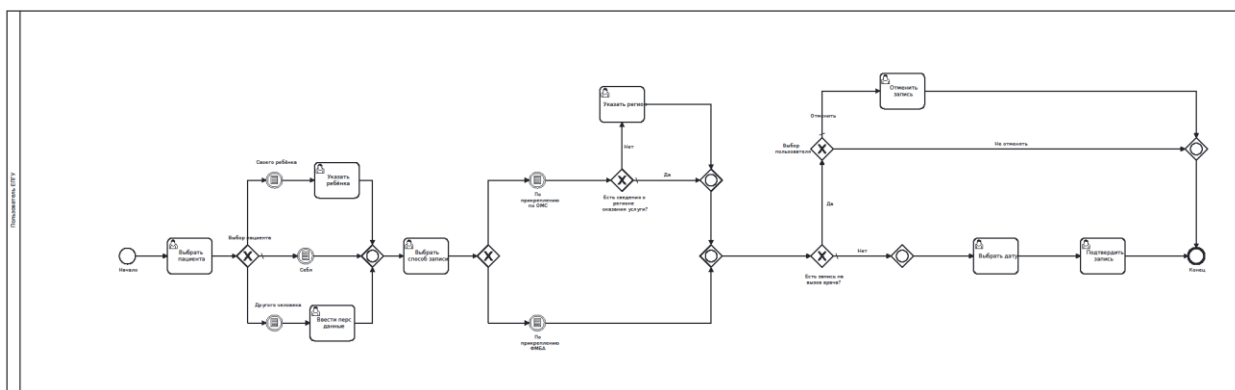


Рисунок 9 – Клиентский путь «Вызов врача на дом» (объект PDF)

3.7.3 Сценарий «Вызов врача на дом»

Цель	Создать запись для вызова врача на дом
Участники	1. Гражданин РФ - пользователь ЕПГУ (Пользователь) 2. ЕПГУ 3. СМЭВ4 4. Федеральная витрина расписаний ЕГИСЗ (Федеральная витрина) 5. Региональная витрина расписаний (Региональная витрина) 6. ГИСЗ субъекта РФ (ГИСЗ)
Входная информация	1. Сведения о расписании 2. Сведения о пациенте

Выходная информация	1. Сведения о записи на приём к врачу (вызове на дом) 2. Сведения о расписании (обновлённые)
---------------------	---

Основной сценарий. Вызов врача на дом для пациента	
Предусловия	1. Пользователь авторизован на ЕПГУ 2. Пользователю доступна услуга «Вызов врача на дом» 3. Информационная связность ЕПГУ и Региональной витрины функционируют в штатном режиме эксплуатации
Триггер	Прецедент начинается, когда Пользователь на ЕПГУ выбирает на ЕПГУ услугу «Вызов врача на дом»
Последовательность (поток) действий	1. ЕПГУ запрашивает у пользователя данные о пациенте (записать себя, своего ребёнка или другого человека) 2. Пользователь указывает пациента: себя, своего ребёнка или другого человека – в случае выбора «себя» ЕПГУ использует персональные данные Пользователя как пациента – в случае выбора «своего ребёнка» Пользователь выбирает ребёнка из списка зарегистрированных на ЕПГУ своих детей или вводит вручную сведения о ребёнке, ЕПГУ использует персональные данные ребёнка как пациента – в случае выбора «другого человека» Пользователь вводит персональные данные пациента в составе: СНИЛС, ФИО, пол, дата рождения, полис ОМС; 3. Пользователь указывает вариант взаимодействия «По прикреплению по ОМС» 4. Пользователь вариант взаимодействия «Запись по специальности» 5. ЕПГУ выполняет следующие действия: – определяет регион оказания услуги (на основании сведений о пациенте или запросом к пользователю)

	– посредством СМЭВ4 считывает расписание с Региональной витрины; 6. ЕПГУ выводит пользователю список доступных для записи специальностей врачей для вызова на дом 7. Пользователь выбирает специальность врача 8. ЕПГУ проверяет наличие у пациента записей на выбранную специальность и далее, в случае отсутствия у пациента записей на выбранную специальность выводит пользователю список слотов (дата, время, врач) 9. Пользователь выбирает слот 10. ЕПГУ посредством СМЭВ4 передаёт в ГИСЗ информацию о запросе на создании записи в выбранный пользователем слот 11. ГИСЗ получает запрос на создании записи в слот, проверяет возможность создания записи и далее, в случае успеха – создаёт запись – передаёт на ЕПГУ сведения об успешном создании записи – вносит изменения в расписание – передаёт обновлённые сведения о расписании в Региональную витрину 12. Региональная витрина получает обновлённые сведения о расписании и далее – обновляет хранимые данные с учётом новых сведений – передаёт посредством СМЭВ4 обновлённые сведения о расписании в Федеральную витрину
Результат (постусловия)	Создана запись о вызове врача на дом

3.8 Телемедицинская консультация

3.8.1 Краткое описание

Сервис предназначен для дистанционного взаимодействия между медицинским работником и пациентом посредством телемедицинских технологий по видам работ (услуг), указанным в лицензии на осуществление медицинской деятельности.

В рамках сервиса «Телемедицинская консультация» обеспечиваются следующие функциональные возможности:

- подача заявки на запись Пациента (себя) на телемедицинскую консультацию;
- автоматическое определение региона получения услуги на основании сведений о прикреплении пациента;
- автоматическое определение МО получения услуги на основании сведений о прикреплении пациента;
- выбор для записи МО и врача на основании данных направления и открытого случая;
- выбор для записи МО, подведомственной ФОИВ;
- выбор доступной для записи специальности врача;
- выбор доступного для записи времени приёма;
- обновления статуса записи со стороны ГИСЗ субъекта РФ;
- отмена записи по инициативе пользователя ЕПГУ;
- отмена записи по инициативе МО;
- предоставление ссылки для подключения к ТМК;
- предоставление видео-аудио-конференцсвязи между врачом консультирующей МО и пациентом;
- управление статусом телемедицинской консультации.

3.8.2 Клиентский путь «Телемедицинская консультация»

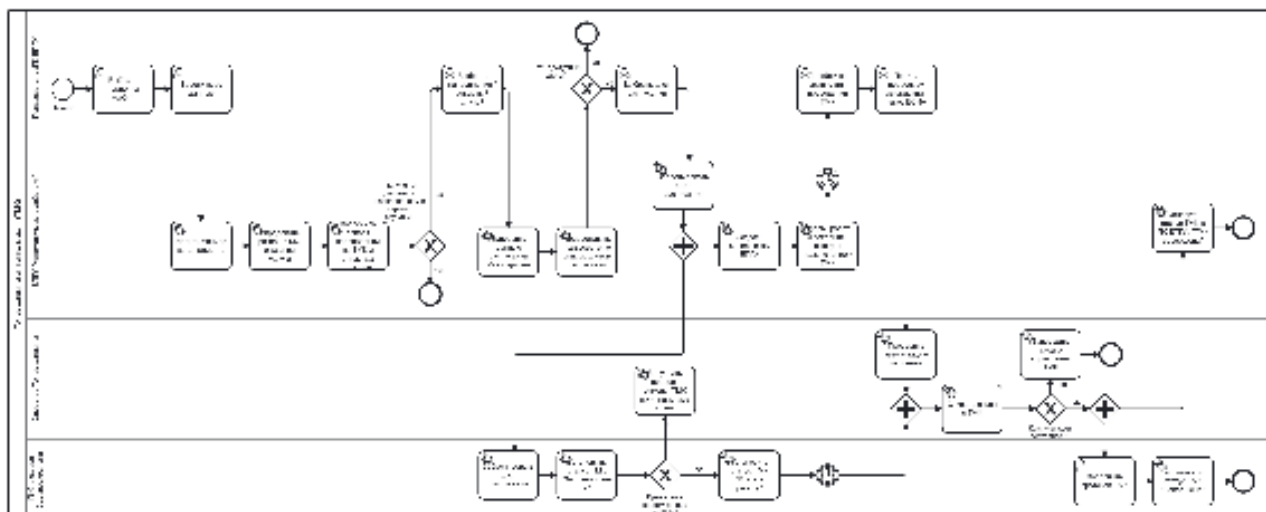


Рисунок 10 – Клиентский путь «Телемедицинская консультация»

3.8.3 Сценарий «Телемедицинская консультация»

Цель	Записать пациента и провести телемедицинскую консультацию
Участники	1. Гражданин РФ - пользователь ЕПГУ (Пользователь) 2. ЕПГУ 3. СМЭВ4 4. Федеральная витрина расписаний ЕГИСЗ (Федеральная витрина) 5. Региональная витрина расписаний (Региональная витрина) 6. ГИСЗ субъекта РФ - поставщик сведений о расписаниях (ГИСЗ) 7. Подсистема ФРМО ЕГИСЗ (ФРМО) 8. Подсистема ФРМР ЕГИСЗ (ФРМР) 9. Подсистема РЭМД ЕГИСЗ (РЭМД) 10. Оператор Телемедицины
Входная информация	1. Сведения о расписании 2. Сведения о пациенте 3. Сведения о направлении
Выходная информация	1. Сведения о записи на ТМК

	2. Сведения о расписании (обновлённые) 3. Сведения о направлении (обновлённые) 4. Протокол телемедицинской консультации
--	---

Основной сценарий. Записать пациента на телемедицинскую консультацию к врачу выбранной специальности, проведение телемедицинской консультации

Предусловия	1. Пользователь авторизован на ЕПГУ 2. Пользователю доступна услуга «Запись на телемедицинскую консультацию» 3. Информационная связность ЕПГУ и Региональной витрины функционируют в штатном режиме эксплуатации 4. Консультирующая МО числится в ФРМО 5. Врач-консультант числится в ФРМР
Триггер	Прецедент начинается, когда Пользователь выбирает на ЕПГУ услугу «Запись на телемедицинскую консультацию»
Последовательность (поток) действий	1. ЕПГУ запрашивает у пользователя данные о пациенте (записать себя) 2. Пользователь указывает пациента: себя 3. ЕПГУ автоматически использует персональные данные Пользователя как пациента 4. ЕПГУ, используя данные о прикреплении пациента: – определяет регион оказания услуги; – предоставляет пользователю выбрать регион самостоятельно; – проводит идентификацию пациента на предмет принадлежности к выбранному региону; 5. ЕПГУ, используя данные о пациенте запрашивает в региональных Витринах сведения о направлениях Пациента и открытых случаях 6. ЕПГУ выводит пользователю список направлений на ТМК и открытых случаев 7. Пользователь выбирает из списка нужное направление / открытый случай

	8. ЕПГУ посредством СМЭВ4 считывает расписание с Региональной витрины 9. ЕПГУ выводит пользователю список доступных для записи специальностей врачей, соответствующих направлению / открытому случаю 10. ЕПГУ запрашивает данные в ФРМР по медицинским работникам / в ФРМО по медицинским организациям 11. ЕПГУ выводит Пользователю список врачей (ФИО) совместно с адресом приёма (МО, СП/филиал МО), соответствующих направлению / ФИО врача, соответствующие открытому случаю 12. Пользователь выбирает врача 13. ЕПГУ выводит Пользователю расписание врача (свободные слоты) в форме календаря 14. Пользователь выбирает в расписании дату и время (слот) для записи и подтверждает бронирование 15. ЕПГУ создаёт в ЛК пользователя запись о запланированной телемедицинской консультации 16. ЕПГУ формирует ссылку для подключения пользователя к телемедицинской консультации 17. ЕПГУ посредством СМЭВ4 передаёт в ГИСЗ информацию о запросе на бронирование слота 18. ГИСЗ получает запрос на бронирование, создаёт и сохраняет информацию о бронировании 19. ГИСЗ передаёт данные о созданном бронировании в Региональную витрину 20. ГИСЗ уведомляет врача о записи на телемедицинскую консультацию 21. ГИСЗ формирует статус телемедицинской консультации «Запланирована» по факту бронирования слота расписания 22. ГИСЗ формирует статус телемедицинской консультации «Взята в работу» по факту принятия врачом
--	--

	консультирующей МО запроса на телемедицинскую консультацию 23. ЕПГУ проводит процедуру авторизации пользователя через ЕСИА по факту перехода по ссылке для проведения телемедицинской консультации в указанное время 24. Оператор Телемедицины предоставляет форму для проведения телемедицинской консультации для врача и пациента в указанное в расписании время 25. ГИСЗ по факту завершения телемедицинской консультации предоставляет возможность врачу составить заключение и заполнить протокол телемедицинской консультации с использованием УКЭП 26. ГИСЗ отправляет протокол телемедицинской консультации в РЭМД 27. ГИСЗ по факту заполнения заключения формирует статус телемедицинской консультации «Завершена» 28. ЕПГУ по факту завершения телемедицинской консультации обновляет статус телемедицинской консультации в ЛК пользователя
Результат (постусловия)	1. Создана запись на телемедицинскую консультацию 2. На основании записи проведен сеанс видео-аудио-конференцсвязи между врачом и пациентом 3. Оформлен протокол телемедицинской консультации

3.9 Запись на прием на плановую госпитализацию

3.9.1 Краткое описание

Сервис предназначен для записи на приём к врачу перед плановой госпитализацией.

Сервис «Запись на прием на плановую госпитализацию» обеспечивает следующие функциональные возможности:

- подача заявки пациента на запись на приём к врачу перед плановой госпитализацией;
- возможность выбора в качестве пациента себя или своего ребенка;
- отображение в ленте ЕПГУ факта записи, включая статус записи;
- автоматическое определение региона получения услуги;
- выбор для записи медицинской организации из списка доступных для данного пациента;
- выбор доступного для записи времени приёма;
- на основании заявки на запись создание записи в выбранную медицинскую организацию;
- обновления статуса записи со стороны ИС Поставщика;
- отмена записи по инициативе пользователя ЕПГУ;
- отмена записи также по инициативе МО.

3.9.2 Клиентский путь «Запись на прием на плановую госпитализацию»

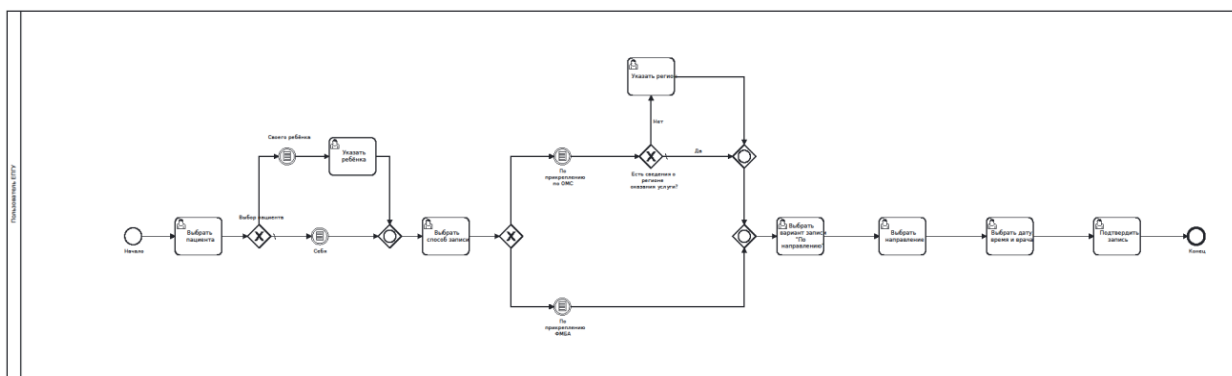


Рисунок 11 – Клиентский путь «Запись на прием на плановую госпитализацию» (объект PDF)

3.9.3 Сценарий «Запись на прием на плановую госпитализацию»

Цель	Записать пациента на приём к врачу в соответствии с направлением на госпитализацию
Участники	1. Пользователь ЕПГУ (Пользователь)

	2. ЕПГУ 3. СМЭВ4 4. Федеральная витрина расписаний ЕГИСЗ (Федеральная витрина) 5. Региональная витрина расписаний (Региональная витрина) 6. ГИСЗ субъекта РФ (ГИСЗ)
Входная информация	1. Сведения о расписании 2. Сведения о пациенте 3. Сведения о направлении
Выходная информация	1. Сведения о записи на приём к врачу 2. Сведения о расписании (обновлённые) 3. Сведения о направлении (обновлённые)

Основной сценарий. Записать пациента на приём на плановую госпитализацию

Предусловия	1. Пользователь авторизован на ЕПГУ 2. Пользователю доступна услуга «Запись на прием на плановую госпитализацию» 3. Информационная связность ЕПГУ и Региональной витрины функционируют в штатном режиме эксплуатации
Триггер	Прецедент начинается, когда Пользователь выбирает на ЕПГУ услугу «Запись на прием на плановую госпитализацию»
Последовательность (поток) действий	1. ЕПГУ запрашивает у пользователя данные о пациенте (записать себя или своего ребёнка) 2. Пользователь указывает пациента: себя или своего ребёнка – в случае выбора «себя» ЕПГУ использует персональные данные Пользователя как пациента – в случае выбора «своего ребёнка» Пользователь выбирает ребёнка из списка зарегистрированных на ЕПГУ своих детей или вводит вручную

	сведения о ребёнке, ЕПГУ использует персональные данные ребёнка как пациента; 3. Пользователь указывает вариант взаимодействия «По прикреплению по ОМС» 4. Пользователь вариант взаимодействия «Запись по направлению на плановую госпитализацию» 5. ЕПГУ выполняет следующие действия: – определяет регион оказания услуги – посредством СМЭВ4 считывает расписание с Региональной витрины; 6. ЕПГУ выводит пользователю список направлений на плановую госпитализацию 7. Пользователь выбирает направление из списка 8. ЕПГУ выводит пользователю список слотов (дата, время, врач) 9. Пользователь выбирает слот 10. ЕПГУ посредством СМЭВ4 передаёт в ГИСЗ информацию о запросе на создании записи в выбранный пользователем слот 11. ГИСЗ получает запрос на создании записи в слот, проверяет возможность создания записи и далее, в случае успеха – создаёт запись – передаёт на ЕПГУ сведения об успешном создании записи – вносит изменения в расписание – передаёт обновлённые сведения о расписании в Региональную витрину 12. Региональная витрина получает обновлённые сведения о расписании и далее – обновляет хранимые данные с учётом новых сведений – передаёт посредством СМЭВ4 обновлённые сведения о расписании в Федеральную витрину
--	--

Результат (постусловия)	Для пациента создана запись на приём к врачу
-------------------------	--

3.10 Запись на прием на медицинское освидетельствование

3.10.1 Краткое описание

Сервис предназначен для записи на приём к врачу для прохождения медицинского освидетельствования проводимого вне рамок реализации программы государственных гарантий бесплатного оказания гражданам медицинской помощи, территориальной программы государственных гарантий бесплатного оказания гражданам медицинской помощи.

Сервис «Запись на прием на медицинское освидетельствование» обеспечивает следующие функциональные возможности:

- подача заявки пациента на запись на приём к врачу для прохождения медицинского освидетельствования;
- возможность выбора в качестве пациента себя;
- отображение в ленте ЕПГУ факта записи, включая статус записи;
- автоматическое определение региона получения услуги;
- выбор для записи медицинской организации из списка доступных для данного пациента;
- выбор доступного для записи времени приёма;
- на основании заявки на запись создание записи в выбранную медицинскую организацию;
- обновления статуса записи со стороны ИС Поставщика;
- отмена записи по инициативе пользователя ЕПГУ;
- отмена записи также по инициативе МО.

[illegible]

Цель	Записать пациента на приём к врачу на медицинское освидетельствование
Участники	1. Пользователь ЕПГУ (Пользователь) 2. ЕПГУ 3. СМЭВ4 4. Федеральная витрина расписаний ЕГИСЗ (Федеральная витрина) 5. Региональная витрина расписаний (Региональная витрина) 6. ГИСЗ субъекта РФ (ГИСЗ)
Входная информация	1. Сведения о расписании 2. Сведения о пациенте
Выходная информация	1. Сведения о записи на приём к врачу 2. Сведения о расписании (обновлённые)

Основной сценарий. Записать пациента на приём к врачу на медицинское освидетельствование

Предусловия	1. Пользователь авторизован на ЕПГУ 2. Пользователю доступна услуга «Запись на приём на медицинское освидетельствование» 3. Информационная связность ЕПГУ и Региональной витрины функционируют в штатном режиме эксплуатации
Триггер	Прецедент начинается, когда Пользователь выбирает на ЕПГУ услугу «Запись на приём на медицинское освидетельствование»
Последовательность (поток) действий	1. ЕПГУ запрашивает у пользователя данные о пациенте (записать себя, своего ребёнка или другого человека) 2. Пользователь указывает пациента: себя, своего ребёнка или другого человека – в случае выбора «себя» ЕПГУ использует персональные данные Пользователя как пациента – в случае выбора «другого человека» Пользователь вводит персональные данные пациента в составе: СНИЛС, ФИО, пол, дата рождения, полис ОМС; 3. ЕПГУ выполняет следующие действия: – определяет регион оказания услуги (на основании сведений о пациенте или запросом к пользователю) – посредством СМЭВ4 считывает расписание с Региональной витрины; 4. ЕПГУ выводит пользователю список доступных для записи специальностей врачей, оказывающих услугу медицинского освидетельствования 5. Пользователь выбирает специальность врача 6. ЕПГУ проверяет наличие у пациента записей на выбранную специальность и далее, в случае отсутствия у пациента записей на выбранную специальность выводит пользователю список слотов (дата, время, врач) 7. Пользователь выбирает слот

	8. ЕПГУ посредством СМЭВ4 передаёт в ГИСЗ информацию о запросе на создании записи в выбранный пользователем слот 9. ГИСЗ получает запрос на создании записи в слот, проверяет возможность создания записи и далее, в случае успеха – создаёт запись – передаёт на ЕПГУ сведения об успешном создании записи – вносит изменения в расписание – передаёт обновлённые сведения о расписании в Региональную витрину 10. Региональная витрина получает обновлённые сведения о расписании и далее – обновляет хранимые данные с учётом новых сведений – передаёт посредством СМЭВ4 обновлённые сведения о расписании в Федеральную витрину
Результат (постусловия)	Создана запись на приём к врачу

4 ПРЕДЛАГАЕМОЕ РЕШЕНИЕ

«РЕГИОНАЛЬНАЯ ВИТРИНА РАСПИСАНИЙ»

4.1 Общие сведения о Витрине данных

Предлагаемое решение «Региональная витрина расписаний субъекта РФ» (далее – Витрина расписаний) является частью НСУД и предназначена для загрузки публикуемых данных в отдельную БД на стороне Поставщика данных. Витрина расписаний представляет собой типовое программное обеспечение, устанавливаемое на стороне поставщиков данных.

Витрина расписаний создаётся для каждого субъекта РФ и обеспечивает накопление сведений из различных медицинских систем этого субъекта и их предоставление потребителям данных с использованием СМЭВ версии 4.х. через ПОДД СМЭВ.

Типовая методика обслуживания витрины данных представлена в Приложении Е.

4.2 Рекомендованные системные требования

Системные требования к серверному оборудованию, необходимому для установки Витрины расписаний данных, описаны с учетом степени внедрения ГИСЗ субъекта РФ в регионах, а также размера населения, и приведены ниже.

Таблица 2 – Рекомендованные системные требования для установки Витрины расписаний на продуктивной среде в регионе с населением до 1,5 миллионов человек

Рекомендовано для количества населения в регионе до 1.5 млн.							
Рекомендованная переподписка по CPU не более x3							
Модуль	Ресурсы на 1 сервер			Общее число серверов	Ресурсы суммарные		
	CPU	RAM,ГБ	Data storage,ТБ		CPU	RAM,ГБ	Data storage,ТБ
Мониторинг / журналирование	6	8	0,1	1	6	8	0,1
Агент ПОДД	4	8	0,05	1	4	8	0,05
PostgreSQL + PGBouncer	8	32	0,5	2	16	64	1

Ядро Prostore	4	8	0,1	1	4	8	0,1
Kafka + ZooKeeper	4	8	0,1	3	12	24	0,3
Адаптер ПОДД	4	8	0,1	1	4	8	0,1
Адаптер Витрины	6	8	0,1	1	6	8	0,1
Итого контур				10	52	128	1,75
*Рекомендуется использовать диски производительностью ≥ 200 IOPS							

Таблица 3 – Рекомендованные системные требования для установки Витрины расписаний на продуктивной среде в регионе с населением от 1,5 до 3,5 миллионов человек

Рекомендовано для количества населения в регионе от 1.5 млн. до 3.5 млн.							
Рекомендованная переподписка по CPU не более x2							
Модуль	Ресурсы на 1 сервер			Общее число серверов	Ресурсы суммарные		
	CPU	RAM, ГБ	Data storage, ТБ		CPU	RAM, ГБ	Data storage, ТБ
Мониторинг / журналирование	8	12	0,3	1	8	12	0,3
Агент ПОДД	6	12	0,05	1	6	12	0,05
PostgreSQL + PGBouncer	10	48	0,5	2	20	96	1
Ядро Prostore	6	12	0,1	1	6	12	0,1
Kafka + ZooKeeper	6	12	0,2	3	18	36	0,6
Адаптер ПОДД	6	12	0,1	1	6	12	0,1
Адаптер Витрины	6	8	0,1	1	6	8	0,1
Итого контур				10	70	188	2,25
*Рекомендуется использовать диски производительностью ≥ 3000 IOPS							
*Рекомендуется использовать диски производительностью ≥ 200 IOPS							

Таблица 4 – Рекомендованные средние системные требования для установки Витрины расписаний на продуктивной среде в регионе с населением более 3,5 миллионов человек

Рекомендовано для количества населения в регионе от 1.5 млн. до 3.5 млн.
Рекомендованная переподписка по CPU не более x2

Модуль	Ресурсы на 1 сервер			Общее число серверов	Ресурсы суммарные		
	CPU	RAM, ГБ	Data storage, ТБ		CPU	RAM, ГБ	Data storage, ТБ
Мониторинг / журналирование	8	12	0,3	1	8	12	0,3
Агент ПОДД	6	12	0,05	1	6	12	0,05
PostgreSQL + PGBouncer	10	48	0,5	2	20	96	1
Ядро Prostore	6	12	0,1	1	6	12	0,1
Kafka + ZooKeeper	6	12	0,2	3	18	36	0,6
Адаптер ПОДД	6	12	0,1	1	6	12	0,1
Адаптер Витрины	6	8	0,1	1	6	8	0,1
Итого контур				10	70	188	2,25
*Рекомендуется использовать диски производительностью ≥ 3000 IOPS							
*Рекомендуется использовать диски производительностью ≥ 200 IOPS							

Таблица 5 – Рекомендованные системные требования для установки Витрины расписаний на тестовой среде в регионе с населением до 1,5 миллионов человек

Модуль	Ресурсы на 1 сервер			Общее число серверов	Ресурсы суммарные		
	CPU	RAM, ГБ	Data storage, ТБ		CPU	RAM, ГБ	Data storage, ТБ
Мониторинг / журналирование	6	8	0,1	1	6	8	0,1
Агент ПОДД	4	4	0,05	1	4	4	0,05
PostgreSQL + PGBouncer	4	8	0,5	2	8	16	1
Ядро Prostore	4	6	0,1	1	4	6	0,1
Kafka + ZooKeeper	4	6	0,1	3	12	18	0,3
Адаптер ПОДД	4	8	0,1	1	4	8	0,1
Адаптер Витрины	6	8	0,1	1	6	8	0,1
Итого контур				10	44	68	1,75

Таблица 6 – Рекомендованные системные требования для установки Витрины расписаний на тестовой среде в регионе с населением от 1,5 до 3,5 миллионов человек

Модуль	Ресурсы на 1 сервер			Общее число серверов	Ресурсы суммарные		
	CPU	RAM, ГБ	Data storage,ТБ		CPU	RAM, ГБ	Data storage,ТБ
Мониторинг / журналирование	6	8	0,3	1	6	8	0,3
Агент ПОДД	4	4	0,05	1	4	4	0,05
PostgreSQL + PGBouncer	4	8	0,5	2	8	16	1
Ядро Prostore	4	6	0,1	1	4	6	0,1
Kafka + ZooKeeper	4	6	0,2	3	12	18	0,6
Адаптер ПОДД	4	8	0,1	1	4	8	0,1
Адаптер Витрины	6	8	0,1	1	6	8	0,1
Итого контур				10	44	68	2,25

Таблица 7 – Рекомендованные средние системные требования для установки Витрины расписаний на тестовой среде в регионе с населением более 3,5 миллионов человек

Модуль	Ресурсы на 1 сервер			Общее число серверов	Ресурсы суммарные		
	CPU	RAM, ГБ	Data storage,ТБ		CPU	RAM, ГБ	Data storage,ТБ
Мониторинг / журналирование	6	8	0,6	1	6	8	0,6
Агент ПОДД	4	4	0,05	1	4	4	0,05
PostgreSQL + PGBouncer	4	8	0,75	2	8	16	1,5
Ядро Prostore	4	6	0,1	1	4	6	0,1
Kafka + ZooKeeper	4	6	0,3	3	12	18	0,9
Адаптер ПОДД	4	8	0,1	1	4	8	0,1
Адаптер Витрины	6	8	0,1	1	6	8	0,1
Итого контур				10	44	68	3,35

4.3 Требования к наполнению данными

Наполнение «Региональной витрины расписаний субъекта РФ» данными, необходимыми для надлежащего оказания Услуг с использованием НСУД должно выполняться в соответствии со следующими требованиями:

1. Периодичность наполнения / обновления:
 - данные о расписании МО (сведения о свободных слотах, таблица slot, см. п.А.1.2) – не реже чем раз в 30 сек;
 - прочие данные (сведения о ресурсах МО, пациентах, направлениях и прочее, см. п. А.1.2) – не реже чем раз в 1 час;
2. Состав и структура данных должна соответствовать модели, определённой в приложении «Приложение А» к настоящему документу.

4.4 Требования к составу и структуре региональных витрин расписаний

ГИСЗ субъекта РФ должны обеспечивать:

- Регулярное обновление данных в витринах с установленной периодичностью:
 - Данные по расписанию (сведения о свободных слотах на ЕПГУ - см таблицу Slot в Приложении А) должны обновляться не реже чем раз в 30 сек;
 - Остальные данные должны обновлять не реже чем раз в час.
- Первоначальную загрузку актуальных данных.

Структура модели данных приведена ниже (см. **Ошибка! Источник ссылки не найден.**).

ГИСЗ субъекта РФ должна обеспечивать предоставление следующих сведений:

- Перечень данных, необходимых для идентификации пользователей (см. Таблица 8);

- Перечень данных для оказания услуги «Запись на прием к врачу» на витринах данных (см. Таблица 9)

Таблица 8 - Перечень данных для идентификации пользователей

№ п.п	Описание
1.	Идентификатор пользователя в региональной витрине
2.	Идентификатор пользователя в ЕСИА
3.	Фамилия, Имя, Отчество пользователя
4.	Дата рождения пользователя
5.	СНИЛС
6.	Номер полиса ОМС (ЕНП)

Таблица 9 - Перечень данных для оказания услуги «Запись на прием к врачу» на витринах данных

№ п.п	Описание
1.	Медицинские организации (структурные подразделения)
2.	Ресурсы для оказания медицинской помощи
3.	Услуги
4.	Слоты
5.	Пациенты
6.	Сведения о прикреплении пациентов
7.	Записи на бирки пациентов
8.	Направления пациентов
9.	Периоды недоступности ресурсов
10.	Профили медицинской помощи
11.	Записи по карте на диспансерное наблюдение
12.	Сведения о мониторинге доступности витрины
13.	Результаты проверки ФЛК

Для идентификации пользователей на ЕПГУ должен использоваться идентификатор пользователя в ЕСИА, для идентификации пользователя в витрине должен использоваться идентификатор пользователя в региональной витрине. Идентификатор пользователя в региональной витрине должен передаваться на ЕПГУ в ответ на запрос валидации пациента по регламентированному запросу с параметрами: ФИО, дата рождения, номер ОМС, СНИЛС.

Проверка по критериям идентификации пациента в БД (по актуальным данным пациента) выполняется в следующей последовательности по указанным наборам входных данных запроса:

Мнемоника рег.запроса	Входные данные	Пример sql-запроса

getPatientByOMS_SNILS_Birthday	• Номер полиса • СНИЛС • Дата рождения	SELECT * FROM misdmXX.getPatientByOMS_SNILS_Birthday ('oms_number', 'snils', 'birthdate')
getPatientByOMS_SNILS_LastName	• Номер полиса • СНИЛС • Фамилия	SELECT * FROM misdmXX.getPatientByOMS_SNILS_LastName ('norm_lastname', 'snils', 'oms_number')
getPatientByOMS_SNILS_FN_MN	• Номер полиса • СНИЛС • Имя • Отчество	SELECT * FROM misdmXX.getPatientByOMS_SNILS_FN_MN ('oms_number', 'snils', 'norm_firstname', 'norm_middlename')
getPatientByOMS_BD_LN_FN	• Номер полиса • Дата рождения • Фамилия • Имя	SELECT * FROM misdmXX.getPatientByOMS_BD_LN_FN ('oms_number', 'birthdate', 'norm_lastname', 'norm_firstname')
getPatientBySNILS_BD_FN	• СНИЛС • Дата рождения • Фамилия • Имя	SELECT * FROM misdmXX.getPatientBySNILS_BD_FN ('snils', 'birthdate', 'norm_lastname', 'norm_firstname')

Номер полиса сравнивается в БД с единым номером полиса нового образца и с номером полиса старого образца, т.е. критерий идентификации в части номера полиса выполняется, если (Номер полиса = Единый номер полиса в БД) или (Номер полиса = Номер полиса старого образца в БД).

Если идентифицировать пользователя не удастся или данные по пользователю будут отсутствовать в региональных витринах, пользователю ЕПГУ будет выведено уведомление «Пользователь не найден. Проверьте правильность указания данных в Личном кабинете», с возможностью перехода в Личный кабинет и отправки жалобы.

Для получения сведений о текущих направлениях пользователя по полученному

идентификатору пациента в витрине запрашивается список направлений пользователя с помощью регламентированного запроса вида:

Мнемоника рег.запроса	Входные данные	Пример sql-запроса
getReferralByPatientId	Идентификатор пациента	SELECT * FROM misdmXX.getReferralByPatientId('patient_id')

Для получения списка медицинских организаций, по полученному идентификатору пациента в витрине запрашивается список доступных по прикреплению медицинских организаций с помощью регламентированного запроса вида:

Мнемоника рег.запроса	Входные данные	Пример sql-запроса
getMOByPatientId	Идентификатор пациента	SELECT * FROM misdmXX.getMOByPatientId('patient_id')

Для получения списка специальностей по полученному идентификатору пациента в витрине запрашивается список доступных специальностей по прикреплению (в том числе во всех медицинских организациях по прикреплению) с помощью регламентированного запроса вида:

Мнемоника рег.запроса	Входные данные	Пример sql-запроса
getSpecialistsByPatientId	Идентификатор пациента	SELECT * FROM misdmXX.getSpecialistsByPatientId ('patient_id')

Для получения списка специальностей по выбранной медицинской организации запрашивается список специальностей, доступных в данной медицинской организации с помощью регламентированного запроса вида

Мнемоника рег.запроса	Входные данные	Пример sql-запроса
getResourcesByMOid	Идентификатор медорганизации	SELECT * FROM misdmXX.getResourcesByMOid('mo_id') where resource_type = 'SPECIALIST'

Для получения списка текущих записей пользователя по полученному идентификатору пациента в витрине запрашивается список текущих записей на прием к врачу с помощью регламентированного запроса вида:

Мнемоника рег.запроса	Входные данные	Пример sql-запроса
	данные	

getBooksByPatientId	Идентификатор пациента Дополнительно стоит фильтр по статусу и времени визита.	SELECT * FROM misdmXX.getBooksByPatientId('Patient_id') where book_status_code = '5000' and slot_visittime > (cast(current_date as timestamp))
---------------------	--	---

Для получения сведений о прикреплении по диспансерному учету по полученному идентификатору пациента в витрине запрашивается информация с помощью регламентированного запроса вида:

Мнемоника рег.запроса	Входные данные	Пример sql-запроса
getObservationByPatientId	Идентификатор пациента	SELECT * FROM misdmXX.getObservationByPatientId (patient_id')

Для получения сведений о доступных слотах для выбранной специальности за указанный интервал времени (14 дней) осуществляется регламентированный запрос вида:

Мнемоника рег.запроса	Входные данные	Пример sql-запроса
geSlotsAllMoByPostId	- Идентификатор должности - Дата/время (от) - Дата/время (до) - Идентификатор пациента	SELECT * FROM misdmXX. geSlotsAllMoByPostId (resource_post_id', TIMESTAMP 'visittime_from', TIMESTAMP 'visittime_to', 'attachment_patient_id'))

Для получения сведений о доступных слотах для записи по направлению на консультацию, во входных параметрах передается идентификатор специальности и идентификатор МО, куда выписано направление.

Мнемоника рег.запроса	Входные данные	Пример sql-запроса
getFreeSlotsBy_SpecId_moId	- Идентификатор специальности - Идентификатор медорганизации	SELECT * FROM misdmXX.getFreeSlotsBy_SpecId_moId (speciality_id', 'mo_id')

Для получения сведений о доступных слотах для выбранного врача за указанный день с дополнительным фильтром по типу слота необходимо выполнить регламентированный запрос вида:

Мнемоника рег.запроса	Входные данные	Пример sql-запроса
	данные	

getFreeSlotsByResourceId_m oId	• Идентификатор ресурса • Время (от) • Время (до) • Идентификатор медроганизации	SELECT * FROM misdmXX.getFreeSlotsByResourceId_m oId ('resource_id', 'visittime_from', 'visittime_to', 'mo_id') where slot_tag_type in ('ORDINARY', 'EPGU', 'REGISTRY', 'REFERRAL', 'D_OBSERVATION')
-----------------------------------	---	--

Для получения сведений о списке недоступных ресурсов в указанный день необходимо выполнить регламентированный запрос вида:

Мнемоника рег.запроса	Входные данные	Пример sql-запроса
getresourcelistbypostid_dates	• Идентификатор пациента • Идентификатор должности специальности • Дата	SELECT * FROM misdmXX.getresourcelistbypostid_dates ('patient_id', 'resource_post_id', 'visitdate', 'visitdate')

Для получения сведений о списке ресурсов, относящихся к указанному профилю медицинской помощи необходимо выполнить следующий регламентированный запрос:

Мнемоника рег.запроса	Входные данные	Пример sql-запроса
getProfilecodeResourceByProfileC ode	• Код профиля	SELECT * FROM misdmXX.getProfilecodeResourceByPr ofileCode ('profile_code')

4.5 Требования к предоставлению данных на ЕПГУ из региональной витрины расписаний

Региональная витрина должна быть создана для каждого субъекта РФ, обеспечивать накопление сведений из различных медицинских систем этого субъекта и их предоставление потребителям данных с использованием системы межведомственного электронного взаимодействия (далее - СМЭВ) версии 4.х. через ПОДД СМЭВ.

СМЭВ предназначена для обмена сведениями в рамках сеансов обмена.

Каждый сеанс обмена с использованием СМЭВ включает в себя:

- передачу запроса от информационной системы участника взаимодействия (далее – ИС УВ), инициирующей данный обмен (Потребитель данных);
- обратную передачу ответа (нескольких ответов) со стороны отвечающей ГИСЗ субъекта РФ (нескольких отвечающих Систем) (Поставщик данных).

СМЭВ поддерживает следующие способы организации обменов:

1. Обмен с использованием Регламентированных запросов – SQL-запросов, выраженных в терминах модели данных, загруженной в ПОДД, и зарегистрированных в Ядре ПОДД под символической мнемоникой, используемой Потребителем данных для выполнения этого запроса.
2. Обмен с использованием Произвольных SQL-запросов со структурой запроса и ответа, задаваемой Потребителем данных;
3. Обмен с использованием зарегистрированных в ПОДД спецификаций OpenAPI.

Потребитель данных, участвующий в информационном взаимодействии с использованием ПОДД СМЭВ, (далее – Потребитель данных ПОДД СМЭВ) имеет возможность:

1. Получать сведения с Витрины Поставщиков данных ПОДД СМЭВ следующими способами:
 - 1) путём отправки регламентированных запросов;
 - 2) путём подписки на изменения сведений и получения сведений в виде:
 - уведомлений об изменениях;
 - копии данных, сформированной в контуре Потребителя данных ПОДД СМЭВ.
2. Выполнять запросы к Поставщикам данных ПОДД СМЭВ с использованием зарегистрированной в ПОДД спецификации OpenAPI.

Поставщики данных ПОДД СМЭВ:

- 1) размещают и актуализируют данные на Витринах и осуществляют подключение Витрин к ПОДД СМЭВ.
- 2) предоставляют доступ к REST-сервисам для выполнения запросов по спецификации OpenAPI.

Каждый Участник может выступать одновременно как в качестве Поставщика данных ПОДД СМЭВ, предоставляя свои сервисы и/или данные со своей Витрины, так и в качестве Потребителя данных ПОДД СМЭВ, выполняя запросы к другим Участникам.

Взаимодействие со СМЭВ 4 должно осуществляться с использованием специализированного типового ПО – Агент ПОДД СМЭВ (инструкция по установке приведена в 6.2).

Агент ПОДД СМЭВ должен обеспечивать взаимодействие с витриной ГИСЗ субъекта РФ с Ядром ПОДД СМЭВ (частью ПОДД СМЭВ, предназначенной для проверки и маршрутизации запросов между участниками взаимодействия).

Потребителем данных должен выступать ЕПГУ. На его ресурсах должны быть развернуты:

- Информационная ГИСЗ субъекта РФ (далее - ИС).
- Агент ПОДД СМЭВ.

Для взаимодействия с Агентом ПОДД СМЭВ, ИС должна включать в себя REST-сервис, обеспечивающий:

- передачу запроса расписания слотов;
- приём ответа на запрос расписания слотов;
- передачу запроса на бронирование слота;
- передача ответа на запрос бронирования.

Для взаимодействия с Агентом ПОДД СМЭВ в части получения расписания слотов также может использоваться JDBC-драйвер ПОДД СМЭВ.

В качестве поставщика данных должна выступать ГИСЗ субъекта РФ (региональная медицинская ГИСЗ субъекта РФ), на ресурсах которой должны быть развёрнуты:

- программного обеспечение «Витрина данных».
- Информационная ГИСЗ субъекта РФ (ИС).
- Агент ПОДД СМЭВ.

Запрос расписания слотов должен обеспечиваться взаимодействием Агента ПОДД СМЭВ с Витриной данных с использованием Apache Kafka.

Запрос бронирования слота должен обеспечиваться взаимодействием Агента ПОДД СМЭВ с ИС через её REST-сервис и ИС с Витриной данных.

Безопасность взаимодействия должна обеспечиваться следующим образом:

- 1) Рядом с Агентом ПОДД СМЭВ устанавливается КриптоПро и размещается закрытый ключ (в соответствии с инструкцией приведённой в), сертификат которого прописывается в ядре СМЭВ 4.
- 2) Взаимодействие между Агентом и Ядром ПОДД должно быть защищено путем выдачи токенов и подписания сообщений.

- 3) Агент ПОДД СМЭВ должен размещаться рядом с Витриной данных в защищенном контуре, инфраструктура которого подлежит отдельной защите: ограничению портов и доступов к каталогам из вне.
- 4) При этом доступ к вызовам (JDBC и REST) Агента ПОДД СМЭВ должен осуществляться без аутентификации и авторизации. Аналогично, Агент взаимодействует с Витриной без шифрования и аутентификации.

Сценарий предоставления пользователю услуги должен состоять из двух частей:

- 1) Предоставление информации из ГИСЗ субъекта РФ пользователю – реализуется с использованием регламентированных запросов.
- 2) Осуществление изменения информации в Системе по инициативе пользователя – реализуется с использованием спецификаций OpenAPI.

Информационный обмен должен осуществляться следующим образом¹:

- 1) Пользователь на ЕПГУ выбирает кого необходимо записать (например, себя).
- 2) Происходит запрос в Цифровой профиль ЕПГУ для идентификации пользователя и получения данных: ФИО, дата рождения, пол, номер полиса ОМС, СНИЛС.
- 3) Модуль «Информационная ГИСЗ субъекта РФ» компонента «ЕПГУ» передаёт SQL-запрос Агенту ПОДД СМЭВ. Передача запроса может осуществляться с использованием:
 - JDBC-подключения;
 - REST-интерфейса.
- 4) Агентом ПОДД СМЭВ из контура ЕПГУ передаёт запрос в Ядро ПОДД СМЭВ.
- 5) Ядро ПОДД СМЭВ передаёт запрос Агенту ПОДД СМЭВ ГИСЗ субъекта РФ.
- 6) Агент ПОДД СМЭВ ГИСЗ субъекта РФ передаёт запрос в Витрину данных для идентификации пользователя в Витрине данных ГИСЗ субъекта РФ.
- 7) Витрина данных возвращает ответ на запрос Агенту ПОДД СМЭВ из контура ГИСЗ субъекта РФ о наличии/отсутствии данных о пользователе в Витрине.
- 8) Агент ПОДД СМЭВ ГИСЗ субъекта РФ передаёт ответ в Ядро ПОДД СМЭВ.
- 9) Ядро ПОДД СМЭВ передаёт ответ Агенту ПОДД СМЭВ ЕПГУ.

¹ Описание шагов проверок и маршрутизации внутри Агентов ПОДД СМЭВ и Ядра ПОДД СМЭВ опущено

- 10) Агенту ПОДД СМЭВ ЕПГУ передаёт ответ в Модуль «Информационная ГИСЗ субъекта РФ» компонента «ЕПГУ».
- 11) Пользователю отображается результат запроса в виде списка доступных медицинских организаций, медицинских работников (специальностей и врачей), слотах, доступных для записи, а также сведения об активных записях.
- 12) Пользователь осуществляет запрос на изменение данных (например, выбор конкретной специальности врача и слота для бронирования), осуществляет запись на выбранный слот (дата и время).
- 13) Модуль «Информационная ГИСЗ субъекта РФ» компонента «ЕПГУ» принимает запрос пользователя.
- 14) Модуль «Информационная ГИСЗ субъекта РФ» компонента «ЕПГУ» передаёт запрос Агенту ПОДД СМЭВ в соответствии со спецификацией OpenAPI (в соответствии со спецификацией OpenAPI происходит передача запросов на бронирование или отмену записи на прием к врачу)
- 15) Агентом ПОДД СМЭВ из контура ЕПГУ передаёт запрос в Ядро ПОДД СМЭВ.
- 16) Ядро ПОДД СМЭВ передаёт запрос Агенту ПОДД СМЭВ ГИСЗ субъекта РФ.
- 17) Агент ПОДД СМЭВ ГИСЗ субъекта РФ передаёт запрос в ГИСЗ субъекта РФ.
- 18) Модуль «Информационная ГИСЗ субъекта РФ» ГИСЗ субъекта РФ возвращает ответ на запрос Агенту ПОДД СМЭВ.
- 19) Агент ПОДД СМЭВ ГИСЗ субъекта РФ передаёт ответ в Ядро ПОДД СМЭВ.
- 20) Ядро ПОДД СМЭВ передаёт ответ Агенту ПОДД СМЭВ ЕПГУ.
- 21) Агенту ПОДД СМЭВ ЕПГУ передаёт ответ в Модуль «Информационная ГИСЗ субъекта РФ» компонента «ЕПГУ» о бронировании выбранного слота для этого пользователя (или отмены).
- 22) Пользователю отображается результат предоставления услуги.
- 23) ГИСЗ субъекта РФ параллельно направляет информацию в Витрину данных.

4.6 Требования к предоставлению данных в витрину из ГИС3 субъекта РФ

НСУД предназначена для управления информацией, содержащейся в информационных системах органов и организаций государственного сектора, а также в информационных ресурсах, созданных в целях реализации полномочий органов и организаций государственного сектора (далее – государственные данные) и для осуществления информационного обмена между поставщиками и получателями данных, присоединившимися к НСУД (далее – Участники НСУД).

Управление процессами информационного обмена между Участниками НСУД осуществляется средствами федеральной государственной информационной системы «Единая информационная платформа Национальной системы управления данными» (далее – ФГИС «ЕИП НСУД»).

Для передачи данных между Участниками НСУД используется среда взаимодействия НСУД, состоящая из Системы межведомственного электронного взаимодействия (далее – СМЭВ) и (или) подсистемы обеспечения доступа к данным СМЭВ (далее – ПОДД СМЭВ) (СМЭВ 4.0), обеспечивающих транспорт и процессинг данных, а также агентов ПОДД СМЭВ, устанавливаемых на стороне Участников НСУД.

Предлагаемые решения «Витрина данных Медиум» (далее – Витрина) являются частью НСУД и предназначены для загрузки публикуемых данных в отдельную БД на стороне поставщика данных. Витрина представляет собой типовое программное обеспечение, устанавливаемое на стороне поставщиков данных (порядок установки представлен в 6.1).

Развертывание Витрины данных² и настройка Агента ПОДД СМЭВ³ в тестовой среде поставщика данных осуществляется в рамках работ по внедрению Витрины данных.

Поставщик данных формирует описание создаваемой Витрины данных (модель Витрины данных см в Приложении А) в ЕИП НСУД и направляет на согласование в Центр компетенции в соответствии с «Инструкцией по работе в ЕИП НСУД».

После согласования модель Витрины данных переходит в статус «Согласовано».

2 Работы выполняются в соответствии с 6.1.

3 Работы выполняются в соответствии с 6.2.

Для модели Витрины данных в статусе «Согласовано» доступно формирование XML-модели Витрины данных для загрузки в тестовую среду ПОДД и создания на ее основе структуры физической Витрины данных.

Для передачи данных из ГИСЗ субъекта РФ в витрину необходимо использовать технологию ЗСПД Минздрава, отвечающий требованиям безопасности по подключению к ГЕОПу.

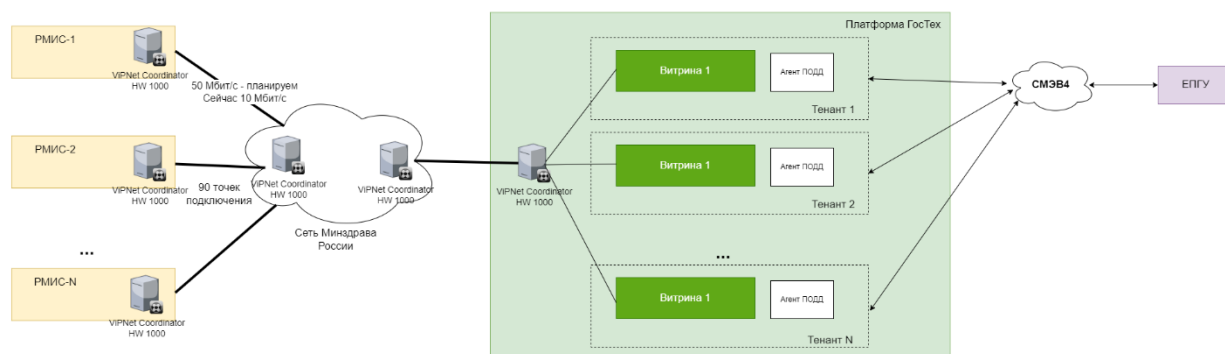


Рисунок 13 – Связность ГИСЗ субъекта РФ– Витрины на платформе ГосТех через ЗСПД Минздрава

Общая схема взаимодействия ГИСЗ субъекта РФ при использовании региональных витрин приведена ниже.

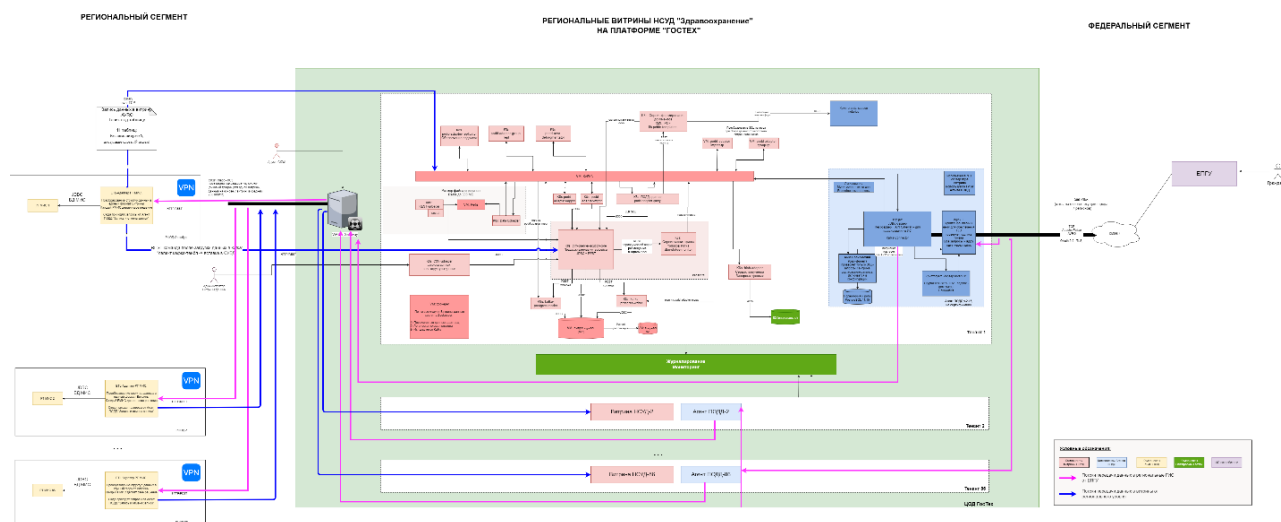


Рисунок 14 – Общая схема взаимодействия региональных государственных информационных систем в сфере здравоохранения с ЕПГУ

Типовое программное обеспечение витрины предоставляется Министерством цифрового развития и массовых коммуникаций РФ на основании лицензионного договора на право использования результатов интеллектуальной деятельности – программ для

электронных вычислительных машин: типового тиражируемого программного обеспечения «Витрина данных» и программного обеспечения Агента ПОДД.

Рекомендуемые требования к аппаратному и программному обеспечению, каналам связи представлены в документации на типовое программное обеспечение витрины и предоставляются вместе с программным обеспечением. Индивидуальная ссылка на скачивание предоставляется субъекту после заключения лицензионного договора.

При наличии свободных ресурсов для размещения региональной витрины субъекту РФ необходимо скачать и заполнить Анкету расчета сайзинга необходимой инфраструктуры (Приложение Г) и направить в адрес Минцифры России. В свою очередь Минцифры подготовит расчет требований к необходимым ресурсам для размещения витрины на инфраструктуре субъекта. В случае размещения региональной витрины в субъекте РФ витрина входит в состав ГИСЗ субъекта РФ и обеспечивает требования информационной безопасности согласно требованиям ГИСЗ субъекта РФ.

В случае отсутствия свободных ресурсов у субъектов РФ для размещения региональной витрины на собственных мощностях возможно ее размещение в государственной единой облачной платформе (ГЕОП). Для этого необходимо скачать и заполнить Анкету расчета сайзинга необходимой инфраструктуры (Приложение Г) и направить в адрес Минцифры России. После получения расчета необходимой инфраструктуры субъекту требуется заполнить заявку на выделения необходимой инфраструктуры (Приложение Д) и официальным письмом направить запрос в адрес Минцифры России. В случае размещения витрины в ГЕОП витрина входит в состав ГИСЗ субъекта РФ и обеспечивает требования информационной безопасности согласно требованиям ГИСЗ субъекта РФ.

Платформа ГосТех - это набор инструментов, позволяющих пользователям Платформы быстро и эффективно создавать бизнес-приложения любой сложности для конечных потребителей, отвечающих новейшим требованиям по обеспечению отказоустойчивости, масштабируемости и безопасности. Интерактивная документация по платформе ГосТех размещена по ссылке <https://platform-docs.v-serv.ru/online-documentation/home/interactive-documentation/>.

Схема взаимодействия с ЕПГУ при использовании витрины ГИСЗ субъекта РФ приведена ниже.

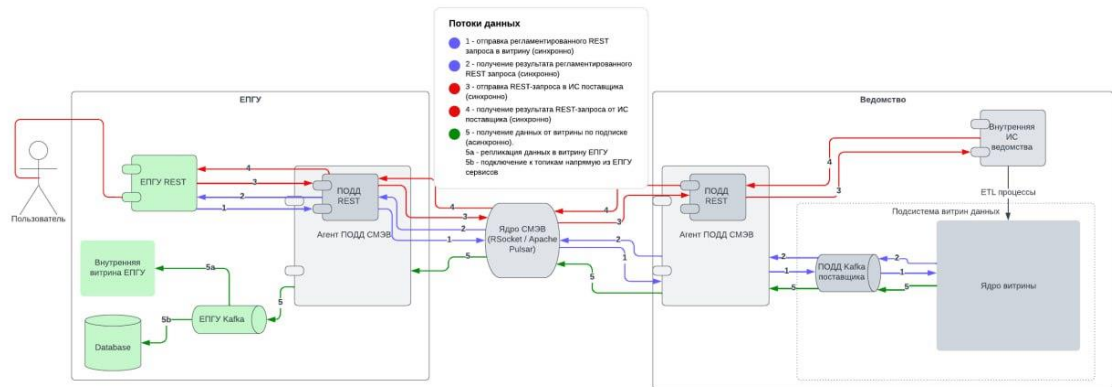


Рисунок 15 – Общая схема взаимодействия с ЕПГУ при использовании региональных витрин

Подключение к сервисам, входящим в состав платформы разработки, используемого в составе базовых сервисов единой цифровой платформы Российской Федерации «ГосТех» и размещенные на средах DEV, TEST, ПСИ. НТ, PROD осуществляется по инструкции, размещенной по ссылке <https://instructions.v-serv.ru/vpn.html> (требуется ввод логина и пароля, запрашивается отдельно у разработчика платформы "ГосТех").

5 ОРГАНИЗАЦИОННЫЕ МЕРОПРИЯТИЯ

5.1 Порядок перехода ГИСЗ субъекта РФ на решение с использованием технологии витрин данных

В соответствии с п.6. приказа Министерства здравоохранения Российской Федерации от 24 декабря 2018 г. N 911н «Требования к государственным информационным системам в сфере здравоохранения субъектов Российской Федерации, медицинским информационным системам медицинских организаций и информационным системам фармацевтических организаций» Оператором ГИС в сфере здравоохранения субъекта РФ (далее – Оператор) является орган исполнительной власти субъекта Российской Федерации, уполномоченный высшими исполнительными органами государственной власти субъектов Российской Федерации на создание, развитие и эксплуатацию государственной информационной системы в сфере здравоохранения субъекта Российской Федерации, организации, назначенные указанными органами, медицинские организации и фармацевтические организации.

Для перехода ГИСЗ субъекта РФ на решение с использованием технологии витрин данных Оператор должен выполнять следующие мероприятия:

1. Заключение соглашения с Минцифры России на предоставление ПО Региональной витрины;
2. В соответствии с указаниями, приведёнными в п.4.2 настоящего документа, определить фактические системные требования к серверному оборудованию, необходимому для установки Региональной витрины расписаний субъекта РФ (далее – Региональная витрина) в ГИСЗ субъекта РФ;
3. Выделить ресурсы, соответствующие фактическим системным требованиям, определённым на предыдущем шаге;
4. В случае отсутствия в ГИСЗ субъекта РФ надлежащих средств криптографической защиты (ПО «КриптоПро CSP» версии 5.0 на сервере и СКЗИ «КриптоПро JCP» версии 2.0 на одном сервере с неограниченным кол-вом ядер), потребных для установки ПО Региональной витрины, произвести закупку, установку и настройку указанного выше программного обеспечения средств криптографической защиты;
5. Установить и настроить ПО Региональной витрины, в соответствии с указаниями, приведёнными в п. 6.1 настоящего документа;

6. Подключить ГИСЗ субъекта РФ к СМЭВ 4, в соответствии с указаниями, приведёнными в п. 5.2;
7. Произвести доработку ПО ГИСЗ субъекта РФ в соответствии с требованиями п.4.3 в целях обеспечения наполнения Региональной витрины данными;
8. Произвести доработку ПО ГИСЗ субъекта РФ в целях обеспечения предоставления данных, содержащихся на Региональной витрине, в «Федеральную агрегированную витрину расписаний Минздрава России» подсистемы ФЭР ЕГИСЗ, а также всем прочим всем Потребителям, определённым в п.2 настоящего документа;
9. Произвести доработку ПО ГИСЗ субъекта РФ в целях обеспечения возможности мониторинга, накопления и передачи в ЕПГУ сведений о состоянии программно-аппаратных компонентов Витрины;
10. Принять административный регламент, определяющий порядок обеспечения согласованности и непротиворечивости (консистентность данных) данных, хранимых в Витрине и включающий в себя следующие положения:
 - Определение эксплуатирующей (подрядной) организации (наименование и ОГРН), а также контактную информацию (Ф.И.О., телефон и адрес электронной почты) ответственного лица или делегированного лица, получившего права доступа к функциям «Личного кабинета участника взаимодействия» СМЭВ4;
 - Определение лиц (контактная информация, включая Ф.И.О., телефон, адрес электронной почты), ответственных за регистрацию ГИСЗ субъекта РФ в СМЭВ4 и имеющего доступ к функциям «Личного кабинета участника взаимодействия» СМЭВ4;
 - Определение лиц (контактная информация, включая Ф.И.О., телефон, адрес электронной почты), ответственных за настройку межсетевого взаимодействия и подключение ГИСЗ субъекта РФ к СМЭВ4;
 - Определение лиц (контактная информация, включая Ф.И.О., телефон, адрес электронной почты), ответственных за доработку ГИСЗ субъекта РФ в части взаимодействия с Региональной витриной;
 - Определение лиц (контактная информация, включая Ф.И.О., телефон, адрес электронной почты), ответственных за администрирование Витрины.

5.2 Порядок подключения ГИСЗ субъекта РФ к СМЭВ 4

Подключение ГИСЗ субъекта РФ к СМЭВ 4 выполняется согласно документу «Методические рекомендации по работе с подсистемой обеспечения доступа к данным федеральной государственной информационной системы «Единая ГИСЗ субъекта РФ межведомственного электронного взаимодействия» версии 2.10.0.0».

Указанный документ размещен в ЕСКС: <https://info.gosuslugi.ru/materials>.

6 ПОРЯДОК РАЗВЁРТЫВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

6.1 Порядок установки программного обеспечения «Витрина данных Медиум»

6.1.1 Предварительные действия

Примечание:

Установка Программы производится в закрытом контуре (без необходимости доступа к сети Интернет).

Перед установкой Программы необходимо выполнить следующие предварительные действия:

1. Создать виртуальные машины согласно схеме развертывания.
2. Установить на серверы ОС (РЕД ОС 7.3, Альт 8 СП Сервер, ASTRA LINUX SPECIAL EDITION 1.7 Воронеж).
3. Создать пользователя `datamart` с правами `sudo` (см. раздел Создание пользователя).
4. Создать приватный и публичный `ssh`-ключ для пользователя `datamart` (см. раздел Создание SSH-ключей).
5. Настроить `ssh`-подключение к серверам, на которых планируется установка программы (см. раздел Подключение к серверу через SSH-клиент PuTTY).
6. Скопировать публичный ключ на сервер (см. раздел Копирование `ssh`-ключей на сервер).
7. Проверить настройки Firewall и отключить при необходимости (см. раздел Отключение межсетевого экрана).
8. Выключить SELinux (см. раздел Отключение SELinux).
9. Указать соответствующий местоположению сервера часовой пояс (см. раздел Выбор часового пояса).
10. Проверить, что на всех серверах установлен сервис синхронизации времени.
11. Проверить, что имена хостов (FQDN) серверов могут взаимно получать IP по имени со всех машин;
12. Убедиться, что хранилище «centos-extras» включено.

13. На серверах, для которых требуется большой объем данных, создать диск большого объема. По умолчанию диск монтируется в /opt.

6.1.1.1 Установка операционной системы

Подробная инструкция по установке операционной системы доступна на сайте производителя.

После установки операционной системы обновите локальную базу пакетов, для этого выполните следующую команду:

```
sudo yum update
```

6.1.1.2 Создание пользователя

Для установки программы рекомендуется создать отдельного пользователя, для этого следует выполнить следующие действия:

- создать пользователя **datamart**;
- отключить для пользователя пароль при вызове **sudo** (необходимо для автоматической установки);
- назначить пользователю права **sudo**.

6.1.1.3 Создание и настройка пользователя datamart

Чтобы создать пользователя **datamart** и установить для него пароль, выполните команды:

```
sudo useradd datamart
sudo password for user:
sudo passwd datamart
sudo password for user:
Changing password for user datamart.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

6.1.1.4 Отключение пароля

Чтобы отключить пароль **sudo** для пользователя **datamart**, надо добавить в настройки (пользователя или группы) директиву **NOPASSWD**. Для этого последовательно выполните команду:

```
sudo visudo
```

В открывшемся конфигурационном файле, с помощью команд редактора **vi**, введите в последней строке:

```
datamart ALL=(ALL) NOPASSWD: ALL
```

Сохраните изменения и закройте файл.

6.1.1.5 Добавление пользователя в группу администраторов

Чтобы добавить пользователя в группу администраторов, у которых есть права выполнения команды **sudo**, выполните следующую команду:

```
sudo usermod -aG wheel datamart
```

Для проверки вы можете переключиться в учетную запись **datamart** и вывести список содержимого директории **/root**, которое обычно доступно только для пользователя **root user**:

```
su datamart
Password:
sudo ls -la /root
```

6.1.1.6 Отключение межсетевого экрана

Для корректной установки потребуется отключить службу **Firewalld** операционной системы ГИСЗ субъекта РФ.

Что просмотреть текущий статус работы приложения используйте команду **firewall-cmd**:

```
sudo firewall-cmd --state
```

В случае, если служба **Firewalld** запущена, команда выше выведет следующее сообщение:

```
running
```

Вы можете временно остановить службу **Firewalld** для этого выполните следующую команду:

```
sudo systemctl stop firewalld
```

Следует учитывать, что данная команда только временно отключит службу, при последующей перезагрузке служба **Firewalld** снова будет запущена. Чтобы полностью отключить службу выполните следующую команду:

Остановите службу:

```
sudo systemctl stop firewalld
```

Отключите автоматический запуск службы **Firewalld** при загрузке операционной системы ГИСЗ субъекта РФ:

```
sudo systemctl disable firewalld
```

После отключения проверьте, что статус службы изменился на **not running**, для этого выполните команду

```
sudo firewall-cmd --state  
not running
```

6.1.1.7 Отключение SELinux

Для корректной установки необходимо отключить SELinux, для этого выполните следующие действия:

1. Проверьте параметры запуска **SELinux** при загрузке системы ГИСЗ субъекта РФ. Для этого выполните следующую команду:

```
cat /etc/selinux/config
```

2. Если параметр **SELINUX** имеет значение **enforcing**, отключите запуск **SELinux** при загрузке системы ГИСЗ субъекта РФ. Для этого следует в файле **/etc/selinux/config** указать значение **SELINUX=disabled** и перезагрузите сервер. **SELinux** будет отключен.

Открыть и отредактировать файл **/etc/selinux/config** можно с помощью редактора **vi**, для этого выполните команду:

```
sudo vi /etc/selinux/config
```

3. Проверьте, что служба отключена. Для этого выполните команду:
sestatus

В ответ вы должны получить:

```
SELinux status: disabled
```

6.1.1.8 Выбор часового пояса

Проверьте, что установлен нужный часовой пояс.

На команду **timedatectl**, должна выводиться строка **Time zone: Europe/Moscow (MSK, +0300)**.

Пример команды:

```
timedatectl
```

Пример ответа:

```
Local time: Mon 2021-12-20 12:06:39 MSK
Universal time: Mon 2021-12-20 09:06:39 UTC
RTC time: Mon 2021-12-20 09:06:49
Time zone: Europe/Moscow (MSK, +0300)
NTP enabled: n/a
NTP synchronized: no
RTC in local TZ: no
DST active: n/a
```

Если результат отличается, укажите соответствующий местоположению сервера часовой пояс.

Пример команды для московского часового пояса:

```
sudo timedatectl set-timezone Europe/Moscow
```

6.1.1.9 Подключение к серверу через SSH-клиент PuTTY

Настройку ssh-подключения к серверу можно выполнить, используя клиент удалённого доступа - [PuTTY](#).

6.1.1.10 Создание SSH-ключей

Внимание:

Действия этого раздела необходимо выполнять под созданной учетной записью **datamart**

Для переключения на учетную запись **datamart** выполните команду:

```
sudo su - datamart
```

Для подключения Ansible к серверу по протоколу SSH необходимо создать ssh-ключи.

Для аутентификации пользователя на сервере используются два ключа: *приватный* и *публичный*.

Чтобы сгенерировать SSH-ключи для операционной ГИСЗ субъекта РФ, выполните следующие действия:

1. Откройте терминал на вашем компьютере и выполните команду

```
ssh-keygen -t rsa
```

Следует оставить все значения по умолчанию.

Примечание:

Если команда ssh-keygen не найдена, установите пакет **openssh**.

2. После выполнения команды вам будет предложено указать имена файлов, в которые будут сохранены ключи и ввести пароль для закрытого ключа. По умолчанию используется имя **id_rsa**, ключи будут созданы в директории **~/.ssh**.

Публичная часть ключа будет сохранена в файле с названием `<имя_ключа>.pub`.

3. Будет выведено следующее сообщение:

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/datamart/.ssh/id_rsa).
```

4. Нажмите клавишу **Enter** - ключ будет сохранен в указанную директорию по умолчанию. Далее вам будет предложено ввести кодовое слово для дополнительной защиты ключа. Вы можете пропустить данный шаг и нажать **Enter**.

5. На запрос указать кодовое слово, не вводя его нажмите **Enter**.

Процедура создания ключей завершена, ключи сохранены в директории `~/.ssh/` в файлах `id_rsa` и `id_rsa.pub`.

6.1.1.11 Копирование ssh-ключей на сервер

Внимание:

Действия этого раздела необходимо выполнять под созданной учетной записью `datamart`

Для переключения на учетную запись `datamart` выполните команду:

```
sudo su - datamart
```

Для копирования ssh-ключей в список разрешенных ключей выполните команды:

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
```

6.1.1.12 Установка сервиса синхронизации времени

Для корректной установки программы необходимо убедиться, что на всех серверах установлен сервис синхронизации времени

6.1.1.13 Настройка имен хостов (FQDN) на серверах.

Для корректной установки программы необходимо проверить, что имена хостов (FQDN) серверов могут взаимно получать IP по имени со всех машин.

6.1.1.14 Включение внешнего репозитория

Репозиторий содержит компоненты, которые обеспечивают дополнительную функциональность для ОС. Этот репозиторий поставляется с ОС и включен, по умолчанию.

6.1.1.15 Создание диска для данных

На серверах, для которых требуется большой объем данных, создать диск большого объема. По умолчанию диск монтируется в /opt.

6.1.2 Базовый план распределения серверов для конфигурации Медиум

Инсталляция приложений производится на подготовленные заранее сервера, на один сервер можно установить несколько приложений. Перед началом установки необходимо выделить 1 сетевой адрес в сети для балансировщика PostgreSQL.

Вариант плана распределения серверов в ЦОД, для примера:

1. dev-nsud-logging 172.29.7.101
2. dev-nsud-agents 172.29.7.102
3. dev-nsud-db01 172.29.7.103
4. dev-nsud-db02 172.29.7.104
5. dev-nsud-prostore 172.29.7.105
6. dev-nsud-kafka01 172.29.7.106
7. dev-nsud-kafka02 172.29.7.107
8. dev-nsud-kafka03 172.29.7.108
9. dev-nsud-adapter 172.29.7.109

6.1.3 Порядок установки конфигурации Медиум

1. Проверить соответствие серверов техническим характеристикам (см. раздел «Настройка на состав технических средств»).
2. Выполнить предварительные действия перед установкой программы (см. раздел «Предварительные действия»).
3. Установить на серверы программное обеспечение, в соответствии с базовым порядком и рекомендуемым распределением, определёнными ниже.

6.1.4 Базовый порядок установки приложений конфигурации Медиум

1. Кластер PostgreSQL
2. Кластер Zookeeper

3. Кластер Kafka
4. Конекторы kafka-postgreSQL
5. Приложения ProStore (DTM)
6. Адаптер ПОДД
7. Приложение csv-uploader-1.0.23

6.1.5 Рекомендуемое распределение компонентов по серверам для конфигурации Медиум

6.1.5.1 Kafka, Zookeeper, PostgreSQL

Рекомендуется устанавливать версию Kafka-2.7.1, ZooKeeper-3.5.8.

Сервер **dev-nsud-kafka01 172.29.7.106** Планировать установку компонент:

1. PostgreSQLcluster manager (на выбор)
2. zookeeper
3. kafka

Сервер **dev-nsud-kafka02 172.29.7.107** Планировать установку компонент:

1. PostgreSQLcluster manager (на выбор)
2. zookeeper
3. kafka

Сервер **dev-nsud-kafka03 172.29.7.108** Планировать установку компонент:

1. PostgreSQLcluster manager (на выбор)
2. zookeeper
3. kafka

6.1.5.2 БД PostgreSQL-13

Рекомендуется устанавливать версию PostgreSQL-13.

Сервер **dev-nsud-db01 172.29.7.103** Планировать установку компонент:

1. PostgreSQL (master)

Сервер **dev-nsud-db02: 172.29.7.104** Планировать установку компонент:

1. PostgreSQL (slave)

6.1.5.3 Prostore, коннекторы к БД

Сервер **dev-nsud-prostore 172.29.7.105** Планировать установку компонент:

1. kafka-postgres-reader
2. kafka-postgres-writer
3. dtm-query-execution-core
4. dtm-status-monitor

6.1.5.4 Адаптер ПОДД

Сервер **dev-nsud-adapter 172.29.7.109** Планировать установку компонент:

1. podd-adapter-group-repl
2. podd-adapter-group-tp
3. podd-adapter-import-tp
4. podd-adapter-mppr
5. podd-adapter-mppw
6. podd-adapter-query
7. podd-adapter-replicator
8. podd-avro-defragmentator

6.1.5.5 Сервер логирования + csv-uploader

Сервер **dev-nsud-logging 172.29.7.101** Планировать установку компонент:

1. csv-uploader
2. Loki

6.1.5.6 Агент ПОДД

Сервер **dev-nsud-agents 172.29.7.102** Планировать установку компонент:

1. Агент – ПОДД (не входит в состав ПО Витрины)

6.1.6 Установка Kafka Zookeeper

Kafka и Zookeeper устанавливается на кластере в соответствии с инструкцией производителя в зависимости от выбранного поставщика.

6.1.7 Установка PostgreSQL

Кластер PostgreSQL устанавливается в конфигурации Мастер-Реплика и использованием дополнительных инструментов балансировки и переключения в соответствии с инструкцией производителя в зависимости, от выбранного поставщика.

6.1.8 Установка ПО ProStore

Действия по установке выполняются через SSH консоль технологического пользователя.

Ядро витрины данных Prostore - ГИСЗ субъекта РФ предоставляет доступ к логической схеме данных, беря на себя управление физической схемой данных в СУБД хранилища. Это позволяет внешней системе работать с системой Prostore, как с обычной реляционной базой данных.

Общий процесс установки состоит из следующих действий:

1. Настроить конфигурацию ядра витрины данных.
2. Создать на сервере директорию для загрузки файлов ядра.
3. Загрузить файлы ядра в созданную директорию.
4. Запустить jar-файл модуля.

Настройка конфигурации выполняется путем редактирования параметров файла **application.yml**. Пример конфигурации файла **application.yml** и возможные настройки конфигурации модуля см. Конфигурация ядра витрины данных Prostore в папке с дистрибутивом. Для установки необходимо выполнить следующие шаги:

1. Создать на сервере ProStore папки: */opt/query-execution* и */opt/status-monitor*.
2. В файлах сборки *adtm-master* перейти в папку *dtm-query-execution-core*.
3. Скопировать файл из сборки *target/dtm-query-execution-core-5.7.0.jar* (или более поздний) и загрузить в папку */opt/query-execution* на сервере.
4. Скопировать файл из сборки *config/application-default.yml* и загрузить в как */opt/query-execution/application.yml* на сервере.

5. Перейти в папку *dtm-status-monitor*.
6. Скопировать файлы из сборки (*target/dtm-status-monitor-5.7.0.jar* (или более поздний) и *dtm-status-monitor/src/main/resources/application.yml* в папку */opt/status-monitor* на сервере.
7. Создать файлы */opt/query-execution/logback.xml* и */opt/status-monitor/logback.xml*.
Содержимое файлов:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
<appender name="FILE"
class="ch.qos.logback.core.rolling.RollingFileAppender">
<file>application.log</file>
<rollingPolicy
class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
<!-- daily rollover -->
<fileNamePattern>application.%d{yyyy-MM-dd}.log</fileNamePattern>
<!-- keep 30 days' worth of history capped at 3GB total size -->
<maxHistory>30</maxHistory>
<totalSizeCap>3GB</totalSizeCap>
</rollingPolicy>
<encoder>
<pattern>%d{yyyy-MM-dd HH:mm:ss.SSS} [%thread] %-5level %logger{35} -
%msg%n</pattern>
</encoder>
</appender>
<root level="INFO">
<appender-ref ref="FILE"/>
</root>
</configuration>
```

8. Создать файл */etc/systemd/system/query-execution.service* со следующим содержимым:

```
[Unit]
Description="Core service"
After=syslog.target
[Service]
Type=simple
WorkingDirectory=/opt/query-execution
User=root
Group=root
ExecStart=/usr/bin/java \\\
-Dlogback.configurationFile=/opt/query-execution/logback.xml \\\
-jar dtm-query-execution-core-5.7.0.jar \\\
--logging.config=/opt/query-execution/logback.xml
```

9. Создать файл */etc/systemd/system/status-monitor.service* со следующим содержимым:

```
[Unit]
Description="Status monitor service"
After=syslog.target
[Service]
Type=simple
WorkingDirectory=/opt/status-monitor
User=root
Group=root
ExecStart=/usr/bin/java \\\
-Dlogback.configurationFile=/opt/status-monitor/logback.xml \\\
-jar dtm-status-monitor-5.7.0.jar \\\
--logging.config=/opt/status-monitor/logback.xml
```

10. Включить автозапуск сервисов:

```
sudo systemctl enable query-execution status-monitor
```

11. Получить драйвер в папке dtm-jdbc-driver
12. Настроить подключения к JDBC-клиентам типа DataGrip и DBeaver
13. Настроить ProStore.
14. Запустить ProStore.
15. Проверить корректность функционирования ProStore путём подачи произвольного SQL- запроса к ProStore и сопоставления ожидаемого эталонного и полученного результатов.

Пример тестового запроса:

```
SELECT 1;
```

Ответ: 1.

6.1.9 Установка коннекторов к СУБД Postgres-13

Общий процесс установки состоит из следующих действий:

1. Настройка конфигурации модуля (см. kafka-postgres-writer-0.3.0, kafka-postgres-reader-0.3.0 (application.yml),).
2. Загрузка jar-файла на сервер.
3. Запуск jar-файла (см. Запуск модуля).

6.1.10 Установка ПОДД-адаптера

6.1.10.1 Установка ПОДД-адаптер-Модуль исполнения запросов

Описание настроек модуля приведено в «Руководстве администратора ПО «Витрина данных НСУД»».

Внимание:

Данное руководство описывает процесс установки *ПОДД-адаптер- Модуль исполнения запросов* в «Витрину данных НСУД»

Общий процесс установки состоит из следующих действий:

1. Настройка конфигурации *ПОДД-адаптер- Модуль исполнения запросов* (файл **application.yml**).
2. Установка docker-образа на сервер
3. Запуск docker-образа.

Установка сервисов и необходимых сервисных баз данных.

Внимание:

Установка данных сервисов выполняется после установки «Core Services».

Действия по установке *ПОДД-адаптера - Модуль исполнения запросов* выполняются на сервере через SSH-консоль технологического пользователя.

Настройка конфигурации выполняется путем редактирования параметров файла **application.yml**.

Пример конфигурации файла **application.yml** для *ПОДД-адаптера - Модуль исполнения запросов* см. Пример файла application.yml.

Установить файлы образов в локальный репозиторий **docker**.

1. Установить docker-образ *ПОДД-адаптера - Модуль исполнения запросов*, файл образа **.tar** должен быть загружен через **docker**:

```
docker image load -i <image_file>
```

2. Перейти в директорию с файлом **docker-compose.yml**

Например:

```
cd ~/direct
```

3. Запустить *ПОДД-адаптер- Модуль исполнения запросов* командой:


```
docker-compose up -d
```

4. Выполнить проверку корректного запуска docker-образов следующей командой:

```
docker ps
```

В списке запущенных образов должен присутствовать **dtm-adapter-reader-das 3.4.0**

6.1.10.2 Установка ПОДД-адаптер-Модуль MPPR

Описание настроек модуля приведено в «Руководстве администратора ПО «Витрина данных НСУД»».

Общий процесс установки состоит из следующих действий:

1. Настройка конфигурации модуля (см. Конфигурация модуля ПОДД-адаптер - Модуль MPPR (application.yml)).
2. Загрузка jar-файла на сервер.
3. Запуск jar-файла (см. Запуск модуля).
4. Проверка установки (см. Проверка).

Настройка конфигурации выполняется путем редактирования параметров файла **application.yml**. Пример конфигурации файла **application.yml** и возможные настройки конфигурации модуля см. Конфигурация модуля ПОДД-адаптер - Модуль MPPR (application.yml).

Для загрузки файла на сервер выполните команду:

```
scp file.jar user_name@IP:/home/dir
```

где,

file.jar - название jar-файла;

user_name - имя пользователя, например, **sudo** или **root**;

IP - адрес сервера;

/home/dir - директория на сервере, в которую будет загружен файл.

6.1.10.3 Установка ПОДД-адаптер-Модуль MPPW

Описание настроек модуля приведено в «Руководстве администратора ПО «Витрина данных НСУД»».

Модуль **ПОДД-адаптер – Модуль MPPW ** поставляется в виде jar-файла. В поставку также входят следующие файлы: файл настроек конфигурации модуля *ПОДД-адаптер – Модуль MPPW* (application.yml).

Общий процесс установки состоит из следующих действий:

1. Настроить конфигурацию модуля.
2. Создать на сервере директорию для загрузки файлов модуля.
3. Загрузить файлы модуля в созданную директорию.
4. Запустить jar-файл модуля.
5. Проверить установку модуля.

Настройка конфигурации выполняется путем редактирования параметров файла **application.yml**. Пример конфигурации файла **application.yml** и возможные настройки конфигурации модуля см. Конфигурация модуля ПОДД-адаптер - Модуль MPPW (application.yml).

Создайте на сервере папку, в которую будут загружены файлы модуля, например, */opt/podd-adapter-mppw*.

В случае, если ранее была установлена старая версия модуля *ПОДД-адаптер – Модуль MPPW*, сделайте его резервную копию.

Для загрузки файла на сервер выполните команду:

```
scp file.jar user_name@IP:/opt/podd-adapter-mppw
```

где,

file.jar - название jar-файла модуля;

user_name - имя пользователя, например, **sudo** или **root**;

IP - адрес сервера;

/opt/podd-adapter-mppw - директория на сервере, в которую будет загружен файл.

6.1.10.4 Установка ПОДД-адаптер – Модуль импорта данных табличных параметров

Описание настроек модуля приведено в «Руководстве администратора ПО «Витрина данных НСУД»».

Действия по установке выполняются через SSH консоль технологического пользователя.

Модуль *ПОДД-адаптер - Модуль импорта данных табличных параметров* поставляется в виде jar-файла. В поставку также входят следующие файлы: файл настроек конфигурации модуля *ПОДД-адаптер - Модуль импорта данных табличных параметров* (application.yml).

Общий процесс установки состоит из следующих действий:

1. Настроить конфигурацию модуля.
2. Создать на сервере директорию для загрузки файлов модуля.
3. Загрузить файлы модуля в созданную директорию.
4. Запустить jar-файл модуля.
5. Проверить установку модуля.

Настройка конфигурации выполняется путем редактирования параметров файла **application.yml**. Пример конфигурации файла **application.yml** и возможные настройки конфигурации модуля см. Конфигурация модуля *ПОДД-адаптер - Модуль импорта данных табличных параметров* (application.yml).

Создайте на сервере папку, в которую будут загружены файлы модуля, например, */opt/podd-adapter-import-tp*.

В случае, если ранее была установлена старая версия модуля *ПОДД-адаптер - Модуль импорта данных табличных параметров*, сделайте его резервную копию.

Для загрузки файла на сервер выполните команду:

```
scp file.jar user_name@IP:/podd-adapter-import-tp
```

где,

file.jar - название jar-файла модуля;

user_name - имя пользователя, например, **sudo** или **root**;

IP - адрес сервера;

/opt/podd-adapter-import-tp - директория на сервере, в которую будет загружен файл.

6.1.10.5 Установка ПОДД-адаптер – Модуль группировки данных табличных параметров

Описание настроек модуля приведено в «Руководстве администратора ПО «Витрина данных НСУД»».

Действия по установке выполняются через SSH консоль технологического пользователя.

Модуль *ПОДД-адаптер - Модуль импорта данных табличных параметров* поставляется в виде jar-файла. В поставку также входят следующие файлы: файл настроек конфигурации модуля *ПОДД-адаптер - Модуль импорта данных табличных параметров* (application.yml).

Общий процесс установки состоит из следующих действий:

1. Настроить конфигурацию модуля.
2. Создать на сервере директорию для загрузки файлов модуля.
3. Загрузить файлы модуля в созданную директорию.
4. Запустить jar-файл модуля.
5. Проверить установку модуля.

Настройка конфигурации выполняется путем редактирования параметров файла **application.yml**. Пример конфигурации файла **application.yml** и возможные настройки конфигурации модуля см. Конфигурация модуля *ПОДД-адаптер - Модуль импорта данных табличных параметров* (application.yml).

Создайте на сервере папку, в которую будут загружены файлы модуля, например, */opt/podd-adapter-import-tp*.

В случае, если ранее была установлена старая версия модуля *ПОДД-адаптер - Модуль импорта данных табличных параметров*, сделайте его резервную копию.

Для загрузки файла на сервер выполните команду:

```
scp file.jar user_name@IP:/podd-adapter-import-tp
```

где,

file.jar - название jar-файла модуля;

user_name - имя пользователя, например, **sudo** или **root**;

IP - адрес сервера;

/opt/podd-adapter-import-tp - директория на сервере, в которую будет загружен файл.

6.1.10.6 Установка ПОДД-адаптер – Wrapper

Описание настроек модуля приведено в «Руководстве администратора ПО «Витрина данных НСУД»».

Действия по установке выполняются через SSH консоль технологического пользователя.

Модуль *ПОДД-адаптер - Wrapper* поставляется в виде jar-файла. В поставку также входят следующие файлы: файл настроек конфигурации модуля *ПОДД-адаптер - Wrapper* (application.yml).

Общий процесс установки состоит из следующих действий:

1. Настроить конфигурацию модуля.
2. Создать на сервере директорию для загрузки файлов модуля.
3. Загрузить файлы модуля в созданную директорию.
4. Запустить jar-файл модуля.
5. Проверить установку модуля.

Настройка конфигурации выполняется путем редактирования параметров файла **application.yml**. Пример конфигурации файла **application.yml** и возможные настройки конфигурации модуля см. в документе «Руководство администратора»..

Создайте на сервере папку, в которую будут загружены файлы модуля, например, */opt/podd-avro-defragmentator*.

В случае, если ранее была установлена старая версия модуля *ПОДД-адаптер - Wrapper*, сделайте его резервную копию.

Загрузите в созданную на предыдущем шаге папку:

jar-файл модуля;
 файл настроек конфигурации модуля *ПОДД-адаптер - Wrapper*
 (application.yml);

Для загрузки файла на сервер выполните команду:

```
scp file.jar user_name@IP:/opt/podd-avro-defragmentator
```

где,

file.jar - название jar-файла модуля;

user_name - имя пользователя, например, **sudo** или **root**;

IP - адрес сервера;

/opt/podd-avro-defragmentator - директория на сервере, в которую будет загружен файл.

6.1.10.7 Установка ПОДД-адаптер – Модуль подписки (опционально)

Описание настроек модуля приведено в «Руководстве администратора ПО «Витрина данных НСУД»».

Модуль *ПОДД-адаптер – Модуль подписки* поставляется в виде jar-файла. В поставку также входят следующие файлы: файл настроек конфигурации модуля *ПОДД-адаптер – Модуль подписки* (application.yml).

Общий процесс установки состоит из следующих действий:

1. Настроить конфигурацию модуля.
2. Создать на сервере директорию для загрузки файлов модуля.
3. Загрузить файлы модуля в созданную директорию.
4. Запустить jar-файл модуля.
5. Проверить установку модуля.

Настройка конфигурации выполняется путем редактирования параметров файла **application.yml**. Пример конфигурации файла **application.yml** и возможные настройки конфигурации модуля см. Конфигурация модуля ПОДД-адаптер - Модуль подписок (application.yml).

Создайте на сервере папку, в которую будут загружены файлы модуля, например, */opt/podd-adapter-replicator*.

В случае, если ранее была установлена старая версия модуля *ПОДД-адаптер – Модуль подписки*, сделайте его резервную копию.

Для загрузки файла на сервер выполните команду:

```
scp file.jar user_name@IP:/opt/podd-adapter-replicator
```

где,

file.jar - название jar-файла модуля;

user_name - имя пользователя, например, **sudo** или **root**;

IP - адрес сервера;

/opt/podd-adapter-replicator - директория на сервере, в которую будет загружен файл.

6.1.10.8 Установка модуля CSV-uploader

Для сборки docker-образа выполните следующую команду:

```
gradle clean build dokkaJavadoc
```

Создать docker-образ

Чтобы собрать docker-образ выполните следующую команду:

```
gradle clean build docker build . -t csv-uploader
```

Действия по установке CSV-uploader выполняются на сервере через SSH-консоль технологического пользователя.

Процесс установки состоит из следующих действий:

1. Настройка конфигурации CSV-uploader (файл application.yml)
2. Установка docker-образа на сервер
3. Запуск docker-образа.

6.1.11 Запуск и остановка программы

«Витрина данных Медиум» не имеют графического интерфейса, при необходимости любой из сервисов/модулей можно остановить и запустить заново.

Для ручной остановки и запуска необходимо подключиться по ssh на сервер и с правами sudo использовать штатный функционал systemctl.

Например:

```
sudo systemctl stop query-execution
sudo systemctl start query-execution
sudo systemctl stop status-monitor
sudo systemctl start status-monitor
```

6.1.12 Проверка программы

6.1.12.1 Проверка ProStore, Kafka и Zookeeper

Для проверки ядра витрины ProStore, Kafka и Zookeeper необходимо отправить SQL+ запрос к ProStore через клиентское JDBC-подключение:

```
CHECK_VERSIONS()
```

Запрос возвращает информацию о версиях следующих программных компонентов:

- компонентов ГИС3 субъекта РФ
(https://prostore.datamart.ru/docs_prostore/overview/components/components.html)
- компонентов, с которыми работает ГИС3 субъекта РФ.

Успешный ответ содержит объект `ResultSet`, где каждая запись — это информация об одном компоненте. Неуспешный ответ содержит исключение

По каждому компоненту возвращается следующая информация:

- `component_name` — имя компонента,
- `version` — номер версии компонента или ошибка, если компонент вернул ошибку при опросе.

Функциональная проверка - создать Витрину в ProStore с помощью SQL-запроса:

```
CREATE DATABASE <имя несуществующей логической базы>, например,  
CREATE DATABASE testdb;
```

Создать таблицу в ProStore со всеми типами колонок с помощью SQL-запроса:

```
CREATE TABLE <имя логической базы из п.1>.all_types (
```

- `id` int not null,
- `double_col` double,
- `float_col` float,
- `char_col` varchar(36),
- `boolean_col` boolean,
- `int_col` int not null,
- `bigint_col` bigint,
- `date_col` date,
- `timestamp_col` timestamp,
- primary key (id) distributed by (id)

Проверить существование и структуру созданной таблицы в ProStore с помощью SQL-запросов:

```
select \* from <имя логической базы из п.1>.all_types  
DATASOURCE_TYPE='ADPG'
```

Удалить таблицу со всеми типами колонок из ProStore с помощью SQL-запроса:

```
DROP TABLE <имя логической базы из п.1>.all_types
```

Удалить Витрину с помощью SQL-запроса:


```
DROP DATABASE <имя логической базы из п.1>.
```

Внимание:

Наличие сообщений об ошибках, а также отличие получаемых состояний ProStore на различных этапах проверки от ожидаемых состояний является индикатором неуспешного прохождения проверки.

6.1.12.2 Проверка ПОДД-адаптера - Модуль исполнения запросов

Для проверки модуля *ПОДД-адаптер - Модуль исполнения запросов* необходимо выполнить запрос к сервису.

Пример запроса

```
curl -s IP:Port/metrics | grep '^liveness '
```

где:

IP - адрес сервера.

Port - адрес сервера.

liveness - параметр проверки работоспособности модуля.

Например:

```
curl -s http://172.16.10.67:9837/metrics | grep '^liveness '
```

Пример успешного ответа

```
liveness 1.0
```

Ответ **1** означает, что модуль работает.

6.1.12.3 Проверка ПОДД-адаптер – Модуль MPPR

Для проверки модуля *ПОДД-адаптер - Модуль MPPR* необходимо выполнить запрос к сервису.

Пример запроса

```
curl -s IP:Port/metrics | grep '^liveness '
```

где:

IP - адрес сервера.

Port - адрес сервера.

liveness - параметр проверки работоспособности модуля.

Например:

```
curl -s http://172.16.10.67:9837/metrics | grep '^liveness '
```

Пример успешного ответа

```
liveness 1.0
```

Ответ **1** означает, что модуль работает.

6.1.12.4 Проверка ПОДД-адаптер-Модуль MPPW

Для проверки модуля *ПОДД-адаптер - Модуль MPPW* необходимо выполнить запрос к сервису.

Пример запроса

```
curl -s IP:Port/metrics | grep '^liveness '
```

где:

IP - адрес сервера.

Port - адрес сервера.

liveness - параметр проверки работоспособности модуля.

Например:

```
curl -s http://172.16.10.67:9837/metrics | grep '^liveness '
```

Пример успешного ответа

```
liveness 1.0
```

Ответ **1** означает, что модуль работает.

6.1.12.5 Проверка ПОДД-адаптер – Модуль импорта данных табличных параметров

Для проверки модуля *ПОДД-адаптер – Модуль импорта данных табличных параметров* необходимо выполнить запрос к сервису.

Пример запроса

```
curl -s IP:Port/metrics | grep '^liveness '
```

где:

IP - адрес сервера.

Port - адрес сервера.

liveness - параметр проверки работоспособности модуля.

Например:

```
curl -s http://172.16.10.67:9837/metrics | grep '^liveness '
```

Пример успешного ответа

```
liveness 1.0
```

Ответ **1** означает, что модуль работает.

6.1.12.6 Проверка ПОДД-адаптер – Модуль группировки данных табличных параметров

Для проверки модуля *ПОДД-адаптер – Модуль импорта данных табличных параметров* необходимо выполнить запрос к сервису.

Пример запроса

```
curl -s IP:Port/metrics | grep '^liveness '
```

где:

IP - адрес сервера.

Port - адрес сервера.

liveness - параметр проверки работоспособности модуля.

Например:

```
curl -s http://172.16.10.67:9837/metrics | grep '^liveness '
```

Пример успешного ответа

```
liveness 1.0
```

Ответ **1** означает, что модуль работает.

6.1.12.7 Проверка ПОДД-адаптер – ПОДД-адаптер – Wrapper

Для проверки модуля *ПОДД-адаптер - Wrapper* необходимо выполнить запрос к сервису.

Пример запроса

```
curl -s IP:Port/metrics | grep '^liveness '
```

где,

IP - адрес сервера.

Port - адрес сервера.

liveness - параметр проверки работоспособности модуля.

Например:

```
curl -s http://172.16.10.67:9837/metrics | grep '^liveness '
```

Пример успешного ответа

```
liveness 1.0
```

Ответ **1** означает, что модуль работает.

6.1.12.8 Проверка ПОДД-адаптер – Модуль подписки

Для проверки модуля *ПОДД-адаптер - Модуль подписок* необходимо выполнить запрос к сервису.

Пример запроса

```
curl -s IP:Port/metrics | grep '^liveness '
```

где:

IP - адрес сервера.

Port - адрес сервера.

liveness - параметр проверки работоспособности модуля.

Например:

```
curl -s http://172.16.10.67:9837/metrics | grep '^liveness '
```

Пример успешного ответа

```
liveness 1.0
```

Ответ **1** означает, что модуль работает.

6.1.12.9 Проверка CSV-uploader

Для проверки успешного завершения установки: Проверьте наличие в log-файле сообщения:

```
(сообщения пока не генерируется)
```

6.1.13 Дополнительные возможности

Инструкции данного раздела не выполняются в рамках первичной установки компонентов ПО «Витрина данных Медиум».

Необходимость выполнения действий данного раздела определяется в процессе эксплуатации программы.

6.1.14 Логирование

Лог-файлы компонентов для ПО «Витрины данных Медиум» могут быть найдены на соответствующих серверах, по относительным путям, описанным ниже:

Наименование	–	Относительный путь
PostgreSQL	–	/var/log/postgre-server/postgre-server.log
	–	/var/log/postgre-server/postgre-server.err.log
ПОДД-адаптер-Модуль исполнения запросов	–	/opt/podd-migration/logs/application.log
	–	/opt/podd-adapter/logs/application.log
REST-адаптер	–	/opt/rest/logs

6.2 Порядок установки программного обеспечения «Агент ПОДД СМЭВ»

6.2.1 Условия работы с Агентом ПОДД СМЭВ

6.2.1.1 Требования к уровню подготовки пользователя

Развертывание Агента ПОДД СМЭВ должно осуществляться в соответствии с настоящим порядком.

Специалист должен обладать следующими знаниями и опытом:

- администрирование КристоПро;
- администрирование «Apache Kafka»;
- администрирование «Kubernetes».

6.2.1.2 Требования к среде для развертывания Агента ПОДД СМЭВ

Минимальные требования к характеристикам сервера для развертывания Агента ПОДД СМЭВ приведены в таблице 10.

Таблица 10 – Минимальные требования к характеристикам сервера

Характеристики	Минимальное требование
ОС	CentOS 7.9
ЦПУ	4-х ядерный процессор

Характеристики	Минимальное требование
Оперативная память	8 ГБ
Дисковое пространство	20 ГБ

6.2.2 Подготовка к работе

6.2.2.1 Состав и содержание дистрибутивного пакета

Дистрибутив Агента ПОДД СМЭВ размещен на технологическом портале СМЭВ3, <https://smev3.gosuslugi.ru/portal/>

6.2.2.2 Порядок установки, запуска и настройки Агента ПОДД СМЭВ

Методические рекомендации и иные инструкции по работе с ПОДД СМЭВ представлены на сайте ЕСКС <https://info.gosuslugi.ru/>, в разделе «Документы СМЭВ4 (ПОДД)» по ссылке https://info.gosuslugi.ru/docs/section/%D0%A1%D0%9C%D0%AD%D0%92_4_%28%D0%9F%D0%9E%D0%94%D0%94%29/

Администратор УВ осуществляет развертывание, запуск и настройку Агента ПОДД СМЭВ согласно Руководству администратора Агента ПОДД СМЭВ, размещенного на сайте ЕСКС <https://info.gosuslugi.ru/>

Типовые ошибки Агента ПОДД - <https://telegra.ph/Tipovye-oshibki-Agenta-Potrebitelya-SMEHV4-PODD-04-18>

ПРИЛОЖЕНИЕ А

СВЕДЕНИЯ ДЛЯ НАПОЛНЕНИЯ РЕГИОНАЛЬНЫХ ВИТРИН РАСПИСАНИЙ

А.1 Сведения о расписании

А.1.1 Диаграмма связей и структуры объектов данных расписания

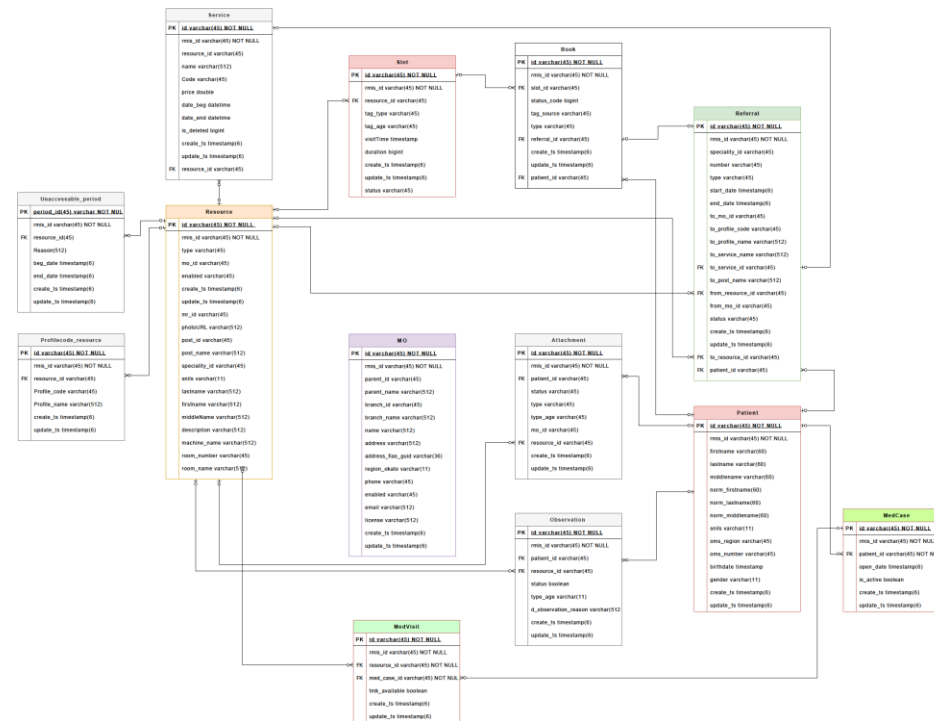


Рисунок 16 – Диаграмма связей и структуры объектов данных расписания (объект PDF)

А.1.2 Описание логической структуры объектов данных расписания

Таблица mo: в таблице содержатся сведения о МО и СП МО.

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи	id	О	VARCHAR (512) NOT NULL		Да
Идентификатор ИС	rmis_id	Н	VARCHAR (512)	Идентификатор ИС, зарегистрированной в ФЭР. Выдается после регистрации ИС в тестовой/промышленной среде ФЭР. Пример: conc-5015-rmis	Нет
Уникальный идентификатор (OID) МО	parent_id	Н	VARCHAR (512)	OID МО должен соответствовать значению в ФРМО	Да
Наименование МО	parent_name	Н	VARCHAR (512)	Наименование МО должно соответствовать значению в ФРМО	Планируется использовать после доработки по хранению структуры МО
Уникальный идентификатор подразделения	branch_id	Н	VARCHAR (512)		Планируется использовать после доработки по хранению структуры МО

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Наименование подразделения	branch_name	Н	VARCHAR (512)		Планируется использовать после доработки по хранению структуры МО
Наименование структурного подразделения	name	Н	VARCHAR (512)		Да
Адрес структурного подразделения	address	Н	VARCHAR (512)		Да
Глобальный идентификатор адреса подразделения в ФИАС	address_fias_guid	Н	VARCHAR (512)	Глобальный уникальный идентификатор адресного объекта ГАР ФИАС	Нет
ОКАТО региона	region_okato	Н	VARCHAR (512)	Содержит код объекта административно-территориального деления по общероссийскому классификатору ОКАТО соответствующего уровня (от субъекта РФ до сельского	Нет

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
				населенного пункта и внутригородских районов или внутригородских округов). Длина кода – 11 разрядов (заполняются все 11 разрядов). Для адресных объектов, не включенных в классификатор ОКАТО, в этом поле указывается код ОКАТО либо старшего административно-территориального объекта, либо расположенного в непосредственной близости к адресуемому объекту административно-территориального объекта, включенного в ОКАТО.	
Телефон подразделения	phone	Н	VARCHAR (512)		Да
Типы	enabled	Н	VARCHAR (512)	ATTACHED_MO - принимает по	Да

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
прикрепления, допустимые к данному подразделению				прикреплениям; ATTACHED_REG - принимает по прикреплению в регионе; ALL - принимает всех без прикрепления; NONE - не принимает	
Время создания записи	create_ts	Н	TIMESTAMP(6)		Нет
Время обновления записи	update_ts	Н	TIMESTAMP(6)		Нет
Адрес электронной почты	email	Н	VARCHAR (512)		Да, для услуги запись на телемедконсультацию
Лицензия на осуществление соответствующих видов деятельности	license	Н	VARCHAR (512)		Да, для услуги запись на телемедконсультацию

Таблица resource: в таблице содержатся сведения о ресурсах МО: врачах, кабинетах, медицинских аппаратах.

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи	id	О	VARCHAR (512) NOT NULL		Да
Идентификатор ИС	rmis_id	Н	VARCHAR (512)	Идентификатор ИС, зарегистрированной в ФЭР. Выдается после регистрации ИС в тестовой/промышленной среде ФЭР. Пример: conc-5015-rmis	Нет
Тип ресурса	type	Н	VARCHAR (512)	SPECIALIST – врач; MACHINE - медицинский аппарат; ROOM - кабинет	Да
Уникальный идентификатор структурного подразделения	mo_id	Н	VARCHAR (512)	Значение уник.идентификатора МО или СП МО, в которому привязан МР (тип ресурса= SPECIALIST) соответствует значению в ФРМР	Да

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Типы прикрепления, допустимые к данному ресурсу	enabled	Н	VARCHAR (512)	ATTACHED_MO - принимает по прикреплениям ATTACHED_REG - принимает по прикреплению в регионе ALL - принимает всех без прикрепления NONE - не принимает	Да
Время создания записи	create_ts	Н	TIMESTAMP(6)		Нет
Время обновления записи	update_ts	Н	TIMESTAMP(6)		Нет
Уникальный идентификатор (OID) МР	mr_id	Н	VARCHAR (45)	Значение уник.идентификатора МР (тип ресурса= SPECIALIST) соответствует значению в ФРМР	Да, для услуги запись на телемедконсультацию
Фотография врача	photoURL	Н	VARCHAR (512)	Ссылка на фото врача (сама фотография хранится в S3 хранилище)	Да, для услуги запись на телемедконсультацию
Код должности	post_id	Н	VARCHAR (512)	Заполняется согласно справочнику ФНСИ Значение кода должности	Да

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
				соответствует значению из справочника ФНСИ «ФРМР. Должности медицинского персонала» (OID 11.2.643.5.1.13.13.99.2.181)	
Наименование должности	post_name	Н	VARCHAR (512)	Обязательно к заполнению при type = SPECIALIST Заполняется согласно справочнику ФНСИ Значение наименования должности соответствует значению из справочника ФНСИ «ФРМР. Должности медицинского персонала» (OID 11.2.643.5.1.13.13.99.2.181)	Да
Идентификатор специальности	speciality_id	Н	VARCHAR (512)	Обязательно к заполнению при type = SPECIALIST Значение Идентификатора специальности соответствует значению из справочника ФНСИ	Да

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
				«Номенклатура специальностей специалистов со средним, высшим и послевузовским медицинским и фармацевтическим образованием в сфере здравоохранения» (OID 1.2.643.5.1.13.13.11.1066)	
Врач, СНИЛС	snils	Н	VARCHAR (512)	Обязательно к заполнению при type = SPECIALIST Значение СНИЛС МР соответствует значению в ФРМР	Нет
Врач, фамилия	lastName	Н	VARCHAR (512)	Обязательно к заполнению при type = SPECIALIST Значение ФИО МР соответствует значению в ФРМР	Да
Врач, имя	firstName	Н	VARCHAR (512)	Обязательно к заполнению при type = SPECIALIST Значение ФИО МР соответствует значению в ФРМР	Да
Врач, отчество	middleName	Н	VARCHAR (512)	Обязательно к заполнению при type = SPECIALIST	Да

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
				Значение ФИО МР соответствует значению в ФРМР	
Описание	description	Н	VARCHAR (512)		Нет
Аппарат, наименование	machine_name	Н	VARCHAR (512)	Обязательно к заполнению при type = MACHINE	Да
Кабинет, номер	room_number	Н	VARCHAR (512)	Обязательно к заполнению при type = MACHINE	Да
Кабинет, наименование	room_name	Н	VARCHAR (512)	Обязательно к заполнению при type = MACHINE	Да

Таблица service: в таблице содержатся сведения об услугах МО.

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи	id	О	VARCHAR (512) NOT NULL		Нет
Идентификатор ИС	rmis_id	Н	VARCHAR (512)	Идентификатор ИС, зарегистрированной в ФЭР. Выдается после регистрации ИС в тестовой/промышленной среде ФЭР.	Нет

				Пример: conc-5015-rmis	
Идентификатор ресурса	resource_id	Н	VARCHAR (512)		Да
Наименование услуги	name	Н	VARCHAR (512)	Заполняется согласно справочнику ФНСИ Значение наименования услуги соответствует значению из справочника ФНСИ «Номенклатура медицинских услуг» (1.2.643.5.1.13.13.11.1070)	Да
Код услуги	code	Н	VARCHAR (512)	Заполняется согласно справочнику ФНСИ Значение кода услуги соответствует значению из справочника ФНСИ «Номенклатура медицинских услуг» OID 1.2.643.5.1.13.13.11.1070	Да
Цена услуги	price	Н	DOUBLE		Нет
Дата открытия	date_beg	Н	DATETIME	Эти данные нужны для: 1. Определения актуальности услуги. Услуга имеющая дату закрытия, признак и дату удаления считается не актуальной.	Нет
Дата закрытия	date_end	Н	DATETIME		Нет
Признак удаления	is_deleted	Н	BOOLEAN		Нет

				2. Разбора проблемных ситуаций в будущем.	
Время создания записи	create_ts	Н	TIMESTAMP(6)		Нет
Время обновления записи	update_ts	Н	TIMESTAMP(6)		Нет

Таблица slot: в таблице содержатся сведения о слотах расписания МО.

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи	id	О	VARCHAR (512) NOT NULL		Да
Идентификатор ИС	rmis_id	О	VARCHAR (512)	Идентификатор ИС, зарегистрированной в ФЭР. Выдается после регистрации ИС в тестовой/промышленной среде ФЭР. Пример: conc-5015-rmis	Нет
Уникальный идентификатор ресурса	resource_id	Н	VARCHAR (512)		Да

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Типы записей	tag_type	Н	VARCHAR (512)	EPGU - для ЕПГУ; ORDINARY – обычный; RESERVE – резервный; PAID – платный; CENTER – для центра записи; REFERRAL – по направлению; D_OBSERVATION – для записи по карте диспансерного наблюдения; INFOMAT – для информата; REGISTRY – для регистратуры EQUEUE – для живой очереди; ONLINE – для приема средствами видеосвязи	Да
Учет возраста	tag_age	Н	VARCHAR (512)	ADULT – взрослый; CHILD – детский; BABY – младенец; ALL – все возрастные группы	Да
Время начала приема	visitTime	Н	TIMESTAMP(6)		Да

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Продолжительность в минутах процедуры оказания государственной услуги по регламенту	duration	Н	BIGINT		Да
Дата создание записи	create_ts	Н	TIMESTAMP(6)		Нет
Дата изменения записи	update_ts	Н	TIMESTAMP(6)		Нет
Статус слота	status	Н	VARCHAR (512)	AVAILABLE – доступен; RECORDED – записан; CANCELED – отменен; RESERVED – забронирован; BLOCKED - заблокирован	Да

Таблица patient: в таблице содержатся сведения о пациентах.

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи	id	О	VARCHAR (512)		Да
Идентификатор ИС	rmis_id	Н	VARCHAR (512)	Идентификатор ИС, зарегистрированной в ФЭР. Выдается после регистрации ИС в тестовой/промышленной среде ФЭР. Пример: conc-5015-rmis	Нет
Имя пациента	firstname	Н	VARCHAR (512)		Да
Фамилия пациента	lastname	Н	VARCHAR (512)		Да
Отчество пациента	middlename	Н	VARCHAR (512)		Да
Имя пациента (нормализованное)	norm_firstname	Н	VARCHAR (512)	Нормализованное значение firstname. Условия: отсутствуют пробелы и специальные символы, буква "ё (Ё)" заменена на "е (Е)".	Да
Фамилия пациента (нормализованное)	norm_lastname	Н	VARCHAR (512)	Нормализованное значение lastname. Условия:	Да

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
				отсутствуют пробелы и специальные символы, буква "ё (Ё)" заменена на "е (Е)".	
Отчество пациента (нормализованное)	norm_middlename	H	VARCHAR (512)	Нормализованное значение middlename. Условия: отсутствуют пробелы и специальные символы, буква "ё (Ё)" заменена на "е (Е)".	Да
СНИЛС пациента	snils	H	VARCHAR (512)		Да
Регион страхования	oms_region	H	VARCHAR (512)		Нет
Номер полиса пациента	oms_number	H	VARCHAR (512)		Да
Дата рождения пациента	birthdate	H	TIMESTAMP(6)		Да
Пол пациента	gender	H	VARCHAR (512)	MALE – мужской; FEMALE – женский	Да
Дата создание записи	create_ts	H	TIMESTAMP(6)		Нет
Дата изменения	update_ts	H	TIMESTAMP(6)		Нет

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
записи					

Таблица attachment: в таблице содержатся сведения о прикреплении пациентов к МО.

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи	id	О	VARCHAR (512) NOT NULL		Да
Идентификатор ИС	rmis_id	Н	VARCHAR (512)	Идентификатор ИС, зарегистрированной в ФЭР. Выдается после регистрации ИС в тестовой/промышленной среде ФЭР. Пример: conc-5015-rmis	Нет
Уникальный идентификатор пациента	patient_id	О	VARCHAR (512)		Да
Идентификатор ресурса	resource_id	Н	VARCHAR (512)		Нет

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Информация о статусе прикрепления	status	Н	VARCHAR (512)	DONE – действительное прикрепление; NONE – недействительное прикрепление	Да
Тип прикрепления к МО	type	Н	VARCHAR (512)	GENERAL – основное; DENTIST – стоматологическое; GYNECOLOG – гинекологическое	Да
Тип прикрепления по возрастной категории	type_age	Н	VARCHAR (512)	ADULT – взрослое; CHILD – детское	Да
Уникальный идентификатор структурного подразделения МО	mo_id	Н	VARCHAR (512)	Значение уник.идентификатора МО или СП МО, в которому привязан МР (тип ресурса= SPECIALIST) соответствует значению в ФРМР	Да
Дата создания записи	create_ts	Н	TIMESTAMP(6)		Нет
Дата изменения	update_ts	Н	TIMESTAMP(6)		Нет

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
записи					
Идентификатор медицинской карты	cards_id	Н	VARCHAR (512)		Нет
Повод для обращения	d_observation_reason	Н	VARCHAR (512)		Нет

Таблица book: в таблице содержатся сведения о записях на приём в МО.

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи	id	О	VARCHAR (512) NOT NULL		Да
Идентификатор ИС	rmis_id	Н	VARCHAR (512)	Идентификатор ИС, зарегистрированной в ФЭР. Выдается после регистрации ИС в тестовой/промышленной среде ФЭР. Пример: conc-5015-rmis	Нет
Уникальный	patient_id	О	VARCHAR (512)		Да

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
идентификатор пациента					
Уникальный идентификатор слота	slot_id	O	VARCHAR (512)		Да
Статус записи, код	status_code	H	BIGINT	5000 (RECORDED) – пациент записан; 5010 (CANCELED) – запись отменена; 5040 (PATIENT_CANCELED) – запись отменена пациентом в МО	Да
Источник записи	tag_source	H	VARCHAR (512)	EPGU – ЕПГУ; INFOMAT – инфомат; DOCTOR – врач; REG – регистратура; OTHER_WEB – другой источник записи	Да
Тип записи	type	H	VARCHAR (512)	APPOINTMENT – прием к врачу; VACCINATION – вакцинация;	Да

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
				COVID_VACCINATION – вакцинация от COVID-19; UDVN_DISPENSARY – усиленная диспансеризация; DISPENSARY – диспансеризация; PROPHYLACTIC – профилактический осмотр; D_OBSERVATION – запись по карте диспансерному наблюдению; ONLINE_CONSULTING – консультация онлайн; PAID_APPOINTMENT – запись на платную услуги BLOOD_DONATION – донация крови.	
Идентификатор направления	referral_id	Н	VARCHAR (512)		Обязательно для записи по направлению

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Дата создание записи	create_ts	Н	TIMESTAMP(6)		Нет
Дата изменения записи	update_ts	Н	TIMESTAMP(6)		Нет
Код услуги	service_code	Н	VARCHAR (512)		Обязательно для записи на услугу (вакцинация)

Таблица referral: в таблице содержатся сведения о имеющихся у пациентов направлениях.

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи	id	О	VARCHAR (512) NOT NULL		Да
Идентификатор ИС	rmis_id	Н	VARCHAR (512)	Идентификатор ИС, зарегистрированной в ФЭР. Выдается после регистрации ИС в тестовой/промышленной среде	Нет

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
				ФЭР. Пример: conc-5015-rmis	
Идентификатор пациента	patient_id	Н	VARCHAR (512)		Да
Идентификатор специальности	speciality_id	Н	VARCHAR (512)	Должно соответствовать коду из справочника ФНСИ «Номенклатура специальностей специалистов со средним, высшим и послевузовским медицинским и фармацевтическим образованием в сфере здравоохранения» (OID 1.2.643.5.1.13.13.11.1066) Обязательно, если тип направления "На консультацию", type = CONSULTATION.	Да
Номер направления	number	Н	VARCHAR (512)		Да
Тип направления	type	Н	VARCHAR (512)	EXAMINATION – на	Да

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
				обследование; CONSULTATION – на консультацию; REHABILITATION – на восстановительное лечение; HOSPITALIZATION – на осмотр с целью госпитализации; RESEARCH – на исследование; CONSULTING_ROOM – в консультационный кабинет; OUTPATIENT_VISIT – на поликлинический прием PROF_EXAMINATION – на проф. осмотр VACCINATION – на осмотр перед вакцинацией VACCINATION_ROOM – в кабинет вакцинации TELEMED - онлайн-	

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
				консультация (телемедицина)	
Дата начала действия направления	start_date	Н	TIMESTAMP(6)	Формат: YYYY-MM-DD	Да
Дата окончания действия направления	end_date	Н	TIMESTAMP(6)	Формат: YYYY-MM-DD	Да
Идентификатор структурного подразделения МО, в которое выдано направление	to_mo_id	Н	VARCHAR (512)	Значение уник.идентификатора МО или СП МО, в которому привязан МР (тип ресурса= SPECIALIST) соответствует значению в ФРМР	Да
Код профиля отделения куда выписано направление	to_profile_code	Н	VARCHAR (512)		Да
Наименование кода профиля отделения куда выписано направление	to_profile_name	Н	VARCHAR (512)		Да
Наименование услуги	to_service_name	Н	VARCHAR (512)	Обязательно, если тип	Да

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
				направления "На исследование", type = RESEARCH.	
Наименование должности	to_post_name	Н	VARCHAR (512)	Обязательно, если тип направления "На консультацию", type = CONSULTATION.	Да
Код медицинской услуги, на которую выдано направление	to_service_id	Н	VARCHAR (512)	Обязательно, если тип направления "На исследование", type = RESEARCH.	Да
Идентификатор структурного подразделения МО, выдавшее направление	from_mo_id	Н	VARCHAR (512)	Значение уник.идентификатора МО или СП МО, в котором привязан МР (тип ресурса= SPECIALIST) соответствует значению в ФРМР	Нет
Идентификатор ресурса, выдавшего направление	from_resource_id	Н	VARCHAR (512)		Нет
Статус направления	status	Н	VARCHAR (512)	AVAILABLE – доступно; RECORDED – записано; CANCELED – отменено	Да

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Дата создание записи	create_ts	Н	TIMESTAMP(6)		Нет
Дата изменения записи	update_ts	Н	TIMESTAMP(6)		Нет
Идентификатор ресурса, к которому выдано направление	to_resource_id	Н	VARCHAR (512)	Обязательно, если тип направления type = CONSULTATION	Нет

Unaccessable_period: в таблице содержатся сведения о периоде и причине недоступности ресурса.

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи	period_id	О	INTEGER NOT NULL		Да
Идентификатор ИС	rmis_id	Н	VARCHAR (512)	Идентификатор ИС, зарегистрированной в ФЭР. Выдается после регистрации ИС в тестовой/промышленной среде ФЭР. Пример: conc-5015-rmis	Нет

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Идентификатор ресурса	resource_id	Н	VARCHAR (512)		Да
Причина	reason	Н	VARCHAR (512)	VACATION – в отпуске; SICK_LEAVE – на больничном; REPAIR – на ремонте.	Да
Начало периода	beg_date	Н	TIMESTAMP(6)		Да
Конец периода	end_date	Н	TIMESTAMP(6)		Да
Дата создание записи	create_ts	Н	TIMESTAMP(6)		Нет
Дата изменения записи	update_ts	Н	TIMESTAMP(6)		Нет

Profilecode_resource: в таблице содержатся сведения о профиле ресурса.

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи	id	О	VARCHAR (512) NOT NULL		Да
Идентификатор ИС	rmis_id	Н	VARCHAR (512)	Идентификатор ИС, зарегистрированной в ФЭР.	Нет

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
				Выдается после регистрации ИС в тестовой/промышленной среде ФЭР. Пример: conc-5015-rmis	
Идентификатор ресурса	resource_id	Н	VARCHAR (512)		Да
Код профиля отделения, к которому привязан ресурс	profile_code	Н	VARCHAR (512)		Да
Наименование профиля	profile_name	Н	VARCHAR (512)		Да
Дата создание записи	create_ts	Н	TIMESTAMP(6)		Нет
Дата изменения записи	update_ts	Н	TIMESTAMP(6)		Нет

Observation: в таблице содержатся сведения о карте диспансерного наблюдения.

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный	id	О	VARCHAR (512)		Да

идентификатор записи			NOT NULL		
Идентификатор ИС	rmis_id	Н	VARCHAR (512)	Идентификатор ИС, зарегистрированной в ФЭР. Выдается после регистрации ИС в тестовой/промышленной среде ФЭР. Пример: conc-5015-rmis	Нет
Уникальный идентификатор пациента	patient_id	Н	VARCHAR (512)	Идентификатор пациента	Да
Идентификатор ресурса	resource_id	Н	VARCHAR (512)	Идентификатор врача	Да
Информация о статусе карты	status	Н	BOOLEAN	Значение TRUE, если: Карта учета актуальная. То есть, если PersonDisp_begDate (дата начала диспансерного наблюдения пациента) меньше или равно текущей дате И PersonDisp_endDate (дата окончания диспансерного наблюдения пациента) не заполнено или больше текущей даты. Карта учета не удалена.	Да

				То есть поле Persondisp_deleted = 1 или NULL (актуальность). Если у пациента существует несколько карт учета, то значение TRUE проставляется всем актуальным картам.	
Возрастная категория пациента	type_age	H	VARCHAR (512)	ADULT – Если 1 (Взрослое население) CHILD – Если 2 (Детское население) Null – Если 3 (Смешанный прием) или Null	Да
Повод для обращения (диагноз)	d_observation_reason	H	VARCHAR (512)	Код и наименование диагноза	Нет
Дата создания записи	create_ts	H	TIMESTAMP(6)		Нет
Дата изменения записи	update_ts	H	TIMESTAMP(6)		Нет

MedCase - Информация об открытых случаях лечения, в данной таблице будет храниться инфо об открытых случаях лечения.

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
--------------	---------	----------------	------------	-------------	-------------------------------------

Уникальный идентификатор записи	id	O	VARCHAR (512) NOT NULL	В инф. взаимодействии с Федеральной витриной расписаний используется составной ключ 'id + rmis_id'	Нет
Идентификатор ИС	rmis_id	H	VARCHAR (512)	Идентификатор ИС, зарегистрированной в ФЭР. Выдается после регистрации ИС в тестовой/промышленной среде ФЭР. Пример: conc-5015-rmis	Нет
Уникальный идентификатор пациента	patient_id	O	VARCHAR (512)	Идентификатор пациента	Да
Дата время открытия случая	open_date	H	TIMESTAMP(6)		Да
Признак закрытия случая	is_active	H	BOOLEAN	True – случай открыт; False – случай закрыт.	Да
Дата создания записи	create_ts	H	TIMESTAMP(6)		Нет
Дата изменения записи	update_ts	H	TIMESTAMP(6)		Нет

MedVisit - Информация о посещениях в рамках открытого случая, в данной таблице будет храниться инфо об очных посещениях (визитах)

Наименование	Атрибут	Обязательность	Тип данных	Комментарий	Влияние на отображение формы услуги
Уникальный идентификатор записи	id	O	VARCHAR (512)	В инф. взаимодействии с Федеральной витриной расписаний используется составной ключ 'id + rmis_id'	Нет
Идентификатор ИС	rmis_id	H	VARCHAR (512)	Идентификатор ИС, зарегистрированной в ФЭР. Выдается после регистрации ИС в тестовой/промышленной среде ФЭР. Пример: conc-5015-rmis	Нет
Уникальный идентификатор ресурса	resource_id	O	VARCHAR (512)		Да
Идентификатор открытого случая	med_case_id	O	VARCHAR (512)		Да
Признак актуальности для ТМК	tmk_available	H	BOOLEAN	Т – очное посещение доступно для последующей записи на ТМК F - очное посещение	Да

				недоступно для последующей записи на ТМК	
Дата создание записи	create_ts	Н	TIMESTAMP(6)		Нет
Дата изменения записи	update_ts	Н	TIMESTAMP(6)		Нет

Monitoring: в таблице содержатся сведения о мониторинге состоянии Витрины данных.

Наименование	Атрибут	Обязательность	Тип данных	Комментарий
Идентификатор записи	id	О	INT NOT NULL	Одна строка. Принимает значение 1
Идентификатор записи таблицы	monitoring_id	Н	INT	Одна строка. Принимает значение 1
Идентификатор ИС	rmis_id	Н	VARCHAR (512)	Идентификатор ИС, зарегистрированной в ФЭР. Выдается после регистрации ИС в тестовой/промышленной среде ФЭР. Пример: conc-5015-rmis

А.2 Сведения о результатах контроля качества расписания

А.2.1 Описание логической структуры объектов данных результатов валидации

Таблица validation: в таблице содержатся сведения о результатах контроля качества расписания.

Наименование	Атрибут	Тип	Обязательность	Описание
Идентификатор записи	id	varchar(45) NOT NULL	O	
Идентификатор ИС	rmis_id	varchar(45) NOT NULL	O	
Идентификатор витрины расписаний	datamart_id	varchar(45) NOT NULL	O	
Статус	status	varchar(45) NOT NULL	H	Возможные значения: OK - проверка выполнена, ошибок нет ERROR - проверка выполнена, есть ошибки
Тип ошибки	error_code	varchar(45) NOT NULL	H	Возможные значения: RESOURCE_ERROR - ошибка ресурса MO_ERROR - ошибка МО SERVICE_ERROR - ошибка услуги SLOT_ERROR - ошибка слота PATIENT_ERROR - ошибка пациента BOOK_ERROR - ошибка записи REFERAL_ERROR - ошибка направления ATTACHMENT_ERROR - ошибка прикрепления

Код группы правил, манифест	manifest_id	varchar(45) NOT NULL	H	
Код правила	rule_id	varchar(45) NOT NULL	H	
Дата проверки	validation_ts	timestamp(6)	H	
Название ошибки	name	varchar(512)	H	
Описание ошибки	description	varchar(512)	H	
Действия по исправлению ошибки	to_fix	varchar(512)	H	
Идентификатор родительского МО	parent_id	varchar(45)	H	
Уникальный идентификатор подразделения	branch_id	varchar(45)	H	
Идентификатор ресурса	resource_id	varchar(45)	H	
Идентификатор МР	mr_id	varchar(45)	H	
Идентификатор услуги	service_id	varchar(45)	H	
Идентификатор слота	slot_id	varchar(45)	H	
Идентификатор пациента	patient_id	varchar(45)	H	
Идентификатор записи	book_id	varchar(45)	H	
Идентификатор направления	referral_id	varchar(45)	H	
Идентификатор прикрепления	attachment_id	varchar(45)	H	
Дата создания записи	create_ts	timestamp(6)	H	

Дата изменения записи	update_ts	timestamp(6)	Н	
-----------------------	-----------	--------------	---	--

ПРИЛОЖЕНИЕ Б

ОПИСАНИЕ ПРАВИЛ ПРОВЕРОК, ОСУЩЕСТВЛЯЕМЫХ В КОМПОНЕНТЕ «МОДУЛЬ УПРАВЛЕНИЯ БИЗНЕС- ПРАВИЛАМИ» ФЭР ЕГИСЗ

Перечень проверок качества данных:

Группа правил	Правило	Краткое описание
Атрибутивный состав расписания	Заполнение и формат обязательных полей таблиц сведений о расписании	Тип и значение атрибутов соответствует требованиям, предъявляемым к Сведениям для наполнения региональных расписаний данных (приложение А к настоящему документу). Например, «Уникальный идентификатор ресурса МО» имеет значение 'varchar (45) NOT NULL'
МО	Уникальный идентификатор медицинской организации имеет корректное значение	Уникальный идентификатор (OID) МО, указанный в расписании, должен существовать в ФРМО
МО	Медицинская организация имеет корректное название	Наименование медицинской организации, используемое в расписании, должно соответствовать значению в ФРМО
Ресурсы МО	Тип ресурса медицинской организации соответствует допустимым значениям	Тип ресурса имеет одно из значений: SPECIALIST, MACHINE, ROOM.
Ресурсы МО	Уникальный идентификатор медицинского работника имеет корректное значение	OID МР существует в ФРМР, OID МР принадлежит медицинскому работнику, Значение атрибута ресурса «Уникальный идентификатор структурного подразделения»

		соответствует значению в ФРМР (в сведениях о расписании МР привязан к той же МО или СП МО, что и в ФРМР)
Ресурсы МО	Должность медицинского работника соответствует значению из ФНСИ	Наименование должности должно быть указано в соответствии со справочником ФНСИ «ФРМР. Должности медицинского персонала» (OID 11.2.643.5.1.13.13.99.2.181)
Ресурсы МО	Специальность медицинского работника соответствует значению из ФНСИ	Код специальности должен быть указан в соответствии со справочником ФНСИ «Номенклатура специальностей специалистов со средним, высшим и послевузовским медицинским и фармацевтическим образованием в сфере здравоохранения» (OID 1.2.643.5.1.13.13.11.1066)
Услуги МО	Код услуги имеет корректное значение	Код услуги в расписании имеет значение в соответствии со справочником ФНСИ «Номенклатура медицинских услуг» (1.2.643.5.1.13.13.11.1070)
Услуги МО	Название услуги имеет корректное значение	Название услуги в расписании имеет значение в соответствии со справочником ФНСИ «Номенклатура медицинских услуг» (1.2.643.5.1.13.13.11.1070)

ПРИЛОЖЕНИЕ В

ОБЩЕЕ ОПИСАНИЕ РЕШЕНИЯ

«ЗАПИСЬ НА ПРИЕМ К ВРАЧУ»

В.1 Бизнес-сценарий решения «Запись на прием к врачу»

У каждой ГИСЗ субъекта РФ есть своя Витрина, позволяющая пользователям ЕПГУ записаться на прием ко врачу, процедуры, исследования и т.п.

Каждая услуга (прием врача, процедура, исследование и т.п.) является «слотом» (определяющим специалиста, клинику, кабинет, дату и время). В самом начале все слоты свободны. Когда пользователь записывается на услугу, выбранный слот бронируется этим пользователем. Существует ряд ограничений на запись: мужчина не может записаться к гинекологу, взрослый не может записаться к педиатру и другие ограничения. Все эти ограничения контролируются.

Занять слот можно как через ЕПГУ, который взаимодействует с Витриной через ПОДД и через ГИСЗ субъекта РФ (терминал или регистратура в поликлинике, врач на приеме и т.п.). При этом должны исключаться конфликты двойной записи на один слот.

Кроме задачи бронирования слотов Витрина должна позволять обновлять другие данные, например, расписание врачей, доступность слотов и пр., без остановки сервиса бронирования.

В.1.1 Архитектура решения

ГИСЗ субъекта РФ из своей внутренней БД выгружает данные в Витрину – периодический ETL процесс обновления расписания и т.п.

- 1) Пользователь ЕПГУ через ПОДД просматривает расписание и текущее состояние слотов в Витрине.
- 2) Пользователь ЕПГУ через ПОДД посылают запрос бронирования выбранного слота, Адаптер бронирования запрашивает проверку доступности бронирования у Контроллера бронирования, который проверяет возможность бронирования (наличие, статус, допустимость бронирования) по своей БД, записывает в нее обновленный статус слота и возвращает результат проверки в Адаптер ПОДД.

Адаптер ПОДД, получив ответ обновляет Витрину и возвращает ответ в ЕПГУ через ПОДД.

- 3) Контроллер бронирования (как часть МИС) периодически (реже, чем «8», но чаще, чем «2») обновляет статус слотов в Витрине – обновление слотов (отмена приема и т.п., отказ в ранее произведенном бронировании, обновление броней, выполненных в РМИС и т.п.).

В.2 Перечень регламентированных запросов

Получить записи пациента (по полученному на шаге 1 patient_id)
getBooksByPatientId Пример: <pre> SELECT * FROM misdm02.5.0.Book INNER JOIN misdm02.5.0.Slot ON misdm02.5.0.Slot."Id" = misdm02.5.0.Book."slot_Id" AND misdm02.5.0.Book."patient_id" = ? INNER JOIN misdm02.5.0.Resource ON misdm02.5.0.Resource."id" = misdm02.5.0.Slot."resource_id" INNER JOIN misdm02.5.0.MO ON misdm02.5.0.MO."id" = misdm02.5.0.Resource."mo_id" </pre>
Поиск доступных слотов по post_id и за определенный период (14 дней) для конкретного patient_id
geslotsallmobypostid Пример: <pre> SELECT * FROM misdm02.4.0.Slot INNER JOIN misdm02.4.0.Resource ON misdm02.4.0.Resource."id" = misdm02.4.0.Slot."resource_id" AND misdm02.4.0.Resource."post_Id" = ? AND misdm02.4.0.Slot."visitTime" >= ? AND misdm02.4.0.Slot."visitTime" <= ? AND misdm02.4.0.Slot."status" in ('AVAILABLE', 'RESERVED', 'RECORDED') </pre>

```

AND misdm02.4.0.Resource.enabled in ('ATTACHED_REG', 'ATTACHED_MO', 'ALL')
INNER JOIN misdm02.4.0.MO ON misdm02.4.0.MO."id" = misdm02.4.0.Resource."mo_id"
AND misdm02.4.0.mo.enabled in ('ATTACHED_REG', 'ATTACHED_MO')
INNER JOIN misdm02.4.0.Attachment
ON misdm02.4.0.Attachment."mo_id" = misdm02.4.0.Resource."mo_id"
AND misdm02.4.0.Attachment."patient_id" = ?
AND misdm02.4.0.Attachment."status" = 'DONE'
LEFT JOIN
( SELECT misdm02.4.0.Unaccessible_period."period_id", misdm02.4.0.Unaccessible_
period."resource_id", misdm02.4.0.Unaccessible_period."Reason", misdm02.4.0.Una
ccessible_period."beg_date", COALESCE( misdm02.4.0.Unaccessible_period."end_dat
e", CURRENT_TIMESTAMP + INTERVAL '14' DAY ) "end_date"
FROM misdm02.4.0.Unaccessible_period ) AS up
ON misdm02.4.0.slot."resource_id" = up."resource_id"
AND misdm02.4.0.slot."visittime" between up."beg_date"
AND up."end_date"
WHERE up."resource_id" is null
UNION ALL
SELECT * FROM misdm02.4.0.Slot
INNER JOIN misdm02.4.0.Resource
ON misdm02.4.0.Resource."id" = misdm02.4.0.Slot."resource_id"
AND misdm02.4.0.Resource."post_Id" = ?
AND misdm02.4.0.Slot."visitTime" >= ?
AND misdm02.4.0.Slot."visitTime" <= ?
AND misdm02.4.0.Slot."status" in ('AVAILABLE', 'RESERVED', 'RECORDED')
AND misdm02.4.0.Resource.enabled in ('ATTACHED_REG', 'ATTACHED_MO', 'ALL')
INNER JOIN misdm02.4.0.MO
ON misdm02.4.0.MO."id" = misdm02.4.0.Resource."mo_id"
AND misdm02.4.0.MO.enabled = 'ALL'
LEFT JOIN (SELECT misdm02.4.0.Unaccessible_period."period_id", misdm02.4.0.Unac
cessible_period."resource_id", misdm02.4.0.Unaccessible_period."Reason", misdm0
2.4.0.Unaccessible_period."beg_date", COALESCE( misdm02.4.0.Unaccessible_period
."end_date", CURRENT_TIMESTAMP + INTERVAL '14' DAY ) "end_date"
FROM misdm02.4.0.Unaccessible_period ) AS up
ON misdm02.4.0.slot."resource_id" = up."resource_id"
AND misdm02.4.0.slot."visittime" between up."beg_date"
AND up."end_date"
WHERE up."resource_id" is null

```


Поиск доступных слотов по идентификатору специальности в указанном МО (для направлений на консультацию)

getFreeSlotsBy_SpecId_moId

Пример:

```
SELECT * FROM misdm02.4.0.Slot
INNER join misdm02.4.0.Resource
on misdm02.4.0.Slot.resource_id = misdm02.4.0.Resource.id
AND misdm02.4.0.Slot.visittime > LOCALTIMESTAMP + INTERVAL '5' HOUR
AND misdm02.4.0.Slot.status IN ('AVAILABLE','RESERVED','RECORDED')
AND misdm02.4.0.Resource."speciality_id" = ?
AND misdm02.4.0.Resource."mo_id" = ?
INNER join misdm02.4.0.MO on misdm02.4.0.Resource.mo_id = misdm02.4.0.MO.id
LEFT JOIN
(SELECT misdm02.4.0.Unaccessible_period."resource_id", misdm02.4.0.Unaccessible
_period."Reason", misdm02.4.0.Unaccessible_period."beg_date", misdm02.4.0.Unacc
essible_period."period_id", misdm02.4.0.Unaccessible_period."update_ts", misdm0
2.4.0.Unaccessible_period."create_ts", COALESCE( misdm02.4.0.Unaccessible_perio
d."end_date", CURRENT_TIMESTAMP + INTERVAL '14' DAY ) "end_date"
FROM misdm02.4.0.Unaccessible_period ) AS up
ON misdm02.4.0.Slot."resource_id" = up."resource_id"
AND misdm02.4.0.Slot."visitTime" between up."beg_date"
AND up."end_date" WHERE up."resource_id" is null
```

Поиск доступных слотов по коду услуги в указанном МО (для направлений на исследование)

getFreeSlotsByServiceCodemoId

Пример:

```
SELECT * FROM misdm02.4.0.Slot
inner join misdm02.4.0.Resource
on misdm02.4.0.Slot.resource_id = misdm02.4.0.Resource.id
AND misdm02.4.0.Slot.visittime > (LOCALTIMESTAMP + INTERVAL '5' HOUR)
AND misdm02.4.0.Slot.status in ('AVAILABLE','RESERVED','RECORDED')
AND misdm02.4.0.Resource."mo_id" = ?
inner join misdm02.4.0.MO on misdm02.4.0.Resource.mo_id = misdm02.4.0.MO.id
inner join misdm02.4.0.Service
on misdm02.4.0.Resource.id = misdm02.4.0.Service.resource_id
```

```

AND misdm02.4.0.Service."Code" = ?
INNER JOIN misdm02.4.0.Profilecode_resource
ON misdm02.4.0.Profilecode_resource."Resource_id"=misdm02.4.0.Resource.id
AND misdm02.4.0.Profilecode_resource."Profile_code" = ?

```

Поиск доступных слотов по resource_id в выбранном МО

getFreeSlotsByResourceId_moId

Пример:

```

SELECT * FROM misdm02.4.0.Slot
INNER JOIN misdm02.4.0.Resource
ON misdm02.4.0.Resource.id = misdm02.4.0.Slot.resource_id
AND misdm02.4.0.Slot.resource_id = ?
AND misdm02.4.0.Slot."status" in ('AVAILABLE', 'RESERVED')
AND misdm02.4.0.Slot.visitTime >= ?
AND misdm02.4.0.Slot.visitTime <= ?
and misdm02.4.0.Resource.mo_id = ?
INNER JOIN misdm02.4.0.MO ON misdm02.4.0.MO.id = misdm02.4.0.Resource.mo_id
LEFT JOIN (SELECT misdm02.4.0.Unaccessible_period."resource_id", misdm02.4.0.Unaccessible_period."Reason", misdm02.4.0.Unaccessible_period."beg_date", misdm02.4.0.Unaccessible_period."period_id", misdm02.4.0.Unaccessible_period."update_ts", misdm02.4.0.Unaccessible_period."create_ts", COALESCE( misdm02.4.0.Unaccessible_period."end_date", CURRENT_TIMESTAMP + INTERVAL '14' DAY ) "end_date"
FROM misdm02.4.0.Unaccessible_period ) AS up
ON misdm02.4.0.Slot."resource_id" = up."resource_id"
AND misdm02.4.0.Slot."visitTime" between up."beg_date"
AND up."end_date"
WHERE up."resource_id" is null

```

Поиск доступных медорганизаций по patient_id

getMOByPatientId

Пример:

```

SELECT misdm02.4.0.MO.*
FROM misdm02.4.0.MO
INNER JOIN misdm02.4.0.Attachment

```

```

ON misdm02.4.0.Attachment.mo_id = misdm02.4.0.MO.id
WHERE misdm02.4.0.MO.enabled in ('ATTACHED_MO', 'ATTACHED_REG')
AND misdm02.4.0.attachment.status = 'DONE'
AND misdm02.4.0.attachment.patient_id = ?
UNION ALL
SELECT *
FROM misdm02.4.0.MO
WHERE misdm02.4.0.MO.enabled = 'ALL'

```

Получение обследований по пациенту

getObservationByPatientId

Пример:

```

SELECT *
from misdm02.4.0.observation
inner join misdm02.4.0.resource
on misdm02.4.0.resource.id = misdm02.4.0.observation.resource_id
and misdm02.4.0.observation.patient_id = ?
and misdm02.4.0.observation.status = 'true'

```

Поиск пациента по Номеру полиса, Дате рождения, Фамилии, Имени

getPatientByOMS_BD_LN_FN

Пример:

```

SELECT misdm02.4.0.Patient."Id", misdm02.4.0.Patient."firstname", misdm02.4.0.P
atient."lastname", misdm02.4.0.Patient."birthdate", misdm02.4.0.Patient."gender
", misdm02.4.0.Patient."create_ts", misdm02.4.0.Patient."norm_lastname", misdm0
2.4.0.Patient."norm_firstname", misdm02.4.0.Patient."OMS_region", misdm02.4.0.P
atient."middlename", misdm02.4.0.Patient."OMS_Number", misdm02.4.0.Patient."SNI
LS", misdm02.4.0.Patient."update_ts", misdm02.4.0.Patient."norm_middlename"
FROM misdm02.4.0.Patient
WHERE misdm02.4.0.Patient."OMS_Number" = ?
AND misdm02.4.0.Patient."birthdate" = ?
AND misdm02.4.0.Patient."norm_lastname" = ?
AND misdm02.4.0.Patient."norm_firstname" = ?

```

Поиск пациента по Номеру полиса, СНИЛС, Дате рождения

getPatientByOMS_SNILS_Birthday

Пример:

```
SELECT misdm02.4.0.Patient."Id", misdm02.4.0.Patient."birthdate"
FROM misdm02.4.0.Patient
WHERE misdm02.4.0.Patient."OMS_Number" = ?
AND misdm02.4.0.Patient."SNILS" = ?
AND misdm02.4.0.Patient."birthdate" = ?
```

Поиск пациента по Номеру полиса, СНИЛС, Фамилии

getPatientByOMS_SNILS_LastName

Пример:

```
SELECT misdm02.4.0.Patient."Id", misdm02.4.0.Patient."firstname", misdm02.4.0.P
atient."lastname", misdm02.4.0.Patient."birthdate", misdm02.4.0.Patient."gender
", misdm02.4.0.Patient."create_ts", misdm02.4.0.Patient."norm_lastname", misdm0
2.4.0.Patient."norm_firstname", misdm02.4.0.Patient."OMS_region", misdm02.4.0.P
atient."middlename", misdm02.4.0.Patient."OMS_Number", misdm02.4.0.Patient."SNI
LS", misdm02.4.0.Patient."update_ts", misdm02.4.0.Patient."norm_middlename"
FROM misdm02.4.0.Patient
WHERE misdm02.4.0.Patient."norm_lastname" = ?
AND misdm02.4.0.Patient."SNILS" = ?
AND misdm02.4.0.Patient."OMS_Number" = ?
```

Поиск пациента по Номеру полиса, СНИЛС, Имени, Отчеству

getPatientByOMS_SNILS_FN_MN

Пример:

```
SELECT misdm02.4.0.Patient."Id", misdm02.4.0.Patient."firstname", misdm02.4.0.P
atient."lastname", misdm02.4.0.Patient."birthdate", misdm02.4.0.Patient."gender
", misdm02.4.0.Patient."norm_lastname", misdm02.4.0.Patient."norm_firstname", m
isdm02.4.0.Patient."OMS_region", misdm02.4.0.Patient."middlename", misdm02.4.0.
Patient."OMS_Number", misdm02.4.0.Patient."SNILS", misdm02.4.0.Patient."norm_mi
ddlename"
FROM misdm02.4.0.Patient
```

```
WHERE misdm02.4.0.Patient."OMS_Number" = ?
AND misdm02.4.0.Patient."SNILS" = ?
AND misdm02.4.0.Patient."norm_firstname" = ?
AND misdm02.4.0.Patient."norm_middlename" = ?
```

Поиск пациента по СНИЛС, Дате рождения, Фамилии, Имени

getPatientBySNILS_BD_FN

Пример:

```
SELECT misdm02.4.0.Patient."Id", misdm02.4.0.Patient."firstname", misdm02.4.0.P
atient."lastname", misdm02.4.0.Patient."birthdate", misdm02.4.0.Patient."gender
", misdm02.4.0.Patient."create_ts", misdm02.4.0.Patient."norm_lastname", misdm0
2.4.0.Patient."norm_firstname", misdm02.4.0.Patient."OMS_region", misdm02.4.0.P
atient."middlename", misdm02.4.0.Patient."OMS_Number", misdm02.4.0.Patient."SNI
LS", misdm02.4.0.Patient."update_ts", misdm02.4.0.Patient."norm_middlename"
FROM misdm02.4.0.Patient
WHERE misdm02.4.0.Patient."SNILS" = ?
AND misdm02.4.0.Patient."birthdate" = ?
AND misdm02.4.0.Patient."norm_lastname" = ?
AND misdm02.4.0.Patient."norm_firstname" = ?
```

Поиск доступных направлений по patient_id

getReferralByPatientId

Пример:

```
SELECT *
FROM misdm02.4.0.Referral
INNER JOIN misdm02.4.0.MO
ON misdm02.4.0.MO."id" = misdm02.4.0.Referral."to_mo_id"
WHERE misdm02.4.0.Referral."patient_id" = ?
AND (end_date is null or end_date >= LOCALTIMESTAMP + INTERVAL '5' HOUR )
```

Поиск доступных ресурсов по медорганизации mo_id, выбранный или определенный ранее

getResourcesByMOid

Пример:

```
SELECT *
FROM misdm02.4.0.Resource
INNER JOIN misdm02.4.0.MO ON misdm02.4.0.MO."id" = misdm02.4.0.Resource."mo_id"
AND misdm02.4.0.Resource."mo_id" = ?
AND misdm02.4.0.Resource."enabled" in ('ATTACHED_MO', 'ATTACHED_REG', 'ALL')
LEFT JOIN misdm02.4.0.slot on misdm02.4.0.slot."resource_id"='0'
WHERE EXISTS (SELECT misdm02.4.0.slot.resource_id
FROM misdm02.4.0.slot
WHERE misdm02.4.0.slot.tag_type in ('ORDINARY', 'EPGU', 'D_OBSERVATION', 'REGIS
TRY'))
AND misdm02.4.0.slot.resource_id = misdm02.4.0.Resource.id
AND misdm02.4.0.slot.visittime >= (LOCALTIMESTAMP + INTERVAL '5' HOUR))
```

Поиск доступных специалистов по patient_id

GetSpecialistsByPatientId

Пример:

```
SELECT misdm02.4.0.Resource.*
FROM misdm02.4.0.Resource
INNER JOIN misdm02.4.0.MO on misdm02.4.0.MO.id = misdm02.4.0.Resource.mo_id
AND misdm02.4.0.Resource."type" = 'SPECIALIST'
AND misdm02.4.0.Resource."enabled" IN ('ATTACHED_REG', 'ATTACHED_MO', 'ALL')
AND misdm02.4.0.Resource."post_id" is not null
AND misdm02.4.0.MO.enabled IN ('ATTACHED_REG', 'ATTACHED_MO')
INNER JOIN misdm02.4.0.Attachment
on misdm02.4.0.Attachment.mo_id = misdm02.4.0.MO.id
AND misdm02.4.0.Attachment.status = 'DONE'
AND misdm02.4.0.Attachment.patient_id = ?
LEFT JOIN misdm02.4.0.Slot ON misdm02.4.0.Slot.id = '-1'
UNION ALL
SELECT *
FROM misdm02.4.0.Resource
LEFT JOIN misdm02.4.0.Slot ON misdm02.4.0.Slot."id"='-1'
INNER JOIN misdm02.4.0.MO on misdm02.4.0.MO.id = misdm02.4.0.Resource.mo_id AND
misdm02.4.0.Resource."type" = 'SPECIALIST'
AND misdm02.4.0.Resource."enabled" IN ('ATTACHED_REG', 'ATTACHED_MO', 'ALL')
```

```

AND misdm02.4.0.Resource."post_id" is not null
AND misdm02.4.0.MO.enabled = 'ALL'
WHERE EXISTS
(SELECT 1 FROM misdm02.4.0.Slot
WHERE misdm02.4.0.Slot."tag_type" in ('ORDINARY', 'EPGU', 'D_OBSERVATION', 'REG
ISTRY')
AND misdm02.4.0.Slot."resource_id" = misdm02.4.0.Resource."id"
AND misdm02.4.0.Slot."visittime" >= (LOCALTIMESTAMP + INTERVAL '5' HOUR) )

```

Поиск ресурсов, относящихся к указанному профилю

getProfilecodeResourceByProfileCode

Пример:

```

SELECT misdm02.4.0.Profilecode_resource."Resource_id", misdm02.4.0.Profilecode_
resource."id", misdm02.4.0.Profilecode_resource."Profile_name", misdm02.4.0.Pro
filecode_resource."Profile_code"
FROM misdm02.4.0.Profilecode_resource
WHERE misdm02.4.0.Profilecode_resource."Profile_code" = ?

```

Поиск недоступных врачей по post_id в указанный день

getresourcelistbypostid_dates

Пример:

```

SELECT *
FROM misdm02.4.0.Unaccessible_period
INNER JOIN misdm02.4.0.Resource
ON misdm02.4.0.Resource.id = misdm02.4.0.Unaccessible_period.resource_id
INNER JOIN misdm02.4.0.MO ON misdm02.4.0.MO.id = misdm02.4.0.Resource.mo_id
INNER JOIN misdm02.4.0.attachment
ON misdm02.4.0.Attachment.mo_id = misdm02.4.0.Resource.mo_id
AND misdm02.4.0.Attachment.patient_id = ?
AND misdm02.4.0.Attachment."status" = 'DONE'
WHERE misdm02.4.0.Resource."post_Id" = ? ) AS fulltable
WHERE fulltable.beg_date <= ?
AND fulltable.end_date >= ?

```

Поиск недоступных кабинетов/аппаратов по коду услуги в указанный день

getlistresourcesbycode_dates

Пример:

```
SELECT * from misdm02.4.0.Unaccessible_period
inner join misdm02.4.0.Resource
on misdm02.4.0.Resource.id = misdm02.4.0.Unaccessible_period.resource_id
inner join misdm02.4.0.MO on misdm02.4.0.MO.id = misdm02.4.0.Resource.mo_id
INNER JOIN misdm02.4.0.Service
on misdm02.4.0.Service.resource_id = misdm02.4.0.Unaccessible_period.resource_id
and misdm02.4.0.Service."Code" = ? ) as fulltable
WHERE fulltable.beg_date <= ?
and fulltable.end_date >= ?
```

В.3 Пример спецификации OpenAPI

Загрузка спецификаций OpenAPI осуществляется в рамках выполнения запроса регистрации API. ПОДД СМЭВ поддерживает спецификации OpenAPI версии 3.0. Спецификации OpenAPI версии 2.0 не поддерживаются.

Используемый формат спецификаций OpenAPI: JSON и YAML.

Допускается использование всех HTTP-методов в спецификации.

Пример спецификации OpenAPI для бронирования слота и отмены:

```
{
  "openapi": "3.0.2",
  "servers": [
    {
      "url": "http://127.0.0.1"
    }
  ],
  "info": {
    «description»: «Спецификация OpenAPI 3.0 для сервиса записи на приём к врачу»,
    «version»: «1.0»,
    «title»: «Запись на приём к врачу – OpenAPI 3.0»
  },
  "paths": {
    "/booking/book": {
```



```

“post”: {
  “summary”: “Бронирование слота”,
  “description”: “Бронирование слота”,
  “operationId”: “book”,
  “requestBody”: {
    “description”: “Запрос”,
    “content”: {
      “application/json”: {
        “schema”: {
          “$ref”: “#/components/schemas/BookRequest”
        }
      }
    },
    “required”: true
  },
  “responses”: {
    “200”: {
      “description”: “Успешная операция”,
      “content”: {
        “application/json”: {
          “schema”: {
            “$ref”: “#/components/schemas/BookResponse”
          }
        }
      }
    },
    “400”: {
      “description”: “Неверные параметры”
    },
    “500”: {
      “description”: “Внутренняя ошибка”
    }
  }
},
“/booking/cancel”: {
  “post”: {
    “summary”: “Отмена брони”,
    “description”: “Отмена брони”,
    “operationId”: “cancel”,
    “requestBody”: {

```

```

    "description": "Запрос",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/CancelRequest"
        }
      }
    },
    "required": true
  },
  "responses": {
    "200": {
      "description": "Успешная операция",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/BookResponse"
          }
        }
      }
    },
    "400": {
      "description": "Неверные параметры"
    },
    "500": {
      "description": "Внутренняя ошибка"
    }
  }
},
"components": {
  "schemas": {
    "BookRequest": {
      "type": "object",
      "properties": {
        "bookId": {
          "type": "string"
        }
      },
      "slotId": {
        "type": "string"
      }
    }
  }
}

```

```

    },
    "patient_id": {
      "type": "string"
    },
    "booking_type": {
      "type": "string"
    },
    "caseNumber": {
      "type": "string"
    },
    "preliminaryReservation": {
      "type": "167ombok167"
    },
    "referral_id": {
      "type": "string"
    },
    "cards_id": {
      "type": "string"
    },
    "email": {
      "type": "string"
    },
    "mobilePhone": {
      "type": "string"
    }
  },
  "required": [
    "bookId",
    "slotId",
    "patient_id"
  ]
},
"BookResponse": {
  "oneOf": [
    { "$ref": "#/components/schemas/BookResponseSuccess" },
    { "$ref": "#/components/schemas/BookResponseError" }
  ]
},
"BookResponseSuccess": {
  "type": "object",
  "properties": {

```

```

“bookExtId”: {
  “type”: “string”
},
“slotId”: {
  “type”: “string”
},
“visitTime”: {
  “type”: “string”
},
“duration”: {
  “type”: “string”
},
“serviceId”: {
  “type”: “string”
},
“organizationId”: {
  “type”: “string”
},
“TMK_url”: {
  “type”: “string”
},
“areaId”: {
  “type”: “string”
},
“queueNumber”: {
  “type”: “string”
},
“pincode”: {
  “type”: “string”
},
“window”: {
  “type”: “string”
},
“status”: {
  “$ref”: “#/components/schemas/Status”
}
},
“required”: [
  “bookExtId”,
  “slotId”,
  “visitTime”,

```

```

    "status"
  ]
},
"BookResponseError": {
  "type": "object",
  "properties": {
    "status": {
      "$ref": "#/components/schemas/Status"
    }
  },
  "required": [
    "status"
  ]
},
"CancelRequest": {
  "type": "object",
  "properties": {
    "bookExtId": {
      "type": "string"
    },
    "patient_id": {
      "type": "string"
    }
  },
  "required": [
    "bookExtId",
    "patient_id"
  ]
},
"Status": {
  "type": "object",
  "properties": {
    "statusCode": {
      "type": "integer"
    },
    "statusMessage": {
      "type": "string"
    }
  },
  "required": [
    "statusCode"
  ]
}

```

```
]
}
}
}
}
```

В.3.1 Бронирование слота

POST /FerDtm/services/fer/booking/book

Описание параметров в запросе:

№	Параметр	Тип данных	Обязательность	Описание
1	bookId	string	О	Идентификатор записи на стороне ЕПГУ
2	slotId	string	О	Идентификатор слота
3	patient_id	string	О	Идентификатор пациента
4	booking_type	string	О	Тип записи (APPOINTMENT / VACCINATION / UDVN_DISPENSARY / D_OBSERVATION / ONLINE_CONSULTING)
5	preliminaryReservation	boolean	О	Переключатель на бронирование/предварительное бронирование true/false (по умолчанию всегда false)
6	caseNumber	string	Н	Идентификатор заявления в ЕПГУ
7	referral_id	string	У	Идентификатор направления. Обязательно при записи по направлению
8	cards_id	string	Н	Идентификатор медицинской карты
9	email	string	Н	Электронная почта пациента
10	mobilePhone	string	Н	Номер телефона пациента

Описание параметров при успешном ответе бронирования:

№	Параметр	Тип данных	Обязательность	Описание
1	type	string	О	Результат бронирования, заполняется значением «BookResponseSuccess»
2	bookExtId	string	О	Идентификатор записи на стороне РМИС Состоит из двух частей: <rmis_id>-<guid> где rmis_id - число из

№	Параметр	Тип данных	Обязательность	Описание
				идентификатора МИС (cons-50116-rmis) guid - уникальный идентификатор, состоит из 32 шестнадцатеричных символов, разделённых дефисами на пять групп по схеме 8-4-4-4-12. Пример: 50116-d01eb415-48b1-4426-81f6-5d027c8105dc
3	slotId	string	O	Идентификатор слота
4	visitTime	string	O	Время посещения
5	duration	string	O	Длительность посещения
6	serviceId	string	H	Идентификатор цели услуги (внутренний параметр услуги на ЕПГУ)
7	organizationId	string	H	Идентификатор подразделения
8	areaId	string	H	Идентификатор помещения
9	queueNumber	string	H	Номер в очереди (если пациент записан не точно по времени)
10	TMK_url	string	H	Ссылка для перехода на ВКС онлайн-консультации Обязательно для услуги Запись на телемедицинскую консультацию
11	status	object	O	Статус ответа
11.1	statusCode	integer	O	Код статуса
11.2	statusMessage	string	H	Информационное сообщение статуса (лучше отправлять)

Описание параметров в ответе для неуспешного бронирования:

№	Параметр	Тип данных	Обязательность	Описание
1	type	string	O	Результат бронирования, заполняется значением «BookResponseError»
2	status	object	O	Статус ответа
2.1	statusCode	integer	O	Код статуса
2.2	statusMessage	string	H	Информационное сообщение статуса (лучше отправлять)

В.3.2 Отмена бронирования выбранного слота

POST /FerDtm/services/fer/booking/cancel

Описание параметров в запросе:

№	Параметр	Тип данных	Обязательность	Описание
1	bookExtId	string	O	Идентификатор записи на стороне РМИС
2	patient_id	string	O	Идентификатор пациента

Описание параметров в ответе для успешной отмены брони:

№	Параметр	Тип данных	Обязательность	Описание
1	type	string	O	Результат отмены бронирования, заполняется значением «BookResponseSuccess»
2	bookExtId	string	O	Идентификатор записи на стороне РМИС
3	slotId	string	O	Идентификатор слота
4	visitTime	string	O	Время посещения
5	duration	string	O	Длительность посещения
6	serviceId	string	H	Идентификатор цели услуги (внутренний параметр услуги на ЕПГУ)
7	organizationId	string	H	Идентификатор подразделения
8	areaId	string	H	Идентификатор помещения
9	queueNumber	string	H	Номер в очереди (если пациент записан не точно по времени)
10	status	object	O	Статус ответа
10.1	statusCode	integer	O	Код статуса
10.2	statusMessage	string	H	Информационное сообщение статуса (лучше отправлять)

Описание параметров в ответе для неуспешной отмены брони:

№	Параметр	Тип данных	Обязательность	Описание
1	type	string	O	Результат отмены бронирования, заполняется значением «BookResponseError»
2	status	object	O	Статус ответа
2.1	statusCode	integer	O	Код статуса
2.2	statusMessage	string	H	Информационное сообщение статуса (лучше отправлять)

В.4 Коды ошибок

http-код	Описание
200	Запрос успешно обработан
400	Ошибка в запросе, информация об ошибке содержится в параметре «error»
403	Нет полномочий на выполнение запроса
404	Результат по заданному идентификатору SQL-запроса не найден
406	Неподдерживаемая версия протокола (после внедрения поддержки обратной совместимости возвращается только для несуществующих версий)
429	ИС УВ временно заблокирована в связи с превышением лимитов
500	Системная ошибка (в связи с принятыми ограничениями по доступности код 500 предполагается только при сбое Агента)
503	Система временно недоступна, возможно повторить запрос через 50 мс.

Коды ошибок в ответе на запрос:

Код ошибки status.statusCode	Описание ошибки	Отображаемый текст для пользователя
0		Переход на экран успешной записи – не отображается пользователю
11	Бирка к врачу занята или заблокирована Слот заблокирован другим пациентом Слот занят или заблокирован Слот заблокирован другим пациентом	Выбранное время занято Вы не успели подтвердить бронирование. Выберите другое время
12	Есть бирка на то же время	У вас есть запись к другому специалисту на то же время У вас уже есть запись к другому специалисту на то же время. Отмените ее или выберите другое время для записи
13	Пациент имеет предстоящую запись к данному врачу или врачу данной специальности Есть бирка на тот же день к тому же врачу	У вас есть активная запись к специалисту на тот же день Записаться снова можно после того, как эта запись станет недействительна. Вы можете отменить её или записаться на другой день
14	Бирка только для пациентов ДО ГОДА Бирка только для пациентов СТАРШЕ ГОДА Бирка только для детей до 18 лет Бирка только для пациентов от 18 лет	Слот не соответствует по возрасту Вы не можете записаться к этому специалисту, так как он ведет прием для другой возрастной категории пациентов
15	Слот аннулирован	Выбранное время недоступно Данный слот был аннулирован. Пожалуйста, выберите другое время
17	Пациент не найден	Пользователь не найден Проверьте правильность указанных данных в

Код ошибки status.statusCode	Описание ошибки	Отображаемый текст для пользователя
		личном кабинете
18	Бирка не соответствующего типа Извините, запись невозможна Передано неверное направление. Слот не найден Несоответствие в переданных данных для записи Ошибка снятия блокировки	Внутренняя ошибка сервиса Ошибка при получении данных из медицинской организации
1	Запись с таким ИД уже отменена	Ошибка при отмене записи Произошла ошибка. Попробуйте снова или зайдите позже. Если ничего не изменится, напишите в службу поддержки

В.5 Примеры запросов/ответов

В.5.1 Получение данных по регламентированным запросам

URL-адрес для выполнения обращений к REST-интерфейсу имеет следующий формат:

http://<адрес>:<порт>/query, где:

- <адрес> – IP-адрес Агента ПОДД СМЭВ;
- <порт> – порт, на котором развернут REST-интерфейс.

Параметры запроса:

№	Параметр	Тип	Обязательность	Описание
header				
1	Content-Type	string	Да	application/x-www-form-urlencoded; charset=utf-8
2	Accept-version	string	Да	Основная (major) часть версии (сейчас 1)
query				
	async	boolean	Нет	Для синхронного режима выполнения запроса параметр должен отсутствовать или иметь значение False
body				
1	priority	string	Да	Приоритет запроса. Варианты: — NORMAL; — HIGH
2	timeout	string	Нет	Предельное время ожидания выполнения запроса.

				В случае отсутствия параметра в запросе таймаут не выставляется
3	sql	string	Да	Текст SQL-запроса к Витринам Поставщиков данных
4	params	array	Нет	Параметры запроса
4.1	type	string	Да	Тип параметра
4.2	value	string	Да	Значение параметра
5	maxrows	string	Нет	Максимальное количество возвращаемых записей таблицы ответа. Если не задан, возвращаются все записи

Пример запроса:

```
POST «https://10.81.4.30:29354/query
Accept-Version:1
Content-Type:application/x-www-form-urlencoded; encoding=utf-8

sql: "SELECT * FROM misdmXX.GetSpecialistsByPatientId('patient_id')"
priority: "NORMAL"
maxrows: "9999"
```

Параметры ответа с http-код возврата 200:

№	Параметр	Тип	Обязательность	Описание
	responseCode	numeric	Да	Код возврата (HTTP-код)
header				
	Content-Type	string	Да	application/vnd.ru.rtlabs.podd.agent+json; version=1.0; charset=utf-8
body				
1	created_at	dateTime	Да	Время формирования ответа
2	query_id	string	Да	Идентификатор запроса
3	rows	array	Нет	Массив записей таблицы ответа (в случае если результат выполнения запроса в виде таблицы). При задании параметра «maxrows» ограничивается его значением
3.1		array	Да	Массив значений. Возможным значением может быть содержимое файла, закодированное в формате BASE64
4	meta	array	Да	Список полей результата
4.1	name	string	Да	Имя поля

4.2	type	string	Да	Тип поля
-----	------	--------	----	----------

Пример ответа с кодом возврата 200:

```
HTTP/1.1 200 OK
{
  "created_at": "2017-12-15T07:36:-03Z",
  "query_id": "c005a0e7-0d26-4ce0-alfa-10c8bdf4dfc5",
  "meta": [
    {
      "name": "count",
      "type": "INTEGER"
    }
  ],
  «rows»: [
    [
      «4994»
    ]
  ]
}
```

Параметры ответа с кодом возврата отличным от 200:

№	Параметр	Тип	Обязательность	Описание
	responseCode	numeric	Да	Код возврата (HTTP-код)
header				
1	Content-Type	string	Да	application/vnd.ru.rtlabs.podd.agent+json; version=1.0; charset=utf-8
body				
1	query_id	string	Нет	Идентификатор запроса
2	created_at	dateTime	Нет	Время формирования ответа
3	error	string	Нет	Текст ошибки

Пример ответа с кодом возврата, отличным от 200:

```
HTTP/1.1 429 Too Many Requests

{
  "created_at": "2021-09-10T15:23:36Z",
  "query_id": "1ec124b1-1aa6-66d6-9d40-f55438428f56",
  "error": "RuntimeException: Ошибка во входящем потоке : CustomRSocketException:
LIMIT_EXCEEDED: Запросы к Ядру ПОДД временно заблокированы до September 10, 2021 3:24:35 PM
UTC, код причины блокировки=2, подробности: 'Превышен лимит по количеству запросов
lockId=fe462d49-3c5a-4350-8bf6-da155225c52f, userId=e92e3fd4-28d9-48e5-8079-e377b676c9b4
reqCountLimit=10, QueriesStatistic(totalSent=10, totalBytes=2870, totalRows=0,
requestIds={1ec124b0-bce0-613b-9d40-c7c6585b14bc=287B, 1ec124b0-caa5-677c-9d40-
ebd5d2c0409b=287B, 1ec124b0-d464-695d-9d40-bbafa86207b7=287B, 1ec124b0-dcd5-63ae-9d40-
171562fc3a47=287B, 1ec124b0-e493-6a6f-9d40-174dab9e6065=287B, 1ec124b0-ec15-60a0-9d40-
f300de6b4e34=287B, 1ec124b0-f6de-6451-9d40-51926f0b5d81=287B, 1ec124b0-fe95-65e2-9d40-
e5e27f87babb=287B, 1ec124b1-0581-6d43-9d40-6301af351da7=287B, 1ec124b1-0c9c-6ad4-9d40-
dfc04d23e0bc=287B})'."
}
```

В.5.2 Пример запроса/ответа на бронирование

POST /booking/book

Пример запроса

Content-Type: application/json

```

Accept-Version: 1
User-Agent: PostmanRuntime/7.32.2
Accept: */*
Postman-Token: a013767c-8dfb-4661-810d-286cef4b2d17
Host: pgu-uat-feddsnlg.test.gosuslugi.ru
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 209
Cookie: nau=8a6d5c15-ed41-49f0-527c-8440998cf7d1; userSelectedLanguage=ru; nau=9fbb2dbc-e521-57ea-11fc-dd7b85902d65; userSelectedLanguage=ru

{
  "bookId": "92bf98eb-2979-49f4-b36c-0a64f8ce233f",
  "slotId": "CB4C984E-4099-4365-AA00-4797BF2D5C20",
  "patient_id": "20930000047068",
  "booking_type": "APPOINTMENT",
  "preliminaryReservation": false,
  "referral_id": "1234587" --опционально
}

```

Пример успешного ответа

```

HTTP/1.1 200 OK

Date: Fri, 21 Apr 2023 06:47:22 GMT
Content-Type: application/json
Connection: keep-alive
Keep-Alive: timeout=300
Set-Cookie: usi_portal=;Domain=.test.gosuslugi.ru;Path=/;Max-Age=1800
Set-Cookie: usi_portal=CvSHAmRCMelynwAHITsYAg==; expires=Fri, 21-Apr-23 07:19:17 GMT;
domain=.test.gosuslugi.ru; path=/
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff always
Set-Cookie: usi_portal=;Domain=.test.gosuslugi.ru;Path=/;Max-Age=1800
Set-Cookie: usi_portal=CIEDHWRCMXocWbtSeILAg==; expires=Fri, 21-Apr-23 07:17:22 GMT;
domain=.test.gosuslugi.ru; path=/
Set-Cookie:
TS01417566=01474e7625dd85d16395c1c4dcad0bd243399333049c9180985f9ec25ee44d0131c9010a1665b1f690a77b
59b9cfab5d3e45644893c161dce6e5c654dd3bafb400b255cea0b6b9c36e77d28016e88c1c5400f7e9873d7d50e5b4415
af5a15b40e922ce500387ed1cd333bb0d4340f9242742ebdad7; Path=/; Domain=.pgu-uat-
feddsnlg.test.gosuslugi.ru
Transfer-Encoding: chunked

{
  "type": "BookResponseSuccess",
  "bookExtId": "00000000-0000-0001-0000-000001017307",
  "slotId": "7E33091C-0962-4C26-8524-E2960B61B437",
  "visitTime": "2022-12-14T08:00:00+05:00",
  "duration": "40",
  "serviceId": null,
  "organizationId": null,
  "areaId": null,
  "queueNumber": null,
  "pincode": null,
  "window": null,
  "status": {
    "statusCode": 0,
    "statusMessage": null
  }
}

```

Пример ответа с бизнес ошибкой

```

HTTP/1.1 200 OK

Date: Fri, 21 Apr 2023 06:47:22 GMT
Content-Type: application/json
Connection: keep-alive
Keep-Alive: timeout=300
Set-Cookie: usi_portal=;Domain=.test.gosuslugi.ru;Path=/;Max-Age=1800
Set-Cookie: usi_portal=CvSHAmRCMelynwAHITsYAg==; expires=Fri, 21-Apr-23 07:19:17 GMT;
domain=.test.gosuslugi.ru; path=/
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff always

```

```

Set-Cookie: usi_portal=;Domain=.test.gosuslugi.ru;Path=/;Max-Age=1800
Set-Cookie: usi_portal=C1EDHWRCMXocwWbtSeILAg==; expires=Fri, 21-Apr-23 07:17:22 GMT;
domain=.test.gosuslugi.ru; path=/
Set-Cookie:
TS01417566=01474e7625dd85d16395c1c4dcad0bd243399333049c9180985f9ec25ee44d0131c9010a1665b1f690a77b
59b9cfab5d3e45644893c161dce6e5c654dd3bafb400b255cea0b6b9c36e77d28016e88c1c5400f7e9873d7d50e5b4415
af5a15b40e922ce500387ed1cd333bb0d4340f9242742ebdad7; Path=/; Domain=.pgu-uat-
feddsnlb.test.gosuslugi.ru
Transfer-Encoding: chunked

{
  "type": "BookResponseError",
  "status": {
    "statusCode": 11,
    "statusMessage": "Слот занят другим пациентом или заблокирован МО"
  }
}

```

B.5.3 Пример запрос/ответа на отмену бронирования

POST /booking/cancel

Пример запроса

```

Content-Type: application/json
Accept-Version: 1
User-Agent: PostmanRuntime/7.32.2
Accept: */*
Postman-Token: a013767c-8dfb-4661-810d-286cef4b2d17
Host: pgu-uat-feddsnlb.test.gosuslugi.ru
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 209
Cookie: nau=8a6d5c15-ed41-49f0-527c-8440998cf7d1; userSelectedLanguage=ru; nau=9fbb2dbc-e521-57ea-
11fc-dd7b85902d65; userSelectedLanguage=ru

{
  "bookExtId": "00000000-0000-0001-0000-000001014bd8",
  "patient_id": "20930000047068"
}

```

Пример успешного ответа

```

HTTP/1.1 200 OK

Date: Fri, 21 Apr 2023 06:47:22 GMT
Content-Type: application/json
Connection: keep-alive
Keep-Alive: timeout=300
Set-Cookie: usi_portal=;Domain=.test.gosuslugi.ru;Path=/;Max-Age=1800
Set-Cookie: usi_portal=CvSHAmRCMelynwAHITsYAg==; expires=Fri, 21-Apr-23 07:19:17 GMT;
domain=.test.gosuslugi.ru; path=/
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff always
Set-Cookie: usi_portal=;Domain=.test.gosuslugi.ru;Path=/;Max-Age=1800
Set-Cookie: usi_portal=C1EDHWRCMXocwWbtSeILAg==; expires=Fri, 21-Apr-23 07:17:22 GMT;
domain=.test.gosuslugi.ru; path=/
Set-Cookie:
TS01417566=01474e7625dd85d16395c1c4dcad0bd243399333049c9180985f9ec25ee44d0131c9010a1665b1f690a77b
59b9cfab5d3e45644893c161dce6e5c654dd3bafb400b255cea0b6b9c36e77d28016e88c1c5400f7e9873d7d50e5b4415
af5a15b40e922ce500387ed1cd333bb0d4340f9242742ebdad7; Path=/; Domain=.pgu-uat-
feddsnlb.test.gosuslugi.ru
Transfer-Encoding: chunked

{
  "type": "BookResponseSuccess",
  "bookExtId": "00000000-0000-0001-0000-000001017307",
  "slotId": "7E33091C-0962-4C26-8524-E2960B61B437",
  "visitTime": "2022-12-14T08:00:00+05:00",
}

```

```

"duration": "40",
"serviceId": null,
"organizationId": null,
"areaId": null,
"queueNumber": null,
"pincode": null,
>window": null,
"status": {
  "statusCode": 0,
  "statusMessage": null
}

```

Пример ответа с бизнес ошибкой

```

HTTP/1.1 200 OK

Date: Fri, 21 Apr 2023 06:47:22 GMT
Content-Type: application/json
Connection: keep-alive
Keep-Alive: timeout=300
Set-Cookie: usi_portal=;Domain=.test.gosuslugi.ru;Path=/;Max-Age=1800
Set-Cookie: usi_portal=CvSHAmRCMelynwAHITsYAg==; expires=Fri, 21-Apr-23 07:19:17 GMT; domain=.test.gosuslugi.ru; path=/
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff always
Set-Cookie: usi_portal=;Domain=.test.gosuslugi.ru;Path=/;Max-Age=1800
Set-Cookie: usi_portal=C1EDHWRCMXocwWbtSeILAg==; expires=Fri, 21-Apr-23 07:17:22 GMT; domain=.test.gosuslugi.ru; path=/
Set-Cookie: TS01417566=01474e7625dd85d16395c1c4dcad0bd243399333049c9180985f9ec25ee44d0131c9010a1665b1f690a77b59b9cfab5d3e45644893c161dce6e5c654dd3bafb400b255cea0b6b9c36e77d28016e88c1c5400f7e9873d7d50e5b4415af5a15b40e922ce500387ed1cd333bb0d4340f9242742ebdad7; Path=/; Domain=.pgu-uat-feddsnlb.test.gosuslugi.ru
Transfer-Encoding: chunked

{
  "type": "BookResponseError",
  "status": {
    "statusCode": 1,
    "statusMessage": "Запись с ИД записи уже отменена"
  }
}

```

В.6 Тестовые сценарии

В.6.1 Проверка реализации сценария записи себя, ребенка или другого человека» по услуге «Запись на прием к врачу»

Предварительные условия:

- 1) Пользователь авторизован на ЕПГУ с подтвержденной УЗ ФЛ.
- 2) Пользователь имеет прикрепление в одном из пилотных регионов для которых открыта услуга «Записи на прием к врачу»
<https://www.gosuslugi.ru/10700/1/form>
- 3) Пользователь перешел на страницу услуги «записи на прием к врачу» по адресу <https://www.gosuslugi.ru/10700/1/form>

№	Шаг	Ожидаемый результат
	Нажать на кнопку «Начать»	Открылся экран «Кого записать на приём?» в форме квиза с вариантами для выбора: Себя Ребенка Другого человека
1	Выбрать пункт «Себя»	Открылся экран «Выберите способ записи на приём» в форме квиза со списком возможных способов записи: По специальности врача По направлению По медорганизации
2	Вернуться на экран «Кого записать на прием»? и выбрать пункт «Ребенка»	Открылся экран «Укажите данные ребёнка»
3	Выбрать данные ребенка из выпадающего списка и нажать кнопку «Продолжить»	Открылся экран «Выберите способ записи на приём» в форме квиза со списком возможных способов записи: По специальности врача По направлению По медорганизации
4	Вернуться на экран «Кого записать на прием»? и выбрать пункт «Добавить нового ребенка», заполнить форму и нажать на кнопку «Продолжить»	Открылся экран «Выберите способ записи на приём» в форме квиза со списком возможных способов записи: По специальности врача По направлению По медорганизации
5	Вернуться на экран «Кого записать на прием»? и выбрать пункт «Другого человека»	Открылся экран «Укажите данные пациента»
6	Заполнить форму данными другого человека и нажать на кнопку «Продолжить»	Открылся экран «Где посетить врача?»
7	Выбрать регион и нажать на кнопку «Продолжить»	Открылся экран «Выберите способ записи на приём» в форме квиза со списком возможных способов записи: По специальности врача По направлению

№	Шаг	Ожидаемый результат
		По медорганизации

В.6.2 Проверка реализации сценария записи и отмены записи к врачу по направлению по услуге «Запись на прием к врачу»

Предварительные условия:

- 1) Пользователь авторизован на ЕПГУ с подтвержденной УЗ ФЛ.
- 2) Пользователь имеет прикрепление в одном из пилотных регионов для которых открыта услуга «Записи на прием к врачу»
<https://www.gosuslugi.ru/10700/1/form>:
- 3) Пользователь перешел на страницу услуги «записи на прием к врачу» по адресу <https://www.gosuslugi.ru/10700/1/form>

№	Шаг	Ожидаемый результат
	Нажать на кнопку «Начать»	Открылся экран «Кого записать на приём?» в форме квиза с вариантами для выбора: Себя Ребенка Другого человека
1	Выбрать пункт «Себя»	Открылся экран «Выберите способ записи на приём» в форме квиза со списком возможных способов записи: По специальности врача По направлению По медорганизации
2	Выбрать пункт «По направлению»	Открылся экран «Направления» со списком доступных направлений»(Если в методе Referral в ответе получения сведения о направлении, при пустом ответе отображается экран с ошибкой)
3	Выбрать направление и нажать на кнопку «Записаться»	Открылся экран «Выберите дату и врача»
4	Выбрать врача и слот для записи, осуществить запись	Открылся экран «Вы записаны на прием»
5	Отменить запись используя ссылку «отменить запись» (в методе Cancel пришел успешный ответ 0)	Запись успешно отменена

В.6.3 Проверка реализации сценария записи и отмены записи в выбранную медицинскую организацию по услуге «Запись на прием к врачу»

Предварительные условия:

- 1) Пользователь авторизован на ЕПГУ с подтвержденной УЗ ФЛ.
- 2) Пользователь имеет прикрепление в одном из пилотных регионов для которых открыта услуга «Записи на прием к врачу»
<https://www.gosuslugi.ru/10700/1/form>:
- 3) Пользователь перешел на страницу услуги «записи на прием к врачу» по адресу
<https://www.gosuslugi.ru/10700/1/form>

№	Шаг	Ожидаемый результат
	Нажать на кнопку «Начать»	Открылся экран «Кого записать на приём?» в форме квиза с вариантами для выбора: Себя Ребенка Другого человека
1	Выбрать пункт «Себя»	Открылся экран «Выберите способ записи на приём» в форме квиза со списком возможных способов записи: По специальности врача По направлению По медорганизации
2	Выбрать пункт «По медорганизации»	Открылся экран с картой для выбора медицинской организации (если в методе get_МО получен список мо, при пустом ответе экран с ошибкой)
3	Выбрать медицинскую организацию	Открылся экран «Выберите специальность врача» (если в методе Get_Resource пришел список доступных специальностей, если пустой ответ – экран с ошибкой)
4	Выбрать специальность	Открылся экран «Выберите дату и врача» (если в методе GetCountFreeSlot пришли доступные слоты для записи с учетом пола и возраста)
5	Выбрать врача и слот для записи, осуществить запись (если в методе Get_Free_Slot пришли доступные для записи слоты по конкретному врачу)	Открылся экран «Вы записаны на прием» (если в метод book пришел успешный ответ с кодом 0, если пришел код 101,11-18 – экран с ошибкой)
6	Отменить запись используя ссылку «отменить запись» (в методе Cancel пришел успешный ответ 0)	Запись успешно отменена

В.6.4 Проверка реализации сценария записи и отмены записи к врачу выбранной специальности по услуге «Запись на прием к врачу»

Предварительные условия:

- 1) Пользователь авторизован на ЕПГУ с подтвержденной УЗ ФЛ.
- 2) Пользователь имеет прикрепление в одном из пилотных регионов для которых открыта услуга «Записи на прием к врачу»
<https://www.gosuslugi.ru/10700/1/form>:
- 3) Пользователь перешел на страницу услуги «записи на прием к врачу» по адресу <https://www.gosuslugi.ru/10700/1/form>

№	Шаг	Ожидаемый результат
	Нажать на кнопку «Начать»	Открылся экран «Кого записать на приём?» в форме квиза с вариантами для выбора: Себя Ребенка Другого человека
1	Выбрать пункт «Себя»	Открылся экран «Выберите способ записи на приём» в форме квиза со списком возможных способов записи: По специальности врача По направлению По медорганизации
2	Выбрать пункт «По специальности врача»	Открылся экран «Выберите специальность врача» (если в методе Get_Specialist пришел список доступных специальностей, если пустой ответ – экран с ошибкой, если код ответа 3 – экран с ошибкой)
3	Выбрать специальность	Открылся экран «Выберите дату и врача» (если в методе GetCountFreeSlot пришли доступные слоты для записи с учетом пола и возраста)
4	Выбрать врача и слот для записи, осуществить запись (если в методе Get_Free_Slot пришли доступные для записи слоты по конкретному врачу)	Открылся экран «Вы записаны на прием» (если в метод book пришел успешный ответ с кодом 0, если пришел код 101,11-18 – экран с ошибкой)
5	Открыть каталог и выбрать услугу Запись на прием к врачу, дойти до шага «Выберите специальность врача». Проверить, что у специальности, по которой была	Информация о записи к специалисту отображается в специальности

№	Шаг	Ожидаемый результат
	осуществлена ранее запись показывается информация о предстоящей записи (метод Get_Books должен передать сведения о записи, если пустой ответ с кодом 7 – ничего не отобразится))	
6	Нажать на специальность, у которой отображается информация о предстоящей записи	Откроется экран
7	Отменить запись используя ссылку «отменить запись» (в методе Cancel пришел успешный ответ 0)	Запись успешно отменена

В.6.5 Проверка реализации сценария отображения информации по временно недоступным расписаниям врачей (отпуск, больничный, командировка, прочее)

Предварительные условия:

- 1) Пользователь авторизован на ЕПГУ с подтвержденной УЗ ФЛ.
- 2) Пользователь имеет прикрепление в одном из пилотных регионов для которых открыта услуга «Записи на прием к врачу»
<https://www.gosuslugi.ru/10700/1/form>:
- 3) Пользователь перешел на страницу услуги «записи на прием к врачу» по адресу <https://www.gosuslugi.ru/10700/1/form>

№	Шаг	Ожидаемый результат
	Нажать на кнопку «Начать»	Открылся экран «Кого записать на приём?» в форме квиза с вариантами для выбора: Себя Ребенка Другого человека
1	Выбрать пункт «Себя»	Открылся экран «Выберите способ записи на приём» в форме квиза со списком возможных способов записи: По специальности врача По направлению По медорганизации
2	Выбрать пункт «По специальности врача»	Открылся экран «Выберите специальность врача»

№	Шаг	Ожидаемый результат
3	Выбрать специальность	Открылся экран «Выберите дату и врача»
4	Просмотреть перечень врачей, у которых нет слотов для записи в связи с периодом недоступности (в методе GetUnaccessibleResources придут записи с периодом недоступности)	Перечень врачей и причина отображается

В.6.6 Проверка реализации проектирования сценария записи и отмены записи по записи по диспансерному наблюдению

Предварительные условия:

- 1) Пользователь авторизован на ЕПГУ с подтвержденной УЗ ФЛ.
- 2) Пользователь имеет прикрепление в одном из пилотных регионов для которых открыта услуга «Записи на прием к врачу»
<https://www.gosuslugi.ru/10700/1/form>:
- 3) Пользователь перешел на страницу услуги «записи на прием к врачу» по адресу
<https://www.gosuslugi.ru/10700/1/form>

№	Шаг	Ожидаемый результат
	Нажать на кнопку «Начать»	Открылся экран «Кого записать на приём?» в форме квиза с вариантами для выбора: Себя Ребенка Другого человека
1	Выбрать пункт «Себя»	Открылся экран «Выберите способ записи на приём» в форме квиза со списком возможных способов записи: По специальности врача По направлению По медорганизации
2	Выбрать пункт «По специальности врача»	Открылся экран «Выберите специальность врача»
3	Выбрать специальность с пометкой «Состоите на диспансерном учете у врача: <i>ФИО</i> » (пометка отобразится, если в методе Get_Observation были переданы сведения о диспансерном прикреплении, в случае если был передан код 3- 7 – ничего не	Открылся экран «Выберите дату и врача»

№	Шаг	Ожидаемый результат
	отобразится))	
4	Выбрать врача и слот для записи, осуществить запись (пользователю дополнительно доступны слоты для записи по диспансерному наблюдению с типом D_OBSERVATION, проверяется в методах Get_Free_Slot)	Открылся экран «Вы записаны на прием»
5	Отменить запись используя ссылку «отменить запись» (в методе Cancel пришел успешный ответ 0)	Запись успешно отменена

В.7 Проверки и фильтры на стороне ЕПГУ

В.7.1 Проверка доступности витрины и редирект на старую форму записи

Реализована бесшовная маршрутизация между услугами Запись на прием к врачу: на витринах данных (10700) и КУ ФЭР (600204). Редирект происходит в случаях:

- Если деактивирован регион, использующий витрину для услуги 10700;
- Если у региона еще не развернута и не функционирует витрина данных;
- Если у региона развернута и функционирует витрина данных, но не поддерживается сценарий записи себя по направлению

Основные положения:

1. Всегда открывается стартовый экран 10700.
2. Возможен обратный переход из 600204 в 10700 по кнопке назад, первой после редиректа, или при изменении региона.
3. Редирект с 10700 осуществляется на конкретный экран 600204, а не на стартовый экран 600204.

В.7.2 Определение региона пользователя

Регион пользователя определяется по его персональным данным из профиля одним из способов:

1. по верифицированному полису (ЕНП)

2. по адресу фактического проживания
3. по адресу прописки
4. по адресу временной прописки

Если ни один из указанных данных в профиле не заполнен – отображаем экран для выбора региона из списка.

В случае если не смогли провалидировать пользователя по его персональным данным в определенной витрине – отображается экран с кнопкой «Изменить регион».

В.7.3 Фильтр списка доступных специальностей

Полученный список специалистов (рег.запросы GetSpecialistsByPatientId и getResourcesByMOid) дополнительно фильтруется по возрасту, полу и сортируется по приоритету согласно справочнику ЕНСИ <https://esnsi.gosuslugi.ru/classifiers/8163/data?pg=1&q=medical&p=1>

В.7.4 Фильтр списка доступных слотов

Полученный список слотов дополнительно фильтруется:

- по возрасту – параметр tag_age:
 - ADULT - взрослый
 - CHILD - детский
 - BABY - младенец
 - ALL - все возрастные группы
- по типу - tag_type. Для записи через ЕПГУ отображаются слоты с типом ORDINARY, EPGU, D_OBSERVATION (если у этого врача на наблюдении), REFERRAL (если сценарий записи по направлению). Дополнительно для информации отображаем слоты с типом REGISTRY, но для записи они не активны.

В.8 Дополнительные настройки

МИС может обновлять статус слота в таблице \${dbName}.slot_hidden Витрины РМИС ETL-процессами (актуальное значение определяем по времени обновления слота - \${dbName}.slot_hidden.update_ts).

Запись в Витрину-кэш из ЕПГУ производится с буферизацией через Адаптер

бронирования (обновления складываются в буфер адаптера, в последующем данные обновляются как 1 дельта (выгружаются в таблицу Витрины-кэш ($\{\text{dbName}\}_{\text{cache.slot}}$) с периодичностью 1 раз в N минут.

В.8.1 Рекомендации по быстродействию

Чтобы вычисление view $\{\text{dbName}\}_{\text{slot}}$ в Витрине РМИС производилось быстрее, таблица $\{\text{dbName}\}_{\text{cache.slot}}$ в Витрине-кэш должна содержать небольшое количество записей. Рекомендуем регулярно удалять из нее устаревшие записи, которые содержат неактуальные данные (информация, которая не является более новой, чем в таблице $\{\text{dbName}\}_{\text{slot_hidden}}$ Витрины РМИС).

Для достижения максимальной производительности Витрины-кэш обязательно требуется создать, на уровне СУБД, индексы в таблице $\{\text{dbName}\}_{\text{slot_hidden}}$, по полям, участвующим в запросах получения слотов.

Например, составной индекс по полям `resource_id` и `visitTime`.

В.8.2 Пример организации Витрины РМИС для выполнения сценария «Запись на прием к врачу»

В процессе работы Витрины РМИС (при выполнении сценария «Запись на прием к врачу») нагрузка на Витрину может возрастать в результате того, что увеличивается число запросов на запись, например, в связи неблагоприятной санитарно-эпидемиологической обстановкой. Администратору Витрины РМИС следует организовать обмен данными так, чтобы максимально сократить время ожидания пользователя.

Обновление данных в Витрине РМИС (из-за механизма дельт) занимает продолжительное время, которое может увеличиваться за счет выполнения параллельных ETL-процессов самой МИС, блокирующих работу Витрины.

Для оптимизации обработки запросов, увеличения производительности Витрины РМИС и сокращения времени ожидания пользователей, рекомендуется добавить дополнительную Витрину-кэш, в которую будет осуществляться запись на прием, если основная Витрина РМИС будет занята внутренними ETL-процессами. В дальнейшем, после завершения внутренних ETL-процессов МИС, данные из Витрины-кэш будут выгружены в основную Витрину РМИС.

В.8.3 Добавление Витрины-кэш

Добавление новой Витрины-кэш и ее настройки выполняются на уровне ProStore (для их выполнения предоставляется java-утилита, которую можно скачать по ссылке <https://storage.yandexcloud.net/dtm-artifacts/jars/DbPatch-fat-1.0.0-SNAPSHOT.jar>). Процесс установки утилиты описан в разделе «**Ошибка! Источник ссылки не найден.**».

Чтобы добавить и настроить Витрину-кэш для работы с Витриной РМИС, выполните следующие действия:

- 1) Создайте Витрину-кэш, например, `${dbName}_cache`.
- 2) В Витрине-кэш создайте таблицу `${dbName}_cache.slot` для хранения части данных (статуса) об изменяемых слотах (все данные хранить не требуется). Пример приведен в таблице .

Таблица 2 – Образец создаваемой таблицы

Название поля	Тип данных	Примечание
id	varchar	ПК
status	varchar	
update_ts	timestamp	

- 3) При обновлении статуса слота, в ответ на бронирование из ЕПГУ, обновляем статус в таблице кэш-витрины `${dbName}_cache.slot`.
- 4) В Витрине РМИС `${dbName}` выполните следующее:
 - Переименуйте таблицу `${dbName}.slot` в `${dbName}.slot_hidden`.
 - Объедините данные слота из основной Витрины РМИС с обновленным статусом из таблицы Витрины-кэш, для этого создайте нематериализованное представление `view` с именем `${dbName}.slot` из которой ЕПГУ будет брать данные и актуальный статус слота.

Пример:

```
create view ${dbName}.slot as
SELECT
  m.resource_id,
  m.id,
  m.tag_type,
  m.tag_age,
  m.visitTime,
  m.duration,
  m.create_ts,
```

```

CASE
  WHEN (c.id is NULL or c.update_ts < m.update_ts)
    THEN m.update_ts
    ELSE c.update_ts END
  AS update_ts,
CASE
  WHEN (c.id is NULL or c.update_ts < m.update_ts)
    THEN m.status
    ELSE c.status
  END AS status
FROM ${dbName}.slot_hidden m
LEFT JOIN ${dbName}_cache.slot c ON m.id = c.id

```

В.8.4 Пример исходного кода для организации Витрины-кэш

В.8.4.1 Пример кода запроса на запись к врачу

```

public ResponseEntity<Response> bookingSlot(String slotId, ru.itone.mis.model.Slot slot) {
    // Запись в буфер
    bufferRepository.offer(
        Slot.builder()
            .slotId(slotId)
            .resourceId(slot.getResourceId())
            .availableDate(LocalDate.parse(slot.getVisitTime()))
            .slotUpdateTs(updatedAt)
            .build()
    );
    //Формирование ответа
    Response response = new Response();
    response.setIsSuccess(true);
    return ResponseEntity.ok(response);
}

```

В.8.4.2 Реализация буфера

```

package ru.itone.mis.repository;

```

```

import 192ombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Component;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Iterator;
import java.util.concurrent.LinkedBlockingQueue;

@Slf4j
@Component
public class LBQBufferRepository<T> implements BufferRepository<T> {

    private final LinkedBlockingQueue<T> repository;

    public LBQBufferRepository() {
        this.repository = new LinkedBlockingQueue<>();
    }

    @Override
    // Положить в буфер слот
    public void offer(T element) {
        log.debug("Offer element {}", element);
        repository.offer(element);
    }

    @Override
    // Получить $count элементов из буфера
    public Collection<T> poll(int count) {
        ArrayList<T> result = new ArrayList<>(count);
        int drained = repository.drainTo(result, count);
        log.debug("Polled {} elements", drained);
        return result;
    }
}

```

```

@Override
// Получить один элемент из буфера
public T poll() {
    T element = repository.poll();
    log.debug("Poll element {}", element);
    return element;
}

```

```

@Override
public long size() {
    return repository.size();
}

```

```

@Override
public Iterator<T> iterator() {
    return repository.iterator();
}
}

```

В.8.4.3 Запись в Витрину-кэш из буфера

```

package ru.itone.mis.uploader;

import 193ombok.RequiredArgsConstructor;
import 193ombok.extern.slf4j.Slf4j;
import org.springframework.context.annotation.Profile;
import org.springframework.scheduling.annotation.Scheduled;
import org.springframework.stereotype.Component;
import org.springframework.util.CollectionUtils;
import ru.itone.mis.config.UploaderProperties;
import ru.itone.mis.dao.ProstoreDao;
import ru.itone.mis.dao.SlotDao;
import ru.itone.mis.domain.Slot;
import ru.itone.mis.repository.BufferRepository;

```

```

import java.util.Collection;
import java.util.stream.Collectors;

@Slf4j
@Component
@RequiredArgsConstructor
@Profile("!local")
public class ProstoreBufferUploader {

    private final BufferRepository<Slot> bufferRepository;
    private final SlotDao slotDao;
    private final ProstoreDao prostoreDao;
    private final UploaderProperties uploaderProperties;

    // Запуск метода с интервалом в n миллисекунд
    @Scheduled(fixedDelayString = "${upload.interval}")
    public void start() {
        long start = System.currentTimeMillis();
        //Получить из буфера слоты для бронирования
        Collection<Slot> values = bufferRepository.poll(uploaderProperties.getBatchSize());
        if (CollectionUtils.isEmpty(values)) {
            log.trace("NO DATA");
            return;
        }
        try {
            log.debug("Start uploading....");
            // открыть дельту
            prostoreDao.beginDelta();
            // записать в дельту данные
            slotDao.upsert(values
                .stream()
                .peek(slot -> slot.setSlotStatus("BUSY"))
                .collect(Collectors.toList()));
            log.debug("End upserting....");
        }
    }
}

```

```

        log.debug("Total time {} ms", System.currentTimeMillis() – start);
        // закрыть дельту
        prostoreDao.commitDelta();
    } catch (Exception e) {
        // откатить дельту
        prostoreDao.rollbackDelta();
    }
    log.debug("End uploading...");
    log.debug("Total time {} ms", System.currentTimeMillis() – start);
    log.info("Data upload successfully, size={}", values.size());
}
}

```

В.8.4.4 Очистка таблицы в Витрине-кэш

```

package ru.itone.mis.dao;

import 195ombok.AllArgsConstructor;
import 195ombok.extern.slf4j.Slf4j;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Repository;
import ru.itone.mis.config.TemporaryTableProperties;

import static java.lang.String.format;

@Repository
@Slf4j
@AllArgsConstructor
public class TemporaryTableDao {

    public static final String TRUNCATE_TEMPLATE = "TRUNCATE HISTORY %s%s.slot
FOR SYSTEM_TIME AS OF 'infinite'\n" +

```

```

        "WHERE update_ts < '%s'";

    public static final String MAX_UPDATE_TS_TEMPLATE = "SELECT max(m.update_ts)
FROM %s.slot_hidden m";

    private final JdbcTemplate jdbcTemplate;
    private final TemporaryTableProperties properties;

    private String maxUpdateTs() {
        String sql = format(MAX_UPDATE_TS_TEMPLATE, properties.getSchema());
        log.debug("Query {}", sql);
        return jdbcTemplate.query(
            sql,
            (rs, rowNum) -> rs.getTimestamp(1)
        ).get(0).toString();
    }

    public void truncate() {
        String sql = format(TRUNCATE_TEMPLATE, properties.getSchema(),
            properties.getCacheSuffix(),
            maxUpdateTs());
        log.debug("Executing {}", sql);
        jdbcTemplate.execute(
            sql
        );
    }
}

```

B.8.5 Оптимизация настроек Postgres

Важно! Все команды в Dbeaver выполняются в postgres, НЕ в Prostore! Для проверки настроек Postgres, выполните в Dbeaver команды:

```

SHOW effective_cache_size;
SHOW checkpoint_completion_target;
SHOW wal_buffers = '16MB';
SHOW default_statistics_target;
SHOW random_page_cost;
SHOW effective_io_concurrency;
SHOW min_wal_size;
SHOW max_wal_size;

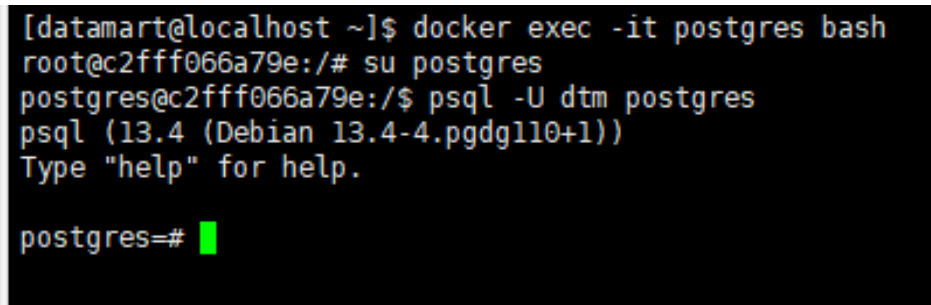
```



```
SHOW max_worker_processes;
SHOW max_parallel_workers_per_gather;
SHOW max_parallel_workers;
SHOW max_parallel_maintenance_workers;
SHOW max_connections;
SHOW shared_buffers;
SHOW temp_buffers;
SHOW work_mem;
SHOW maintenance_work_mem;
SHOW enable_seqscan;
```

Каждая из команд вернет значение настройки. Например, команда **SHOW shared_buffers;** по умолчанию вернет значение «128Mb». Подробнее о [Константах стоимости для планировщика](#). Для изменения настроек, необходимо подключиться к Postgres из консоли. Для этого выполните:

```
docker exec -it postgres bash
su postgres
psql -U dtm postgres
```



```
[datamart@localhost ~]$ docker exec -it postgres bash
root@c2fff066a79e:/# su postgres
postgres@c2fff066a79e:/ $ psql -U dtm postgres
psql (13.4 (Debian 13.4-4.pgdg110+1))
Type "help" for help.

postgres=#
```

Рисунок 4 – Подключение к Postgres

Далее, скорректируйте настройки, выполнив запросы:

```
ALTER SYSTEM SET effective_cache_size = '12GB';
ALTER SYSTEM SET checkpoint_completion_target = '0.9';
ALTER SYSTEM SET wal_buffers = '16MB';
ALTER SYSTEM SET default_statistics_target = '100';
ALTER SYSTEM SET random_page_cost = '4';
ALTER SYSTEM SET effective_io_concurrency = '2';
ALTER SYSTEM SET min_wal_size = '1GB';
ALTER SYSTEM SET max_wal_size = '4GB';
ALTER SYSTEM SET max_worker_processes = '10';
ALTER SYSTEM SET max_parallel_workers_per_gather = '4';
ALTER SYSTEM SET max_parallel_workers = '10';
ALTER SYSTEM SET max_parallel_maintenance_workers = '4';
ALTER SYSTEM SET max_connections = 40;
ALTER SYSTEM SET shared_buffers = '4GB';
ALTER SYSTEM SET temp_buffers = '128MB';
ALTER SYSTEM SET work_mem = '26214kB';
ALTER SYSTEM SET maintenance_work_mem = '1GB';
ALTER SYSTEM SET enable_seqscan = 'OFF';
```

Затем отключаемся и перезапускаем Postgres:

```
\q
exit
exit
docker restart postgres
```

```

postgres=# \q
postgres@c2fff066a79e:/$ exit
exit
root@c2fff066a79e:/# exit
exit
[datamart@localhost ~]$ docker restart postgres

```

Рисунок 5 – Выход и рестарт контейнера Postgres

После рестарта контейнера, можно проверить применение настроек. Для этого выполните команды в Dbeaver. Например, команда **SHOW shared_buffers;** вернет значение «4GB».

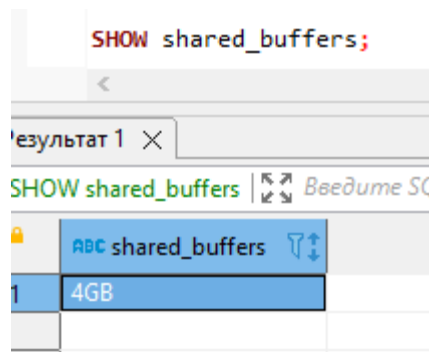


Рисунок 6 - SHOW shared_buffers

Для информации: команда проверки версии Postgres в Dbeaver: `select version()`

ПРИЛОЖЕНИЕ Г

АНКЕТА «СВЕДЕНИЯ ОБ ИТ-ЛАНДШАФТЕ ВЕДОМСТВА»

ЧАСТЬ 1

**характеристик информационного ресурса ведомства,
данные которого размещаются на витрине НСУД**

(курсивом приводятся примеры для гипотетического информационного ресурса (ИР), содержащего сведения о лицензиях на право осуществления определенной деятельности)

Название информационного ресурса ведомства:

1. **Общий объем данных в информационном ресурсе для размещения на витрине** – логический (не физический объем хранилища, который существенно зависит от способа хранения), здесь и ниже учитываются только данные в ИР, предназначенные для размещения на витрине, в т.ч. для межведомственного обмена.

Ответ:

1.1. из него – **объем бинарных данных** (изображения, документы – любые данные, не являющиеся текстовыми либо численными).

Ответ:

2. **Количество записей (элементов данных) информационного ресурса для размещения на витрине с разбивкой по сервисам.**

Ответ:

3. **Частота изменения записей информационного ресурса, размещаемых на витрине с разбивкой по сервисам.**

Ответ:

4. **Прогноз изменения за год объема данных, размещаемых на витрине**

Ответ:

5. **Перечень межведомственных получателей данных ИР, размещаемого на витрине** (по всем каналам, не только по СМЭВ, учитываются только получатели, являющиеся государственными органами, госкорпорациями, включая направляющих свои запросы через ГИС и ЕПГУ/ЕСИА), если таких получателей значительное количество – допускается указать только основных получателей и привести оценку общего количества получателей без их перечисления.

Ответ:

5.1. из них – кто обращается по СМЭВ2, кто – по СМЭВ3, кто – иными способами (указать).

6. **Существующие сервисы СМЭВ, предоставляющие сведения из информационного ресурса (ссылки на портал СМЭВ).**

Ответ:

7. **Средний размер набора данных, соответствующему одному объекту ИР** (*передаваемой межведомственно в ответ на запрос одной лицензии*).

Ответ:

8. Частота межведомственных запросов сведений из ИР.

Ответ:

9. Планируются ли в ближайшее время изменения структуры хранимых/предоставляемых данных в ИС ведомства?

Ответ:

ЧАСТЬ 2

Сведения о программно-техническом комплексе ведомства

1. Масштаб информационной ГИСЗ субъекта РФ: указывается один из вариантов - Федеральный, Региональный, Объектовый.

Ответ:

2. Наличие свободных мощностей, которые могут быть выделены для функционирования витрины данных.

Ответ:

1) CPU в целом:

2) ОЗУ в целом:

3) Дисковое пространство в целом:

ПРИЛОЖЕНИЕ Д

ФОРМА ЗАПРОСА ОБЪЕМА УСЛУГ ГОСОБЛАКА (ОБЩЕСТВЕННОЕ ОБЛАКО)

1. Информация о Пользователе и информационной системе

Наименование ФОИВ|РОИВ:

Наименование информационной ГИСЗ субъекта РФ:

Наименование ресурса для размещения в ГосОблаке:

2. Требования к предоставлению вычислительных ресурсов и ресурсов хранения данных.

Общие количественные требования к вычислительным ресурсам и ресурсам хранения данных для всех информационных систем

Потребителя, предоставляемые в рамках услуги, представлены в [Таблице 1](#).

Таблица 1 – Вычислительные ресурсы и ресурсы хранения данных

Наименование ресурса	Тип платформы виртуализации ⁴⁾	Кол-во
Виртуальный процессор vCPU (переподписка 1), шт.	Тип 2	
Виртуальный процессор vCPU	Тип 2	

4 Платформа виртуализации Тип №1 - VMWare или Тип №2 - Digital Energy

Наименование ресурса	Тип платформы виртуализации ⁴⁾	Кол-во
(переподписка 3), шт.		
Виртуальный процессор vCPU (переподписка 5), шт.	Тип 2	
Виртуальная оперативная память vRAM (1 ГБ), шт.	Тип 2	
Виртуальное дисковое пространство типа «vSSD» (1 ГБ), шт.	Тип 2	
Виртуальное дисковое пространство типа «vSAS» (1 ГБ), шт.	Тип 2	
Виртуальное дисковое пространство типа «vSATA» (1 ГБ), шт.	Тип 2	

Максимальные требуемые параметры для одной ВМ, указаны в [Таблице 2](#).

Таблица 2 – Максимальные параметры виртуальной машины.

Наименование параметра виртуальной машины	Макс. кол-во	Ед. измерения
Виртуальный процессор vCPU		шт.
Виртуальная оперативная память		ГБ

3. Требования к предоставлению управляемого кластера контейнеров Kubernetes

Количественные требования к оказанию услуги по предоставлению управляемого кластера контейнеров Kubernetes представлены в [Таблице 3](#).

Таблица 3 – Управляемые кластеры контейнеров Kubernetes

Наименование ресурса	Кол-во
K8s: виртуальный процессор vCPU (переподписка 1), шт.	
K8s: виртуальная оперативная память vRAM (1 ГБ), шт.	
K8s: виртуальное дисковое пространство типа «vSSD» (1 ГБ), шт.	

4. Требования к базовому конфигурированию и сопровождению виртуальных машин

Количественные требования к услуге по базовому конфигурированию и сопровождению виртуальных машин Потребителя, размещаемых в информационно-технологической инфраструктуре Исполнителя, представлены в [Таблице 4](#).

Таблица 4 – Количество виртуальных машин

Наименование ресурса	Кол-во
Базовое конфигурирование и сопровождение виртуальных машин (1 VM), шт.	

5. Требования к предоставлению ГИСЗ субъекта РФ резервного копирования

Количественные требования к дисковому пространству для хранения резервных копий, объему хранимых данных информационных систем и количества объектов (VM) управления резервным копированием, представлены в [Таблице 5](#).

Таблица 5 – Дисковое пространство для хранения резервных копий

Наименование ресурса	Кол-во
Объем дискового пространства для хранения резервных копий (50 ТБ), шт.	
Объем данных объектов резервного копирования информационных систем Потребителя (1 ТБ), шт.	

Наименование ресурса	Кол-во
Управление резервным копированием (1 VM), шт.	

6. Требования к услуге по предоставлению балансировки нагрузки

Количество балансировщиков нагрузки, предоставляемых в рамках услуги, представлены в [Таблице 6](#).

Таблица 6 – Балансировщик нагрузки.

Наименование ресурса	Кол-во
Балансировщик нагрузки, шт.	0

7. Требования к предоставлению выделенного сервера для платформы создания и проверки электронной подписи

Количественные требования по предоставлению сервера для платформы создания и проверки электронной подписи, предоставляемого в рамках услуги, представлены в [Таблице 7](#).

Таблица 7 – Сервер для платформы создания и проверки электронной подписи

Наименование ресурса	Кол-во
Сервер для платформы создания и проверки электронной подписи, шт.	0

8. Требования к предоставлению системного ПО

Количественные требования к лицензиям программного обеспечения, предоставляемого в рамках услуги, представлены в [Таблице 8](#).

Таблица 8 – Программные ресурсы и количество лицензий

Наименование программного ресурса	Кол-во
Программный ресурс Тип 1 (1 VM), шт.	
Программный ресурс Тип 2 (1 VM), шт.	0

9. Требования к предоставлению каналов связи

Количественные требования к каналам связи, предоставляемых в рамках услуги, и требуемой пропускной способности, представлены в [Таблице 9](#).

Таблица 9 – Каналы связи и пропускная способность

№	Наименование ресурса	Кол-во
	Канал связи к сети Интернет 100 Мбит/с (основной), шт.	0
	Канал связи к сети Интернет 100 Мбит/с (резервный), шт.	0
	Канал связи к сети Интернет 1 Гбит/с (основной), шт.	0
	Канал связи к сети Интернет 1 Гбит/с (резервный), шт.	0
	Канал связи к сети Интернет 5 Гбит/с (основной), шт.	0
	Канал связи к сети Интернет 5 Гбит/с (резервный), шт.	0
	Канал связи к сети Интернет 10 Гбит/с (основной), шт.	0
	Канал связи к сети Интернет 10 Гбит/с (резервный), шт.	0
	Канал связи к сети Интернет с защитой от DDoS-атак 100 Мбит/с (основной), шт.	0

№	Наименование ресурса	Кол-во
0	Канал связи к сети Интернет с защитой от DDoS-атак 100 Мбит/с (резервный), шт.	0
1	Канал связи к сети Интернет с защитой от DDoS-атак 1 Гбит/с (основной), шт.	0
2	Канал связи к сети Интернет с защитой от DDoS-атак 1 Гбит/с (резервный), шт.	0
3	Канал связи к сети Интернет с защитой от DDoS-атак 5 Гбит/с (основной), шт.	0
4	Канал связи к сети Интернет с защитой от DDoS-атак 5 Гбит/с (резервный), шт.	0
5	Канал связи к сети Интернет с защитой от DDoS-атак 10 Гбит/с (основной), шт.	0
6	Канал связи к сети Интернет с защитой от DDoS-атак 10 Гбит/с (резервный), шт.	0
7	Обеспечение подключения к СМЭВ 100 Мбит/с (основной), шт.	0
8	Обеспечение подключения к СМЭВ 100 Мбит/с (резервный), шт.	0
9	Обеспечение подключения к СМЭВ 1 Гбит/с (основной), шт.	0
0	Обеспечение подключения к СМЭВ 1 Гбит/с (резервный), шт.	0
1	Канал связи L2VPN 100 Мбит/с (основной), шт.	0
2	Канал связи L2VPN 100 Мбит/с (резервный), шт.	0
3	Канал связи L2VPN 1 Гбит/с (основной), шт.	0
4	Канал связи L2VPN 1 Гбит/с (резервный), шт.	0
5	Канал связи L2VPN 5 Гбит/с (основной), шт.	0

№	Наименование ресурса	Кол-во
6	Канал связи L2VPN 5 Гбит/с (резервный), шт.	0
7	Канал связи L2VPN 10 Гбит/с (основной), шт.	0
8	Канал связи L2VPN 10 Гбит/с (резервный), шт.	0
9	Канал связи L3VPN 100 Мбит/с (основной), шт.	0
0	Канал связи L3VPN 100 Мбит/с (резервный), шт.	0
1	Канал связи L3VPN 1 Гбит/с (основной), шт.	2
2	Канал связи L3VPN 1 Гбит/с (резервный), шт.	2
3	Канал связи L3VPN 5 Гбит/с (основной), шт.	0
4	Канал связи L3VPN 5 Гбит/с (резервный), шт.	0
5	Канал связи L3VPN 10 Гбит/с (основной), шт.	0
6	Канал связи L3VPN 10 Гбит/с (резервный), шт.	0
7	Обеспечение подключения к ВЧС Потребителя 100 Мбит/с (основной), шт.	0
8	Обеспечение подключения к ВЧС Потребителя 100 Мбит/с (резервный), шт.	0
9	Обеспечение подключения к ВЧС Потребителя 1 Гбит/с (основной), шт.	0
0	Обеспечение подключения к ВЧС Потребителя 1 Гбит/с (резервный), шт.	0
1	Обеспечение подключения к ВЧС Потребителя 5 Гбит/с (основной), шт.	0

№	Наименование ресурса	Кол-во
2	Обеспечение подключения к ВЧС Потребителя 5 Гбит/с (резервный), шт.	0
3	Обеспечение подключения к ВЧС Потребителя 10 Гбит/с (основной), шт.	0
4	Обеспечение подключения к ВЧС Потребителя 10 Гбит/с (резервный), шт.	0

Адреса площадок Потребителя для подключения каналов связи L2VPN или L3VPN и тип интерфейса подключения оконечного оборудования в ЦОД Потребителя, указаны в [Таблице 10](#).

Таблица 10 – Адреса площадок Потребителя и типы интерфейса подключения

№	Тип канала связи	Объекты подключения услуги		Назначение (основной/ резервный)	Пропускная способность, Гбит/с
		Адрес площадки Потребителя	Тип интерфейса*		
1					
2					
(*) - Тип интерфейса при оказании услуги может быть: RJ45(8P8C), SFP, SFP+					

Требуемая информация при заказе услуги обеспечение подключения к ВЧС представлена в [Таблице 11](#) и [Таблице 12](#).

Таблица 11 – Обеспечение подключения к ВЧС

№	Версия ПО СКЗИ, включая версии ПО управления	Номер криптографической сети Потребителя	Предполагаемый тип взаимодействия	Туннелируемые IP-сети (узлы)

Таблица 12 – Обеспечение маршрутизации при подключении к ВЧС.

№	В чьей зоне ответственности находится ВЧС	Кем обслуживается IP адресация ВЧС	Маршрутизация (BGP, статика)	Кем выделяются IP-сети ВЧС
1				

10. Требования к предоставлению публичных IPv4-адресов

Количественные требования к публичным IPv4-адресам, предоставляемых в рамках услуги, представлены в [Таблице 13](#).

Таблица 13 – Публичные IPv4-адреса

№	Наименование ресурса	Кол-во
1	Публичные IPv4-адреса, шт.	0

11. Требования к обеспечению информационной безопасности

Количественные требования к оказанию услуг по обеспечению информационной безопасности представлены в [Таблице 14](#).

Таблица 14 – Защита информационно-технологической инфраструктуры Исполнителя

№	Наименование ресурса	Кол-во
1	Защита информационно-технологической инфраструктуры Исполнителя (1 ГБ), шт.	
2	Подключение привилегированного пользователя, шт.	
3	Предоставление межсетевого экрана уровня приложений, шт.	

Количественные требования к услуге по обеспечению подключения по протоколу TLS, представлены в [Таблице 15](#).

Таблица 15 – Обеспечение подключения по протоколу TLS

№	Наименование ресурса	Кол-во
1	Обеспечение подключения по протоколу TLS (до 150 пользователей), шт.	
2	Обеспечение подключения по протоколу TLS (до 700 пользователей), шт.	
3	Обеспечение подключения по протоколу TLS (до 2000 пользователей), шт.	

12. Требования к оказанию услуг по предоставлению ПАК

Количественные требования ПАК, предоставляемые в рамках услуги, представлены в Таблице 16.

Таблица 16 – Программно-аппаратные комплексы

Наименование услуги	Состав услуги	Кол-во
Предоставление ПАК машины баз данных	Базовый модуль (предоставление оборудования), шт.	
	Модуль баз данных тип 2 (предоставление оборудования), шт.	
	Модуль баз данных тип 2 (предоставление ПО для баз данных), шт.	
	Модуль расширения хранения (предоставление оборудования), шт.	
Предоставление ПАК машины аналитической обработки больших данных	Базовый модуль (предоставление оборудования), шт.	
	Базовый модуль управления и распределения тип 2 (предоставление оборудования), шт.	
	Базовый модуль управления и распределения тип 2 (предоставление	

Наименование услуги	Состав услуги	Кол-во
	ПО для баз данных), шт.	
	Модуль вычисления и хранения D тип 2 (предоставление оборудования), шт.	
	Модуль вычисления и хранения D тип 2 (предоставление ПО для баз данных), шт.	
	Модуль вычисления и хранения T тип 2 (предоставление оборудования), шт.	
	Модуль вычисления и хранения T тип 2 (предоставление ПО для баз данных), шт.	
Предоставление ПАК машины резидентной СУБД	Базовый модуль (предоставление оборудования), шт.	
	Модуль вычисления и хранения D тип 2 (предоставление оборудования), шт.	
	Модуль вычисления и хранения D тип 2 (предоставление ПО для баз данных), шт.	
	Модуль вычисления и хранения T тип 2 (предоставление оборудования), шт.	
	Модуль вычисления и хранения T тип 2 (предоставление ПО для баз данных), шт.	
Предоставление ПАК машины статистической обработки больших данных	Базовый модуль (предоставление оборудования), шт.	
	Базовый модуль управления и распределения тип 2 (предоставление оборудования), шт.	
	Базовый модуль управления и распределения тип 2 (предоставление ПО для баз данных), шт.	
	Модуль вычисления и хранения D тип 2 (предоставление	

Наименование услуги	Состав услуги	Кол-во
	оборудования), шт.	
	Модуль вычисления и хранения D тип 2 (предоставление ПО для баз данных), шт.	
Предоставление ПАК машины потоковой обработки больших данных	Базовый модуль (предоставление оборудования), шт.	
	Базовый модуль управления и распределения тип 2 (предоставление оборудования), шт.	
	Базовый модуль управления и распределения тип 2 (предоставление ПО для баз данных), шт.	
	Модуль вычисления и хранения тип 2 (предоставление оборудования), шт.	
	Модуль вычисления и хранения тип 2 (предоставление ПО для баз данных), шт.	
Предоставление ПАК машины обработки неструктурированных больших данных	Базовый модуль (предоставление оборудования), шт.	
	Базовый модуль управления и распределения тип 2 (предоставление оборудования), шт.	
	Базовый модуль управления и распределения тип 2 (предоставление ПО для баз данных), шт.	
	Модуль вычисления и хранения D тип 2 (предоставление оборудования), шт.	
	Модуль вычисления и хранения D тип 2 (предоставление ПО для баз данных), шт.	
	Модуль вычисления и хранения T тип 2 (предоставление оборудования), шт.	
	Модуль вычисления и хранения T тип 2 (предоставление ПО для	

Наименование услуги	Состав услуги	Кол-во
	баз данных), шт.	
Предоставление ПАК машины хранения данных	Базовый модуль (предоставление оборудования), шт.	
	Модуль хранения D тип 1 (предоставление оборудования), шт.	
	Модуль хранения T тип 1 (предоставление оборудования), шт.	
Предоставление ПАК БД	Базовый модуль ПАК СУБД №1 (предоставление оборудования и ПО), шт.	
	Базовый модуль ПАК СУБД №2 (предоставление оборудования и ПО), шт.	
	Базовый модуль ПАК прочих БД №1 (предоставление оборудования и ПО), шт.	
	Базовый модуль ПАК прочих БД №2 (предоставление оборудования и ПО), шт.	

13. Форма списка авторизованных лиц для формирования обращений (создание инцидентов, запросов на обслуживание, запросов на изменение) представлена в Таблице 17.

Таблица 17 – Список авторизованных лиц Потребителя

№	ФИО	Контактный телефон	Контактные адрес электронной почты

Таблица 18. Дополнительные ресурсы

№	Наименование ресурса	Кол-во
1		

ПРИЛОЖЕНИЕ Е

ТИПОВАЯ МЕТОДИКА ОБСЛУЖИВАНИЯ

ТИПОВОГО ПО «ВИТРИНА ДАННЫХ»

1035009567450.СМЭВ-ЕСНСИ-ЕИП НСУД-2023-1.М 01

ОП1

Назначение документа

Национальная система управления данными (далее – НСУД) представляет собой систему, состоящую из взаимосвязанных элементов информационно-технологического, организационного, методологического, кадрового и нормативно-правового характера и обеспечивающую достижение целей, и выполнение задач, обозначенных в Концепции Национальной системы управления данными, утвержденной распоряжением Правительства Российской Федерации от 3 июня 2019 года № 1189-р.

НСУД предназначена для управления информацией, содержащейся в информационных системах органов и организаций государственного сектора, а также в информационных ресурсах, созданных в целях реализации полномочий органов и организаций государственного сектора (далее – государственные данные) и для осуществления информационного обмена между Поставщиками и Получателями данных, присоединившимися к НСУД (далее – Участники НСУД).

Управление процессами информационного обмена между Участниками НСУД осуществляется средствами федеральной государственной информационной системы «Единая информационная платформа Национальной системы управления данными» (далее – ФГИС «ЕИП НСУД»).

Для передачи данных между Участниками НСУД используется среда взаимодействия НСУД, состоящая из Системы межведомственного электронного взаимодействия 3.0 (далее – СМЭВ) и (или) подсистемы обеспечения доступа к данным СМЭВ (далее – ПОДД СМЭВ) (СМЭВ 4.0), обеспечивающих транспорт и процессинг данных, а также агентов ПОДД СМЭВ, устанавливаемых на стороне Участников НСУД.

ВД является частью НСУД и предназначена для загрузки публикуемых данных в отдельную БД на стороне Поставщика данных. ВД представляет собой комплекс программных и технических средств, использующий типовое программное обеспечение, устанавливаемое на стороне поставщиков или потребителей данных.

Методика обслуживания типового ПО «Витрина данных» предназначена для специалистов поддерживающих и обеспечивающих работу ВД и направлена на ускорение процедур обслуживания и сокращение времени на типовые работы.

Документ покажет описание набора инструментов для работы с ВД НСУД, описание настройки и обслуживания PostgreSQL, настройки и обслуживания Kafka и Zookeeper, описание дополнительных операций, резервного копирования, логирования, очистки данных, описание мониторинга ВД.

Е.1 Описание набора инструментов для работы с витриной данных НСУД

Е.1.1 DBeaver. Описание инструмента и его функциональность

Е.1.1.1 Описание программы

DBeaver - это программное обеспечение, которое действует как универсальный инструмент базы данных. Предназначен для управления базами данных с помощью интерфейса JDBC.

Е.1.1.2 Установка и подключение

Установка и подключение к витрине данных описано в статье [Установка и настройка Dbeaver для работы с витриной данных Лайт](#)

Е.1.1.3 Работа в программе

Рассмотрим основные команды, которые чаще всего встречаются в пользовательских сценариях. База данных в ProStore - это и есть витрина с данными. Имя Базы данных является мнемоникой витрины из НСУД.

Е.1.1.4 Создание и удаление БД

База данных. Для создания базы данных (витрина/датамарт), выполните запрос:

```
CREATE DATABASE testDBname
```

где testDBname - имя базы данных.

Результат:

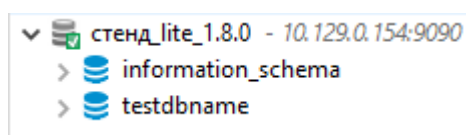


рис.1 Результат создания testDBname

Для удаления базы данных (витрина/датамарт), выполните запрос:

```
DROP DATABASE testDBname
```

где testDBname - имя базы данных.

Результат:

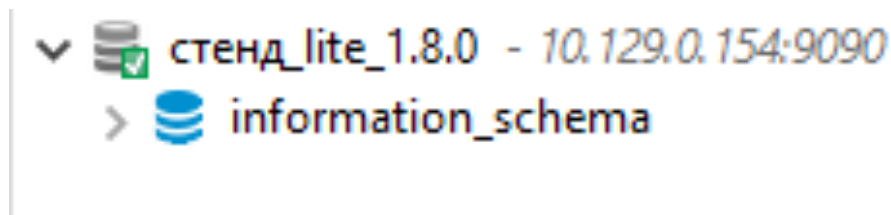


рис. 2 Результат удаления testDBname

Е.1.1.5 Создание и удаление таблиц

Для создания таблицы, выполните запрос:

```
CREATE TABLE testDBname.testTBLname (
  ID INT NOT NULL,
  NAME_REG VARCHAR NULL,
  CODE_MAN INT32 NULL,
  PRIMARY KEY (ID))
DISTRIBUTED BY (ID)
DATASOURCE_TYPE (ADP)
```

где testTBLname - имя создаваемой таблицы.

Результат:

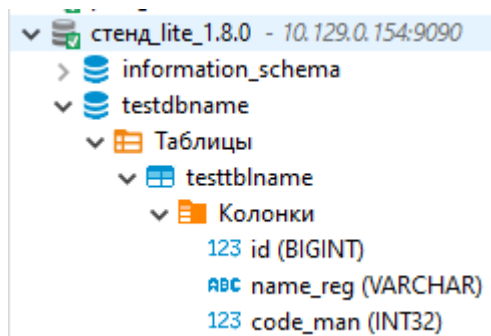


рис. 3 Результат создания testTBLname

Для удаления таблицы, выполните запрос:

```
DROP TABLE testDBname.testTBLname
```

где testTBLname - имя удаляемой таблицы.

Результат:

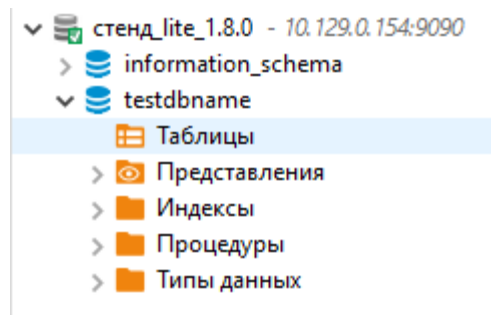


рис. 4 Результат удаления testTBLname

Е.1.1.6 Создание и удаление standalone-таблиц

Для создания standalone-таблиц, выполните запрос:

```
CREATE READABLE EXTERNAL TABLE testDBname.testTBLname_readable (
  ID INT NOT NULL,
  NAME_REG VARCHAR NULL,
  CODE_MAN INT32 NULL,
  PRIMARY KEY (ID))
DISTRIBUTED BY (ID)
LOCATION 'core:adp://testDBname.testTBLname_standalone'
OPTIONS ('auto.create.table.enable=true');

CREATE WRITABLE EXTERNAL TABLE testDBname.testTBLname_writable (
  ID INT NOT NULL,
  NAME_REG VARCHAR NULL,
  CODE_MAN INT32 NULL,
  PRIMARY KEY (ID))
DISTRIBUTED BY (ID)
LOCATION 'core:adp://testDBname.testTBLname_standalone'
```

По умолчанию система создает внешнюю таблицу в логической базе данных и не создает связанную с ней standalone-таблицу в датасорсе. Это может быть полезно, если standalone-таблица уже существует в датасорсе.

- READABLE-таблица служит для чтения записей
- WRITABLE-таблица служит для вставки, изменения и удаления записей

Результат:

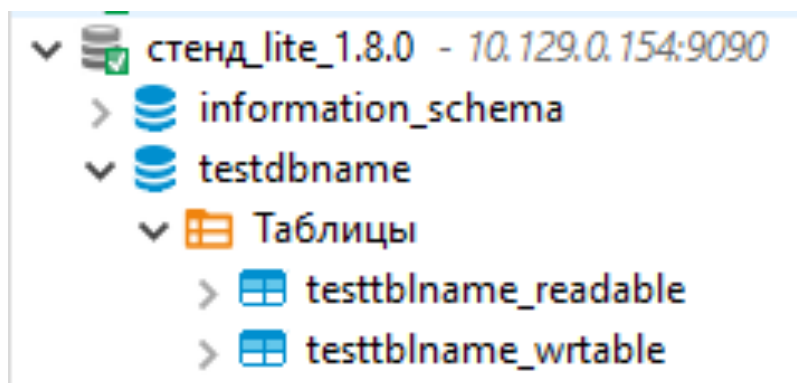


рис. 5 Результат создания внешних таблиц

Для удаления standalone-таблиц, выполните запрос:

```

DROP READABLE EXTERNAL TABLE testDBname.testTBLname_readable;

DROP WRITABLE EXTERNAL TABLE testDBname.testTBLname_writable

```

Результат:

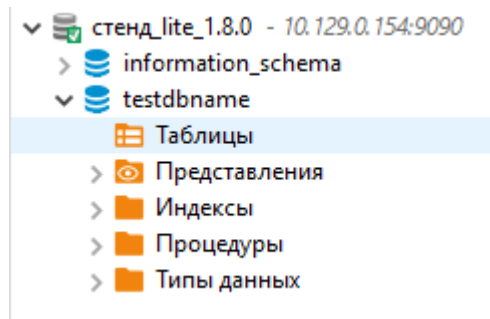


рис. 6 Результат удаления внешних таблиц

Е.1.1.7 Создание и удаление таблиц загрузок и выгрузок

Для создания таблиц загрузок и выгрузок, выполните запрос:

```

CREATE DOWNLOAD EXTERNAL TABLE testDBname.testTBLname_download (
  ID INT NOT NULL,
  NAME_REG VARCHAR NULL,
  CODE_MAN INT32 NULL,
  SYS_OP INT32)
LOCATION 'kafka://zookeeper:2181/test TOPIC name'
FORMAT 'avro'
CHUNK_SIZE 100;

CREATE UPLOAD EXTERNAL TABLE testDBname.testTBLname_upload (
  ID INT NOT NULL,
  NAME_REG VARCHAR NULL,
  CODE_MAN INT32 NULL)
LOCATION 'kafka://zookeeper:2181/test TOPIC name'
FORMAT 'avro'
MESSAGE_LIMIT 100

```

где test_TOPIC_name - имя топика, в который выгружаются данные при вставке данных в DOWNLOAD EXTERNAL таблицу.

UPLOAD EXTERNAL таблица служит для выгрузки данных из топика, например, в другую логическую таблицу.

Результат:

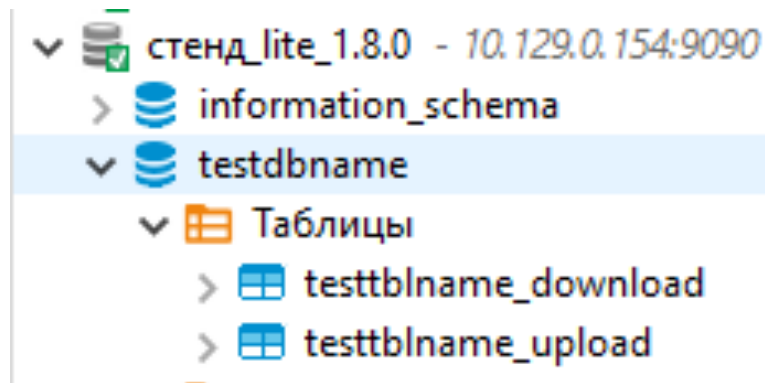


рис. 7 Результат создания таблиц Загрузки и Выгрузки

Для удаления таблиц загрузок и выгрузок, выполните запрос:

```
DROP DOWNLOAD EXTERNAL TABLE testDBname.testTBLname_download;

DROP UPLOAD EXTERNAL TABLE testDBname.testTBLname_upload
```

Результат:

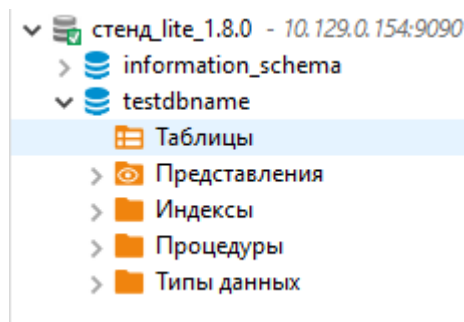


рис. 8 Результат удаления таблиц Загрузки и Выгрузки

Во время работы CSV-uploader создает UPLOAD EXTERNAL таблицу с постфиксом "_ext". После выполнения загрузки, модуль удаляет эти таблицы. Но иногда они могут не удалиться. В этом случае их можно удалить самостоятельно, выполнив запрос:

```
DROP UPLOAD EXTERNAL TABLE testDBname.testTBLname_ext
```

где testDBname - имя Базы данных,
testTBLname_ext - имя удаляемой таблицы с постфиксом "_ext"

Е.1.1.8 Добавление и переименование полей в таблицах

В данном примере рассмотрим изменение таблицы testTBLname.

Функция ALTER TABLE доступна в версии ProStore 6.0 и выше.

Добавим новое поле CODE_DESCR с типом данных VARCHAR. Для этого выполните запрос:

```
ALTER TABLE testDBname.testtblname
ADD COLUMN CODE_DESCR VARCHAR
```

А так же переименуем поле CODE_MAN в CODE_MANUAL Для этого выполните запрос:

```
ALTER TABLE testDBname.testtblname
RENAME COLUMN CODE_MAN TO CODE_MANUAL
```

Результат:

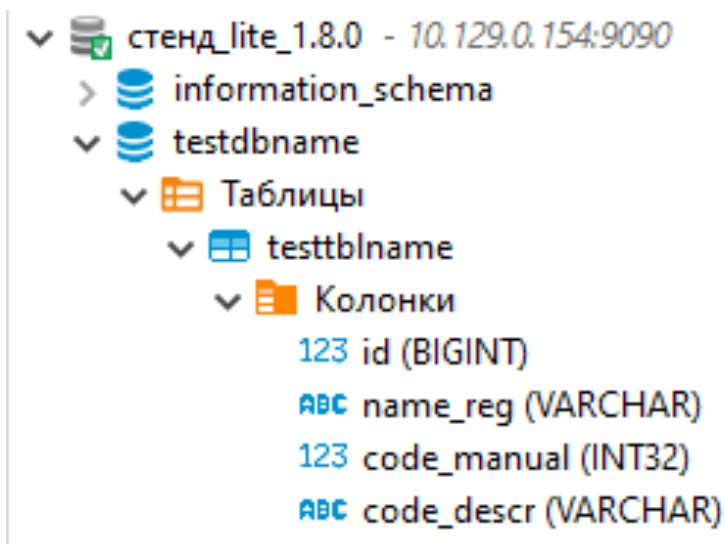


рис. 9 Результат добавления и переименования полей

Е.1.1.9 Получение информации о версиях компонентов

Для получения информации о версиях программных компонентов, с которыми работает система, выполните запрос:

```
CHECK_VERSIONS ()
```

Результат выполнения запроса на витрине данных Лайт версии 1.8.0:

CHECK_VERSIONS		Введите SQL выражение чтобы отфильтровать результаты
component_name	version	
query-execution-core	6.0.0	
ADP: adp instance	PostgreSQL 13.4 (Debian 13.4-4.pgdg110+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 10.2.1-6) 10.2.1 20210110, 64-bit	
ADP: kafka-postgres connector reader	0.5.0	
ADP: kafka-postgres connector writer	0.5.0	
status-monitor	6.0.0	

рис. 10 Результат выполнения команды CHECK_VERSIONS

Е.1.1.10 Получение DDL таблицы

Для получения DDL таблицы, т.е. код запроса для создания таблицы, выполните запрос:

```
GET_ENTITY_DDL(testDBname.testTBLname)
```

где testDBname - имя Базы данных,
testTBLname_ext - имя таблицы

Е.1.1.11 Получение информации о выполнении SELECT-запроса

Для получения информации о выполнении SELECT-запроса без его фактического выполнения в датасорсе, выполните запрос:

```
SELECT * FROM testDBname.testTBLname ESTIMATE_ONLY
```

где ESTIMATE_ONLY - ключевое слово

Результат:

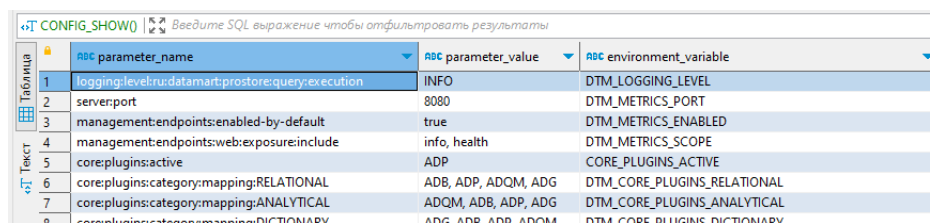
```
{
  "datasource": "ADP",
  "estimation": [
    {
      "Plan": {
        "Node": "Bitmap", "Type": "Bitmap", "Heap": "Scan",
        "Parallel": "Aware": false,
        "Relation": "Name": "testtblname_actual",
        "Alias": "testtblname_actual",
        "Startup": "Cost": 5.8,
        "Total": "Cost": 19.05,
        "Plan": "Rows": 72,
        "Plan": "Width": 76,
        "Recheck": "Cond": "(sys_from <= '-1'::integer)",
        "Filter": "(COALESCE(sys_to, '9223372036854775807'::bigint) >=
        '-1'::integer)",
        "Plans": [
          {
            "Node": "Bitmap", "Type": "Bitmap", "Index": "Scan",
            "Parent": "Relationship": "Outer",
            "Parallel": "Aware": false,
            "Index": "Name": "testtblname_actual_sys_from_idx",
            "Startup": "Cost": 0.0,
            "Total": "Cost": 5.78,
            "Plan": "Rows": 217,
            "Plan": "Width": 0,
            "Index": "Cond": "(sys_from <= '-1'::integer)"
          }
        ]
      }
    }
  ],
  "query": "SELECT id, name_reg, code_manual, code_descr FROM
testdbname.testtblname_actual WHERE sys_from <= -1 AND COALESCE(sys_to,
9223372036854775807) >= -1"
}
```

Е.1.1.12 Информация об одном или всех параметрах конфигурации системы

Для получения информации об одном или всех параметрах конфигурации системы, выполните запрос:

```
CONFIG_SHOW()
```

Результат:



parameter_name	parameter_value	environment_variable
logging:level:ru:datamart:prostore:query:execution	INFO	DTM_LOGGING_LEVEL
server:port	8080	DTM_METRICS_PORT
management:endpoints:enabled-by-default	true	DTM_METRICS_ENABLED
management:endpoints:web:exposure:include	info, health	DTM_METRICS_SCOPE
core:plugins:active	ADP	CORE_PLUGINS_ACTIVE
core:plugins:category:mapping:RELATIONAL	ADB, ADP, ADQM, ADG	DTM_CORE_PLUGINS_RELATIONAL
core:plugins:category:mapping:ANALYTICAL	ADQM, ADB, ADP, ADG	DTM_CORE_PLUGINS_ANALYTICAL
core:plugins:category:mapping:DICTIONARY	ADG, ADB, ADP, ADQM	DTM_CORE_PLUGINS_DICTIONARY

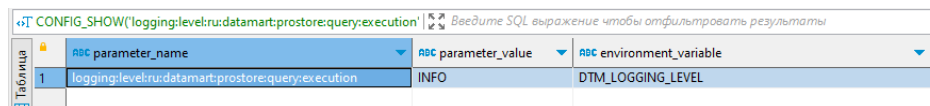
рис. 11 Результат выполнения команды CONFIG_SHOW

Для получения информации об одном параметре, выполните запрос:

```
CONFIG_SHOW('logging:level:ru:datamart:prostore:query:execution')
```

где 'logging:level:ru:datamart:prostore:query:execution' - имя параметра конфигурации системы

Результат:



parameter_name	parameter_value	environment_variable
logging:level:ru:datamart:prostore:query:execution	INFO	DTM_LOGGING_LEVEL

рис. 12 Результат выполнения команды CONFIG_SHOW с параметром

Е.1.2 Zoonavigator. Описание инструмента и его функциональность

Е.1.2.1 Описание программы

ZooNavigator — это веб-интерфейс и редактор для работы с ZooKeeper.

Это проект с открытым исходным кодом, доступным на GitHub.

Основная цель этого проекта — предоставить способ эффективного управления нодами ZooKeeper. Основные возможности программы:

Управление несколькими кластерами ZooKeeper одновременно;

Расширенные операции с нодами - перемещение, копирование и вставка, экспорт\импорт, массовое удаление;

Умный редактор данных в нодах - проверка данных, подсветка синтаксиса и автоматическое форматирование, редактирование данных;

Е.1.2.2 Установка и подключение

ZooNavigator поставляется в виде Docker-контейнера. Для установки, необходимо выполнить команду (Важно, чтобы машина находилась в той же сети, что Zookeeper):

```
docker run -d -p 9090:9090 -e HTTP_PORT=9090 --name zoonavigator --restart
unless-stopped elkozmon/zoonavigator:latest
```

Параметры запуска:

- `-d` Запустить контейнер в фоновом режиме и показать идентификатор контейнера
- `-p 9099:9000` Проброс порта с хоста 9099 на порт контейнера 9000
- `-e HTTP_PORT=9000` Запись переменной `HTTP_PORT` в параметры контейнера
- `--name zoonavigator` Имя контейнера
- `--restart unless-stopped` Политика перезапуска контейнера

После успешной установки, необходимо перейти по адресу `http://IP:Port` для доступа к веб-интерфейсу Приложения и в поле «Connection string» (рис. 13) ввести IP-адрес сервера с Zookeeper и нажать «Connect» для подключения

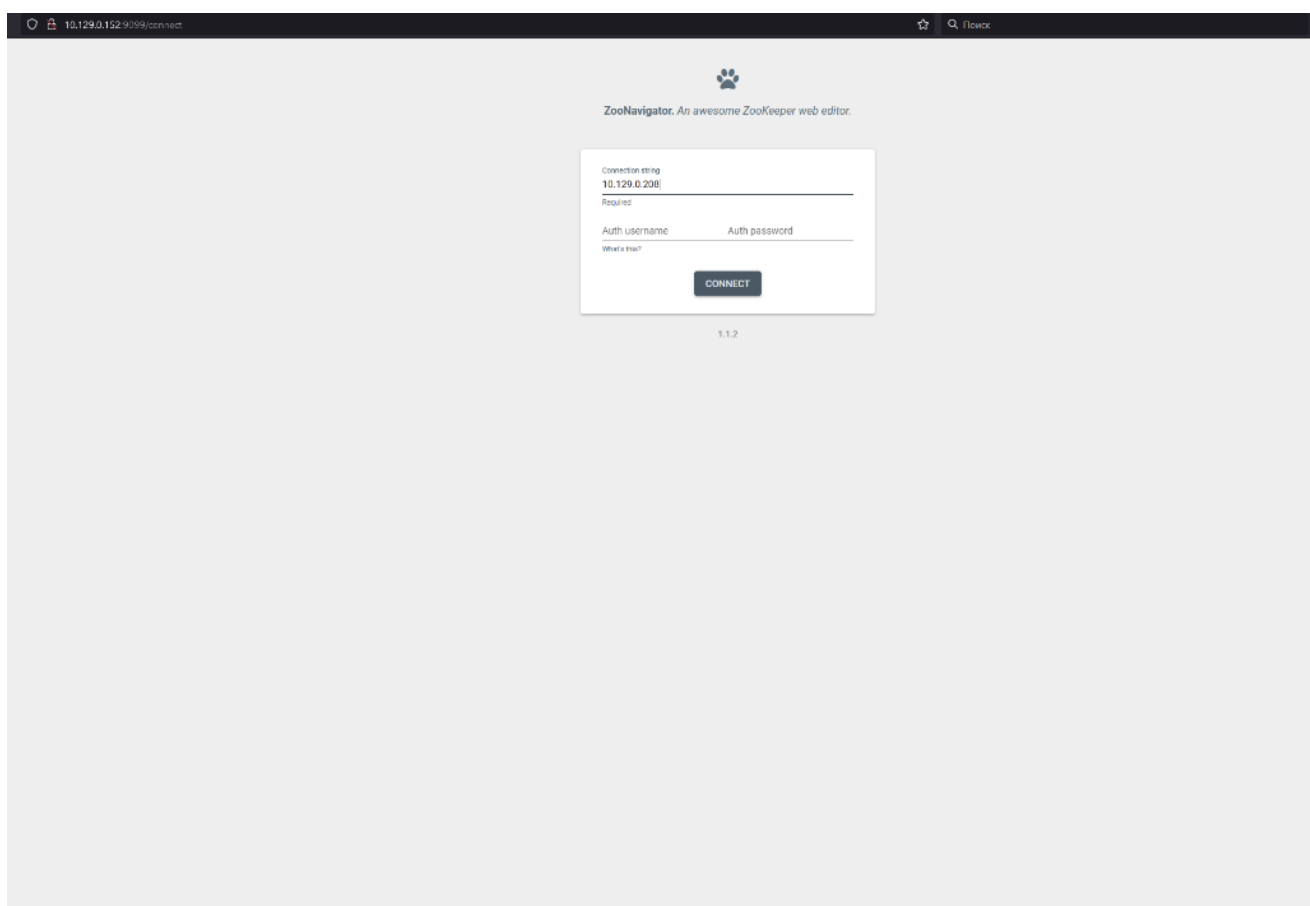


рис. 13 Окно подключения к Zoonavigator

В случае успешного подключения станет доступен интерфейс Zoonavigator (рис. 14)

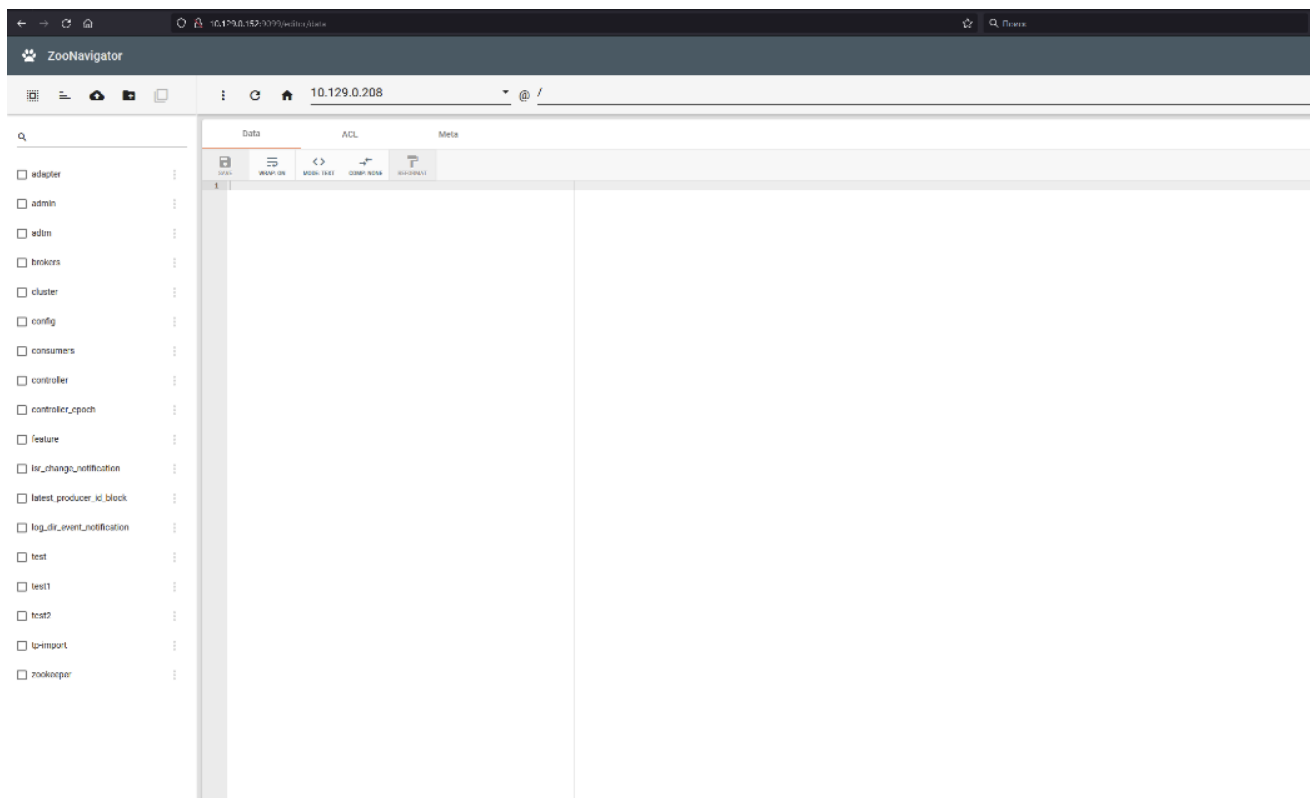


рис. 14 Стартовая страница Zoonavigator

Е.1.2.3 Работа в программе

В левой части интерфейса расположен список все нод в текущем каталоге (первоначально открывается корневой каталог /). Над списком располагается строка поиска. В центральной части интерфейса расположено окно работы с данными, записанными в ноду.

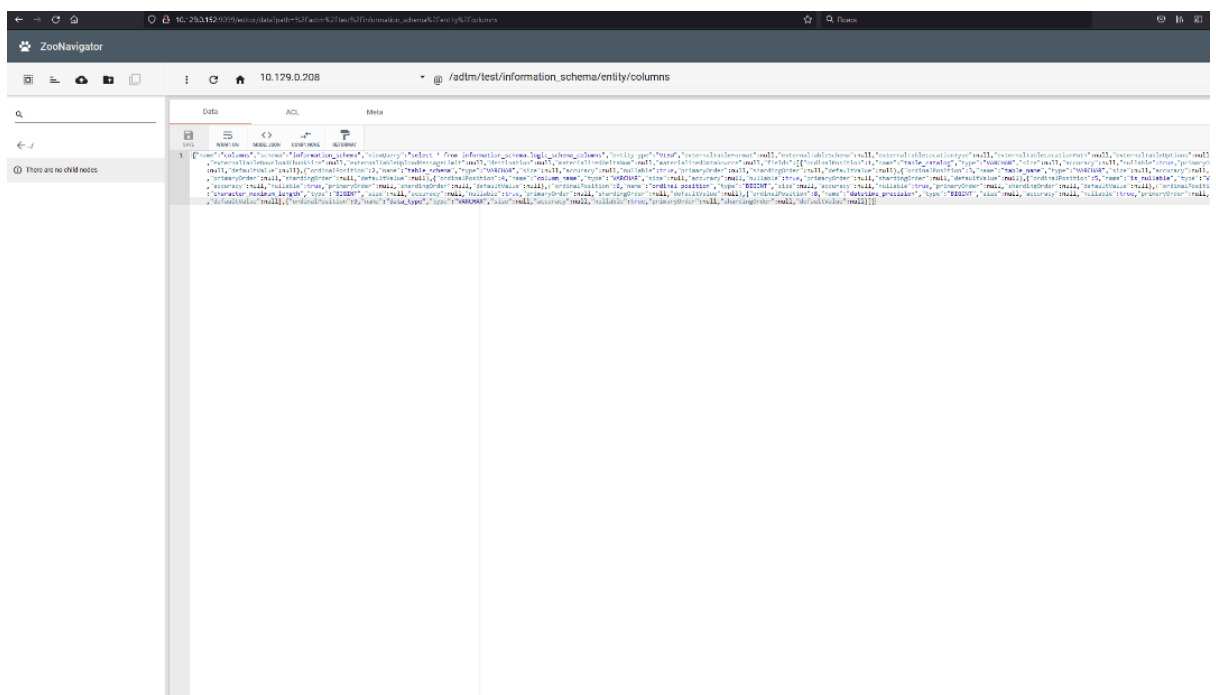


рис. 15 Данные для редактирования

Данные можно редактировать, изменять формат отображения, форматировать, сохранять с помощью кнопок, расположенных выше.

У каждой ноды есть область выделения (для совместных действий над множеством нод) и меню, для выполнения действий (символ 3 точки):

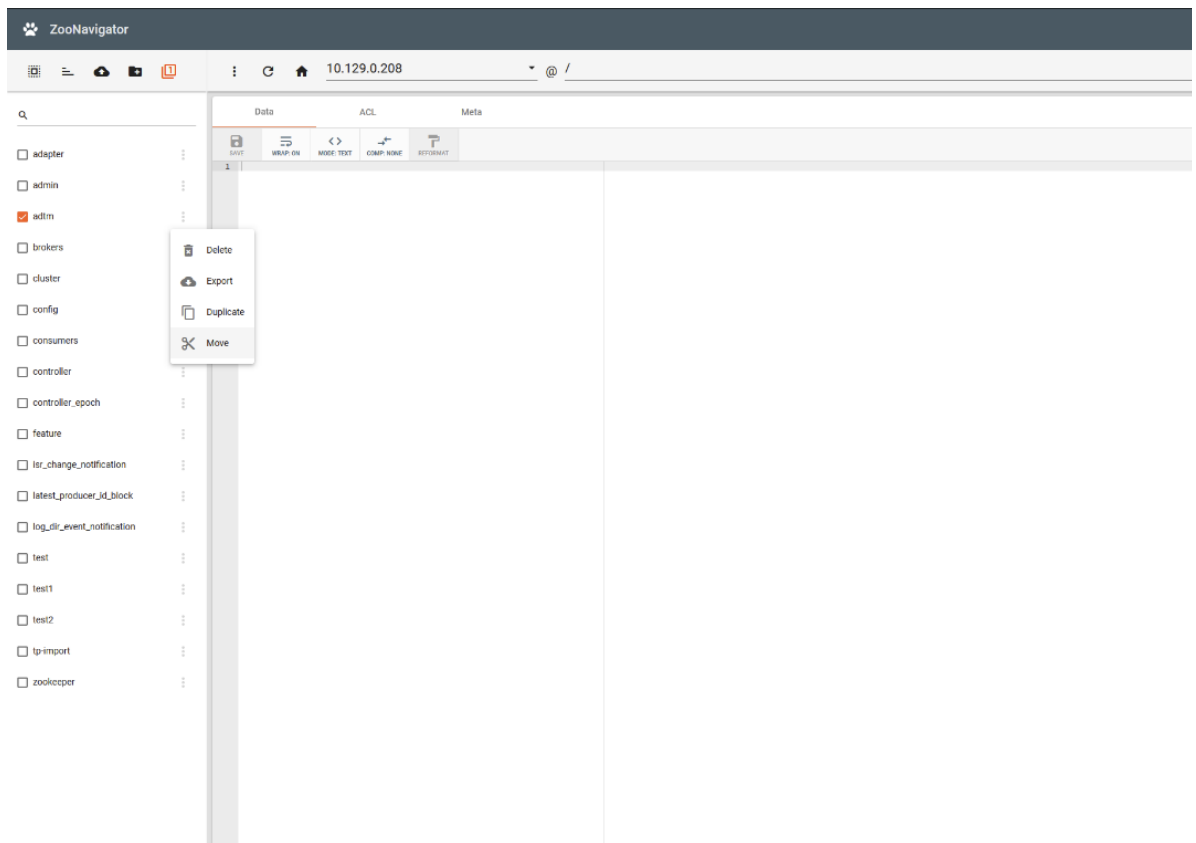


рис. 16 Нода adtm выделена, открыто меню действий

Е.1.2.4 Создание, удаление и перемещение нод

Для создания новой ноды, необходимо нажать кнопку «Create node» (рис. 17) в левом верхнем углу интерфейса, в открывшемся окне указать абсолютный путь и нажать кнопку «Create»:

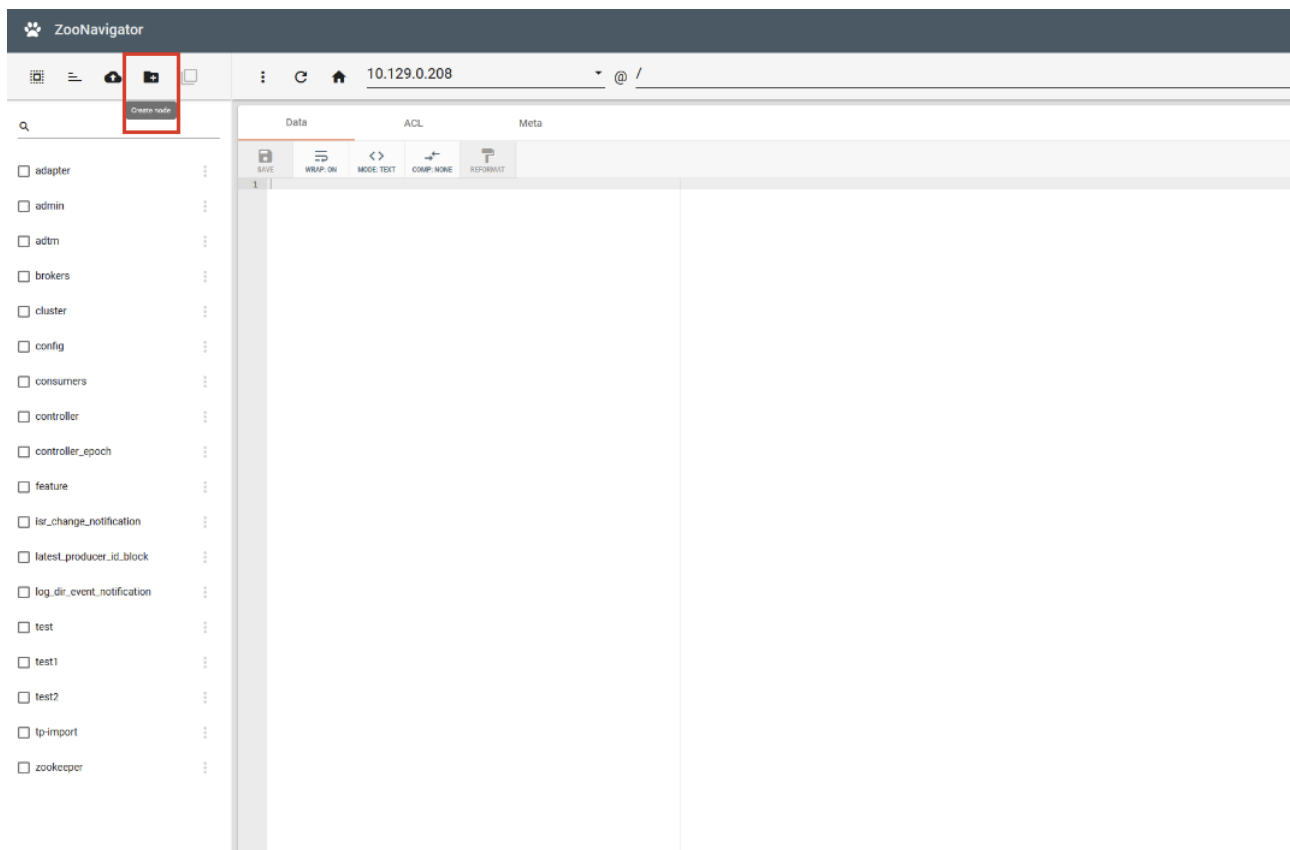


рис. 17 Кнопка «Create node»

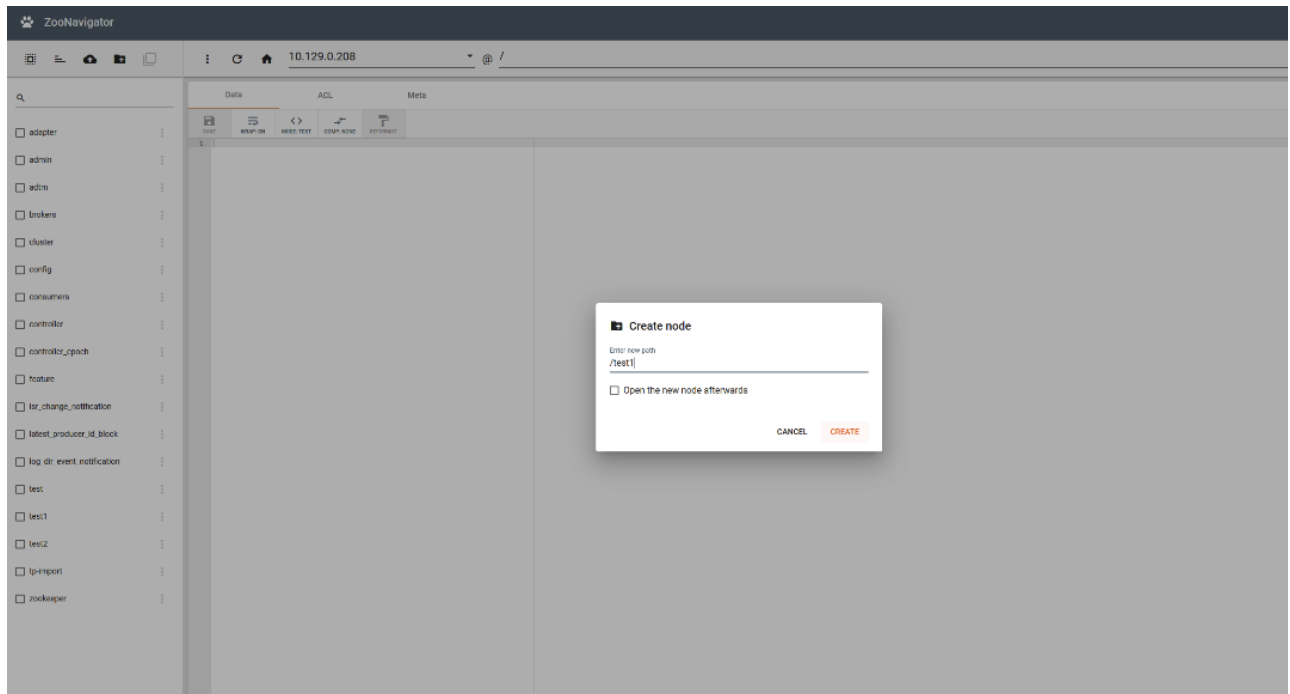


рис. 18 Окно создания ноды

Для удаления ноды, необходимо выделить ее, открыть меню, выбрать удаление и подтвердить действие во всплывающем окне

Удаление происходит рекурсивно т.е. удаляется сама нода и все ее потомки

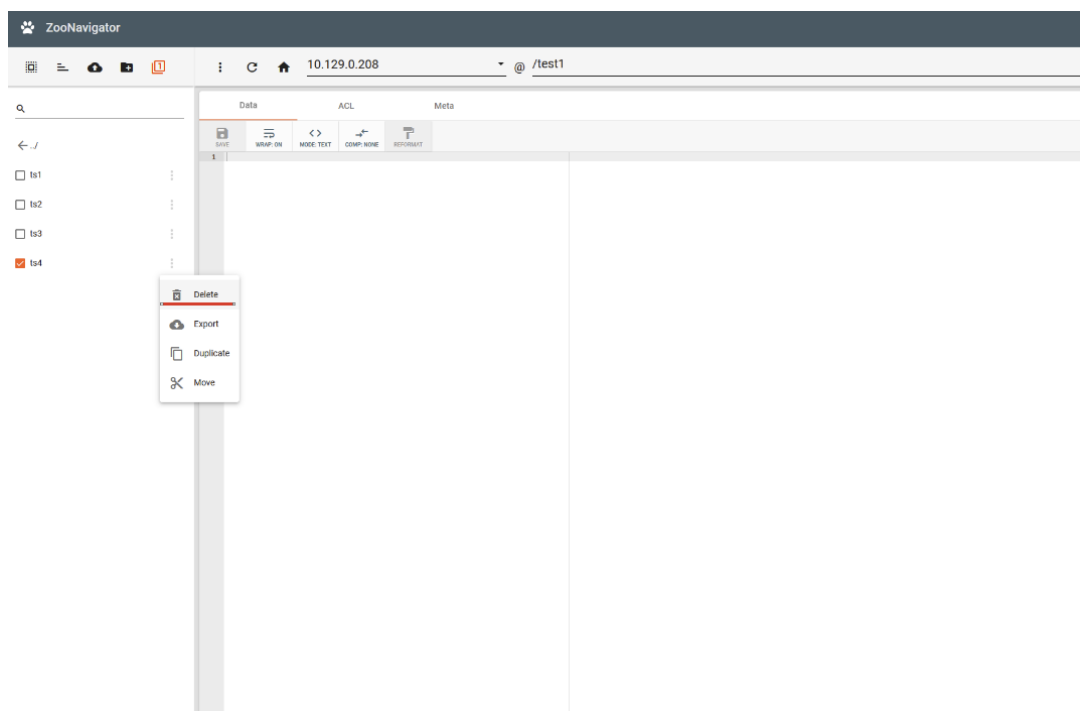


рис. 19 Удаление ноды ts4

Для массового удаления нод, можно воспользоваться кнопками «Toggle select all» и «Selected node actions» в левом верхнем углу. «Toggle select all» выделяет все ноды в текущем каталоге, а «Selected node actions» дает возможность удалить все выбранные ноды или экспортировать их.

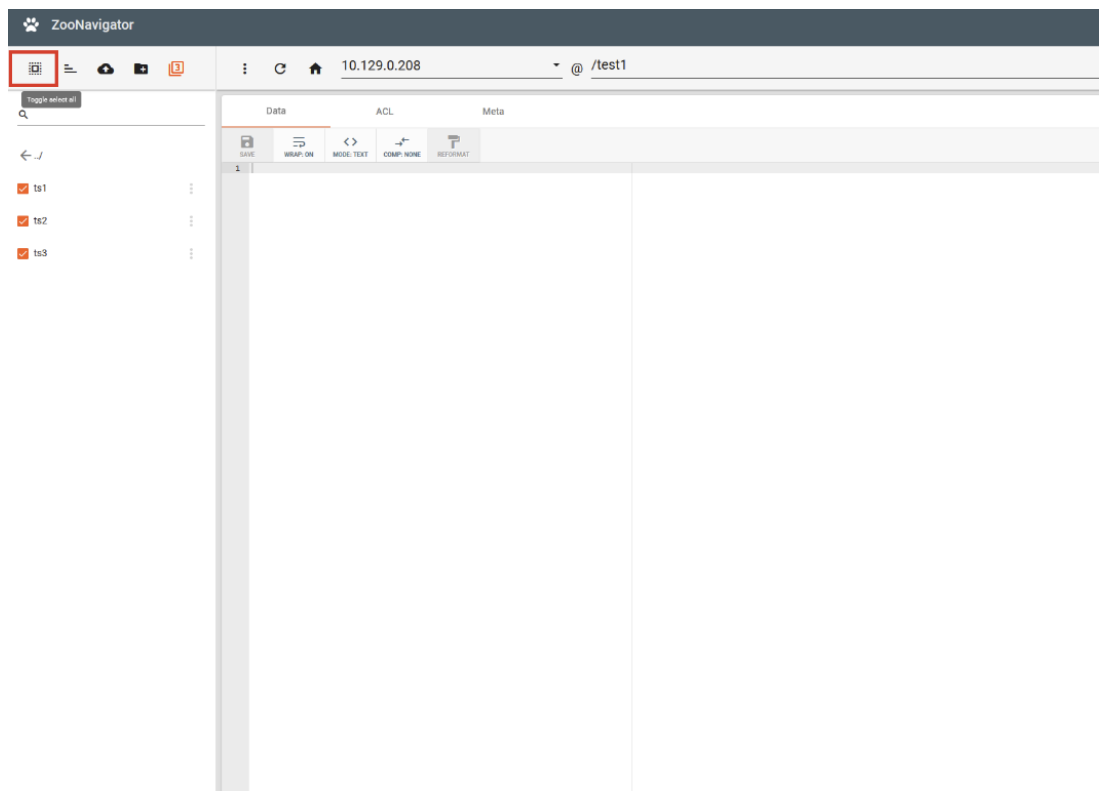


рис. 20 Кнопка «Toggle select all»

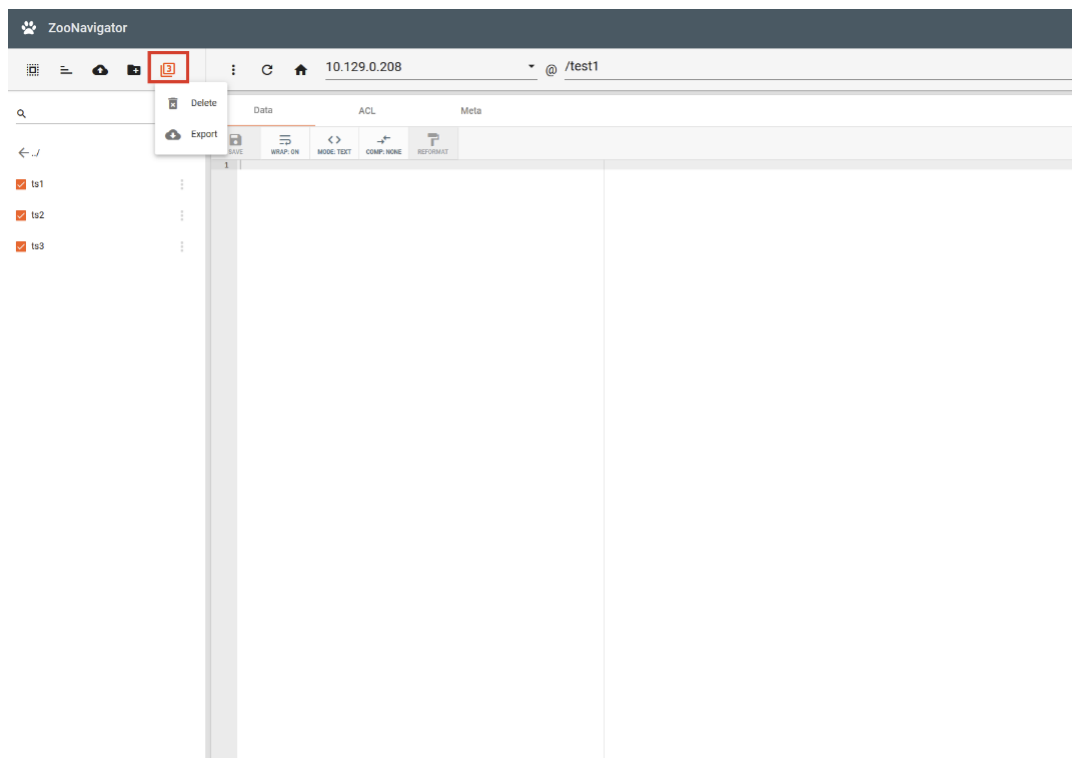


рис. 21 Кнопка «Selected node actions»

Для перемещения ноды в другой каталог, необходимо выделить ноду, в меню выбрать «Move» и указать новый путь

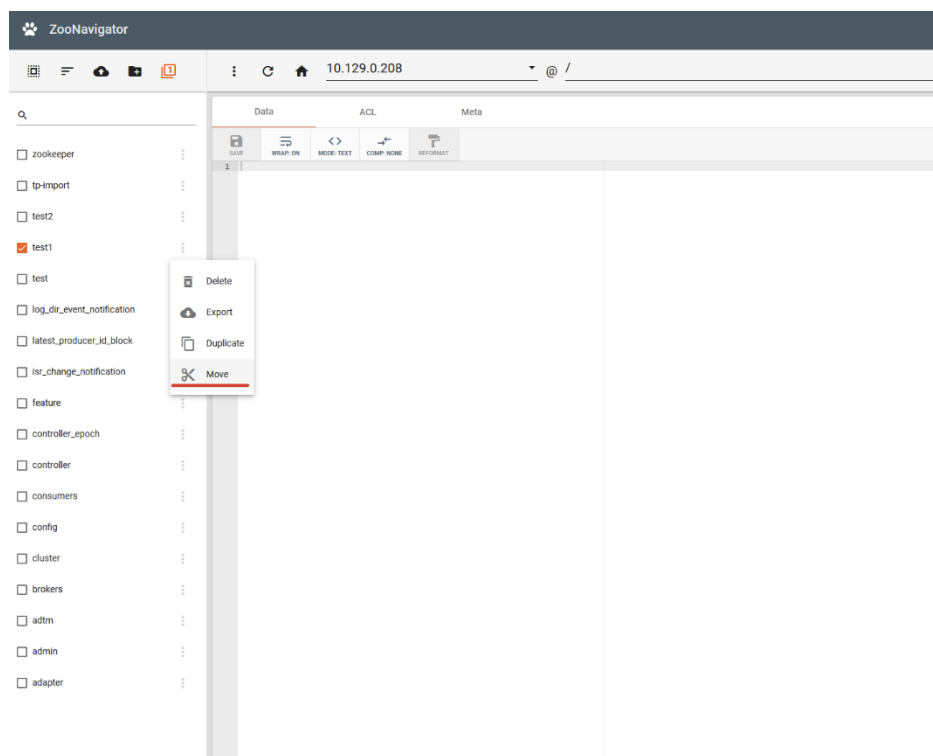


рис. 22 Перенос ноды test1 в каталог test2

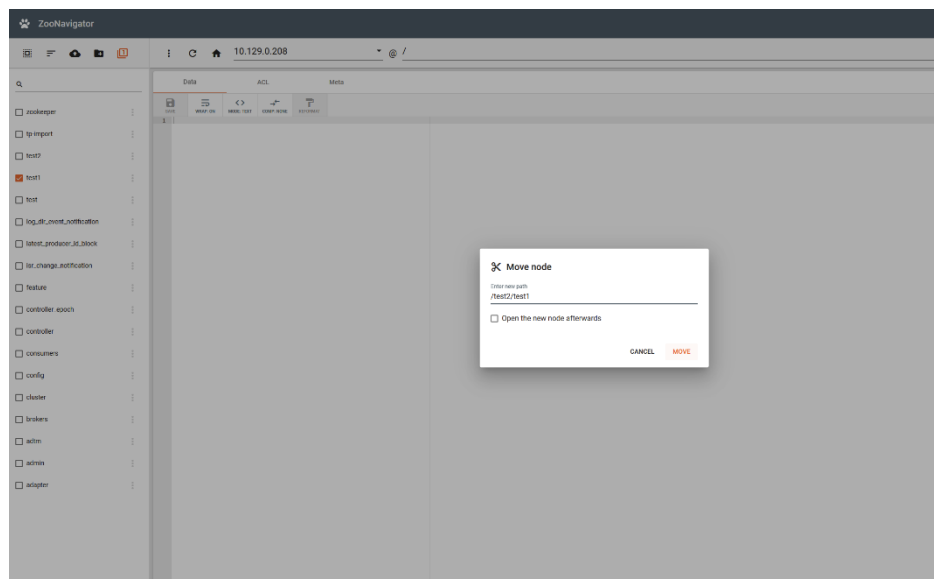


рис. 23 Перенос ноды test1 в каталог test2

При переносе необходимо указывать абсолютный путь, иначе может возникнуть ошибка KeeperErrorCode = NodeExists

3.2 Изменение информации

Для изменения информации, необходимо выбрать ноду (если в ней что-то записано информация отобразится в центральной части интерфейса), изменить какое-либо значение и нажать кнопку «Save» для сохранения изменений

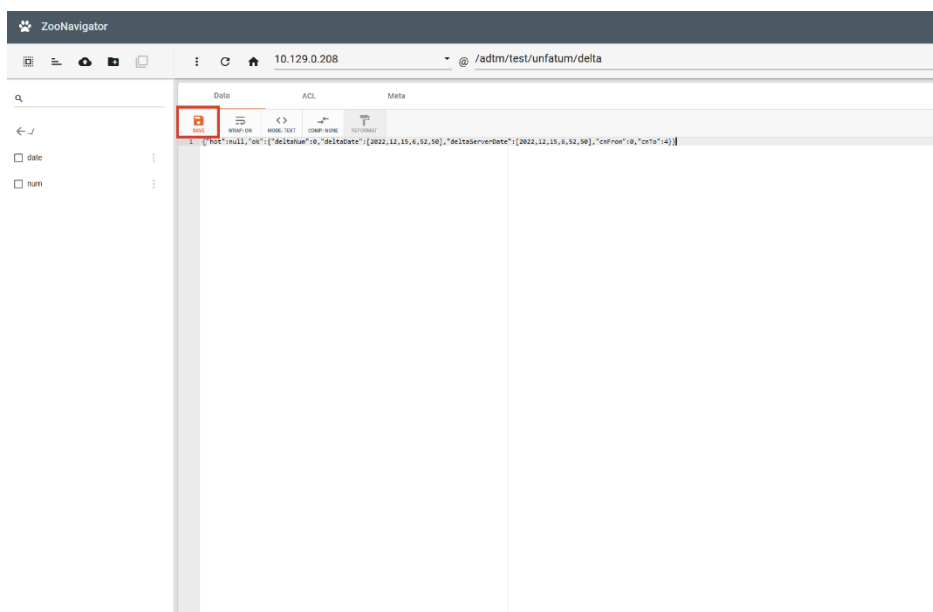


рис. 24 Кнопка «Save»

Е.1.3 OffsetExplorer. Описание инструмента и его функциональность

Е.1.3.1 Описание программы

Offset Explorer - это приложение с графическим интерфейсом для работы с кластерами Kafka. Основная цель - предоставить интуитивно понятный пользовательский интерфейс, который позволяет быстро просматривать объекты в кластере Kafka, а также сообщения, хранящиеся в темах кластера.

Основные особенности:

- Получение списка топиков и информации от разных кластеров Kafka на едином интерфейсе
- Чтение сообщений внутри топиков
- Выгрузка отдельных сообщений
- Изменение конфигурации топиков
- Добавление\удаление топиков из Kafka
- Чтение сообщений нескольких типов: строка, массив байт, Avro

Е.1.3.2 Установка и подключение

Скачать и установить OffsetExplorer можно через официальный сайт. Доступны версии для Windows, Linux и Mac OS.

Руководство не добавляет кластеры/сервера в Kafka, а лишь пытается подключиться к указанным

Для начала работы с Kafka необходимо добавить в утилиту хотя бы один кластер. Сделать это можно через окно "Add new cluster"

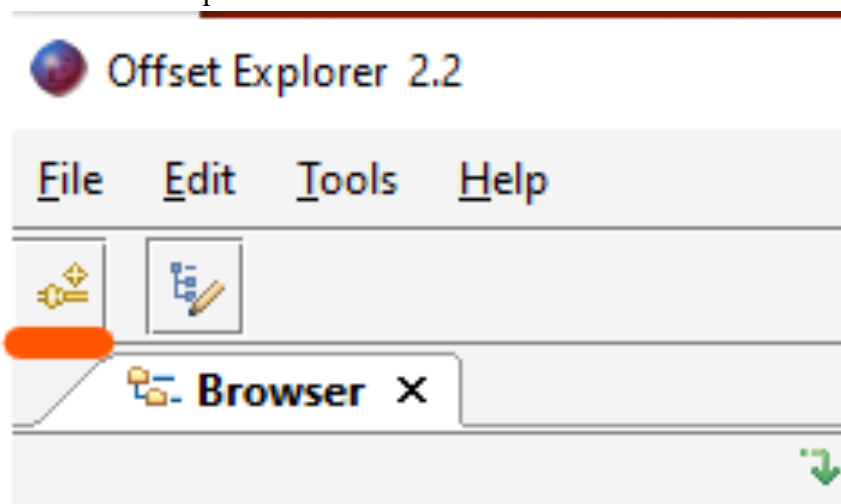


рис. 25 Расположение кнопки "Add new cluster"

После нажатия на клавишу откроется окно добавления кластера, где необходимо заполнить параметры подключения и нажать кнопку "Add"

Параметры подключения:

- Cluster name - пользовательское имя добавляемого кластера. Необходимо для отображения в Приложении.

- Kafka Cluster Version - версия установленной Kafka.
- Zookeeper Host - адрес хоста с Zookeeper или одного из хостов кластера Zookeeper, который используется Kafka. Кнопка "Ping" позволяет проверить доступность адреса и порта.
- Zookeeper Port - порт, используемый Zookeeper на хосте, указанном в Zookeeper Host.
- chroot path — путь, по которому данные кластера kafka отображаются в Zookeeper.

Можно проверить корректность введенных данных, используя кнопку "Test" внизу окна. В случае успеха Offset Explorer предложит добавить подключение

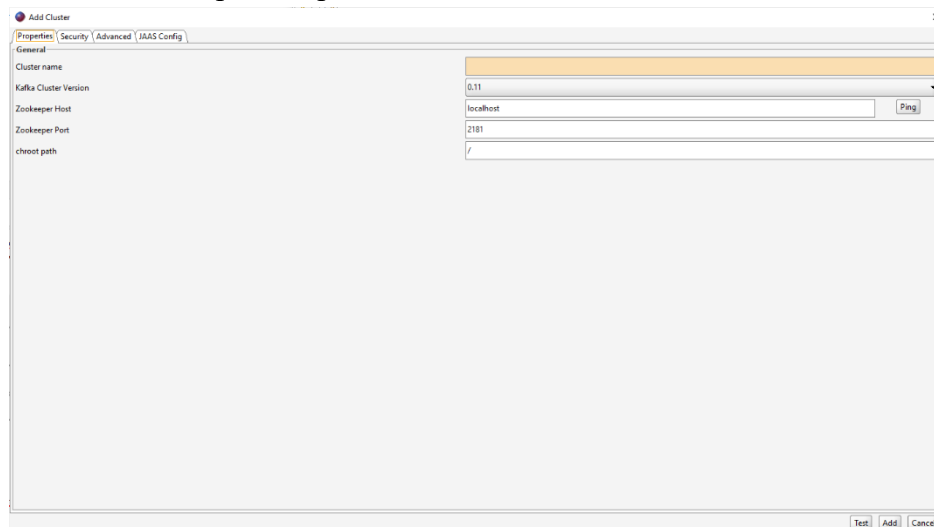


рис. 26 Окно добавления кластера

Е.1.3.3 Работа в программе

Е.1.3.3.1 Создание и удаление топиков

Для создания нового топика, необходимо выбрать вкладку "Topics" в нужном кластере и нажать кнопку "Add New Topic".

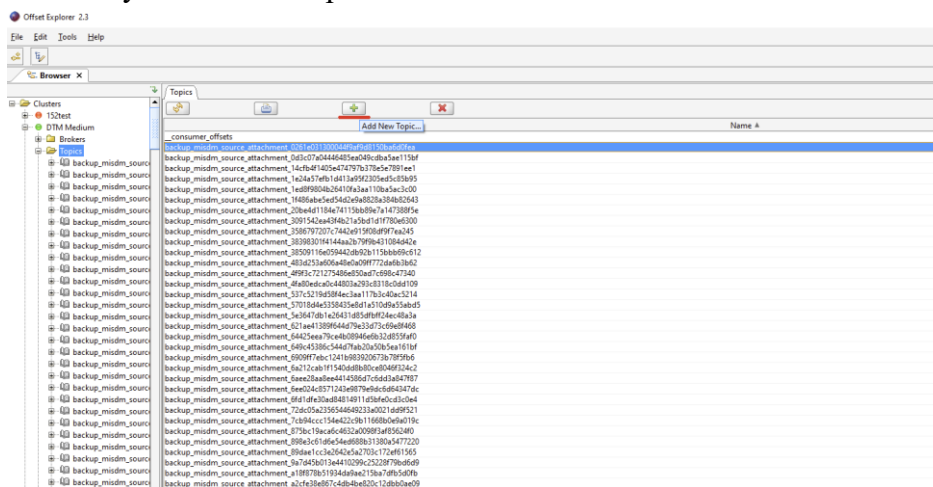


рис. 27 Расположение кнопки "Add New Topic"

В открывшемся окне необходимо указать название топика, количество партиций, количество реплик(так же есть возможность изменить стандартную конфигурацию, при необходимости) и нажать на кнопку "Add" для добавления топика в кластер

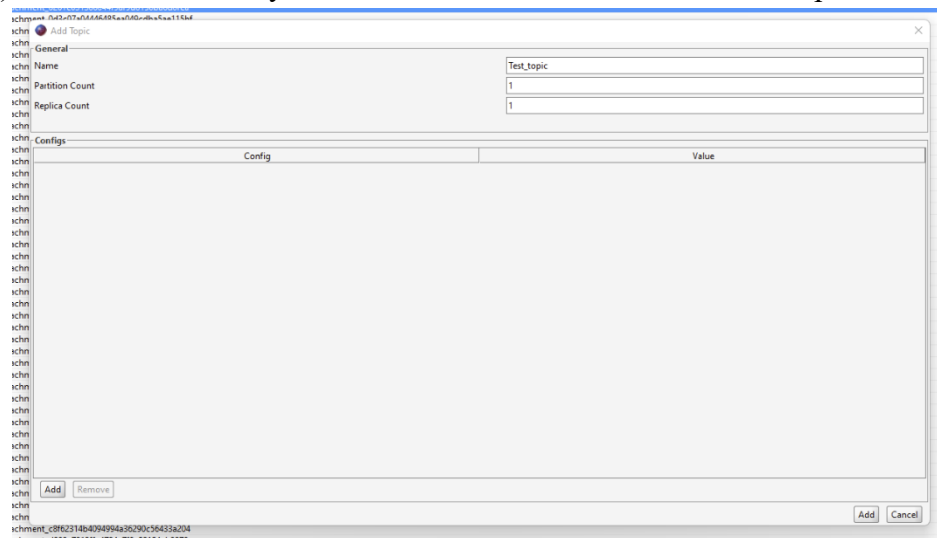


рис. 28 Окно создания нового топика

Для удаления топика необходимо выбрать вкладку "Topics" в нужном кластере и нажать кнопку "Delete Topic". Во всплывающем окне необходимо подтвердить удаление топика со всеми данными.

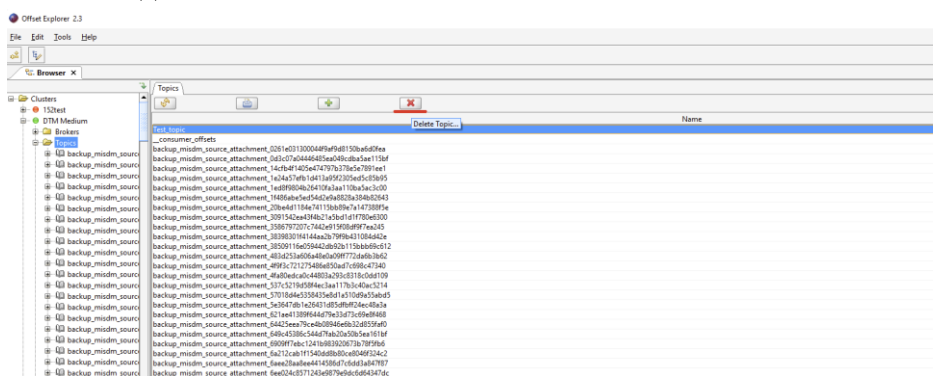


рис. 29 Расположение кнопки "Delete Topic"

Топики не удаляются сразу, а помечаются на удаление. Кафка удаляет их спустя какое-то время.

Е.1.3.3.2 Проверка наличия сообщений в топике

Для того, чтобы узнать, есть ли сообщения в топике можно воспользоваться вкладкой "Properties" на окне обзора топика. После чего, в блоке "Messages" нажать кнопку "Refresh" и получить информацию во вкладке "Data".

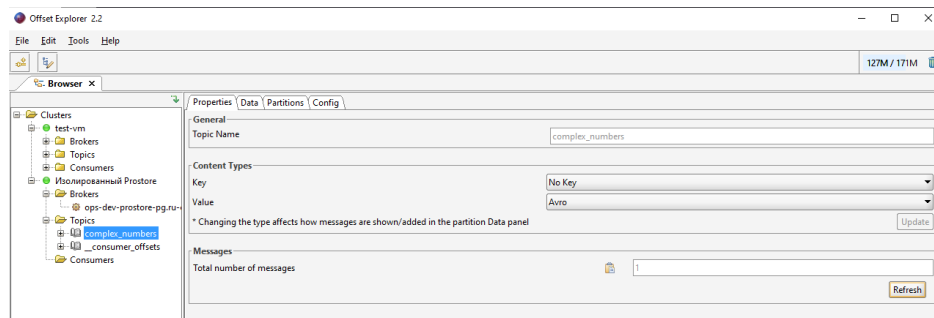


рис. 30 Окно обзора топика

Е.1.3.3.3 Добавление сообщений в топик

Для добавления сообщения в топик, необходимо выбрать топик, перейти в во вкладку "Partitions", нажать на кнопку "Add Message" и выбрать в выпадающем списке добавление одного или нескольких сообщений.

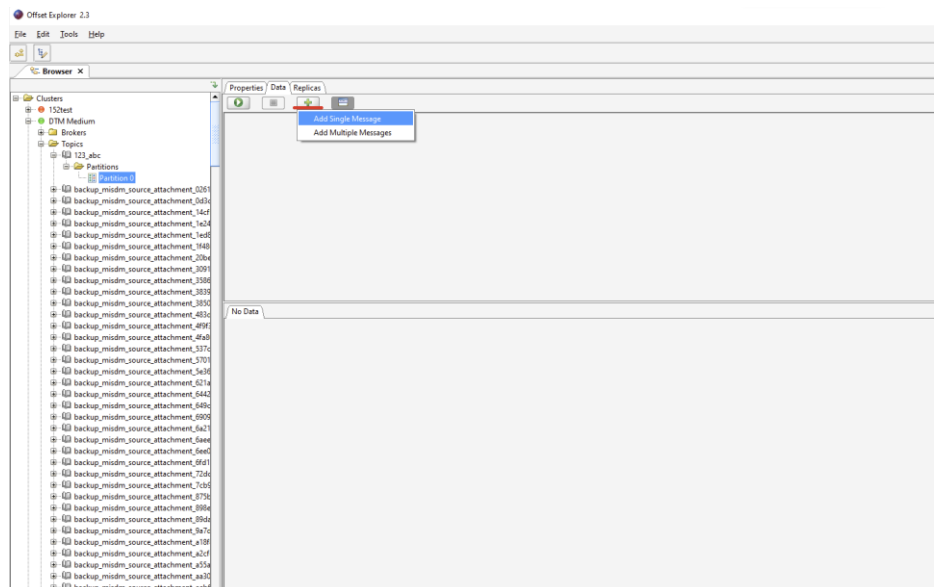


рис. 31 Расположение кнопки "Add Message"

В открывшемся окне можно выбрать загрузку сообщения из файла или ввести ручную (отдельные настройки для ключа и значения). Для добавления сообщения необходимо нажать кнопку "Add".

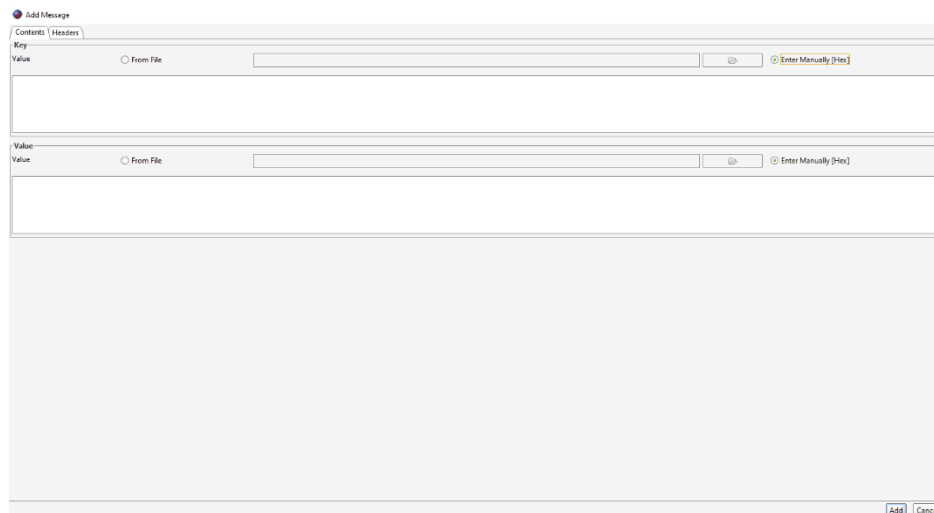


рис. 32 Окно добавления сообщения

Е.1.3.3.4 Выгрузка отдельных сообщений

Для выгрузки сообщений необходимо перейти на вкладку "Data" в Окне обзора топика и получить сообщения (нажать на кнопку "Retrieve messages", подчёркнута зелёным), после чего выбрать то сообщение, которое необходимо скачать, открыть вкладку с информацией (кнопка "Show/Hide Detail Panel", подчёркнута синим), выбрать вкладку "Value" (подчёркнута жёлтым) и нажать кнопку "Save data to file..." (подчёркнута красным), после чего указать файл, куда поместить содержимое.

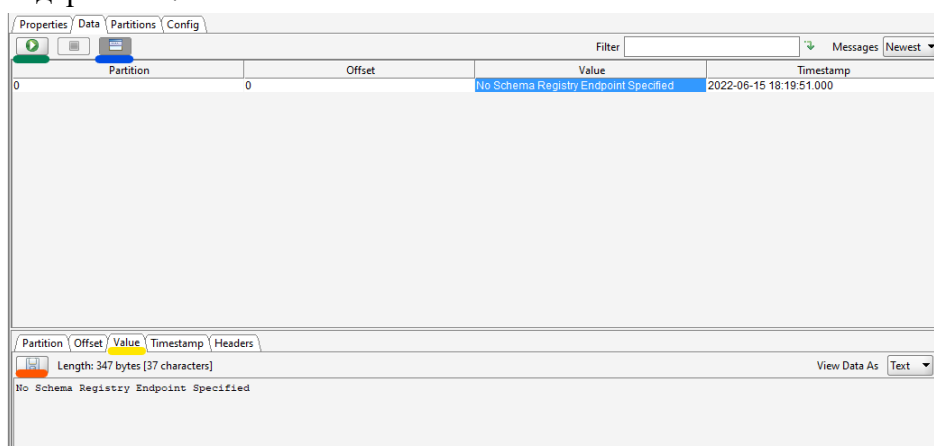


рис. 33 Окно обзора сообщений внутри топика

Е.1.4 VisualVM. Описание инструмента и его функциональность

Е.1.4.1 Описание программы

Утилита VisualVM применяется для сбора и анализа JVM метрик и решения задач профилирования Java приложений.

Е.1.4.2 Установка и подключение

Установка VisualVM описана в официальной документации.

Рассмотрим подключение VisualVM к ПОДД адаптерам, JVM метрики которых собираются в формате, подходящем для анализа утилитами профилирования.

Необходимо скачать данную утилиту на компьютер, с которого будет производиться мониторинг, и произвести некоторые дополнительные настройки в Portainer витрины.

Заходим в Portainer, выбираем контейнер с нужным ПОДД-адаптером, например, **podd-adapter-query**.

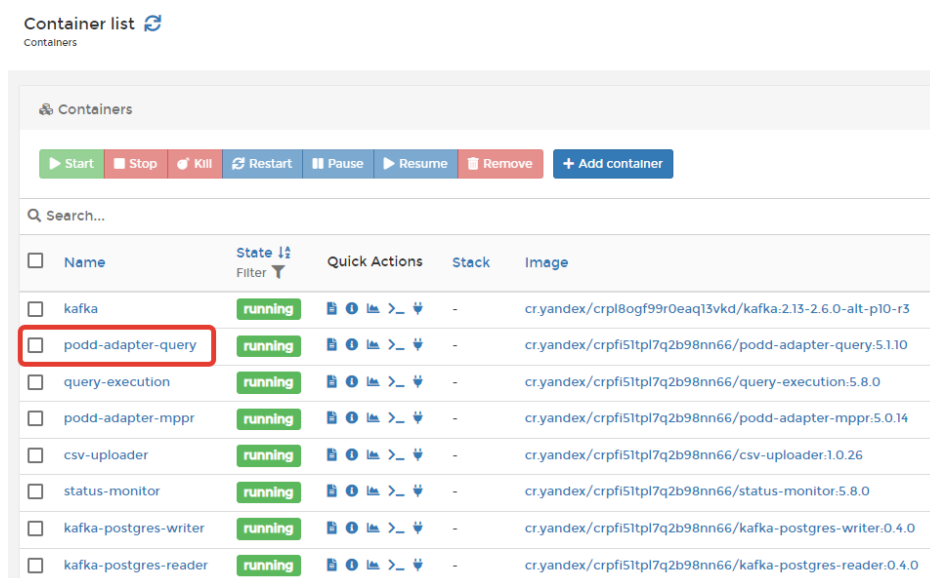


рис. 34 podd-adapter-query

Порты для сбора метрик на адаптерах обычно имеют формат **98xx** и могут быть указаны в переменной контейнера **METRICS_PORT**

Наличие переменной **METRICS_PORT** в разделе **ENV** в Portainer не обязательно, внутри контейнера она существует с дефолтным значением.

Но сам порт для сбора метрик обязательно должен быть проброшен из хостовой машины витрины внутрь контейнера:

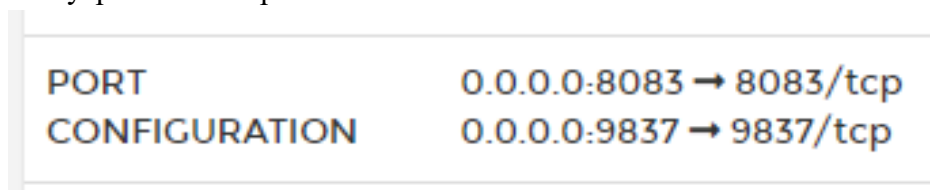


рис. 35 Порт для сбора метрик 9837

Далее для корректной работы **VisualVM** нужно добавить в окружении контейнера новую переменную **JAVA_OPTS** и значение **-Dcom.sun.management.jmxremote=true** -
Dcom.sun.management.jmxremote.port=9837 -
Dcom.sun.management.jmxremote.local.only=false -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false -
Dcom.sun.management.jmxremote.rmi.port=9837 -Djava.rmi.server.hostname=10.129.0.127

В переменной необходимо указать верные значения для порта метрик (в двух местах) и ip-адрес хостовой машины витрины (не внутренний адрес контейнера)

JAVA_OPTS	-Dcom.sun.management.jmxremote=true -Dcom.sun.management.jmxremote.port=9837 -Dcom.sun.management.jmxremote.local.only=false -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.rmi.port=9837 -Djava.rmi.server.hostname=10.129.0.127
-----------	--

рис. 36 переменную **JAVA_OPTS** и значение

Е.1.4.3 Работа в программе

Интерфейс VisualVM прозрачен и интуитивно понятен.

Е.1.4.3.1 Вкладка "Monitor"

На данной вкладке строятся графики с основными показателями проверяемой системы.

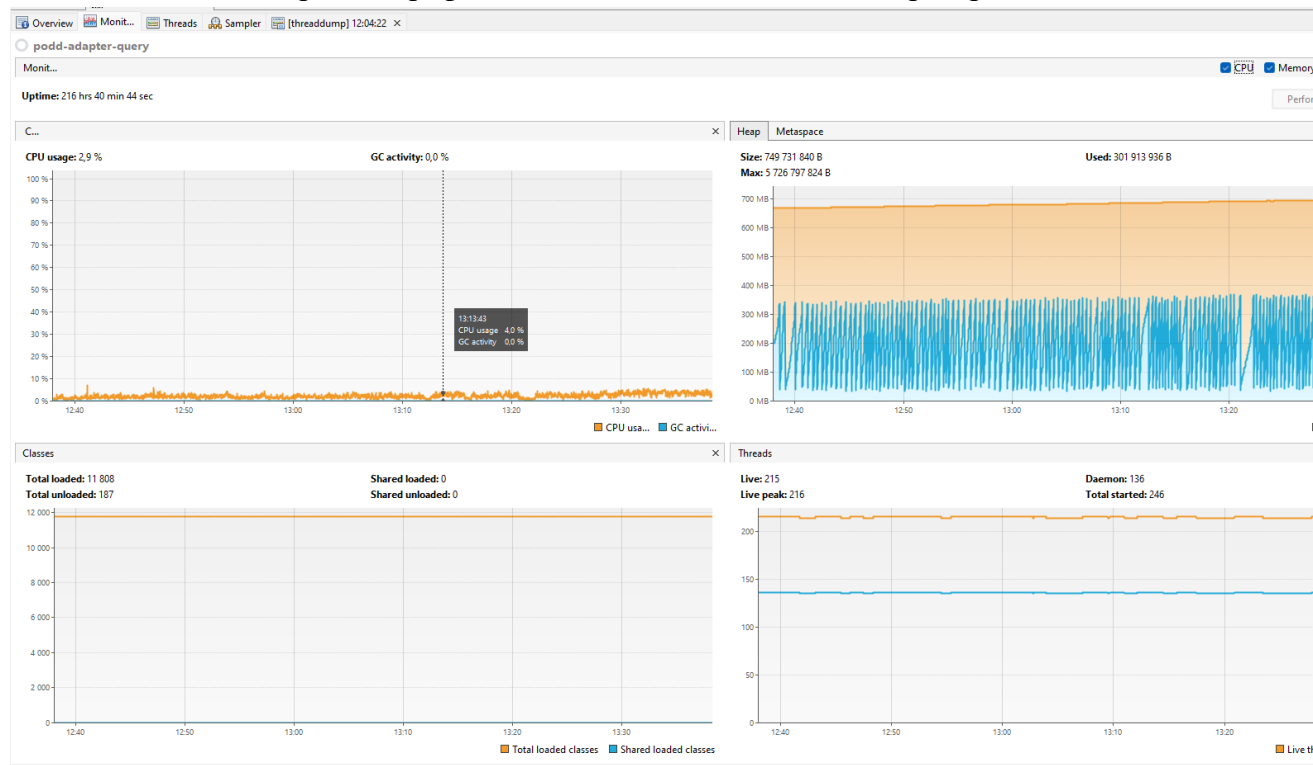


рис. 37 Графики

1. График "CPU"

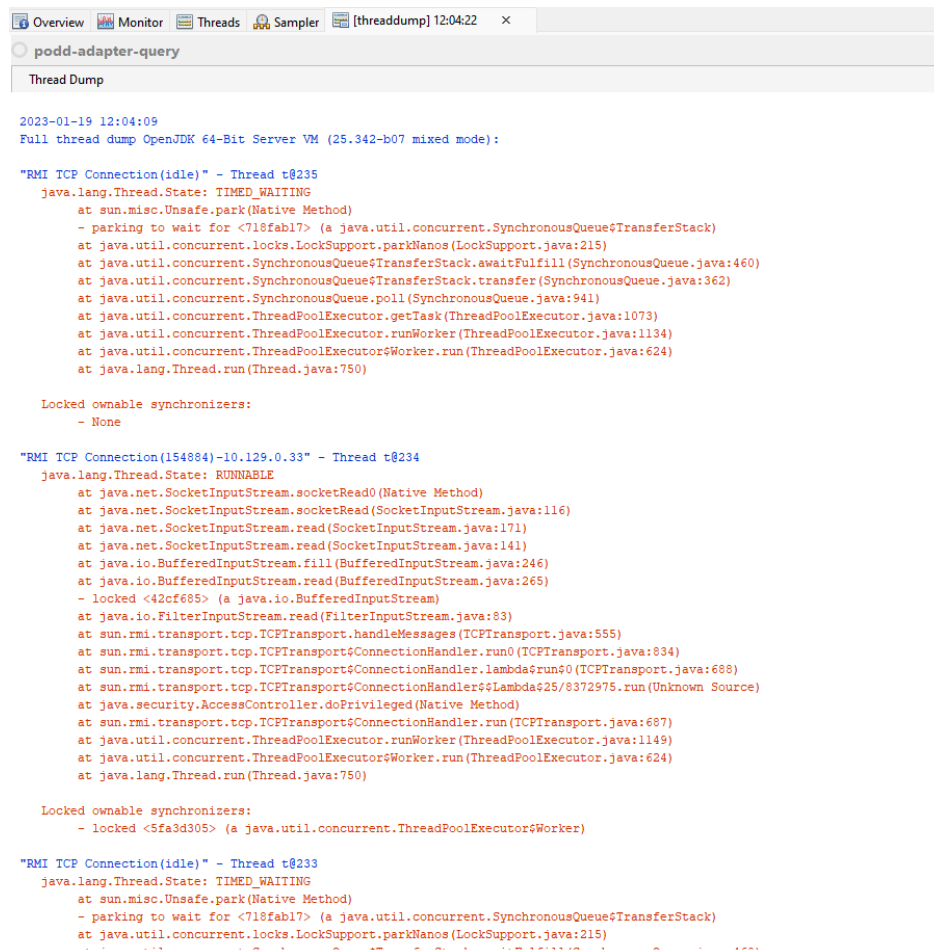


рис. 39 дамп потока

Е.1.4.3.2 Вкладка "Sampler"

Семплирование - мощный инструмент для анализа скорости работы JVM приложения. Он позволяет выполнять замеры времени работы каждого метода, смотреть на каком именно этапе приложение потратило больше всего времени и ресурсов, сколько объектов было создано и тд.

Статистика собирается по потреблению ресурсов процессора и памяти отдельно, кроме того, в разных проекциях, что позволяет использовать множество подходов для анализа.

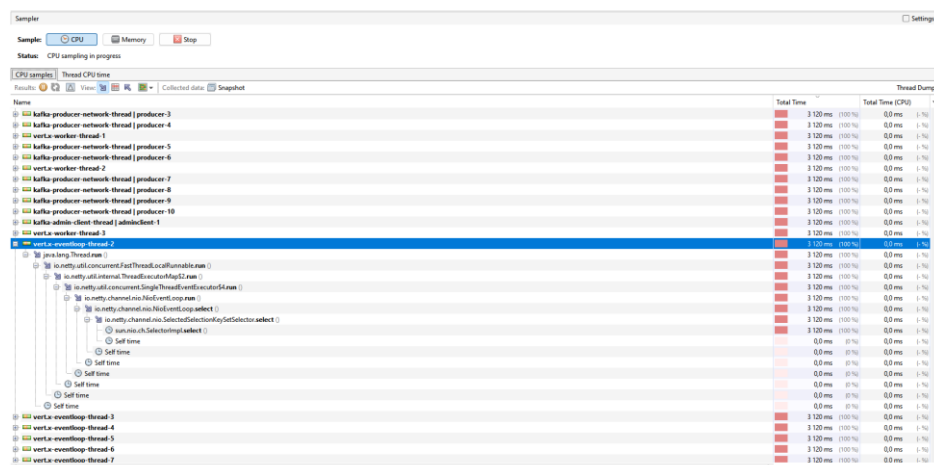


рис. 40 Семплирование

Тут, помимо времени выполнения, следует обратить внимание на статус потока. Особенно на статусы *Park*, *Sleep*, *Wait*

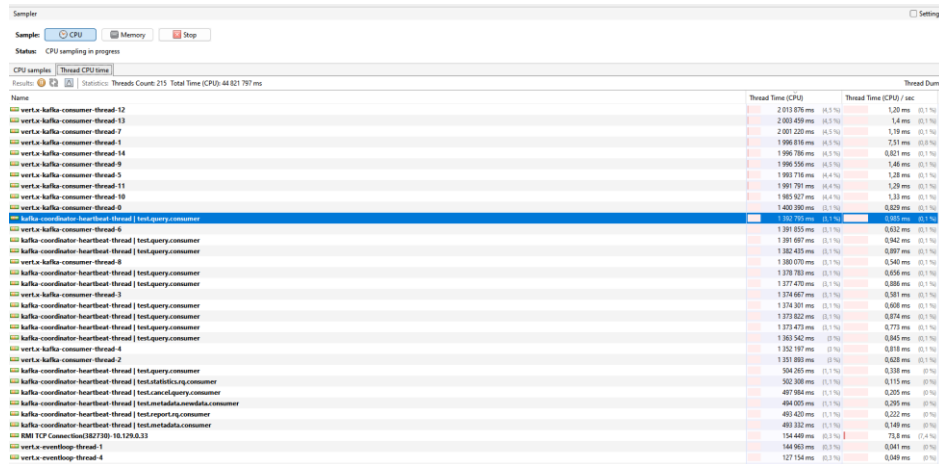


рис. 41 Процессорное время по потокам

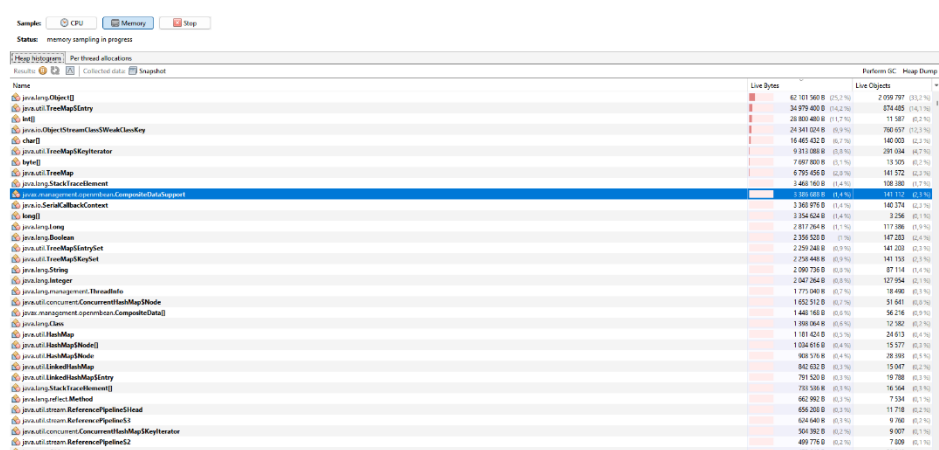


рис. 42 Потребление динамической памяти

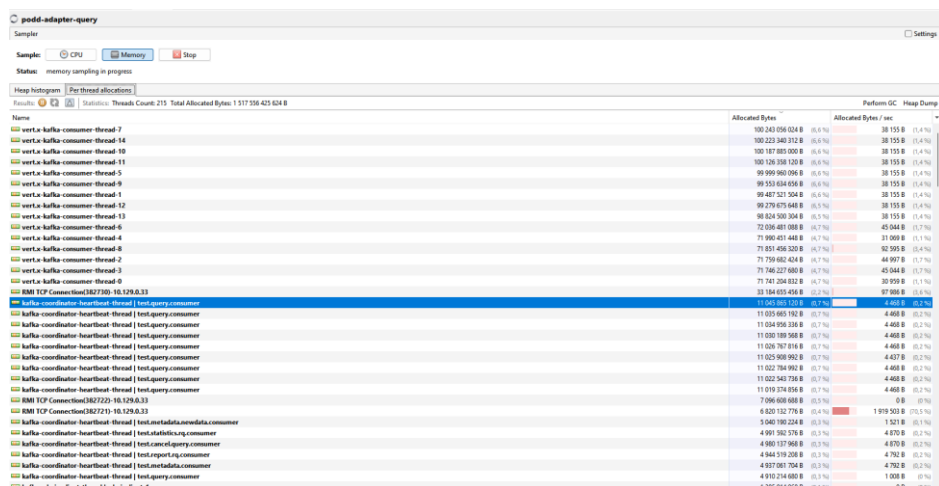


рис. 43 Статистика динамической памяти по потокам

E.2 POSTGRESQL. Настройка и обслуживание

E.2.1 Периодическое обслуживание

E.2.1.1 VACUUM

Для удаления накопившегося мусора и анализа базы данных, рекомендуется выполнять команду VACUUM.

E.2.1.1.1 Описание

VACUUM высвобождает пространство, занимаемое «мёртвыми» кортежами. При обычных операциях Postgres Pro кортежи, удалённые или устаревшие в результате обновления, физически не удаляются из таблицы; они сохраняются в ней, пока не будет выполнена команда VACUUM. Таким образом, периодически необходимо выполнять VACUUM, особенно для часто изменяемых таблиц.

Без параметра команда VACUUM обрабатывает все таблицы в текущей базе данных, которые может очистить текущий пользователь. Если в параметре передаётся имя таблицы, VACUUM обрабатывает только эту таблицу.

VACUUM ANALYZE выполняет очистку (VACUUM), а затем анализ (ANALYZE) всех указанных таблиц. Это удобная комбинация для регулярного обслуживания БД. За дополнительной информацией об анализе обратитесь к описанию ANALYZE.

Простая команда VACUUM (без FULL) только высвобождает пространство и делает его доступным для повторного использования. Эта форма команды может работать параллельно с обычными операциями чтения и записи таблицы, так она не требует исключительной блокировки. Однако освобождённое место не возвращается операционной системе (в большинстве случаев); оно просто остаётся доступным для размещения данных этой же таблицы. VACUUM FULL переписывает всё содержимое таблицы в новый файл на диске, не содержащий ничего лишнего, что позволяет вернуть неиспользованное пространство операционной системе. Эта форма работает намного медленнее и запрашивает исключительную блокировку для каждой обрабатываемой таблицы.

Когда список параметров заключается в скобки, параметры могут быть записаны в любом порядке. Без скобок параметры должны указываться именно в том порядке, который показан выше. Синтаксис со скобками появился в PostgreSQL 9.0; вариант записи без скобок считается устаревшим.

E.2.1.1.2 Синтаксис

```
VACUUM [ ( { FULL | FREEZE | VERBOSE | ANALYZE } [, ...] ) ] [ имя_таблицы [
(имя_столбца [, ...] ) ] ]
```

Параметры

FULL

Выбирает режим «полной» очистки, который может освободить больше пространства, но выполняется гораздо дольше и запрашивает исключительную блокировку таблицы. Этот режим

также требует дополнительное место на диске, так как он записывает новую копию таблицы и не освобождает старую до завершения операции. Обычно это следует использовать, только когда требуется высвободить значительный объём пространства, выделенного таблице.

`FREEZE`

Выбирает агрессивную «заморозку» кортежей. Добавление указания `FREEZE` равносильно выполнению команды `VACUUM` с параметрами `vacuum_freeze_min_age` и `vacuum_freeze_table_age`, равными нулю. Агрессивная заморозка всегда выполняется при перезаписи таблицы, поэтому в режиме `FULL` это указание избыточно.

`VERBOSE`

Выводит подробный отчёт об очистке для каждой таблицы.

`ANALYZE`

Обновляет статистику, которую использует планировщик для выбора наиболее эффективного способа выполнения запроса.

`имя_таблицы`

Имя (возможно, дополненное схемой) определённой таблицы, подлежащей очистке. По умолчанию очищаются все таблицы в текущей базе данных.

`имя_столбца`

Имя столбца, подлежащего анализу. По умолчанию анализируются все столбцы. Если указан список столбцов, подразумевается операция `ANALYZE`.

Е.2.1.1.3 Пример

Очистка одной `имя_таблицы = testTBLname_actual`, проведение её анализа для оптимизатора и печать подробного отчёта о действиях операции очистки:

```
VACUUM (VERBOSE, ANALYZE) testTBLname_actual;
```

Существует два варианта `VACUUM`: обычный `VACUUM` и `VACUUM FULL`. Команда `VACUUM FULL` может высвободить больше дискового пространства, однако работает медленнее. Кроме того, обычная команда `VACUUM` может выполняться параллельно с использованием производственной базы данных. (При этом такие команды как `SELECT`, `INSERT`, `UPDATE` и `DELETE` будут выполняться нормально, хотя нельзя будет изменить определение таблицы командами типа `ALTER TABLE`.) Команда `VACUUM FULL` требует блокировки обрабатываемой таблицы в режиме `ACCESS EXCLUSIVE` и поэтому не может выполняться параллельно с другими операциями с этой таблицей. По этой причине администраторы, как правило, должны стараться использовать обычную команду `VACUUM` и избегать `VACUUM FULL`.

Команда `VACUUM` порождает существенный объём трафика ввода/вывода, который может стать причиной низкой производительности в других активных сеансах.

Рекомендуемая периодичность: 1-2 раза в месяц в период минимальной активности базы (выходные, ночью).

Е.2.2 Планировщик запросов

Е.2.2.1 ANALYZE

Планировщик запросов в PostgreSQL, выбирая эффективные планы запросов, полагается на статистическую информацию о содержимом таблиц. Эта статистика собирается командой ANALYZE, которая может вызываться сама по себе или как дополнительное действие команды VACUUM. Статистика должна быть достаточно точной, так как в противном случае неудачно выбранные планы запросов могут снизить производительность базы данных.

Е.2.2.1.1 Описание

ANALYZE собирает статистическую информацию о содержимом таблиц в базе данных и сохраняет результаты в системном каталоге pg_statistic. Впоследствии планировщик запросов будет использовать эту статистику для выбора наиболее эффективных планов выполнения запросов.

Без списка `таблица_и_столбцы` команда ANALYZE обрабатывает все таблицы и материализованные представления в текущей базе данных, на анализ которых текущий пользователь имеет право. Со списком ANALYZE обрабатывает только указанные в нём таблицы. Дополнительно можно задать для таблицы список имён столбцов, в этом случае статистика будет собираться только по этим столбцам.

Е.2.2.1.2 Синтаксис

```
ANALYZE [ ( параметр [, ...] ) ] [ таблица_и_столбцы [, ...] ]
```

Параметры
VERBOSE

Включает вывод сообщений о процессе выполнения.

SKIP_LOCKED

Указывает, что команда ANALYZE не должна ждать освобождения конфликтующих блокировок, начиная обработку отношения: если отношение не удаётся заблокировать сразу, без ожидания, оно пропускается. Заметьте, что даже с этим указанием ANALYZE может заблокироваться, открывая индексы отношения или получая выборку строк из секций, потомков в иерархии наследования или некоторых видов сторонних таблиц. Учтите также, что при наличии конфликтующей блокировки в секционированной таблице команда ANALYZE пропускает все её секции, тогда как обычно все они обрабатываются.

логическое_значение

Включает или отключает заданный параметр. Для включения параметра можно написать TRUE, ON или 1, а для отключения — FALSE, OFF или 0. Параметр логическое_значение можно опустить, в этом случае подразумевается TRUE.

имя_таблицы

Имя (возможно, дополненное схемой) определённой таблицы, подлежащей анализу. Если опущено, анализироваться будут все обычные и секционированные таблицы, а также материализованные представления в текущей базе данных (но не сторонние таблицы). Если задано имя секционированной таблицы, обновлена будет как статистика наследования для этой таблицы, так и статистика отдельных её секций.

имя_столбца

Имя столбца, подлежащего анализу. По умолчанию анализируются все столбцы.

Е.2.2.1.3 Пример

```
ANALYZE (VERBOSE) testdbname.testtblname_actual;
```

Результат:

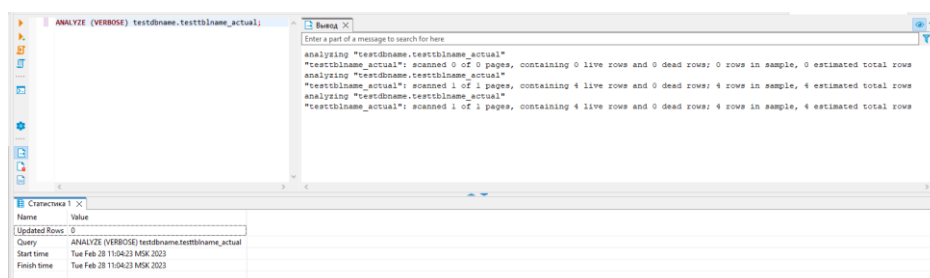


рис. 44 Результат выполнения команды Analyze с параметром Verbose

Е.2.3 Перестройка Индексов

В некоторых ситуациях стоит периодически перестраивать индексы, выполняя команду REINDEX или последовательность отдельных шагов по восстановлению индексов.

Е.2.3.1 REINDEX

Е.2.3.1.1 Описание

REINDEX перестраивает индекс, обрабатывая данные таблицы, к которой относится индекс, и в результате заменяет старую копию индекса. Команда REINDEX применяется в следующих ситуациях:

- Индекс был повреждён, его содержимое стало некорректным.
- Индекс стал «раздутым», то есть в нём оказалось много пустых или почти пустых страниц.
- Параметр хранения индекса (например, фактор заполнения) был изменён, и теперь требуется, чтобы это изменение вступило в силу в полной мере.
- Построение индекса с параметром CONCURRENTLY завершилось ошибкой, в результате чего индекс оказался «нерабочим».

Е.2.3.1.2 Синтаксис

```
REINDEX [ ( VERBOSE ) ] { INDEX | TABLE | SCHEMA | DATABASE | SYSTEM } имя
INDEX
```

Перестраивает указанный индекс.

TABLE

Перестраивает все индексы в указанной таблице. Если у таблицы имеется дополнительная таблица «TOAST», она так же переиндексируется.

SCHEMA

Перестраивает все индексы в указанной схеме. Если таблица в этой схеме имеет вторичную таблицу «TOAST», она также будет переиндексирована. При этом обрабатываются и индексы в общих системных каталогах. Эту форму REINDEX нельзя выполнить в блоке транзакции.

DATABASE

Перестраивает все индексы в текущей базе данных. При этом обрабатываются также индексы в общих системных каталогах. Эту форму REINDEX нельзя выполнить в блоке транзакции.

SYSTEM

Перестраивает все индексы в системных каталогах текущей базы данных. При этом обрабатываются также индексы в общих системных каталогах, но индексы в таблицах пользователя не затрагиваются. Эту форму REINDEX нельзя выполнить в блоке транзакции.

ИМЯ

Имя определённого индекса, таблицы или базы данных, подлежащих переиндексации. В настоящее время REINDEX DATABASE и REINDEX SYSTEM могут переиндексировать только текущую базу данных, так что их параметр должен соответствовать имени текущей базы данных.

VERBOSE

Выводит отчёт о прогрессе после переиндексации каждого индекса.

Е.2.3.1.3 Пример

```
REINDEX TABLE testTBLname_actual;
```

Е.2.3.1.4 Пересоздание Индексов

При необходимости можно удалить существующие индексы и создать их заново. Но уже с учетом текущих потребностей.

Для просмотра существующих Индексов, выполните запрос:

```
SELECT * FROM pg_indexes WHERE schemaname = 'testDBname';
```

Результат:

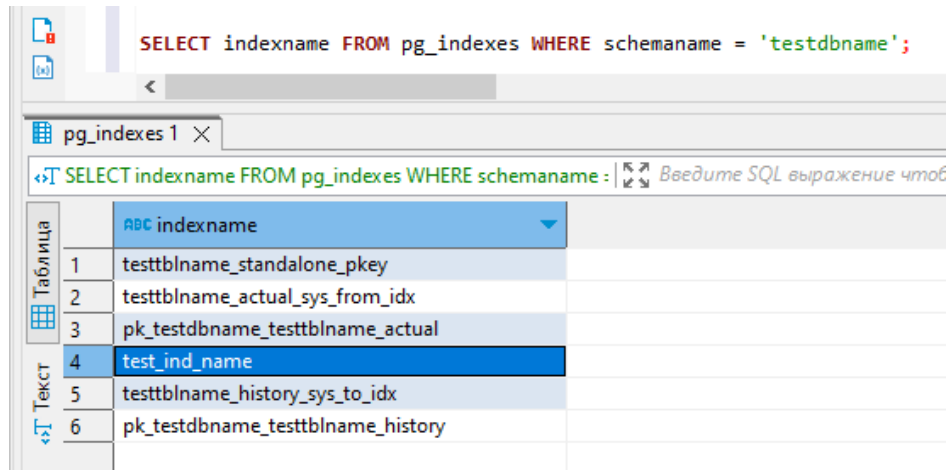


рис. 45 Список Индексов

Е.2.3.1.5 Описание команд

DROP INDEX удаляет существующий индекс из базы данных.

CREATE INDEX создаёт индексы по указанному столбцу(ам) заданного отношения, которым может быть таблица или материализованное представление.

Е.2.3.1.6 Пример

```
DROP INDEX test_ind_name;
```

```
CREATE INDEX test_ind_name_new ON testdbname.testtblname_actual (name_reg);
```

Результат:

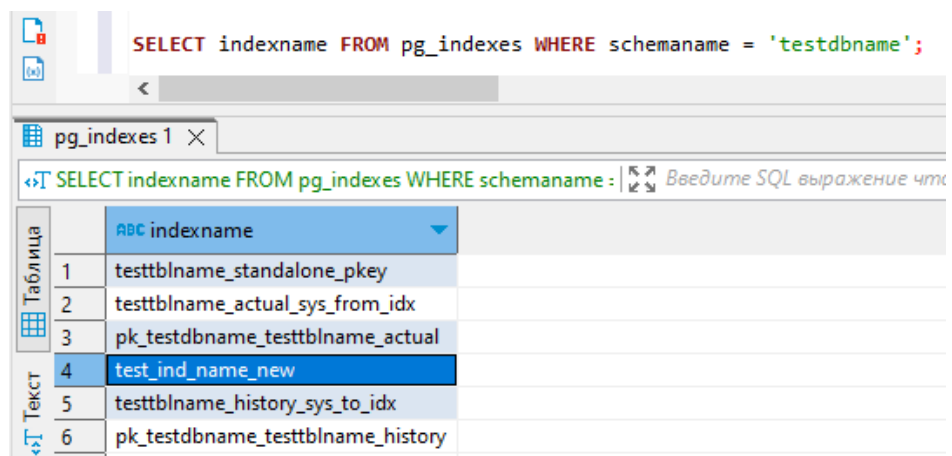


рис. 46 Новый список Индексов

Е.2.4 Динамическое статистическое представление - pg_stat_activity

Команда `pg_stat_activity` отображает одну строку для каждого серверного процесса с информацией по текущей активности процесса, такой как состояние и текущий запрос. Значения в столбцах `wait_event` и `state` не зависят друг от друга. Если обслуживающий процесс находится в состоянии `active` (активен), он может ожидать какое-то событие, или не ожидать никакого. Если состояние `active` и поле `wait_event` содержит не `NULL`, это означает, что запрос выполняется, но заблокирован чем-то в системе.

Е.2.4.1 Примеры

Определение зависшего процесса:

```
SELECT * FROM pg_stat_activity WHERE state = 'active';
```

Смотрим на PID процесса который завис и пытаемся его завершить:

```
SELECT pg_cancel_backend(<pid of the process>)
```

Если процесс не удалось завершить таким образом, то можно попробовать его убить:

```
SELECT pg_terminate_backend(<pid of the process>)
```

Е.2.5 Просмотр активности PostgreSQL

Иногда полезно видеть, чем сейчас занимается сервер.

Для этого можно выполнить в консоли команду:

```
watch -n 1 'ps auxww | grep ^postgres'
```

Результат

```
Every 1.0s: ps auxww | grep ^postgres                                Wed Mar 1 10:27:37 2023
postgres 1699  0.0  0.2 397568 17504 ?        Ss   05:31   0:00 /usr/postgresql-13/bin/postmaster -D /var/lib/postgresql/13/data/
postgres 1938  0.0  0.0 249804 2048 ?        Ss   05:31   0:00 postgres: logger
postgres 2004  0.0  0.0 397604 3564 ?        Ss   05:31   0:00 postgres: checkpoint
postgres 2005  0.0  0.0 397700 3344 ?        Ss   05:31   0:00 postgres: background writer
postgres 2086  0.0  0.0 397568 6244 ?        Ss   05:31   0:00 postgres: walwriter
postgres 2087  0.0  0.0 398112 3300 ?        Ss   05:31   0:00 postgres: autovacuum launcher
postgres 2088  0.0  0.0 252200 2360 ?        Ss   05:31   0:00 postgres: stats collector
postgres 2089  0.0  0.0 398124 2816 ?        Ss   05:31   0:00 postgres: logical replication launcher
postgres 2011  0.0  0.0 399100 6104 ?        Ss   05:31   0:00 postgres: dtm test 10.129.0.166(39134) idle
postgres 2027  0.0  0.0 398512 5420 ?        Ss   05:31   0:00 postgres: dtm test 10.129.0.166(39116) idle
postgres 2092  0.0  0.0 398512 5352 ?        Ss   05:31   0:00 postgres: dtm test 10.129.0.166(39120) idle
postgres 2251  0.0  0.0 398512 5352 ?        Ss   05:31   0:00 postgres: dtm test 10.129.0.166(39144) idle
postgres 2252  0.0  0.0 398512 5356 ?        Ss   05:31   0:00 postgres: dtm test 10.129.0.166(39146) idle
postgres 2352  0.0  0.0 398512 5360 ?        Ss   05:32   0:00 postgres: dtm test 10.129.0.166(39150) idle
postgres 2483  0.0  0.0 398512 5356 ?        Ss   05:32   0:00 postgres: dtm test 10.129.0.166(39154) idle
postgres 2486  0.0  0.0 398512 5356 ?        Ss   05:32   0:00 postgres: dtm test 10.129.0.166(39156) idle
postgres 4169  0.0  0.0 398512 5356 ?        Ss   05:32   0:00 postgres: dtm test 10.129.0.166(39178) idle
```

рис. 47 Пример результата выполнения команды

Е.3 Настройка и обслуживание Kafka и Zookeeper

Е.3.1 Описание инструментов, входящих в комплект поставки Kafka

В случае отсутствия возможности использования сторонних инструментов для работы с kafka можно воспользоваться стандартными скриптам (Kafka CLI), идущими в комплекте (рис 48).

```
[kafka@44428c53f642 bin]$ ls -al
total 140
drwxr-xr-x 1 kafka root    6 Feb  2 12:12 .
drwxr-xr-x 1 kafka root   69 Feb 27 16:23 ..
-rwxr-xr-x 1 kafka root 1423 Jul 28 2020 connect-distributed.sh
-rwxr-xr-x 1 kafka root 1396 Jul 28 2020 connect-mirror-maker.sh
-rwxr-xr-x 1 kafka root 1420 Jul 28 2020 connect-standalone.sh
-rwxr-xr-x 1 kafka root  861 Jul 28 2020 kafka-acls.sh
-rwxr-xr-x 1 kafka root  873 Jul 28 2020 kafka-broker-api-versions.sh
-rwxr-xr-x 1 kafka root  864 Jul 28 2020 kafka-configs.sh
-rwxr-xr-x 1 kafka root  945 Jul 28 2020 kafka-console-consumer.sh
-rwxr-xr-x 1 kafka root  944 Jul 28 2020 kafka-console-producer.sh
-rwxr-xr-x 1 kafka root  871 Jul 28 2020 kafka-consumer-groups.sh
-rwxr-xr-x 1 kafka root  948 Jul 28 2020 kafka-consumer-perf-test.sh
-rwxr-xr-x 1 kafka root  871 Jul 28 2020 kafka-delegation-tokens.sh
-rwxr-xr-x 1 kafka root  869 Jul 28 2020 kafka-delete-records.sh
-rwxr-xr-x 1 kafka root  866 Jul 28 2020 kafka-dump-log.sh
-rwxr-xr-x 1 kafka root  870 Jul 28 2020 kafka-leader-election.sh
-rwxr-xr-x 1 kafka root  863 Jul 28 2020 kafka-log-dirs.sh
-rwxr-xr-x 1 kafka root  862 Jul 28 2020 kafka-mirror-maker.sh
-rwxr-xr-x 1 kafka root  886 Jul 28 2020 kafka-preferred-replica-election.sh
-rwxr-xr-x 1 kafka root  959 Jul 28 2020 kafka-producer-perf-test.sh
-rwxr-xr-x 1 kafka root  874 Jul 28 2020 kafka-reassign-partitions.sh
-rwxr-xr-x 1 kafka root  874 Jul 28 2020 kafka-replica-verification.sh
-rwxr-xr-x 1 kafka root 10074 Jul 28 2020 kafka-run-class.sh
-rwxr-xr-x 1 kafka root 1376 Jul 28 2020 kafka-server-start.sh
-rwxr-xr-x 1 kafka root 1217 Jul 28 2020 kafka-server-stop.sh
-rwxr-xr-x 1 kafka root  945 Jul 28 2020 kafka-streams-application-reset.sh
-rwxr-xr-x 1 kafka root  863 Jul 28 2020 kafka-topics.sh
-rwxr-xr-x 1 kafka root  958 Jul 28 2020 kafka-verifiable-consumer.sh
-rwxr-xr-x 1 kafka root  958 Jul 28 2020 kafka-verifiable-producer.sh
-rwxr-xr-x 1 kafka root 1714 Jul 28 2020 trogdor.sh
drwxr-xr-x 2 kafka root 4096 Jul 28 2020 windows
-rwxr-xr-x 1 kafka root  867 Jul 28 2020 zookeeper-security-migration.sh
-rwxr-xr-x 1 kafka root 1393 Jul 28 2020 zookeeper-server-start.sh
-rwxr-xr-x 1 kafka root 1221 Jul 28 2020 zookeeper-server-stop.sh
-rwxr-xr-x 1 kafka root 1019 Jul 28 2020 zookeeper-shell.sh
[kafka@44428c53f642 bin]$
```

рис. 48 Стандартные sh-скрипты Kafka

По умолчанию они расположены по пути:

/opt/applications/kafka/bin

в случае установки kafka в docker-контейнере, необходимо подключиться к контейнеру и перейти в директорию /bin:

```
$ docker exec -it kafka bash
$ cd bin/
```

Е.3.1.1 Основные команды для работы

Вывести список топиков (рис 49):

```
./kafka-topics.sh --list --zookeeper localhost:2181
#или:
./kafka-topics.sh --list --zookeeper $(cat ZooKeeperList.txt)
#Где ZooKeeperList.txt — это список хостов самого зуклипера (zookeeper:port).
```

```
[datamart@dev-dtm-kafka03 bin]$ ./kafka-topics.sh --list --zookeeper localhost:2181
__consumer_offsets
backup_topic_misdmsource80
cancel.rq
metadata.newdata.rq
metadata.rq
mppw.delegate.rq
mppw.delta.rq
mppw.delta.in.rq
mppw.tp
mppw.upload.data.eduejd27.Ymarks
mppw.upload.data.eduejd27.students
mppw.upload.data.eduejd27.test_null_table3
mppw.upload.data.misdmsource.monitoring
mppw.upload.rq
mppw.upload.rs
query.rq
report.rq
statistics.rq
status.event
[datamart@dev-dtm-kafka03 bin]$
```

рис. 49 Вывод списка топиков

Получить информацию о топике (рис 50):

```
./kafka-topics.sh --describe --zookeeper localhost:2181 --topic test_topic
```

```
[datamart@dev-dtm-kafka03 bin]$ ./kafka-topics.sh --describe --zookeeper localhost:2181 --topic status.event
Topic: status.event PartitionCount: 1 ReplicationFactor: 1 Configs:
Topic: status.event Partition: 0 Leader: 1290208 Replicas: 1290208 Isr: 1290208
[datamart@dev-dtm-kafka03 bin]$
```

рис. 50 Вывод информации о топике

Создание топика (рис 51):

```
./kafka-topics.sh --create --zookeeper localhost:2181 --topic test_topic --
replication-factor 1 --partitions 1
```

```
[datamart@dev-dtm-kafka03 bin]$ ./kafka-topics.sh --create --zookeeper localhost:2181 --topic test_topic --replication-factor 1 --partitions 1
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use either, but not both.
Created topic test_topic.
[datamart@dev-dtm-kafka03 bin]$ ./kafka-topics.sh --list --zookeeper localhost:2181
__consumer_offsets
backup_topic_misdm_source00
cancel.rq
metadata.newdata.rq
metadata.rq
mppr.delegate.rq
mppr.delta.rq
mppw.delta.in.rq
mppw.tp
mppw.upload.data.eduejd27.Ymarks
mppw.upload.data.eduejd27.students
mppw.upload.data.eduejd27.test_null_table3
mppw.upload.data.misdm_source.monitoring
mppw.upload.rq
mppw.upload.rs
query.rq
report.rq
statistics.rq
status.event
test_topic
[datamart@dev-dtm-kafka03 bin]$
```

рис. 51 Создание топика test_topic

Удалить топик (рис 52):

```
./kafka-topics.sh --delete --zookeeper localhost:2181 --topic test_topic
```

```
[datamart@dev-dtm-kafka03 bin]$ ./kafka-topics.sh --delete --zookeeper localhost:2181 --topic test_topic
Topic test_topic is marked for deletion.
Note: This will have no impact if delete.topic.enable is not set to true.
[datamart@dev-dtm-kafka03 bin]$ ./kafka-topics.sh --list --zookeeper localhost:2181
__consumer_offsets
backup_topic_misdm_source00
cancel.rq
metadata.newdata.rq
metadata.rq
mppr.delegate.rq
mppr.delta.rq
mppw.delta.in.rq
mppw.tp
mppw.upload.data.eduejd27.Ymarks
mppw.upload.data.eduejd27.students
mppw.upload.data.eduejd27.test_null_table3
mppw.upload.data.misdm_source.monitoring
mppw.upload.rq
mppw.upload.rs
query.rq
report.rq
statistics.rq
status.event
[datamart@dev-dtm-kafka03 bin]$
```

рис. 52 Удаление топика test_topic

Е.3.2 Описание zkCli, входящего в комплект поставки Zookeeper

zkCli - Интерфейс командной строки для взаимодействия с Zookeeper. По умолчанию он расположен по пути:

```
/opt/applications/zookeeper/bin/zkCli.sh
```

в случае установки zookeeper в docker-контейнере, необходимо подключиться к контейнеру и перейти в директорию и выполнить подключение к серверу с помощью zkCli.sh (рис. 53):

```
$ docker exec -it zookeeper bash
$ ./bin/zkCli.sh
```

[illegible]

рис. 53 Успешное подключение к Zookeeper с помощью zkCli

Е.3.2.1 Основные команды для работы

Просмотр узлов в каталоге `ls` (рис. 54):

```
ls /adtm/test
```

```
[zk: localhost:2181(CONNECTED) 0] ls /
adapter, admin, adfs, brokers, cluster, config, consumers, controller, controller_epoch, feature, isr_change_notification, latest_producer_id_block, log_dir_event_notification, test, zookeeper
[zk: localhost:2181(CONNECTED) 1] ls /admin/test/
Path must not end with / character
[zk: localhost:2181(CONNECTED) 2] ls /admin/test
value[2], file_info, information schema, misde_source, testdb
[zk: localhost:2181(CONNECTED) 3]
```

рис. 54 Просмотр узлов в различных каталогах Zookeeper

Получение данных узла по определенному пути **get** (рис. 56):

```
get /adtm/test/testdb/delta
```

```
[zk: localhost:2181(CONNECTED) 5] ls /adtm/test/testdb
[changelog, changelog_erased, delta, entity]
[zk: localhost:2181(CONNECTED) 6] get /adtm/test/testdb/delta
{"hot":null,"ok":{"deltaNum":2,"deltaDate":{"2023,2,10,15,18,37"},"deltaServerDate":{"2023,2,10,15,18,37"},"cnFrom":2,"cnTo":2}}
```

рис. 56 Получение информации о дельте витрины testdb

Изменение данных по пути **set** (рис. 57):

```
set /adtm/test/testdb/delta
```

```
{"hot":null,"ok":{"deltaNum":2,"deltaDate":[2023,2,10,15,18,37],"deltaServerDate":[2023,2,10,15,18,37],"cnFrom":0,"cnTo":2}}
```

```
[*zk: localhost:2181(CONNECTED) 6] get /adtm/test/testdb/delta
{"hot":null,"ok":{"deltaTalm":2,"deltaDate":"2023,2,10,15,18,37","deltaServerDate":"2023,2,10,15,18,37","cnFrom":2,"cnTo":2}}
```

рис. 57 Изменение поля "cnFrom" в дельте витрины testdb

Создание новых узлов **create** (рис.58):

```
create /adtm/test2
```

```
[zk: localhost:2181(CONNECTED) 1] create /adtm/test2
Created /adtm/test2
[zk: localhost:2181(CONNECTED) 2] ls /adtm
[test, test2]
[zk: localhost:2181(CONNECTED) 3] █
```

рис. 58 Создание узла test2 в каталоге /adtm

Удаление узлов по определенному пути **delete** (рис. 59):

```
delete /adtm/test2
```

```
[zk: localhost:2181(CONNECTED) 3] ls /adtm
[test, test2]
[zk: localhost:2181(CONNECTED) 4] delete /adtm/test2
[zk: localhost:2181(CONNECTED) 5]
[zk: localhost:2181(CONNECTED) 5] ls /adtm
[test]
[zk: localhost:2181(CONNECTED) 6] █
```

рис. 59 Удаление узла test2 в каталоге /adtm

Удаление всех узлов по определенному пути **deleteall** (рис. 60):

```
deleteall /adtm/test2
```

```
[zk: localhost:2181(CONNECTED) 10] ls /adtm/test2
[test1, test2, test3]
[zk: localhost:2181(CONNECTED) 11] deleteall /adtm/test2
[zk: localhost:2181(CONNECTED) 12] ls /adtm/test2
Node does not exist: /adtm/test2
[zk: localhost:2181(CONNECTED) 13] █
```

рис. 60 Удаление узла test2 и всех дочерних узлов в каталоге /adtm

Е.3.3 Схема данных для Prostore в Zookeeper

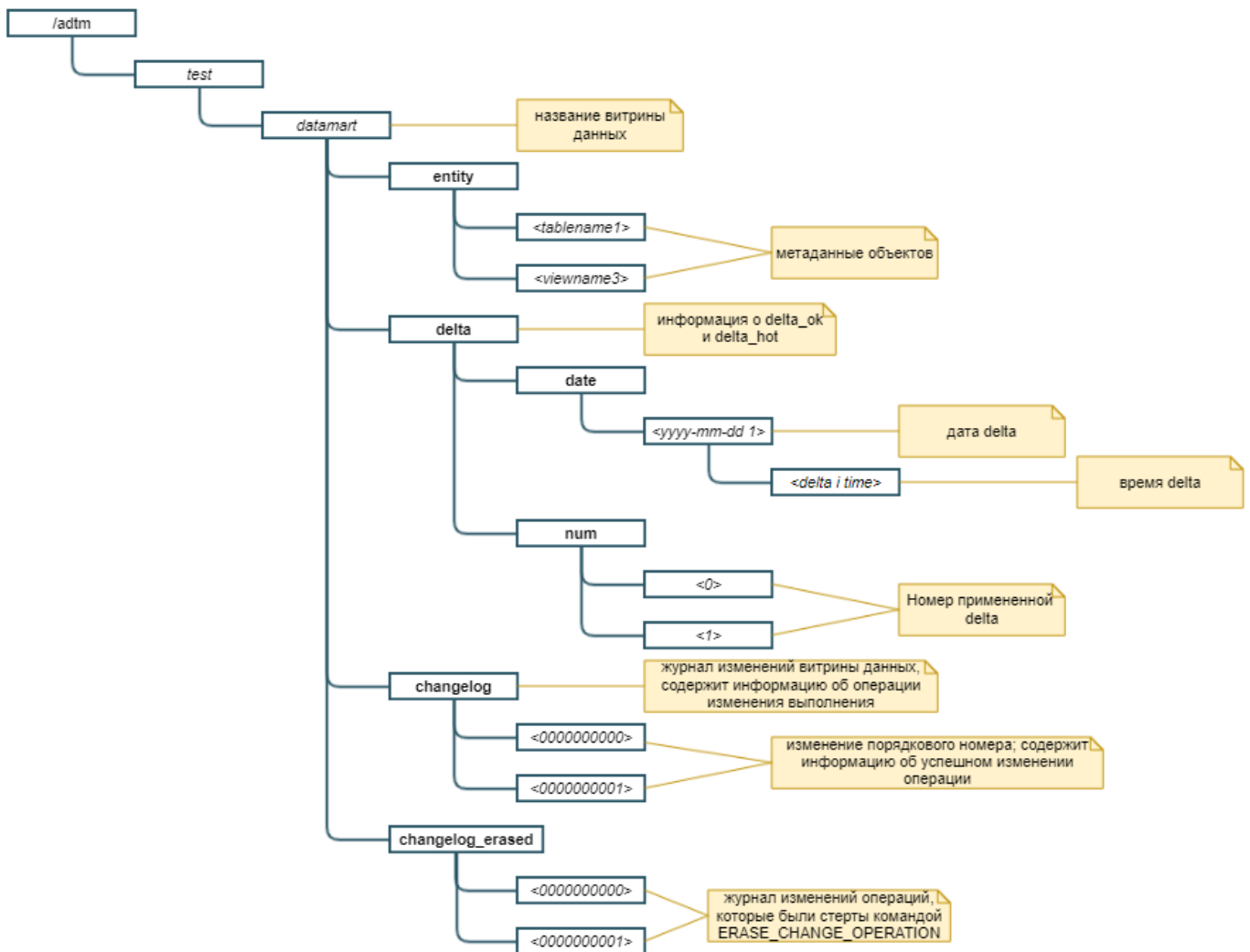


Рис 61. Схема данных для Prostore в Zookeeper

Е.3.4 Схема данных для ПОДД-Адаптеров в Zookeeper

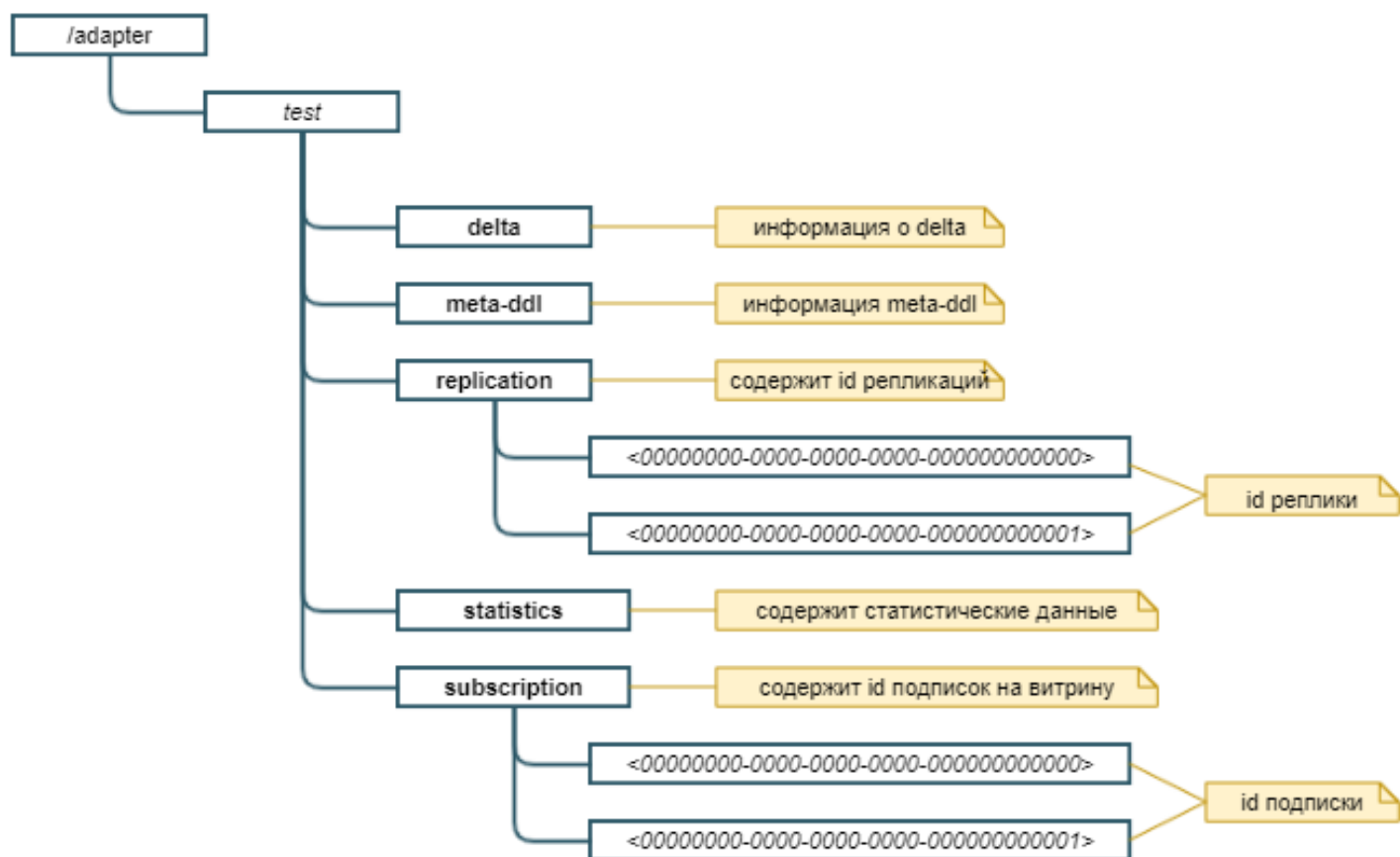


Рис 62. Схема данных для ПОДД-Адаптеров в Zookeeper

Е.3.5 Схема данных для Брокеров в Zookeeper

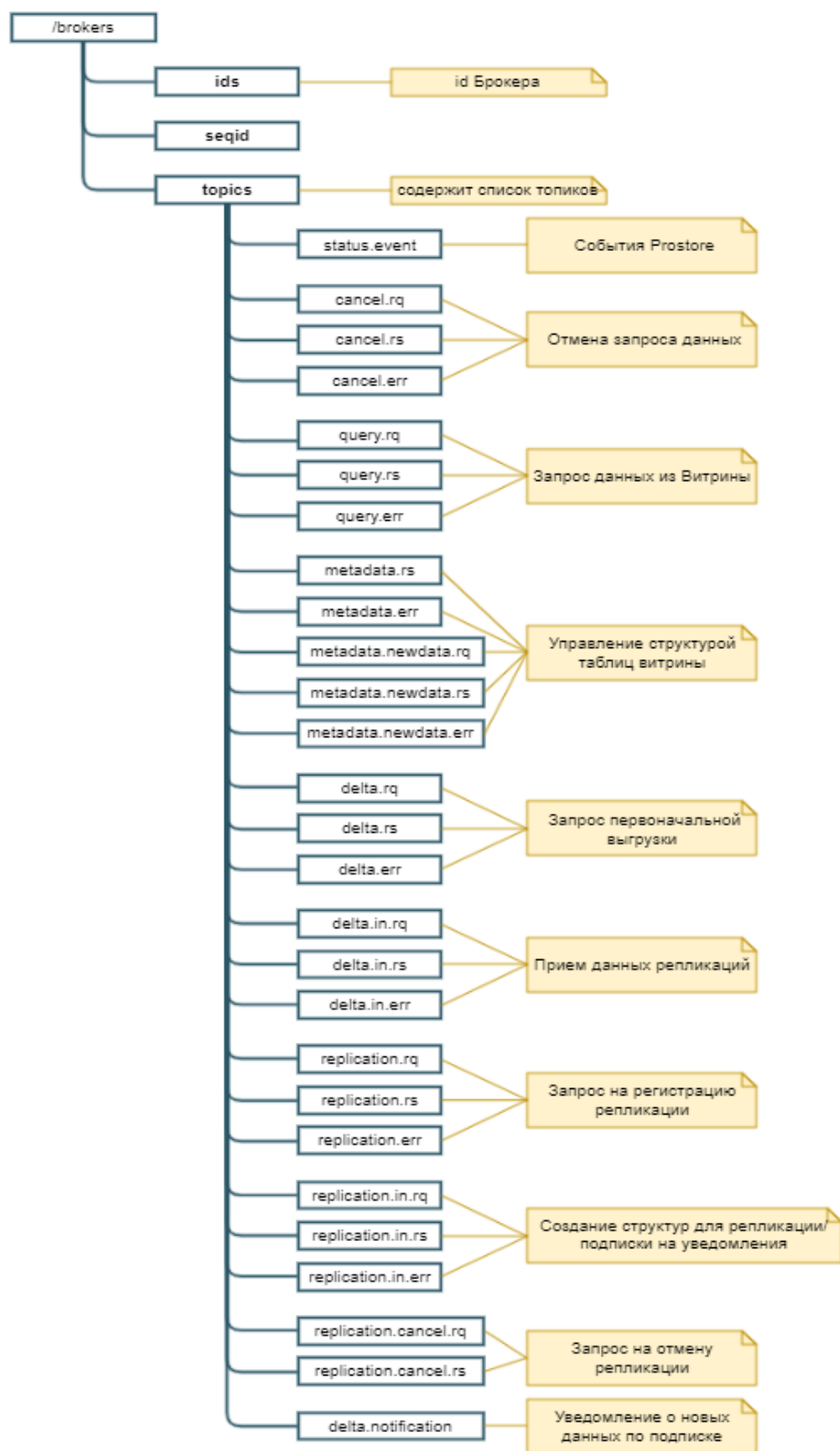


Рис 63. Схема данных для Брокеров в Zookeeper

Е.3.6 Описание топиков Kafka

status.event - содержит сервисную информацию о работе Prostore (получение уведомлений об изменениях логической модели)

cancel.rq - Передача Агентом ПОДД идентификатора запроса, выполнение которого в Витрине необходимо отменить.

cancel.rs - Получение Агентом ПОДД успешной отмены запроса от Витрины.

cancel.err - Получение Агентом ПОДД ошибки при отмене запроса от Витрины.

query.rq - Передача Агентом ПОДД подзапроса данных в Витрину.

query.rs - Получение Агентом ПОДД результата запроса данных от Витрины.

query.err - получение Агентом ПОДД ошибки выполнения запроса данных от Витрины.

metadata.rs - Передача Агенту ПОДД ответа на запрос метаданных витрины из Витрины.

metadata.newdata.rq - Передача Агентом ПОДД запроса на создание таблиц в Витрину.

metadata.newdata.rs - Передача Агенту ПОДД ответа на запрос создания структуры таблиц в витрине из Витрины.

metadata.newdata.err - Передача Агенту ПОДД ошибки при обработке запроса на создания структуры таблиц в витрине из Витрины.

delta.rq - запрос от Агента ПОДД на получение дельты/снапшота изменений по подписке в Витрину.

delta.rs - ответ на запрос на получение снапшота / дельты

delta.err - ответ на запрос на получение снапшота / дельты к Витрине в случае ошибки.

delta.in.rq - запрос на прием дельты / снапшота.

delta.in.rs - ответ на запрос на прием дельты / снапшота.

delta.in.err - Получение Агентом ПОДД ошибки при обработке репликации данных от Витрины.

replication.rq - запрос на регистрацию подписки. Передача Агентом ПОДД запроса (к Витрине-источнику данных) на регистрацию подписки.

replication.rs - ответ на запрос на регистрацию подписки.

replication.err - ответ на запрос на регистрацию подписки. Получение Агентом ПОДД ошибки при обработке подписки на репликацию от Витрины.

replication.in.rq - Передача Агентом ПОДД запроса на создание таблиц для хранения данных подписки на репликацию в Витрину потребителя данных.

replication.in.rs - ответ на запрос на создание структуры данных.

replication.in.err - ответ об ошибке на запрос на создание структуры данных. Негативный ответ на запрос подписки на репликацию - описание причины ошибки, передается только в случае невозможности выполнения запроса (в т.ч. наличие в витрине до запроса таблицы, указанной в запросе).

replication.cancel.rq - запрос на отмену подписки. Передача Агентом ПОДД идентификатора отменяемой подписки на репликацию в Витрину источник данных.

replication.cancel.rs - ответ на запрос на отмену подписки. Получение Агентом ПОДД результата (успешного или ошибки) отмены подписки на репликацию от Витрины.

delta.notification - уведомление об изменении данных на витрине Источнике

Е.3.7 Скрипт для удаления топиков Kafka

Увеличение количества топиков (в случае создания большого количества внешних таблиц без их удаления) может негативно сказываться на работе Kafka и Zookeeper. В этом случае можно воспользоваться скриптом для удаления. Удалять информацию лучше напрямую из Zookeeper.

```
#!/bin/bash

# Останавливаем контейнеры
sudo docker container stop csv-uploader query-execution

touch /tmp/kafka_topic_list.txt
rm /tmp/kafka_topic_list.txt

# Формируем список топиков для удаления
docker exec --tty kafka /opt/kafka/bin/kafka-topics.sh --list --zookeeper
zookeeper:2181 | grep remove* | awk -F " " '{print $1}' > /tmp/kafka_topic_list.txt
# Удаляем символы перевода строки для корректной передачи в контейнер
sed -i -e 's/\r//g' /tmp/kafka_topic_list.txt

while IFS= read -r topic_del
do
    # Удаляем топики по списку
    docker exec --tty zookeeper /opt/zookeeper/bin/zkCli.sh deleteall
/brokers/topics/$topic_del
done < /tmp/kafka_topic_list.txt

# Удаляем список
rm /tmp/kafka_topic_list.txt

# рестартуем кафку и зукипер
sudo docker restart zookeeper
sudo docker restart kafka

# Стартуем контейнеры по порядку
docker container start query-execution
docker container start csv-uploader
```

Е.3.7.1 Пример работы скрипта

В Kafka создаем множество топиков, которые необходимо будет удалить (рис. 64)

```
[kafka@44428c53f642 bin]$ ./kafka-topics.sh --list --zookeeper zookeeper:2181 | grep remove*
remove-me1
remove-me10
remove-me100
remove-me11
remove-me12
remove-me13
remove-me14
remove-me15
remove-me16
remove-me17
remove-me18
remove-me19
remove-me2
remove-me20
remove-me21
remove-me22
remove-me23
remove-me24
remove-me25
remove-me26
remove-me27
remove-me28
remove-me29
remove-me3
remove-me30
remove-me31
remove-me32
remove-me33
remove-me34
remove-me35
remove-me36
remove-me37
remove-me38
remove-me39
```

рис. 64 Топики, которые необходимо удалить

В скрипте задаем условие поиска и запускаем на выполнение (рис. 65)

```
#!/bin/bash
# Останавливаем контейнеры
sudo docker container stop csv-uploader query-execution
touch /tmp/kafka_topic_list.txt
rm /tmp/kafka_topic_list.txt

# Формируем список топиков для удаления
docker exec --tty kafka /opt/kafka/bin/kafka-topics.sh --list --zookeeper zookeeper:2181 | grep remove* | awk -F ' ' '{print $1}' > /tmp/kafka_topic_list.txt
# Удаляем символы перевода строки для корректной передачи в контейнер
sed -i -e 's/\r//g' /tmp/kafka_topic_list.txt

while IFS= read -r topic_del
do
    # Удаляем топик по списку
    docker exec --tty zookeeper /opt/zookeeper/bin/zkCli.sh deleteall /brokers/topics/$topic_del
done < /tmp/kafka_topic_list.txt

# Удаляем список
rm /tmp/kafka_topic_list.txt

# Restartуем kafka и zookeeper
sudo docker restart zookeeper
sudo docker restart kafka

# Startуем контейнеры по порядку
docker container start query-execution
# Проверка для возобновления процесса синхронизации с РМИС
docker container start csv-uploader
```

рис. 65 Условие по которому будет производиться поиск и удаление топиков

Начнется процесс удаления топиков (рис. 66)

```
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --list --zookeeper zookeeper:2181 | grep remove*
remove-me1
remove-me10
remove-me100
remove-me11
remove-me12
remove-me13
remove-me14
remove-me15
remove-me16
remove-me17
remove-me18
remove-me19
remove-me2
remove-me20
remove-me21
remove-me22
remove-me23
remove-me24
remove-me25
remove-me26
remove-me27
remove-me28
remove-me29
remove-me3
remove-me30
remove-me31
remove-me32
remove-me33
remove-me34
remove-me35
remove-me36
remove-me37
remove-me38
remove-me39

kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me1
Deleting topic remove-me1 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me10
Deleting topic remove-me10 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me100
Deleting topic remove-me100 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me11
Deleting topic remove-me11 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me12
Deleting topic remove-me12 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me13
Deleting topic remove-me13 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me14
Deleting topic remove-me14 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me15
Deleting topic remove-me15 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me16
Deleting topic remove-me16 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me17
Deleting topic remove-me17 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me18
Deleting topic remove-me18 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me19
Deleting topic remove-me19 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me2
Deleting topic remove-me2 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me20
Deleting topic remove-me20 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me21
Deleting topic remove-me21 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me22
Deleting topic remove-me22 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me23
Deleting topic remove-me23 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me24
Deleting topic remove-me24 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me25
Deleting topic remove-me25 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me26
Deleting topic remove-me26 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me27
Deleting topic remove-me27 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me28
Deleting topic remove-me28 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me29
Deleting topic remove-me29 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me3
Deleting topic remove-me3 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me30
Deleting topic remove-me30 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me31
Deleting topic remove-me31 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me32
Deleting topic remove-me32 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me33
Deleting topic remove-me33 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me34
Deleting topic remove-me34 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me35
Deleting topic remove-me35 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me36
Deleting topic remove-me36 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me37
Deleting topic remove-me37 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me38
Deleting topic remove-me38 (2 partitions).
kafka@44428c53f642:~/bin$ ./kafka-topics.sh --delete --zookeeper zookeeper:2181 --topic remove-me39
Deleting topic remove-me39 (2 partitions).
```

рис. 66 Процесс удаления топиков

Убедимся, что топики действительно удалены (рис. 67)

```
[datamart@test-vm-2 ~]$ docker exec -it kafka bash
[kafka@44428c53f642 kafka]$ cd bin/
[kafka@44428c53f642 bin]$ ./kafka-topics.sh --list --zookeeper zookeeper:2181 | grep remove*
[kafka@44428c53f642 bin]$
[kafka@44428c53f642 bin]$
[kafka@44428c53f642 bin]$
[kafka@44428c53f642 bin]$
[kafka@44428c53f642 bin]$
[kafka@44428c53f642 bin]$
[kafka@44428c53f642 bin]$
[kafka@44428c53f642 bin]$
[kafka@44428c53f642 bin]$
```

рис. 67 Все топики с именем remove удалены

Е.3.8 Контроль количества транзакционных логов и снэпшотов в Zookeeper

Для снижения нагрузки на Zookeeper и экономии ресурсов необходимо контролировать количество транзакционных логов и снэпшотов. В ручном и автоматическом режиме.

Е.3.8.1 Ручной режим.

```
$ cd /opt/applications/zookeeper/bin
$ sudo ln -s zookeeper.cfg zoo.cfg
$ sudo ./zkCleanup.sh /opt/applications/zookeeper/var/zookeeper -n 10
```

Оставляем только последние 10 файлов.

Е.3.8.2 Автоматический режим.

1. Необходимо добавить в конфиг /opt/applications/zookeeper/conf/zookeeper.cfg строчки:

```
autopurge.snapRetainCount=3
autopurge.purgeInterval=1
```

2. Перезапустить Zookeeper:

```
$ sudo systemctl restart zookeeper
```

Теперь Zookeeper самостоятельно следит за тем, чтобы в директории было не больше 3-х логов и снэпшотов.

Е.4 Дополнительные операции, резервное копирование, логирование, очистка данных, восстановление данных

Е.4.1 Восстановление витрины данных

Инструкция позволяет восстановить витрину и подтянуть в нее актуальные данные в случае возникновения сбоев в ЦОД или отказе некоторых компонентов (напр. Kafka, Zookeeper, Ядро Prostore). Все данные хранятся в <tablename>_actual-таблицах в postgres. Если эти таблицы не повреждены, данные всегда можно восстановить.

ПЛОЖЕНИЕ А Создаем логическую БД с ключевым словом LOGICAL_ONLY

```
CREATE DATABASE testdb LOGICAL_ONLY
```

ДЛОЖЕНИЕ Б Восстанавливаем структуру удаленной БД, путем создания таблиц с ключевым словом LOGICAL_ONLY (чтобы не затереть данные)

```
CREATE TABLE testdb.marks (
    id VARCHAR(200) NOT NULL,
    student_id VARCHAR(200) NOT NULL,
    subject_id VARCHAR NOT NULL,
    period_id VARCHAR(200) NOT NULL,
    class_id VARCHAR NOT NULL,
    mark_value1 VARCHAR NOT NULL,
    mark_value2 VARCHAR,
    weight DOUBLE NOT NULL,
    mark_date DATE NOT NULL,
    subject_name VARCHAR NOT NULL,
    work_type_code VARCHAR NOT NULL,
    work_type_description VARCHAR NOT NULL,
    work_name VARCHAR,
    comment VARCHAR,
    scale_code VARCHAR NOT NULL,
    scale_description VARCHAR NOT NULL,
    school_id VARCHAR(200) NOT NULL,
    work_kind VARCHAR,
    PRIMARY KEY (id)
) distributed by (id) LOGICAL_ONLY
```

DLL для создания таблиц можно получить, выполнив GET_ENTITY_DDL(test.tablename) до удаления логической БД. Если БД удалена, необходимо взять их из postgres (табл. <tablename>_actual), удалив служебные поля:

```
        sys_from int8 NOT NULL,
    sys_to int8 NULL,
    sys_op int4 NULL,
    CONSTRAINT pk_vit02_tablename_actual PRIMARY KEY (sys_from)
    CREATE INDEX tablename_actual_sys_from_idx ON vit02.tablename_actual USING
btree (sys_from);
```

После создания базы и таблиц, необходимо убедиться, что они успешно создались в Prostore (рис. 68)

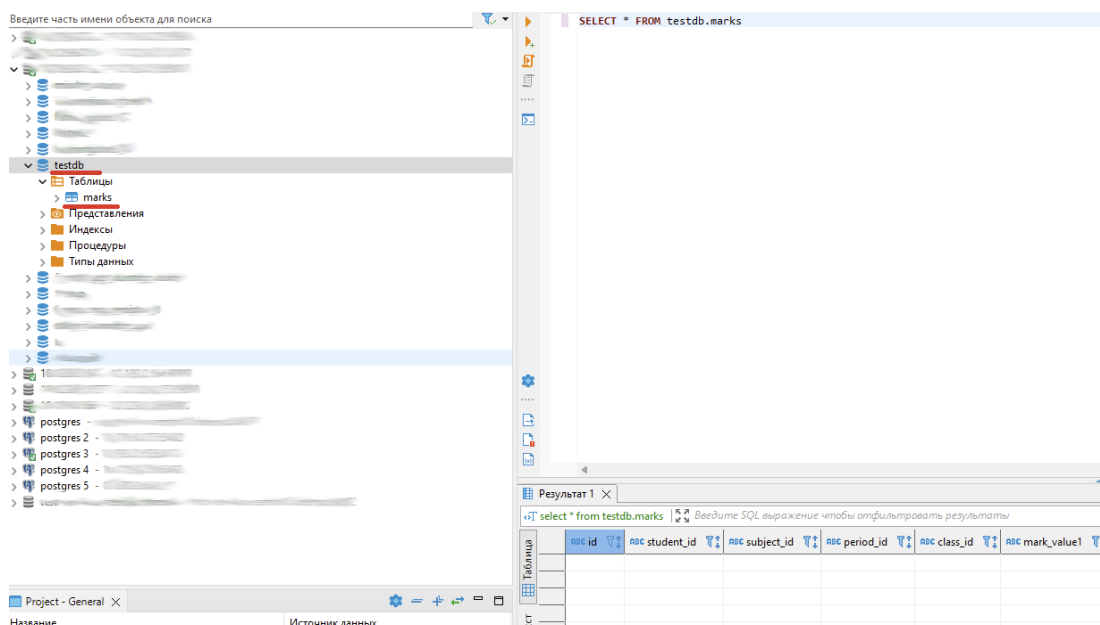


рис. 68 База testdb с таблицей marks создана и не наполнена данными

ПРИЛОЖЕНИЕ В Создаем необходимые Standalone-таблицы (т.к. для них нет версионирования, данные всегда актуальны и автоматически подтянутся в витрину после создания таблиц)

ПРИЛОЖЕНИЕ Г Уточняем наибольшее значение sys_to и sys_from во всех таблицах БД.

Необходимо выполнить запрос `select max(sys_from) sys, max(sys_to) sys_to_max from test.tablename_actual` из DBeaver к postgres, либо подключиться через терминал (рис. 69)

Необходимо выполнить запрос

```
SELECT MAX(sys_from) sys, max(sys_to) sys_to_max FROM test.tablename_actual
```

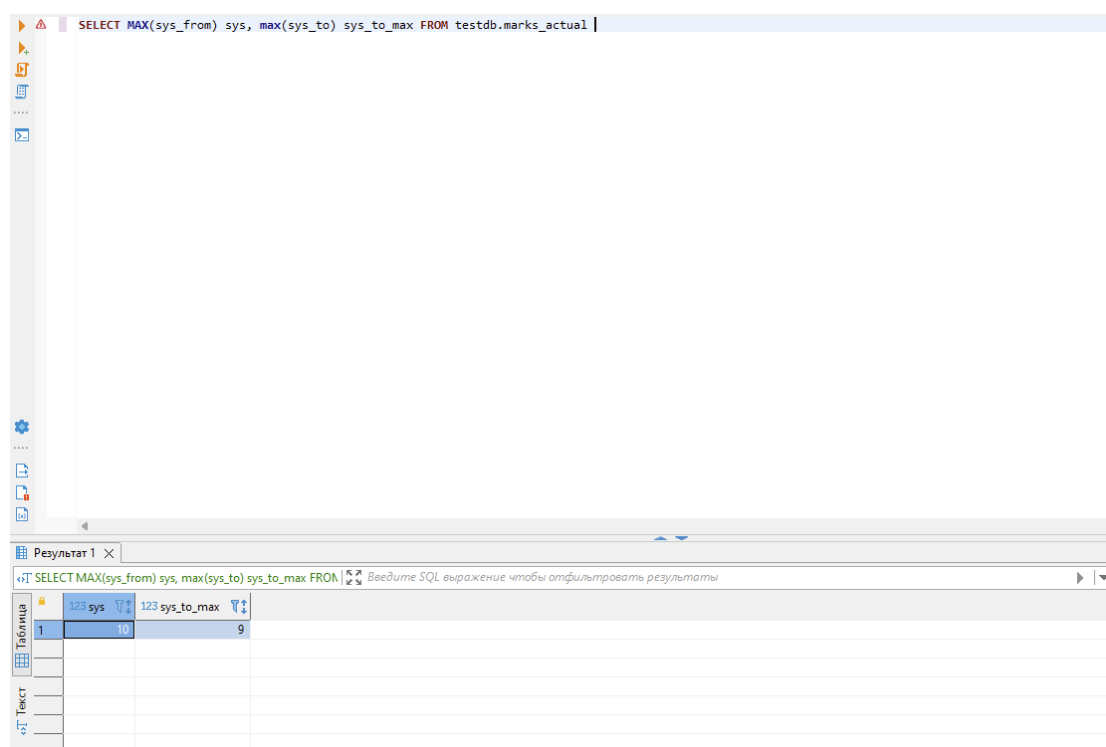


рис. 69 Значения sys_to и sys_from для testdb.marks

ПРИЛОЖЕНИЕ Д Подключиться к Zookeeper с помощью Zoonavigator (если его установка невозможна, необходимо использовать zkCli - см. П.8) и проверить активность дельты по пути - adtm/Название БД postgres/Название БД с данными/delta

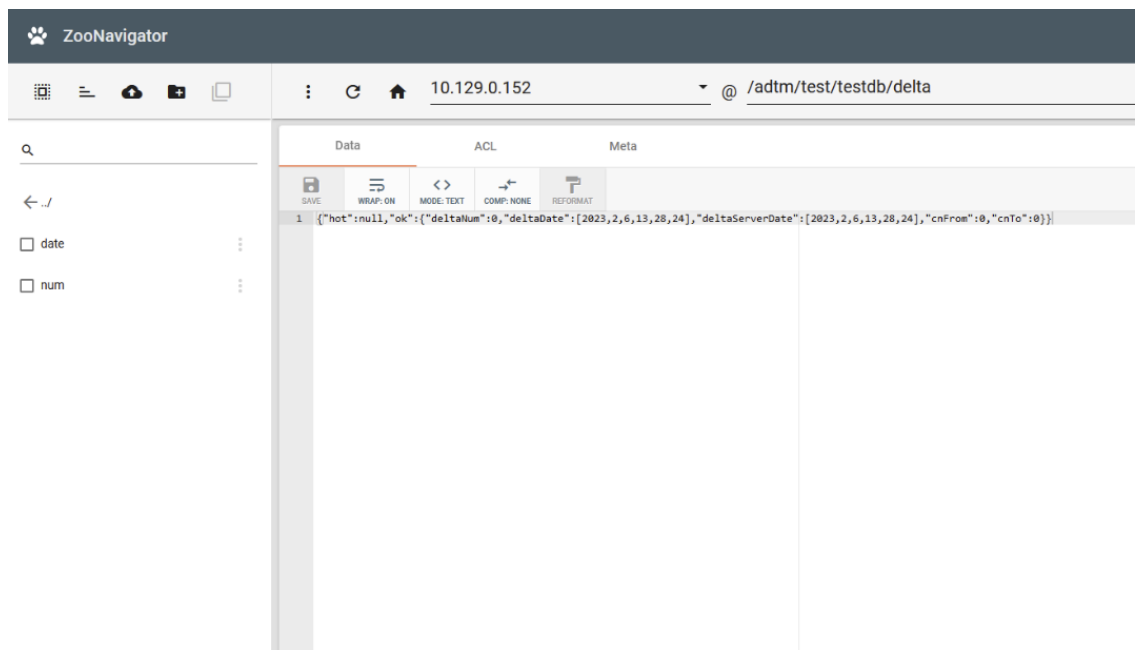


рис. 70 Пример активной дельты

При получении ответа (рис.71)

```
{"hot":null,"ok":null}
```

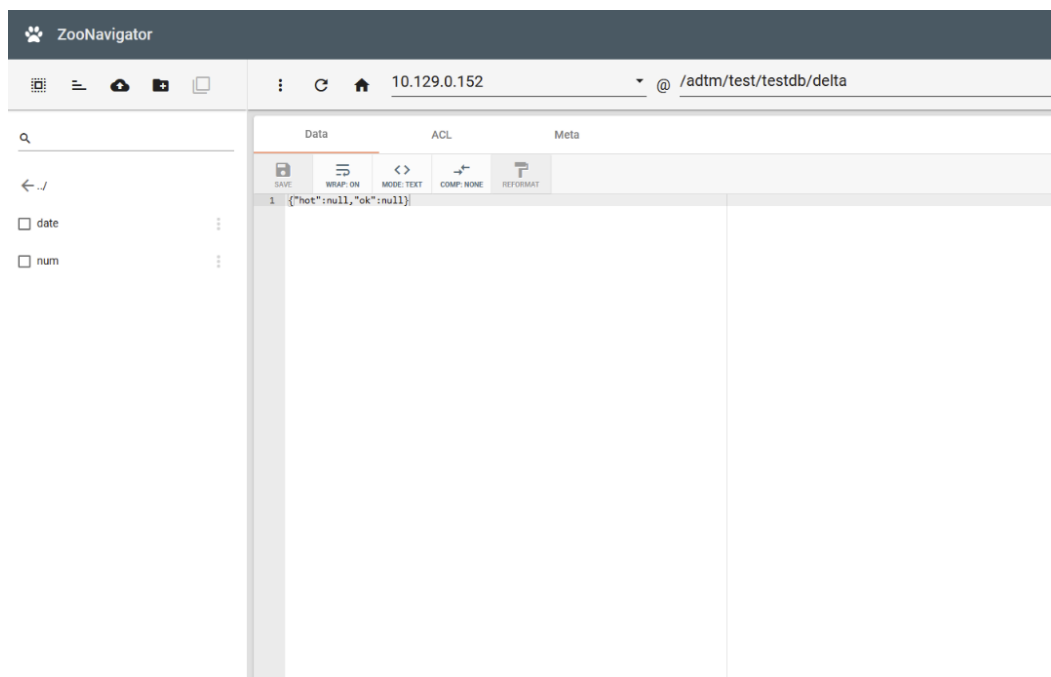


рис. 71 Информация о дельте отсутствует

Необходимо в DBeaver открыть дельту и сразу ее закрыть

```
Use vit02
BEGIN delta;
COMMIT delta;
```

ДЛОЖЕНИЕ Е Установить в Zookeeper наибольшее значение sys_to из П.4 в поле "cnTo" и сохранить изменения (рис. 72)

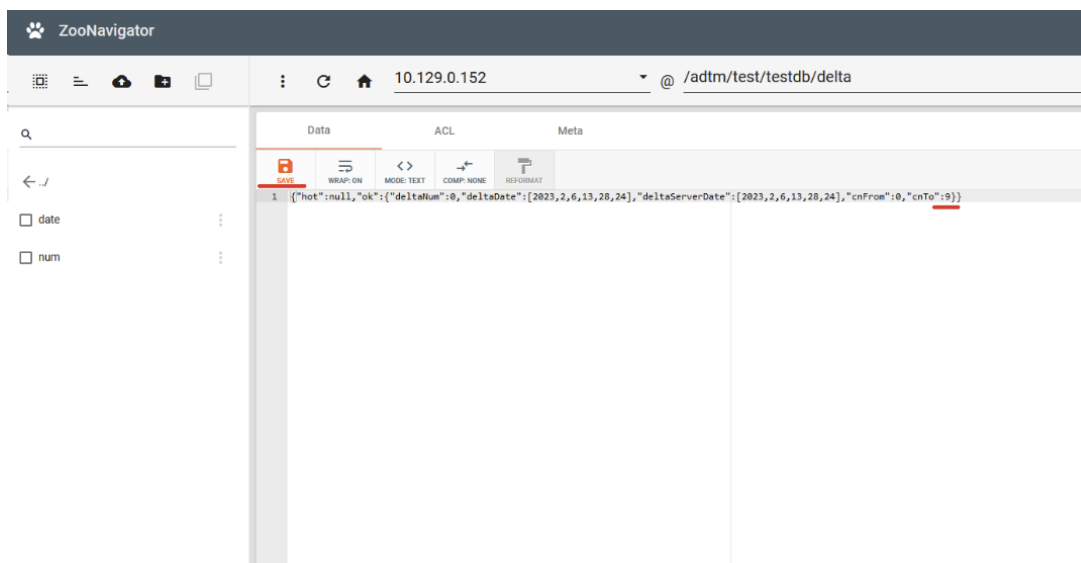


рис. 72 Изменение записи дельты

ДЛОЖЕНИЕ Ж Проверить актуальность данных в витрине (Например посчитать количество записей)(рис. 73)

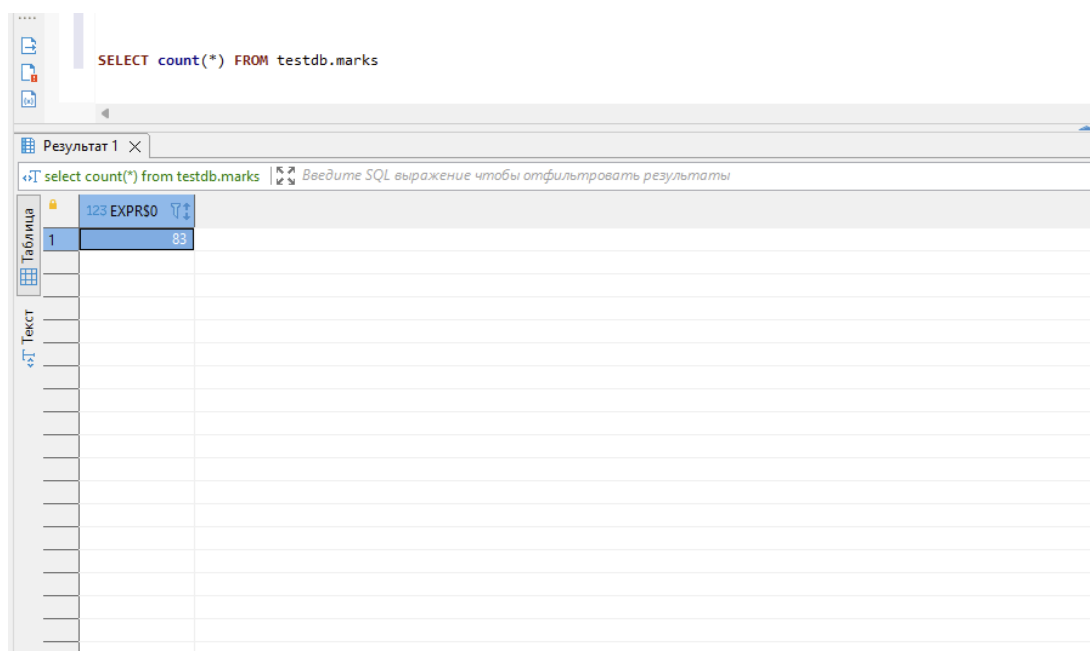


рис. 73 Проверка актуальности данных

ДЛОЖЕНИЕ З Подключение к Zookeeper при помощи zkCli

При отсутствии Zoonavigator можно подключиться к Zookeeper при помощи zkCli

```
docker exec -it zookeeper bash ./bin/zkCli.sh
```

Проверить активность дельты (В Zookeeper папка adtm/Название БД postgres/Название БД с данными/delta)

```
get /adtm/test/test/delta
```

Успешный ответ

```
{"hot":null,"ok":{"deltaNum":0,"deltaDate":[2022,11,21,14,39,11],"deltaServerDate":[2022,11,21,14,39,11],"cnFrom":0,"cnTo":0}}
```

Установить в Zookeeper наибольшее значение `sys_to` из П.4 в поле `"cnTo"`

```
set /adtm/test/vit02/delta
```

```
{"hot":null,"ok":{"deltaNum":0,"deltaDate":[2022,11,21,14,39,11],"deltaServerDate":[2022,11,21,14,39,11],"cnFrom":0,"cnTo":7}}
```

Проверить актуальность данных в витрине (П.7).

Е.4.2 Резервное копирование и восстановление витрины при помощи DTM-Tools

Е.4.2.1 Создание бэкапа логической базы данных

Команда **`backup_to_kafka`** выгружает части бэкапа логической базы данных (далее — инкременты) в топик Kafka. Из целостного набора инкрементов можно развернуть резервную копию логической БД или восстановить сбойную логическую БД с помощью команды **`restore_from_kafka`**.

Каждый инкремент содержит:

- изменения данных, внесенные в дельте и в отдельных операциях записи, которые имеют ту же метку времени, что и дельта;
- журнал, содержащий историю изменений логической схемы данных по состоянию на момент запуска команды.

По логическим таблицам в инкремент включаются как структура, так и данные, по материализованным представлениям — только структура. Материализованные представления автоматически наполняются данными при разворачивании логической БД из бэкапа.

Внешние таблицы не включаются в инкремент.

За один запуск команды **`backup_to_kafka`** можно выгрузить один или несколько инкрементов. Если в команде указан номер дельты, выгружается инкремент этой дельты, иначе — инкременты тех дельт, которых еще нет в топике Kafka.

Е.4.2.1.1 Ограничения

- Запуск команды недоступен, если в исходной логической БД есть незавершенные операции по изменению схемы данных.
- Во время работы команды недоступно изменение схемы данной исходной логической БД.
- Команда не проверяет полноту журнала перед выгрузкой инкрементов. Если журнал отсутствует или заполнен частично, перед запуском команды **`backup_to_kafka`** необходимо выполнить команду **`populate_changelog`**, иначе данные будут неполными и их разворачивание приведет к некорректным результатам.
- Выгружаются данные только тех дельт, которые были закрыты на момент запуска команды. Данные без метки времени, а также данные дельт, закрытых во время работы команды, не выгружаются.

- Инкременты создаются в разрезе закрытых дельт. Это означает, что если на момент запуска команды топик уже содержит инкременты всех закрытых дельт логической БД, команда ничего не выгружает даже при наличии новых изменений данных или новых записей журнала.

Для работы утилиты DTM-Tools необходимо в компоненте DTM-query-execution в параметре ZOOKEEPER_KAFKA_ADDRESS явно указать IP-адрес Кафки. Без указания порта.

ZOOKEEPER_KAFKA_ADDRESS	10.129.0.152
-------------------------	--------------

Для запуска утилиты необходимо, чтобы в операционной системе была установлена Java.

Для работы с Prostore 6.0. необходимо использовать DTM-Tools 1.9.0

Е.4.2.1.2 Синтаксис

```
java -jar dtm-tools-<version>.jar --connect <dtm_host:dtm_port/db_name>
backup_to_kafka [--datasourceType <datasource>] --topic <backup_topic> [--
kafkaBrokers <backup_kafka_broker_list>] [--deltaNum <delta_num>] [--partitionsCount
<partitions_count>] [--threadsCount <threads_count>] [--chunkSize <chunk_size>]
```

Параметры:

- version — номер версии утилиты dtm-tools;
- dtm_host — IP-адрес или имя хоста, на котором установлена система;
- dtm_port — номер порта для подключения к системе (равен значению параметра DTM_CORE_HTTP_PORT в конфигурации системы);
- db_name — имя логической базы данных, для которой создается бэкап;
- datasource — имя датасорса, из которого выгружаются инкременты. Если параметр datasourceType не указан, данные каждой логической таблицы выгружаются из наиболее оптимального датасорса (запрос по выгрузке инкрементов относится к категории «Реляционный» и подкатегории «Все узлы»);
- backup_topic — имя топика, куда выгружаются инкременты;
- backup_kafka_broker_list — список адресов брокеров сообщений Kafka. Если параметр не указан, используется брокер, заданный в конфигурации системы (см. параметр ZOOKEEPER_KAFKA_ADDRESS);
- delta_num — номер дельты, для которой выгружается инкремент. Если параметр не указан, выгружаются инкременты всех «новых» дельт — дельт, чьи номера больше максимального номера дельты среди инкрементов, хранящихся в топике;
- partitions_count — количество партиций, создаваемых в топике backup_topic. Значение по умолчанию — 1. Параметр учитывается при создании нового топика; иначе используются существующие партиции;
- threads_count — количество параллельных потоков, выгружающих данные. Значение по умолчанию — 1;
- chunk_size — количество записей, сохраняемых в одном сообщении Kafka. Значение по умолчанию — 1000.

Е.4.2.2 Пример создания бэкапа

10.129.0.152 - витрина Источник - витрина Лайт 1.8.0 (ProStore 6.0). Все компоненты развернуты на одном сервере.

10.129.0.166 - целевая витрина Приемник - витрина Медиум (ProStore 6.0). Три Kafka и Zookeeper установлены в кластере на выделенном отдельном сервере.

Для создания Бэкапа выполним команду (рис.74):

```
java -jar dtm-tools-1.9.0.jar --connect 10.129.0.152:9090/misdm_source
backup_to_kafka --topic backup_topic_misdm_source --partitionsCount 1 --threadsCount 1
```

```
[datamart@test-vm-2 ~]$ java -jar dtm-tools-1.9.0.jar --connect 10.129.0.152:9090/misdm_source backup_to_kafka --topic backup_topic_misdm_source --partitionsCount 1 --threadsCount 1
SLF4J: class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/datamart/dtm-tools-1.9.0.jar!/BOOT-INF/lib/dtm-jdbc-6.0.0-SNAPSHOT.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/datamart/dtm-tools-1.9.0.jar!/BOOT-INF/lib/logback-classic-1.2.11.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [ch.qos.logback.classic.util.ContextSelectorStaticBinder]
2023-01-31 10:06:06 INFO ru.datamart.prostore.tools.DtmTools - Starting DtmTools v1.9.0 using Java 11.0.17 on test-vm-2.ru-central1.internal with PID 18388 (/home/datamart/dtm-tools-1.9.0.jar)
2023-01-31 10:06:06 INFO ru.datamart.prostore.tools.DtmTools - Running with Spring Boot v2.7.1, Spring v5.3.21
2023-01-31 10:06:06 INFO ru.datamart.prostore.tools.DtmTools - No active profile set, falling back to 1 default profile: "default"
2023-01-31 10:06:08 INFO ru.datamart.prostore.tools.DtmTools - Started DtmTools in 1.878 seconds (JVM running for 2.605)
2023-01-31 10:06:08 INFO r.d.p.t.command.sub.BackupCommand - Started BACKUP process with params:
connect = 10.129.0.152:9090/misdm_source
datasource type = null
topic = backup_topic_misdm_source
kafkaBrokers = null
deltaNum = null
partitionsCount = 1
threadsCount = 1
chunkSize = 1000
2023-01-31 10:06:08 INFO r.d.prostore.tools.dao.ChangesDao - Deny changes in datamart [misdm_source] with code [backup_misdm_source_31]
2023-01-31 10:06:17 INFO r.d.p.t.b.service.BackupService - Delta range [11, 16] to backup
2023-01-31 10:06:17 INFO r.d.prostore.tools.dao.ChangesDao - Got changes for datamart misdm_source
2023-01-31 10:06:17 INFO r.d.p.t.b.service.BackupService - Tables received for datamart misdm_source: [ATTACHMENT, MO, BOOK, MONITORING, PROFILECODE_RESOURCE, SLOT, REFERRAL, PATIENT, SERVICE, PERIOD, RESOURCE]
2023-01-31 10:06:17 INFO r.d.p.t.b.s.task.BackupDeltaTask - [a6ddbc6606e6474685cffe6824fe86ef-backup-delta-11] Started backup delta processing
2023-01-31 10:06:17 INFO r.d.p.t.b.s.task.BackupDeltaTask - [a6ddbc6606e6474685cffe6824fe86ef-backup-delta-11] Got delta date for datamart misdm_source, delta num 11: 1674741390000000
2023-01-31 10:06:17 INFO r.d.p.t.b.s.t.BackupDeltaTableTask - [a6ddbc6606e6474685cffe6824fe86ef-backup-delta-11-table-attachment] Started backup delta table processing
2023-01-31 10:06:17 INFO r.d.p.t.b.s.t.BackupDeltaTableTask - [a6ddbc6606e6474685cffe6824fe86ef-backup-delta-11-table-mo] Started backup delta table processing
2023-01-31 10:06:17 INFO r.d.p.t.b.s.t.BackupDeltaTableTask - [cd507a3eba63464695a2315d31c76cba-backup-delta-12] Started backup delta processing
2023-01-31 10:06:17 INFO r.d.p.t.b.s.task.BackupDeltaTask - [cd507a3eba63464695a2315d31c76cba-backup-delta-12] Got delta date for datamart misdm_source, delta num 12: 1674742803000000
2023-01-31 10:06:17 INFO r.d.p.t.b.s.t.BackupDeltaTableTask - [cd507a3eba63464695a2315d31c76cba-backup-delta-12-table-attachment] Started backup delta table processing
2023-01-31 10:06:17 INFO r.d.p.t.b.s.t.BackupDeltaTableTask - [a6ddbc6606e6474685cffe6824fe86ef-backup-delta-11-table-book] Started backup delta table processing
2023-01-31 10:06:17 INFO r.d.p.t.b.s.task.BackupDeltaTask - [18be93088aee46b4affda109a7451a52-backup-delta-13] Started backup delta processing
2023-01-31 10:06:17 INFO r.d.p.t.kafka.service.KafkaService - Topic backup_misdm_source_attachment_a6ddbc6606e6474685cffe6824fe86ef with 1 partitions successfully created
2023-01-31 10:06:17 INFO r.d.p.t.b.s.task.BackupDeltaTask - [18be93088aee46b4affda109a7451a52-backup-delta-13] Got delta date for datamart misdm_source, delta num 13: 1674743969000000
2023-01-31 10:06:17 INFO r.d.p.t.b.s.t.BackupDeltaTableTask - [18be93088aee46b4affda109a7451a52-backup-delta-13-table-attachment] Started backup delta table processing
2023-01-31 10:06:17 INFO r.d.p.t.kafka.service.KafkaService - Topic backup_misdm_source_no_a6ddbc6606e6474685cffe6824fe86ef with 1 partitions successfully created
2023-01-31 10:06:17 INFO r.d.prostore.tools.dao.DatamartDao - Successfully created download_ext table misdm_source.attachment_ext_download_a6ddbc6606e6474685cffe6824fe86ef
misdm_source.attachment_ext_download_a6ddbc6606e6474685cffe6824fe86ef
(
  MO_ID VARCHAR,
  PATIENT_ID VARCHAR,
  ID VARCHAR,
  CARDS_ID VARCHAR,
  TYPE VARCHAR,
  RESOURCE_ID VARCHAR,
  STATUS VARCHAR,
  CREATE_TS TIMESTAMP(6),
  TYPE_AGE VARCHAR,
  O_OBSERVATION_REASON VARCHAR,
  UPDATE_TS TIMESTAMP(6),
  sys_op int32
)
```

рис. 74 Выполнение команды бэкапирования

Е.4.3 Развертывание логической базы данных из бэкапа

Команда *restore_from_kafka* разворачивает логическую базу данных из бэкапа, сохраненного командой *backup_to_kafka*.

Команда имеет режим *readonly*, в котором проверяет корректность и полноту бэкапа в топике Kafka без его фактического развертывания.

При периодическом дополнении резервной копии новыми данными рекомендуется вносить изменения, загружая бэкап, и не вносить изменения в копию вручную. Иначе нумерация дельт и записей журнала в резервной копии может отличаться от источника.

Как работает команда

Команда последовательно загружает инкременты из топика в логическую базу данных. При этом команда создает недостающие логические сущности и загружает недостающие дельты и записи журнала. Номера и метки времени дельт сохраняются такими же, какими они были в исходной

логической БД. Операции записи, выполненные в исходной логической БД вне дельт, включаются в состав ближайшей дельты, следующей за ними.

С каждой дельтой разворачивается вся схема данных, актуальная на момент создания инкремента. Для логических таблиц из бэкапа загружается структура и данные, для логических и материализованных представлений — только структура. После создания материализованные представления автоматически наполняются данными из развернутых таблиц.

Е.4.3.1 Ограничения

- Команда не проверяет полноту журнала, сохраненного в топике, перед развертыванием логической БД. Если журнал отсутствует или заполнен частично, развертывание логической базы данных приведет к некорректным результатам.
- Во время работы команды не допускаются следующие действия с разворачиваемой логической БД:
 - открытие, закрытие и откат дельт,
 - загрузка и обновление данных,
 - создание, удаление и изменение логических сущностей.
- Во время своей работы команда игнорирует ошибки, не связанные с загрузкой данных из бэкапа, и продолжает развертывание. Например, команда пропускает создание логического представления, если оно связано с другой логической БД, которая отсутствует на момент запуска команды.
- Нумерация операций записи в развернутых дельтах может отличаться от нумерации операций в исходной логической БД.
- Значения `delta_num` в журналах исходной и развернутой логических БД могут отличаться. Это связано с тем, что каждой записи разворачиваемого журнала в поле `delta_num` записывается номер дельты, последней на момент развертывания этой записи, а не на момент ее создания в исходной логической БД.

Для запуска утилиты необходимо, чтобы в операционной системе была установлена Java. Для работы с Prostore 6.0. необходимо использовать DTM-Tools 1.9.0

Е.4.3.2 Синтаксис

```
java -jar dtm-tools-<version>.jar --connect <dtm_host:dtm_port/db_name>
restore_from_kafka --topic <backup_topic> [--kafkaBrokers <backup_kafka_broker_list>]
[--delta <delta_num>] [--partitionsCount <partitions_count>] [--threadsCount
<threads_count>] [--messageLimit <message_limit>] [--readonly [<log_level>]] [--
deltaFrom <delta_from>]
```

Параметры:

- `version` — номер версии утилиты dtm-tools;
- `dtm_host` — IP-адрес или имя хоста, на котором установлена система;

- `dtm_port` — номер порта для подключения к системе (равен значению параметра `DTM_CORE_HTTP_PORT` в конфигурации системы);
- `db_name` — имя разворачиваемой логической базы данных. Имя должно совпадать с именем логической БД, для которой был создан бэкап;
- `backup_topic` — имя топика с бэкапом;
- `backup_kafka_broker_list` — список адресов брокеров сообщений Kafka. Если параметр не указан, используются брокеры, заданные в конфигурации системы (см. параметр `ZOOKEEPER_KAFKA_ADDRESS`);
- `delta_num` — номер дельты, по состоянию на которую разворачивается логическая БД. Если параметр не указан, логическая БД разворачивается по состоянию на максимальную дельту из непрерывного диапазона дельт, инкременты которых есть в топике `backup_topic` на момент запуска команды;
- `partitions_count` — количество партиций, создаваемых в промежуточных топиках, куда копируется бэкап. Значение по умолчанию — 1;
- `threads_count` — количество параллельных потоков, загружающих бэкап в логическую базу данных. Значение по умолчанию равно числу партиций в топике с бэкапом;
- `message_limit` — максимальное количество сообщений Kafka, записываемых в один промежуточный топик. Значение по умолчанию — 1000;
- `log_level` — уровень детализации логов, выводимых в режиме `readonly`. Возможные значения:
 - 0 (по умолчанию) — ключи сообщений Kafka и список некорректных, неполных и отсутствующих дельт в топике с бэкапом,
 - 1 — список некорректных, неполных и отсутствующих дельт в топике с бэкапом;
- `delta_from` — номер дельты, с которой начинает выводиться лог в режиме `readonly`. Значение по умолчанию — 0.

Ключевое слово `readonly`

Ключевое слово `readonly` позволяет вывести метаинформацию бэкапа, который хранится в топике Kafka.

Если ключевое слово указано, в лог командной строки выводится метаинформация с заданной детализацией. Иначе команда загружает данные в логическую БД и не выводит метаинформацию.

Е.4.3.3 Пример развертывания из бэкапа

10.129.0.152 - витрина Источник - витрина Лайт 1.8.0 (ProStore 6.0). Все компоненты развернуты на одном сервере.

10.129.0.166 - целевая витрина Приемник - витрина Медиум (Prostore 6.0). Три Kafka и Zookeeper установлены в кластере на выделенном отдельном сервере.

Для развертывания из бэкапа, сохраненного в топике на другом сервере, необходимо временно переконфигурировать на целевой витрине компоненты DTM-query-execution и DTM-Status-Monitor. Для этого:

1. в DTM-query-execution в параметре `ZOOKEEPER_KAFKA_ADDRESS` укажите ip-адрес Кафки источника

ZOOKEEPER_KAFKA_ADDRESS

сетевой адрес хоста Zookeeper для брокера сообщений Kafka

10.129.0.152 

2. в DTM-Status-Monitor в параметре STATUS_MONITOR_BROKERS укажите ip-адрес Кафки источника

STATUS_MONITOR_BROKERS

список адресов брокеров сообщений
Kafka

10.129.0.152 

Для создания Бэкапа выполним команду (рис 75):

```
java -jar dtm-tools-1.9.0.jar --connect 10.129.0.166:9090/misdmsource restore_from_kafka --topic backup_topic_misdmsource --partitionsCount 1 --threadsCount 1 --messageLimit 2000
```

```
[datamart@test-vm-4 ~]$ java -jar dtm-tools-1.9.0.jar --connect 10.129.0.166:9090/misdmsource restore_from_kafka --topic backup_topic_misdmsource --partitionsCount 1 --threadsCount 1 --messageLimit 2000
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/datamart/dtm-tools-1.9.0.jar!/BOOT-INF/lib/dtm-jdbc-6.0.0-SNAPSHOT.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/datamart/dtm-tools-1.9.0.jar!/BOOT-INF/lib/logback-classic-1.2.11.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [ch.qos.logback.classic.util.ContextSelectorStaticBinder]
2023-01-31 07:34:32 INFO ru.datamart.prostore.tools.DtmTools - Starting DtmTools v1.9.0 using Java 1.8.0_352 on test-vm-4.ru-central1.internal with PID 31017 (/home/datamart/dtm-tools-1.9.0.jar)
datamart in (/home/datamart)
2023-01-31 07:34:32 DEBUG ru.datamart.prostore.tools.DtmTools - Running with Spring Boot v2.7.1, Spring v5.3.21
2023-01-31 07:34:32 INFO ru.datamart.prostore.tools.DtmTools - No active profile set, falling back to 1 default profile: "default"
2023-01-31 07:34:33 INFO ru.datamart.prostore.tools.DtmTools - Started DtmTools in 1.76 seconds (JVM running for 2.423)
2023-01-31 07:34:33 INFO r.d.p.t.command.sub.RestoreCommand - Started RESTORE process with params:
host = 10.129.0.166
port = 9090
datamart = misdmsource
topic = backup_topic_misdmsource
kafkaBrokers = null
deltaNum = null
partitionsCount = 1
threadsCount = 1
messageLimit = 2000
readOnly = false
```

рис. 75 Выполнение команды восстановления

После развертывания из бэкапа, необходимо восстановить корректные значения у параметров ZOOKEEPER_KAFKA_ADDRESS в DTM-query-execution и STATUS_MONITOR_BROKERS в DTM-Status-Monitor.

Е.4.4 Логирование витрины

Е.4.4.1 Логирование приложений в docker-контейнерах

По умолчанию логи приложений, запущенных в контейнерах docker хранятся по пути:

/var/lib/docker/containers/ID контейнера/ID контейнера-json.log

Настройки ротирования логов контейнеров указываются в файле /etc/docker/daemon.json

```
{
  "log-opts": {
    "max-file": "1",      #Количество лог-файлов
    "max-size": "300m"    #Максимальный размер лог-файла
  }
}
```

Е.4.4.2 Логирование Kafka и Zookeeper

По умолчанию логи kafka хранятся по пути:

/opt/applications/kafka/logs

Существует директория kafka-logs где kafka хранит служебную информацию для работы с топиками

В kafka-logs хранится только служебная информация для работы с топиками! Не рекомендуется удалять, изменять или переносить информацию из этой директории во избежание потери информации.

По умолчанию логи Zookeeper хранятся по пути

`/opt/applications/zookeeper/logs`

Е.4.4.3 Логирование Postgres

По умолчанию логи Postgres хранятся по пути:

`/var/lib/pgsql/13/data/log`

Дополнительно параметры логирования можно настроить в файле `postgresql.conf` по пути:

`/var/lib/pgsql/13/data/postgresql.conf`

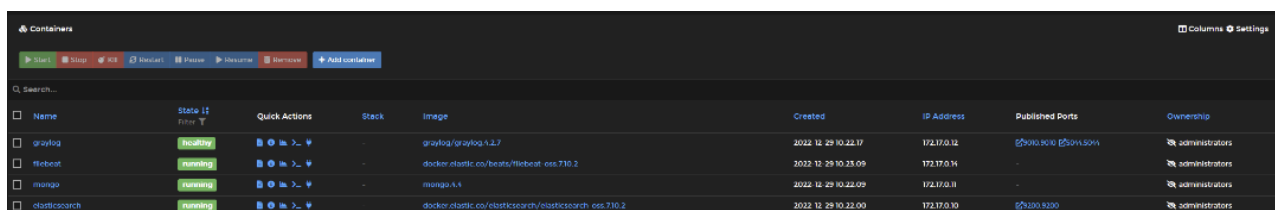
Параметры:

- `log_destination` - выбор метода протоколирования сервера (по умолчанию "stderr");
- `logging_collector` - включение сборщика сообщений (если он отключен, логирование не работает);
- `log_directory` - определяет каталог, в котором создаются журнальные файлы;
- `log_filename` - параметр, задающий имена журнальных файлов;
- `log_file_mode` - параметр задаёт права доступа к журнальным файлам;
- `log_truncate_on_rotation` - включение перезаписи существующих журнальных файлов (перезапись при переключении на новый файл возможна только в результате ротации по времени);
- `log_rotation_age` - определение максимального времени жизни отдельного журнального файла;
- `log_rotation_size` - определение максимального размера отдельного журнального файла;

Е.4.4.4 Логирование Витрины Лайт

Несмотря на то, что Витрина Лайт полностью работает в docker-контейнерах, в комплект поставки входит Graylog для агрегации всех логов в одном месте и удобной работы с ними. Для работы с ним, необходимо запустить контейнеры (рис. 76):

- graylog
- elasticsearch
- mongo
- filebeat



Name	State	Quick Actions	Stack	Image	Created	IP Address	Published Ports	Ownership
graylog	running	[stop] [restart] [logs]	-	graylog/graylog:5.2.7	2022-12-29 10:22:17	172.17.0.12	5555:5555	administrators
filebeat	running	[stop] [restart] [logs]	-	docker.elastic.co/beats/filebeat:oss-7.10.2	2022-12-29 10:23:09	172.17.0.14	-	administrators
mongo	running	[stop] [restart] [logs]	-	mongo:4.4	2022-12-29 10:22:09	172.17.0.11	-	administrators
elasticsearch	running	[stop] [restart] [logs]	-	docker.elastic.co/elasticsearch/elasticsearch:oss-7.10.2	2022-12-29 10:22:00	172.17.0.10	9200:9200	administrators

рис. 76 Контейнеры, необходимы для работы Graylog

После успешного старта контейнера с Graylog, переходим в веб-интерфейс по адресу <http://yourIP:9010>, откроется окно авторизации (рис. 77). Логин для входа - admin, пароль указывается в настройка контейнера в параметре **GRAYLOG_PASSWORD_SECRET**

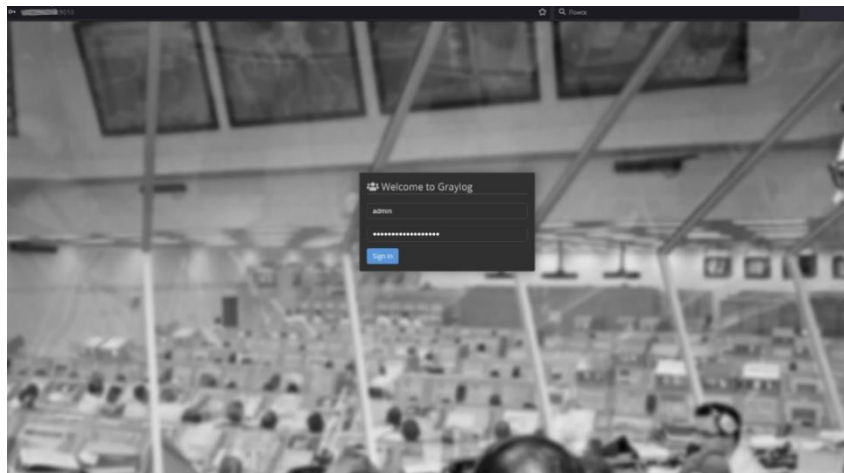


рис. 77 Окно авторизации Graylog

После успешной авторизации откроется стартовая страница Graylog с подсказками по работе в программе, переходим во вкладку "Search" (рис. 78)

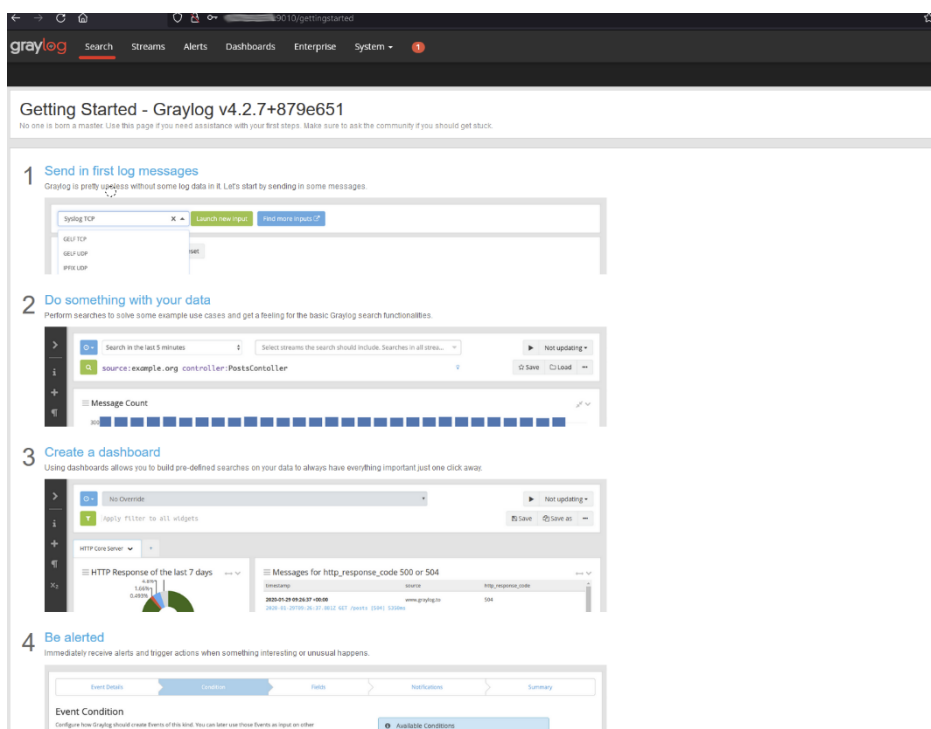


рис. 78 Стартовая страница Graylog

Вкладка Search позволяет просматривать логи в реальном времени или в заданном временном отрезке, осуществлять поиск по ключевым словам, создавать шаблоны поиска, выбирать потоки и выгружать результаты (рис. 79)

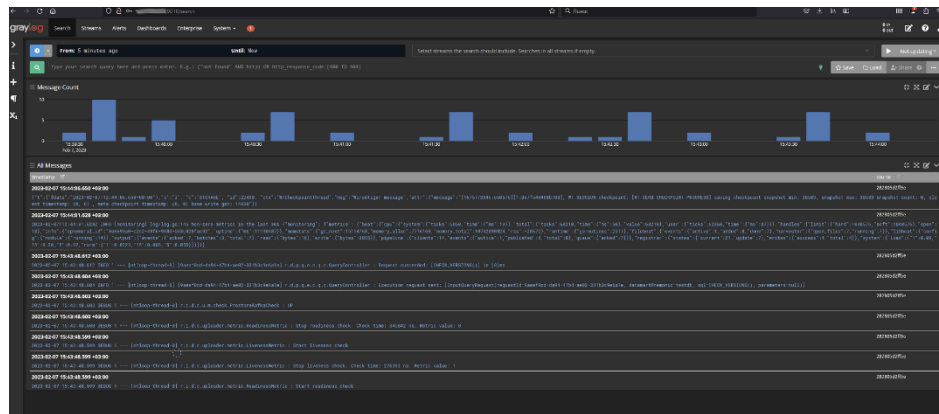


рис. 79 Вкладка Search

Для выбор временного отрезка необходимо нажать на кнопку "Search Time Range" и указать необходимый интервал (рис. 80)

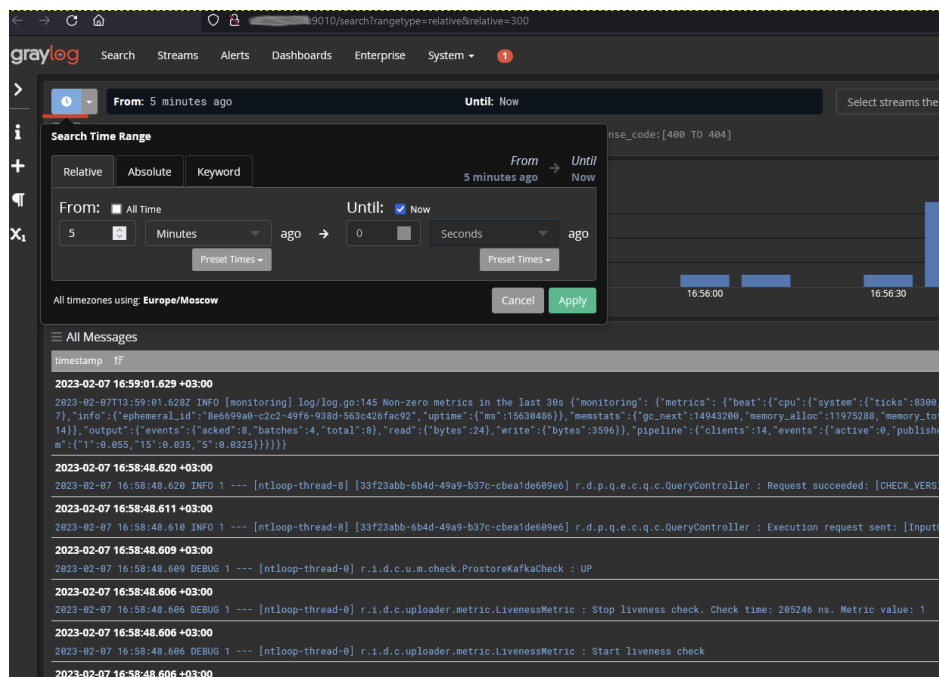


рис. 80 Кнопка Search Time Range

Для поиска по шаблону, необходимо ввести его в поле поиска (рис. 81)

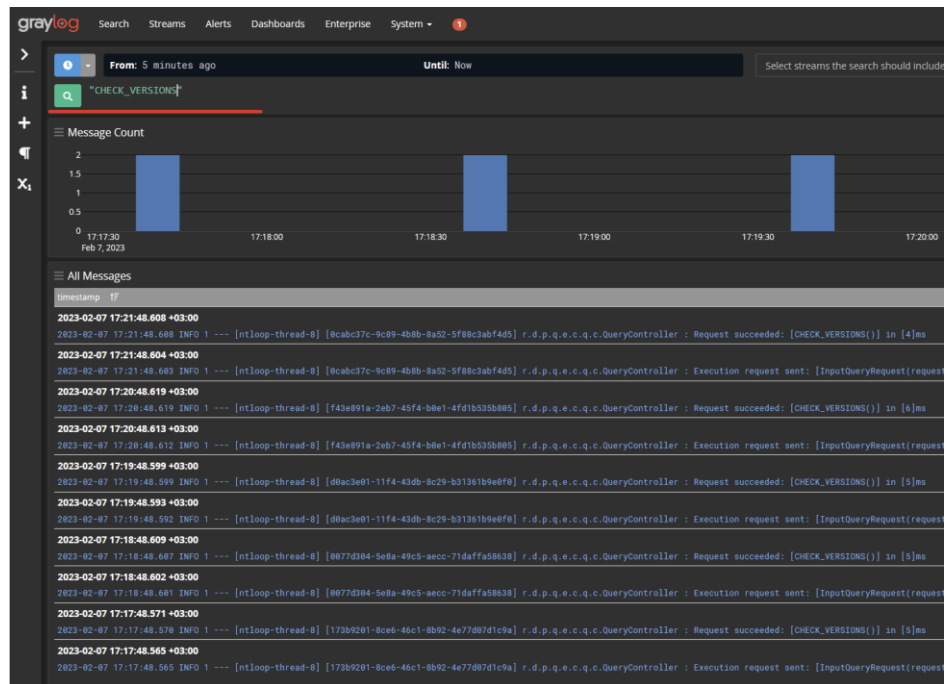


рис. 81 Поиск по шаблону

При необходимости, можно выводить сообщения только из определенных потоков и работать с ними (рис. 82)

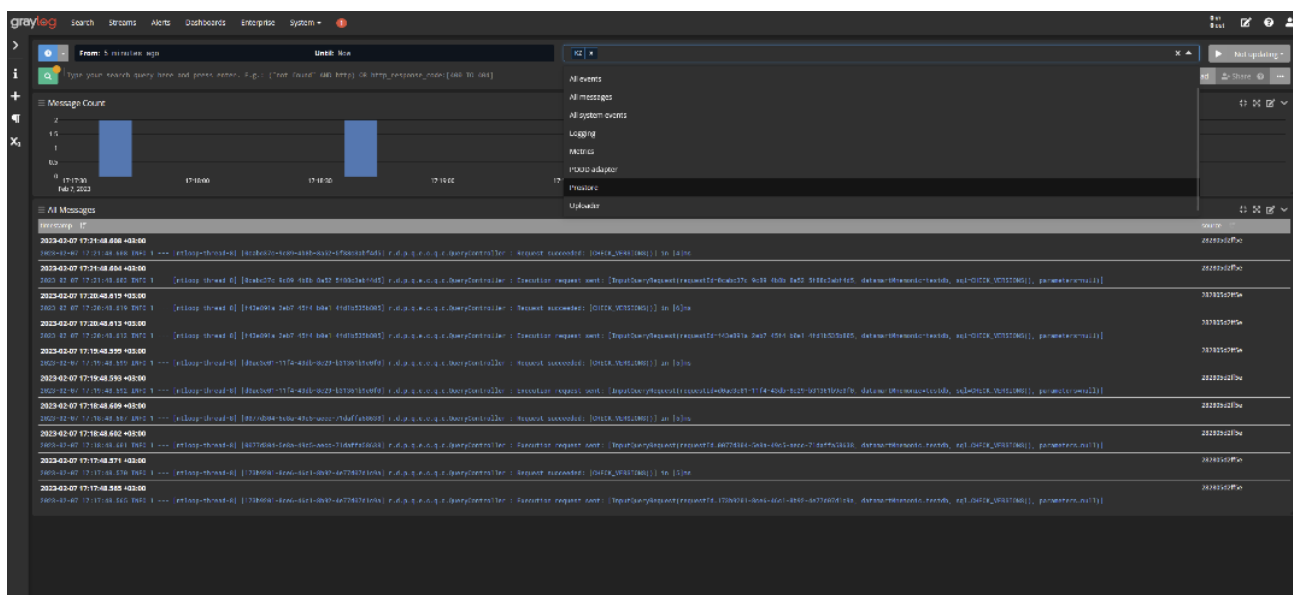


рис. 82 Выбор потоков

Для просмотра логов в реальном времени, необходимо выбрать временной интервал и запустить обновление (рис. 83)

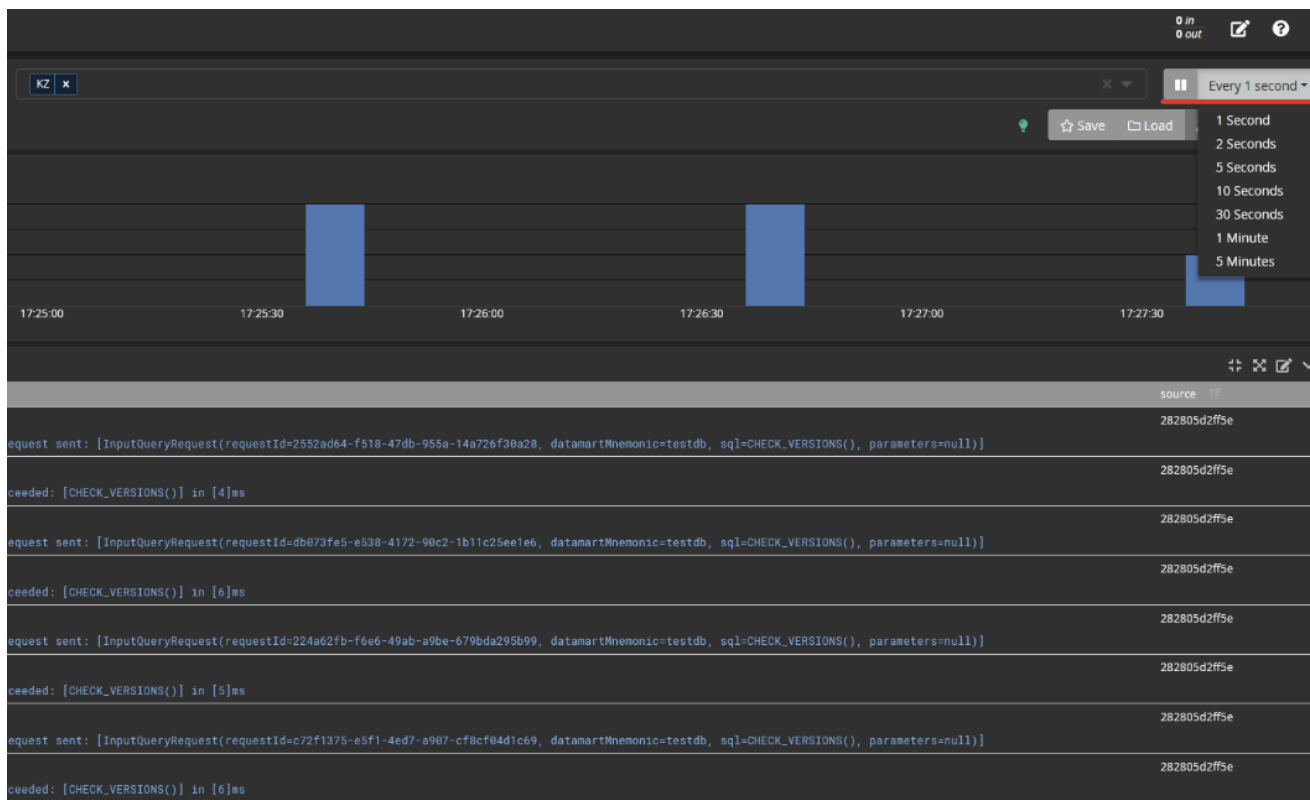


рис. 83 Кнопка просмотра логов в реальном времени

Для выгрузки логов, необходимо нажать на "многоточие" в правом верхнем углу и выбрать "Export" (рис. 84)

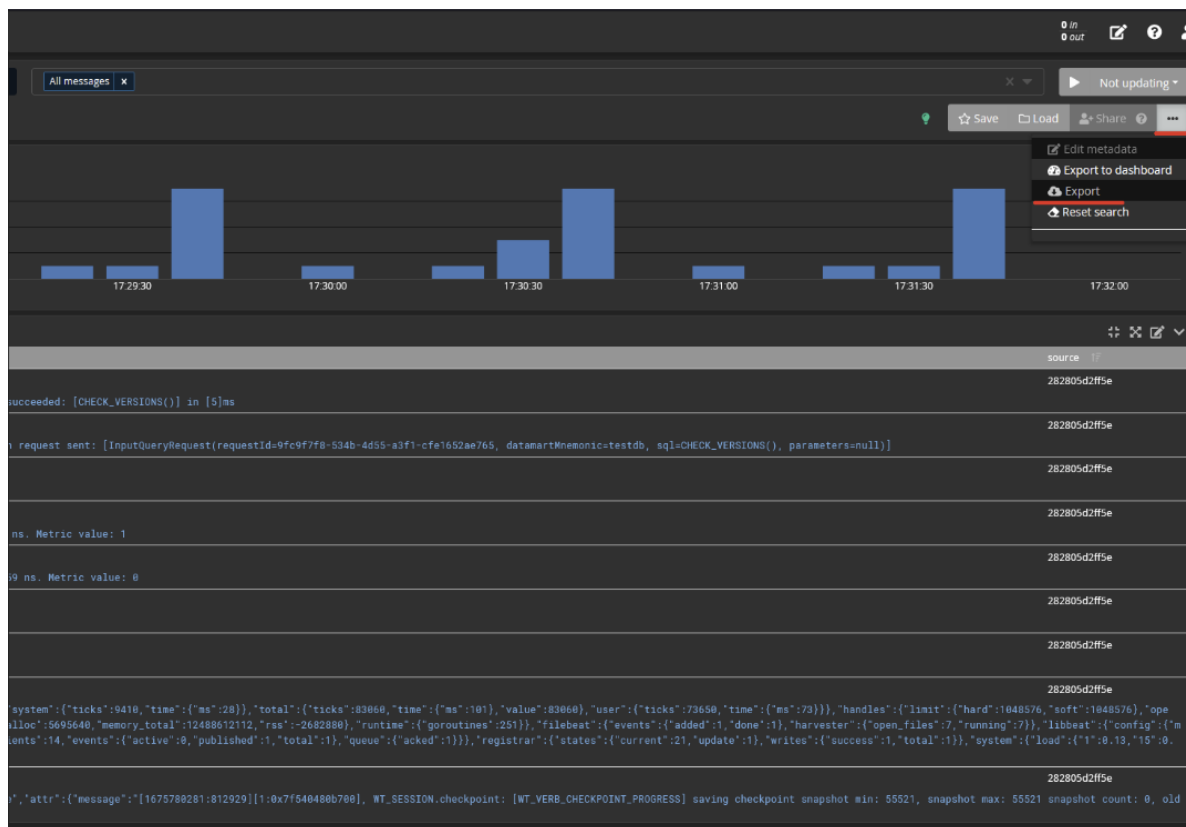


рис.84 Кнопка для выгрузки логов

Вкладка "Streams" показывает потоки из которых собираются сообщения (рис. 85).

Предварительно настроено 10 потоков:

- ADP - Сообщения от Postgres
- All events - Все события созданные в Graylog
- All messages - Все сообщения
- All system events - Все системные события
- KZ - Сообщения Kafka и Zookeeper
- Logging - Сообщения от систем сбора логов (Graylog, Mongo, Elasticsearch, Filebeat)
- Metrics - Сообщения от систем сбора метрик (Grafana, Prometheus, Node Exporter)
- Podd adapter - Сообщения от ПОДД-адаптеров
- Prostore - Сообщения от ядра (query-execution, status-monitor)
- Uploader - Сообщения от CSV-Uploader

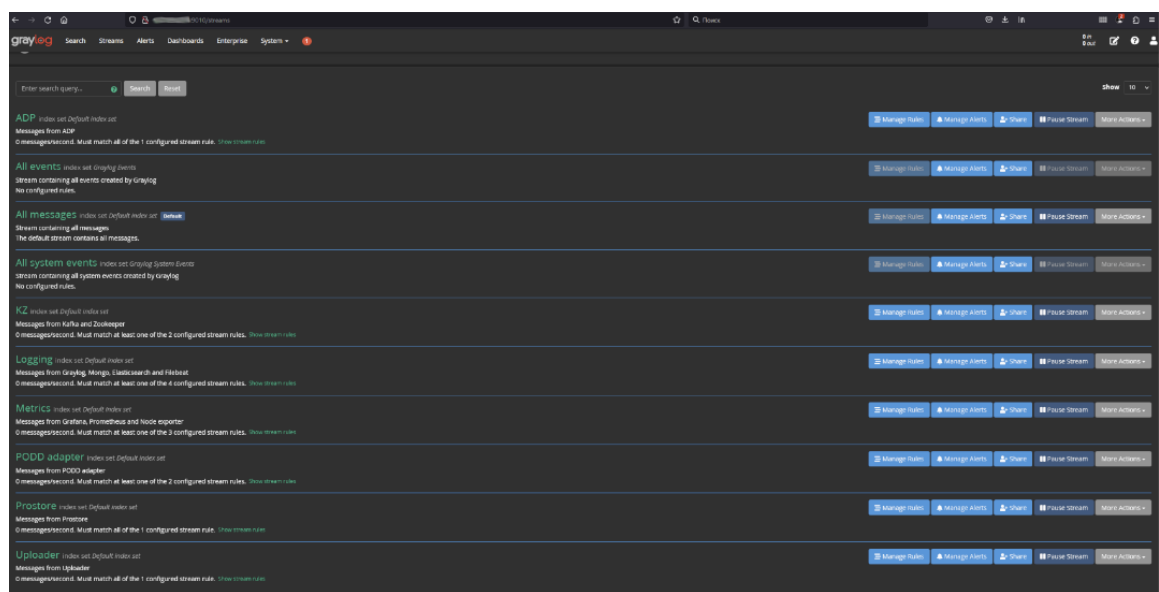


рис. 85 Вкладка Streams

Вкладка "Alerts" позволяет настраивать события через различные условия и создавать оповещения.(рис. 86)

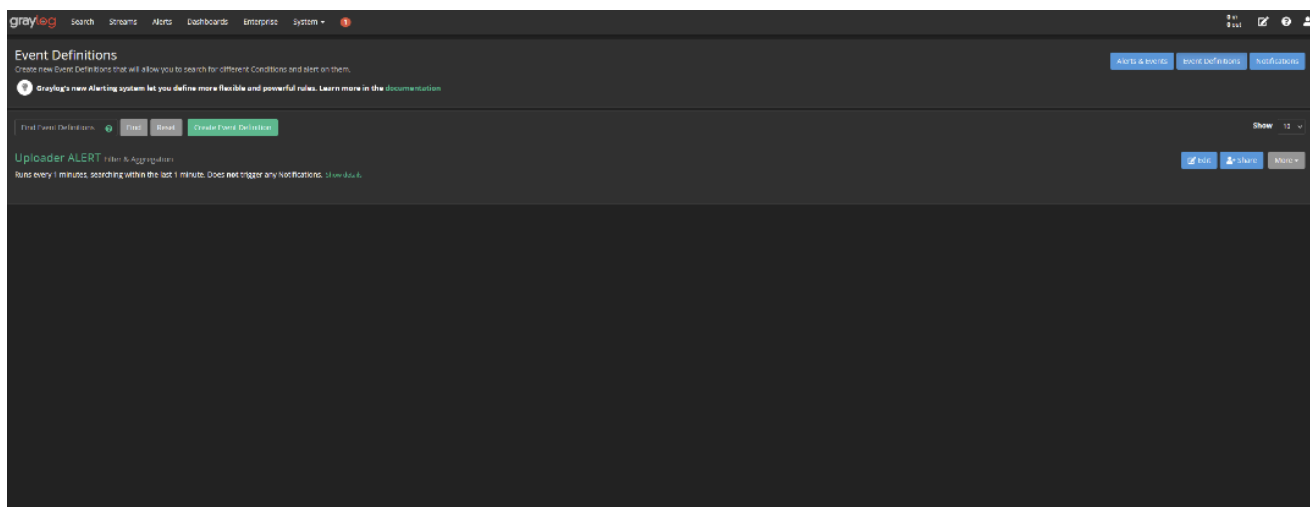


рис. 86 Вкладка Alerts

Вкладка "Dashboards" позволяет создавать дашборды по определенному набору данных. (рис. 87)

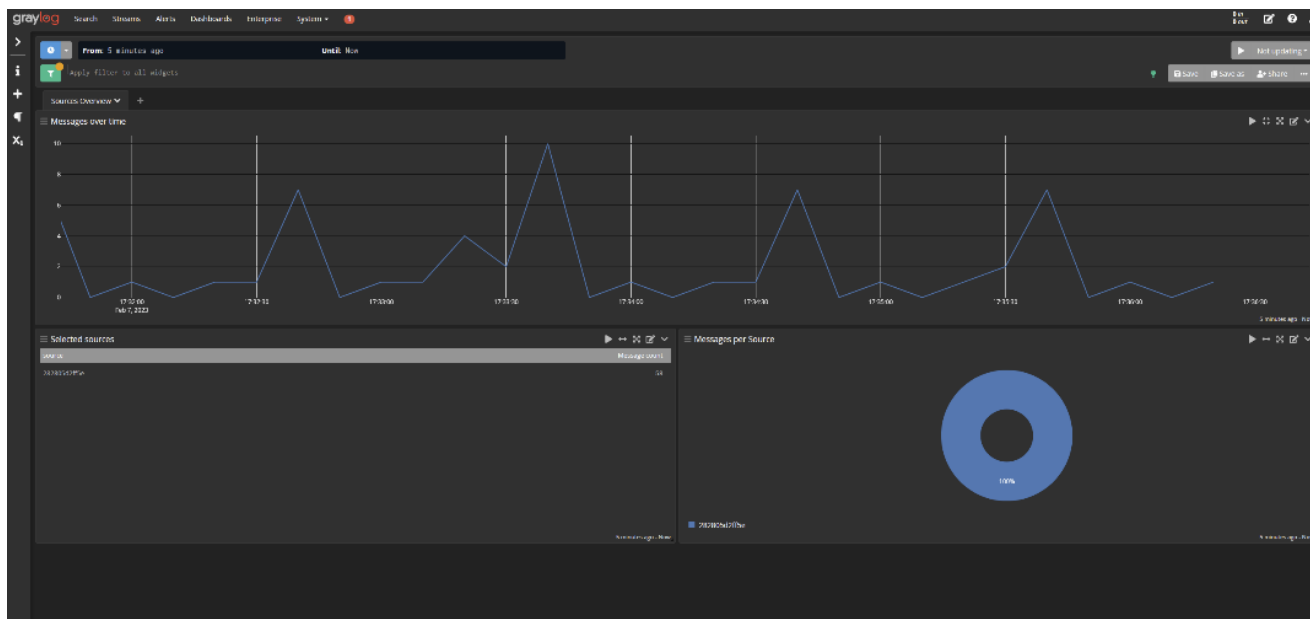


рис. 87 Вкладка Dashboards

Е.4.5 Очистка данных при помощи TRUNCATE HISTORY

В процессы работы с Prostore в СУБД могут накапливаться исторические данные, увеличивающие ее размер. Если нет необходимости в их сохранении, то для сокращения размера бэкапа можно воспользоваться запросом TRUNCATE_HISTORY. Этот запрос удаляет изменения данных, неактуальные на указанный момент времени (при указании метки времени).

Е.4.5.1 Синтаксис

```
TRUNCATE HISTORY [db_name.]table_name FOR SYSTEM_TIME AS OF
date_time_expression[WHERE filter_expression]
```

Е.4.5.2 Пример

В таблице **misd_source.resource** витрины данных 165562 записи, а в этой же таблице в Postgres 331124 (рис.88). Это происходит из-за того, что в СУБД сохраняется историчность и хранятся все изменения записей.

Результат 1 X	Результат 1 X
SELECT count(*) qty_in_prostore FROM misd_source.resource; 1 165562	SELECT count(*) qty_in_postgres FROM misd_source.resource_actual; 1 331124

рис. 88 Количество записей в табл. таблице **misd_source.resource** Витрины данных и СУБД

Выполним запрос TRUNCATE_HISTORY с меткой времени 2023-02-26, чтобы удалить все записи из СУБД старше этой даты и оставить только актуальные (рис. 89):

```
TRUNCATE HISTORY misdm_source.resource FOR SYSTEM_TIME AS OF '2023-02-26 00:00:00';
```

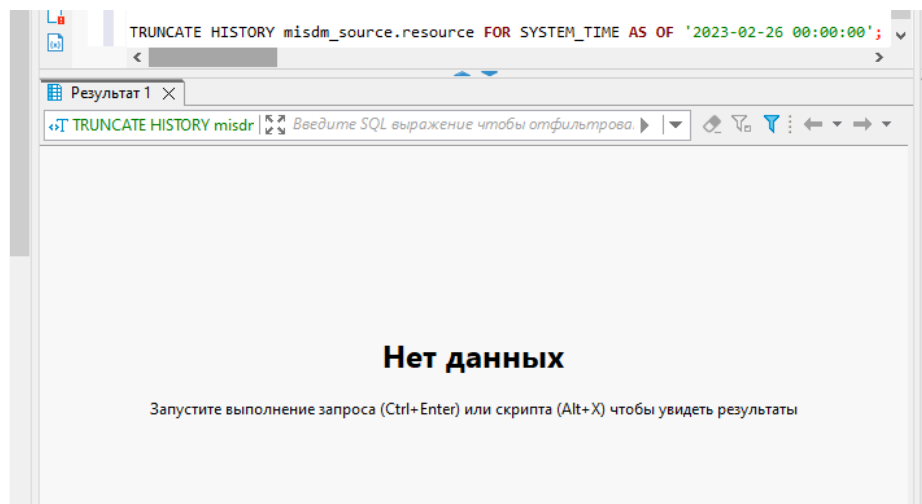


рис. 89 Успешное выполнение запроса TRUNCATE_HISTORY

После выполнения запроса видно, что в СУБД остались только актуальные данные (рис.90)

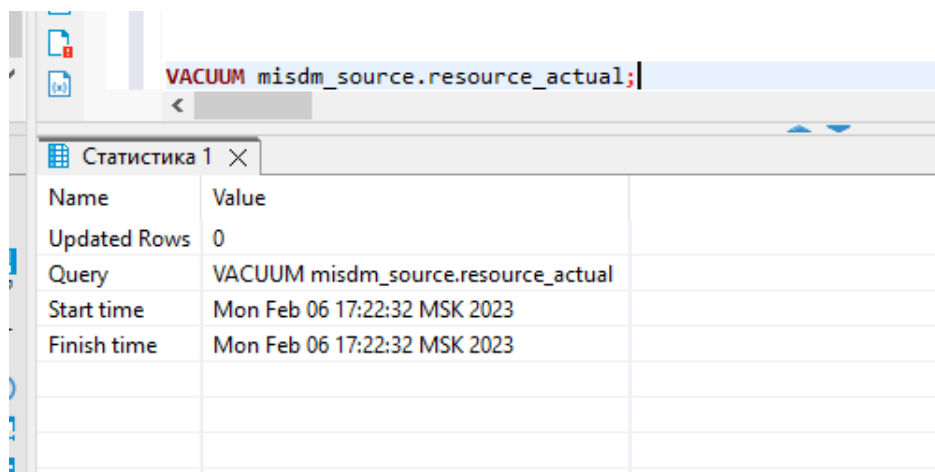
Таблица	1	qty_in_prostore
Текст	1	165 562

Таблица	1	qty_in_postgres
Текст	1	165 562

рис. 90 Количество записей в табл. таблице misd_source.resource Витрины данных и СУБД после выполнения запроса

После выполнения запроса, для полного удаления из СУБД информации о записях, необходимо выполнить VACUUM (подробнее о команде в статье "Настройка и обслуживание PostgreSQL")(рис. 91):

```
VACUUM misdm_source.resource_actual;
```



Name	Value
Updated Rows	0
Query	VACUUM misdsm_source.resource_actual
Start time	Mon Feb 06 17:22:32 MSK 2023
Finish time	Mon Feb 06 17:22:32 MSK 2023

рис. 91 Выполнение команды VACUUM

Е.5 Мониторинг витрины данных

Е.5.1 Базовые настройки Grafana для витрины данных

Из коробки в витрине данных поставляются базовые дашборды для Grafana, позволяющие мониторить работу всех основных компонентов витрины.

Для подключения к UI Grafana необходимо, чтобы контейнеры мониторинга (grafana, prometheus, status-monitor, node_exporter, filebeat, elasticsearch) были запущены.

- порт для UI Grafana по умолчанию 3000
- логин и пароль для UI Grafana по умолчанию admin/admin

Подключившись к UI Grafana доступны два подготовленных дашборда: Lite и Node Exporter Full

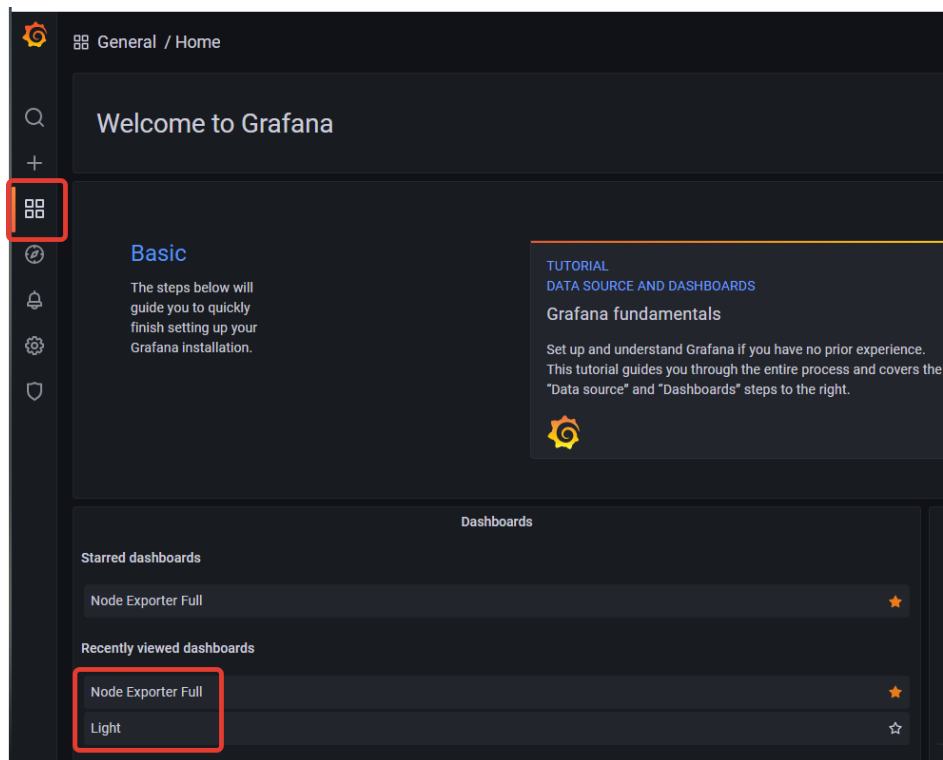


рис. 92 Доступные Дашборды

Е.5.2 Дашборд Lite

Дашборд Lite содержит логические показатели жизнедеятельности витрины.

HealthCheck - содержит две основные метрики, получаемые аплоадерами, показывающие доступность ключевых компонентов витрины: Liveness - состояние вертикалов Vertx и Readiness - доступность Prostore и Kafka

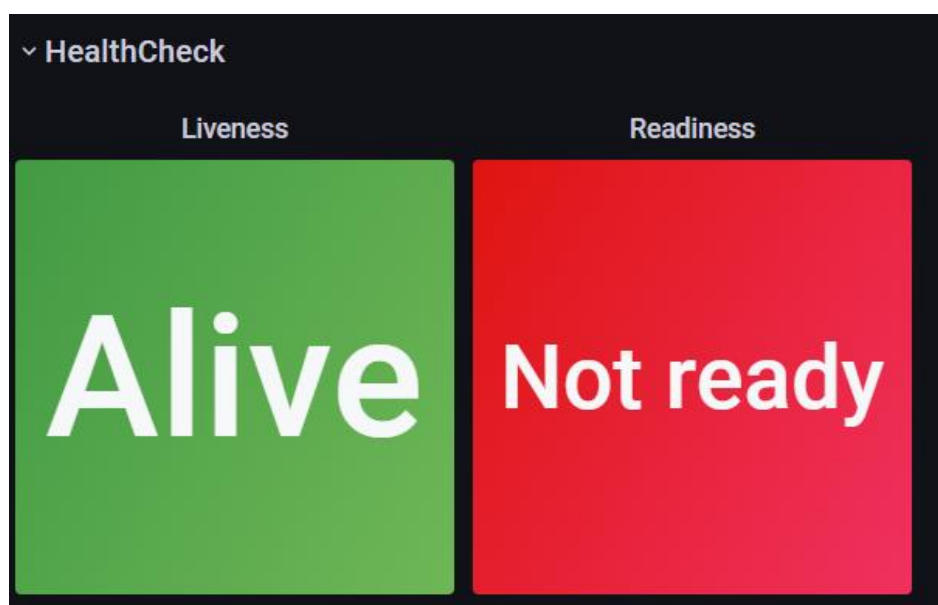


рис. 93 HealthCheck

CSV-uploader - показывает статистику по успешным запросам, статистику по неуспешным запросам и статистику по времени их выполнения

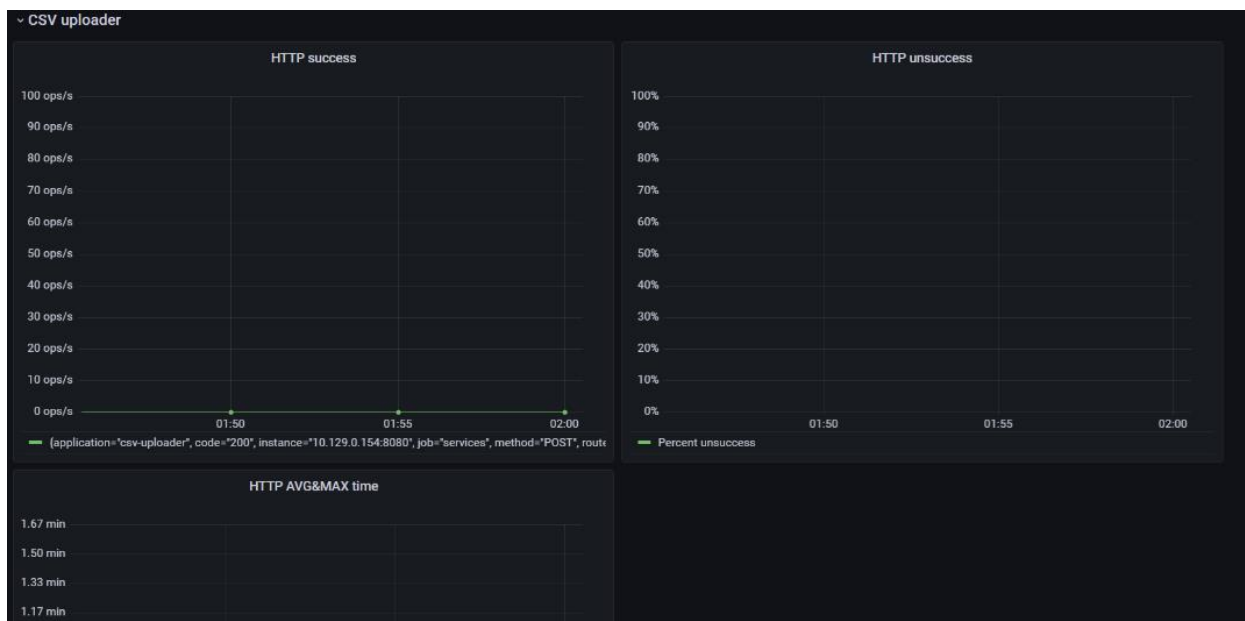


рис. 94 CSV-uploader

PODD-adapter - содержит статистику по скорости обработки очереди подзапросов в витрину данных



рис. 95 PODD-adapter

Е.5.3 Дашборд Node Exporter Full

Дашборд Node Exporter Full собирает множество технических параметров работы витрины: утилизацию процессора, памяти, загруженность и состояние дисков, сетевой трафик и многое другое.

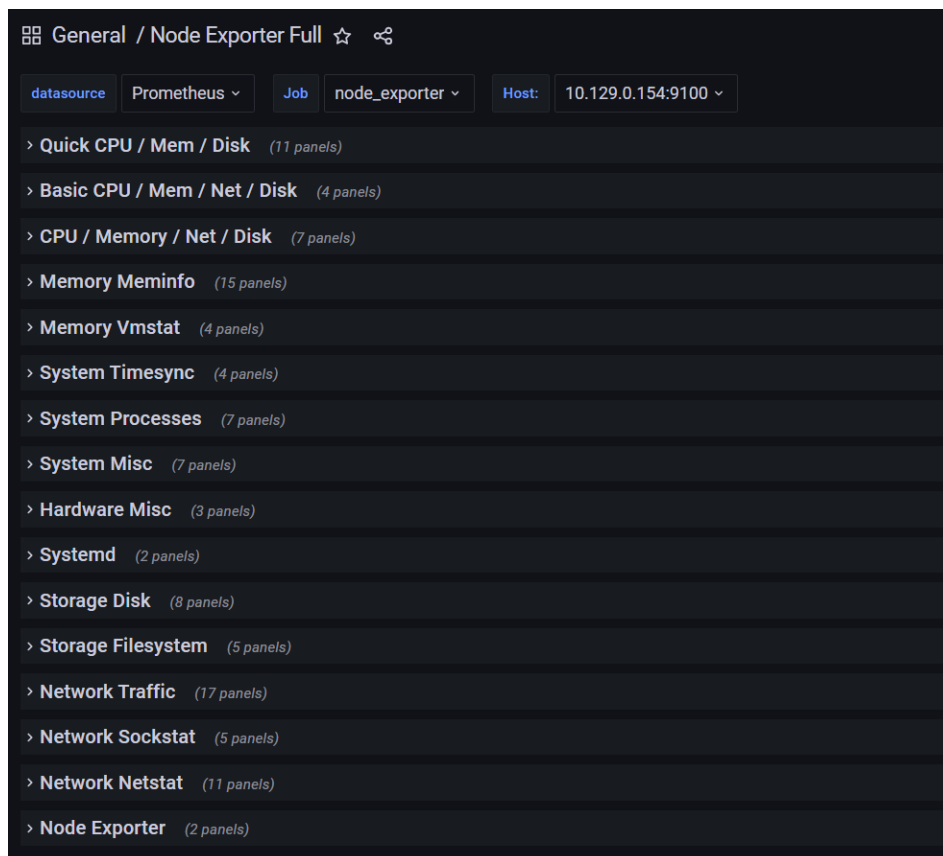


рис. 96 Node Exporter Full

Рассмотрим наиболее важные из них:

Quick CPU / Mem / Disk - общие показатели утилизации процессора и памяти

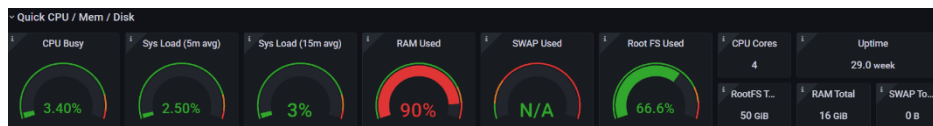


рис. 97 Quick CPU / Mem / Disk

Basic CPU / Mem / Net / Disk, CPU / Memory / Net / Disk - детальный мониторинг утилизации CPU, памяти и диска, а так же сводка по сетевому трафику

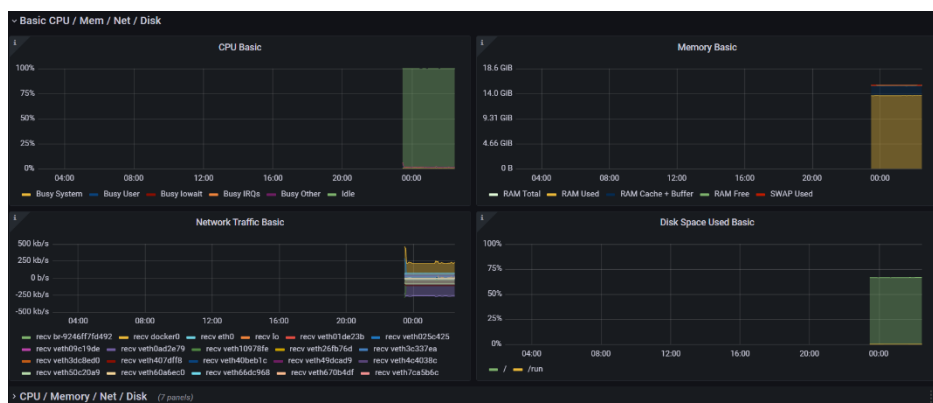
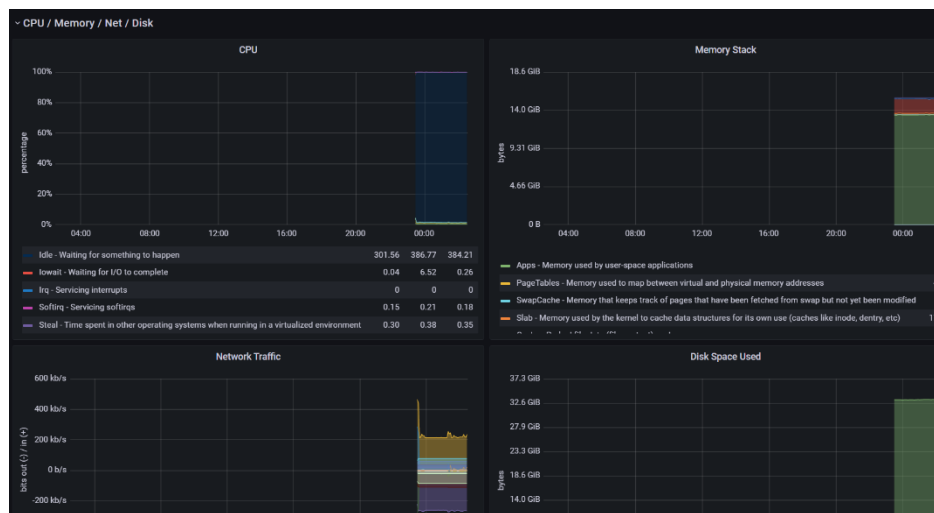
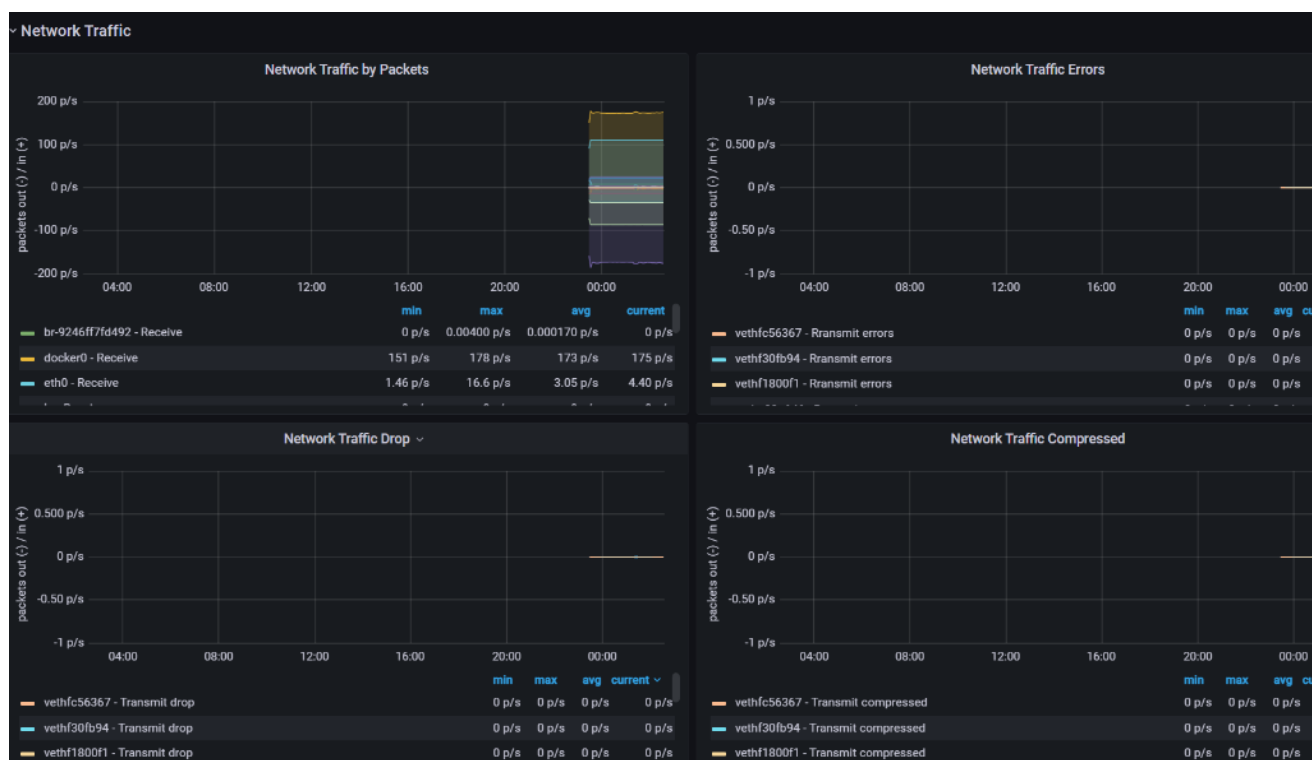


рис. 98 Basic CPU / Mem / Net / Disk



pic. 99 CPU / Memory / Net / Disk

Network Traffic, Network Sockstat, Network Netstat, Node Exporter - детальная статистика по сетевому трафику



pic. 100 Network Traffic

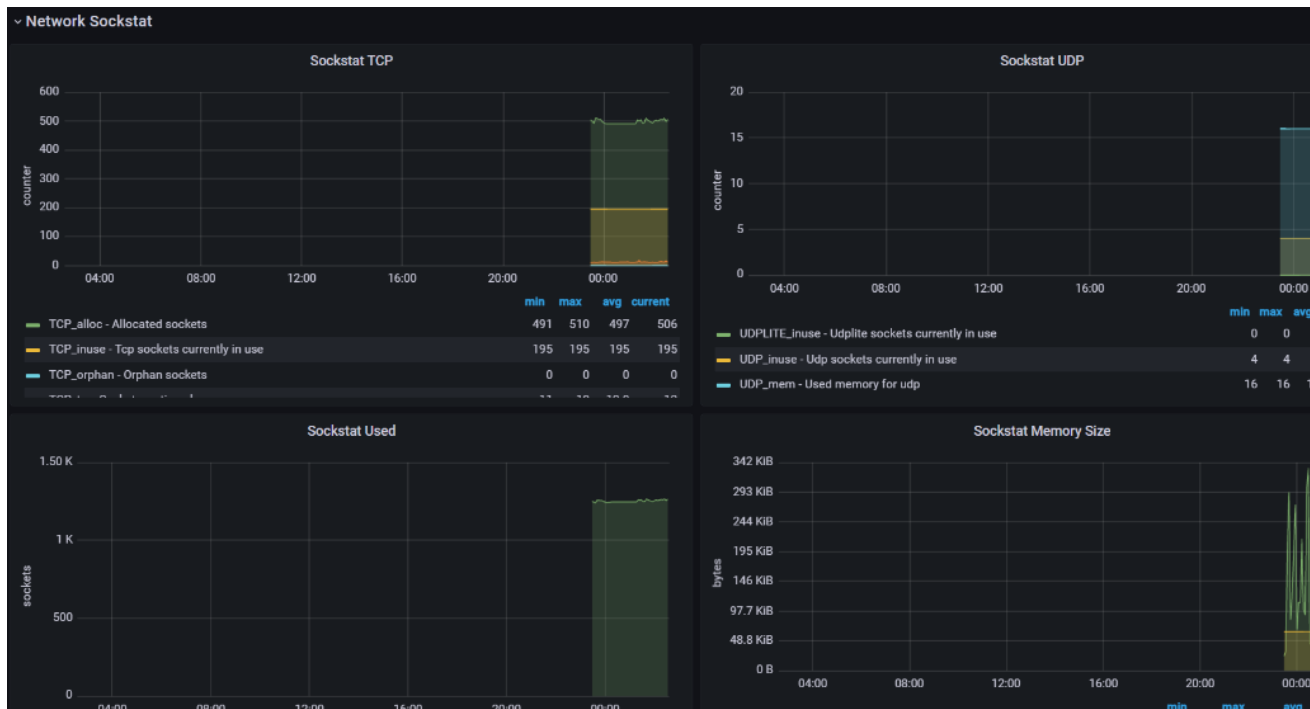


рис. 101 Network Sockstat

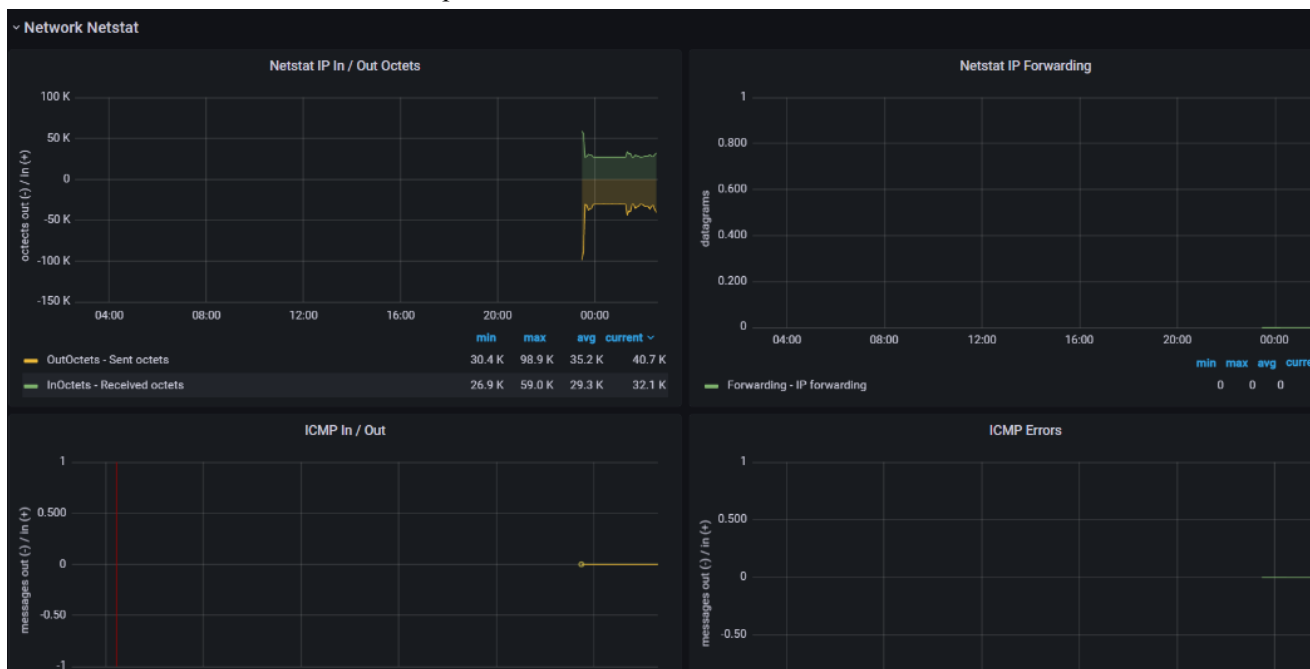


рис. 102 Network Netstat

Grafana позволяет создавать свои дашборды, выбирая нужные метрики из списка доступных в **Prometheus**, или копируя в них необходимые элементы из **Lite** и **Node Exporter Full**

Е.5.4 Загрузка дашбордов

Для **Grafana** существует множество готовых дашбордов для любых целей. Найти их можно на [официальном сайте Grafana](https://grafana.com/)

Для загрузки нужного дашборда необходимо скачать его в **JSON** формате и импортировать в **Grafana**

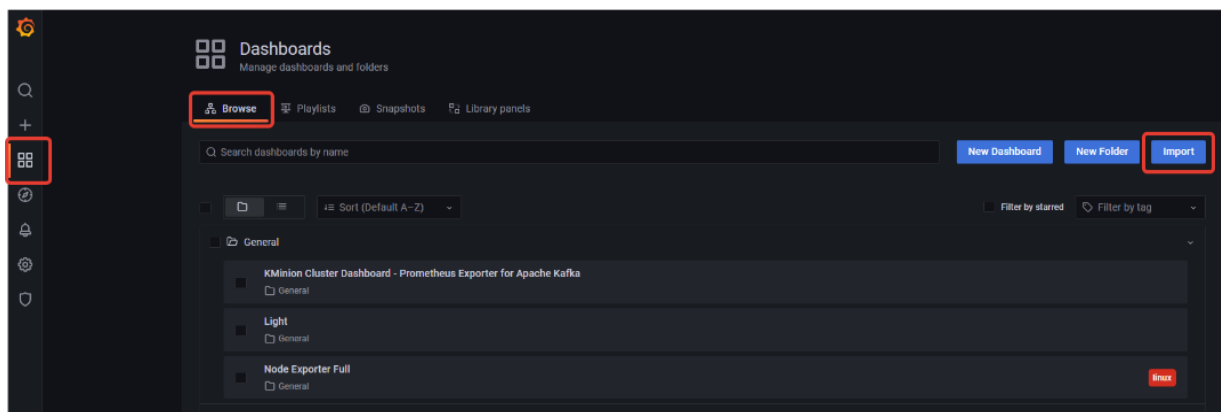


рис. 103 Выбор Дашборда

Во вложении представлены *JSON* файлы дашбордов:

- Light.json
- Node Exporter Full.json
- Kminion-cluster_rev1.json (Prometheus Exporter for Apache Kafka)

ПРИЛОЖЕНИЕ Ж

МЕТОДИКА ОЦЕНКИ КАЧЕСТВА РАБОТЫ УСЛУГИ «ЗАПИСЬ НА ПРИЕМ К ВРАЧУ» С ИСПОЛЬЗОВАНИЕМ ЕДИНОГО ПОРТАЛА ГОСУДАРСТВЕННЫХ И МУНИЦИПАЛЬНЫХ УСЛУГ (ФУНКЦИЙ)

Ж.1 Назначение методики

Настоящая методика предназначена для расчета показателей, характеризующих качество оказания услуги «Запись на прием к врачу» с использованием Единого портала государственных и муниципальных услуг (функций), а также для определения пороговых значений показателя и описания реакции в случае отклонения от порогового значения.

Ж.2 Перечень отчетов и источников данных

Ж.2.1 Перечень отчетов услуги «Запись на прием к врачу (на витринах данных)»

Перечень отчетов, для построения которых источником данных служит Clickhouse, получающий данные из региональной витрины данных и ЕПГУ:

1. Запись к врачу 10700. Воронка. Пользователи
2. Запись к врачу 10700. Воронка. Сессии
3. Сводная. Запись к врачу 10700. Воронка. Пользователи
4. Сводная. Запись к врачу 10700. Воронка. Сессии
5. Отчет по конверсии, источникам записи и недоступности витрины региона

Источником данных для построения отчетов по показателям качества данных в витринах субъектов Российской Федерации, размещаемых в витринах, служит ЕИП НСУД:

Перечень отчетов:

1. Качество данных о прикреплении к МО
2. Доступность врачебных специальностей
3. Доступность слотов, размещаемых на ЕПГУ
4. Конкурентные слоты
5. Прикрепление к МО по полу и возрасту

Ж.2.2 Перечень отчетов услуги «Запись на прием к врачу (КУ ФЭР)»

Перечень отчетов, для построения которых источником данных служит Clickhouse, получающий данные из КУ ФЭР и ЕПГУ:

1. Запись к врачу 600204. Воронка. Пользователи
2. Запись к врачу 600204. Воронка. Сессии
3. Сводная. Запись к врачу 600204. Воронка. Пользователи
4. Сводная. Запись к врачу 600204. Воронка. Сессии

Ж.2.3 Перечень общих отчетов

Перечень отчетов, для построения которых источником данных служит Clickhouse, получающий данные из региональной витрины данных, КУ ФЭР и ЕПГУ:

1. Сравнение форм записи к врачу. Пользователи
2. Сравнение форм записи к врачу. Сессии
3. Отчет по событиям записи на прием к врачу (Пилот запись к врачу 2.0)

Ж.3 Перечень отслеживаемых показателей

Ж.3.1 Перечень показателей услуги «Запись на прием к врачу (на витринах данных)»

Ж.3.1.1 Бизнес-метрики

В таблице Таблица 11 представлен перечень бизнес-метрик услуги «Запись на прием к врачу (на витринах данных)». Показатели рассчитываются на основе данных о событиях, представленных в разделе Ж.4.1. Данные поступают из региональных витрин субъекта РФ и ЕПГУ в БД Clickhouse.

Таблица 11 Бизнес-метрики

Показатель	Отчет
Количество уникальных пользователей	Запись к врачу 10700. Воронка. Пользователи Сравнение форм записи к врачу. Пользователи
Количество уникальных сессий	Запись к врачу 10700. Воронка. Сессии Сравнение форм записи к врачу. Сессии
Количество успешно идентифицированных	Запись к врачу 10700. Воронка. Пользователи
Доля успешно идентифицированных в общем количестве результатов шага	Запись к врачу 10700. Воронка. Сессии
Количество полученных списков специальностей	Сравнение форм записи к врачу. Пользователи
Доля полученных списков специальностей в общем количестве результатов шага	Сравнение форм записи к врачу. Сессии
Количество полученных списков доступных слотов	
Доля полученных списков доступных слотов в общем количестве результатов шага	
Количество успешно записавшихся	

Показатель	Отчет
Доля успешно записавшихся в общем количестве результатов шага	Запись к врачу 10700. Воронка. Пользователи Запись к врачу 10700. Воронка. Сессии
Количество организационных ошибок РМИС	
Доля организационных ошибок РМИС в общем количестве результатов шага	
Количество ошибок ЕПГУ	
Доля ошибок ЕПГУ в общем количестве результатов шага	
Количество ушедших пользователей	
Доля ушедших пользователей в общем количестве результатов шага	
Количество ошибок в разрезе типов ошибок	
Доля ошибок в разрезе типов ошибок в общем количестве ошибок	
Количество ответов на опрос в разрезе типа ответа	
Доля ответов на опрос в разрезе типа ответа в общем количестве ответов	
Длительность недоступности услуги в регионе	Отчет по конверсии, источникам записи и недоступности витрины региона
Доля времени недоступности услуги к общему времени исследуемого периода	Отчет по конверсии, источникам записи и недоступности витрины региона
Изменение недоступности витрины региона по отношению к предыдущему периоду	Отчет по конверсии, источникам записи и недоступности витрины региона
Целевая конверсия	Отчет по конверсии, источникам записи и недоступности витрины региона
Отклонение конверсии от целевой	Отчет по конверсии, источникам записи и недоступности витрины региона
Изменение конверсии по отношению к предыдущему периоду	Отчет по конверсии, источникам записи и недоступности витрины региона
Количество записей за период	Отчет по конверсии, источникам записи и недоступности витрины региона
Изменение количества записей по отношению к предыдущему периоду	Отчет по конверсии, источникам записи и недоступности витрины региона
Количество записей в разрезе источников записи	Отчет по конверсии, источникам записи и недоступности витрины региона
Доля записей в разрезе источников записи	Отчет по конверсии, источникам записи и недоступности витрины региона

Ж.3.1.2 Технические метрики

В таблице Таблица 12 представлен перечень технических метрик услуги «Запись на прием к врачу (на витринах данных). Показатели рассчитываются на основе данных о событиях, представленных в разделе Ж.4.1. Данные поступают из региональных витрин субъекта РФ и ЕПГУ в БД Clickhouse.

Таблица 12 Технические метрики

Показатель	Отчет
Количество технических ошибок РМИС	Запись к врачу 10700. Воронка. Пользователи Запись к врачу 10700. Воронка. Сессии
Доля технических ошибок РМИС в общем количестве результатов шага	
Количество ошибок ЕПГУ	
Доля ошибок ЕПГУ в общем количестве результатов шага	
Количество ошибок ПОДД	
Доля ошибок ПОДД в общем количестве результатов шага	
Количество ошибок в разрезе типов ошибок	
Доля ошибок в разрезе типов ошибок в общем количестве ошибок	

Ж.3.1.3 Метрики качества данных

В таблице Таблица 13 представлен перечень метрик качества данных услуги «Запись на прием к врачу (на витринах данных)». Показатели рассчитываются на основе данных, полученных из ЕИП НСУД.

Таблица 13 Метрики качества данных

Показатель	Отчет
Доля пациентов в регионе, имеющих прикрепление к медицинской организации и прикрепленных к существующим и актуальным медицинским организациям	Качество данных о прикреплении к МО
Процент доступных врачей специальности в МО	Доступность врачебных специальностей
Факт отсутствия доступных слотов на ЕПГУ в разрезе врачебных специальностей и медицинских организаций по информации, размещенной на витрине данных»	
Среднее значение инцидентов по специальности в МО	
Среднее значение инцидентов по специальности в Субъекте РФ	
Доля доступных слотов, размещаемых для ЕПГУ	Доступность слотов, размещаемых на ЕПГУ
Количество пациентов	Конкурентные слоты
Количество врачей	
Общее количество слотов всех типов (врачей указанной специальности в указанной МО) на заданный период времени (14 дней)	
Количество свободных слотов всех типов (врачей указанной специальности в указанной МО) на заданный период времени (14 дней)	
Общее количество слотов с типом, предназначенным для записи с ЕПГУ (врачей указанной специальности в указанной МО) на заданный период времени (14 дней)	
Количество свободных слотов с типом, предназначенным для записи с ЕПГУ (врачей указанной специальности в указанной МО) на заданный период времени (14 дней)	
Количество прикреплений пациентов к медицинским организациям в разрезе пола и возраста	Прикрепление к МО по полу и возрасту
Количество прикреплений пациентов к медицинским организациям всего	
Процент отклонений прикреплений пациентов к медицинским организациям в разрезе пола и возраста в сравнении с данными ФОМС	

Ж.3.2 Перечень показателей услуги «Запись на прием к врачу (КУ ФЭР)»

Ж.3.2.1 Бизнес-метрики

В таблице Таблица 14 представлен перечень бизнес-метрик услуги «Запись на прием к врачу (КУ ФЭР)». Показатели рассчитываются на основе данных о событиях, представленных в разделе Ж.4.1.2. Данные поступают из КУ ФЭР и ЕПГУ в БД Clickhouse.

Таблица 14 Бизнес-метрики

Показатель	Отчет
Количество уникальных пользователей	Запись к врачу 600204. Воронка. Пользователи
Количество уникальных сессий	Запись к врачу 600204. Воронка. Сессии
Количество успешно идентифицированных	Запись к врачу 600204. Воронка. Пользователи
Доля успешно идентифицированных в общем количестве результатов шага	Запись к врачу 600204. Воронка. Сессии
	Сравнение форм записи к врачу. Пользователи Сравнение форм записи к врачу. Сессии

Показатель	Отчет
Количество совершенных выборов медорганизаций	Запись к врачу 600204. Воронка. Пользователи Запись к врачу 600204. Воронка. Уникальные сессии
Доля совершенных выборов медорганизаций в общем количестве результатов шага	
Количество полученных списков специальностей	Запись к врачу 600204. Воронка. Пользователи Запись к врачу 600204. Воронка. Сессии Сравнение форм записи к врачу. Пользователи Сравнение форм записи к врачу. Сессии
Доля полученных списков специальностей в общем количестве результатов шага	
Количество полученных списков фамилий и кабинетов	Запись к врачу 600204. Воронка. Пользователи Запись к врачу 600204. Воронка. Сессии
Доля полученных списков фамилий и кабинетов в общем количестве результатов шага	
Количество полученных списков доступных слотов	Запись к врачу 600204. Воронка. Пользователи Запись к врачу 600204. Воронка. Сессии Сравнение форм записи к врачу. Пользователи Сравнение форм записи к врачу. Сессии
Доля полученных списков доступных слотов в общем количестве результатов шага	
Количество успешно записавшихся	
Доля успешно записавшихся в общем количестве результатов шага	
Количество организационных ошибок РМИС	Запись к врачу 600204. Воронка. Пользователи Запись к врачу 600204. Воронка. Сессии
Доля организационных ошибок РМИС в общем количестве результатов шага	
Количество ушедших пользователей	
Доля ушедших пользователей в общем количестве результатов шага	
Количество ошибок в разрезе типов ошибок	
Доля ошибок в разрезе типов ошибок в общем количестве ошибок	

Ж.3.2.2 Технические метрики

В таблице Таблица 15 представлен перечень технических метрик услуги «Запись на прием к врачу (КУ ФЭР)». Показатели рассчитываются на основе данных о событиях, представленных в разделе Ж.4.1.2. Данные поступают из КУ ФЭР в БД Clickhouse.

Таблица 15 Технические метрики

Показатель	Отчет
Количество технических ошибок РМИС	Запись к врачу 600204. Воронка. Пользователи Запись к врачу 600204. Воронка. Сессии
Доля технических ошибок РМИС в общем количестве результатов шага	
Количество ошибок в разрезе типов ошибок	
Доля ошибок в разрезе типов ошибок в общем количестве ошибок	

Ж.4 Методика расчета показателей

Ж.4.1 Передача данных по событиям услуги «Запись на прием к врачу»

Ж.4.1.1 Данные по событиям услуги «Запись на прием к врачу (на витринах данных)»

В отчет по событиям записи на прием к врачу (Пилот запись к врачу 2.0) передаются показатели событий при прохождении сценария пользователем. На каждое событие с фронта

отправляется по REST /api/zdrav-metrics/v1/form-event. Обработка запроса происходит внутри REST /api/zdrav-metrics/v1/form-event, конвертируется в плоскую структуру и дополняется данными пользователя: oid из токена, персональные данные из ЕСИА. Событие направляется через топик kafka formEventZdravMetricsTopic в ClickHouse. Параметры отправляемого сообщения в kafka указаны в таблице Таблица 16, для передачи информации о доступности услуги в таблице Таблица 17. Каждое событие (Таблица 18) кладется в отдельную таблицу.

Таблица 16 Параметры отправляемого сообщения в kafka

Элемент	Описание	Правило заполнения
type	Тип события	zdrav-event
oid	Идентификатор пользователя в ЕСИА	заполняется данными из ЕСИА
assuranceLevel	Уровень учетной записи	
dateTime	Системное время	системное время
service	Услуга	значением из запроса
event	Наименование события	
session	ID сессии пользователя	
data_result	Результат	
data_errorDescription	Описание кода ошибки	
data_timing	Время выполнения запроса	
data_data1	Бизнес атрибут	

Таблица 17 Параметры отправляемого сообщения в kafka, если передача информации о доступности услуги

Элемент	Описание	Правило заполнения
type	Тип события	zdrav-event
oid	Идентификатор пользователя в ЕСИА	пустое значение
assuranceLevel	Уровень учетной записи	пустое значение
dateTime	Системное время	системное время
service	Услуга	10700
event	Наименование события	check_service
session	ID сессии пользователя	пустое значение
data_result	Результат	всегда «0»
data_errorDescription	Описание кода ошибки	пустое значение
data_timing	Время выполнения запроса	пустое значение
data_data1	Код ОКATO региона	значение okato из ответа
data_data2	Доступность в проде	значение prod_in_active из ответа

Таблица 18 Передаваемые события

Событие	Условие возникновения события	event	event_data	Правило заполнения
Переход на услугу	Открыта страница /10700/1/form	service_init	data1 data2 result errorDescription timing	result = 0 - успешно открыта экранная форма result = 1 - при открытии формы произошла ошибка data1, data2, errorDescription, timing - передать пустое значение.
Выбор сценария записи	Пользователь на экране «Кого записать на приём?» выбирает один из вариантов	choice	data1 data2 result errorDescription timing	data1 = choice_self choice_child choice_other (в зависимости от выбора пользователя: себя ребенка другого человека) result = 0 - успешно открыта экранная форма result = 1 - при открытии формы произошла ошибка

Событие	Условие возникновения события	event	event_data	Правило заполнения
				data2, errorDescription, timing - передать пустое значение.
Уже есть активная запись	Пользователь на экране «Выберите специальность врача» выбирает специалиста из списка, к которому уже есть активная запись.	show_active_record	data1 data2 result errorDescription timing	result = 0 - успешно открыта экранная форма result = 1 - при открытии формы произошла ошибка data1, data2, errorDescription, timing - передать пустое значение.
Определение региона	Пытаемся определить регион - в какую витрину направить запрос. Регион определяем по полису ОМС или адресам. Далее рест на проверку доступности услуги в регионе.	search_region	data1 data2 result errorDescription timing	data1 - передать регион, который был определен data2 - передавать значение по какому признаку определили регион: MDCL_PLCY - если по полису ОМС, PLV_ADDRESS - если по адресу фактического проживания, PRG_ADDRESS - если по адресу постоянной регистрации, PTA_ADDRESS - если по адресу временной регистрации. OTHER - иное (регион выбран пользователем вручную) result = 0 - регион доступен для записи через витрины result = 1 - регион не доступен
Идентификация пользователя (шаг 1)	Получен ответ на рест валидации /api/hsi/v1/dictionary/ValidatePatient	validate_patient	data1 data2 result errorDescription timing	data1 = validateType (Событие отправляется не на каждый рест валидации, а на итоговый результат. Передаем тип валидации 1, 2, 3, 4, 5, которым получен идентификатор пациента. Если пациент найден по старому полису ОМС - к типу прибавить b, т.е. в data1 будет передаваться одно из значений: 1b, 2b, 3b, 4b. Если пациент найден по полису, который был введен вручную - к типу прибавить "с" (в data1 будет передаваться одно из значений: 1c, 2c, 3c, 4c) data2 = patient_id из ответа реста на валидацию (пустое значение - если не нашли пользователя). result = errorCode из ответа реста (или httpCode, если != 200) errorDescription - описание ошибки из ответа реста на валидацию. Время ответа
Запрос списка специальностей (шаг 2)	Получен ответ на рест для получения списка специалистов /api/hsi/v1/dictionary/Get_Specialist или получен ответ на рест /api/hsi/v1/dictionary/Get_Resource (в сценарии записи в МО)	get_specialist	data1 data2 result errorDescription timing	data1 = mo_id из запроса (для сценария записи МО) или пусто; data2 = список МО в разрезе должностей специалистов (для сценария "По специальности") result = errorCode из ответа реста (или httpCode, если != 200) errorDescription = описание ошибки из ответа реста. Время ответа
Запрос активных	Получен ответ на рест для получения	get_booking	data1 data2	data1 = количество записей

Событие	Условие возникновения события	event	event_data	Правило заполнения
записей пользователя	текущих/прошедших записей /api/hsi/v1/dictionary/Get_Books		result errorDescription timing	result = errorCode из ответа реста (или httpCode, если != 200) errorDescription = описание ошибки из ответа реста. Время ответа
Запрос расписания врача выбранной специальности (Шаг3)	Получен ответ на рест /api/hsi/v1/dictionary/GetCountFreeSlot	get_slots	data1 data2 result errorDescription timing	data1 = в зависимости от входных параметров запроса передать: – значение post_id из запроса (идентификатор выбранного специалиста). Наименование должности приведено в приложении ПРИЛОЖЕНИЕ А. – значение speciality_id из запроса (идентификатор специалиста) - для сценария по направлению на консультацию – значение service_code из запроса (код услуги) - для сценария по направлению на исследование data2 - передавать количество доступных слотов в разрезе МО; result = errorCode из ответа реста (или httpCode, если != 200) errorDescription = описание ошибки из ответа реста. Время ответа
Запрос на запись к врачу (Шаг 4)	Получен ответ на рест api/lk/v1/equeue/agg/book	request_booking	data1 data2 result errorDescription timing	data1 = значение post_id (id специалиста, к которому запись). Наименование должности приведено в приложении Приложение Ж.1. data2 = значение слота на бронирование (slotId) errorDescription = описание ошибки из ответа реста. Время ответа result = errorCode из ответа реста (или httpCode, если != 200) result При успешной записи передавать <i>result=0</i> . При получении ошибки - значение из errorCode.
Запрос на отмену текущей записи	На экране "Уже есть запись" или в ЛК пользователь нажимает "Отменить", отправляется запрос в витрину. Получен ответ на рест /api/v1/booking/cancel или /api/lk/v1/calendar/events/{extId}/cancel	request_cancel	data1 data2 result errorDescription timing	data1 = значение идентификатора букинга на отмену (bookId). result = errorCode из ответа реста (или httpCode, если != 200) errorDescription = описание ошибки из ответа реста. Время ответа
Прохождение опросов по услуга Здрава	Пользователь на экране «Не смогли записаться к врачу?» выбирает один из вариантов	survey_results	data1 data2 service result errorDescription timing	service - источник опроса/id опроса data1 = номер выбранного ответа по порядку. Возможные значения: 0, 1 - N, где: 0 - перешел на страницу с опросом, но не выбрал ни одно из значений. При передачи значения 0 поле data2 оставлять пустым

Событие	Условие возникновения события	event	event_data	Правило заполнения
				1-N : порядковый номер запроса на форме data2 = расшифровка выбранного ответа - Текстовое описание варианта ответа на форме (в том числе вариант введенный вручную) Например, (data1 = data2): 1= Нет нужной мне медицинской организации 2= Нет нужной специальности врача 3= Не было моего лечащего врача 4=Не удобное время приема 5=Передумал(а) записываться 6=Не получается записаться на выбранное время 7=Другое: текст пользователя result = всегда 0 errorDescription = передать пустое значение timing = 0
Выбор способа записи на прием	Пользователь на экране «Выберите способ записи на прием» выбирает один из вариантов	choice_method	data1 data2 result errorDescription timing	data1 = specialty/ referral / medorg (в зависимости от выбора пользователя: по специальности врача/по направлению/по медорганизации result = 0 - успешно открыта экранная форма result = 1 - при открытии формы произошла ошибка data2, errorDescription, timing - передать пустое значение.
Запрос списка направлений	Получен ответ на рест /api/nsi/v1/dictionary/Referral	get_referral	data1 data2 result errorDescription timing	data1 = количество записей result = errorcode из ответа реста (или httpCode, если != 200) errorDescription = errormessage описание ошибки из ответа реста. Время ответа
Запрос списка медорганизаций	Получен ответ на рест /api/nsi/v1/dictionary/Get_MO	get_mo	data1 data2 result errorDescription timing	data1 = количество записей result = errorcode из ответа реста (или httpCode, если != 200) errorDescription = errormessage описание ошибки из ответа реста. Время ответа
Запрос ресурсов, которые имеют период недоступности в выбранный день	Получен ответ на рест /api/nsi/v1/dictionary/GetUnaccessibleResources	get_unaccessible_resource	data1 data2 result errorDescription timing	data1 - в зависимости от входных параметров запроса передать: значение post_id из запроса (идентификатор выбранного специалиста) значение service_code из запроса (код услуги) - для <i>сцен. по направлению на исследование</i> . data1 - дата, выбранная в календаре (из запроса) data2 - количество уникальных ресурсов result = errorCode из ответа реста (или httpCode, если != 200) errorDescription - пустое значение timing = время ответа

Событие	Условие возникновения события	event	event_data	Правило заполнения
Запрос доступных слотов для выбранного врача	Получен ответ на рест /api/nsi/v1/dictionary/Get_Free_Slot (type=1) (при клике на врача на экране выбора слотов)	get_slots_by_resource	data1 data2 result errorDescription timing	data1 - значение resource_id из запроса data2 - количество слотов result = errorCode из ответа реста (или httpCode, если != 200) errorDescription = описание ошибки из ответа реста. timing = время ответа
Запрос информации о прикреплении по д/учету	Получен ответ на рест /api/nsi/v1/dictionary/Get_Observation	get_observation	data1 data2 result errorDescription timing	data1 = id пациента из запроса data2 - result = errorCode из ответа реста (или httpCode, если != 200) errorDescription = описание ошибки из ответа реста. timing = время ответа
Проверка пользователя в витрине ФМБА	Получен ответ на рест /api/zdrav-appointments/v1/checkfmbapatient/1	check_fmba_patient	data1 data2 result errorDescription timing	data1 - patient_id из ответа реста /api/zdrav-appointments/v1/checkfmbapatient/1 (или пустое значение, если не найден) result = значение http-кода errorDescription - передать пустое значение. timing - время ответа реста. data2 - пустое значение
Информация о доступности услуги	Запрошены с заданной периодичностью данные о настройках по услуге 10700 - вызов рест /api/nsi/v1/epgu/region/SE RVICEREGIONS	check_service	data1 data2 result errorDescription timing	data1 значение okato из ответа data2 = значение prod_in_active из ответа result = 0 errorDescription - пустое значение timing = пустое значение
Редирект	Произошел переход с формы «Запись на прием к врачу (на витринах данных)» на форму «Запись на прием к врачу (КУ ФЭР)» или наоборот	redirect	data1 data2 result errorDescription timing	data1: – 10700_606204, если переход с 10700 на 606204 – 606204_10700, если если переход с 606204 на 10700 data2 – код экрана 10700, с которого/на который осуществляется переход result – 0 errorDescription – пустое значение timing – пустое значение

Для передачи данных по подпискам используется топик kafka zdravAppointmentsMetricsTopic. Параметры отправляемого сообщения в kafka, если происходит обработка дельты из витрины указаны в таблице Таблица 19, если при обработке отмены записи в таблице Таблица 20. Каждое событие (Таблица 21) кладется в отдельную таблицу.

Таблица 19 Параметры отправляемого сообщения в kafka, если обработка дельты из витрины

Элемент	Описание	Правило заполнения
type	Тип события	const.: zdrav-event
service	Услуга	10701 - если в сообщении vitrinaType = MSDM 10731 - если vitrinaType = FMBA (для всех остальных типов шагов, а также для result = SUCCESS если book_type != "BLOOD_DONATION")
event	Наименование события	const.: notice_handling_result

Элемент	Описание	Правило заполнения
patientId	Идентификатор пользователя в витрине	значение patient_id из сообщения
bookId	Идентификатор записи	значение id из сообщения
dateTime	Время события	Системное время
region	Код ОКАТО региона	значение items.okato полученное в ответ на /api/hsi/{version}/epgu/region/SERVICEREGIONS ; если получили пустой массив - передать пустое значение
oid	Идентификатор пользователя в ЕСИА	значение из ответа ЕСИА, если найден; иначе - пустое значение
assuranceLevel	Уровень учетной записи	значение assuranceLevel, если в ЕСИА найден; иначе - пустое значение
result	Результат	зависит от того, на каком шаге прервалась обработка, см. таблицу
errorDescription	Описание кода ошибки	см.Таблица 21
data_data1	Идентификатор подписки	значение subscriptionId из сообщения
data_data2	Источник записи/обновления	значение tag_source из сообщения
data_data3	Статус записи	значение status_code из сообщения

Таблица 20 Параметры отправляемого сообщения в kafka, если обработка отмены записи

Элемент	Описание	Правило заполнения
type	Тип события	const.: zdrav-event
service	Услуга	10701 - если в сообщении vitrinaType = MSDM 10731 - если vitrinaType = FMBA (для всех остальных типов шагов, а также для result = SUCCESS если book_type != "BLOOD_DONATION")
event	Наименование события	const.: booking_cancel
patientId	Идентификатор пользователя в витрине	значение patient_id из запроса
bookId	Идентификатор записи	значение bookId из запроса
dateTime	Время события	Системное время
region	Код ОКАТО региона	Передать пустое значение
oid	Идентификатор пользователя в ЕСИА	Передать пустое значение
assuranceLevel	Уровень учетной записи	Передать пустое значение
result	Результат	значение status.statusCode из ответа (или http-код если не 200)
errorDescription	Описание кода ошибки	значение status.statusMessage из ответа
data_data1	Код ИС региона	значение routeNumber из запроса
data_data2	-	Передать пустое значение
data_data3	-	Передать пустое значение

Таблица 21 Передаваемые события

Событие	Условие возникновения события	event_name	event_data	Правило заполнения
Результат обработки дельт из витрины	Обработано уведомление, полученное из витрины	notice_handling_result	data1 data2 data3 result	data1 = значение subscriptionId из сообщения data2 = значение tag_source из сообщения data3 = значение status_code из сообщения Отправлять в Result и errorDescription Если не удалось получить значение мнемоники витрины по subscriptionId (получен пустой массив в ответе на /api/hsi/{version}/epgu/region/SERVICEREGIONS) : Result = NOT_FOUND_REGION, errorDescription = По подписке #subscriptionId нет настроек

Событие	Условие возникновения события	event_name	event_data	Правило заполнения
				Если не получены перс.данные из витрины (получен пустой массив в ответе) Result = NOT_FOUND_PATIENT, errorDescription = Не найден пациент Если Не найден пользователь в ЕСИА (при поиске УЗ в ЕСИА не найдена УЗ ни одним из методов). Получен http-код 404 Result = NOT_FOUND_OID, errorDescription = Не найден пользователь в ЕСИА Если не получена детальная информация о записи в витрине (получен пустой массив в ответе) Result = NOT_FOUND_BOOK, errorDescription = Нет данных о записи в витрине Если время визита по записи в прошлом Result = VISITTIME_PASSED, errorDescription = Время визита по записи в прошлом Если Успешно создана/отменена запись Result = SUCCESS, errorDescription = Успешно обработано сообщение Если на любом шаге получили техническую ошибку (http-код 5xx/4xx). Если при отправке запроса в LK получили http-код 200, и errorCode!=0 Result = TECHNICAL_ERROR, errorDescription = Текст полученной ошибки
Отмена записи, полученной из витрины	Обработано уведомление об отмене, полученное из витрины	booking_cancel	errorDescription data1 data2 data3 result errorDescription	data1 Код ИС региона значение routeNumber из запроса data2 - Передать пустое значение data3 - Передать пустое значение result Результат значение status.statusCode из ответа (или http-код если не 200) errorDescription Описание кода ошибки значение status.statusMessage из ответа

Ж.4.1.2 Данные по событиям услуги «Запись на прием к врачу (КУ ФЭР)»

В отчет по событиям записи на прием к врачу (Пилот запись к врачу 2.0) передаются показатели событий при прохождении сценария пользователем. Данные по событиям передаются в топики `analitic_dictionary` и `analitic_queue`. Сообщения из этих топиков, относящиеся к услуге передаются в топик `formEventZdravMetricsTopic`. Через `formEventZdravMetricsTopic` событие направляется в ClickHouse. Параметры отправляемого сообщения в kafka указаны в таблице Таблица 22. Каждое событие (Таблица 23) кладется в отдельную таблицу.

Таблица 22 Параметры отправляемого сообщения в kafka

Элемент	Описание	Правило заполнения
type	Тип события	const.: zdrav-event
service	Услуга	const.: 600204, если "eserviceId" = "10000025167"
assuranceLevel	Уровень учетной записи	пустое значение
dateTime	Время события	Системное время
data_timing	Время ответа	пустое значение
oid	Идентификатор пользователя в ЕСИА	см. Таблица 23
session	ID сессии пользователя	
event	Наименование события	
data_result	Результат	
data_errorDescription	Описание кода ошибки	
data_data1	Бизнес атрибут	
data_data2	Бизнес атрибут	

Таблица 23 Передаваемые события

Событие	Условие возникновения события	event_name	Топик	Правило заполнения параметров
Валидация пользователя (шаг 1)	Пользователь подтвердил свои данные, выбрал регион, где хочет посетить врача Вызывается рест /api/lk/v1/equeue/10000025167/agg/ref/items (ref_name="getValidatePatient")	validate_patient	analitic_equeue Фильтр по: "eserviceId": "10000025167" refName: getValidatePatient	event - validate_patient oid - значение esiald из сообщения; session - значение Session_Id из request; data_result - значение errorDetail.errorCode из response data_errorDescription - значение errorDetail.errorMessage из response data_data1 - 0 data_data2 - значение version из response
Выбор сценария записи	Пользователь на экране "Кого записать к врачу?" выбирает один из вариантов	choice	Фильтр "eserviceId": "10000025167" refName: getValidatePatient	event - choice oid - значение esiald из сообщения; session - значение Session_Id из request; data1 = choice_self choice_child choice_other (в зависимости от выбора пользователя: себя ребенка другого человека) result = 0 - успешно открыта экранная форма result = 1 - при открытии формы произошла ошибка data2, errorDescription, timing - передать пустое значение.
Запрос направлений	Отправляется запрос в ведомство для получения списка направлений	get_referral	Фильтр "eserviceId": "10000025167" refName: getValidatePatient	data1 = количество записей из totalItems result = errorcode из ответа реста (или httpCode, если != 200) errorDescription = errormessage описание ошибки из ответа реста.
Запрос доступных МО (шаг 2)	Отправляется запрос в ведомство для получения списка доступных МО Вызывается рест: /api/hsi/v1/dictionary/mzrf_lpu_equeue_smev3	get_mo	analitic_dictionary Фильтр по: "formId": "600204/1" либо "606204/1"	event - get_mo oid - значение oid из сообщения; session - значение Session_Id из request; data_result - значение error.code из response

Событие	Условие возникновения события	event_name	Топик	Правило заполнения параметров
			"dictionaryName": "mzrf_lpu_equeue_smev3"	data_errorDescription - значение error.message из response data_data1 - значение total из response data_data2 - пустое значение.
Запрос специальностей в выбранном МО (шаг 3)	На экране карты пользователь выбирает МО. Отправляется запрос для получения списка специальностей, доступных в этой МО. Вызывается рест: /api/lk/v1/equeue/10000 025167/agg/ref/items (ref_name="ServiceOrSpecs")	get_specialist	analitic_equeue Фильтр по: "eserviceId": "10000 025167" refName: ServiceOrSpecs	event - get_specialist oid - значение oid из сообщения; session - значение Session_Id из request; data_result - значение errorDetail.errorCode из response data_errorDescription - значение errorDetail.errorMessage из response data_data1 - количество полученных специалистов data_data2 - пустое значение.
Запрос ресурсов (врачей) по выбранной специальности (шаг 4)	Из списка, пользователь выбирает специальность врача. Отправляется запрос на получение списка доступных врачей по выбранной специальности. Вызывается рест: /api/lk/v1/equeue/10000 025167/agg/ref/items (ref_name="Resource")	get_resource	analitic_equeue Фильтр по: "eserviceId": "10000 025167" refName: Resource	event - get_resource oid - значение oid из сообщения; session - значение Session_Id из request; data_result - значение errorDetail.errorCode из response data_errorDescription - значение errorDetail.errorMessage из response data_data1 - количество доступных врачей data_data2 - пустое значение.
Запрос доступных слотов для выбранного врача (шаг 5)	Из списка пользователь выбирает врача (ФИО). Отправляется запрос на получение доступных слотов для этого врача. Вызывается рест: /api/lk/v1/equeue/10000 025167/agg/slots	get_slots	analitic_equeue Фильтр по: "eserviceId": "10000 025167" в response передан массив "slots"	event - get_slots oid - значение oid из сообщения; session - значение Session_Id из request; data_result - значение errorDetail.errorCode из response data_errorDescription - значение errorDetail.errorMessage из response data_data1 - значение Resource_Id из request data_data2 - кол-во слотов.
Запрос на букинг слота (шаг 6)	Пользователь выбирает день и время, нажимает "Записаться". Отправляется запрос на букинг выбранного слота. Вызывается рест: /api/lk/v1/equeue/10000 025167/agg/book?srcSystem=BETA	request_booking	analitic_equeue Фильтр по: "eserviceId": "10000 025167" в response передан "bookId" и timeSlot	event - request_booking oid - значение oid из сообщения; session - значение Session_Id из request; data_result: значение status.statusCode (если успех) значение errorDetail.errorCode (при ошибке) из response data_errorDescription: значение status.statusMessage (если успех) значение errorDetail.errorMessage (при ошибке) из response data_data1 - значение doctorid из request data_data2 - значение slotId из request.

Событие	Условие возникновения события	event_name	Топик	Правило заполнения параметров
Запрос на отмену текущей записи	Пользователь отменяет запись в личном кабинете	request_cancel	analitic_equeue Фильтр по: "path": "/api/lk/{version}/calendar/events/{extId}/cancel" "requestParams"."e serviceId":"100000 00204"	data1 = значение идентификатора букинга ("requestParams"."extId"). result = значение response.statusCode errorDescription = описание ошибки из ответа песта (response.statusMessage). Время ответа

Ж.4.2 Расчет показателей по результатам прохождения услуги «Запись на прием к врачу»

Ж.4.2.1 Расчет показателей услуги «Запись на прием к врачу (на витрине данных)»

Расчет результатов шагов по пользователям

Пользователь – конкретный oid, может делать множество попыток записей (session). Для расчета воронки записи к врачу берется в расчет все попытки пройти тот или иной шаг. Существуют следующие шаги:

- идентификация пациента;
- получение списка специальностей;
- получение списка доступных слотов;
- запись.

Для расчета времени считается время выбора прохождения первого шага – идентификация пациента. Идентификация происходит после определения региона (search_region). Если регион недоступен для записи, то пользователь далее не попадет ни в один из шагов и не будет участвовать в подсчете конверсии.

Если пользователь в течение суток (00:00-23:59:59) начал проходить шаги, а именно попал в воронку, то он засчитывается как уникальный пользователь. Вне зависимости от того, сколько раз он пройдет тот или иной шаг или сколько раз запишется, при успешном прохождении шага, считается, что пользователь достиг того или иного шага всего один раз.

Например, пользователь 14 февраля записал к врачу 10 человек, следовательно, он будет считаться как один пользователь, который прошел все шаги воронки.

Для подсчета результата конкретного шага делается группировка каждого пользователя и всех его сессий по суткам. Если пользователь успешно достиг конкретного шага (data_result = 0), то это считается в конверсии этого шага. То есть по отношению к уникальным пользователям рассчитываются доли успешных событий по каждому из перечисленных шагов.

Если пользователь не смог пройти шаг, считается data_result > 0. Это говорит о том, что у пользователя оборвалась сессия.

Расчет ошибок шага по пользователям

Для определения ошибки берется минимальный показатель ошибки из всех ошибок за сутки. То есть, если пользователь на шаге записи сталкивается из 5 попыток с кодами ошибок 200, 107, 3, 7 и 5, то в отчет попадает ошибка с минимальной величиной – 3. Соответственно, если пользователь успешно проходит шаг, у него код будет 0.

Считается общее количество ошибок, сгруппированных по типам, обозначенным в приложении Приложение Ж.2. По отношению ко всем результатам шага рассчитываются доли по каждому из типов ошибок.

Детализация ошибок предполагает, что считаются все конкретные ошибки (ошибки группируются по описанию), возникшие у пользователей, то есть в расчет берутся все неуспешные попытки. По отношению к общему числу ошибок считается доля конкретных описаний ошибок.

Расчет результатов опроса

Результаты опроса фиксируются, когда пользователь при прохождении услуги пользователь пытался покинуть форму. Считается количество ответов каждого типа и доля ответов в общем количестве ответов. Маппинг с регионом осуществляется через идентификатор сессий.

Расчет показателей результатов шагов и ошибок по сессиям происходит аналогично как по пользователям, но группировка идет по сессии.

Примеры скриптов представлены в приложении Приложение Ж.3.

Ж.4.2.2 Расчет показателей услуги «Запись на прием к врачу (КУ ФЭР)»

Расчет показателей результатов шагов и ошибок шага услуги «Запись на прием к врачу (КУ ФЭР)» считается аналогичным образом, как для услуги «Запись на прием к врачу (на витринах данных)». Расчет результатов опроса отсутствует ввиду отсутствия опроса на форме услуги.

При расчете воронки учитываются следующие шаги:

- идентификация пациента;
- получение списка медорганизаций;
- получение списка специальностей;
- получение списка фамилий врачей/кабинетов;
- получение списка доступных слотов;
- запись.

При подсчете конверсии учитывается, что последнем шаге записи успешная запись имеет код 201.

Примеры скриптов представлены в приложении Приложение Ж.4.

Ж.4.3 Расчет длительности недоступности услуги в регионе

Каждые 10 минут запрашиваются данные о настройках по услуге посредством вызова рест `/api/hsi/v1/epgu/region/SERVICEREGIONS`. Длительность недоступности витрины региона за выбранный период времени в минутах определяется как количество раз полученного значения `false` в параметре `data_data2` из передаваемого события `check_service` (Таблица 18), умноженное на 10.

Доля недоступности услуги считается как отношение длительности недоступности к общему времени за исследуемый период. Общее время в минутах считается как количество раз полученного события `check_service` (Таблица 18), умноженное на 10.

Пример определения периода недоступности представлен в таблице Таблица 24.

Таблица 24 Пример выборки периодов недоступности услуги

data time	data_data2
2023-02-22 15:00:00	true
2023-02-22 15:10:00	true
2023-02-22 15:20:00	false
2023-02-22 15:30:00	false
2023-02-22 15:40:00	true
2023-02-22 15:50:00	true
2023-02-22 16:00:00	true
2023-02-22 16:10:00	false
2023-02-22 16:20:00	false
2023-02-22 16:30:00	false
2023-02-22 16:40:00	true
2023-02-22 16:50:00	true
2023-02-22 17:00:00	true

В таблице выделены периоды, когда было получено событие со значением в параметре `data2` – `false`. За исследуемый период количество деактиваций составило 5 или длительность недоступности составила $5 \cdot 10 = 50$ минут, а количество всех полученных событий составило 12 или общее время за исследуемый период составило $12 \cdot 10 = 120$ минут. Соответственно, доля недоступности услуги составляет $50/120 = 0,4167$ или 41,67%.

Данный показатель следует рассчитывать кратно 1 дню (за день количество передаваемых событий должно равняться 144 разам). Если подключение услуги к мониторингу было позднее, чем начало исследуемого периода, то общее время считается с момента подключения услуги.

Ж.4.4 Расчет конверсии

Ж.4.4.1 Расчет целевой конверсии

Изменяемые параметры:⁵

- период (например, последние 30 дней)
- минимальный порог дневной конверсии (например, 5%)
- минимальный порог количества уникальных пользователей (например, 200 пользователей)
- процентиль, который служит нормативом (например, 70-й процентиль)

Порядок расчета:

1. Для каждого региона считаются дневные конверсии за все дни выбранного периода.

⁵ Показатели могут быть скорректированы

2. Из полученного списка результатов убираются те дни, где конверсия ниже минимального порога.

3. Из оставшихся результатов убираются те дни, где пользователей было меньше, чем указано в минимальном пороге пользователей.

4. По оставшемуся списку проводится расчет 70-ого перцентиля.

5. Полученный результат принимается за целевую конверсию.

В итоге целевой конверсией служит граница, выше которой дневная конверсия региона поднималась в 30% дней выбранного периода (за исключением непоказательных дней аварий и малой активности на услуге). Пример скрипта для расчета расположен в приложении Приложение Ж.5

Ж.4.4.2 Расчет фактической конверсии

В зависимости от выбранного способа расчета, конверсией является отношение количества успешно записавшихся пользователей / сессий к числу пользователей / сессий, начавших путь на форме.

При расчете используется группировка в рамках суток. То есть внутри одних суток пользователь / сессия считаются 1 раз. Результатов по пользователю / сессии может быть получено несколько в рамках суток. Успешно сконвертированным считается пользователь / сессия, которые получили за сутки хотя бы одну успешную запись.

Пример расчета по пользователю:

За сутки было 3 пользователя. Каждый совершил 5 попыток записи. Из них пользователю А удалось записаться 3 раза из 5, пользователю Б – 2 раза из 5, пользователю В – ни разу.

Пользователей, которые успешно записались хотя бы раз – было двое (А и Б). А всего пользователей на форме – трое (А, Б, В). Конверсия = $2/3$, то есть 66,7%.

Аналогичный расчет по сессиям. В этом случае `oid` пользователя в группировках заменяется на `session_id`.

Ж.4.5 Методика оценки качества данных, размещаемых в витринах субъектов РФ

Ж.4.5.1 Основные сведения об Информационном обмене и составе информации, размещенных на витринах данных

Наименование услуги: «Услуга по записи на прием к врачу».

Схема функционирования услуги при обращении к Витрине данных ведомства приведена на рисунке Рисунок 17.

Услуга предоставляется при самостоятельном обращении пациента в медицинскую организацию в электронной форме для осуществления предварительной записи на прием к врачу, включая диагностические исследования и лечебные процедуры, а также по направлению.

Услуга предоставляется в той медицинской организации, к которой у гражданина есть прикрепление, либо при наличии у пациента направления установленной формы в медицинскую организацию, отличную от медицинской организации по прикреплению.

Услуга предоставляется в следующих случаях:

- а) при обращении пациента в медицинскую организацию к врачам тех специальностей, к которым открыта самостоятельная запись на прием;
- б) при обращении пациента к лечащему врачу;
- в) при обращении к врачу (или врачу, его замещающему по соответствующей специальности) пациента, состоящего у него на диспансерном наблюдении;
- г) при наличии у пациента направления установленной формы на прием к врачу-специалисту медицинской организации для получения первичной специализированной медико-санитарной помощи;
- д) при наличии направления установленной формы на диагностические исследования или лечебные процедуры в подразделения медицинских организаций.
- е) при обращении пациента в медицинскую организацию для записи на прием к следующим специалистам, если более широкий перечень не предусмотрен иными актами:

- врач-терапевт участковый;
- врач-педиатр участковый;
- врач общей практики (семейный врач);
- врач-хирург;
- врач-акушер-гинеколог;
- врач-офтальмолог;
- врач-оториноларинголог;
- врач-стоматолог;
- врач-стоматолог-терапевт;
- врач-стоматолог детский;
- врач-психиатр детский;
- врач-психиатр подростковый;
- врач-фтизиатр;
- врач-психиатр-нарколог;
- врач-психиатр детский участковый;
- врач-психиатр подростковый участковый;
- врач-психиатр-нарколог участковый;
- врач-детский хирург.

При предоставлении услуги по записи на прием к врачу в электронной форме гражданам предоставляется следующая информация:

- а) о медицинских организациях, в которых гражданину доступна услуга по записи на прием к врачу. Перечень медицинских организаций зависит от территориального прикрепления гражданина к медицинской организации;
- б) о медицинских работниках, к которым гражданину доступна запись на прием;
- в) о доступном времени для записи на прием к медицинскому работнику;
- г) о направлениях гражданина (запись по которым доступна).

Организация записи пользователя на прием к врачу в электронной форме с учетом требования обеспечения возможности записи на прием к врачу в период не менее 14 дней с текущей даты.

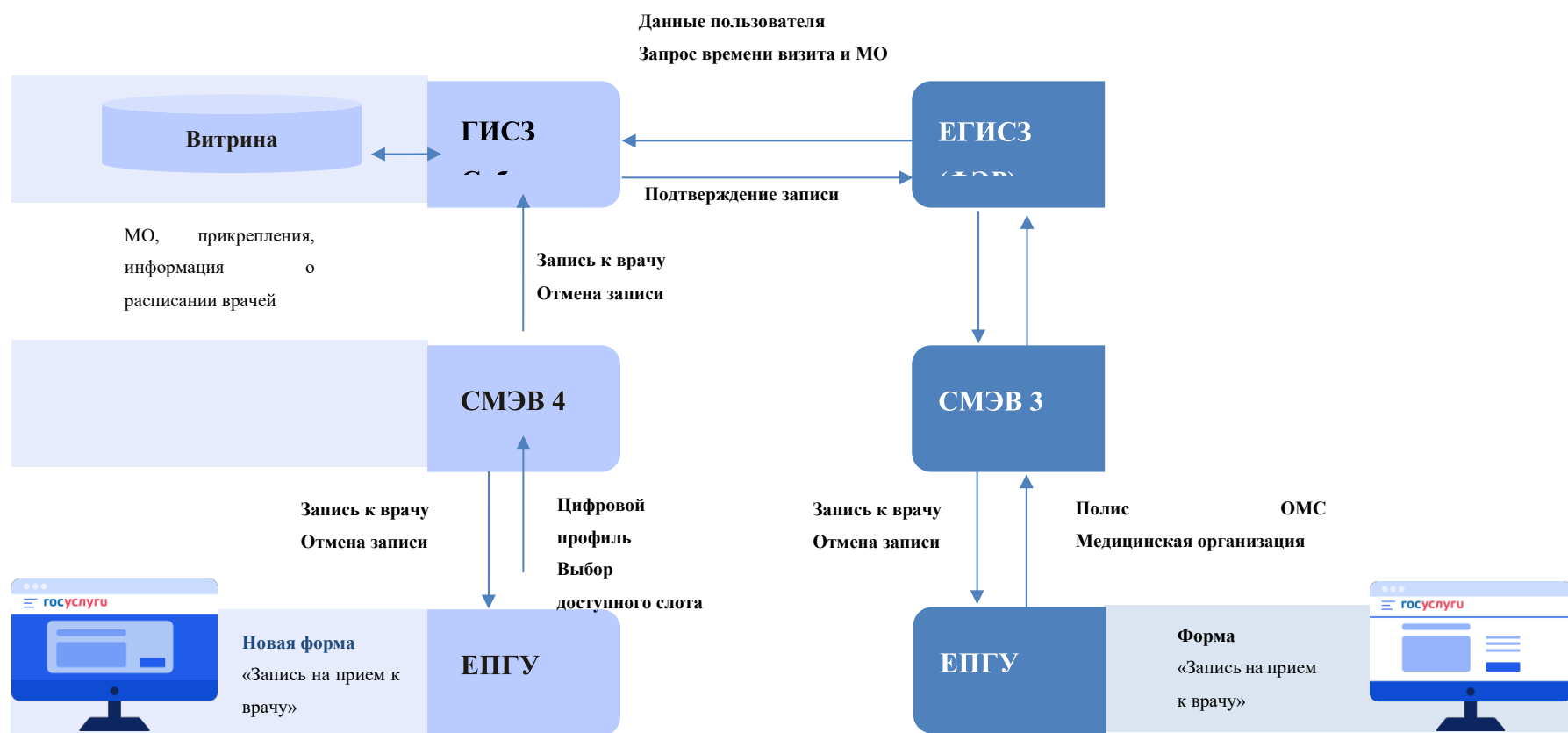


Рисунок 17. Функционирование услуги при обращении к Витрине данных ведомства

Ж.4.5.2 Требования к качеству данных, размещаемых на витринах в виде нормативных значений показателей качества

Целевые значения показателей, определяющих качество данных, размещаемых на витринах, приведены в таблице Таблица 25.

Описание показателей качества функционирования ИС, включая методику расчета их количественных значений, приведено в разделе Ж.4.5.3.

Таблица 25 Целевые значения показателей качества данных, размещаемых в региональных витринах

Показатель качества	Значение показателя ⁶
Доля пациентов в регионе (в РМИС), имеющих прикрепление к медицинской организации и прикрепленных к существующим и актуальным медицинским организациям	> 80%
Среднее значение количества инцидентов недоступности расписаний по определенному перечню специальностей Субъекта РФ за 7 календарных дней	< 5%
Доля слотов в расписании с возможностью для записи на ЕПГУ по отношению к общему объему доступных слотов для записи по всем каналам (в т.ч. очно в регистратуре)	> 80%
Доля ошибочных прикреплений пациентов к медицинским организациям в разрезе пола и возраста	< 10%

Ж.4.5.3 Показатели качества данных, размещаемых на витринах

Ж.4.5.3.1 Доля пациентов в регионе, имеющих прикрепление к медицинской организации и прикрепленных к существующим и актуальным медицинским организациям

Определяется как количество пациентов в витрине данных, имеющих прикрепление к актуальным МО, деленное на общее количество пациентов в регистре пациентов РМИС (Таблица 26).

Значение показателя рассчитывается по следующей формуле:

$$\% \text{ Актуального прикрепления к мед организациям} = 100 * \frac{Q}{Z}$$

где:

Q - количество пациентов с прикреплением к МО в витрине;

Z – общее количество записей о пациентах в витрине;

% Актуальные прикрепления – процентное соотношение Q к Z .

Таблица 26 Расчет доли пациентов региона с актуальным прикреплением к МО

Наименование субъекта	Количество пациентов всего	Количество пациентов с прикреплением к МО	% Актуального прикрепления к МО
Субъект 1	1 000 000	999 999	~ 100

⁶ Значения показателей могут быть скорректированы после получения реальных данных согласно методам, определённым документом

Пример скрипта расчета показателя представлен в приложении Приложение Ж.6.

Ж.4.5.3.2 Доля инцидентов недоступности расписаний по определенному перечню специальностей в соответствии с утвержденными критериями доступности

Значения показателя является результатом расчета, состоящего из следующих шагов вычислений:

Шаг 1 «Расчет доступных врачей специальности для записи через ЕПГУ за период в разрезе медицинских организаций по информации, размещенной на витрине данных»

В каждой медицинской организации с ненулевым количеством прикрепленного населения, а также по каждой врачебной специальности, на основании информации, размещенной в витрине данных, ежедневно выполняется расчет доли слотов, доступных для записи через ЕПГУ в соответствии со списком специальностей, приведенных в приложении Приложение Ж.7 (Таблица 27).

При расчете доступных слотов для врачей, у которых оценка производится на ближайшие сутки, учитываются только будние дни (исключаются часы при выпадении поиска на субботу и воскресенье).

Таким образом, при расчете ближайшего доступного слота в ближайшие 24 часа и начала расчета в пятницу 18:30, время окончания «суточного» периода будет в понедельник в 18:30.

Таблица 27 Расчет доступных врачебных специальностей

Наименование специальности	Наименование МО	сегодня	завтра	С 3 по 14 день
Терапевт участковый 1	Наименование МО 1	нет	да	да
Терапевт участковый 2	Наименование МО 1	нет	нет	нет

Расчет процента доступных врачей специальности в МО происходит по формуле:

$$\% \text{ Доступных врачей специальности в МО} = 100 * \frac{D}{S}$$

D - количество доступных врачей специальности (врач специальности считается доступным если есть хотя бы один свободный слот на ЕПГУ на момент запроса в период времени в соответствии со списком специальностей и временем доступности, приведенных в приложении Приложение Ж.7 (Таблица 28).

S – общее количество врачей соответствующей специальности размещенной в витрине данных.

Таблица 28 Расчет доли доступных врачей специальности в медицинских организациях

Наименование специальности	МО1	МО2	МО3	МО3
Терапевт участковый	% Доступных врачей специальности в МО 1	% Доступных врачей специальности в МО 2	% Доступных врачей специальности в МО 3	% Доступных врачей специальности в МО 4

Пример скрипта расчета показателя представлен в приложении Приложение Ж.8.

Шаг 2 «Расчет факта отсутствия доступных слотов на ЕПГУ в разрезе врачебных специальностей и медицинских организаций по информации, размещенной на витрине данных»

В случае отсутствия свободных слотов на ЕПГУ в медицинской организации по врачебным специальностям выявленных на шаге 1 фиксируется инцидент недоступности в соответствии с критерием:

Фиксируется полная недоступность по специальности в случае отсутствия у всех врачей специальности хотя бы одного доступного слота за указанный период времени.

Шаг 3 «Среднее значение количества инцидентов недоступности расписаний по определенному перечню специальностей Субъекта РФ»

На данном шаге по каждой врачебной специальности в соответствии с приложением Приложение Ж.7 определяются медицинские организации, по которым были зафиксированы инциденты на шаге 2 (Таблица 29).

Значение показателя рассчитывается по следующей формуле:

$$\text{Среднее значение инцидентов по специальности в МО} = \frac{\sum_i P}{i}$$

где:

P – количество зафиксированных инцидентов недоступности по врачебной специальности МО в соответствии с шагом 2;

i – количество календарных дней, когда выполнялся расчёт (например, 7 календарных дней)

Значение показателя рассчитывается по следующей формуле:

$$\text{Среднее значение инцидентов по специальности в Субъекте РФ} = \frac{\sum C}{\sum M}$$

где:

C – среднее значение инцидентов по специальности в МО

M – общее количество МО;

Таблица 29 Расчет среднего значения инцидентов по специальности в субъекте РФ

Наименование субъекта	Наименование специальности	Среднее значение инцидентов по специальности в Субъекте за 7 дней
Субъект 1	Терапевт участковый	

Ж.4.5.3.3 Доля слотов в расписании медицинских организаций с возможностью для записи на ЕПГУ по отношению к общему объему доступных слотов для записи по всем каналам (в т.ч. очно в регистратуре)

Значение показателя рассчитывается в разрезе медицинских организаций, врачебных специальностях в соответствии с приложением Приложение Ж.7 по следующей формуле:

$$\% \text{ доступных слотов, размещаемых для ЕПГУ} = 100 * \frac{X}{Y}$$

где:

X - количество доступных слотов врачебных специальностей в соответствии с приложением Приложение Ж.7, размещенных для записей на ЕПГУ;

Y – количество доступных слотов врачебных специальностей в соответствии с приложением Приложение Ж.7, размещенных для записи по всем источникам записи (в т.ч. регистратура)(Таблица 30).

Таблица 30 Расчет доли слотов для записи на ЕПГУ

Наименование МО	Наименование специальности	% Слотов, размещаемых для ЕПГУ
Наименование МО 1	Терапевт участковый	

Пример скрипта расчета показателя представлен в приложении Приложение Ж.9.

Ж.4.5.3.4 Расчет количества слотов доступных для записи с ЕПГУ на ближайшие 14 дней в разрезе врачебных специальностей и медицинских организаций по размещенной на витрине информации

В ходе данного расчета, по каждой медицинской организации с ненулевым количеством прикрепленного населения, а также по каждой врачебной специальности, представители которой трудоустроены в данной медицинской организации и которая входит в список специальностей, приведенных в приложении Приложение Ж.7, опираясь на данные размещенные в оцениваемой витрине, выполняется расчет количества слотов доступных для записи с ЕПГУ на ближайшие 14 дней.

Пример скрипта расчета показателя для Калужской области представлен в приложении Приложение Ж.10.

В таблице Таблица 31 представлен примерный результат работы скрипта.

Таблица 31 Расчет количества слотов, доступных для записи с ЕПГУ на ближайшие 14 дней

post_id	post_name	med	patients	doctors	Slot count	Epgu slot count	All slot count	all epgu slot count
152	Зубной врач	1.2.643...	60028	11	289	210	438	344
59	Врач-педиатр участковый	1.2.643...	12568	16	533	475	833	770

all_slot_count – общее количество слотов всех типов (врачей указанной специальности в указанной МО) на заданный период времени (14 дней).

slot_count – количество свободных слотов всех типов (врачей указанной специальности в указанной МО) на заданный период времени (14 дней).

all_epgu_slot_count и epgu_slot_count – аналогичный расчет, но для слотов с типом, предназначенным для записи с ЕПГУ.

Ж.4.5.3.5 Доля ошибочных прикреплений пациентов к медицинским организациям в разрезе пола и возраста в конкретном регионе

Шаг 1 «Расчет количества прикреплений пациентов к медицинским организациям в разрезе пола и возраста»

Пример скрипта расчета показателя представлен в приложении Приложение Ж.11.

В таблице Таблица 32 представлен примерный результат работы скрипта

Таблица 32 Расчет количества прикреплений пациентов к МО в разрезе пола и возраста

Наименование МО	0-1 года		1-4 лет		5-17 лет		18-64 лет		65 лет и старше		ИТОГО
	муж.	жен.	муж.	жен.	муж.	жен.	муж.	жен.	муж.	жен.	
МО	347	344	1760	1589	7058	6821	20724	24145	3516	7583	73889

Шаг 2 «Процент отклонений прикреплений пациентов к медицинским организациям в разрезе пола и возраста в сравнении с данными ФОМС»

Определяется показатель как отношение прикреплений, представленных в витрине данных, к данным ФОМС с прикреплениями в МО в разрезе пола и возраста.

На данном этапе выявляются отклонения, которые обнаруживаются в витрине данных в разрезе пола и возраста конкретной МО. К примеру, обнаруживаются прикрепления взрослых в детские поликлиники, или отклонения, связанные с итоговым количественным различием двух витрин данных.

$$\% \text{ Отклонения прикреплений в МО} = 100 - 100 * \frac{Cr}{Cf}$$

Cr – количество прикреплений пациентов к медицинским организациям по данным РМИС

Cf – количество прикреплений пациентов к медицинским организациям по данным ФОМС

Ж.5 Мониторинг показателей и реагирование на их отклонение от нормативных значений для услуги «запись на прием к врачу (на витринах данных)»

Ж.5.1 Соглашение о коммуникациях

Участниками данного процесса являются службы эксплуатации, обеспечивающие функционирование услуги «Записи на прием к врачу»:

- Служба эксплуатации ЕПГУ, обеспечивающая функционирование услуги со стороны ЕПГУ;
- Служба эксплуатации Субъекта РФ, подключенного к услуге «Запись на прием к врачу».

Основным каналом для взаимодействия между участниками является СЦ. Это интернет-портал <https://sc.digital.gov.ru/>, предназначенный для обеспечения коммуникации по вопросам получения и предоставления информационно-методической поддержки Участникам. Операторам СЦ является Минцифры России.

Для взаимодействия по каналу связи через СЦ Службе эксплуатации Субъекта РФ необходимо получить доступ к СЦ. Для этого требуется направить письмо в Минцифры России о необходимости формирования группы из числа ответственных лиц службы эксплуатации Субъекта РФ по услуге «Запись на прием к врачу», включая информацию с контактными данными членов группы (Ф.И.О, должность, организация, ОГРН, телефон, адрес электронной почты).

Служба эксплуатации Субъекта РФ должна самостоятельно определить в своей организации группу лиц, ответственных за осуществление коммуникаций по вопросам контроля и функционирования сервисов, находящихся в их зоне ответственности. После получения доступа и формирования группы в СЦ, Службе эксплуатации Субъекта РФ необходимо будет отслеживать поступающие на их группу запросы, а также самостоятельно заводить запросы в случае такой необходимости.

Дополнительным каналом коммуникации может служить чат в telegram для оперативного взаимодействия и решения аварийной ситуации. Чат создает Служба эксплуатации ЕПГУ и включает ответственных специалистов, указанных в составе группы Субъекта РФ для осуществления коммуникаций по вопросам контроля и функционирования сервисов (Название чата «МVP-ФЭР-ЕПГУ 2.0 «МVP-ФЭР-ЕПГУ 2.0 Техническое Сопровождение»).

Ж.5.2 Мониторинг качества функционирования услуги

Ж.5.2.1 Требования к качеству функционирования услуги в виде нормированных значений показателей качества

Целевые значения показателей, определяющих качество функционирования услуги «Запись на прием к врачу» (на витринах данных), участвующих в информационном обмене с использованием СМЭВ 4, приведены в таблице Таблица 33

Описание показателей качества функционирования услуги «Запись на прием к врачу» (на витринах данных), включая методику их расчета приведено в разделе Ж.5.2.2.

Таблица 33 Показатели качества функционирования услуги «Запись на прием к врачу»

Показатель качества	Значение показателя
Доступность запросов по бизнес-данным из витрины Субъекта РФ	85% и более
Доступность информационного обмена для ЕПГУ	95% и более

Ж.5.2.2 Показатели качества функционирования услуги

Для каждого из показателей качества, включенных в таблице Таблица 33, приведены их описания и методики расчета с учетом используемого способа информационного обмена – СМЭВ 4.

Ж.5.2.2.1 Доступность запросов по бизнес-данным из витрины Субъекта РФ

Доступность запросов по бизнес-данным оценивается исходя из полученных успешных и неуспешных ответов на запросы, отправленных с формы услуги на ЕПГУ в витрины поставщиков данных (Субъекты РФ), по методам, указанным в таблице Таблица 34. На каждый метод настраивается метрика и определяются пороговые значения, при превышении которых создаются события мониторинга. В случае фиксации отрицательного результата создается запрос в СЦ на ответственную команду.

Таблица 34 Оценка полученных успешных и неуспешных ответов на запросы в рамках функционирования услуги «Запись на прием к врачу»

Метод	Описание	Результат	
		Положительный ⁷	Отрицательный
Загрузка формы	Переход на экран https://www.gosuslugi.ru/10700/1/form .	Открыт стартовый экран услуги	Ошибка при открытии
Валидация пациента	Отправлен запрос в витрину для получения идентификатора пользователя (вызов рест <code>/api/hsi/v1/dictionary/ValidatePatient</code> с формы услуги)	Получен id пользователя в витрине или пустой ответ	Получена ошибка
Получение списка специальностей	Отправлен запрос в витрину для получения доступных специалистов (вызов рест <code>/api/hsi/v1/dictionary/Get_Specialist</code> с формы услуги)	Получен список специальностей для записи или пустой ответ	Получена ошибка
Получение списка МО	Отправлен запрос в витрину для получения списка доступных медорганизаций (вызов <code>/api/hsi/v1/dictionary/Get_MO</code> с формы услуги)	Получен список МО или пустой ответ	Получена ошибка
Получение списка направлений	Отправлен запрос в витрину для получения списка выданных направлений (вызов <code>/api/hsi/v1/dictionary/Referral</code> с формы услуги)	Получен список направлений или пустой ответ	Получена ошибка

⁷ Получение пустого ответа или ошибки бронирования считается отрицательным результатом в случае массовости ошибок за определенный период времени

Метод	Описание	Результат	
		Положительный ⁷	Отрицательный
Получение списка доступных слотов по специальности	Отправлен запрос в витрину для получения списка слотов (вызов рест <code>/api/hsi/v1/dictionary/GetCountFreeSlot</code> с формы услуги).	Получен список слотов по выбранной специальности для записи или пустой ответ	Получена ошибка
Запрос на запись к врачу	Отправлен запрос на запись к врачу на выбранное время (вызов рест <code>/api/lk/v1/equeue/agg/book</code> с формы услуги)	Получен id бронирования записи или ошибка 12, 13, 17	Получена ошибка при бронировании
Отмена записи к врачу	Отправлен запрос на отмену записи к врачу (вызов рест <code>/api/lk/v1/equeue/agg/cancel</code> с формы услуги или <code>/api/lk/v1/calendar/events/{extId}/cancel</code> из Ленты)	Запись отмена	Получена ошибка при отмене

Метрики собираются в разрезе региона (параметр `region_code`). Справочник регионов представлен в приложении Приложение Ж.12.

Загрузка формы услуги «Запись на прием к врачу»

Оценивается успешная и неуспешная загрузка формы услуги «Запись на прием к врачу» в процентном соотношении (%) ко всем результатам события.

Успешная загрузка рассчитывается по формуле:

$$event = service_init \text{ AND } data_result = 0$$

Неуспешная загрузка рассчитывается по формуле:

$$event = service_init \text{ AND } data_result = 1$$

где:

`event = service_init` – событие по загрузке формы услуги. Переход на экран [https://www.gosuslugi.ru/10700/1/form](https://www.gosuslugi.ru/10700/1/form;);

`data_result` – результат загрузки формы.

Валидация пользователя

Показатель, который показывает процент ошибок по методу валидации пациента в витрине данных. Оцениваются успешные запросы, организационные и технические ошибки в процентах (%) ко всем результатам события.

Успешные запросы рассчитываются по формуле:

$$event = validate_patient \text{ AND } data_result = 0$$

Организационные ошибки рассчитываются по формуле:

$$event = validate_patient \text{ AND } data_result = 7$$

Технические ошибки рассчитываются по формуле:

$$event = validate_patient \text{ AND } data_result = 3 \text{ OR } data_result \geq 400$$

где:

event = validate_patient – событие по получению данных по валидации пациента. Отправлен запрос в витрину для получения идентификатора пользователя (вызов рест */api/hsi/v1/dictionary/ValidatePatient*);

data_result – код ошибки, полученный в ответе на запрос.

Получение списка специальностей

Оцениваются успешные запросы, организационные и технические ошибки в процентах (%) ко всем результатам события при получении списка специальностей врачей, к которым может записаться пользователь.

Успешные запросы рассчитываются по формуле:

$$event = get_specialist \text{ AND } data_result = 0$$

Организационные ошибки рассчитываются по формуле:

$$event = get_specialist \text{ AND } data_result = 7$$

Технические ошибки рассчитываются по формуле:

$$event = get_specialist \text{ AND } data_result = 3 \text{ OR } data_result \geq 400$$

где:

event = get_specialist – событие по получению данных по специальности. Отправлен запрос в витрину для получения доступных специалистов (вызов рест */api/hsi/v1/dictionary/Get_Specialist*);
data_result – код ошибки, полученный в ответе на запрос.

Получение списка МО

Оцениваются успешные запросы, организационные и технические ошибки в процентах (%) ко всем результатам события при получении списка МО, в которые может записаться пользователь.

Успешные запросы рассчитываются по формуле:

$$event = get_mo \text{ AND } data_result = 0$$

Организационные ошибки рассчитываются по формуле:

$$event = get_mo \text{ AND } data_result = 7$$

Технические ошибки рассчитываются по формуле:

$$event = get_mo \text{ AND } data_result = 3 \text{ OR } data_result \geq 400$$

где:

event = get_MO – событие по получению данных по медорганизациям. Отправлен запрос в витрину для получения списка доступных медорганизаций (вызов */api/hsi/v1/dictionary/Get_MO*);
data_result – код ошибки, полученный в ответе на запрос.

Получение списка направлений

Оцениваются успешные запросы, организационные и технические ошибки в процентах (%) ко всем результатам события при получении списка направлений для пациента.

Успешные запросы рассчитываются по формуле:

$$event = get_refferal \text{ AND } data_result = 0 \text{ OR } data_result = 7$$

Технические ошибки рассчитываются по формуле:

$$event = get_refferal \text{ AND } data_result = 3 \text{ OR } data_result \geq 400$$

где:

event = *get_refferal* – событие по получению данных по направлениям. Отправлен запрос в витрину для получения списка выданных направлений (*вызов /api/nsi/v1/dictionary/Referral*);
data_result – код ошибки, полученный в ответе на запрос.

Получение списка доступных слотов по специальности

Оцениваются успешные запросы, организационные и технические ошибки в процентах (%) ко всем результатам события при получении перечня доступных слотов для услуги «Записи на прием к врачу».

Успешные запросы рассчитываются по формуле:

$$event = get_slots \text{ AND } data_result = 0$$

Организационные ошибки рассчитываются по формуле:

$$event = get_slots \text{ AND } data_result = 7$$

Технические ошибки рассчитываются по формуле:

$$event = get_slots \text{ AND } data_result = 3 \text{ OR } data_result \geq 400$$

где:

event = *get_slots* – событие по получению списка доступных слотов по специальности. Отправлен запрос в витрину для получения списка слотов (*вызов рест /api/nsi/v1/dictionary/GetCountFreeSlot*);
data_result – код ошибки, полученный в ответе на запрос.

Запрос на запись к врачу

Оцениваются успешные запросы, организационные и технические ошибки в процентах (%) ко всем результатам события при записи на выбранное время, на которое может записаться пользователь к врачу. Рассчитываются по формулам:

Успешные запросы рассчитываются по формуле:

$$event = request_booking \text{ AND } data_result = 0 \text{ OR } data_result = 12 \text{ OR } data_result = 13 \text{ OR } data_result = 17$$

Организационные ошибки рассчитываются по формуле:

$$event = request_booking \text{ AND } data_result = 11 \text{ OR } data_result = 14 \text{ OR } data_result = 15$$

Технические ошибки рассчитываются по формуле:

$$event = request_booking \text{ AND } data_result = 18 \text{ OR } data_result = 101 \text{ OR } data_result \geq 400$$

где:

event = request_booking – событие по получению данных по бронированию слота. Отправлен запрос на запись к врачу на выбранное время (вызов рест *api/lk/v1/queue/agg/book*);
data_result – код ошибки, полученный в ответе на запрос.

Отмена записи к врачу

Оцениваются успешные запросы и технические ошибки в процентах (%) ко всем результатам события при отмене записи к врачу на выбранную дату и время.

Успешные запросы рассчитываются по формуле:

$$event = request_cancel \text{ AND } data_result = -1$$

Технические ошибки рассчитываются по формуле:

$$event = request_cancel \text{ AND } data_result \geq 0$$

где:

event = request_cancel – событие по получению данных по отмене записи на прием. Отправлен запрос на отмену записи к врачу (вызов рест */api/lk/v1/queue/agg/cancel* с формы услуги или */api/lk/v1/calendar/events/{extId}/cancel* из Ленты);

data_result – код ошибки, полученный в ответе на запрос.

Ж.5.2.2.2 Доступность информационного обмена для ЕПГУ

Определение доступности витрины данных поставщика для услуги «Запись к врачу» в ПОДД

Доступность витрины определяется путем отправки регламентированного запроса со стороны ядра ПОДД, которое представляет собой централизованную часть ПОДД, предназначенную для маршрутизации запросов и контроля полномочий участников взаимодействия, в витрину поставщика данных. Запрос отправляется с периодичностью 1 раз в минуту. После 5 неудачных попыток подряд формируется сообщение в мониторинге. При получении события мониторинга Служба эксплуатации ЕПГУ информирует Службу эксплуатации Субъекта РФ о недоступности витрины поставщика и необходимости исправить ошибку.

Регламентированный запрос имеет следующий вид:

```
SELECT 1 AS res FROM misdmXX.getpatientbyoms_snils_lastname ('norm_lastname', 'snils',
'oms_number') UNION SELECT 1 AS res limit 1
```

где:

misdmXX – мнемоника витрины поставщика, в которую отправлен запрос.

getpatientbyoms_snils_lastname – метод валидации пациента по фамилии, номеру полиса и СНИЛС.

('norm_lastname', 'snils', 'oms_number') – данные пользователя, по которому отправлен запрос: фамилия, номер полиса, СНИЛС

При успешном выполнении запроса в ответ ожидается получить http код 200.

Ж.5.2.3 Мероприятия при сохранении функционирования информационных систем

Ж.5.2.3.1 Правила обработки запросов в СЦ между участниками

В рамках данного регламента Участникам взаимодействия предлагается использовать типы запросов, предусмотренных в СЦ: инцидент – любое событие, не являющееся частью нормальной работы услуги/сервиса, ведущее/ способное привести к остановке услуги или снижению ее качества.

Инцидент заводится в случае наступления события мониторинга по показателям качества функционирования услуги. На основе показателей качества, указанных в разделе Ж.5.2.2, заведены метрики мониторинга. Обновление метрик происходит каждую минуту за предыдущую минуту. При превышении пороговых значений данных метрик срабатывает событие мониторинга. В зависимости от типа метрики и уровня порога может заводится Инцидент и/или отправляться только оповещение в чат «[MVP-ФЭР-ЕПГУ 2.0] мониторинг». Также в зависимости от уровня порога Инциденты могут быть блокирующими или не блокирующими работу услуги «Запись на прием к врачу» (Таблица 35).

Для визуального представления событий в системе мониторинга требуется отображать графики с названием по следующей маске: Минздрав.%Номер шага%.%Название шага мониторинга%.%Регион%.

Таблица 35 Пороговые значения по событиям мониторинга услуги «Запись на прием к врачу»

Название шага мониторинга	Номер шага	Наименование события	Показатель	Уровень порога	Интервал проведения проверок	Заведение инцидента	Канал уведомления	Типы ошибок	Зона ответственности
Загрузка формы	-	clickhouse_zdrav_service_init	Доля ошибок при открытии стартового экрана	1. Если больше 15% неуспешных загрузок; 2. Если больше 50% неуспешных загрузок; 3. Если 0 успешных загрузок.	Каждые 15 минут.	-	Оповещение в чат [MVP-ФЭР-ЕПГУ 2.0] мониторинг	result = 1	Служба эксплуатации ЕПГУ

Название шага мониторинга	Номер шага	Наименование события	Показатель	Уровень порога	Интервал проведения проверок	Заведение инцидента	Канал уведомления	Типы ошибок	Зона ответственности
Валидация пациента	Шаг №1	clickhouse_zdrav_validate_patient	Доля технических ошибок	См. Методика мониторинга витрин по техническим ошибкам	См. Методика мониторинга витрин по техническим ошибкам	Заведение Инцидента 3 приоритета в СЦ	Оповещение в чаты: – [MVP-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» – [MVP-ФЭР-ЕПГУ 2.0] Техническое сопровождение	4xx или 5xx - недоступна компонента NSI (ЕПГУ). 3 - получена ошибка от витрины данных	Служба эксплуатации ЕПГУ Служба эксплуатации Субъекта РФ
			Доля организационных ошибок	См. Методика мониторинга витрин по организационным ошибкам	См. Методика мониторинга витрин по организационным ошибкам	Заведение Инцидента 3 приоритета в СЦ	Оповещение в чаты: – [MVP-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» – [MVP-ФЭР-ЕПГУ 2.0] Техническое сопровождение	7 – получен пустой ответ	Служба эксплуатации Субъекта РФ
Получение списка специально стей	Шаг №2	clickhouse_zdrav_get_specialist	Доля технических ошибок	См. Методика мониторинга витрин по техническим ошибкам	См. Методика мониторинга витрин по техническим ошибкам	Заведение Инцидента 3 приоритета в СЦ	Оповещение в чаты: – [MVP-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» – [MVP-ФЭР-ЕПГУ 2.0] Техническое сопровождение	4xx или 5xx - недоступна компонента NSI (ЕПГУ). 3 - получена ошибка от витрины данных	Служба эксплуатации ЕПГУ Служба эксплуатации Субъекта РФ
			Доля организационных ошибок	См. Методика мониторинга	См. Методика мониторинга витрин по	Заведение Инцидента 3	Оповещение в чаты:	7 – получен пустой ответ	Служба эксплуатации Субъекта РФ

Название шага мониторинга	Номер шага	Наименование события	Показатель	Уровень порога	Интервал проведения проверок	Заведение инцидента	Канал уведомления	Типы ошибок	Зона ответственности
				витрин по организационным ошибкам	организационным ошибкам	приоритета в СЦ	– [MVP-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» – [MVP-ФЭР-ЕПГУ 2.0] Техническое сопровождение		
Получение списка доступных слотов по специально сти	Шаг №3	clickhouse_zdrav_get_slots	Доля технических ошибок	См. Методика мониторинга витрин по техническим ошибкам	См. Методика мониторинга витрин по техническим ошибкам	Заведение Инцидента 3 приоритета в СЦ	Оповещение в чаты: – [MVP-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» – [MVP-ФЭР-ЕПГУ 2.0] Техническое сопровождение	4xx или 5xx - недоступна компонента NSI (ЕПГУ)	Служба эксплуатации ЕПГУ
								3 - получена ошибка от витрины данных	Служба эксплуатации Субъекта РФ
			Доля организационных ошибок	См. Методика мониторинга витрин по организационным ошибкам	См. Методика мониторинга витрин по организационным ошибкам	Заведение Инцидента 3 приоритета в СЦ	Оповещение в чаты: – [MVP-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» – [MVP-ФЭР-ЕПГУ 2.0] Техническое сопровождение	7 – получен пустой ответ	Служба эксплуатации Субъекта РФ
Получение списка МО	-	clickhouse_zdrav_get_mo	Доля технических ошибок	См. Методика мониторинга витрин по	См. Методика мониторинга витрин по	Заведение Инцидента 3 приоритета в СЦ	Оповещение в чаты:	4xx или 5xx - недоступна компонента NSI (ЕПГУ)	Служба эксплуатации ЕПГУ

Название шага мониторинга	Номер шага	Наименование события	Показатель	Уровень порога	Интервал проведения проверок	Заведение инцидента	Канал уведомления	Типы ошибок	Зона ответственности
				техническим ошибкам	техническим ошибкам		– [MVP-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» – [MVP-ФЭР-ЕПГУ 2.0] Техническое сопровождение	3 - получена ошибка от витрины данных	Служба эксплуатации Субъекта РФ
			Доля организационных ошибок	См. Методика мониторинга витрин по организационным ошибкам	См. Методика мониторинга витрин по организационным ошибкам	Заведение Инцидента 3 приоритета в СЦ	Оповещение в чаты: – [MVP-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» – [MVP-ФЭР-ЕПГУ 2.0] Техническое сопровождение	7 – получен пустой ответ	Служба эксплуатации Субъекта РФ
Получение списка направлений	-	clickhouse_zdrav_get_refferal	Доля технических ошибок	См. Методика мониторинга витрин по техническим ошибкам	См. Методика мониторинга витрин по техническим ошибкам	Заведение Инцидента 3 приоритета в СЦ	Оповещение в чаты: – [MVP-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» – [MVP-ФЭР-ЕПГУ 2.0] Техническое сопровождение	4xx или 5xx - недоступна компонента NSI (ЕПГУ)	Служба эксплуатации ЕПГУ
								3 - получена ошибка от витрины данных	Служба эксплуатации Субъекта РФ
Запрос на запись врачу	Шаг №4	clickhouse_zdrav_request_booking	Доля технических ошибок	См. Методика мониторинга витрин по	См. Методика мониторинга витрин по	Заведение Инцидента 3 приоритета в СЦ	Оповещение в чаты:	4xx или 5xx - недоступна компонента LK (ЕПГУ).	Служба эксплуатации ЕПГУ

Название шага мониторинга	Номер шага	Наименование события	Показатель	Уровень порога	Интервал проведения проверок	Заведение инцидента	Канал уведомления	Типы ошибок	Зона ответственности
				техническим ошибкам	техническим ошибкам		– [MVP-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» – [MVP-ФЭР-ЕПГУ 2.0] Техническое сопровождение	101 - получена ошибка от агента ПОДД.	Служба эксплуатации ЕПГУ
								18 - внутренняя ошибка РМИС	Служба эксплуатации Субъекта РФ
			Доля организационных ошибок	См. Методика мониторинга витрин по организационным ошибкам	См. Методика мониторинга витрин по организационным ошибкам	Заведение Инцидента 3 приоритета в СЦ	Оповещение в чаты: – [MVP-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» – [MVP-ФЭР-ЕПГУ 2.0] Техническое сопровождение	11, 14, 15 – организационная ошибка от РМИС.	Служба эксплуатации Субъекта РФ
Отмена записи к врачу	-	clickhouse_zdrav_request_cancel	Доля технических ошибок	См. Методика мониторинга витрин по техническим ошибкам	См. Методика мониторинга витрин по техническим ошибкам	Заведение Инцидента 3 приоритета в СЦ	Оповещение в чаты: – [MVP-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» – [MVP-ФЭР-ЕПГУ 2.0] Техническое сопровождение	4xx или 5xx - недоступна компонента LK (ЕПГУ).	Служба эксплуатации ЕПГУ
								101 - получена ошибка от агента ПОДД.	Службой эксплуатации ЕПГУ
								0 - ошибка РМИС.	Службой эксплуатации Субъекта РФ
Доступность витрин для услуги "Запись к	-	-	Количество неудачных попыток	5 неудачных попыток подряд	Раз в минуту отправляются регламентированные запросы в	Если получено сообщение в чате telegram о том, что	Оповещение в чат [MVP-ФЭР-ЕПГУ 2.0] мониторинг		Служба эксплуатации Субъекта РФ

Название шага мониторинга	Номер шага	Наименование события	Показатель	Уровень порога	Интервал проведения проверок	Заведение инцидента	Канал уведомления	Типы ошибок	Зона ответственности
врачу" в ПОДД					витрины поставщика (Субъекты РФ)	витрина Субъекта РФ не отвечает, Службе эксплуатации Субъекта РФ необходимо проверить работоспособность агента-витрины. Если на момент проверки агент и витрина в Субъекте РФ работают стабильно, необходимо запросить дополнительное подтверждение у Службы эксплуатации ЕПУ о доступности витрины через мониторинг, дождаться и провести анализ работы витрины за промежуток времени, в который			

Название шага мониторинга	Номер шага	Наименование события	Показатель	Уровень порога	Интервал проведения проверок	Заведение инцидента	Канал уведомления	Типы ошибок	Зона ответственности
						приходило оповещение о недоступности			

Методика мониторинга витрин по техническим ошибкам

1. Интервал проведения проверок каждые 15 минут.
2. Триггеры настраиваются для шагов записи к врачу:
 - валидация пациента;
 - получение списка специальностей;
 - получение списка доступных слотов по специальности;
 - получение списка МО;
 - получение списка направлений;
 - запрос на запись к врачу;
 - отмена записи к врачу.

2.1. Если % количества неуспешных запросов к общему числу запросов $\geq 1\%$ и количество неуспешных запросов ≥ 5 , тогда услуга деактивируется, заводится ведомственный инцидент.

2.2. В остальных случаях услуга не деактивируется, ведомственный инцидент не заводится.

Методика мониторинга витрин по организационным ошибкам⁸

1. Интервал проведения проверок каждые 30 минут.
2. Триггеры настраиваются для шагов записи к врачу:
 - валидация пациента;
 - получение списка специальностей;
 - получение списка доступных слотов по специальности;
 - получение списка МО;
 - запрос на запись к врачу.

2.1. Если % количества неуспешных запросов к общему числу запросов $\geq 5\%$ и количество неуспешных запросов ≥ 5 , тогда услуга не деактивируется, заводится ведомственный инцидент.

2.2. В остальных случаях услуга не деактивируется, ведомственный инцидент не заводится.

Порядок действий службы эксплуатации ЕПГУ при срабатывании события мониторинга:

1. Определить тип метрики (Таблица 35) по которой создалось событие монитора. В зависимости от типа метрики:

– Оповестить в чате Оповещение в чат «[MVP-ФЭР-ЕПГУ 2.0] мониторинг» и/или завести Инцидент в СЦ.

2. Завести неблокирующий Инцидент

2.1 Завести запрос в СЦ с типом «Инцидент» 3 приоритета на ответственную группу службы эксплуатации Субъекта РФ. Текст шаблона:

⁸ Уровень порога и интервал проверок может быть скорректирован по итогу получения реальных данных

*Тема: Ведомственная ИС. Минздрав. "Витрина Минздрава" - (Наименование Субъекта РФ). Отсутствуют успешные ответы на шаге *****.*

Описание: Добрый день.

*Средствами мониторинга зафиксировано отсутствие успешных ответов на шаге ***** от витрины Субъекта РФ.*

Данная ситуация влияет на услугу ЕПГУ

«Запись на прием к врачу» <https://www.gosuslugi.ru/10700>

Просьба восстановить штатный режим работы витрины.

К инциденту прикладываются файлы с ошибкой из п. 2.3 Обеспечение качества данных, раздел План реакции.

2.2 Осуществить информирование ответственного за услугу «Запись на прием к врачу» со стороны ЕПГУ об обнаружении нештатной ситуации.

2.3 Осуществить информирование Службы эксплуатации Субъекта РФ в чате telegram «MVP-ФЭР-ЕПГУ 2.0 Техническое Сопровождение».

3 Завести блокирующий Инцидент.

3.1 Завести запрос с типом «Инцидент» 3 приоритета на ответственную группу Службы эксплуатации Субъекта РФ через СЦ:

*Тема: Ведомственная ИС. Минздрав. "Витрина Минздрава" - (Наименование Субъекта РФ). Отсутствуют успешные ответы на шаге *****.*

Описание: Добрый день.

*Средствами мониторинга зафиксировано отсутствие успешных ответов на шаге ***** от витрины Субъекта РФ.*

Данная ситуация влияет на услугу ЕПГУ

«Запись на прием к врачу» <https://www.gosuslugi.ru/10700>

Просьба восстановить штатный режим работы витрины.

К инциденту прикладываются файлы с ошибкой из п. 2.3 Обеспечение качества данных, раздел План реакции.

3.2 Деактивировать услугу «Запись на прием к врачу» в Субъекте РФ <https://www.gosuslugi.ru/10700>.

3.3 Осуществить информирование ответственного за услугу «Запись на прием к врачу» со стороны ЕПГУ об обнаружении аварийной ситуации и отключении Субъекта РФ от услуги.

3.4 Осуществить информирование Службы эксплуатации Субъекта РФ в чате telegram «MVP-ФЭР-ЕПГУ 2.0 Техническое Сопровождение»

При получении такого Инцидента Служба эксплуатация Субъекта РФ выполняет следующие действия:

1. Принимает Инцидент в работу в СЦ и приступает к анализу проблемы.
2. В случае, если проблема на стороне Субъекта РФ:

2.1 Выполняет необходимые действия и/или проводит работы для решения проблемы.

2.2 После решения проблемы и проверки работоспособности закрывает Инцидент в СЦ.

2.3 Информировывает Службу эксплуатации ЕПГУ в чате telegram «MVP-ФЭР-ЕПГУ 2.0 название Субъекта РФ» о решении Инцидента и восстановлении работоспособности.

3. В случае, если проблему не удалось решить силами Службы эксплуатации Субъекта РФ:

3.1 Запрашивает дополнительную информацию от Службы эксплуатации ЕПГУ в Инциденте и чате telegram «MVP-ФЭР-ЕПГУ 2.0 Техническое Сопровождение».

3.2 Если информации недостаточно или она не способствует решению проблемы допускается обсуждение проблемы в чате telegram «MVP-ФЭР-ЕПГУ 2.0 Техническое Сопровождение».

3.3 После решения проблемы и проверки работоспособности закрывает Инцидент в СЦ.

3.4 Информировывает Службу эксплуатации ЕПГУ в чате telegram «MVP-ФЭР-ЕПГУ 2.0 Техническое Сопровождение» о решении Инцидента и восстановлении работоспособности.

При решении Инцидента Службой эксплуатации Субъекта РФ Служба эксплуатации ЕПГУ осуществляет следующие действия:

1. Активирует услугу «Запись на прием к врачу» в Субъекте РФ <https://www.gosuslugi.ru/10700>, в случае если Инцидент был блокирующим.

2. Проверяет доступность и функционирование услуги в Субъекте РФ.

3. В случае успешной проверки подтверждает выполнение Инцидента Службой эксплуатации Субъекта РФ в СЦ, а также информирует в чате telegram «MVP-ФЭР-ЕПГУ 2.0 Техническое Сопровождение».

4. В случае неуспешной проверки Инцидент возвращается в работу через СЦ Службе эксплуатации Субъекта РФ до ее устранения и информирует в чате telegram «MVP-ФЭР-ЕПГУ 2.0 Техническое Сопровождение». Деактивирует услугу «Запись на прием к врачу» в Субъекте РФ <https://www.gosuslugi.ru/10700> в случае, если Инцидент был блокирующим.

В случае необходимости Служба эксплуатации Субъекта РФ может самостоятельно завести Инцидент при возникновении аварийной ситуации на своей стороне или обнаружении нестабильной работы ПОДД путем направления запроса с использованием Личного кабинета СЦ <https://sc.digital.gov.ru/>.

В случае возникновения ошибок на ПО «Витрины данных» необходимо оформить запрос в СЦ следующего типа:

- соглашение/услуга – Инциденты ИС ИЭП;
- тип запроса – Запрос/инцидент;
- система ИЭП – ПО «Витрина данных».

В теме запроса указать возникшую проблему. В описании подробно описать проблему.

План реакции на возникающие проблемы.

В описании должна быть предоставлена следующая информация для анализа:

1. Конфигурация витрины данных (Лайт/Медиум/Стандарт); отдельно указать версию витрины.
2. Приложить выгрузку из мониторинга: Системные ресурсы сервера (CPU, RAM, HDD)
3. Приложить выгрузку из мониторинга: Утилизация сервера (результат команды `top` или `htop`)
4. Проверить сетевую связанности компонентов витрины, прислать подтверждение.
5. Проверить портов компонентов витрины, агента ПОДД, прислать подтверждение.
6. Приложить информацию с названием и версией компонентов витрины с ошибкой.
7. Приложить конфигурационные файлы затронутых компонентов (`application.yml` и/или содержимое `ENV` компонента из `Portainer`)
8. Приложить логи компонента с ошибкой
9. Сделать описание проблемы, при которой возникает ошибка, с обязательным указанием даты и времени ошибки.

В случае возникновения ошибок в процессе работы Агента ПОДД СМЭВ/ПОДД СМЭВ необходимо оформить запрос в СЦ следующего типа:

- соглашение/услуга – Инциденты ИС ИЭП;
- тип запроса – Запрос/инцидент;
- система ИЭП – ПОДД.

В случае возникновения ошибок в процессе работы ЕПГУ необходимо оформить запрос в СЦ следующего типа:

- соглашение/услуга – Инциденты ИС ИЭП;
- тип запроса – Запрос/инцидент;
- система ИЭП – ЕПГУ.

В теме запроса указать возникшую проблему. В описании подробно описать проблему.

В случае проведения технических работ на витрине данных Службе эксплуатации Субъекта РФ необходимо оповестить заинтересованных сторон по электронной почте с указанием даты и периода недоступности витрины. Список рассылки представлен в таблице Таблица 36. Допускается дополнительно информировать о проведении технических работ в чате telegram «MVP-ФЭР-ЕПГУ 2.0 Техническое Сопровождение».

Таблица 36 Список рассылки для информирования

Наименование службы	Электронная почта
Служба эксплуатации услуги ЕПГУ	support-egov@rtlabs.ru support-podd@rtlabs.ru
Ответственный от Минцифры России	e.volev@digital.gov.ru
Ответственный от ЕПГУ	maxim.sumskoy@rtlabs.ru

Ж.5.2.3.2 Уровни приоритета решения запросов

На время подключения всех Субъектов РФ к ПОДД и подключения им услуги «Запись на прием к врачу» Инциденту всегда присваивается уровень приоритета – 3. После

подключения всех Субъектов РФ к ПОДД, будут добавлены другие уровни приоритета запроса (1, 2) с последующей актуализацией данного раздела.

Время решения Инцидентов зависит от уровня приоритета, типа запроса, от того, находится ли витрина Субъект РФ после подключения к ПОДД в опытной или промышленной эксплуатации, а также предоставлен ли доступ к услуге «Запись на прием к врачу» в продуктивной среде (Таблица 37)

Продолжительность опытной эксплуатации – 3 месяца со дня открытия услуги «Запись на прием к врачу» в продуктивной среде на ЕПГУ для Субъекта РФ. Начиная с 4 месяца Субъект РФ переводится в промышленную эксплуатацию.

Таблица 37 Время решения запросов

Уровень Приоритета	Тип запроса	Время решения запроса с момента создания	
		Опытная эксплуатация	Промышленная эксплуатация
3	Инцидент блокирующий	96 часов	24 часа
3	Инцидент не блокирующий	120 часов	72 часа

Ж.5.2.3.3 Проверка работоспособности услуги в Субъекте РФ

Для проверки работоспособности услуги «Запись на прием к врачу» в «Субъекте РФ» службой эксплуатации ЕПГУ будет заведен тестовый пользователь в продуктивной среде ЕПГУ. Тестовые данные пользователя будут переданы Службе эксплуатации Субъекта РФ для прикрепления тестового пользователя к медицинской организации Субъекта РФ и заведения в продуктивной версии Витрины Субъекта РФ. Медицинскую организацию следует выбирать исходя из наибольшего количество доступных слотов для записи через ЕПГУ.

Ж.5.2.3.4 Предоставление Отчетности

По запросу Минцифры России может быть предоставлен отчет о количестве Субъектов РФ, подключенных и/или отключенных от услуги «Запись на прием к врачу». По отдельному запросу может быть предоставлен отчет о выполнении Инцидентов Службой эксплуатации Субъекта РФ.

Ж.5.3 Мониторинг качества данных, размещаемых в витринах субъекта РФ

Ж.5.3.1 Мероприятия при сохранении функционирования информационных систем

Ж.5.3.1.1 Правила обработки запросов в СЦ между участниками

Пороговые значения представлены в Таблица 38.

Таблица 38 Пороговые значения по показателям качества данных

Показатель качества	Значение показателя ⁹	Интервал проведения проверок	Канал уведомления
Доля пациентов в регионе (в РМИС), имеющих прикрепление к медицинской организации и прикрепленных к существующим и актуальным медицинским организациям	> 80%	Ежедневно в 18:30	Оповещение в чаты: – [МВР-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» – [МВР-ФЭР-ЕПГУ 2.0] Техническое сопровождение
Среднее значение количества инцидентов недоступности расписаний по определенному перечню специальностей Субъекта РФ за 7 календарных дней	< 5%	Ежедневно в 18:30	Оповещение в чаты: – [МВР-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» [МВР-ФЭР-ЕПГУ 2.0] Техническое сопровождение
Доля слотов в расписании с возможностью для записи на ЕПГУ по отношению к общему объему доступных слотов для записи по всем каналам (в т.ч. очно в регистратуре)	> 80%	Ежедневно в 18:30	Оповещение в чаты: – [МВР-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» [МВР-ФЭР-ЕПГУ 2.0] Техническое сопровождение
Доля ошибочных прикреплений пациентов к медицинским организациям в разрезе пола и возраста	< 10%	Ежедневно в 18:30	Оповещение в чаты: – [МВР-ФЭР-ЕПГУ 2.0] мониторинг – «ИНЦИДЕНТЫ» [МВР-ФЭР-ЕПГУ 2.0] Техническое сопровождение

Порядок действий службы эксплуатации ЕПГУ при срабатывании события мониторинга:

⁹ Значения показателей могут быть скорректированы после получения реальных данных согласно методам, определенным документом

1. Определить тип метрики Таблица 38 по которой создано событие мониторинга. Оповестить в чате Оповещение в чат «[MVP-ФЭР-ЕПГУ 2.0] мониторинг».
2. Завести неблокирующий Инцидент.
3. Завести запрос в СЦ с типом «Инцидент» 3 приоритета на ответственную группу службы эксплуатации Субъекта РФ. Текст шаблона:

Тема: Ведомственная ИС. Минздрав. "Витрина Минздрава" - (Наименование Субъекта РФ). Показатель качества данных, размещаемых в региональных витринах ниже целевого значения.

Описание: Добрый день.

Средствами мониторинга зафиксировано несоответствие показателя качества данных, размещаемых в витрине («Наименование показателя»), ниже целевого значения Данная ситуация влияет на услугу на ЕПГУ

«Запись на прием к врачу» <https://www.gosuslugi.ru/10700>

Просьба провести работы по повышению качества данных витрины.

К Инциденту прикладываются отчет с оценкой качества данных.

4. Осуществить информирование ответственного за услугу «Запись на прием к врачу» со стороны ЕПГУ об обнаружении отклонения показателя от целевого.
5. Осуществить информирование Службы эксплуатации Субъекта РФ в чате telegram «MVP-ФЭР-ЕПГУ 2.0 Техническое Сопровождение».

При получении такого Инцидента Служба эксплуатация Субъекта РФ выполняет следующие действия:

1. Принимает Инцидент в работу в СЦ и приступает к анализу проблемы.
2. Выполняет необходимые действия и/или проводит работы для решения проблемы.
3. После решения проблемы и проверки повышения качества данных закрывает Инцидент в СЦ.
4. Информировует Службу эксплуатации ЕПГУ в чате telegram «MVP-ФЭР-ЕПГУ 2.0 название Субъекта РФ» о решении Инцидента и повышении качества данных.

При решении Инцидента Службой эксплуатации Субъекта РФ Служба эксплуатации ЕПГУ осуществляет следующие действия:

1. Проверяет превышение показателем целевого значения в Субъекте РФ.
2. В случае успешной проверки подтверждает выполнение Инцидента Службой эксплуатации Субъекта РФ в СЦ, а также информирует в чате telegram «MVP-ФЭР-ЕПГУ 2.0 Техническое Сопровождение».
3. В случае неуспешной проверки Инцидент возвращается в работу через СЦ Службе эксплуатации Субъекта РФ до ее устранения и информирует в чате telegram «MVP-ФЭР-ЕПГУ 2.0 Техническое Сопровождение».

В случае возникновения ошибок на ПО «Витрины данных» необходимо оформить запрос в СЦ следующего типа:

- соглашение/услуга – Инциденты ИС ИЭП;
- тип запроса – Запрос/инцидент;
- система ИЭП – ПО «Витрина данных».

В теме запроса указать возникшую проблему. В описании подробно описать проблему.

План реакции на возникающие проблемы.

В описании должна быть предоставлена следующая информация для анализа:

1. Конфигурация витрины данных (Лайт/Медиум/Стандарт); отдельно указать версию витрины.
2. Приложить выгрузку из мониторинга: Системные ресурсы сервера (CPU, RAM, HDD)
3. Приложить выгрузку из мониторинга: Утилизация сервера (результат команды `top` или `htop`)
4. Проверить сетевую связанности компонентов витрины, прислать подтверждение.
5. Проверить портов компонентов витрины, агента ПОДД, прислать подтверждение.
6. Приложить информацию с названием и версией компонентов витрины с ошибкой.
7. Приложить конфигурационные файлы затронутых компонентов (`application.yml` и/или содержимое `ENV` компонента из `Portainer`)
8. Приложить логи компонента с ошибкой
9. Сделать описание проблемы, при которой возникает ошибка, с обязательным указанием даты и времени ошибки.

Ж.5.3.1.2 Рекомендации по выполнению требований к качеству данных, размещаемых на витринах

Для определения причин отклонения значений показателей качества данных, размещаемых на витринах, и достижения целевых показателей в РМИС Субъектов РФ рекомендуется осуществлять указанные мероприятия:

1. Субъектам РФ регулярно проводить сверку данных регистра пациентов с данными реестра застрахованных лиц и прикреплений ТФОМС.
2. Субъектам РФ регулярно проверять наличие расписаний врачей в витрине данных со справочником ФРМР.
3. Субъектам РФ обеспечить настройку конкурентных слотов для записи.

Приложение Ж.1

Справочник должностей медицинских работников

Код должности – значение data_data1 в событиях get_slots и request_booking, перечисленных в разделе Ж.4.1.

Код должности	Наименование должности
1	Должности работников медицинских организаций
2	Должности медицинских работников
3	Должности руководителей медицинских организаций
4	главный врач (начальник) медицинской организации
5	директор больницы (дома) сестринского ухода, хосписа
6	заместитель руководителя (начальника) медицинской организации
7	заведующий (начальник) структурного подразделения (отдела, отделения, лаборатории, кабинета, отряда и другое) медицинской организации - врач-специалист
8	заведующий (главный врач, начальник) структурного подразделения, осуществляющего медицинскую деятельность, иной организации
9	главная медицинская сестра (главная акушерка, главный фельдшер)
10	Должности специалистов с высшим профессиональным (медицинским) образованием (врачи)
11	врач-стажер
12	врачи-специалисты
13	врач-акушер-гинеколог
14	врач-акушер-гинеколог цехового врачебного участка
15	врач-аллерголог-иммунолог
16	врач-анестезиолог-реаниматолог
17	врач-бактериолог
18	врач-вирусолог
19	врач-гастроэнтеролог
20	врач-гематолог
21	врач-генетик
22	врач-гериатр
23	врач-дезинфектолог
24	врач-дерматовенеролог
25	врач - детский кардиолог
26	врач - детский онколог
27	врач - детский уролог-андролог
28	врач - детский хирург
29	врач - детский эндокринолог
30	врач-диабетолог
31	врач-диетолог
32	врач здравпункта
33	врач-инфекционист
34	врач-кардиолог
35	врач клинической лабораторной диагностики
36	врач - клинический миколог
37	врач - клинический фармаколог
38	врач-колопроктолог
39	врач-косметолог
40	врач-лаборант
41	врач - лабораторный генетик
42	врач - лабораторный миколог
43	врач мануальной терапии
44	врач-методист
45	врач-невролог
46	врач-нейрохирург
47	врач-неонатолог

Код должности	Наименование должности
48	врач-нефролог
49	врач общей практики (семейный врач)
50	врач-онколог
51	врач-ортодонт
52	врач-остеопат
53	врач-оториноларинголог
54	врач-офтальмолог
55	врач-офтальмолог-протезист
56	врач-паразитолог
57	врач-патологоанатом
58	врач-педиатр
59	врач-педиатр участковый
60	врач-педиатр городской (районный)
61	врач - пластический хирург
62	врач по авиационной и космической медицине
63	врач по водолазной медицине
64	врач по гигиене детей и подростков
65	врач по гигиене питания
66	врач по гигиене труда
67	врач по гигиеническому воспитанию
68	врач по коммунальной гигиене
69	врач по лечебной физкультуре
70	врач по медико-социальной экспертизе
71	врач по медицинской профилактике
72	врач по медицинской реабилитации
73	врач по общей гигиене
74	врач по паллиативной медицинской помощи
75	врач по радиационной гигиене
76	врач по рентгенэндоваскулярным диагностике и лечению
77	врач по санитарно-гигиеническим лабораторным исследованиям
78	врач по спортивной медицине
79	врач приемного отделения
80	врач-профпатолог
81	врач-психиатр
82	врач-психиатр участковый
83	врач-психиатр детский
84	врач-психиатр детский участковый
85	врач-психиатр подростковый
86	врач-психиатр подростковый участковый
87	врач-психиатр-нарколог
88	врач-психиатр-нарколог участковый
89	врач-психотерапевт
90	врач-пульмонолог
91	врач-радиолог
92	врач-радиотерапевт
93	врач-ревматолог
94	врач-рентгенолог
95	врач-рефлексотерапевт
96	врач-сексолог
97	врач-сердечно-сосудистый хирург
98	врач скорой медицинской помощи
99	врач-статистик
100	врач-стоматолог
101	врач-стоматолог детский
102	врач-стоматолог-ортопед
103	врач-стоматолог-терапевт

Код должности	Наименование должности
104	врач-стоматолог-хирург
105	врач - судебно-медицинский эксперт
106	врач - судебно-психиатрический эксперт
107	врач-сурдолог-оториноларинголог
108	врач-сурдолог-протезист
109	врач-терапевт
110	врач-терапевт участковый
111	врач-терапевт участковый цехового врачебного участка
112	врач-терапевт подростковый
113	врач-токсиколог
114	врач-травматолог-ортопед
115	врач-трансфузиолог
116	врач ультразвуковой диагностики
117	врач-уролог
118	врач-физиотерапевт
119	врач-фтизиатр
120	врач-фтизиатр участковый
121	врач функциональной диагностики
122	врач-хирург
123	врач - торакальный хирург
124	врач - челюстно-лицевой хирург
125	врач-эндокринолог
126	врач-эндоскопист
127	врач-эпидемиолог
128	старший врач станции (отделения) скорой медицинской помощи
129	старший врач станции (отделения) скорой медицинской помощи горноспасательных частей
130	судовой врач
131	Должности специалистов с высшим профессиональным (немедицинским) образованием
132	биолог
133	зоолог
134	эмбриолог
135	энтомолог
136	инструктор-методист по лечебной физкультуре
137	логопед
138	медицинский физик
139	медицинский психолог
140	эксперт-физик по контролю за источниками ионизирующих и неионизирующих излучений
141	судебный эксперт (эксперт-биохимик, эксперт-генетик, эксперт-химик)
142	химик-эксперт медицинской организации
143	Должности специалистов со средним профессиональным (медицинским) образованием (средний медицинский персонал)
144	заведующий здравпунктом - фельдшер (медицинская сестра)
145	заведующий фельдшерско-акушерским пунктом - фельдшер (акушер, медицинская сестра)
146	заведующий кабинетом медицинской профилактики - фельдшер (медицинская сестра)
147	заведующий производством учреждений (отделов, отделений, лабораторий) зубопротезирования
148	заведующий молочной кухней
149	старшая медицинская сестра (акушер, фельдшер, операционная медицинская сестра, зубной техник)
150	акушер
151	гигиенист стоматологический
152	зубной врач
153	зубной техник
154	инструктор-дезинфектор
155	инструктор по гигиеническому воспитанию
156	инструктор по лечебной физкультуре

Код должности	Наименование должности
157	инструктор по трудовой терапии
158	лаборант
159	медицинская сестра
160	медицинская сестра - анестезист
161	медицинская сестра врача общей практики (семейного врача)
162	медицинская сестра диетическая
163	медицинская сестра медико-социальной помощи
164	медицинская сестра палатная (постовая)
165	медицинская сестра патронажная
166	медицинская сестра перевязочной
167	медицинская сестра по косметологии
168	медицинская сестра по массажу
169	медицинская сестра (фельдшер) по приему вызовов скорой медицинской помощи и передаче их выездным бригадам скорой медицинской помощи
170	медицинская сестра приемного отделения
171	медицинская сестра процедурной
172	медицинская сестра по реабилитации
173	медицинская сестра стерилизационной
174	медицинская сестра участковая
175	медицинская сестра по физиотерапии
176	медицинский дезинфектор
177	медицинский лабораторный техник (фельдшер-лаборант)
178	медицинский оптик-оптометрист
179	медицинский регистратор
180	медицинский статистик
181	медицинский технолог
182	операционная медицинская сестра
183	помощник врача
184	помощник врача-эпидемиолога
185	помощник врача-паразитолога
186	помощник врача по гигиене детей и подростков
187	помощник врача по гигиене питания
188	помощник врача по гигиене труда
189	помощник врача по гигиеническому воспитанию
190	помощник врача по коммунальной гигиене
191	помощник врача по общей гигиене
192	помощник врача по радиационной гигиене
193	помощник энтомолога
194	рентгенолаборант
195	фельдшер
196	фельдшер скорой медицинской помощи
197	фельдшер-нарколог
198	фельдшер-водитель скорой медицинской помощи
199	прочий средний медицинский персонал
200	Должности иных медицинских работников (младший медицинский персонал)
201	младшая медицинская сестра по уходу за больными
202	санитар
203	сестра-хозяйка
204	санитар-водитель
205	Должности прочего общеучрежденческого персонала медицинских организаций
206	ИТ-специалист
207	водитель скорой медицинской помощи
208	бухгалтер
209	специалист отдела кадров
210	гардеробщик
211	библиотекарь

Код должности	Наименование должности
212	лифтер
213	иные профессии рабочих и служащих
214	Должности работников фармацевтических организаций
215	Должности фармацевтических работников
216	Должности руководителей фармацевтических организаций
217	директор (заведующий, начальник) аптечной организации
218	заместитель директора (заведующего, начальника) аптечной организации
219	заведующий складом организации оптовой торговли лекарственными средствами
220	заведующий медицинским складом мобилизационного резерва
221	заместитель заведующего складом организации оптовой торговли лекарственными средствами
222	заведующий (начальник) структурного подразделения (отдела) аптечной организации
223	Должности специалистов с высшим профессиональным (фармацевтическим) образованием (провизоры)
224	провизор-стажер
225	провизор
226	провизор-технолог
227	провизор-аналитик
228	старший провизор
229	Должности специалистов со средним профессиональным (фармацевтическим) образованием (средний фармацевтический персонал)
230	старший фармацевт
231	фармацевт
232	младший фармацевт
233	Должности иных фармацевтических работников (младший фармацевтический персонал)
234	санитар (мойщик)
235	фасовщик
236	Должности прочего общеучрежденческого персонала фармацевтических организаций
237	иные профессии рабочих
238	Должности работников образовательных организаций и НИИ
239	Должности руководителей образовательных организаций и НИИ
240	руководитель (ректор, директор, заведующий, начальник) образовательной организации
241	президент образовательной организации высшего профессионального образования
242	заместитель руководителя (директора, заведующего, начальника) образовательной организации
243	руководитель (директор, заведующий, начальник, управляющий) структурного подразделения
244	руководитель (директор, заведующий, начальник, управляющий) обособленного структурного подразделения
245	заместитель руководителя (директора, заведующего, начальника, управляющего) структурного подразделения
246	заведующий (начальник) техническим архивом, чертежно-копировальным бюро; лабораторией (компьютерного и фото-, кинооборудования, оргтехники, средств связи)
247	заведующий (начальник) отделом научно-технической информации
248	начальник (руководитель) бригады (группы)
249	заведующий (начальник) научно-исследовательским (конструкторским) отделом (лабораторией, отделением, сектором)
250	заведующий (начальник) научным центром
251	начальник (заведующий) отдела (управления) аспирантуры
252	первый проректор
253	проректор
254	помощник ректора
255	помощник проректора
256	советник при ректорате
257	ученый секретарь совета образовательной организации
258	ученый секретарь совета факультета

Код должности	Наименование должности
259	руководитель (заведующий) учебной (учебно-производственной, производственной) практики
260	заведующий (директор, начальник, руководитель) питомника, учебного вивария, лесхоза, опытного поля, ботанического сада (дендрария)
261	старший мастер
262	директор (начальник, руководитель) студенческого дворца культуры
263	директор (начальник, заведующий) студенческого общежития
264	директор (руководитель, заведующий) издательства учебной литературы и учебных пособий
265	Должности работников образовательных организаций
266	Должности педагогических работников, отнесенных к профессорско-преподавательскому составу
267	декан факультета
268	начальник факультета
269	директор института
270	начальник института
271	заведующий кафедрой
272	начальник кафедры
273	заместитель начальника кафедры
274	профессор
275	доцент
276	старший преподаватель
277	преподаватель
278	ассистент
279	заведующий курсом
280	Должности иных педагогических работников образовательных организаций
281	воспитатель
282	инструктор-методист
283	инструктор по труду
284	инструктор по физической культуре
285	концертмейстер
286	логопед
287	мастер производственного обучения
288	методист
289	музыкальный руководитель
290	педагог дополнительного образования
291	педагог-библиотекарь
292	педагог-организатор
293	педагог-психолог
294	преподаватель
295	преподаватель-организатор основ безопасности жизнедеятельности
296	руководитель физического воспитания
297	социальный педагог
298	старший вожатый
299	старший воспитатель
300	старший инструктор-методист
301	старший методист
302	старший педагог дополнительного образования
303	старший тренер-преподаватель
304	тренер-преподаватель
305	тьютор
306	учитель
307	учитель-дефектолог
308	учитель-логопед
309	Должности научных работников
310	младший научный сотрудник
311	научный сотрудник

Код должности	Наименование должности
312	старший научный сотрудник
313	ведущий научный сотрудник
314	главный научный сотрудник
315	стажер-исследователь
316	лаборант-исследователь
317	Должности учебного-вспомогательного персонала образовательных организаций
318	заведующий учебным кабинетом
319	старший лаборант
320	лаборант
321	старший препаратор
322	рабочий кафедры
323	Должности административно-хозяйственного персонала образовательных и научно-исследовательских организаций
324	диспетчер факультета
325	ИТ-специалист
326	инженер
327	инженер-программист
328	водитель
329	специалист по учебно-методической работе
330	учебный мастер
331	бухгалтер
332	специалист отдела кадров
333	иные профессии рабочих
334	главный врач медицинской организации
335	начальник медицинской организации
336	заместитель руководителя медицинской организации
337	заместитель начальника медицинской организации
340	заведующий структурным подразделением, осуществляющим медицинскую деятельность, иной организации
341	главный врач структурного подразделения, осуществляющего медицинскую деятельность, иной организации
342	начальник структурного подразделения, осуществляющего медицинскую деятельность, иной организации
343	главная медицинская сестра
344	главная акушерка
345	главный фельдшер
346	судебный эксперт
347	судебный эксперт-биохимик
348	судебный эксперт-генетик
349	судебный эксперт-химик
350	заведующий здравпунктом - фельдшер
351	заведующий здравпунктом - медицинская сестра
352	заведующий фельдшерско-акушерским пунктом - фельдшер
353	заведующий фельдшерско-акушерским пунктом - акушер
354	заведующий фельдшерско-акушерским пунктом - медицинская сестра
355	заведующий кабинетом медицинской профилактики - фельдшер
356	заведующий кабинетом медицинской профилактики - медицинская сестра
357	старшая медицинская сестра
358	старший акушер
359	старший фельдшер
360	старшая операционная медицинская сестра
361	старший зубной техник
362	медицинская сестра по приему вызовов скорой медицинской помощи и передаче их выездным бригадам скорой медицинской помощи
363	фельдшер по приему вызовов скорой медицинской помощи и передаче их выездным бригадам скорой медицинской помощи

Код должности	Наименование должности
364	медицинский лабораторный техник
365	фельдшер-лаборант
366	продавец оптики
367	директор аптечной организации
368	заведующий аптечной организации
369	начальник аптечной организации
370	заместитель директора аптечной организации
371	заместитель заведующего аптечной организации
372	заместитель начальника аптечной организации
373	заведующий структурного подразделения аптечной организации
374	начальник структурного подразделения аптечной организации
375	руководитель образовательной организации
376	ректор образовательной организации высшего профессионального образования
377	директор образовательной организации
378	заведующий образовательной организации
379	начальник образовательной организации
380	заместитель руководителя образовательной организации
381	заместитель директора образовательной организации
382	заместитель заведующего образовательной организации
383	заместитель начальника образовательной организации
384	руководитель структурного подразделения
385	директор структурного подразделения
386	заведующий структурного подразделения
387	начальник структурного подразделения
388	управляющий структурного подразделения
389	руководитель обособленного структурного подразделения
390	директор обособленного структурного подразделения
391	заведующий обособленного структурного подразделения
392	начальник обособленного структурного подразделения
393	управляющий обособленного структурного подразделения
394	заместитель руководителя структурного подразделения
395	заместитель директора структурного подразделения
396	заместитель заведующего структурного подразделения
397	заместитель начальника структурного подразделения
398	заместитель управляющего структурного подразделения
401	заведующий отделом научно-технической информации
402	начальник отдела научно-технической информации
403	начальник бригады
404	руководитель бригады
407	заведующий научным центром
408	начальник научного центра
409	начальник отдела аспирантуры
410	заведующий отделом аспирантуры
417	директор студенческого дворца культуры
418	начальник студенческого дворца культуры
419	руководитель студенческого дворца культуры
420	директор студенческого общежития
421	начальник студенческого общежития
422	заведующий студенческого общежития
423	директор издательства учебной литературы и учебных пособий
424	руководитель издательства учебной литературы и учебных пособий
425	заведующий издательства учебной литературы и учебных пособий
426	заместитель декана
427	директор больницы (дома) сестринского ухода
428	директор хосписа
429	заведующий отделом медицинской организации

Код должности	Наименование должности
430	заведующий отделением медицинской организации
431	заведующий лабораторией медицинской организации
432	заведующий кабинетом медицинской организации
433	заведующий отрядом медицинской организации
434	начальник отдела медицинской организации
435	начальник отделения медицинской организации
436	начальник лаборатории медицинской организации
437	начальник кабинета медицинской организации
438	начальник отряда медицинской организации
439	заведующий отделом аптечной организации
440	начальник отдела аптечной организации
441	начальник группы
442	руководитель группы
444	ученый секретарь совета института
445	заведующий производством учреждений зубопротезирования
446	заведующий производством отделов зубопротезирования
447	заведующий производством отделений зубопротезирования
448	заведующий производством лабораторий зубопротезирования
449	старший врач станции скорой медицинской помощи
450	старший врач отделения скорой медицинской помощи
451	старший врач станции скорой медицинской помощи горноспасательных частей
452	старший врач отделения скорой медицинской помощи горноспасательных частей
453	руководитель учебно-производственной практики
454	руководитель производственной практики
455	заведующий учебной практики
456	заведующий учебно-производственной практики
457	заведующий производственной практики
458	заведующий питомника
459	заведующий учебного вивария
460	заведующий лесхоза
461	заведующий опытного поля
462	заведующий ботанического сада (дендрария)
463	директор питомника
464	директор учебного вивария
465	директор лесхоза
466	директор опытного поля
467	директор ботанического сада (дендрария)
468	начальник питомника
469	начальник учебного вивария
470	начальник лесхоза
471	начальник опытного поля
472	начальник ботанического сада (дендрария)
473	руководитель питомника
474	руководитель учебного вивария
475	руководитель лесхоза
476	руководитель опытного поля
477	руководитель ботанического сада (дендрария)
478	начальник управления аспирантуры
479	заведующий управлением аспирантуры
480	руководитель (директор, заведующий, начальник) научной организации
481	руководитель научной организации
482	директор научной организации
483	заведующий научной организации
484	начальник научной организации
485	заместитель руководителя (директора, заведующего, начальника) научной организации
486	заместитель руководителя научной организации

Код должности	Наименование должности
487	заместитель директора научной организации
488	заместитель заведующего научной организации
489	заместитель начальника научной организации
490	заведующий техническим архивом
491	заведующий чертежно-копировальным бюро
492	заведующий лабораторией (компьютерного и фото-, кинооборудования, оргтехники, средств связи)
493	начальник технического архива
494	начальник чертежно-копировального бюро
495	начальник лаборатории (компьютерного и фото-, кинооборудования, оргтехники, средств связи)
496	заведующий научно-исследовательским отделом
497	заведующий научно-исследовательской лабораторией
498	заведующий научно-исследовательским отделением
499	заведующий научно-исследовательским сектором
500	заведующий конструкторским отделом
501	заведующий конструкторской лабораторией
502	заведующий конструкторским отделением
503	заведующий конструкторским сектором
504	начальник научно-исследовательского отдела
505	начальник научно-исследовательской лаборатории
506	начальник научно-исследовательским отделением
507	начальник научно-исследовательского сектора
508	начальник конструкторского отдела
509	начальник конструкторской лаборатории
510	начальник конструкторского отделения
511	начальник конструкторского сектора
512	руководитель учебной практики
513	руководитель бюро - врач по медико-социальной экспертизе
514	руководитель экспертного состава - врач по медико-социальной экспертизе
515	руководитель – главный эксперт по медико – социальной экспертизе
516	специалист по социальной работе
517	врач - детский онколог-гематолог
518	врач - медицинский микробиолог
519	врач физической и реабилитационной медицины
520	заведующий станцией скорой медицинской помощи
521	заведующий подстанцией скорой медицинской помощи
522	руководитель – федеральный эксперт по медико – социальной экспертизе
523	заместитель руководителя – федерального эксперта по медико – социальной экспертизе
524	заместитель руководителя – главного эксперта по медико – социальной экспертизе
525	заведующий центром медицинской организации

Приложение Ж.2

Справочник ошибок

Код ошибки – значение в поле data_result передаваемых событий, перечисленных в разделе Ж.4.1.

Код ошибки	Описание ошибки	Категория ошибки	Комментарий
0	Успех	Успех	
7	Нет данных	Организационные	Получен пустой массив из витрины на регламентированный запрос. Считаем как бизнес ошибка (но возможно, что была ошибка при загрузке данных в витрину на стороне МИС)
3	Ошибка при вызове регламентированного запроса	Технические	Ошибка при вызове агента ПОДД (вызов REST /internal/api/einfahrt). Это может быть проблема агента на стороне ЕПГУ, ядра ПОДД, агента ПОДД на стороне поставщика или в витрине (техническая ошибка/таймаут).
11	Слот занят или заблокирован	Организационные	Бизнес-ошибка МИС
12	Есть запись на то же время	Организационные	Бизнес-ошибка МИС
13	Есть запись к специалисту на тот же день	Организационные	Бизнес-ошибка МИС
14	Тип слота не подходит по возрасту	Организационные	Бизнес-ошибка МИС
15	Слот аннулирован	Организационные	Бизнес-ошибка МИС
17	Пользователь не найден в витрине	Организационные	Бизнес-ошибка МИС
18	Внутренняя ошибка МИС	Технические	Ошибка в МИС при букировании слота
101	Ошибка при вызове СМЭВ	Ошибки ПОДД	Для 10700 это ошибка при вызове агента ПОДД (вызов REST openapi /einfahrt/internal/api/einfahrt/openapi/{Код_ИС}/FerDtm/services/fer/booking/book на букирование или /einfahrt/internal/api/einfahrt/openapi/{Код_ИС}/FerDtm/services/fer/booking/cancel на отмену) Для 600204 ошибка возникает при отправке запроса по ВС
1	Неуспех	Ошибки ЕПГУ	Ошибка при открытии страницы или регион не доступен для услуги. Только для метрик service_init, choice, show_active_record, search_region, choice_method
405	Техническая ошибка ЕПГУ	Ошибки ЕПГУ	Ошибка балансирующего или компоненты (NSI/LK)
500	Техническая ошибка ЕПГУ	Ошибки ЕПГУ	Ошибка балансирующего или компоненты (NSI/LK)
502	Техническая ошибка ЕПГУ	Ошибки ЕПГУ	Ошибка балансирующего или компоненты (NSI/LK)
504	Техническая ошибка ЕПГУ	Ошибки ЕПГУ	Ошибка балансирующего или компоненты (NSI/LK)
6	Услуга временно не доступна	Технические	Ошибка тайм-аута
200	Запись не возможна	Организационные	Одна из бизнес ошибок: слот занят, уже есть запись, слот не подходит по возрасту пациента
201	Успех	Успех	Успешная запись по форме 600204

Код ошибки	Описание ошибки	Категория ошибки	Комментарий
202	Превышен лимит бронирования слота по заявке	Ошибки ЕПГУ	Защита от ботов, agg/book по конкретному заявлению разрешается вызывать не чаще чем раз в минуту

Приложение Ж.3

Скрипты для расчета показателей услуги «Запись на прием к врачу (на витрине данных)»

Пример скрипта расчета результатов шагов воронки по пользователям для Республики Башкортостан

```

SELECT `Этапы воронки` AS `Этапы воронки`,
      AVG(`Мера`) AS `AVG(Мера)`
FROM
  (WITH reg_dict AS
    (SELECT region_code,
      regions_name
    FROM
      (SELECT ['00','79', '80', '81', '84', '82', '26', '83', '85', '91', '86', '87',
        '88', '89',
        '98', '90', '92', '93', '94', '95', '96', '97', '01', '03', '04',
        '05', '07', '08',
        '10', '11', '12', '14', '15', '17', '18', '19', '20', '24', '25',
        '27', '29', '30',
        '32', '33', '34', '37', '38', '41', '42', '44', '46', '47', '22',
        '49', '50', '52',
        '53', '54', '56', '57', '58', '60', '61', '36', '63', '64', '65',
        '66', '68', '28',
        '69', '70', '71', '73', '75', '76', '78', '45', '40', '99', '111',
        '711', '77', '7114', '35', '67', '55'] AS code_ok,
        ['ФМБА','Республика Адыгея (Адыгея)', 'Республика Башкортостан', 'Республика Бурятия',
        'Республика Алтай', 'Республика Дагестан',
        'Республика Ингушетия', 'Кабардино-Балкарская Республика', 'Республика
        Калмыкия', 'Карачаево-Черкесская Республика', 'Республика Карелия',
        'Республика Коми', 'Республика Марий Эл', 'Республика Мордовия', 'Республика
        Саха (Якутия)', 'Республика Северная Осетия - Алания',
        'Республика Татарстан (Татарстан)', 'Республика Тыва', 'Удмуртская
        Республика', 'Республика Хакасия', 'Чеченская Республика',
        'Чувашская Республика - Чувашия', 'Алтайский край', 'Краснодарский край',
        'Красноярский край', 'Приморский край',
        'Ставропольский край', 'Хабаровский край', 'Амурская область', 'Архангельская
        область',
        'Астраханская область', 'Белгородская область', 'Брянская область', 'Владимирская
        область', 'Волгоградская область',
        'Вологодская область', 'Воронежская область', 'Ивановская область', 'Иркутская
        область', 'Калининградская область',
        'Калужская область', 'Камчатский край', 'Кемеровская область', 'Кировская
        область', 'Костромская область', 'Курганская область',
        'Курская область', 'Ленинградская область', 'Липецкая область', 'Магаданская
        область', 'Московская область', 'Мурманская область',
        'Нижегородская область', 'Новгородская область', 'Новосибирская область',
        'Омская область', 'Оренбургская область',
        'Орловская область', 'Пензенская область', 'Пермский край', 'Псковская
        область', 'Ростовская область', 'Рязанская область',
        'Самарская область', 'Саратовская область', 'Сахалинская область', 'Свердловская
        область', 'Смоленская область', 'Тамбовская область',
        'Тверская область', 'Томская область', 'Тульская область', 'Тюменская
        область', 'Ульяновская область', 'Челябинская область',
        'Забайкальский край', 'Ярославская область', 'г. Москва', 'г. Санкт-Петербург',
        'Еврейская автономная область',
        'Ненецкий автономный округ', 'Ханты-Мансийский автономный округ - Югра',
        'Чукотский автономный округ', 'Ямало-Ненецкий автономный округ',
        'Республика Крым', 'г. Севастополь', 'Иные территории, включая город и космодром
        Байконур'] AS reg) ARRAY

```

```

JOIN reg AS regions_name,
      code_ok AS region_code
GROUP BY 1,
      2),
oid_data AS
(SELECT dt,
      oid,
      MIN(region) AS region_oid,
      MAX(step1_validate) AS step1_validate_oid,
      MIN(step1_dt) AS step1_dt_oid,
      MIN(step1_result) AS step1_result_oid,
      MIN(step1_text) AS step1_text_oid,
      MAX(step2_mo) AS step2_mo_oid,
      MIN(step2_dt) AS step2_dt_oid,
      MIN(step2_result) AS step2_result_oid,
      MIN(step2_text) AS step2_text_oid,
      MAX(step3_specialist) AS step3_specialist_oid,
      MIN(step3_dt) AS step3_dt_oid,
      MIN(step3_result) AS step3_result_oid,
      MIN(step3_text) AS step3_text_oid,
      MAX(step4_resource) AS step4_resource_oid,
      MIN(step4_dt) AS step4_dt_oid,
      MIN(step4_result) AS step4_result_oid,
      MIN(step4_text) AS step4_text_oid,
      MAX(step5_slots) AS step5_slots_oid,
      MIN(step5_dt) AS step5_dt_oid,
      MIN(step5_result) AS step5_result_oid,
      MIN(step5_text) AS step5_text_oid,
      MAX(step6_booking) AS step6_booking_oid,
      MIN(step6_dt) AS step6_dt_oid,
      MIN(step6_result) AS step6_result_oid,
      MIN(step6_text) AS step6_text_oid
FROM zdrav.sess_funnel
WHERE region_code =
      (SELECT region_code
      FROM reg_dict
      WHERE regions_name = 'Республика Башкортостан')
AND step1_validate = 1
AND service = '10700'
AND dt BETWEEN parseDateTimeBestEffort('2023-02-28T00:00:00') AND
parseDateTimeBestEffort('2023-03-01T00:00:00')
GROUP BY dt,
      oid) SELECT "Этапы воронки",
      "Мера"
FROM
      (SELECT [
      'Успешно идентифицированы (%)',
      'Получен список специальностей (%)',
      'Получен список доступных слотов (%)',
      'Успешно записались (%)'] AS funnel,
      [
      SUM(if(step1_validate_oid = 1 and step1_result_oid = 0, 1, 0)) / COUNT(*),
      SUM(if(step3_specialist_oid = 1 and step3_result_oid = 0, 1, 0)) / COUNT(*),
      SUM(if(step5_slots_oid = 1 and step5_result_oid = 0, 1, 0)) / COUNT(*),
      SUM(if(step6_booking_oid = 1 and step6_result_oid = 0, 1, 0)) / COUNT(*)] AS value
      FROM oid_data) ARRAY
JOIN funnel AS "Этапы воронки",
      value AS "Мера"
GROUP BY 1,
      2) AS `virtual_table`

```

```
GROUP BY `Этапы воронки`
ORDER BY `AVG(Мера)` DESC
LIMIT 10
```

Пример скрипта расчета результатов для шага Идентификация пациента по пользователям для Республики Башкортостан

```
SELECT `Название` AS `Название`,
       `Параметр` AS `Параметр`,
       max(`Значение`) AS `MAX(Значение)`
FROM
(WITH reg_dict AS
 (SELECT region_code,
         regions_name
  FROM
  (SELECT ['00','79', '80', '81', '84', '82', '26', '83', '85', '91', '86', '87',
          '88', '89',
          '98', '90', '92', '93', '94', '95', '96', '97', '01', '03', '04',
          '05', '07', '08',
          '10', '11', '12', '14', '15', '17', '18', '19', '20', '24', '25',
          '27', '29', '30',
          '32', '33', '34', '37', '38', '41', '42', '44', '46', '47', '22',
          '49', '50', '52',
          '53', '54', '56', '57', '58', '60', '61', '36', '63', '64', '65',
          '66', '68', '28',
          '69', '70', '71', '73', '75', '76', '78', '45', '40', '99', '111',
          '711', '77', '7114', '35', '67', '55'] AS code_ok,
          ['ФМБА','Республика Адыгея (Адыгея)', 'Республика Башкортостан', 'Республика Бурятия',
          'Республика Алтай', 'Республика Дагестан',
          'Республика Ингушетия','Кабардино-Балкарская Республика', 'Республика Калмыкия',
          'Карачаево-Черкесская Республика', 'Республика Карелия',
          'Республика Коми', 'Республика Марий Эл', 'Республика Мордовия', 'Республика
          Саха (Якутия)', 'Республика Северная Осетия - Алания',
          'Республика Татарстан (Татарстан)', 'Республика Тыва', 'Удмуртская
          Республика', 'Республика Хакасия', 'Чеченская Республика',
          'Чувашская Республика - Чувашия', 'Алтайский край', 'Краснодарский край',
          'Красноярский край', 'Приморский край',
          'Ставропольский край', 'Хабаровский край', 'Амурская область', 'Архангельская
          область',
          'Астраханская область', 'Белгородская область', 'Брянская область', 'Владимирская
          область',
          'Волгоградская область',
          'Вологодская область', 'Воронежская область', 'Ивановская область', 'Иркутская
          область',
          'Калининградская область',
          'Калужская область', 'Камчатский край', 'Кемеровская область', 'Кировская
          область',
          'Костромская область', 'Курганская область',
          'Курская область', 'Ленинградская область','Липецкая область', 'Магаданская
          область',
          'Московская область', 'Мурманская область',
          'Нижегородская область', 'Новгородская область', 'Новосибирская область',
          'Омская область', 'Оренбургская область',
          'Орловская область', 'Пензенская область', 'Пермский край', 'Псковская
          область',
          'Ростовская область', 'Рязанская область',
          'Самарская область', 'Саратовская область', 'Сахалинская область', 'Свердловская
          область',
          'Смоленская область', 'Тамбовская область',
          'Тверская область', 'Томская область', 'Тульская область', 'Тюменская
          область',
          'Ульяновская область', 'Челябинская область',
          'Забайкальский край', 'Ярославская область', 'г. Москва', 'г. Санкт-Петербург',
          'Еврейская автономная область',
          'Ненецкий автономный округ', 'Ханты-Мансийский автономный округ - Югра',
          'Чукотский автономный округ', 'Ямало-Ненецкий автономный округ',
```

```

        'Республика Крым',      'г. Севастополь', 'Иные территории, включая город и космодром
Байконур'] AS reg) ARRAY
    JOIN reg AS regions_name,
        code_ok AS region_code
    GROUP BY 1,
        2),
    oid_data AS
(SELECT dt,
    oid,
    MIN(region) AS region_oid,
    MAX(step1_validate) AS step1_validate_oid,
    MIN(step1_dt) AS step1_dt_oid,
    MIN(step1_result) AS step1_result_oid,
    MIN(step1_text) AS step1_text_oid,
    MAX(step2_mo) AS step2_mo_oid,
    MIN(step2_dt) AS step2_dt_oid,
    MIN(step2_result) AS step2_result_oid,
    MIN(step2_text) AS step2_text_oid,
    MAX(step3_specialist) AS step3_specialist_oid,
    MIN(step3_dt) AS step3_dt_oid,
    MIN(step3_result) AS step3_result_oid,
    MIN(step3_text) AS step3_text_oid,
    MAX(step4_resource) AS step4_resource_oid,
    MIN(step4_dt) AS step4_dt_oid,
    MIN(step4_result) AS step4_result_oid,
    MIN(step4_text) AS step4_text_oid,
    MAX(step5_slots) AS step5_slots_oid,
    MIN(step5_dt) AS step5_dt_oid,
    MIN(step5_result) AS step5_result_oid,
    MIN(step5_text) AS step5_text_oid,
    MAX(step6_booking) AS step6_booking_oid,
    MIN(step6_dt) AS step6_dt_oid,
    MIN(step6_result) AS step6_result_oid,
    MIN(step6_text) AS step6_text_oid
FROM zdrav.ssess_funnel
WHERE region_code =
    (SELECT region_code
    FROM reg_dict
    WHERE regions_name = 'Республика Башкортостан')
AND step1_validate = 1
AND service = '10700'
AND dt BETWEEN parseDateTimeBestEffort('2023-02-28T00:00:00') AND
parseDateTimeBestEffort('2023-03-01T00:00:00')
GROUP BY dt,
    oid) SELECT "Название",
        "Параметр",
        "Значение"
FROM
    (SELECT [',',',',',',',',','] AS name,
        [
            '1. Успешно идентифицированы (%)',
            '2. Ошибка ЕПГУ (%)',
            '3. Техническая ошибка РМИС (%)',
            '4. Организационная ошибка РМИС (%)',
            '5. Ошибка ПОДД (%)',
            '6. Ушедшие пользователи (%)'] AS funnel,
        [
            SUM(if(step1_validate_oid = 1 and step1_result_oid = 0,1,0)) / COUNT(*),
            SUM(if(step1_result_oid >= 400 and step1_result_oid != 999,1,0)) / COUNT(*),
            SUM(if(step1_result_oid = 3,1,0)) / COUNT(*),

```

```

SUM(if(step1_result_oid = 7,1,0)) / COUNT(*),
SUM(if(step1_result_oid >= 101 and step1_result_oid < 400,1,0)) / COUNT(*),
(COUNT(*) - SUM(if(step1_result_oid != 999,1,0))) / COUNT(*) ] AS value
FROM oid_data) ARRAY
JOIN name AS "Название",
funnel AS "Параметр",
value AS "Значение"
GROUP BY 1,
2,
3) AS `virtual_table`
GROUP BY `Название`,
`Параметр`
ORDER BY `MAX(Значение)` DESC
LIMIT 100;

```

Пример скрипта для расчета количества ошибок по описанию для шага
Идентификация пациента по пользователю для Республики Башкортостан

```

SELECT `Описание` AS `Описание`,
max(`Количество`) AS `MAX(Количество)`
FROM
(WITH reg_dict AS
(SELECT region_code,
regions_name
FROM
(SELECT ['00','79', '80', '81', '84', '82', '26', '83', '85', '91', '86', '87',
'88', '89',
'98', '90', '92', '93', '94', '95', '96', '97', '01', '03', '04',
'05', '07', '08',
'10', '11', '12', '14', '15', '17', '18', '19', '20', '24', '25',
'27', '29', '30',
'32', '33', '34', '37', '38', '41', '42', '44', '46', '47', '22',
'49', '50', '52',
'53', '54', '56', '57', '58', '60', '61', '36', '63', '64', '65',
'66', '68', '28',
'69', '70', '71', '73', '75', '76', '78', '45', '40', '99', '111',
'711', '77', '7114', '35', '67', '55'] AS code_ok,
['ФМБА','Республика Адыгея (Адыгея)', 'Республика Башкортостан', 'Республика Бурятия',
'Республика Алтай', 'Республика Дагестан',
'Республика Ингушетия','Кабардино-Балкарская Республика', 'Республика
Калмыкия', 'Карачаево-Черкесская Республика', 'Республика Карелия',
'Республика Коми', 'Республика Марий Эл', 'Республика Мордовия', 'Республика
Саха (Якутия)', 'Республика Северная Осетия - Алания',
'Республика Татарстан (Татарстан)', 'Республика Тыва', 'Удмуртская
Республика', 'Республика Хакасия', 'Чеченская Республика',
'Чувашская Республика - Чувашия', 'Алтайский край', 'Краснодарский край',
'Красноярский край', 'Приморский край',
'Ставропольский край', 'Хабаровский край', 'Амурская область', 'Архангельская
область',
'Астраханская область', 'Белгородская область', 'Брянская область', 'Владимирская
область', 'Волгоградская область',
'Вологодская область', 'Воронежская область', 'Ивановская область', 'Иркутская
область', 'Калининградская область',
'Калужская область', 'Камчатский край', 'Кемеровская область', 'Кировская
область', 'Костромская область', 'Курганская область',
'Курская область', 'Ленинградская область','Липецкая область', 'Магаданская
область', 'Московская область', 'Мурманская область',
'Нижегородская область', 'Новгородская область', 'Новосибирская область',
'Омская область', 'Оренбургская область',

```



```

        'Орловская область', 'Пензенская область', 'Пермский край', 'Псковская
область', 'Ростовская область', 'Рязанская область',
        'Самарская область', 'Саратовская область', 'Сахалинская область', 'Свердловская
область', 'Смоленская область', 'Тамбовская область',
        'Тверская область', 'Томская область', 'Тульская область', 'Тюменская
область', 'Ульяновская область', 'Челябинская область',
        'Забайкальский край', 'Ярославская область', 'г. Москва', 'г. Санкт-Петербург',
        'Еврейская автономная область',
        'Ненецкий автономный округ', 'Ханты-Мансийский автономный округ - Югра',
        'Чукотский автономный округ', 'Ямало-Ненецкий автономный округ',
        'Республика Крым', 'г. Севастополь', 'Иные территории, включая город и космодром
Байконур'] AS reg) ARRAY
JOIN reg AS regions_name,
code_ok AS region_code
GROUP BY 1,
2),
oid_data AS
(SELECT dt,
oid,
MIN(region) AS region_oid,
MAX(step1_validate) AS step1_validate_oid,
MIN(step1_dt) AS step1_dt_oid,
MIN(step1_result) AS step1_result_oid,
MIN(step1_text) AS step1_text_oid,
MAX(step2_mo) AS step2_mo_oid,
MIN(step2_dt) AS step2_dt_oid,
MIN(step2_result) AS step2_result_oid,
MIN(step2_text) AS step2_text_oid,
MAX(step3_specialist) AS step3_specialist_oid,
MIN(step3_dt) AS step3_dt_oid,
MIN(step3_result) AS step3_result_oid,
MIN(step3_text) AS step3_text_oid,
MAX(step4_resource) AS step4_resource_oid,
MIN(step4_dt) AS step4_dt_oid,
MIN(step4_result) AS step4_result_oid,
MIN(step4_text) AS step4_text_oid,
MAX(step5_slots) AS step5_slots_oid,
MIN(step5_dt) AS step5_dt_oid,
MIN(step5_result) AS step5_result_oid,
MIN(step5_text) AS step5_text_oid,
MAX(step6_booking) AS step6_booking_oid,
MIN(step6_dt) AS step6_dt_oid,
MIN(step6_result) AS step6_result_oid,
MIN(step6_text) AS step6_text_oid
FROM zdrav.sess_funnel
WHERE region_code =
(SELECT region_code
FROM reg_dict
WHERE regions_name = 'Республика Башкортостан')
AND step1_validate = 1
AND service = '10700'
AND dt BETWEEN parseDateTimeBestEffort('2023-02-28T00:00:00') AND
parseDateTimeBestEffort('2023-03-01T00:00:00')
GROUP BY dt,
oid) SELECT if(length(step1_text_oid)>150, SUBSTRING(step1_text_oid, 1, 40), step1_text_oid) AS
"Описание",
COUNT(*) AS "Количество"
FROM oid_data
WHERE step1_validate_oid = 1
AND step1_result_oid != 0

```

```

GROUP BY 1
ORDER BY "Количество" DESC) AS `virtual_table`
GROUP BY `Описание`
ORDER BY `MAX(Количество)` DESC
LIMIT 100;

```

Пример скрипта для расчета показателей опроса для Республики Башкортостан

```

SELECT `Причина отказа` AS `Причина отказа`,
       sum(`Количество ответов`) AS `SUM(Количество ответов)`
FROM
  (WITH reg_dict AS
    (SELECT region_code,
            regions_name
    FROM
      (SELECT ['00','79', '80', '81', '84', '82', '26', '83', '85', '91', '86', '87',
                '88', '89',
                '98', '90', '92', '93', '94', '95', '96', '97', '01', '03', '04',
                '05', '07', '08',
                '10', '11', '12', '14', '15', '17', '18', '19', '20', '24', '25',
                '27', '29', '30',
                '32', '33', '34', '37', '38', '41', '42', '44', '46', '47', '22',
                '49', '50', '52',
                '53', '54', '56', '57', '58', '60', '61', '36', '63', '64', '65',
                '66', '68', '28',
                '69', '70', '71', '73', '75', '76', '78', '45', '40', '99', '111',
                '711', '77', '7114', '35', '67', '55'] AS code_ok,
                ['ФМБА','Республика Адыгея (Адыгея)', 'Республика Башкортостан', 'Республика Бурятия',
                'Республика Алтай', 'Республика Дагестан',
                'Республика Ингушетия','Кабардино-Балкарская Республика', 'Республика
Калмыкия', 'Карачаево-Черкесская Республика', 'Республика Карелия',
                'Республика Коми', 'Республика Марий Эл', 'Республика Мордовия', 'Республика
Саха (Якутия)', 'Республика Северная Осетия - Алания',
                'Республика Татарстан (Татарстан)', 'Республика Тыва', 'Удмуртская
Республика', 'Республика Хакасия', 'Чеченская Республика',
                'Чувашская Республика - Чувашия', 'Алтайский край', 'Краснодарский край',
                'Красноярский край', 'Приморский край',
                'Ставропольский край', 'Хабаровский край', 'Амурская область', 'Архангельская
область',
                'Астраханская область', 'Белгородская область', 'Брянская область', 'Владимирская
область', 'Волгоградская область',
                'Вологодская область', 'Воронежская область', 'Ивановская область', 'Иркутская
область', 'Калининградская область',
                'Калужская область', 'Камчатский край', 'Кемеровская область', 'Кировская
область', 'Костромская область', 'Курганская область',
                'Курская область', 'Ленинградская область', 'Липецкая область', 'Магаданская
область', 'Московская область', 'Мурманская область',
                'Нижегородская область', 'Новгородская область', 'Новосибирская область',
                'Омская область', 'Оренбургская область',
                'Орловская область', 'Пензенская область', 'Пермский край', 'Псковская
область', 'Ростовская область', 'Рязанская область',
                'Самарская область', 'Саратовская область', 'Сахалинская область', 'Свердловская
область', 'Смоленская область', 'Тамбовская область',
                'Тверская область', 'Томская область', 'Тульская область', 'Тюменская
область', 'Ульяновская область', 'Челябинская область',
                'Забайкальский край', 'Ярославская область', 'г. Москва', 'г. Санкт-Петербург',
                'Еврейская автономная область',
                'Ненецкий автономный округ', 'Ханты-Мансийский автономный округ - Югра',
                'Чукотский автономный округ', 'Ямало-Ненецкий автономный округ',

```

```

        'Республика Крым',      'г. Севастополь', 'Иные территории, включая город и космодром
Байконур'] AS reg) ARRAY
    JOIN reg AS regions_name,
        code_ok AS region_code
    GROUP BY 1,
        2),
    sel_region AS
    (SELECT DISTINCT session
    FROM zdrav.search_region
    WHERE dateTime BETWEEN parseDateTimeBestEffort('2023-02-28T00:00:00') AND
    parseDateTimeBestEffort('2023-03-01T00:00:00')
    AND service = '10700'
    AND CASE
        when substring(data_data1
            from 1
            for 3) = '111' then '111'
        when (substring(data_data1
            from 1
            for 3) = '711'
            AND substring(data_data1
            from 1
            for 4) != '7114') then '711'
        when substring(data_data1
            from 1
            for 4) = '7114' then '7114'
        else substring(data_data1
            from 1
            for 2)
        END =
    (SELECT region_code
    FROM reg_dict
    WHERE regions_name = 'Республика Башкортостан') ),
    pre_data AS
    (SELECT *
    FROM zdrav.survey_results vp
    WHERE session IN
    (SELECT *
    FROM sel_region) ) SELECT COUNT(data_data1) as "Количество ответов",
        (CASE
            WHEN data_data1 = '1' THEN 'Нет нужной мне медицинской организации'
            WHEN data_data1 = '2' THEN 'Нет нужной специальности врача'
            WHEN data_data1 = '3' THEN 'Не было моего лечащего врача'
            WHEN data_data1 = '4' THEN 'Не удобное время приема'
            WHEN data_data1 = '5' THEN 'Передумал записываться'
            WHEN data_data1 = '6' THEN 'Не получается записаться на выбранное время'
            WHEN data_data1 = '7' THEN 'Другое'
            ELSE 'Без ответа'
        END) as "Причина отказа"
    FROM pre_data
    WHERE data_data1 != '0'
    GROUP BY "Причина отказа"
    ORDER BY "Количество ответов" DESC) AS `virtual_table`
GROUP BY `Причина отказа`
ORDER BY `SUM(Количество ответов)` DESC
LIMIT 100;

```

Приложение Ж.4

Скрипты для расчета показателей услуги «Запись на прием к врачу (КУ ФЭР)»

Пример скрипта расчета результатов шагов воронки по пользователям для Республики Башкортостан

```

SELECT "Этапы воронки" AS "Этапы воронки",
      AVG("Мера") AS "AVG(Мера)"
FROM
  (WITH reg_dict AS
    (SELECT region_code,
            regions_name
    FROM
      (SELECT ['00','79',      '80', '81', '84', '82', '26', '83', '85', '91', '86',
              '87', '88', '89',
              '98', '90', '92', '93', '94', '95', '96', '97', '01', '03', '04',
              '05', '07', '08',
              '10', '11', '12', '14', '15', '17', '18', '19', '20', '24', '25',
              '27', '29', '30',
              '32', '33', '34', '37', '38', '41', '42', '44', '46', '47', '22',
              '49', '50', '52',
              '53', '54', '56', '57', '58', '60', '61', '36', '63', '64', '65',
              '66', '68', '28',
              '69', '70', '71', '73', '75', '76', '78', '45', '40', '99', '111',
              '711', '77', '7114', '35', '67', '55'] AS code_ok,
              ['ФМБА','Республика Адыгея (Адыгея)', 'Республика Башкортостан', 'Республика
Бурятия',      'Республика Алтай', 'Республика Дагестан',
              'Республика Ингушетия',      'Кабардино-Балкарская Республика', 'Республика
Калмыкия',      'Карачаево-Черкесская Республика', 'Республика Карелия',
              'Республика Коми', 'Республика Марий Эл',      'Республика Мордовия',
              'Республика Саха (Якутия)', 'Республика Северная Осетия - Алания',
              'Республика Татарстан (Татарстан)', 'Республика Тыва',      'Удмуртская
Республика', 'Республика Хакасия', 'Чеченская Республика',
              'Чувашская Республика - Чувашия', 'Алтайский край',      'Краснодарский
край', 'Красноярский край', 'Приморский край',
              'Ставропольский край',      'Хабаровский край', 'Амурская область',
              'Архангельская область',
              'Астраханская область',      'Белгородская область',      'Брянская область',
              'Владимирская область',      'Волгоградская область',
              'Вологодская область','Воронежская область',      'Ивановская область',
              'Иркутская область', 'Калининградская область',
              'Калужская область', 'Камчатский край',      'Кемеровская область','Кировская
область',      'Костромская область','Курганская область',
              'Курская область',      'Ленинградская область',      'Липецкая область',
              'Магаданская область','Московская область', 'Мурманская область',
              'Нижегородская область',      'Новгородская область',      'Новосибирская
область',      'Омская область',      'Оренбургская область',
              'Орловская область', 'Пензенская область', 'Пермский край',      'Псковская
область',      'Ростовская область', 'Рязанская область',
              'Самарская область', 'Саратовская область', 'Сахалинская область',
              'Свердловская область',      'Смоленская область', 'Тамбовская область',
              'Тверская область', 'Томская область',      'Тульская область',      'Тюменская
область',      'Ульяновская область','Челябинская область',

```

```

        'Забайкальский край', 'Ярославская область', 'г. Москва', 'г. Санкт-Петербург',
        'Еврейская автономная область',
        'Ненецкий автономный округ', 'Ханты-Мансийский автономный округ -
Югра', 'Чукотский автономный округ', 'Ямало-Ненецкий автономный округ',
        'Республика Крым', 'г. Севастополь', 'Иные территории, включая город и
космодром Байконур'] AS reg) ARRAY
    JOIN reg AS regions_name,
        code_ok AS region_code
    GROUP BY 1,
        2),
    oid_data AS
(SELECT dt,
    oid,
    MIN(region) AS region_oid,
    MAX(step1_validate) AS step1_validate_oid,
    MIN(step1_dt) AS step1_dt_oid,
    MIN(step1_result) AS step1_result_oid,
    MIN(step1_text) AS step1_text_oid,
    MAX(step2_mo) AS step2_mo_oid,
    MIN(step2_dt) AS step2_dt_oid,
    MIN(step2_result) AS step2_result_oid,
    MIN(step2_text) AS step2_text_oid,
    MAX(step3_specialist) AS step3_specialist_oid,
    MIN(step3_dt) AS step3_dt_oid,
    MIN(step3_result) AS step3_result_oid,
    MIN(step3_text) AS step3_text_oid,
    MAX(step4_resource) AS step4_resource_oid,
    MIN(step4_dt) AS step4_dt_oid,
    MIN(step4_result) AS step4_result_oid,
    MIN(step4_text) AS step4_text_oid,
    MAX(step5_slots) AS step5_slots_oid,
    MIN(step5_dt) AS step5_dt_oid,
    groupArray(step5_result) AS step5_result_oid,
    MIN(step5_text) AS step5_text_oid,
    MAX(step6_booking) AS step6_booking_oid,
    MIN(step6_dt) AS step6_dt_oid,
    groupArray(step6_result) AS step6_result_oid,
    MIN(step6_text) AS step6_text_oid
FROM zdrav.ssess_funnel
WHERE region_code =
    (SELECT region_code
    FROM reg_dict
    WHERE regions_name = 'Республика Башкортостан')
AND step1_validate = 1
AND service = '600204'
AND dt BETWEEN parseDateTimeBestEffort('2023-02-28T00:00:00') AND
parseDateTimeBestEffort('2023-03-01T00:00:00')
GROUP BY dt,
    oid) SELECT "Этапы воронки",
        "Мера"
FROM
(SELECT [
    'Успешно идентифицированы (%)',
    'Получен список медорганизаций (%)',
    'Получен список специальностей (%)',

```

```

'Получен список фамилий врачей / кабинетов (%)',
'Получен список доступных слотов (%)',
'Успешно записались (%)'] AS funnel,
[
SUM(if(step1_validate_oid = 1 and step1_result_oid = 0,1,0)) / COUNT(*),
SUM(if(step2_mo_oid = 1 and step2_result_oid = 0,1,0)) / COUNT(*),
SUM(if(step3_specialist_oid = 1 and step3_result_oid = 0,1,0)) / COUNT(*),
SUM(if(step4_resource_oid = 1 and step4_result_oid = 0,1,0)) / COUNT(*),
SUM(if(step5_slots_oid = 1 and hasAny(step5_result_oid,[0,201]),1,
0)) / COUNT(*),
SUM(if(step6_booking_oid = 1
and hasAny(step6_result_oid, [0,201]), 1, 0)) /
COUNT(*) ] AS value
FROM oid_data) ARRAY
JOIN funnel AS "Этапы воронки",
value AS "Мера"
GROUP BY 1,
2) AS virtual_table
GROUP BY "Этапы воронки"
ORDER BY "AVG(Мера)" DESC
LIMIT 10;

```

Пример скрипта расчета результатов для шага Идентификация пациента по пользователям для Республики Башкортостан

```

SELECT "Название" AS "Название",
"Параметр" AS "Параметр",
AVG("Значение") AS "AVG(Значение)"
FROM
(WITH reg_dict AS
(SELECT region_code,
regions_name
FROM
(SELECT ['00','79',      '80',  '81',  '84',  '82',  '26',  '83',  '85',  '91',  '86',
'87',  '88',  '89',
'98',  '90',  '92',  '93',  '94',  '95',  '96',  '97',  '01',  '03',  '04',
'05',  '07',  '08',
'10',  '11',  '12',  '14',  '15',  '17',  '18',  '19',  '20',  '24',  '25',
'27',  '29',  '30',
'32',  '33',  '34',  '37',  '38',  '41',  '42',  '44',  '46',  '47',  '22',
'49',  '50',  '52',
'53',  '54',  '56',  '57',  '58',  '60',  '61',  '36',  '63',  '64',  '65',
'66',  '68',  '28',
'69',  '70',  '71',  '73',  '75',  '76',  '78',  '45',  '40',  '99',  '111',
'711', '77',  '7114', '35',  '67',  '55'] AS code_ok,
['ФМБА','Республика Адыгея (Адыгея)', 'Республика Башкортостан', 'Республика
Бурятия', 'Республика Алтай', 'Республика Дагестан',
'Республика Ингушетия', 'Кабардино-Балкарская Республика', 'Республика
Калмыкия', 'Карачаево-Черкесская Республика', 'Республика Карелия',
'Республика Коми', 'Республика Марий Эл', 'Республика Мордовия',
'Республика Саха (Якутия)', 'Республика Северная Осетия - Алания',
'Республика Татарстан (Татарстан)', 'Республика Тыва', 'Удмуртская
Республика', 'Республика Хакасия', 'Чеченская Республика',

```

'Чувашская Республика - Чувашия', 'Алтайский край', 'Краснодарский край', 'Красноярский край', 'Приморский край',
 'Ставропольский край', 'Хабаровский край', 'Амурская область',
 'Архангельская область',
 'Астраханская область', 'Белгородская область', 'Брянская область',
 'Владимирская область', 'Волгоградская область',
 'Вологодская область', 'Воронежская область', 'Ивановская область',
 'Иркутская область', 'Калининградская область',
 'Калужская область', 'Камчатский край', 'Кемеровская область', 'Кировская область',
 'Костромская область', 'Курганская область',
 'Курская область', 'Ленинградская область', 'Липецкая область',
 'Магаданская область', 'Московская область', 'Мурманская область',
 'Нижегородская область', 'Новгородская область', 'Новосибирская область',
 'Омская область', 'Оренбургская область',
 'Орловская область', 'Пензенская область', 'Пермский край', 'Псковская область',
 'Ростовская область', 'Рязанская область',
 'Самарская область', 'Саратовская область', 'Сахалинская область',
 'Свердловская область', 'Смоленская область', 'Тамбовская область',
 'Тверская область', 'Томская область', 'Тульская область', 'Тюменская область',
 'Ульяновская область', 'Челябинская область',
 'Забайкальский край', 'Ярославская область', 'г. Москва', 'г. Санкт-Петербург',
 'Еврейская автономная область',
 'Ненецкий автономный округ', 'Ханты-Мансийский автономный округ - Югра',
 'Чукотский автономный округ', 'Ямало-Ненецкий автономный округ',
 'Республика Крым', 'г. Севастополь', 'Иные территории, включая город и космодром Байконур'] AS reg) ARRAY

```
JOIN reg AS regions_name,
  code_ok AS region_code
GROUP BY 1,
  2),
oid_data AS
(SELECT dt,
  oid,
  MIN(region) AS region_oid,
  MAX(step1_validate) AS step1_validate_oid,
  MIN(step1_dt) AS step1_dt_oid,
  MIN(step1_result) AS step1_result_oid,
  MIN(step1_text) AS step1_text_oid,
  MAX(step2_mo) AS step2_mo_oid,
  MIN(step2_dt) AS step2_dt_oid,
  MIN(step2_result) AS step2_result_oid,
  MIN(step2_text) AS step2_text_oid,
  MAX(step3_specialist) AS step3_specialist_oid,
  MIN(step3_dt) AS step3_dt_oid,
  MIN(step3_result) AS step3_result_oid,
  MIN(step3_text) AS step3_text_oid,
  MAX(step4_resource) AS step4_resource_oid,
  MIN(step4_dt) AS step4_dt_oid,
  MIN(step4_result) AS step4_result_oid,
  MIN(step4_text) AS step4_text_oid,
  MAX(step5_slots) AS step5_slots_oid,
  MIN(step5_dt) AS step5_dt_oid,
  groupArray(step5_result) AS step5_result_oid,
  MIN(step5_text) AS step5_text_oid,
  MAX(step6_booking) AS step6_booking_oid,
```



```

MIN(step6_dt) AS step6_dt_oid,
groupArray(step6_result) AS step6_result_oid,
MIN(step6_text) AS step6_text_oid
FROM zdrav.ssess_funnel
WHERE region_code =
  (SELECT region_code
   FROM reg_dict
   WHERE regions_name = 'Республика Башкортостан')
AND step1_validate = 1
AND service = '600204'
AND dt BETWEEN parseDateTimeBestEffort('2023-02-28T00:00:00') AND
parseDateTimeBestEffort('2023-03-01T00:00:00')
GROUP BY dt,
oid) SELECT "Название",
"Параметр",
"Значение"
FROM
(SELECT ['','',''] AS name,
[
'1. Идентифицированы (%)',
-- '2. Ошибка ЕПГУ (%)',
'2. Техническая ошибка РМИС (%)',
'3. Организационная ошибка РМИС (%)',
-- '5. Ошибка ПОДД (%)',
'4. Ушедшие пользователи (%)'] AS funnel,
[
SUM(if(step1_validate_oid = 1 and step1_result_oid = 0,1,0)) / COUNT(*),
SUM(if(step1_result_oid = 101,1,0)) / COUNT(*),
SUM(if(step1_result_oid = 6,1,0)) / COUNT(*),
(COUNT(*) - SUM(if(step1_result_oid != 999,1,0))) / COUNT(*) ] AS value
FROM oid_data) ARRAY
JOIN name AS "Название",
funnel AS "Параметр",
value AS "Значение"
GROUP BY 1,
2,
3) AS virtual_table
GROUP BY "Название",
"Параметр"
ORDER BY "AVG(Значение)" DESC
LIMIT 1000;

```

Пример скрипта для расчета количества ошибок по описанию для шага Идентификация пациента по пользователя для Республики Башкортостан

```

SELECT "Описание" AS "Описание",
AVG("Количество") AS "AVG(Количество)"
FROM
(WITH reg_dict AS
(SELECT region_code,
regions_name
FROM
(SELECT ['00','79', '80', '81', '84', '82', '26', '83', '85', '91', '86',
'87', '88', '89',

```



```

'05', '98', '90', '92', '93', '94', '95', '96', '97', '01', '03', '04',
'07', '08',
'10', '11', '12', '14', '15', '17', '18', '19', '20', '24', '25',
'27', '29', '30',
'32', '33', '34', '37', '38', '41', '42', '44', '46', '47', '22',
'49', '50', '52',
'53', '54', '56', '57', '58', '60', '61', '36', '63', '64', '65',
'66', '68', '28',
'69', '70', '71', '73', '75', '76', '78', '45', '40', '99', '111',
'711', '77', '7114', '35', '67', '55'] AS code_ok,
['ФМБА', 'Республика Адыгея (Адыгея)', 'Республика Башкортостан', 'Республика
Бурятия', 'Республика Алтай', 'Республика Дагестан',
'Республика Ингушетия', 'Кабардино-Балкарская Республика', 'Республика
Калмыкия', 'Карачаево-Черкесская Республика', 'Республика Карелия',
'Республика Коми', 'Республика Марий Эл', 'Республика Мордовия',
'Республика Саха (Якутия)', 'Республика Северная Осетия - Алания',
'Республика Татарстан (Татарстан)', 'Республика Тыва', 'Удмуртская
Республика', 'Республика Хакасия', 'Чеченская Республика',
'Чувашская Республика - Чувашия', 'Алтайский край', 'Краснодарский
край', 'Красноярский край', 'Приморский край',
'Ставропольский край', 'Хабаровский край', 'Амурская область',
'Архангельская область',
'Астраханская область', 'Белгородская область', 'Брянская область',
'Владимирская область', 'Волгоградская область',
'Вологодская область', 'Воронежская область', 'Ивановская область',
'Иркутская область', 'Калининградская область',
'Калужская область', 'Камчатский край', 'Кемеровская область', 'Кировская
область', 'Костромская область', 'Курганская область',
'Курская область', 'Ленинградская область', 'Липецкая область',
'Магаданская область', 'Московская область', 'Мурманская область',
'Нижегородская область', 'Новгородская область', 'Новосибирская
область', 'Омская область', 'Оренбургская область',
'Орловская область', 'Пензенская область', 'Пермский край', 'Псковская
область', 'Ростовская область', 'Рязанская область',
'Самарская область', 'Саратовская область', 'Сахалинская область',
'Свердловская область', 'Смоленская область', 'Тамбовская область',
'Тверская область', 'Томская область', 'Тульская область', 'Тюменская
область', 'Ульяновская область', 'Челябинская область',
'Забайкальский край', 'Ярославская область', 'г. Москва', 'г. Санкт-Петербург',
'Еврейская автономная область',
'Ненецкий автономный округ', 'Ханты-Мансийский автономный округ -
Югра', 'Чукотский автономный округ', 'Ямало-Ненецкий автономный округ',
'Республика Крым', 'г. Севастополь', 'Иные территории, включая город и
космодром Байконур'] AS reg) ARRAY
JOIN reg AS regions_name,
code_ok AS region_code
GROUP BY 1,
2),
oid_data AS
(SELECT dt,
oid,
MIN(region) AS region_oid,
MAX(step1_validate) AS step1_validate_oid,
MIN(step1_dt) AS step1_dt_oid,
MIN(step1_result) AS step1_result_oid,

```

```

MIN(step1_text) AS step1_text_oid,
MAX(step2_mo) AS step2_mo_oid,
MIN(step2_dt) AS step2_dt_oid,
MIN(step2_result) AS step2_result_oid,
MIN(step2_text) AS step2_text_oid,
MAX(step3_specialist) AS step3_specialist_oid,
MIN(step3_dt) AS step3_dt_oid,
MIN(step3_result) AS step3_result_oid,
MIN(step3_text) AS step3_text_oid,
MAX(step4_resource) AS step4_resource_oid,
MIN(step4_dt) AS step4_dt_oid,
MIN(step4_result) AS step4_result_oid,
MIN(step4_text) AS step4_text_oid,
MAX(step5_slots) AS step5_slots_oid,
MIN(step5_dt) AS step5_dt_oid,
groupByArray(step5_result) AS step5_result_oid,
MIN(step5_text) AS step5_text_oid,
MAX(step6_booking) AS step6_booking_oid,
MIN(step6_dt) AS step6_dt_oid,
groupByArray(step6_result) AS step6_result_oid,
MIN(step6_text) AS step6_text_oid
FROM zdrav.ssess_funnel
WHERE region_code =
  (SELECT region_code
   FROM reg_dict
   WHERE regions_name = 'Республика Башкортостан')
AND step1_validate = 1
AND service = '600204'
AND dt BETWEEN parseDateTimeBestEffort('2023-02-28T00:00:00') AND
parseDateTimeBestEffort('2023-03-01T00:00:00')
GROUP BY dt,
  oid) SELECT if(length(step1_text_oid)>150, substringUTF8(step1_text_oid, 1, 120),
step1_text_oid) AS "Описание",
COUNT(*) AS "Количество"
FROM oid_data
WHERE step1_validate_oid = 1
AND step1_result_oid != 0
GROUP BY 1
ORDER BY "Количество" DESC) AS virtual_table
GROUP BY "Описание"
ORDER BY "AVG(Количество)" DESC
LIMIT 100;

```

Приложение Ж.5

Расчет целевой конверсии

```

with reg_dict as (
  SELECT
    code,
    reg_name
  FROM
    (
      SELECT
        ['00','79', '80', '81', '84', '82', '26', '83', '85', '91', '86', '87', '88', '89',
        '98', '90', '92', '93', '94', '95', '96', '97', '01', '03', '04', '05', '07', '08',
        '10', '11', '12', '14', '15', '17', '18', '19', '20', '24', '25', '27', '29', '30',
        '32', '33', '34', '37', '38', '41', '42', '44', '46', '47', '22', '49', '50', '52',
        '53', '54', '56', '57', '58', '60', '61', '36', '63', '64', '65', '66', '68', '28',
        '69', '70', '71', '73', '75', '76', '78', '45', '40', '99', '111', '711', '77', '7114', '35', '67',
        '55'] AS code_ok,
        ['ФМБА','Республика Адыгея (Адыгея)', 'Республика Башкортостан', 'Республика Бурятия',
        'Республика Алтай', 'Республика Дагестан',
        'Республика Ингушетия', 'Кабардино-Балкарская Республика', 'Республика Калмыкия',
        'Карачаево-Черкесская Республика', 'Республика Карелия',
        'Республика Коми', 'Республика Марий Эл', 'Республика Мордовия', 'Республика Саха (Якутия)',
        'Республика Северная Осетия - Алания',
        'Республика Татарстан (Татарстан)', 'Республика Тыва', 'Удмуртская Республика', 'Республика
        Хакасия', 'Чеченская Республика',
        'Чувашская Республика - Чувашия', 'Алтайский край', 'Краснодарский край', 'Красноярский
        край', 'Приморский край',
        'Ставропольский край', 'Хабаровский край', 'Амурская область', 'Архангельская область',
        'Астраханская область', 'Белгородская область', 'Брянская область', 'Владимирская область',
        'Волгоградская область',
        'Вологодская область', 'Воронежская область', 'Ивановская область', 'Иркутская область',
        'Калининградская область',
        'Калужская область', 'Камчатский край', 'Кемеровская область', 'Кировская область',
        'Костромская область', 'Курганская область',
        'Курская область', 'Ленинградская область', 'Липецкая область', 'Магаданская область',
        'Московская область', 'Мурманская область',
        'Нижегородская область', 'Новгородская область', 'Новосибирская область', 'Омская область',
        'Оренбургская область',
        'Орловская область', 'Пензенская область', 'Пермский край', 'Псковская область', 'Ростовская
        область', 'Рязанская область',
        'Самарская область', 'Саратовская область', 'Сахалинская область', 'Свердловская область',
        'Смоленская область', 'Тамбовская область',
        'Тверская область', 'Томская область', 'Тульская область', 'Тюменская область', 'Ульяновская
        область', 'Челябинская область',
        'Забайкальский край', 'Ярославская область', 'г. Москва', 'г. Санкт-Петербург', 'Еврейская
        автономная область',
        'Ненецкий автономный округ', 'Ханты-Мансийский автономный округ - Югра', 'Чукотский
        автономный округ', 'Ямало-Ненецкий автономный округ',
        'Республика Крым', 'г. Севастополь', 'Иные территории, включая город и космодром Байконур']
      AS reg
    )
  ARRAY JOIN
    reg AS reg_name,
    code_ok AS code
  GROUP BY
    1,2
),

current_goals as (
  select

```

```

    region,
    current_goal
from (
    select
        ['Кабардино-Балкарская Республика','Республика Дагестан','Ямало-Ненецкий автономный
округ','Брянская область','Республика Башкортостан','Омская область',
        'Калужская область','Республика Карелия','Пензенская область','Астраханская область','Белгородская
область','Вологодская область',
        'Удмуртская Республика','Курганская область','Нижегородская область','Оренбургская
область','Владимирская область',
        'Пермский край','Кировская область','Республика Крым','Республика Саха (Якутия)','Республика
Хакасия'] as region,
        [46,38,33,30,28,28,27,27,26,26,26,26,25,25,24,23,22,22,31,29,28,27] as current_goal
    )
ARRAY JOIN
    region,
    current_goal
GROUP BY 1, 2
),

users_cnt as (
    select
        multiIf(
            substring(sr.data_data1, 1, 4) in ('7114', '7115', '7116', '7117'), '7114',
            substring(sr.data_data1, 1, 3) in ('111', '711'), substring(sr.data_data1, 1, 3),
            substring(sr.data_data1, 1, 2)
        ) as region_code,
        cast(tmp.dateTime as date) as dt,
        count(distinct tmp.oid) as users

    from zdrav.validate_patient tmp
        ANY LEFT JOIN zdrav.search_region sr
            ON sr.session = tmp.session

    where tmp.service = '10700'
        and tmp.dateTime between '{{ period.start }}' and '{{ period.end }}'

    group by 1, 2
),

successes as (
    select
        multiIf(
            substring(sr.data_data1, 1, 4) in ('7114', '7115', '7116', '7117'), '7114',
            substring(sr.data_data1, 1, 3) in ('111', '711'), substring(sr.data_data1, 1, 3),
            substring(sr.data_data1, 1, 2)
        ) as region_code,
        cast(tmp.dateTime as date) as dt,
        count(distinct tmp.oid) as users

    from zdrav.request_booking tmp
        ANY LEFT JOIN zdrav.search_region sr
            ON sr.session = tmp.session

    where tmp.service = '10700'
        and tmp.data_result = 0
        and tmp.dateTime between '{{ period.start }}' and '{{ period.end }}'

    group by 1, 2
)

```

```

select
  region,
  current_goal,
  floor(goal_conversion_rate, 0) goal_conversion_rate,
  floor(goal_conversion_rate - current_goal, 0) diff,
  observations

from (
  select
    r.reg_name region,
    avg(g.current_goal) current_goal,
    quantile({{ percentile_% }} / 100)(round(s.users / u.users * 100, 4)) goal_conversion_rate,
    -- quantile({{ percentile_% }} / 100)(round(s.users / u.users * 100, 4)) - avg(g.current_goal) diff,
    count(s.users) observations
  from users_cnt u
  left join successes s on s.dt = u.dt and s.region_code = u.region_code
  left join reg_dict r on u.region_code = r.code
  left join current_goals g on r.reg_name = g.region
  where
    round(s.users / u.users * 100, 4) > '{{ min_cr_% }}'
    and length(r.reg_name) > 1
    and u.users >= '{{ min_users }}'
  group by 1
)

order by 4 desc

```

Приложение Ж.6
Скрипт для определения доли пациентов в регионе, имеющих актуальное
прикрепление к МО

Приложение содержит скрипт для определения показателей качества данных согласно пункту Ж.4.5.3.1 на примере Калужской области.

```
select count(*)  
from (  
  select distinct p.id, max(m.id) mo_id  
  from misdm40.patient p  
    left outer join misdm40.attachment a on (a.patient_id = p.id and a.status = 'DONE')  
    left outer join misdm40.mo m on (m.id = a.mo_id and m.enabled <> 'NONE')  
  group by p.id)  
where mo_id is not null;
```

Приложение Ж.7

Критерии оценки «плохо» в разрезе врачей

Приложение содержит критерии оценки «плохо» за несвоевременное размещение в витрине информации о слотах, доступных для ЕПГУ в разрезе врачебных специальностей

Врачебная специальность	Критерий оценки «плохо» за несвоевременность размещения в витрине информации о слотах, доступных на ЕПГУ по данной специальности
ВРАЧ-ТЕРАПЕВТ УЧАСТКОВЫЙ ВРАЧ ОБЩЕЙ ПРАКТИКИ (СЕМЕЙНЫЙ ВРАЧ) ВРАЧ-ПЕДИАТР УЧАСТКОВЫЙ	В витрине не размещена информация о слотах, доступных на ЕПГУ, на даты, отстающие от даты оценки на 24 часа (без учета выходных дней).
ВРАЧ-ОТОЛАРИНГОЛОГ ВРАЧ-АКУШЕР-ГИНЕКОЛОГ ВРАЧ-ТЕРАПЕВТ ВРАЧ-ПЕДИАТР ВРАЧ-АКУШЕР-ГИНЕКОЛОГ ВРАЧ-ОТОРИНОЛАРИНГОЛОГ ВРАЧ-ОФТАЛЬМОЛОГ ВРАЧ - ДЕТСКИЙ УРОЛОГ-АНДРОЛОГ ВРАЧ-УРОЛОГ ВРАЧ - ДЕТСКИЙ ХИРУРГ ВРАЧ-ХИРУРГ ВРАЧ-ПСИХИАТР ДЕТСКИЙ ВРАЧ-СТОМАТОЛОГ-ТЕРАПЕВТ ВРАЧ-ПСИХИАТР ДЕТСКИЙ УЧАСТКОВЫЙ ВРАЧ-ПСИХИАТР ПОДРОСТКОВЫЙ ВРАЧ-ПСИХИАТР ПОДРОСТКОВЫЙ УЧАСТКОВЫЙ ВРАЧ-СТОМАТОЛОГ ВРАЧ-СТОМАТОЛОГ ДЕТСКИЙ ВРАЧ-СТОМАТОЛОГ-ХИРУРГ ВРАЧ-НЕВРОЛОГ ВРАЧ-КАРДИОЛОГ ВРАЧ - ДЕТСКИЙ КАРДИОЛОГ ВРАЧ - ДЕТСКИЙ ОНКОЛОГ ВРАЧ-ОНКОЛОГ ВРАЧ-ФТИЗИАТР ВРАЧ-ПСИХИАТР-НАРКОЛОГ ВРАЧ-ДЕТСКИЙ ХИРУРГ ВРАЧ-ПСИХИАТР	В витрине своевременно не размещена информация о слотах, доступных на ЕПГУ на даты, отстающие от даты оценки на 0 - 14 календарных дней (хотя бы один слот в указанный промежуток времени)

Приложение Ж.8

Скрипт для определения доли доступных врачей специальности в медицинских организациях региона

Приложение содержит скрипт для определения показателей качества данных согласно пункту Ж.4.5.3.2 на примере Калужской области.

```

select mo_name,
       rname,
       sum(valD) / sum(valS) * 100.0 postPercent,
       sum(valS)          sumValS,
       sum(valD)          sumValD,
       sum(valD) = 0      isIncident,
       cast(current_date as timestamp) AS loading_dt
from (select trim(substring(m.name, 1, position(',') in m.name || ',') - 1)) as mo_name,
       r.rname,
       a.valD * 1.0          valD,
       a.valS * 1.0          valS
from (select mo_id, post_id, sum(case when slot_exists is not null then 1 else 0 end) valD, count(1)
valS
from (select r.mo_id, r.id, r.post_id, max(s.id) slot_exists
from misdm40.resource r
left outer join misdm40.slot s on (s.resource_id = r.id and s.status = 'AVAILABLE'
and s.visittime > cast(current_date as timestamp)
and s.visittime <
timestampadd(day, case /*для субботы +3 дня, для ВС +2, для остальных дней +1
день */
when cast(extract(dow from cast(current_date as timestamp)) as int) = 6
and
r.post_id in ('110', '49', '59') then 3
when cast(extract(dow from cast(current_date as timestamp)) as int) = 0
and
r.post_id in ('110', '49', '59') then 2
when cast(extract(dow from cast(current_date as timestamp)) as int) <= 5
and
r.post_id in ('110', '49', '59') then 1
else 14 end,

```



```

        cast(current_date as timestamp) ))
    where r.enabled IN ('ATTACHED_MO','ATTACHED_REG','ALL')
and r.post_id in
    ('81', '87', '119', '50', '26', '25', '34', '45', '104', '101', '86', '85', '84',
    '103', '83', '122', '28', '117', '27', '54', '53', '58', '109', '13', '59', '49',
    '110','100')
    group by r.mo_id, r.id, r.post_id)
group by mo_id, post_id) a,
misd40.mo m,
(select max('ВРАЧ-ТЕРАПЕВТ УЧАСТКОВЫЙ') rname, max('110') rid
from misd40.mo
union all
select max('ВРАЧ-СТОМАТОЛОГ') rname, max('100') rid
from misd40.mo
union all
select max('ВРАЧ ОБЩЕЙ ПРАКТИКИ (СЕМЕЙНЫЙ ВРАЧ)') rname, max('49') rid
from misd40.mo
union all
select max('ВРАЧ-ПЕДИАТР УЧАСТКОВЫЙ') rname, max('59') rid
from misd40.mo
union all
select max('ВРАЧ-АКУШЕР-ГИНЕКОЛОГ') rname, max('13') rid
from misd40.mo
union all
select max('ВРАЧ-ТЕРАПЕВТ') rname, max('109') rid
from misd40.mo
union all
select max('ВРАЧ-ПЕДИАТР') rname, max('58') rid
from misd40.mo
union all
select max('ВРАЧ-ОТОРИНОЛАРИНГОЛОГ') rname, max('53') rid
from misd40.mo
union all

```

```

select max('ВРАЧ-ОФТАЛЬМОЛОГ') rname, max('54') rid
from misdm40.mo
union all
select max('ВРАЧ - ДЕТСКИЙ УРОЛОГ-АНДРОЛОГ') rname, max('27') rid
from misdm40.mo
union all
select max('ВРАЧ-УРОЛОГ') rname, max('117') rid
from misdm40.mo
union all
select max('ВРАЧ - ДЕТСКИЙ ХИРУРГ') rname, max('28') rid
from misdm40.mo
union all
select max('ВРАЧ-ХИРУРГ') rname, max('122') rid
from misdm40.mo
union all
select max('ВРАЧ-ПСИХИАТР ДЕТСКИЙ') rname, max('83') rid
from misdm40.mo
union all
select max('ВРАЧ-СТОМАТОЛОГ-ТЕРАПЕВТ') rname, max('103') rid
from misdm40.mo
union all
select max('ВРАЧ-ПСИХИАТР ДЕТСКИЙ УЧАСТКОВЫЙ') rname, max('84') rid
from misdm40.mo
union all
select max('ВРАЧ-ПСИХИАТР ПОДРОСТКОВЫЙ') rname, max('85') rid
from misdm40.mo
union all
select max('ВРАЧ-ПСИХИАТР ПОДРОСТКОВЫЙ УЧАСТКОВЫЙ') rname, max('86') rid
from misdm40.mo
union all
select max('ВРАЧ-СТОМАТОЛОГ ДЕТСКИЙ') rname, max('101') rid
from misdm40.mo
union all

```

```

select max('ВРАЧ-СТОМАТОЛОГ-ХИРУРГ') rname, max('104') rid
from misdm40.mo
union all
select max('ВРАЧ-НЕВРОЛОГ') rname, max('45') rid
from misdm40.mo
union all
select max('ВРАЧ-КАРДИОЛОГ') rname, max('34') rid
from misdm40.mo
union all
select max('ВРАЧ - ДЕТСКИЙ КАРДИОЛОГ') rname, max('25') rid
from misdm40.mo
union all
select max('ВРАЧ - ДЕТСКИЙ ОНКОЛОГ') rname, max('26') rid
from misdm40.mo
union all
select max('ВРАЧ-ОНКОЛОГ') rname, max('50') rid
from misdm40.mo
union all
select max('ВРАЧ-ФТИЗИАТР') rname, max('119') rid
from misdm40.mo
union all
select max('ВРАЧ-ПСИХИАТР-НАРКОЛОГ') rname, max('87') rid
from misdm40.mo
union all
select max('ВРАЧ-ПСИХИАТР') rname, max('81') rid
from misdm40.mo) r
where m.id = a.mo_id
and r.rid = a.post_id)
group by mo_name, rname

```

Приложение Ж.9

Скрипт для определения доли слотов врачебных специальностей в расписании медицинских организаций, размещаемых для ЕПГУ

Приложение содержит скрипта для определения показателей качества данных согласно пункту Ж.4.5.3.3 на примере Калужской области.

```
select mo_name, rname, sum(epgu_slots) / sum(total_slots) * 100.0 egpuPercent
from (select trim(substring(m.name, 1, position(',') in m.name || ',')) as mo_name,
      r.rname,
      egpu_slots * 1.0,
      total_slots * 1.0,
      from (select r.mo_id,
        r.post_id,
        sum(case when s.tag_type = 'EPGU' OR s.tag_type = 'ORDINARY' then 1 else 0 end) egpu_s
        lots,
        count(1)
        total_slots
      from misdm40.resource r, misdm40.slot s
      where s.resource_id = r.id
      and s.status = 'AVAILABLE'
      and s.visittime >= current_date
      and r.post_id in
        ('81', '87', '119', '50', '26', '25', '34', '45', '104', '101', '86', '85', '84',
        '103', '83', '122', '28', '117', '27', '54', '53', '58', '109', '13', '59', '49',
        '110')
      group by r.mo_id, r.post_id) a, misdm40.mo m,
(select max('ВРАЧ-ТЕРАПЕВТ УЧАСТКОВЫЙ') rname, max('110') rid
from misdm40.mo
union all
select max('ВРАЧ ОБЩЕЙ ПРАКТИКИ (СЕМЕЙНЫЙ ВРАЧ)') rname, max('49') rid
from misdm40.mo
union all
select max('ВРАЧ-ПЕДИАТР УЧАСТКОВЫЙ') rname, max('59') rid
from misdm40.mo
union all
select max('ВРАЧ-АКУШЕР-ГИНЕКОЛОГ') rname, max('13') rid
from misdm40.mo
union all
select max('ВРАЧ-ТЕРАПЕВТ') rname, max('109') rid
from misdm40.mo
union all
select max('ВРАЧ-ПЕДИАТР') rname, max('58') rid
from misdm40.mo
union all
select max('ВРАЧ-АКУШЕР-ГИНЕКОЛОГ') rname, max('13') rid
from misdm40.mo
union all
select max('ВРАЧ-ОТОРИНОЛАРИНГОЛОГ') rname, max('53') rid
```

```

from misdm40.mo
union all
select max('ВРАЧ-ОФТАЛЬМОЛОГ') rname, max('54') rid
from misdm40.mo
union all
select max('ВРАЧ - ДЕТСКИЙ УРОЛОГ-АНДРОЛОГ') rname, max('27') rid
from misdm40.mo
union all
select max('ВРАЧ-УРОЛОГ') rname, max('117') rid
from misdm40.mo
union all
select max('ВРАЧ - ДЕТСКИЙ ХИРУРГ') rname, max('28') rid
from misdm40.mo
union all
select max('ВРАЧ-ХИРУРГ') rname, max('122') rid
from misdm40.mo
union all
select max('ВРАЧ-ПСИХИАТР ДЕТСКИЙ') rname, max('83') rid
from misdm40.mo
union all
select max('ВРАЧ-СТОМАТОЛОГ-ТЕРАПЕВТ') rname, max('103') rid
from misdm40.mo
union all
select max('ВРАЧ-ПСИХИАТР ДЕТСКИЙ УЧАСТКОВЫЙ') rname, max('84') rid
from misdm40.mo
union all
select max('ВРАЧ-ПСИХИАТР ПОДРОСТКОВЫЙ') rname, max('85') rid
from misdm40.mo
union all
select max('ВРАЧ-ПСИХИАТР ПОДРОСТКОВЫЙ УЧАСТКОВЫЙ') rname, max('86') rid
from misdm40.mo
union all
select max('ВРАЧ-СТОМАТОЛОГ ДЕТСКИЙ') rname, max('101') rid
from misdm40.mo
union all
select max('ВРАЧ-СТОМАТОЛОГ-ХИРУРГ') rname, max('104') rid
from misdm40.mo
union all
select max('ВРАЧ-НЕВРОЛОГ') rname, max('45') rid
from misdm40.mo
union all
select max('ВРАЧ-КАРДИОЛОГ') rname, max('34') rid
from misdm40.mo
union all
select max('ВРАЧ - ДЕТСКИЙ КАРДИОЛОГ') rname, max('25') rid
from misdm40.mo
union all
select max('ВРАЧ - ДЕТСКИЙ ОНКОЛОГ') rname, max('26') rid

```

```

from misdm40.mo
union all
select max('ВРАЧ-ОНКОЛОГ') rname, max('50') rid
from misdm40.mo
union all
select max('ВРАЧ-ФТИЗИАТР') rname, max('119') rid
from misdm40.mo
union all
select max('ВРАЧ-ПСИХИАТР-НАРКОЛОГ') rname, max('87') rid
from misdm40.mo
union all
select max('ВРАЧ - ДЕТСКИЙ ХИРУРГ') rname, max('28') rid
from misdm40.mo
union all
select max('ВРАЧ-ПСИХИАТР') rname, max('81') rid
from misdm40.mo) r
where m.id = a.mo_id
and r.rid = a.post_id)
group by mo_name, rname;

```

Приложение Ж.10

Скрипт для определения количества слотов, доступных для записи с ЕПГУ на ближайшие 14 дней

Приложение содержит скрипта для определения показателей качества данных согласно пункту Ж.4.5.3.4 на примере Калужской области.

```

select
  m_res.post_id,
  m_res.post_name,
  coalesce(m_res.parent_id, m_att.parent_id) med,
  sum(patients) patients,
  max(m_res.doctors) doctors,
  max(m_res.slots) slot_count,
  max(m_res.epgu_slots) epgu_slot_count,
  max(m_res.all_slots) all_slot_count,
  max(m_res.all_epgu_slots) all_epgu_slot_count
from (
  select
    t.parent_id,
    t.post_id,
    t.post_name,
    true female,
    t.post_id <> '13' male,
    case when t.post_id in ('101','25','27','28','29','7','83','174','58','59') then 'CHILD' else 'ADULT' end age,
    count(distinct t.all_slot_id) all_slots,
    count(distinct t.all_epgu_slot_id) all_epgu_slots,
    count(distinct t.slot_id) slots,
    count(distinct t.epgu_slot_id) epgu_slots,
    count(distinct t.resource_id) doctors
  from (
    select
      mo.parent_id,
      r.post_id,
      p.post_name,
      s.id all_slot_id,
      case when s.tag_type in ('EPGU', 'ORDINARY') then s.id end all_epgu_slot_id,
      case when s.status = 'AVAILABLE' then s.id end slot_id,
      case when s.status = 'AVAILABLE' and s.tag_type in ('EPGU', 'ORDINARY') then s.id end epgu_slot
    _id,
    r.id resource_id
    from misdm40.mo for system_time as of '2022-10-27 08:00:00' mo
    left join misdm40.resource for system_time as of '2022-10-27 08:00:00' r on r.mo_id = mo.id
      and r."type" = 'SPECIALIST' and r.enabled in ('ATTACHED_REG', 'ATTACHED_MO', 'ALL')
    left join misdm40.slot for system_time as of '2022-10-27 08:00:00' s on s.resource_id = r.id
      and s.visittime >= '2022-10-27' and s.visittime < '2022-11-10'
    left join (
      select post_id, post_name from (

```

```

select post_id, post_name, row_number() over(partition by post_id order by count(*) desc) rn
  from misdm40.resource for system_time as of '2022-10-27 08:00:00' r
 group by post_id, post_name
) t
where rn = 1) p on p.post_id = r.post_id
where mo.enabled <> 'NONE'
)t
group by t.parent_id, t.post_id, t.post_name
) m_res
full join (
select mo.parent_id, p.age, p.male, p.female, count(distinct a.patient_id) patients
from misdm40.mo for system_time as of '2022-10-27 08:00:00' mo
join misdm40.attachment for system_time as of '2022-10-
27 08:00:00' a on a.mo_id = mo.id and a.status = 'DONE'
join (
select
  id,
  case when timestampadd(year, 18, birthdate) < '2022-10-27' then 'ADULT' else 'CHILD' end age,
  gender = 'MALE' male,
  gender = 'FEMALE' female
from misdm40.patient for system_time as of '2022-10-27 08:00:00'
) p on p.id = a.patient_id
where mo.enabled <> 'NONE'
group by mo.parent_id, p.age, p.male, p.female
) m_att
on m_res.parent_id = m_att.parent_id and m_res.age = m_att.age
and ((m_att.male and m_res.male) or (m_att.female and m_res.female))
group by m_res.post_id, m_res.post_name, coalesce(m_res.parent_id, m_att.parent_id)
limit 100000;

```


Приложение Ж.11

Скрипт для определения количества прикреплений пациентов к медицинским организациям в разрезе пола и возраста

Приложение содержит скрипта для определения показателей качества данных согласно пункту Ж.4.5.3.5 на примере Астраханской области.

```

SELECT
MO_name,
COALESCE(male_age1, 0) male_age1,
COALESCE(female_age1, 0) female_age1,
-- COALESCE(nogender_age1, 0) nogender_age1,
COALESCE(male_age2, 0) male_age2,
COALESCE(female_age2, 0) female_age2,
-- COALESCE(nogender_age2, 0) nogender_age2,
COALESCE(male_age3, 0) male_age3,
COALESCE(female_age3, 0) female_age3,
-- COALESCE(nogender_age3, 0) nogender_age3,
COALESCE(male_age4, 0) male_age4,
COALESCE(female_age4, 0) female_age4,
-- COALESCE(nogender_age4, 0) nogender_age4,
COALESCE(male_age5, 0) male_age5,
COALESCE(female_age5, 0) female_age5,
-- COALESCE(nogender_age5, 0) nogender_age5,
COALESCE(total, 0) total
FROM (
SELECT
MO_name,
SUM(cnt_male_age1) male_age1,
SUM(cnt_female_age1) female_age1,
-- SUM(cnt_nogender_age1) nogender_age1,
SUM(cnt_male_age2) male_age2,
SUM(cnt_female_age2) female_age2,
-- SUM(cnt_nogender_age2) nogender_age2,
SUM(cnt_male_age3) male_age3,
SUM(cnt_female_age3) female_age3,
-- SUM(cnt_nogender_age3) nogender_age3,
SUM(cnt_male_age4) male_age4,
SUM(cnt_female_age4) female_age4,
-- SUM(cnt_nogender_age4) nogender_age4,
SUM(cnt_male_age5) male_age5,
SUM(cnt_female_age5) female_age5,
-- SUM(cnt_nogender_age5) nogender_age5,
SUM(cnt_total) total
FROM (
SELECT
MO_name,
CASE

```

```

    WHEN grp = 1 AND gender = 'MALE' THEN cnt
END
cnt_male_age1,
CASE
    WHEN grp = 1 AND gender = 'FEMALE' THEN cnt
END
cnt_female_age1,
--
CASE
    WHEN grp = 1 AND gender NOT LIKE '%MALE' THEN cnt
END
cnt_nogender_age1,
CASE
    WHEN grp = 2 AND gender = 'MALE' THEN cnt
END
cnt_male_age2,
CASE
    WHEN grp = 2 AND gender = 'FEMALE' THEN cnt
END
cnt_female_age2,
--
CASE
    WHEN grp = 2 AND gender NOT LIKE '%MALE' THEN cnt
END
cnt_nogender_age2,
CASE
    WHEN grp = 3 AND gender = 'MALE' THEN cnt
END
cnt_male_age3,
CASE
    WHEN grp = 3 AND gender = 'FEMALE' THEN cnt
END
cnt_female_age3,
--
CASE
    WHEN grp = 3 AND gender NOT LIKE '%MALE' THEN cnt
END
cnt_nogender_age3,
CASE
    WHEN grp = 4 AND gender = 'MALE' THEN cnt
END
cnt_male_age4,
CASE
    WHEN grp = 4 AND gender = 'FEMALE' THEN cnt
END
cnt_female_age4,
--
CASE

```

```

    WHEN grp = 4 AND gender NOT LIKE '%MALE' THEN cnt
END
cnt_nogender_age4,
CASE
    WHEN grp = 5 AND gender = 'MALE' THEN cnt
END
cnt_male_age5,
CASE
    WHEN grp = 5 AND gender = 'FEMALE' THEN cnt
END
cnt_female_age5,
--
CASE
    WHEN grp = 5 AND gender NOT LIKE '%MALE' THEN cnt
ELSE
    0
END
cnt_nogender_age5,
cnt cnt_total
FROM (
    SELECT
        CASE
            WHEN lg_pivot.lg = 1 THEN short_name
        ELSE
            'TOTAL'
        END
    AS MO_name,
    gender,
    grp,
    cnt
FROM (
    SELECT
        short_name,
        gender,
        grp,
        MAX(cnt) AS cnt
    FROM (
        SELECT
            TRIM(SUBSTRING(name, 1, position(',') IN name || ',' - 1)) AS short_name,
            name,
            gender,
            grp,
            COUNT(1) cnt
        FROM (
            SELECT
                a.name,
                CASE
                    WHEN timestampadd(year, 65, CAST(p.birthdate AS timestamp)) < base_date.dt THEN 5

```

```

ELSE
CASE
    WHEN timestampadd(year, 18, CAST(p.birthdate AS timestamp)) < base_date.dt THEN 4
ELSE
CASE
    WHEN timestampadd(year, 5, CAST(p.birthdate AS timestamp)) < base_date.dt THEN 3
ELSE
CASE
    WHEN timestampadd(year, 1, CAST(p.birthdate AS timestamp)) < base_date.dt THEN 2
ELSE
1
END
END
END
END
grp,
p.gender
FROM (
    SELECT
        MAX( -- current_date CAST('2022-09-
01' AS timestamp) ) dt /* Дата, относительно которой считается возраст */
        FROM
            misdm30.mo) base_date,
        misdm30.patient p,
        (
            SELECT
                DISTINCT m.name,
                a.patient_id
            FROM
                misdm30.mo m,
                misdm30.attachment a
            WHERE
                m.id = a.mo_id
                AND a.status = 'DONE'
                AND m.enabled <> 'NONE') a
        WHERE
            a.patient_id = p.id ) DATA
GROUP BY
    name,
    gender,
    grp)
WHERE
1 = 1
GROUP BY
    short_name,
    gender,
    grp),
(

```

```
SELECT
  MAX(1) lg
FROM
  misdm30.mo
UNION ALL
SELECT
  MAX(2) lg
FROM
  misdm30.mo) lg_pivot)
WHERE
  1 = 1 )
GROUP BY
  mo_name);
```

Приложение Ж.12
Справочник регионов

Наименование региона	Код региона
Алтайский край	01
Амурская область	10
Архангельская область	11
Астраханская область	12
Белгородская область	14
Брянская область	15
Владимирская область	17
Волгоградская область	18
Вологодская область	19
Воронежская область	20
Еврейская автономная область	99
Забайкальский край	76
Ивановская область	24
Иркутская область	25
Кабардино-Балкарская Республика	83
Калининградская область	27
Калужская область	29
Камчатский край	30
Карачаево-Черкесская Республика	91
Кемеровская область	32
Кировская область	33
Костромская область	34
Краснодарский край	03
Красноярский край	04
Курганская область	37
Курская область	38
Ленинградская область	41
Липецкая область	42
Магаданская область	44
Москва	45
Московская область	46
Мурманская область	47
Ненецкий автономный округ	111
Нижегородская область	22
Новгородская область	49
Новосибирская область	50
Омская область	52
Оренбургская область	53
Орловская область	54
Пензенская область	56
Пермский край	57
Приморский край	05
Псковская область	58

Республика Адыгея	79
Республика Алтай	84
Республика Башкортостан	80
Республика Бурятия	81
Республика Дагестан	82
Республика Ингушетия	26
Республика Калмыкия	85
Республика Карелия	86
Республика Коми	87
Республика Крым	35
Республика Марий Эл	88
Республика Мордовия	89
Республика Саха (Якутия)	98
Республика Северная Осетия-Алания	90
Республика Татарстан	92
Республика Тыва	93
Республика Хакасия	95
Ростовская область	60
Рязанская область	61
Самарская область	36
Санкт-Петербург	40
Саратовская область	63
Сахалинская область	64
Свердловская область	65
Севастополь	67
Смоленская область	66
Ставропольский край	07
Тамбовская область	68
Тверская область	28
Томская область	69
Тульская область	70
Тюменская область	71
Удмуртская Республика	94
Ульяновская область	73
Хабаровский край	08
Ханты-Мансийский автономный округ - Югра	711
Челябинская область	75
Чеченская Республика	96
Чувашская Республика - Чувашия	97
Чукотский автономный округ	77
Ямало-Ненецкий автономный округ	7114
Ярославская область	78
г. Байконур	55
ФМБА	00

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№	Версия документа	Изменения
1	v8	В Приложении К внесены изменения в пункте «Бронирование записи на прием и отмена», в booking/book добавлены referral_id Идентификатор направления, при записи по направлению и cards_id Идентификатор медицинской карты
2	v9	В Приложение К внесено описание и коды ошибок, возвращаемые витриной региона и отображаемые пользователю на ЕПГУ
3	v10	В Приложении К исправлено описание OpenApi, изменено наименование book_gmis на bookExtId, в ответе на бронирование добавлены не обязательные поля
4	v11	В Приложении К актуализирован список и описание ошибок
5	v12	В Приложении К добавлены комментарии к полям в таблицы
6	v13	Добавлен раздел 4, где описаны рекомендованные и минимальные системные требования для реализации решения «Витрины данных»
7	v14	Исключена допустимость использования ПО «Витрина данных Лайт» (п.4) Внесены изменения в Модель витрины данных (Приложение К) Внесены технические правки по тексту документа
8	v15	Расширен состав услуг, предоставляемых на ЕПГУ посредством Региональной витрины данных субъекта РФ, добавлены услуги: «Запись на прием по направлению», «Запись на прием к врачам, осуществляющим диспансерное наблюдение», «Запись на вакцинацию», «Запись на прием на углубленную диспансеризацию для лиц, переболевших COVID-19». Добавлено общее описание информационного взаимодействия при предоставлении Сервисов на ЕПГУ с использованием типового решения «Витрины данных» НСУД. Внесены технические правки в структуре и по тексту документа.
9	v16	Комментарии по тексту документа

№	Версия документа	Изменения
10	v17	Внесены правки в структуру и содержание документа, добавлено: правовое основание для оказания услуги; информация о текущей схеме взаимодействия, ссылка на метод. рекомендации текущие Внесены изменения в модель данных, таблица validation.
11	v18	В приложения добавлен список правил проверки сведений о расписании и бронировании, осуществляемых в компоненте «Модуль управления бизнес-правилами» ФЭР ЕГИСЗ (приложение 10). Добавлено описание процесса перехода ГИС в сфере здравоохранения субъекта Российской Федерации на решение с использованием технологии витрин данных (п.6)
12	v19	В приложении 3 порядок установки ПО «Витрина данных Лайт» заменён на порядок установки ПО «Витрина данных Медиум» В приложении 4, а также в формах заявок внесены правки, соответствующие справочной и методической информации, размещённой на технологическом. Портале СМЭВ В приложении 10 модель данных расписания заменена на актуальную. Внесены технические правки в структуре и по тексту документа.
13	v20	Внесены изменения в состав сведений для наполнения региональных витрин данных (приложение 10): внесены изменения в соотв. с реализованными Сервисами на момент декабрь 2022; добавлены сведения из федеральных реестров и справочников, обеспечивающие контроль качества расписания.
14	v21	В п.3.2.2 «Сведения о НСУД и ПОДД СМЭВ 4» определены Поставщики и Потребители – участники информационного взаимодействия. Добавлены указания по применению типового программного обеспечения «Витрина данных Лайт» (п. Ошибка! Источник ссылки не найден.). Актуализированы системные требования для установки витрины в регионе с населением до 1,5 миллионов человек (Таблица 3) и в регионе с населением от 1,5 млн до 3,5 млн человек (Таблица 4). В пункте 5.2 добавлена информация о системных требованиях к тестовому контуру. Уточнены указания по настройке часового пояса в UTC,

№	Версия документа	Изменения
		п.7.3.1.8. Внесены изменения в состав сведений для наполнения региональных витрин данных (приложение 10), таблица «service»: исключены поля «Идентификатор связи Служба+Услуга», «Идентификатор услуги из мастера системы», «Дата удаления записи из витрины»
15	v22	Актуализированы основные определения документа; изменён формат и содержание порядка оказания услуг; внесены уточнения в описания технических и организационных мероприятий по переходу на решение с использованием Витрин данных; обновлено описание модели данных для наполнения Региональной витрины; внесены множественные тех правки по структуре и вёрстке документа.
16	v23	Добавлено описание порядка оказания отдельных услуг; внесены изменения в «Сведения для наполнения региональных витрин данных».
17	v24	Изменён заголовок документа; внесены изменения в используемые по тексту документа термины.
18	v24	Сокращён состав Услуг; внесены изменения в Сведения для наполнения региональных витрин данных (Приложение А); учтены отдельные замечания к МР в ред #23.
19	v25	Внесены изменения в: порядок оказания услуг и сервисов (п.3); рекомендованные системные требования (п. 4.2), порядок перехода ГИСЗ субъекта РФ на решение с использованием технологии витрин данных (п. 5.1); сведения для наполнения региональных витрин данных (приложение А); описание правил проверок, осуществляемых в компоненте «Модуль управления бизнес-правилами» ФЭР ЕГИСЗ (приложение Б); пример решения с использованием OpenAPI (приложение В). Исключено дублирование методических материалов и указаний, в размещенных в ЕСКС: https://info.gosuslugi.ru/materials .
20	v26	Внесены множественные изменения в содержание и структуру документа, документ приведён к форме Методических рекомендаций в составе Единых требования по обеспечению взаимодействия информационных систем в сфере здравоохранения субъектов Российской Федерации с программным обеспечением «Витрина данных».

№	Версия документа	Изменения
21	v27	Добавлено Приложение Ж, актуализирован раздел Термины и сокращения
22	v28	Внесены изменения в порядок оказания отдельных услуг и сервисов в сфере здравоохранения на епгу (п.3)
23	v29	Внесены изменения: в модель данных Витрины, скорректированы значения в колонке «Обязательность» и «Влияние на отображение формы услуги»; в описание программного интерфейса openAPI (В.3), в параметры ответа добавлен необязательный параметр ТМК_url - ссылка на ВКС на онлайн-консультацию.
24	v30	В состав оказываемых услуг (1.1) и в порядок оказания отдельных Услуг (п. 3) добавлены сервисы: «Запись на прием на плановую госпитализацию», «Запись на прием на медицинское освидетельствование».