

BI.Qube  
РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

## ОГЛАВЛЕНИЕ

<b>BI.QUBE .....</b>	<b>4</b>
<b>ОРГАНИЗАЦИЯ РАБОТЫ С BI.QUBE .....</b>	<b>6</b>
<b>METACOMMON .....</b>	<b>8</b>
<b>Введение .....</b>	<b>8</b>
<b>Общее описание концепции построения визуального интерфейса .....</b>	<b>8</b>
<b>Виды endpoints .....</b>	<b>9</b>
<b>Создание подключений .....</b>	<b>10</b>
Создание подключения к Postgresql .....	12
Создание подключения к S3 .....	13
<b>METASTAGING .....</b>	<b>14</b>
<b>Введение .....</b>	<b>14</b>
<b>Создание профилей на странице «Staging» .....</b>	<b>14</b>
<b>Создание и редактирование команд для загрузки данных .....</b>	<b>18</b>
Общие параметры .....	18
Команда «Запрос» .....	21
<i>Запрос извлечения файлов с компьютера пользователя .....</i>	<i>21</i>
<i>Запрос извлечения данных из ИС Предприятие .....</i>	<i>23</i>
<i>Запрос извлечения данных из СУБД .....</i>	<i>27</i>
<i>Запрос извлечения данных из веб-сервисов REST API .....</i>	<i>27</i>
<i>Запрос извлечения данных из файлового хранилища S3 .....</i>	<i>29</i>
<i>Запрос извлечения данных из общих папок Windows .....</i>	<i>29</i>
Типы загрузки .....	29
<i>Полная загрузка .....</i>	<i>29</i>
<i>Полная загрузка с сохранением истории .....</i>	<i>30</i>
<i>Инкрементальная загрузка .....</i>	<i>31</i>
<i>Секции .....</i>	<i>32</i>

<b>Запуск на выполнение.....</b>	<b>35</b>
<b>Сессии .....</b>	<b>36</b>
<b>Данные.....</b>	<b>37</b>
<b>DATA&amp;MODEL .....</b>	<b>39</b>
<b>Введение .....</b>	<b>39</b>
<b>Профиль .....</b>	<b>39</b>
<b>Домен .....</b>	<b>40</b>
<b>Создание модели.....</b>	<b>41</b>
Создание сущности в модели .....	43
Просмотр и редактирование данных.....	48
Создание связей между сущностями .....	51
Сборка сущности.....	52
Работа с моделью в графическом режиме .....	53
<b>METACONTROL.....</b>	<b>59</b>
<b>Введение .....</b>	<b>59</b>
<b>Профиль .....</b>	<b>59</b>
<b>Список рассылки .....</b>	<b>61</b>
<b>Проверка .....</b>	<b>63</b>
<b>Статус рассылки .....</b>	<b>64</b>

## BI.QUBE

BI.Qube – платформа (фреймворк, набор инструментов) для комплексного анализа данных и метаданных начиная от извлечения их на источнике до обогащения в хранилище данных, с организацией эффективного хранения.

Применение BI.Qube позволяет существенно снизить требования к уровню подготовки специалистов по построению корпоративных хранилищ данных (КХД) с использованием методологии DataVault, и в большинстве случаев позволяет отказаться от написания программного кода и вести проектирование КХД в подходе no code/ low code.

BI.Qube включает в себя ряд подсистем, позволяющих полноценно решать определенный круг задач, появляющихся при построении КХД, не зависимо от остальных подсистем, с другой стороны, каждая подсистема предоставляет полноценный интерфейс доступа к данным о своей деятельности, что позволяет организовать бесшовную сквозную связь между задачами разных подсистем. Так, сторонний оркестратор позволяет организовать ETL процесс оптимальным образом с точки зрения временных (ресурсных) затрат. В ряде случаев данные, извлекаемые из источников, в автоматическом режиме укладываются в хранилище в модель DataVault (автоматическое определение бизнес ключей на основании метаданных источника), и пользователю нет необходимости выполнять какие-то дополнительные действия. Концепция построения подсистемы MDM непосредственно в хранилище DataVault существенно снижает трудозатраты связанные с работой с нормативно справочной информацией (НСИ) в том смысле, что интеграция всех справочников НСИ с хранилищем уже реализованы на уровне системы (хранилища) и не требует от пользователей никакого вмешательства в виде программного кода, что существенно удешевляет разработку хранилища данных в целом, сопровождения его в будущем и самое главное позволяет бизнес-пользователям самим создавать и

настраивать работу с НСИ, «золотой» записью, обогащением новыми данными без привлечения программистов. Построение хранилища и подсистемы MDM на основе модели DataVault существенно расширяют возможности по управлению доступа к данным, одновременной работе с данными, сохранения истории появления и изменения данных.

Продукт BI.Qube и его компоненты используют общий подход к организации артефактов разработки. Это позволяет унифицировать процесс разворачивания и тестирования средств разработки и отладки. Это также упрощает перенос и объединение изменений между разными средами разработки.

В состав BI.Qube входят следующие компоненты:

- MetaCommon – компонент имеющий развитый визуальный интерфейс, реализующий работу в режиме no code/ low code и предназначенный для выполнения всех необходимых настроек, которые в последующем использует все остальные компоненты;
- MetaStaging – компонент имеющий развитый визуальный интерфейс, реализующий работу в режиме no code/ low code и предназначенный для извлечения данных из источников и доставки их в точку назначения;
- Data&Model – компонент имеющий развитый визуальный интерфейс, реализующий работу в режиме no code/ low code, предназначенный для создания аналитической модели данных, работой со справочниками, обогащения данных. Компонент работает с данными, доставляемыми с использованием компонента MetaStaging. Включает в себя компонент MetaVault и MetaMasterData;
- MetaVault – компонент имеющий развитый визуальный интерфейс, реализующий работу в режиме no code/ low code и предназначенный для организации хранения данных в модели DataVault. Пользователь может не иметь представления об особенностях модели DataVault система все

необходимые действия выполняет сама и предоставляет доступ к автоматически сгенерированным представлениям;

- **MetaMasterData** – компонент имеющий развитый визуальный интерфейс, реализующий работу в режиме no code/ low code и предназначенный для работы с нормативно-справочной информацией, обогащения данными, вводимыми в ручном режиме через веб интерфейс, создания новых данных. Данный компонент работает только в связке с MetaVault и отдельно работать не может. Компонент реализует возможности MDM систем и создание с его помощью объекты не требуют интеграции с объектами MetaVault;

- **MetaControl** – компонент имеющий развитый визуальный интерфейс, реализующий работу в режиме no code/ low code и предназначенный для создания различных бизнес-правил, например, контроля за ETL-процессами. Компонент выполняет бизнес-правило, которое может быть представлено, например, запросом, выполняет сопоставление полученного результата с эталонным (ожидаемым) и при обнаружении расхождений (с учетом заданной точности) выполняет рассылку по e-mail или по средствам telegram-канала информации о выполненных действиях, всем заинтересованным получателям.

## **ОРГАНИЗАЦИЯ РАБОТЫ С BI.Qube**

Визуальный интерфейс BI.Qube представлен веб-сервисом, организующим диалоговый режим работы с пользователем.

В настоящее время пользователям доступны следующие разделы и страницы:

- **MetaCommon (Общее) – компонент**
  - **Profiles (Профили)** – страница предназначена для просмотра группировки команд в исполняемый объект, а так же запуск на выполнение

- Endpoints (Подключения) – страница для создания подключений к источникам данных
- Data (Данные) – страница для просмотра данных на стороне источника, для которых настроены подключения
- **Data&Model (Данные и модель данных) – компоненты MetaVault и MetaMasterData**
  - Models (Модель) – страница предназначена для создания модели данных и работы с данными
  - Profiles (Профиль) – страница предназначена для просмотра группировки команд в исполняемый объект
  - Domains (Домен) – страница предназначена для создания домена
  - ER-Model (ER-модель) – страница предназначена для работы с моделью данных в графическом режиме
- **Staging (Стейджинг) – компонент**
  - Profiles (Профили) – страница предназначена для просмотра группировки команд в исполняемый объект
  - Commands (Команды) – страница предназначена для создания запросов к источникам для извлечения данных
  - Partitions (Секции) – страница предназначена для настройки различных схем секционирования данных источника для организации инкрементальной загрузки
  - Sessions (Сессии) – страница предназначена для отслеживания статуса о выполнении загрузки данных
  - Data (Данные) – страница предназначена для отображения детальной информации о каждой загрузке

# METACOMMON

## *Введение*

Данный компонент предназначен для концентрации в одном месте всех общих настроек, параметров системы, которые в своей работе используют все остальные компоненты.

В данном компоненте выполняется настройка контейнеров – Profiles (Профилей), в которые помещаются все команды, создаваемые другими компонентами. Использование контейнеров позволяет группировать задачи и запускать их на выполнение в режиме параллельной обработки.

Так же выполняется настройка подключений к различным источникам данных, точкам назначения (Endpoints), система поддерживает работы с широким перечнем источников данных. баз данных, веб-сервисов, файловых хранилищ, сетевых дисков...

А также компонент позволяет визуально просмотреть данные, включая метаданные доступные в любом endpoint, доступ к которому настроен.

## *Общее описание концепции построения визуального интерфейса*

Все страницы системы VI.Qube имеют похожую структуру и представлены в виде трёхколоночного макета. Левая колонка (1) содержит пункты главного меню, позволяющие осуществить переход на интересующую страницу программы. В средней части (2) размещается основной набор визуальных элементов, позволяющих увидеть все необходимые настройки, в большинстве случаев эта часть представлена в табличном виде. Редактирование осуществляется с использованием правой колонки (3), в которой размещается «скрываемое» окно свойств каждой строки таблицы.



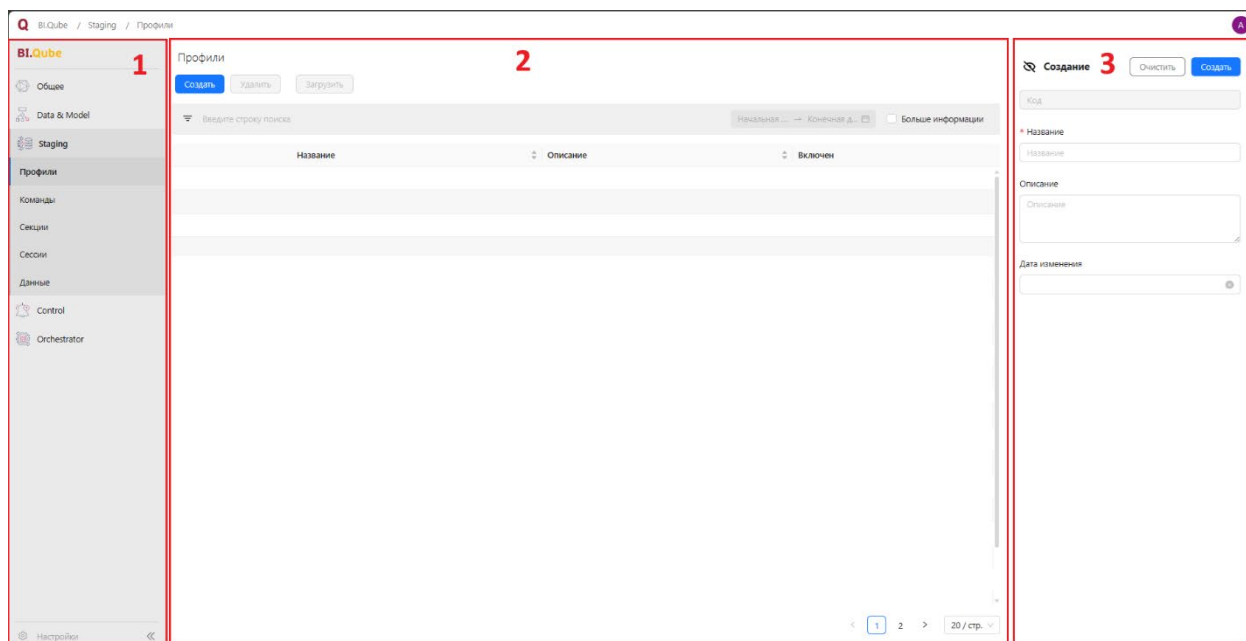


Рисунок 1. Макет типовой страницы

Переход по страницам программы осуществляется с использованием бокового меню, наименования страниц имеют логичные названия и позволяют понять, какие настройки могут быть размещены на странице.

### Виды endpoints

В BI.Qube в настоящее время поддерживаются следующие типы endpoints, которые могут быть настроены как источник данных:

- СУБД – 1C, PG, GP, MS SQL, Oracle, MySQL;
- Web-сервисы REST API – JSON;
- Файловые сервисы S3 – parquet, xls, xlsx;
- Локальные файлы – xls, xlsx.

Для организации точки назначения – платформы, в которую доставляются данные, могут быть созданы следующие endpoints:

- СУБД – PG, GP (Table, External table);
- S3 - parquet, xls, xlsx.

Перечень типов источников данных и видов данных постоянно обновляется и расширяется.

## Создание подключений

Для создания нового endpoints – подключения к источнику или создания точки назначения необходимо выбрать ссылку в боковом меню endpoints (подключения).

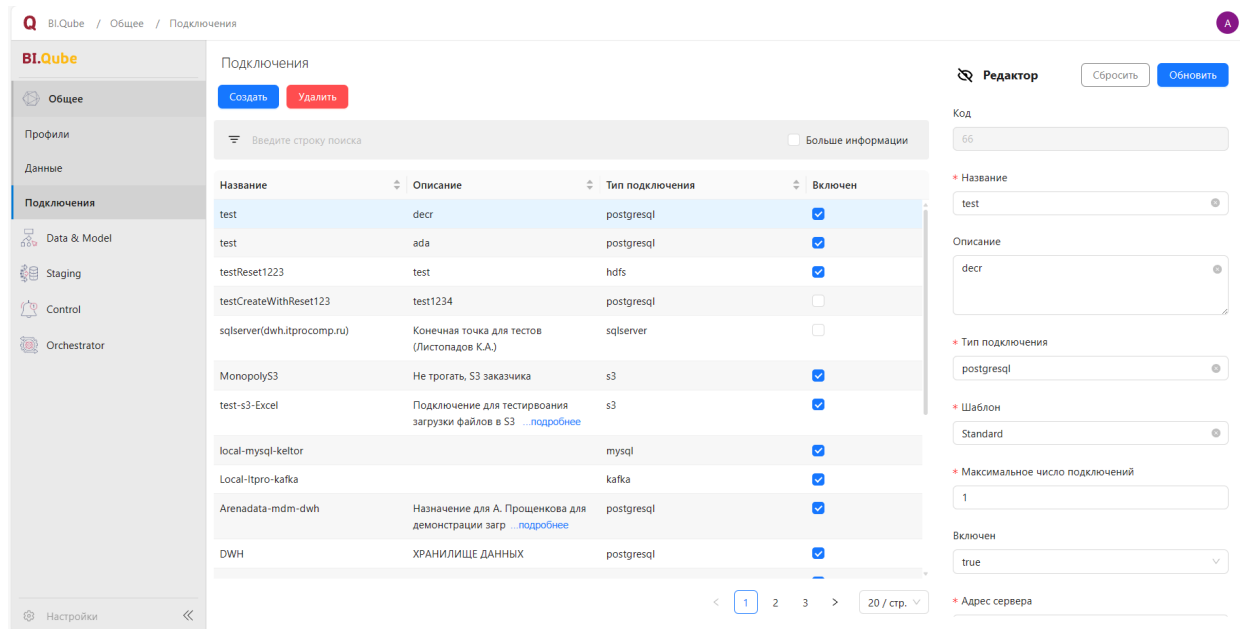


Рисунок 2. Страница создания/редактирования подключений

В основной части страницы появится список созданных ранее подключений, если ранее ничего не было создано, то таблица будет пустой. Создание нового подключения осуществляется нажатием кнопки «Создать». Справой стороны экрана в окне свойств необходимо заполнить следующие поля (поля, указанные со звёздочкой, обязательны для заполнения.):

- Code (Код) – уникальный идентификатор записи в базе данных;
- Name (Название) – имя подключения, вводится без пробелов;
- Description (Описание) – бизнес описание подключения;
- Endpoint Type (Тип подключения) – тип подключения, система содержит большой перечень подключений к различным источникам и типов источников;
- Template (Шаблон) – шаблон строки подключения;
- Max connections (Максимальное число подключений) – максимальное количество одновременных подключений к источнику;

- Enabled (Состояние) – опция указывает на доступность подключения в системе;
- Connection string (Строка подключения) – поле из которого можно скопировать автоматически сформированную строку подключения;
- TestConnection – процедура проверки доступности подключения.

В зависимости от выбранного типа подключения автоматически в интерфейсе появляются дополнительные поля требующие заполнения.

Редактирования уже созданного подключения также производится в окне свойств. После внесения изменений в строки необходимо нажать на кнопку Update (Обновить) в верхней части окна свойств.

Код

Код

\* Название

Название

Описание

description

\* Тип подключения

Тип подключения

\* Шаблон

Шаблон

\* Максимальное число подключений

Максимальное число подключений

Включен

true

Строка подключения

Проверить подключение

Скопировать

Строка подключения

Рисунок 3. Поля для настройки подключений

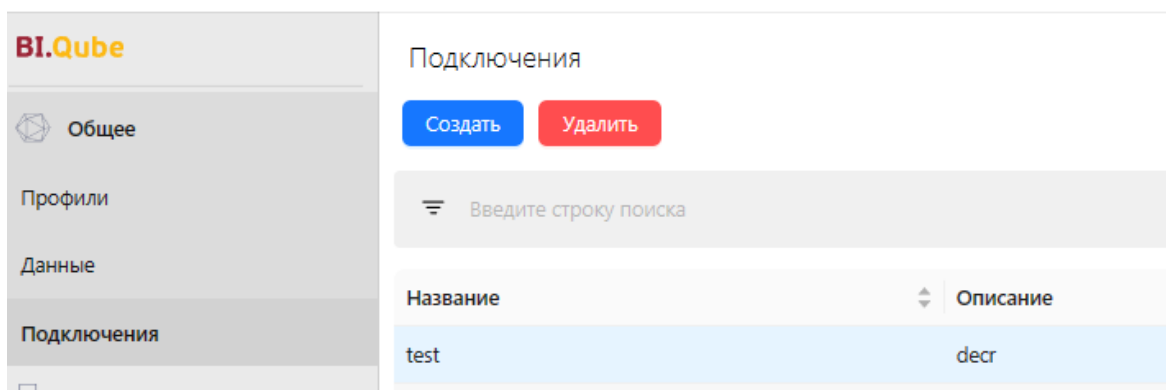


Рисунок 4. Удаление подключения

Для удаления ранее созданного подключения необходимо выделить нужную строку одним щелчком левой кнопки мыши и нажать кнопку «Удалить».

## Создание подключения к Postgresql

Для данного типа источника доступно несколько вариантов шаблонов подключения.

В зависимости от выбранного шаблона создается перечень полей для заполнения.

- Endpoint Type (Тип подключения) - Postgresql;
- Template (Шаблон) – Standart (выбирается в зависимости от настроек endpoint);
- Host (Адрес сервера) – указывается сетевой адрес, где размещен endpoint;
- Port (Порт) – указывается порт, по которому доступен endpoint;
- Database (Имя базы данных) – указывается имя базы данных;
- User (Пользователь) – указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) – пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- Include Error Detail (Выводить детальный лог) – сохраняет детальный лог в процессе подключения к endpoint.

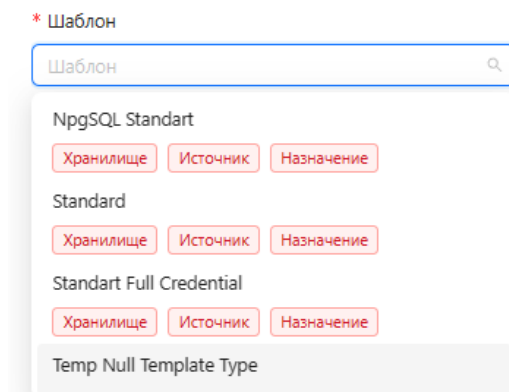


Рисунок 5. Шаблоны строки подключения

Рисунок 6. Пример настройки

## Создание подключения к S3

Для данного типа источника доступные шаблоны подключения приведены в одноименном поле. После выбора интересующего шаблона необходимо заполнить появившиеся поля:

- Endpoint Type (Тип подключения) – файловое хранилище S3;
- Template (Шаблон) – выбрать шаблон строки подключения;
- Access Key (Ключ доступа) ввести ключ доступа к бакету файлового хранилища;
- Secret Key (Секретный ключ) – секретный ключ;
- Bucket Name (Имя бакета данных) – имя бакета, к которому настраивается доступ;
- IP Endpoint (Адрес сервера) адрес сервера, где размещено файловое хранилище;
- Region (Регион) указать для какого региона выполнены настройки в файловом хранилище.

\* Тип подключения  
s3

\* Шаблон  
S3 Compatible Storage

\* Максимальное число подключений  
10

Включен  
true

\* Ключ доступа  
4WWmbPVBRWdSZfXq

\* Секретный ключ  
\*\*\*\*\*

\* Имя бакета данных  
metacomponents-dev

\* Адрес сервера  
http://192.168.72.47:9000

Регион  
Регион

Проверить подключение  
Скопировать

Строка подключения  
Key=4WWmbPVBRWdSZfXq;  
Secret=\*\*\*\*\*;  
BucketName=metacomponents-dev;

Рисунок 7. Пример настройки

# METASTAGING

## *Введение*

При построении хранилищ данных наиболее частой задачей является извлечение данных из источника и их копирование в слой, предназначенный для хранения. Под таким слоем в зависимости от целевой архитектуры понимают DataLake, детальный слой данных (DDS), стейджинговый слой – далее обобщенно этот слой называется стейджингом. Более простыми словами можно сказать, что это может быть либо файловое хранилище данных, либо реляционное хранилище данных.

Другими словами можно сказать, что MetaStaging предназначен для консолидации данных в стейджинговом слое хранилища данных из гетерогенных источников с поддержанием целостности и унифицированности метаданных, также уменьшает нагрузку на операционные базы данных при выполнении запросов, и кроме того, обеспечивает надежное подключение различных БД из разнородных источников для помещения данных в единый слой стейджинга (staging area) с поддержанием целостности метаданных в системе-назначения.

## *Создание профилей на странице «Staging»*

Создание профилей выполняется на одноименной странице Profiles (Профили), данная страница предназначена для создания и редактирования профилей. Следует иметь в виду, что профили созданные здесь доступны во всех компонентах и также отображаются в общих настройках.

По умолчанию в только что развернутой системе не создано ни одного профиля, система должна иметь хотя бы один профиль, в который будут сгруппированы команды, без профиля нет возможности запустить выполнение команд.

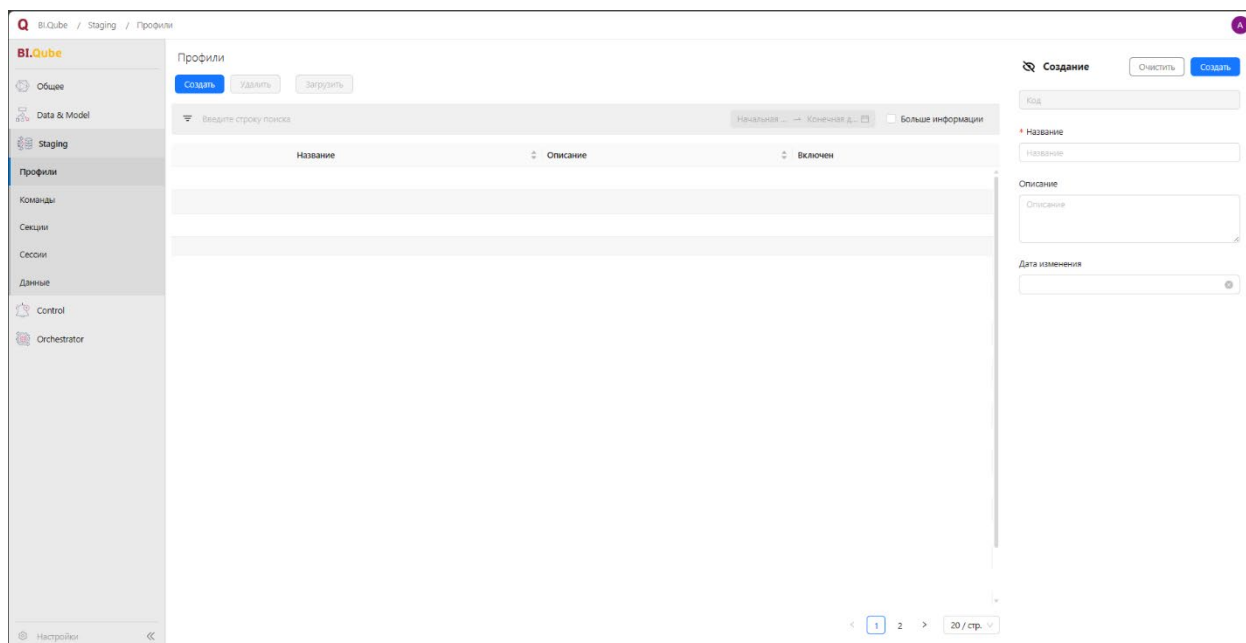


Рисунок 8. Пустая страница Profiles (Профили)

Для создания нового профиля необходимо нажать на кнопку Create (создать). Справа появится (если оно ранее было скрыто) окно свойств, в котором необходимо заполнить следующие поля:

- Name (Название) – уникальное имя профиля, позволяющее отделять один профиль от другого, как правило, даётся осмысленное имя, поясняющее назначение профиля, не должно содержать пробелов;
- Description (Описание) – расширенное описание назначения профиля (необязательное поле, введено для удобства пользователей);
- Start date (Дата изменения) – автоматически создаваемое поле, содержит дату создания поля, при необходимости дата может быть отредактирована.

Создание

Очистить Создать

Код

\* Название

Описание

Дата изменения

04.12.2023

*Рисунок 9. Заполнение свойств профиля*

Кроме вышеперечисленных свойств, в базу данных автоматически попадают учётные данные о текущем авторизованном пользователе. В базе данных программы хранятся только сведения о последнем внесённом изменении. При редактировании профиля, история внесённых изменений не сохраняется. Данную информацию можно увидеть, нажав на кнопку More info (Больше информации), в строке фильтров над таблицей, в основной части экрана.

После заполнения всех свойств необходимо нажать кнопку Create (Создать), и только что созданный профиль отобразится в таблице.



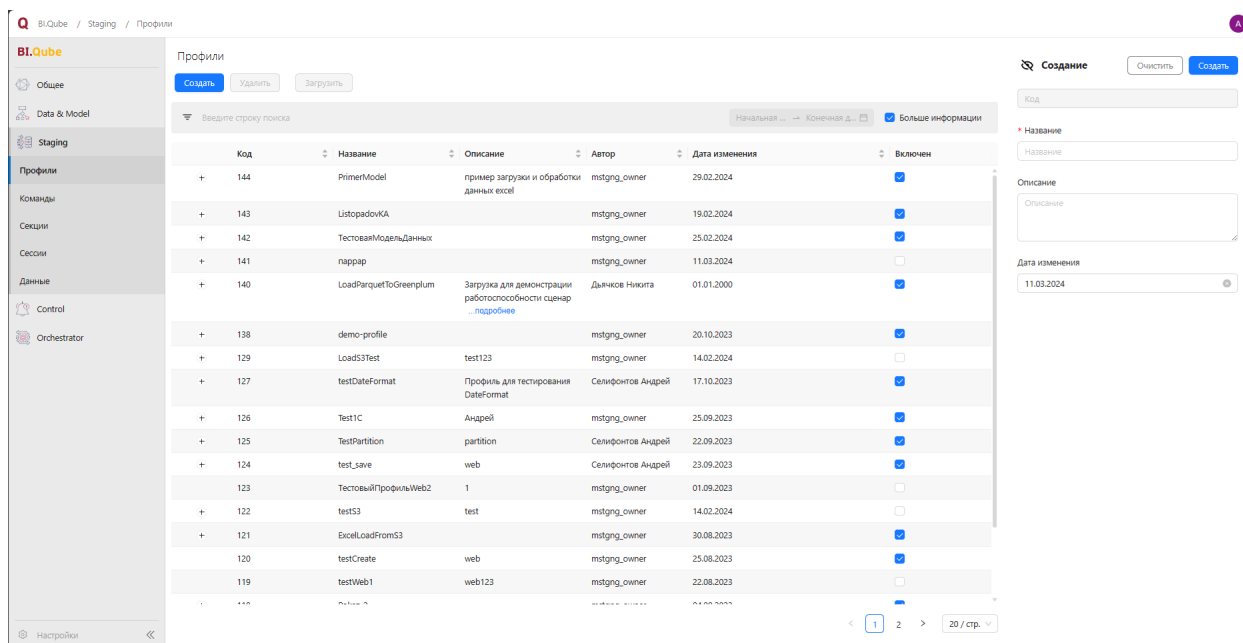


Рисунок 10. Пример заполненной страницы Profiles (Профили) с включённым фильтром More info (больше информации)

Для редактирования свойств профиля необходимо щёлкнуть левой кнопкой мыши по интересующей строке в таблице, внести необходимые изменения в поля свойств в правой части экрана и нажать кнопку Обновить (Update).



Рисунок 11. Команда обновления данных о профиле

Для удаления профиля также необходимо щёлкнуть левой кнопкой мыши по строке в таблице, появится кнопка Delete (Удалить), нажатие на которую приведет к безвозвратному удалению выбранного профиля.

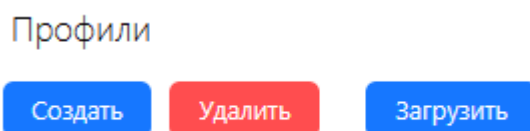


Рисунок 12. Команда удаления профиля

## Создание и редактирование команд для загрузки данных

### Общие параметры

При создании команды для загрузки данных из источника в точку назначения потребуется некоторый опыт работы с СУБД и начальное понимание SQL-запросов.

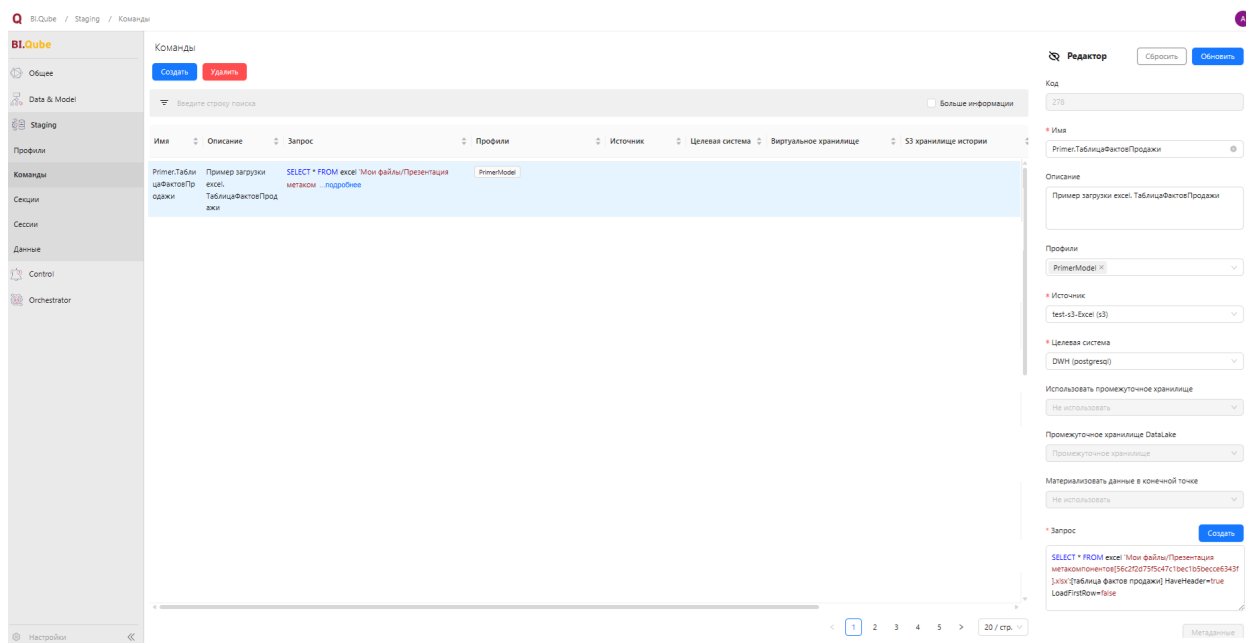


Рисунок 13. Пример настроенной команды

Для создания настроек параметров команды необходимо заполнить следующие поля:

1. Name (Имя) – уникальное наименование команды без пробелов;
2. Description (Описание) – бизнес-описание команды;
3. Profiles (Профили) – команда помещается в один или более профилей (контейнеров);
4. Source (Источники) – система – источник данных для загрузки;
5. Destination (Целевая система) – система, в которую планируется загрузить данные из источников;
6. Use intermediate storage (Использовать промежуточное хранилище) – иногда при построении хранилищ требуется использовать дополнительное файловое хранилище, например S3, используя данную

- опцию можно организовать доставку данных сначала в одно хранилище, а затем в следующее с использованием одной команды;
7. Intermediate storage DataLake (Промежуточное хранилище DataLake) – выбирается endpoint для промежуточного хранилища;
  8. Materialize data at endpoint (Материализовать данные в конечной точке) – в случае использования опции «Использовать промежуточное хранилище» есть возможность сгенерировать External Table в конечной точке или генерировать таблицы с данными, которые продублированы в промежуточном хранилище;
  9. Query (Запрос) – запрос к источнику данных (ниже приведено детальное описание возможных вариантов);
  10. Destination object (Имя объекта) – наименование объекта в точке назначения, для реляционного слоя задается в формате ИмяСхемы.ИмяТаблицы;
  11. Save history (Сохранять историю) – использовать/не использовать;
  12. Specify intermediate storage (Укажите промежуточное хранилище) – выбрать из выпадающего списка нужное хранилище, в которое будут сохраняться результаты всех выполнений команд;
  13. Load type (Тип загрузки) – тип загрузки (инкрементальная, инкрементальная по дате, инкрементальная по идентификатору, перезагрузка таблицы, полная загрузка, полная с сохранением истории);
  14. Batch size (Размер пакета данных) – размер пакета данных;
  15. Partition schema (Схема секционирования) – задается исходя из назначения секции и зависит от типа секционирования;
  16. Partition column (Поле секционирования) – поле, по которому осуществляется секционирование;
  17. Partition column convert (Условия для секции) – условия, характерные для выбранной секции.



## Команда «Запрос»

Команда Query (Запрос) открывает диалоговое окно для пользователя, в котором создается запрос (команда), которая будет выполнена на стороне endpoint для извлечения данных. Окно создания запроса зависит от типа endpoint:

- запрос извлечения файлов с компьютера пользователя (xls,xlsx, csv, xml);
- запрос извлечения данных из 1С Предприятие (на основе MS SQL Server, PostgreSQL);
- запрос извлечения данных из СУБД (MS SQL Server, Oracle, MySQL, PG, GP);
- запрос извлечения данных из веб-сервисов REST API;
- запрос извлечения данных из общих каталогов windows.

Тип нужного диалогового окна определяется автоматически, на основе выбранного endpoint, используемого в качестве источника данных.

### Запрос извлечения файлов с компьютера пользователя

После нажатия кнопки Create (Создать) появится диалоговое окно, в котором можно оформить запрос на извлечение данных из источника в диалоговом режиме или ввести запрос с клавиатуры на языке источника данных или, в отдельных случаях, на внутреннем языке системы.

Окно создания запроса разделено на две зоны, слева зона отображения кода запроса к источнику, и под ним зона предварительного просмотра результата запроса и зона создания кода запроса.

Для загрузки файла с локального компьютера пользователя, файл необходимо обязательно поместить в промежуточное хранилище, чаще всего это хранилище типа S3, endpoint для которого должен быть создан заранее. В правой зоне окна создания запроса отображается файловая структура выбранного хранилища и файлы доступные в хранилище. Для загрузки файла

в хранилище, если нужного файла еще нет, нужно нажать кнопку Upload file (Загрузить файл), в результате откроется стандартное диалоговое окно Windows выбора файла. Далее необходимо выбрать интересующий файл на компьютере пользователя с использованием открытого диалогового окна и щёлкнуть по кнопке «ОК». Выбранный файл автоматически загрузится в хранилище и подсветится в структуре каталогов. С этого момента файл доступен для анализа.

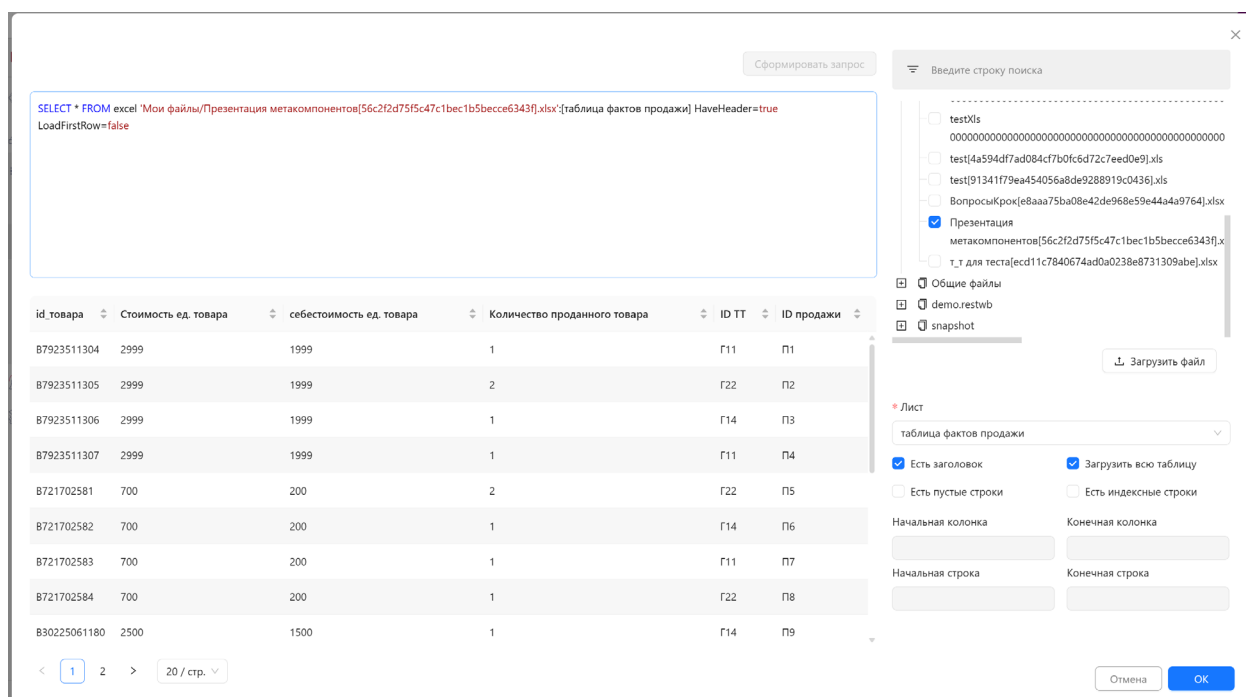


Рисунок 15. Загрузка файла с компьютера пользователя

Для настройки команды загрузки выбранного файла необходимо в выпадающем списке Sheet (Лист) выбрать лист, данные из которого необходимо будет загрузить, после чего задать нужные опции:

- Have header (Есть заголовок) – опция, позволяющая использовать первую строку диапазона данных использовать как строку заголовков таблицы;
- Load full table (Загрузить всю таблицу) – автоматическое определение диапазона данных;

- Have empty rows (Есть пустые строки) – позволяет из диапазона данных удалять пустые строки;
- Have index rows (Есть индексные строки) – добавляется колонка с номерами строк.

Если опция Load full table (Загрузить всю таблицу) не выбрана, то пользователь может ввести нужный диапазон данных вручную.

Для просмотра результирующего запроса и результатов его работы необходимо нажать на кнопку Form a query (Сформировать запрос), автоматически сформируется текст запроса и в зоне предварительного просмотра появятся результаты выполнения этого запроса.

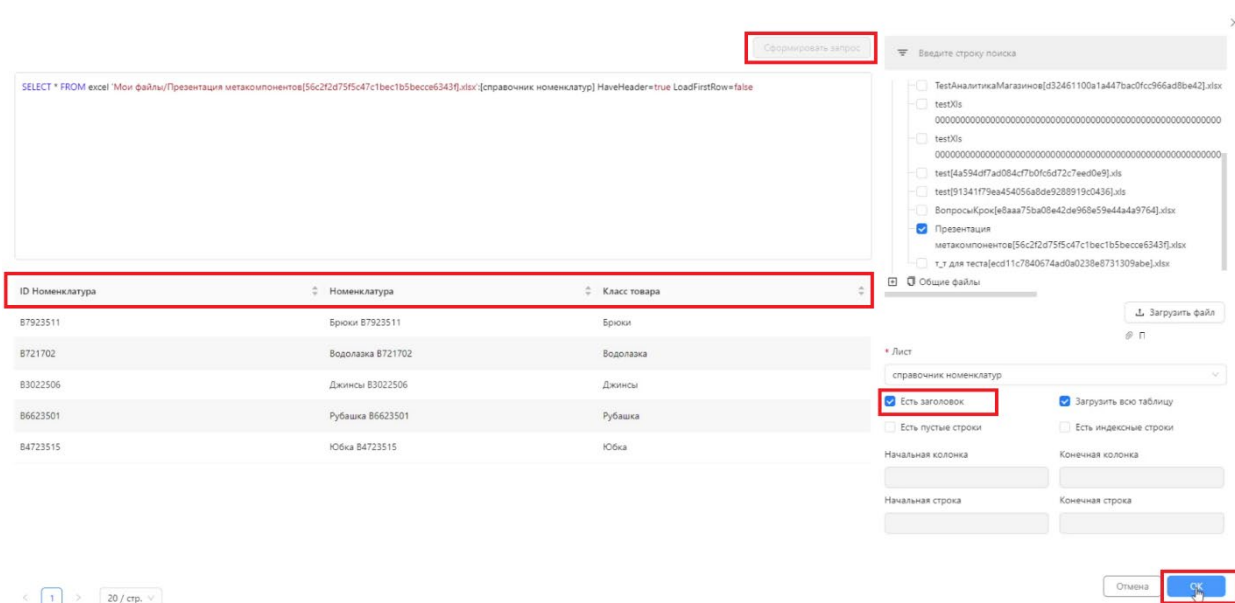


Рисунок 16. Сверка загруженного файла и добавление заголовков в таблицу

После окончания настройки запроса следует нажать кнопку «ОК», данный запрос загрузит данные из файла в хранилище при запуске команды на выполнение.

## Запрос извлечения данных из 1С Предприятие

Для создания команды загрузки данных из 1С Предприятие, должен быть выбран соответствующий endpoint в выпадающем списке Source

(Источник). После нажатия на кнопку Create (Создать), автоматически сформируется окно для настройки команды. В этом окне справа расположено дерево объектов 1С той конфигурации данных, к которой настроен endpoint.

Для автоматического формирования текста запроса необходимо выбрать интересующий объект 1С. При этом следует помнить, что некоторые объекты 1С представлены одной таблицей, например справочники, некоторые представлены набором таблиц, например документы. В связи с этим необходимо понимание у пользователя данные из какого объекта и связанные с этим объектом нужны пользователю. В примере выбран справочник «Физические Лица» и все атрибуты этого справочника. После выбора нужного объекта необходимо нажать кнопку Form a query (Сформировать запрос) в результате чего будет сформирован запрос и данные из выбранного объекта появятся в зоне предварительного просмотра.

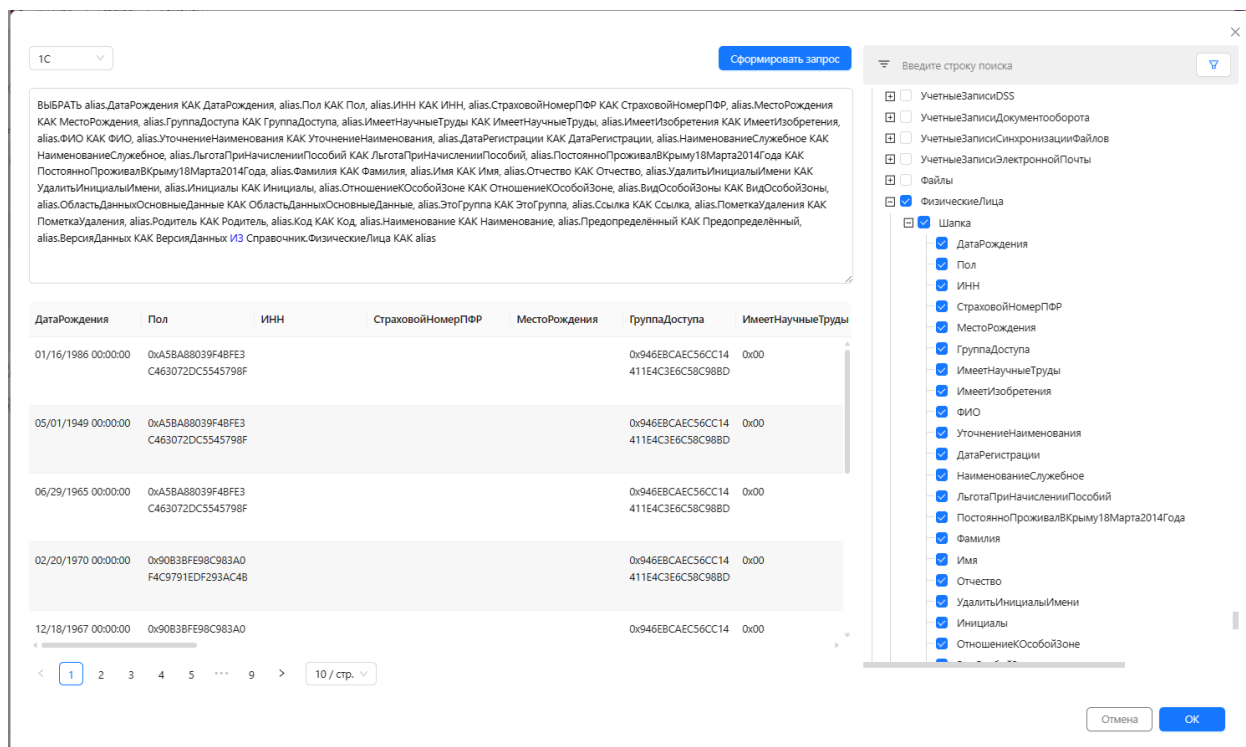
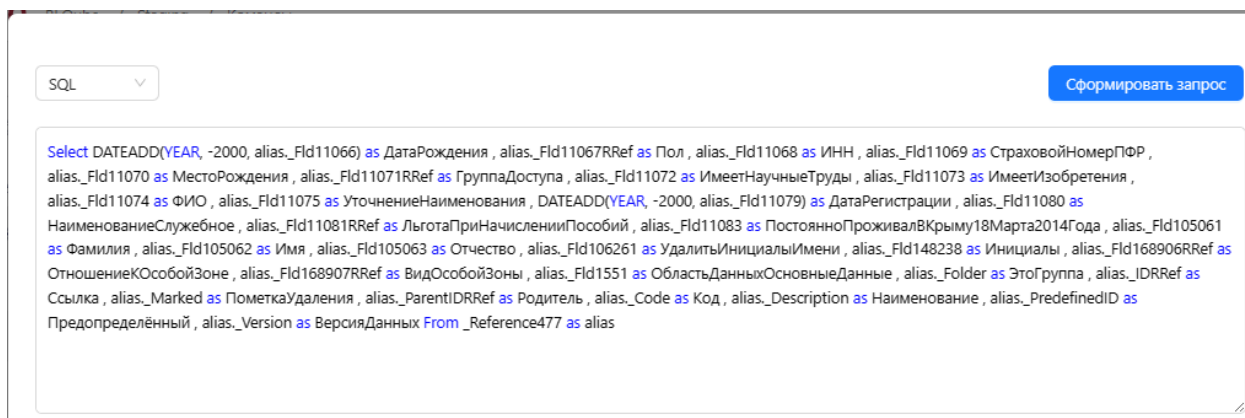


Рисунок 17. Пример отображения данных в 1С запросе

Сформированный запрос также может быть отображен не только в нотации 1С, но и в нотации SQL.





*Рисунок 18. Пример отображения данных в SQL запросе*

Кроме этого, для анализа связей между объектами 1С предусмотрен графический режим работы. В этом режиме пользователь может увидеть с какими объектами связан выбранный объект (простыми словами можно сказать так: из каких связанных таблиц подтягиваются данные в выбранную таблицу).

В графическом режиме доступно два вида отображения:

- концептуальный – в этом режиме все объекты, в том числе и сложные (составные) объекты представляются одним графически элементом и отображаются связи между ними, таким образом можно понять, например, на какие справочники ссылается выбранный документ;
- детальный – в этом режиме все объекты отображаются в отдельном графическом объекте, с перечислением атрибутов и указанием по каким атрибутам установлены связи.

*Следует помнить, что отображаются все связи выбранного объекта, связи между зависимыми объектами не отображаются. Таким образом для детального анализа связей необходимо исследовать каждый объект по отдельности.*

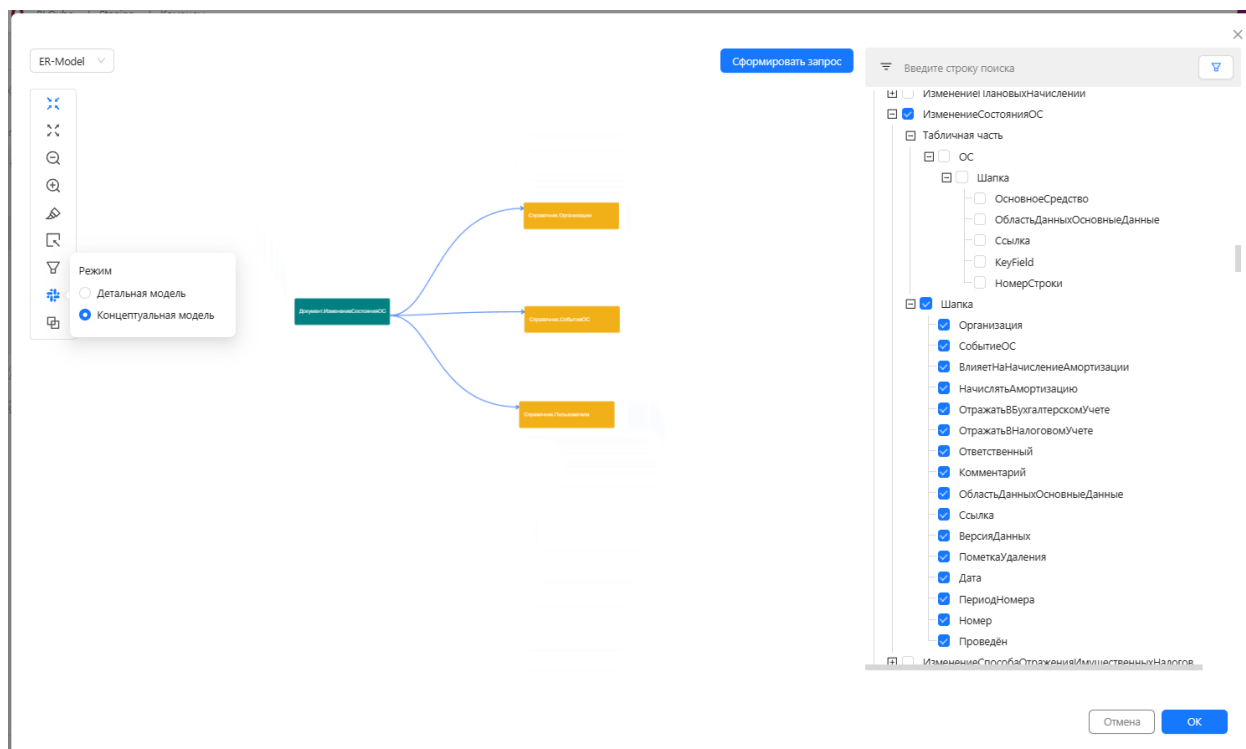


Рисунок 19. Концептуальная ER - модель отображения данных

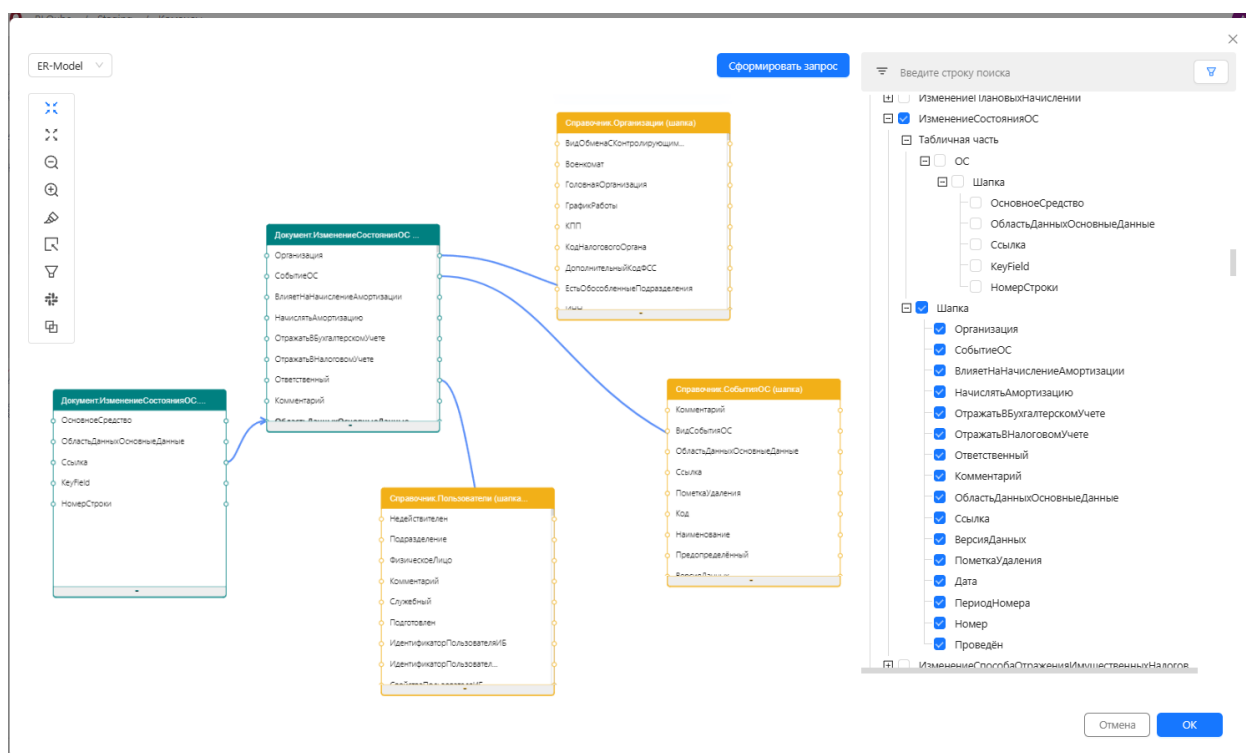


Рисунок 20. Детальная ER – модель отображения данных

Работа с окном заканчивается после того, как настроен запрос и нажата кнопка «ОК» в правом нижнем углу модального окна.

## Запрос извлечения данных из СУБД

Для создания команды загрузки данных из СУБД, должен быть выбран соответствующий endpoint в выпадающем списке Source (Источник). После нажатия на кнопку Create (Создать), автоматически сформируется окно для настройки команды. В этом окне справа расположено дерево объектов СУБД той конфигурации данных, к которой настроен endpoint.

Далее нужно выбрать объект СУБД и нажать кнопку Form a query (Сформировать запрос). В результате чего автоматически формируется простой запрос на выборку всех данных, данный запрос можно редактировать, при этом, следует помнить, что нотация SQL запроса зависит от выбранного endpoint.

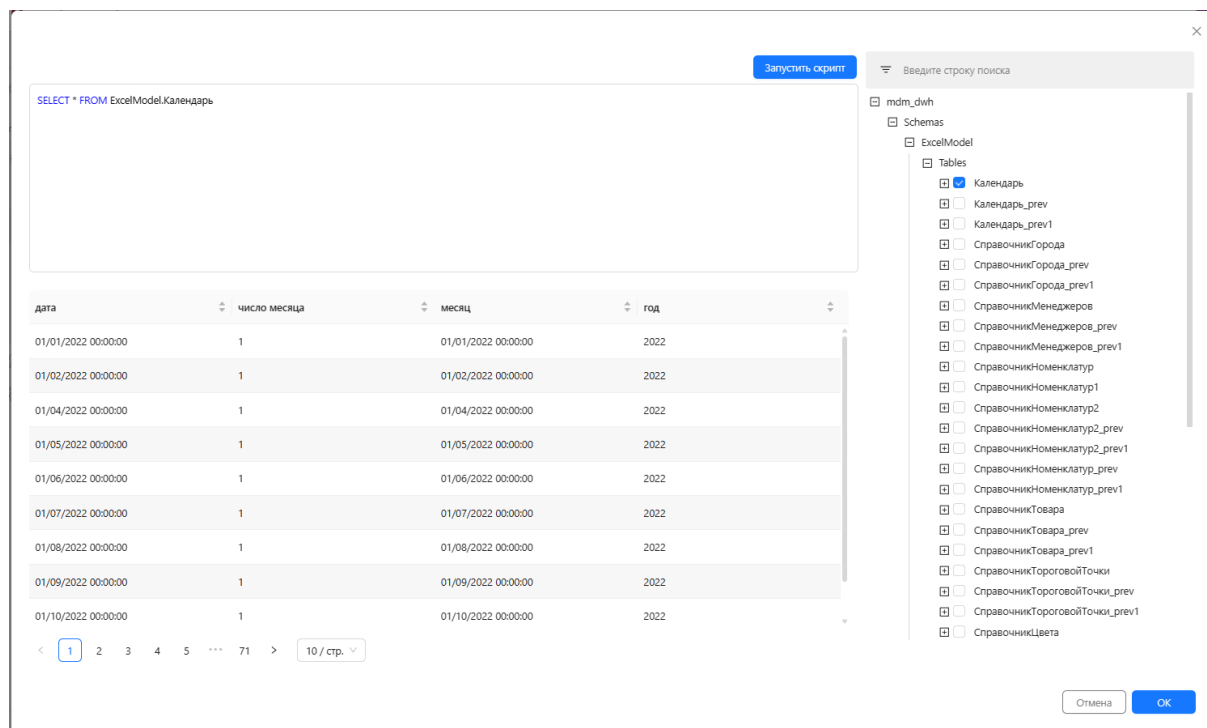


Рисунок 21. Пример запуска скрипта извлечённых данных из СУБД

После окончания настройки запроса необходимо нажать кнопку «ОК» в правом нижнем углу модального окна.

## Запрос извлечения данных из веб-сервисов REST API

Для создания запроса для извлечения данных из веб-сервисов типа REST API (должен быть выбран соответствующий endpoint). Для этого необходимо

открыть окно создания команд. В строку (Введите адрес файла) необходимо ввести адрес файла на веб-сервере, данные из которого необходимо загружать. После чего нажать кнопку «Загрузить файл». Данные из файла отобразятся в поле просмотра ниже. Для автоматического создания запроса загрузки данных из файла необходимо нажать кнопку Form a query (сформировать запрос), в результате сформируется запрос, который позволит получить данные из файла, для просмотра результата нужно нажать кнопку «Выполнить запрос». Предварительный просмотр доступен в двух режимах:

- табличный – данные будут разложены в нормализованные табличные структуры. Не всегда простой запрос может в полностью автоматическом режиме выполнить эту операции.
- Json – данные преобразуются в более простую структуру.

Если в результате предварительного просмотра появился неожиданный результат, это означает, что используемые алгоритмы не могут в автоматическом режиме обработать исходный json-файл. Чтобы добиться ожидаемого результата следует подготовить JOLT-инструкции для парсинга файла и проверить полученный результат.

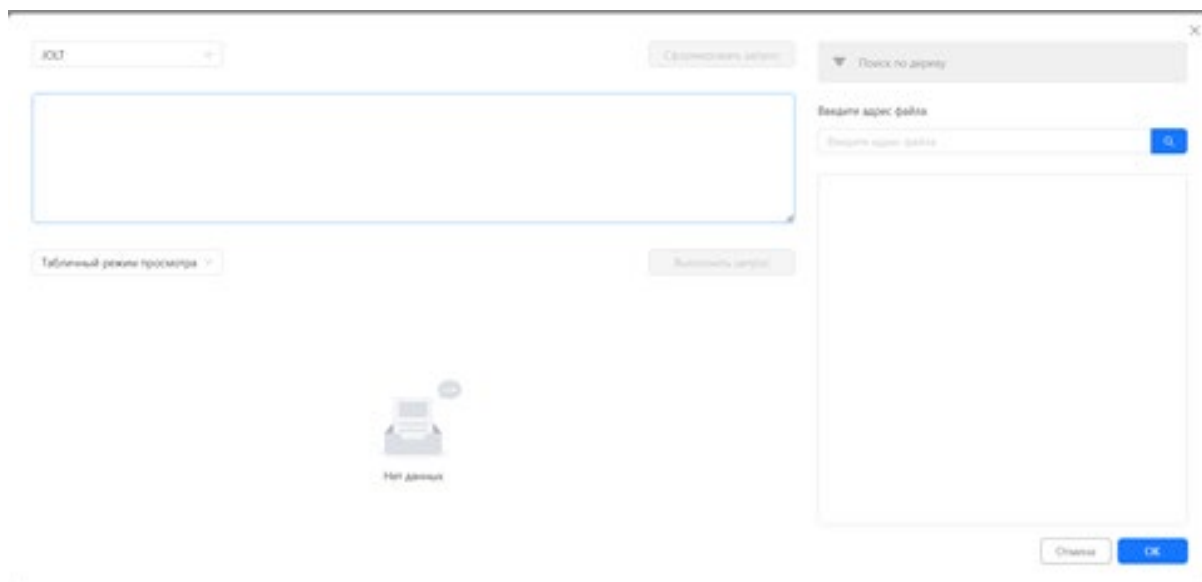


Рисунок 22. Подготовка запроса для извлечения данных из файла JSON

При нажатии на кнопку «Ок», всё сохраняется в настроечные таблицы, откуда и читается при открытии окна.

Запрос извлечения данных из файлового хранилища S3

В разработке аналогично Excel

Запрос извлечения данных из общих папок Windows

В разработке

### Типы загрузки

В поле Load type (Тип загрузки) можно выбрать наиболее подходящий тип загрузки данных для определенной команды.

#### Полная загрузка

Полная загрузка – это загрузка данных без параметризации. Применяется, когда необходима полная перезагрузка всех данных в таблице на источнике (например, при отсутствии столбца, подходящего для секционирования).

\* Тип загрузки

Полная загрузка



Размер пакета данных

1000

Схема секционирования

Схема секционирования



Поле секционирования

Поле секционирования

Условия для секции

Условия для секции

*Рисунок 23. Выбор полной загрузки*

### Полная загрузка с сохранением истории

Полная загрузка с сохранением истории – это загрузка данных без параметризации. Но представления перенацеливаются на новые Parquet-файлы, а старые не удаляются из хранилища.

\* Тип загрузки

Полная загрузка с сохранением истории



Размер пакета данных

1000

Схема секционирования

Схема секционирования



Поле секционирования

Поле секционирования

Условия для секции

Условия для секции

*Рисунок 24. Выбор полной загрузки с сохранение истории*

## Инкрементальная загрузка

Инкрементальная загрузка (загрузка с параметрами) – это регулярная загрузка данных. При этом, извлекаются актуальные данные с даты последней загрузки.

The image shows a web form for configuring an incremental load. It consists of several labeled input fields:

- \* Тип загрузки**: A dropdown menu with the selected option "Инкрементальная загрузка".
- Размер пакета данных**: A text input field containing the value "1000".
- Схема секционирования**: A dropdown menu with the selected option "Схема секционирования".
- Поле секционирования**: A text input field containing the value "Поле секционирования".
- Условия для секции**: A text input field containing the value "Условия для секции".

Рисунок 25. Выбор инкрементальной загрузки

Запрос для инкрементальной загрузки (или загрузки с параметрами) данных отличается от полной и требует настройки дополнительных параметров:

- Partition schema (Схема секционирования) задается исходя из назначения секции и зависит от типа секционирования. Выбирается из выпадающего списка. Создание секции производится на странице Partition (Создание секций);
- Partition column (Поле секционирования) - поле, по которому осуществляется секционирование;
- Partition column convert (Условия секционирования) - условия, характерные для выбранной секции.

## Секции

Страница Partition (Секции) предназначена для создания схем секционирования, необходимых для работы инкрементальной загрузки данных.

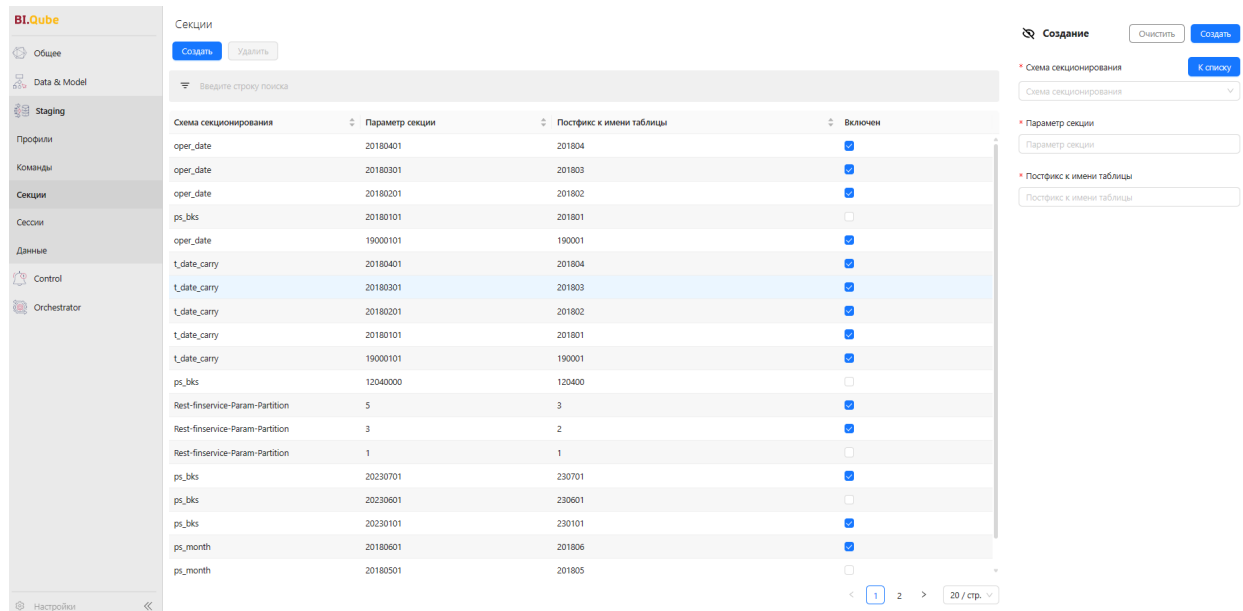


Рисунок 26. Страница Partition (Секции)

Данные новой схемы заполняются нажатием на кнопку Create (Создать).  
Далее следует заполнить поля в правой части экрана:

- Partition schema (Схема секционирования) – заполняется выбором из выпадающего списка;
- Partition value (Параметр секции);
- Partition postfix (Постфикс к имени таблицы).



Рисунок 27. Параметры секционирования

Секция может редактироваться и настраиваться под потребности пользователя, для этого необходимо нажать на кнопку To the list (К списку).

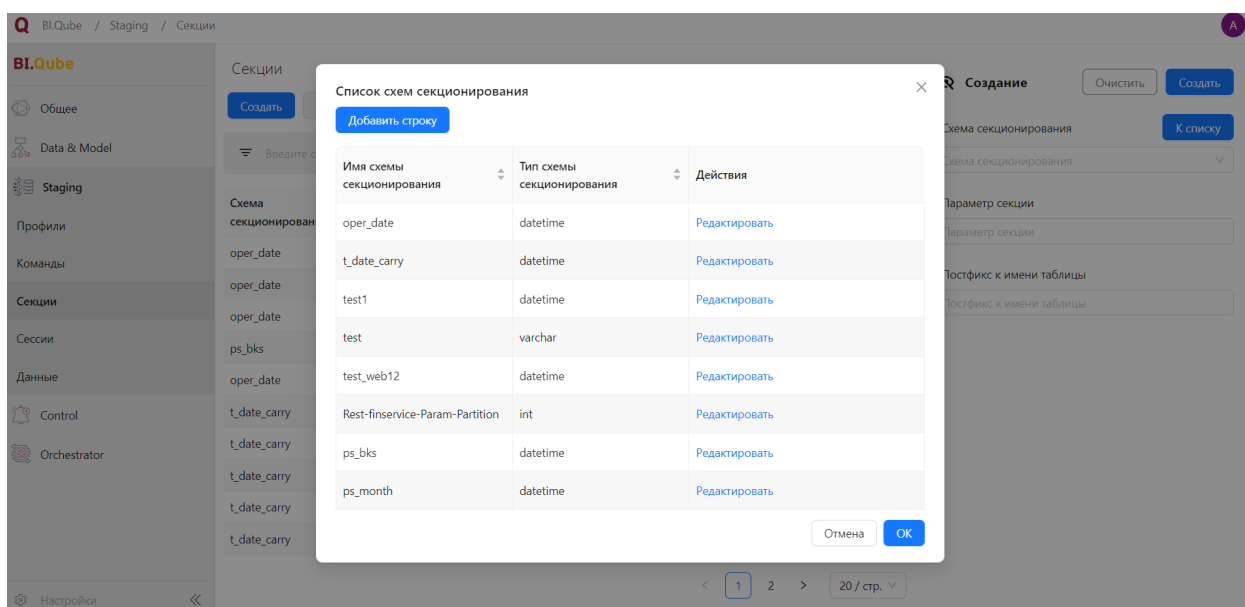


Рисунок 28. Модальное окно «Список схем секционирования»

Появится модальное окно, в нем три колонки:

- Partition schema name (Имя схемы секционирования);
- Partition schema column type (Тип схемы секционирования);
- Operations (Действия). При нажатии в данной колонке (Редактировать). Появляется возможность изменить название схемы секционирования, а также задать тип схемы секционирования из выпадающего

списка. По окончании редактирования следует нажать «Ок», чтобы сохранить внесенные изменения и Create (Создать).

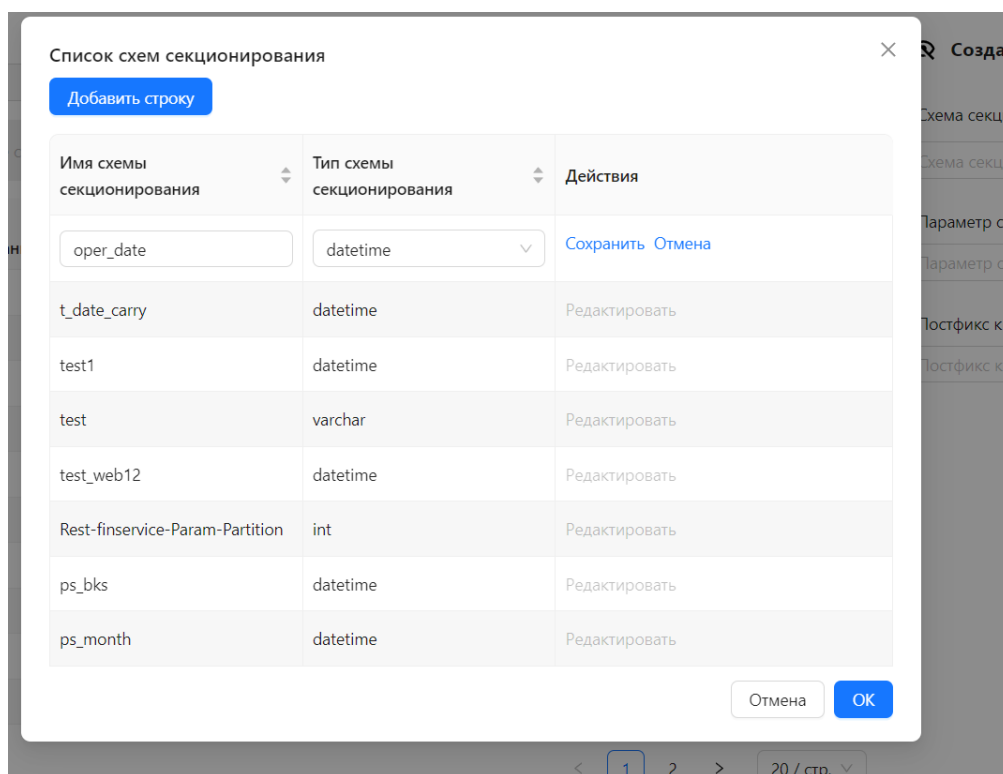


Рисунок 29. Список схем секционирования. Редактирование

Помимо редактирования можно добавить схему, для этого следует нажать кнопку Add a row (Добавить строку). Далее действия аналогичны процессу редактирования.

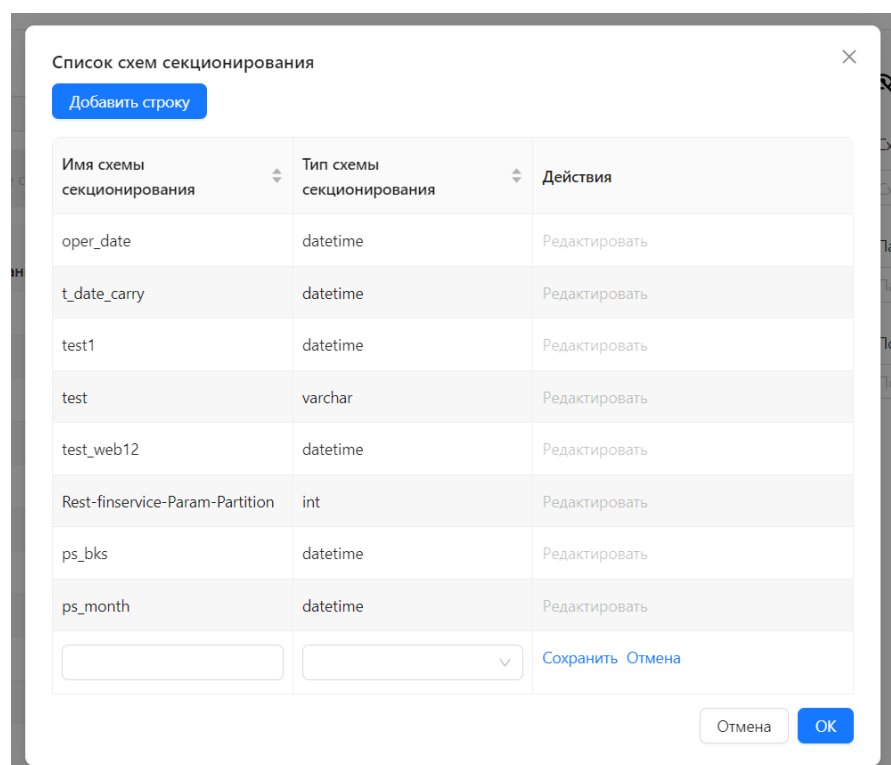


Рисунок 30. Список схем секционирования. Добавление строки

### Запуск на выполнение

Для того, чтобы загрузить данные в хранилище с использованием созданных команд необходимо на странице Profiles (Профили) выбрать интересующий профиль и нажать кнопку Load (Загрузить), затем в появившемся диалоговом окне подтвердить действия нажатием кнопки Yes (Да).

Если необходимо в данный момент времени выполнить не все команды профиля, то раскрыв содержимое профиля можно отключить команды, данные из которых не нужны в текущей загрузке.

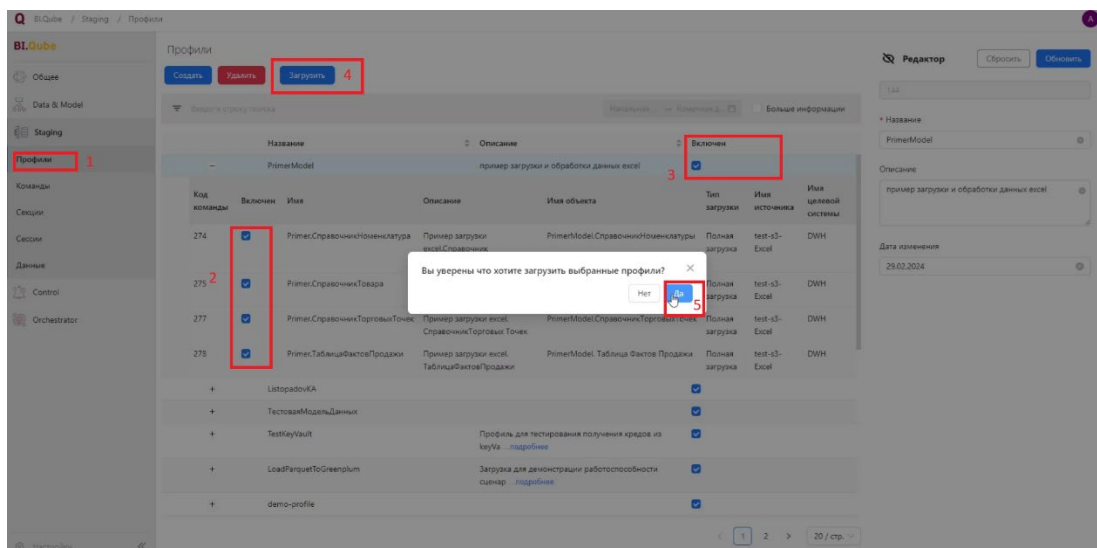


Рисунок 31. Выбор загруженных данных

Нумерацией на рисунке показан алгоритм выполняемых действий на странице.

## Сессии

На странице «Сессии» отображаются все сессии загрузки данных (важно не путать с предварительным просмотром при создании команды загрузки). Каждая загрузка подробно логируется и для каждой команды доступна вся история загрузок.

Для просмотра детализации сессии, просмотра какие команды выполнялись в рамках этой сессии, нужно раскрыть знак «+». Для просмотра деталей выполнения команды следует дважды щелкнуть мышкой по интересующей команде.

Старт	Профиль
13.03.2024 08:03:21	igor
12.03.2024 11:03:21	parpar
04.03.2024 15:03:45	AFL-2.0
04.03.2024 14:03:15	AFL-2.0
04.03.2024 14:03:41	AFL-2.0
04.03.2024 14:03:08	
01.03.2024 11:03:21	LoadParquetToGreenplum
01.03.2024 08:03:48	LoadParquetToGreenplum
01.03.2024 08:03:39	LoadParquetToGreenplum
01.03.2024 08:03:06	LoadParquetToGreenplum
01.03.2024 08:03:35	LoadParquetToGreenplum
01.03.2024 08:03:44	LoadParquetToGreenplum
29.02.2024 11:02:36	LoadParquetToGreenplum
29.02.2024 07:02:53	PrimerModel
29.02.2024 07:02:43	PrimerModel
29.02.2024 07:02:19	PrimerModel
28.02.2024 14:02:29	ТестоваяМодельДанных
28.02.2024 09:02:33	LoadParquetToGreenplum
28.02.2024 08:02:17	LoadParquetToGreenplum
28.02.2024 08:02:13	LoadParquetToGreenplum

Рисунок 32 Страница Session (Сессии)

Имя	Источник	Объект источника	Целевая система	Имя объекта	Статус
sqlserver(dwh.fprocomp.ru)	SELECT TOP (1000) [id], [DocumentCode], [DocumentName], [DocumentNumber], [РегистрационныйНомер], [DocumentDate] FROM [BIQube].[ais].[DocumentCards];		DocumentCards		Ошибка

Рисунок 33. Состав сессии

Код	Время старта	Сообщение стейджинга	Сообщение ошибки	Трассировка стека	Команда для целевой системы	Статус
18252	12.03.2024 11:18	Старт программы загрузки				Старт
18253	12.03.2024 11:18	Установлено соединение с источником				Успех
18254	12.03.2024 11:18	Прочитаны метаданные источника				Успех
18255	12.03.2024 11:18	Выполнено соединение с целевой системой				Успех
18256	12.03.2024 11:18	Сгенерирована таблица: "demo"."TestEnum_1c_v2_temp"			create schema if not exists "demo"; drop table if exists "demo"."TestEnum_1c_v2_temp";	Успех
18257	12.03.2024 11:18	Загружена группа строк 1 из источника				Успех
18258	12.03.2024 11:18	Данные загружены в целевую систему			copy "demo"."TestEnum_1c_v2_temp" ("Ссылка", "П")	Успех
18259	12.03.2024 11:18	Успешно выполнено переименование для таблицы demo.			do language plpgsql \$\$ begin if exists ( select 1 from pg_tables where tablename = 'demo' ) then rename table demo to demo_1; end if; end \$\$	Успех
18260	12.03.2024 11:18	Исходные данные источника сохранены в промежуточно				Успех

Рисунок 34. Окно, демонстрирующее детали выполнения команды

## Данные

Страница Data (Данные) позволяет пользователю посмотреть визуально загруженные данные в хранилище, здесь же есть возможность выполнить

любые запросы, на основе которых можно убедиться в качестве полученных данных.

Справа в строке необходимо выбрать тот тип загрузки, который выбирали ранее. Затем раскрываем дерево файлов, нажатием на плюсик, и находим данные.

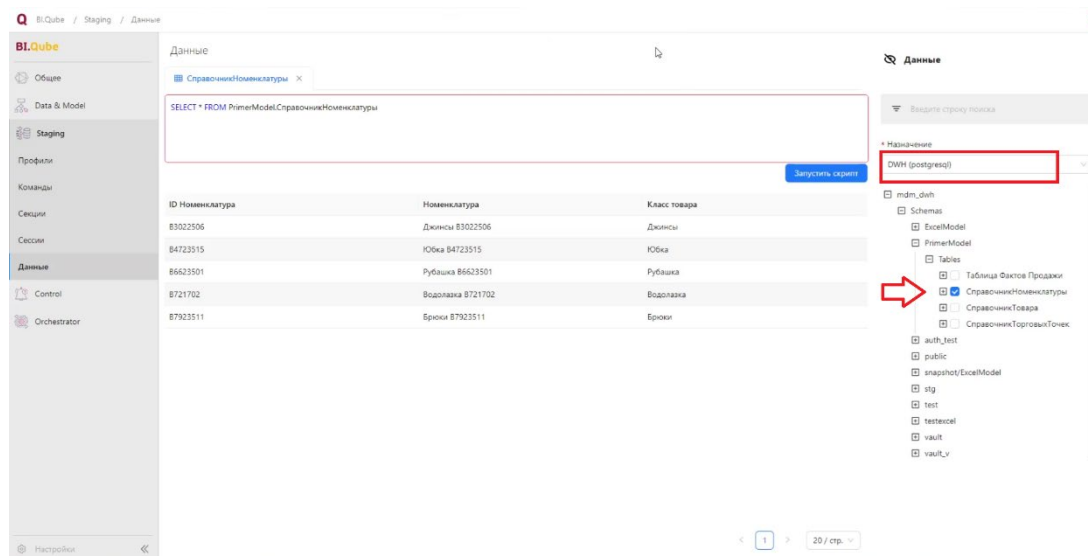


Рисунок 35. Просмотр загруженных данных

## DATA&MODEL

### *Введение*

Сущность (таблица) – это часть (элемент) модели, поэтому для создания сущности необходимо либо выбрать уже существующую модель, либо создать новую модель.

Создать новую сущность или отредактировать существующую можно в рабочей области с правой стороны. При этом сущности условно делятся на два типа:

- Пустые сущности – такие сущности создаются и заполняются вручную с использованием интерфейса BI.Qube эти сущности создаются компонентом MetaMasterData.
- Созданные на основе имеющейся таблицы или представления;

Кроме этого, доступен так называемый гибридный тип, когда к сущностям второго типа можно добавлять новые поля и редактировать данные в таких полях, редактировать или удалять поля и данные созданные автоматически на основе метаданных источников невозможно.

### *Профиль*

Страница Profiles (Профиль) предназначена для создания профилей, которые являются контейнерами для команд.

Для создания профиля нужно нажать кнопку Create (Создать) после чего в правой части экрана заполнить поля:

- Name (Имя) – имя профиля;
- Description (Описание) – бизнес-описание, указывается назначение данного профиля;
- Enabled (Видимость) – видимость профиля в остальных частях системы.

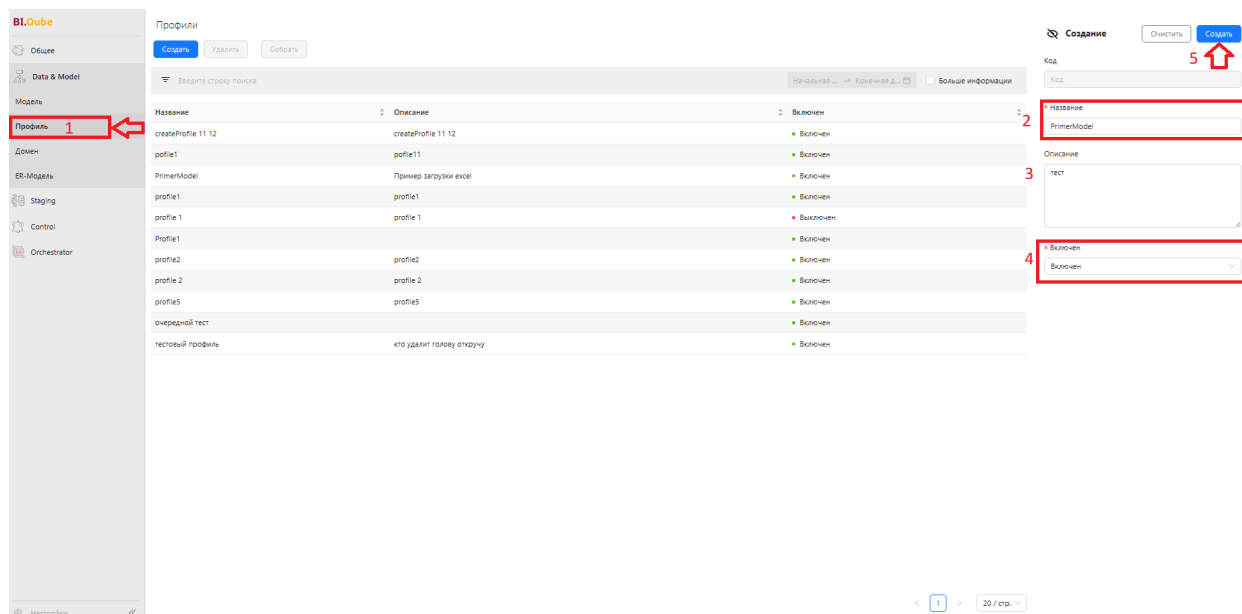


Рисунок 36. Создание профиля

Нумерацией на рисунке показан алгоритм выполняемых действий на странице. Для редактирования имени и/или описания профиля необходимо щелкнуть по строке в центральной части экрана мышкой и в правом окне внести нужные изменения, после чего нажать кнопку Update (Обновить).

## Домен

Для создания Domains (Домена) необходимо нажать кнопку Create (Создать), затем заполнить поля: Name (Название) без пробелов, Description (Описание) не обязательно для заполнения и снова нажать кнопку Create (Создать).

Для удаления домена нужно выделить одинарным щелчком левой кнопки мыши нужную строку и нажать на кнопку Delete (Удалить).



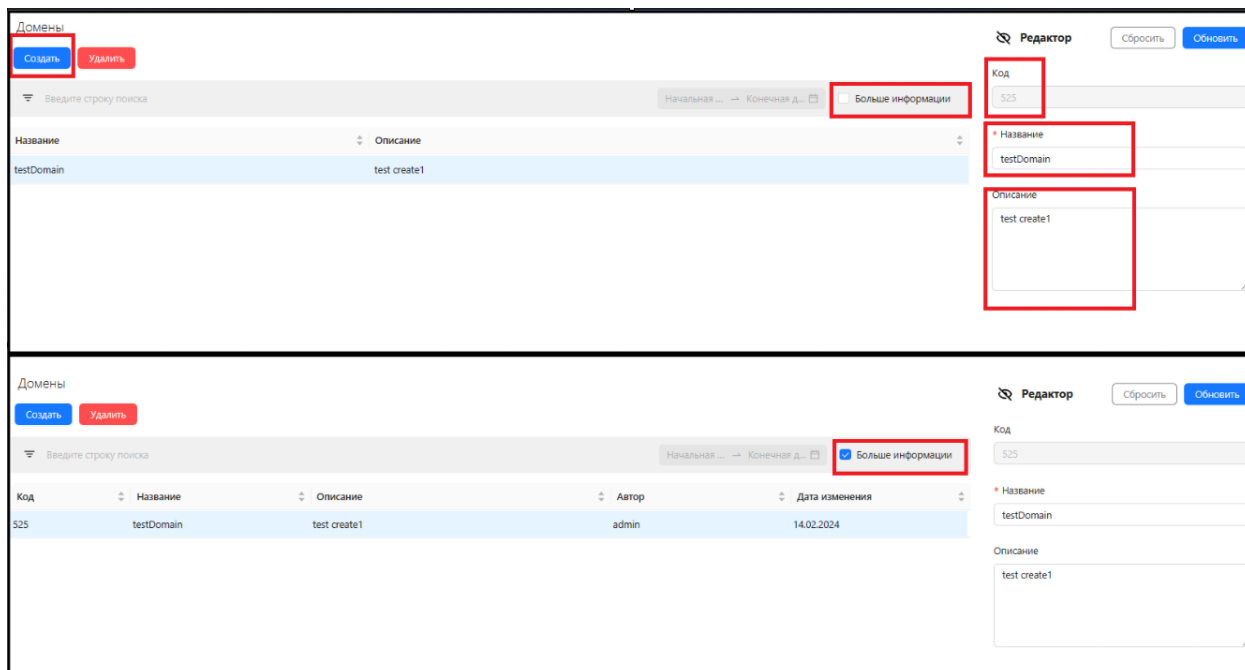


Рисунок 37. Пример заполненного домена и изменение полей отображения домена при нажатой more info (больше информации)

При нажатии More info (Больше информации) в центральном поле отображения появляются дополнительные столбцы: Id (Код), Owner (Автор), Modified date (Дата изменения).

### Создание модели

Создание модели данных осуществляется на странице Models (Модель), нажатием на кнопку Create (Создать) создаются поля для заполнения в правой части экрана. Необходимо заполнить поля:

- Name (Имя) – имя модели данных;
- Description (Описание) – бизнес-описание модели данных, как правило, дается описание назначения модели данных.

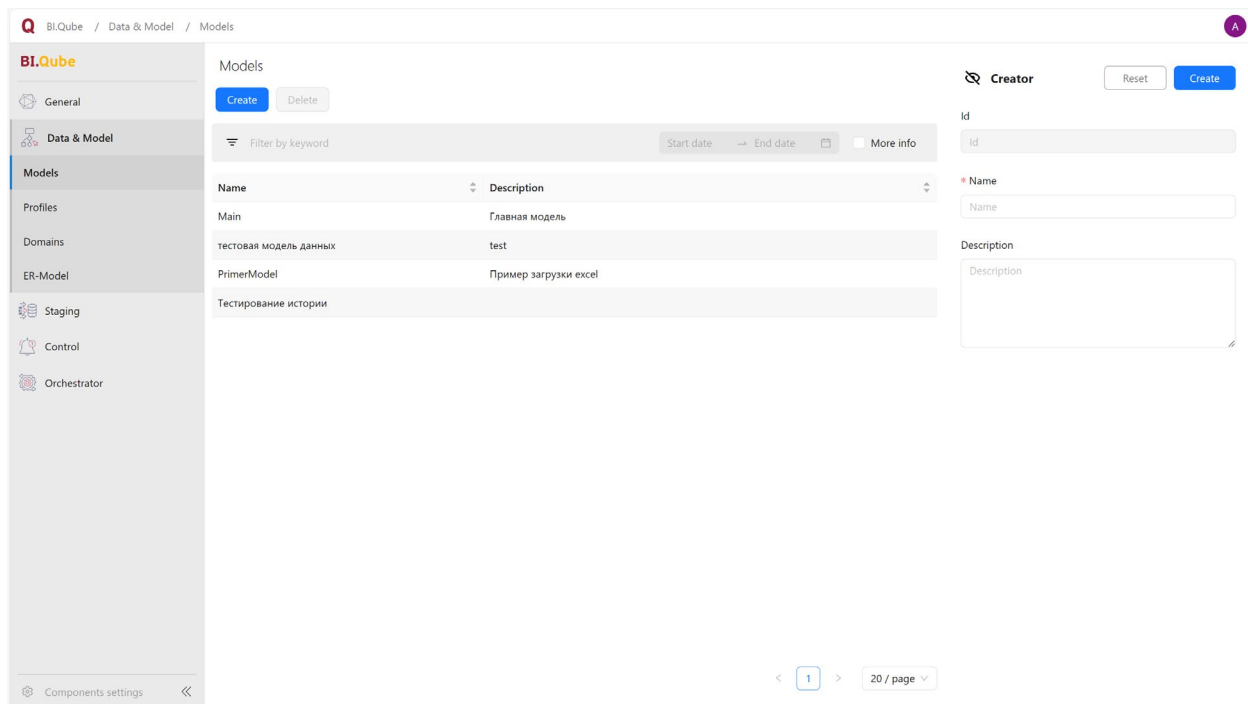


Рисунок 38. Создание модели данных

Редактирование имени и описание выполняется аналогичным образом, щелкнуть левой кнопки мыши по строке модели в центральной части экрана, внести в правой части, в окне свойств необходимые изменения и нажать кнопку Update (Обновить).

Для просмотра содержимого модели данных необходимо дважды щелкнуть левой кнопкой мыши по строке модели после чего на экране появится список сущностей входящих в эту модель.

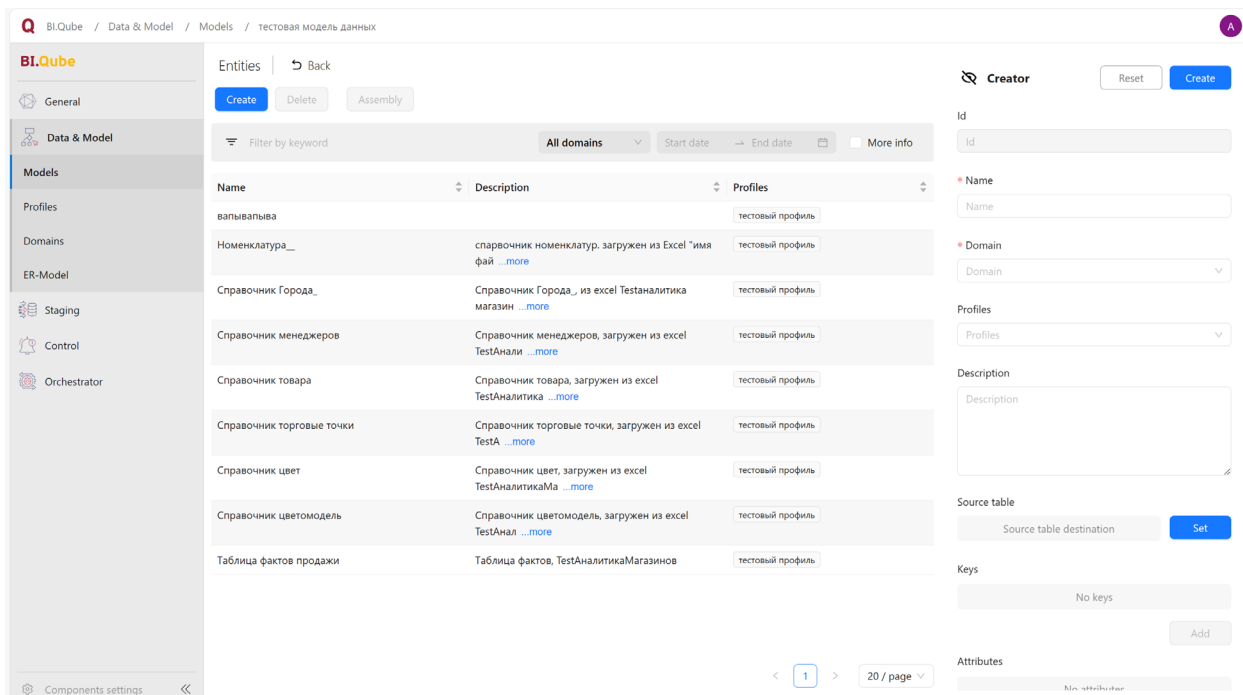


Рисунок 39. Сущности модели данных

## Создание сущности в модели

Создание сущности происходит стандартным образом необходимо, находясь в модели нажать на кнопку Create (Создать) справа в окне свойств появится перечень свойств которые нужно заполнить:

- Name (Название) – имя сущности;
- Domain (Домен) – выбрать домен в который будет входить сущность, одна сущность может принадлежать только одному домену;

- Profile (Профиль) – выбрать профиль к которому будет принадлежать сущность, сущность может принадлежать нескольким профилям;
- Description (Описание) – бизнес описание назначения сущности;
- Source table (Таблица-источник) – ссылка на таблицу (привязка источника данных к создаваемой сущности), из которой данные будут попадать в создаваемую сущность;
- Keys (Ключи) – создание ключевых полей сущности (доступно, только при наличии источника данных);
- Attributes (Поля) – поля создаваемой сущности. Поля создаются либо путем выбора команды «Add manual» (Создать ручной) для создания нового атрибута или команды «Add» (Добавить) скопировать атрибут из источника привязанного к создаваемой сущности;
- Links (Ссылки) – инструмент создания связей между сущностями;
- Materialize active view (Активное представление) – материализация данных в бизнес-представлении;

Код

Код

\* Название

Название

\* Домен

Домен

Профили

Профили

Описание

Описание

Таблица-источник

Целевая таблица-источник

Выбрать

Ключи

Нет ключей

Добавить

Поля

Нет полей

Добавить ручной

Добавить

Ссылки

Нет ссылок

Добавить

Настройки

Активное представление

false

Историчное представление

false

Рисунок 40. Свойства сущности

- Materialize historical view (историческое представление) - материализация данных изменений в бизнес представлении.

Создание пустой сущности в хранилище чаще всего происходит для создания новых данных (справочников), нормативно-справочной информации (НСИ), которых нет в имеющихся учетных системах. Такие сущности (справочники) обычно заполняются в ручном режиме, с клавиатуры оператором системы и используются как «центр правды» для всех остальных учетных систем.

Для пустых сущностей не нужно выбирать таблицу-источник и создавать ключи, необходимо сразу нажать на кнопку Add manual (Добавить ручной) в результате на экране появится окно Creating attribute (Добавление поля) в котором ввести имя создаваемого атрибута, выбрать свойство Nullable (Обязательный), выбрать тип данных и, если доступно, указать свойства выбранного типа данных.

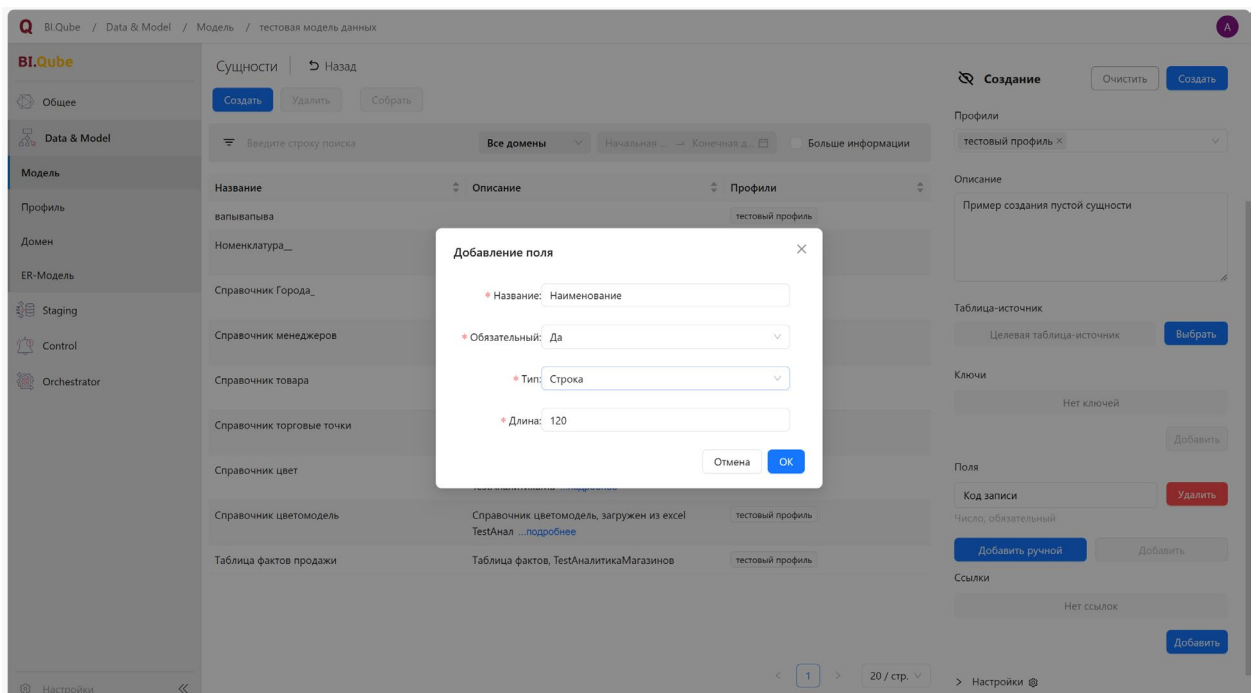


Рисунок 41. Создание атрибутов сущности

Система поддерживает достаточно разнообразный набор типов данных, который зависит от месторасположения хранилища (PostgreSQL, MS SQL).

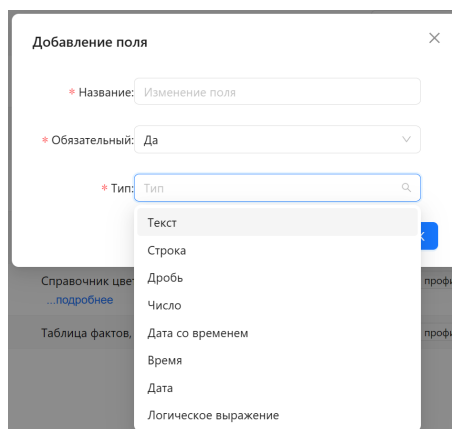


Рисунок 42. Доступные типы данных

После создания сущности необходимо нажать кнопку Create (Создать), новая запись о созданной сущности появится в списке сущностей текущей модели данных.

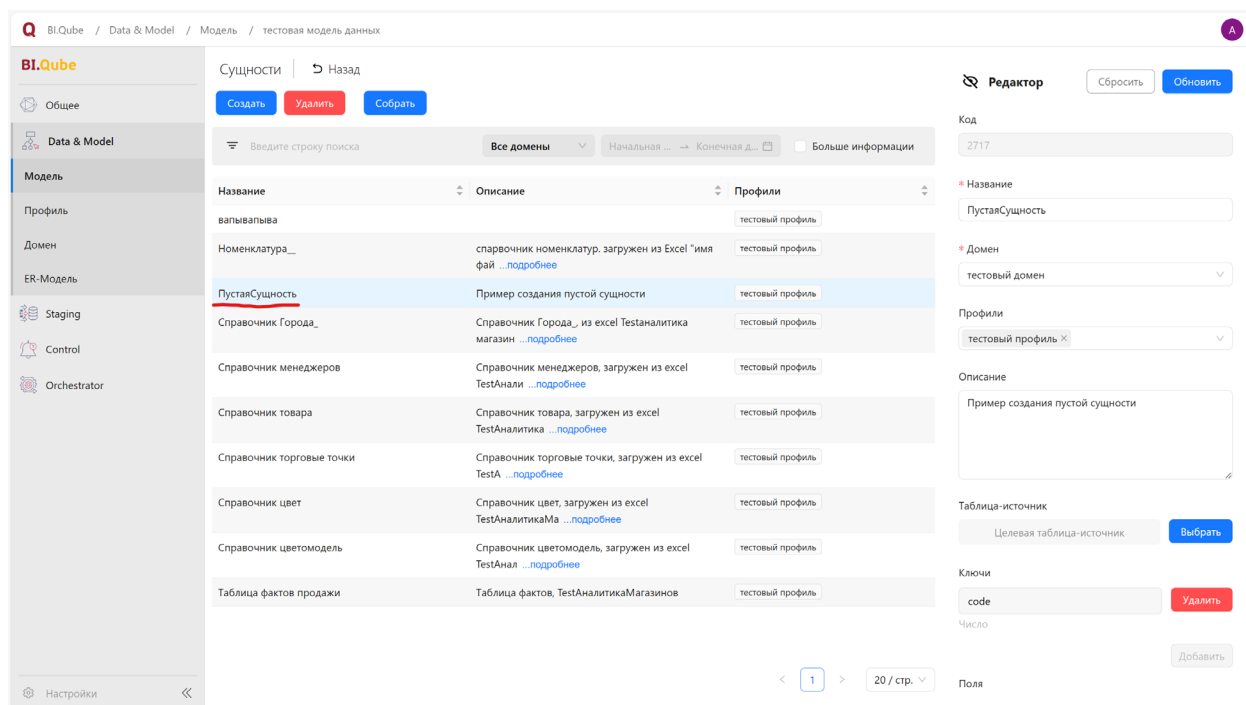


Рисунок 43. Созданная сущность

Для создания сущности, которая может быть заполнена данными из источника данных необходимо указать источник данных (Таблица-источник) для это нужно нажать кнопку Set (Выбрать), появляется диалоговое окно. Данное окно настроено по умолчанию на определённую базу данных, в которой могут находиться таблицы источники данных. Вверху в выпадающем списке выбрать схемы данных базы данных, после чего указать таблицу,

данные из которой будут загружаться в создаваемую сущность. После сделанных настроек нажать кнопку Set (Выбрать).

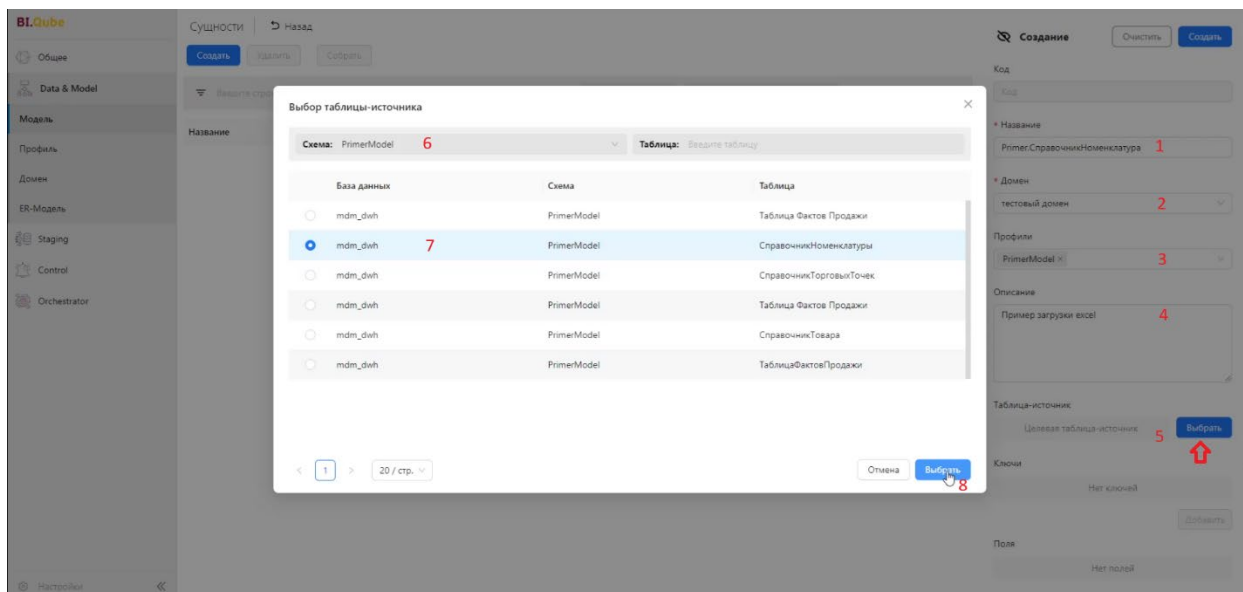


Рисунок 44. Описание заполнения сущности

После указания таблицы-источника появляется возможность создать ключевые поля сущности для этого для поля Keys (Ключи) следует нажать кнопку Add (Добавить) и в появившемся диалоговом окне выбрать тот ключ, который является уникальным для создаваемой сущности – поставить галочку напротив него и нажать на кнопку Add (Добавить).

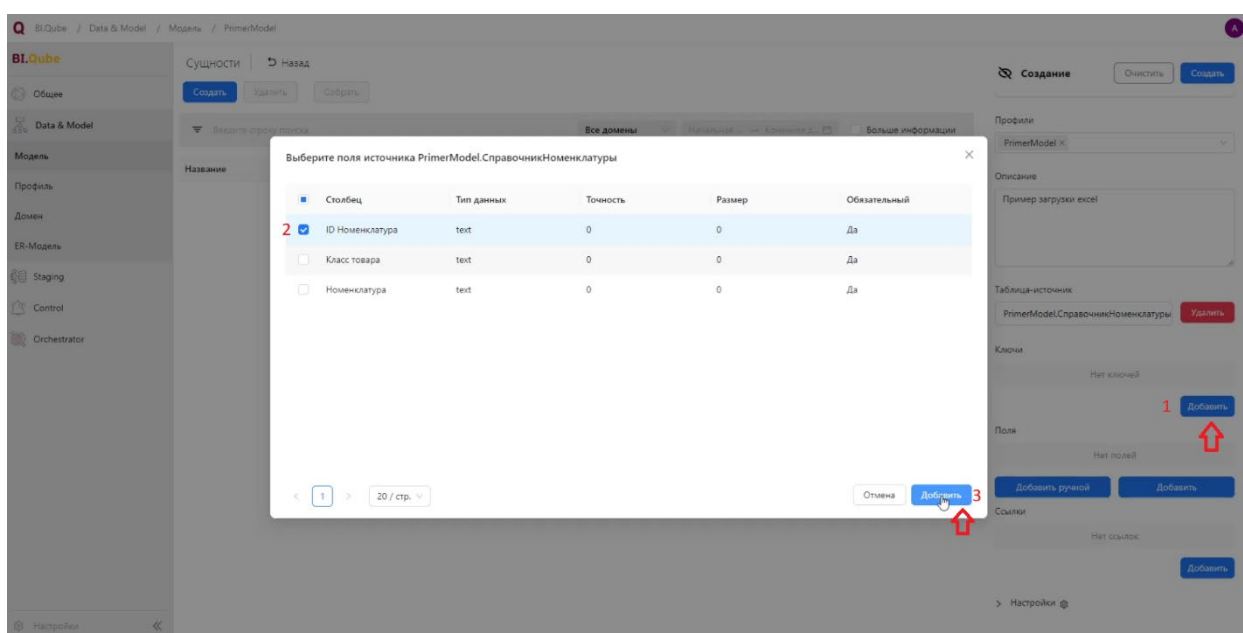


Рисунок 45. Заполнение поля «Ключи»

После создания ключа, который может быть составным, т.е. состоять более чем из одного поля, можно создать остальные поля, при этом не все поля из источника могут попасть в создаваемую сущность. Выбрать команду Add (Добавить) появится диалоговое окно Selecting source attributes (Выбор поля источника), в котором будут перечислены поля таблицы, ранее привязанной к создаваемой сущности и доступные для добавления, затененные поля не доступны для выбора так как они уже добавлены в качестве ключа.

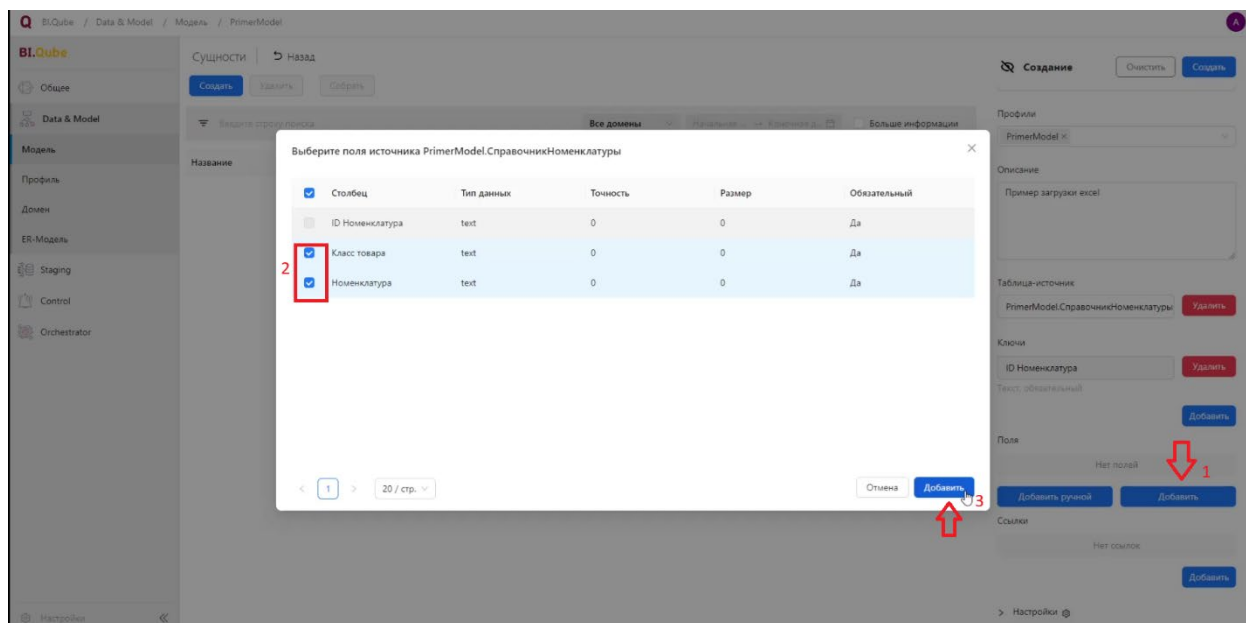


Рисунок 46. Поэтапное добавление полей

За один шаг можно создать сразу все нужные поля, для каждого поля в окне свойств будут созданы отдельные записи.

## Просмотр и редактирование данных

Просмотр содержимого сущности – данных, осуществляется после двойного нажатия левой кнопки мыши по строке сущности в модели. Происходит проваливание в сущность (таблицу) для просмотра данных и редактирования полей в созданных атрибутах.

В этом режиме доступно построчное редактирование, при этом данные, которые попали из таблицы-источника не могут быть изменены. Заполнять можно только те поля, которые не привязаны к таблице-источнику редактировать сколько угодно раз. При этом, система сохраняет всю историю



изменений выполнимых пользователем, после заполнения полей выбранной строки в окне свойств необходимо нажать кнопку Update (Обновить).

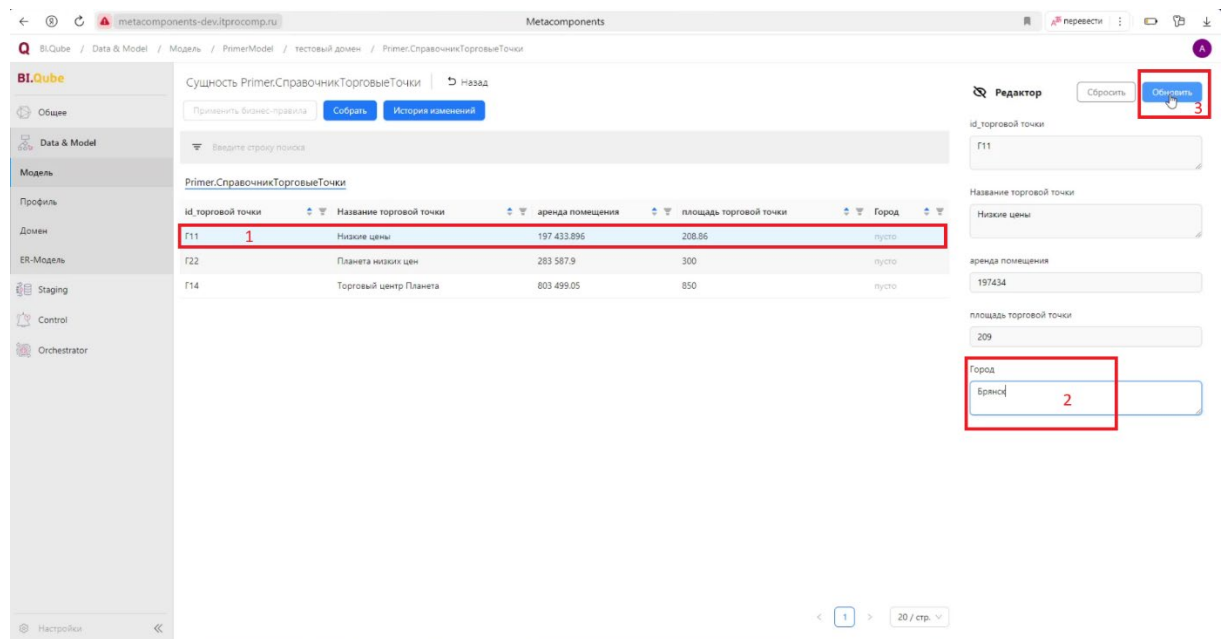


Рисунок 47. Заполнение добавленного поля в сущности (таблице)

Для просмотра изменений в строке необходимо нажать кнопку History changes (История изменений), появится окно, в котором отобразятся все изменения данных этой строки при этом каждое последующее изменение относительно предыдущего выделяется цветом.

История изменений

code	f1	f2	hh	modified_by	modified_at	date_from	date_to	deleted
2	пусто	пусто	{ 1, 9, empty, empty }	admin	2024-03-13T12:37:27	0001-01-01T00:00:00Z	2024-03-13T14:16:31Z	false
2	4	8	{ 1, 9, empty, empty }	admin	2024-03-13T14:16:31	2024-03-13T14:16:31Z	2024-03-14T12:34:25Z	false
2	4	8	{ 1, 9, empty, empty }	admin	2024-03-14T12:34:24	2024-03-14T12:34:25Z	2024-03-14T12:35:14Z	false
2	4	8	пусто	admin	2024-03-14T12:35:14	2024-03-14T12:35:14Z	2024-03-14T12:36:56Z	false
2	4	8	{ 2, 4, выфмывмыав, empty }	admin	2024-03-14T12:36:55	2024-03-14T12:36:56Z	2024-03-14T12:39:34Z	false
2	4	пусто	{ 2, 4, выфмывмыав, empty }	admin	2024-03-14T12:39:34	2024-03-14T12:39:34Z	2024-03-14T12:54:47Z	false
2	4	пусто	пусто	admin	2024-03-14T12:54:46	2024-03-14T12:54:47Z	2024-03-14T12:55:39Z	false
2	4	пусто	{ 2, 4, выфмывмыав, empty }	admin	2024-03-14T12:55:38	2024-03-14T12:55:39Z	2024-03-14T12:56:12Z	false
2	4	пусто	пусто	admin	2024-03-14T12:56:12	2024-03-14T12:56:12Z	2024-03-14T12:59:18Z	false
2	4	4	пусто	admin	2024-03-14T12:59:17	2024-03-14T12:59:18Z	2024-03-18T15:27:32Z	false
2	4	выфмывмыав	пусто	admin	2024-03-18T15:27:31	2024-03-18T15:27:32Z	9999-01-01T00:00:00Z	false

1

20 / стр.

Рисунок 48. Просмотр истории изменения записей

В созданной модели данных существует функция фильтрации данных. Чтобы перейти к фильтрации необходимо двойным щелчком левой кнопки мыши щёлкнуть по модели, затем выбрать также двойным щелчком левой кнопки мыши необходимую сущность (таблицу).

BI.Qube

Общие

Data & Model

Модель

Профиль

Домен

ER-Модель

Staging

Control

Orchestrator

Настройки

Сущность Номенклатура\_

Назад

Применить бизнес-правила

Собрать

История изменений

Введите строку поиска

Номенклатура

ID Номенклатура	Номенклатура	Класс товара
B1722528	Блузка B1722528	Блузка
B1723507	Блузка B1723507	Блузка
B1723510	Блузка B1723510	Блузка
B1723515	Блузка B1723515	Блузка
B1723518	Блузка B1723518	Блузка
B1723519	Блузка B1723519	Блузка
B1723520	Блузка B1723520	Блузка
B1723521	Блузка B1723521	Блузка
B1723522	Блузка B1723522	Блузка
B1723524	Блузка B1723524	Блузка
B1723525	Блузка B1723525	Блузка
B9323501	Боди B9323501	Боди
B2922530	Брюки B2922530	Брюки
B2923503	Брюки B2923503	Брюки
B2923506	Брюки B2923506	Брюки
B2923508	Брюки B2923508	Брюки
B2923509	Брюки B2923509	Брюки
B2923511	Брюки B2923511	Брюки

1

20 / стр.

Рисунок 49. Данные выбранной сущности с отображением по столбцам

Диалоговое окно фильтрации открывается одинарным нажатием левой кнопкой мыши по значку «Фильтр». Filter (Фильтр) можно настраивать для каждого столбца. В появившемся диалоговом окне нужно выбрать из выпадающего списка необходимый вид фильтрации и нажать кнопку Apply (Применить).

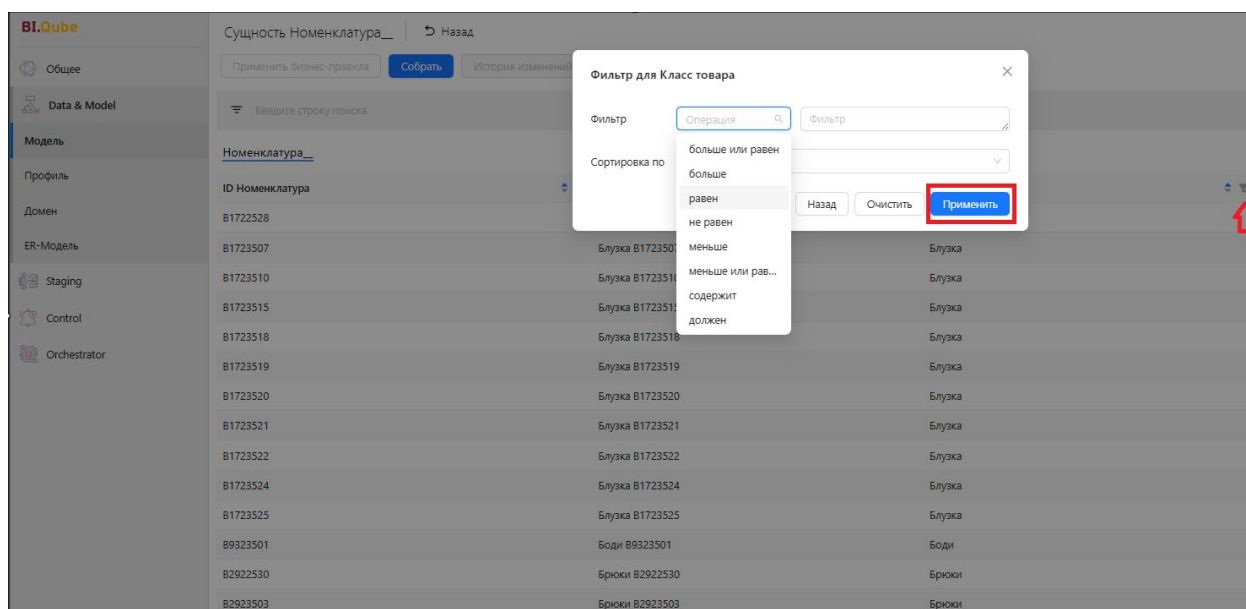


Рисунок 50. Фильтрация сущностей в модели

Важно! Поиск по словам осуществляется с учётом регистра.

### Создание связей между сущностями

Для создания связей («линков») между сущностями необходимо зайти в свойства сущности, к которой будут привязываться другие сущности.

В зоне Links (Связи) необходимо выбрать команду Add (Добавить) и заполнить появившееся диалоговое окно.

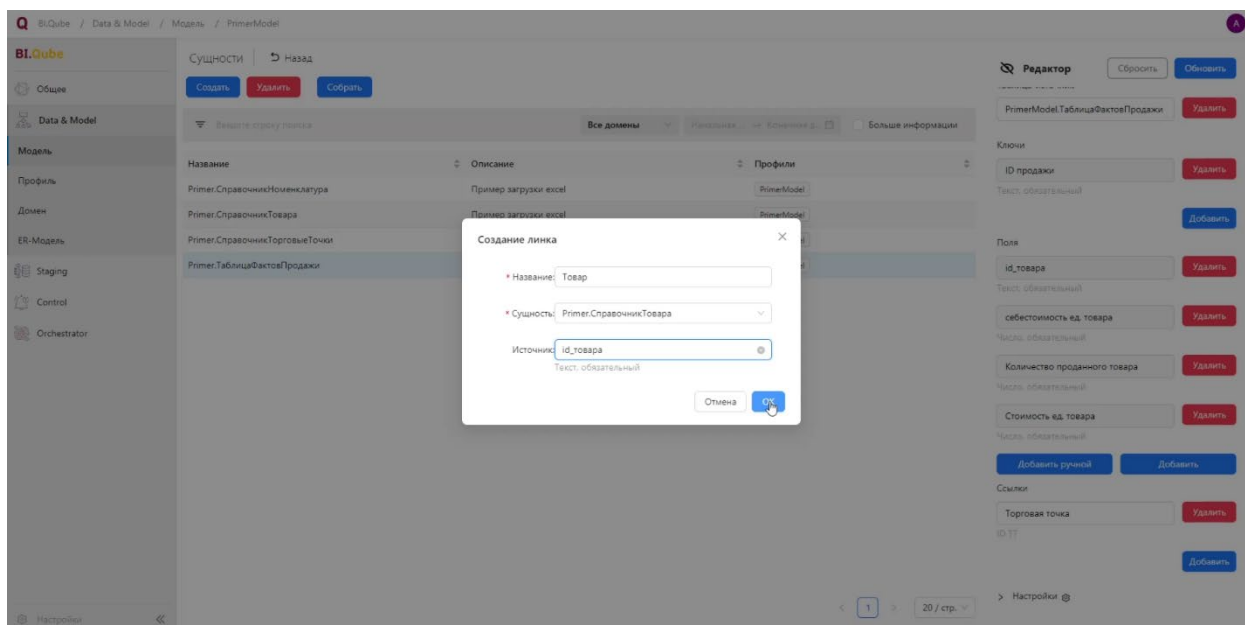


Рисунок 51. Создание линка (связи)

Следует указать имя связи, выбрать связываемый объект, и выбрать атрибут текущей сущности, к которому привязываются данные сторонней сущности. Следует отметить, что связываемая сущность связывается с текущей только по бизнес ключу об этом, следует помнить при создании бизнес-ключей.

## Сборка сущности

После создания сущности необходимо её собрать, операция сборки запускает ряд внутренних процедур, связанных с формированием большого количества программного кода, представления сущности в модель DataVault. Так же для сущностей, созданных на основе таблицы-источника происходит загрузка данных из источника.

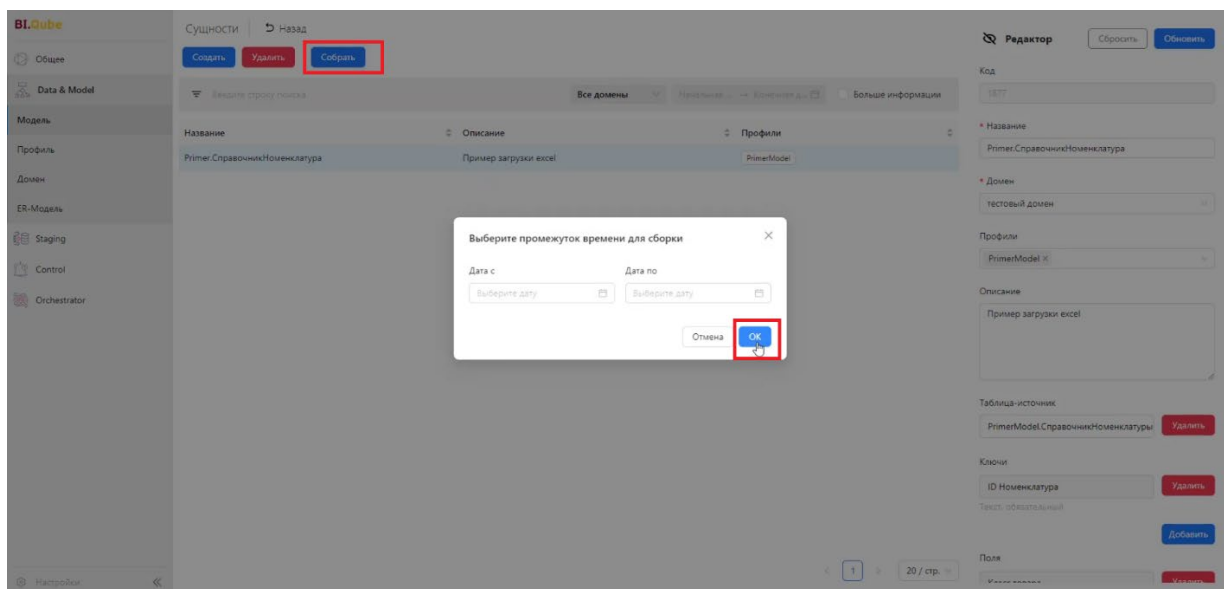


Рисунок 52. Сборка сущности

В основном окне нажать кнопку «Собрать», затем в появившемся диалоговом окне нажать «ОК». Содержание можно посмотреть двойным щелчком левой кнопки мыши.

В сущность, созданную на основе таблицы-источника можно добавлять атрибуты, не привязанные к таблице источнику, такие атрибуты в последующем можно будет заполнить данными в ручном режиме.

### Работа с моделью в графическом режиме

Система BI.Qube предоставляет пользователю возможность работы в графическом режиме. В таком режиме можно визуальное увидеть весь состав модели, все связи посмотреть свойства созданных сущностей и создать новые, здесь же можно увидеть, как раскладываются метаданные на объекты модели DataVault. Для всех этих задач используется страница Er-model (Er-модель).

Для просмотра графического представления модели необходимо выбрать модель, а также один или более доменов, нажать кнопку Load ER-model (Загрузить ER-модель). В данной модели каждый графический объект несет определенный смысл, так прямоугольниками показаны сущности, внутри прямоугольников могут быть перечислены атрибуты сущности

(зависит от режима отображения), линии между прямоугольниками символизируют связи



Рисунок 53. Выбор ER – модели в BI.Qube

При нажатии на любой графический объект в окне свойств справа появляется возможность редактирования свойств выбранного объекта.

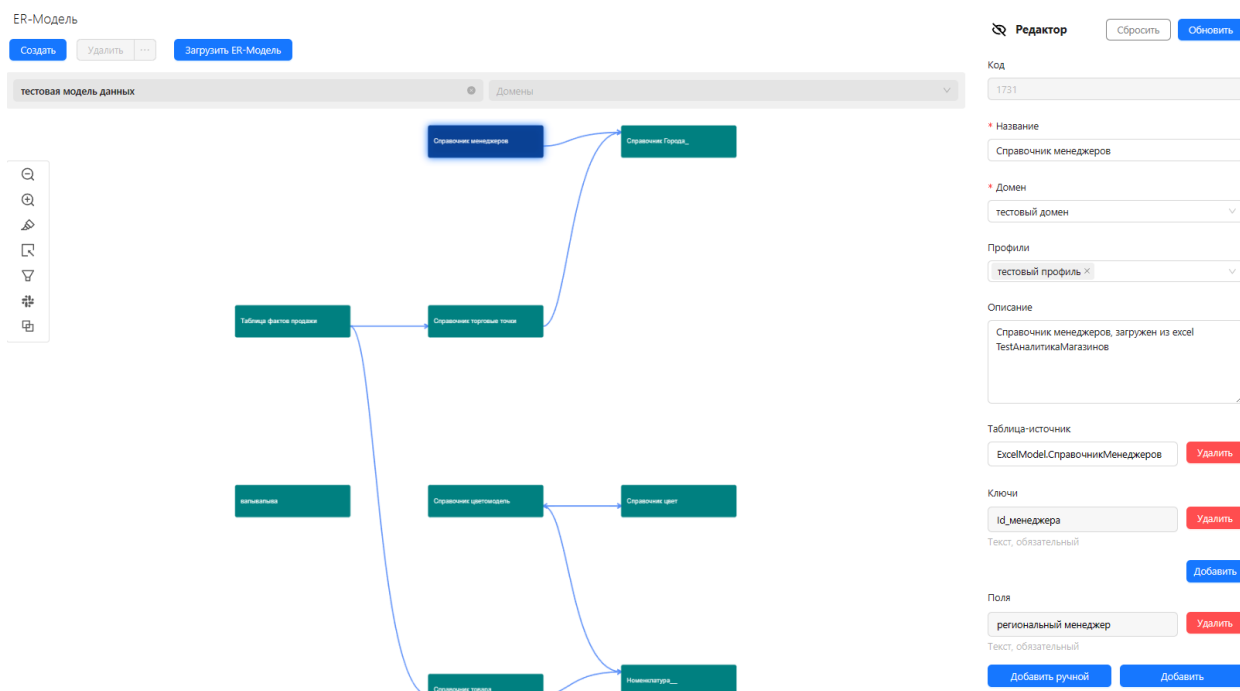


Рисунок 54 Редактирование таблицы в ER – модели



Рисунок 55. Панель инструментов (1 и 2 – лупа (уменьшить/увеличить); 3 – подсветка связей; 4 – фильтр слоёв; 5 – фильтр связей; 6 – режим отрисовки ER – модели; 7 – макет)

При выборе в панели инструментов фильтра связи предлагается возможность выбора объектов для отображения в ER – модели. Формат отображения зависит от режима отрисовки (концептуальная, детальная модель, DataVault).

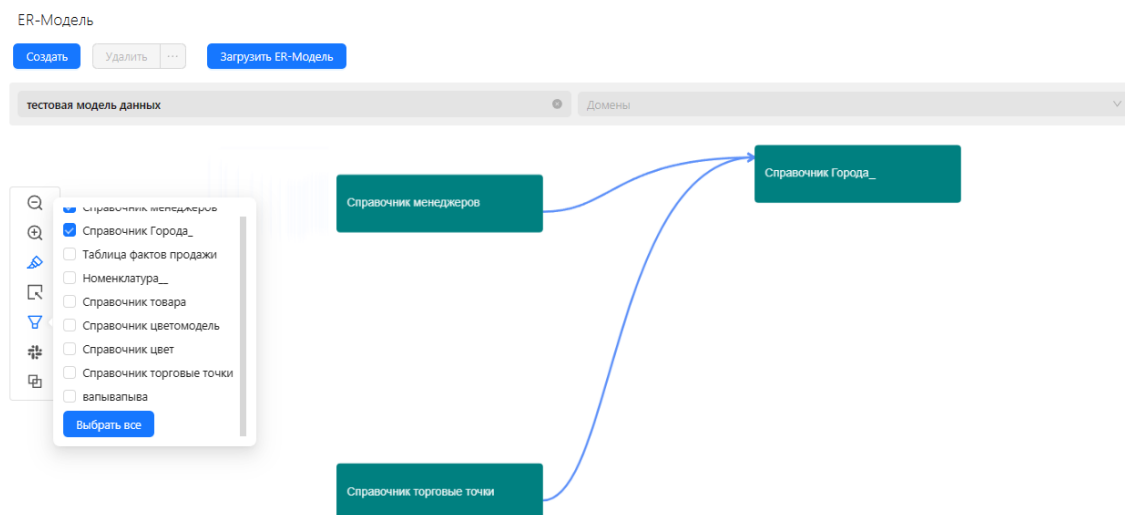


Рисунок 56. Работа с фильтром связей в режиме концептуальная модель

При визуализации ER – модель имеет три режима отображения модели:

- conceptual model (концептуальная модель) – в этом режиме отображаются все сущности в виде прямоугольников со связями;
- detail model (детальная модель) – в этом режиме в прямоугольниках отображаются атрибуты сущности;

- Data Vault – в этом режиме отображаются все объекты модели DataVault.

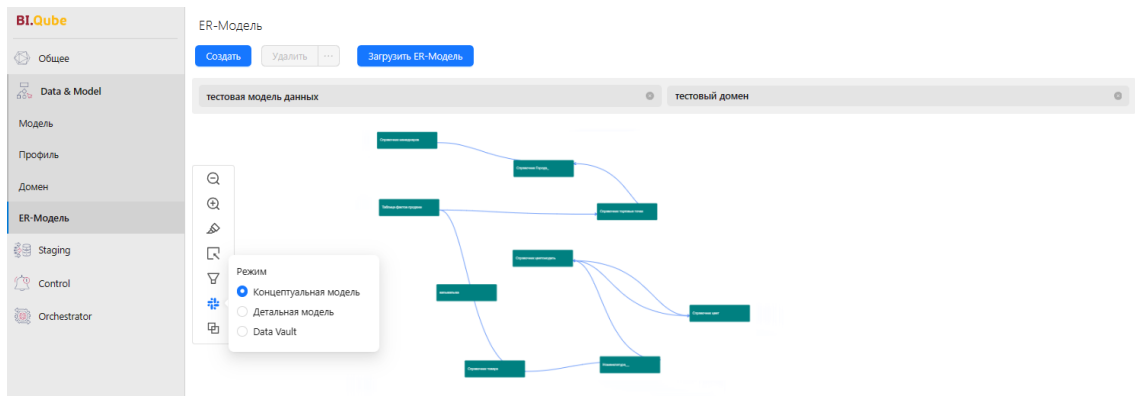


Рисунок 57. Пример отображения связей в концептуальной модели

Концептуальная модель данных удобна для отображения большого количества сущностей. Она определяет структуру моделируемой системы, свойства её элементов и причинно-следственные связи, присущие системе и существенные для достижения цели моделирования.

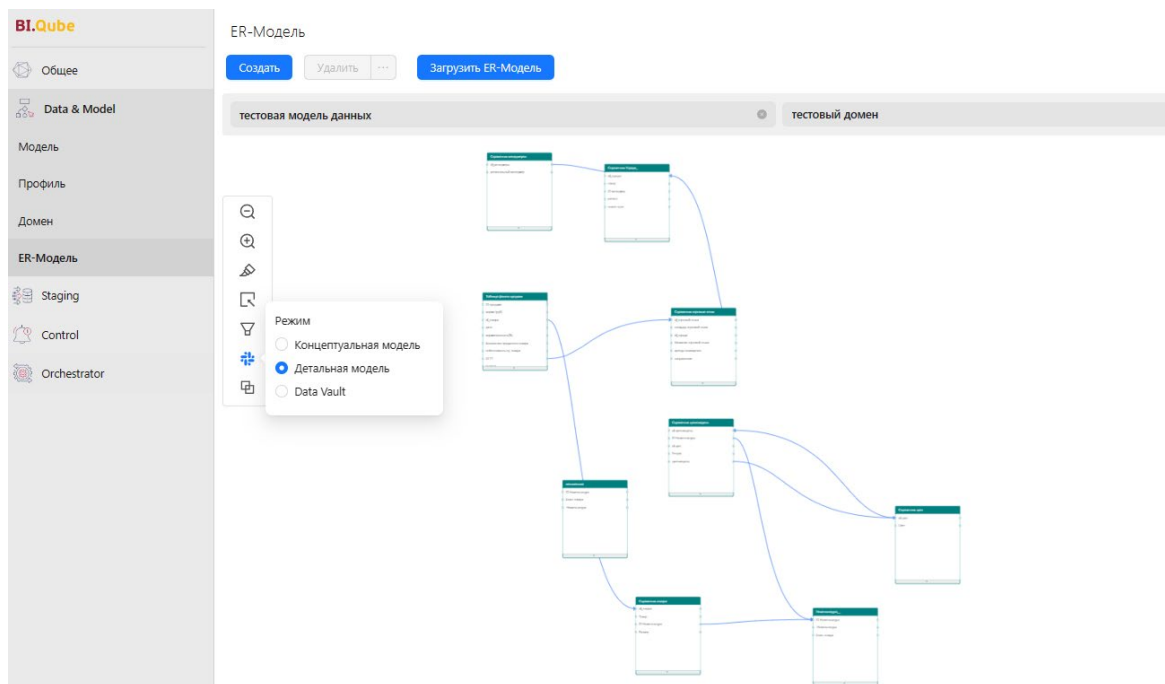


Рисунок 58. Пример отображения связей в детальной модели

Детальная модель позволяет пользователю более конкретизировано рассмотреть связи между полями сущностей для решения поставленных задач.



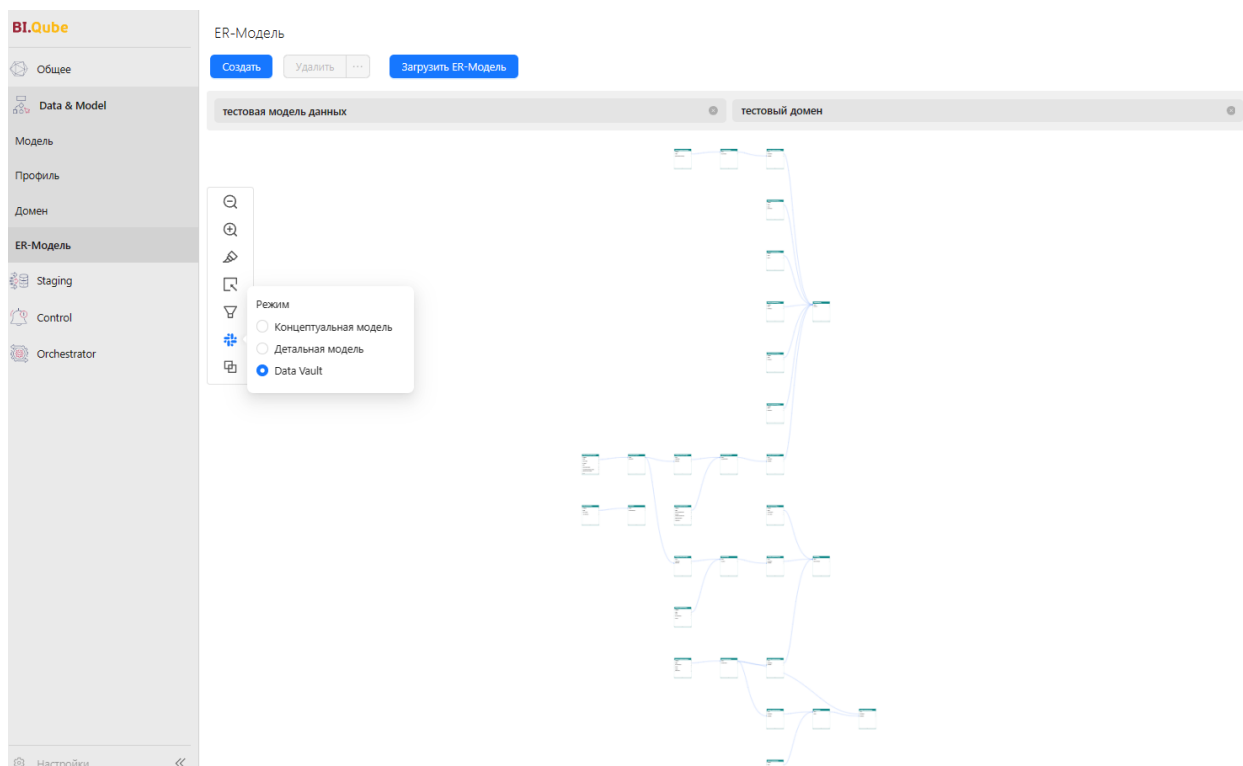


Рисунок 59. Пример отображения DataVault

При работе с ER – моделью возможен выбор двух вариантов отображения макета (слои, концентрический).

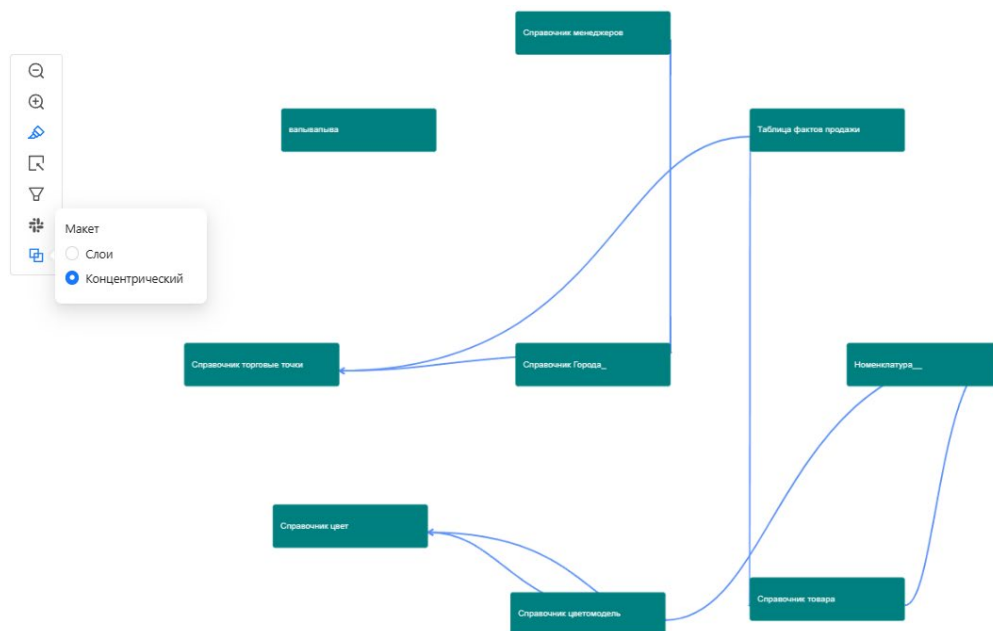


Рисунок 60. Концентрический макет

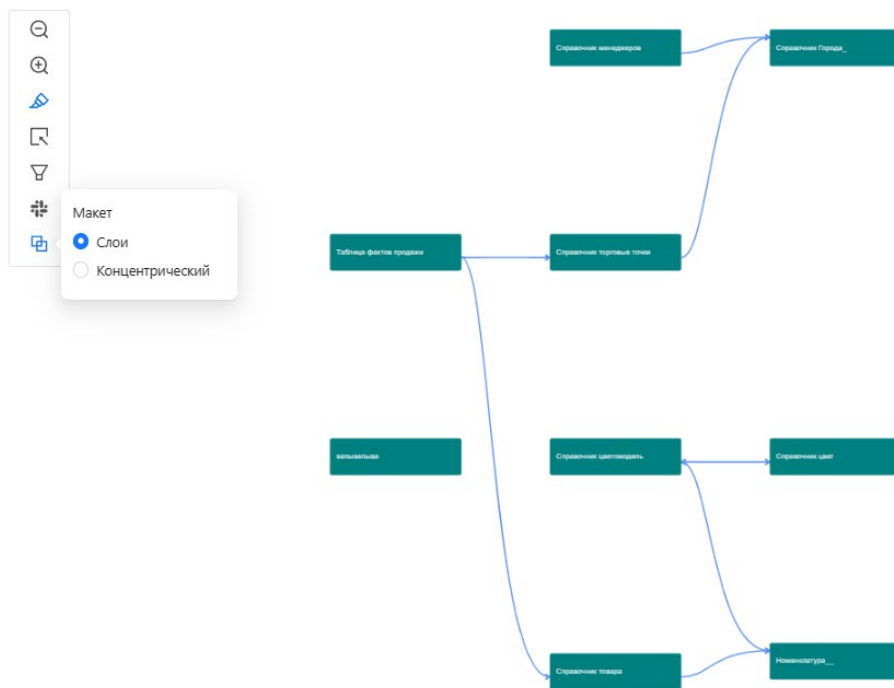


Рисунок 61. Макет в режиме «слои»

# METACONTROL

## *Введение*

MetaControl применяется, когда необходимо своевременно получать уведомления о статусе событий, происходящих с данными в хранилище. Другими словами, компонент позволяет выполнять заранее подготовленные бизнес-правила, представленные в виде SQL-запросов или процедур на определенных этапах ETL с целью выполнения контроля за происходящими действиями.

Результатом работы является отправка уведомлений в удобочитаемой форме для человека содержащие сообщения о результатах выполнения-бизнес правила. Уведомления отправляются на электронную почту и/или в telegram-канал.

## *Профиль*

Страница Profiles (Профили данных) предназначена для создания и редактирования содержимого профиля. Профиль предназначен для хранения общей справочной информации. В проекте профиль может быть один, может быть несколько, используется для распараллеливания задач для загрузки данных. Все дальнейшие настройки обязательно привязываются к выбранному профилю.

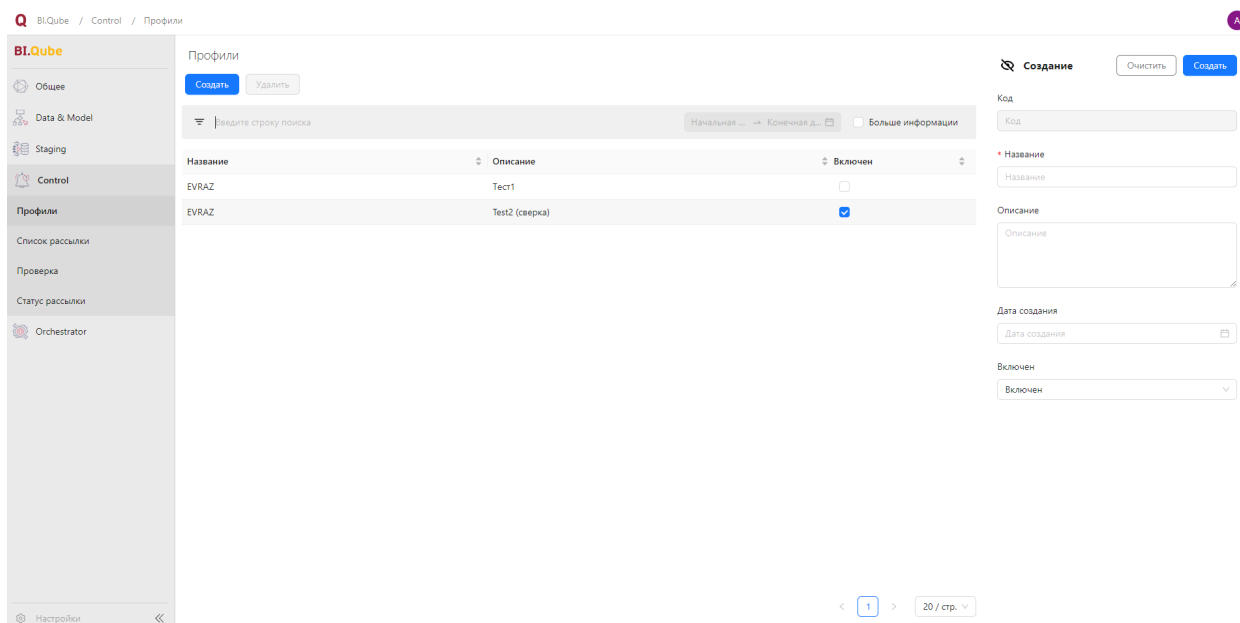


Рисунок 62. Профиль в MetaControl

Для создания профиля необходимо нажать на кнопку Create (Создать) над таблицей в центральной части экрана. Справа в окне свойств заполнить поля для заполнения:

- Name (Название) - имя профиля;
- Description (Описание) - описание профиля;
- Start date (Дата создания) заполняется автоматически — это дата создания профиля, предусмотрена возможность задать желаемую дату из календаря.

Далее нужно поставить значение «галочку» в поле Enabled (Включен) для активации нужного профиля. Профиль в любой момент может быть отключен тогда

Рисунок 63. Окно свойств для создания Profiles (Профиля)

задачи данного профиля не будут выполняться.

При необходимости нажать кнопку More info (Больше информации), в этом случае в таблице появятся дополнительные поля: Owner (Автор) и Start date (Дата создания).

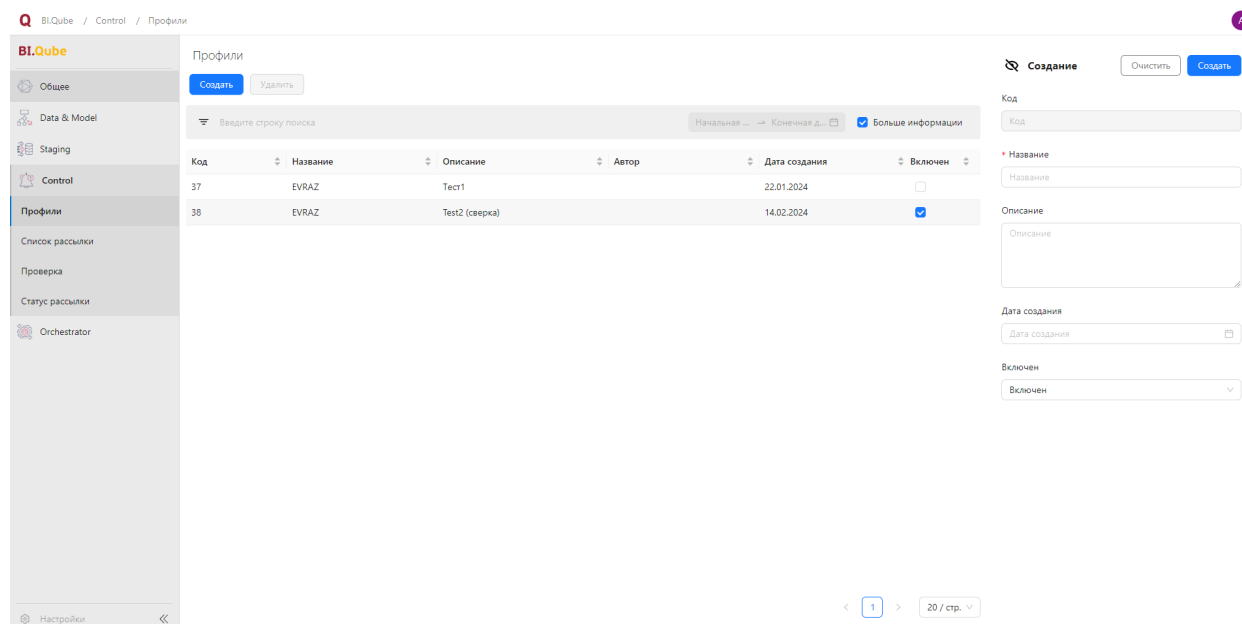


Рисунок 64. Содержимое профиля с включённой функцией More info (Больше информации)

Далее нужно нажать кнопку Create (Создать), чтобы завершить создание профиля. Данные созданного профиля появятся в таблице.

Процесс редактирования профиля аналогичен созданию профиля – работа по редактированию производится в правом окне свойств. Для редактирования определённого профиля его нужно выделить нажатием на строку

### Список рассылки

Страница Mailing list (Список рассылки) предназначена для создания и редактирования справочника адресов. При этом, список рассылок к профилям не привязывается.

Главное меню слева, справа окно свойств, по центру основные данные, представленные в виде таблицы

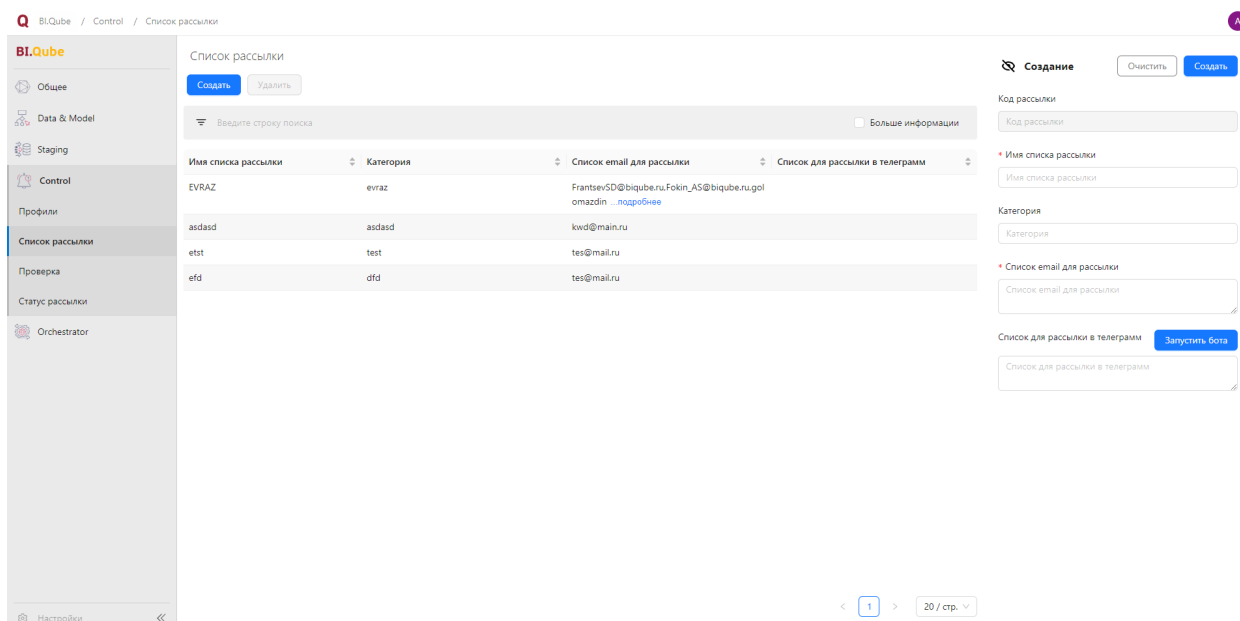


Рисунок 65. Страница для создания Mailing list (Список рассылки)

Для создания списка адресов необходимо нажать на строку таблицы Create (Создать) двойным щелчком. Заполнить данные создаваемого источника в таблице Mailing list (Список рассылки):

- Mailing list name (Имя списка рассылки);
- Category (Категория);
- Mailing list (Список email для рассылки) внести почту получателей рассылки;
- Telegram list (Список рассылок телеграмм) имя пользователя (login, учетная запись) в Telegram.

Далее нажать на значок Run bot (Запустить бота) (<https://t.me/MetaControlComponentTGBot>) и написать с указанного в поле Telegram\_list (Список для рассылки в телеграм) аккаунта слово «start». Нажать кнопку Create (Создать) чтобы завершить создание списка рассылки.

Для удаления нужно выделить список адресов нажатием на строку и нажать кнопку Delete (Удалить).

Редактирование осуществляется аналогично созданию списка рассылки – работа по редактированию производится в правом окне свойств. Для этого нужно выделить список адресов нажатием на строку.

## Проверка

Страница Validation (Проверка) предназначена для создания и редактирования условий контроля.

Главное меню слева, справа окно свойств, по центру основные данные, представленные в виде таблицы.

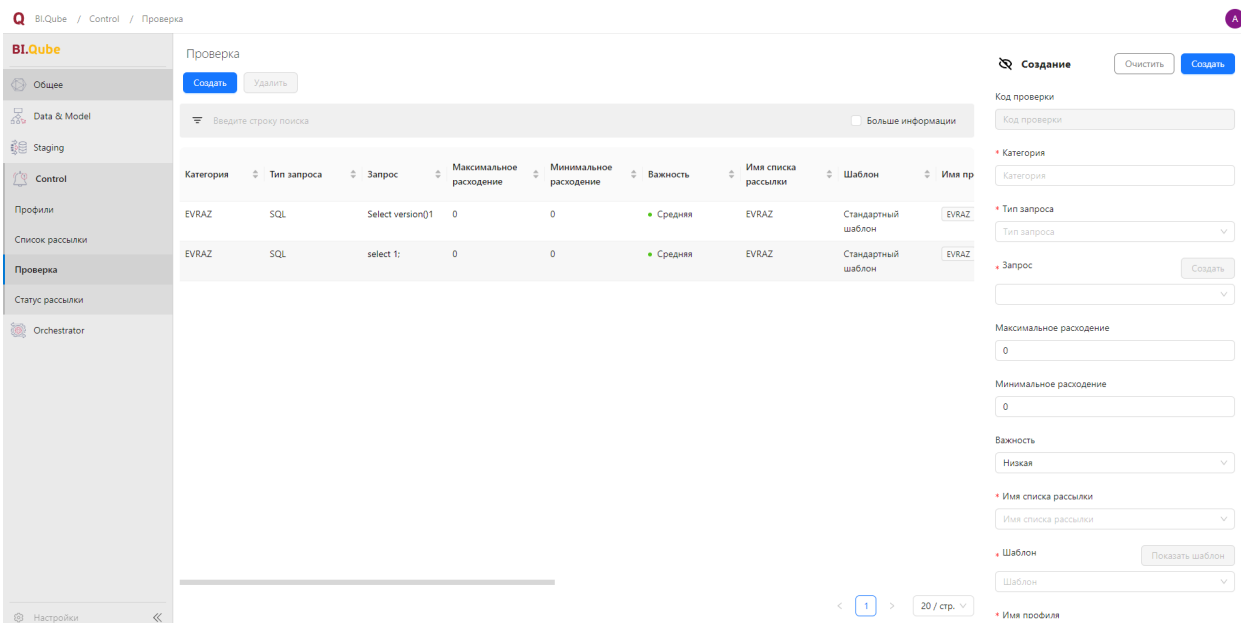


Рисунок 66. Содержание страницы Validation (Проверка)

Для создания контроля необходимо нажать на строку таблицы Create (Создать) двойным щелчком. Заполнить данные создаваемого источника в таблице Validation (Проверка):

- Category (Категория) - наименование категории контроля;
- Выбрать в выпадающем списке «type of query» (это может быть процедура, функция или представление);
- Query (Запрос);
- Min threshold (Минимальное расхождение);
- Max threshold (Максимальное расхождение).

Выбрать из выпадающего списка:

- Importance (Важность);
- Mailing name (Имя списка рассылки);

- Template (Шаблон) - это тип вывода: вывод минимальной информации о выполнении контроля, вывод полной информации по контролю, вывод расхождений по контролю;
- Profile name (Имя профиля). Выбор должен осуществляться по наименованию уже созданных профилей.

Заполнить поля:

- Description (Описание);
- Mail text (Описание текста письма);
- Default report (Отчет по умолчанию) для пометки стандартного отчета;
- Send data in email (Отправить по электронной почте) для вкл/откл отправки всех данных из проверки;
- Max rowcount data in email (Максимальное количество строк в письме) Появляется после включения Send data in email (Отправить по электронной почте);
- Xslt templates (Xslt Шаблон). Появляется после включения Send data in email (Отправить по электронной почте);
- Url report (Отчет).

Нажать кнопку Create (Создать) чтобы завершить создание контроля.

Для удаления контроля его необходимо выделить и нажать кнопку Delete (Удалить).

Редактирование осуществляется аналогично созданию, при выделении нужного контроля в таблице.

### *Ст ат ус рассылки*

Страница Mailing status (Статус рассылки) предназначена для запуска и отслеживания статус по контролю.



Главное меню слева, справа окно свойств, по центру основные данные, представленные в виде таблицы

BI.Qube / Control / Статус рассылки

Статус рассылки

Начать

Имя профиля	Категория	Отправлено
—	EVRAZ	false
	EVRAZ	
+	EVRAZ	false

Настройка <<

< 1 > 20 / стр. ▾

Рисунок 67. Содержимое страницы Mailing status (Статус рассылки)

Для проверки статуса контроля необходимо открыть нужный профиль в графе Profile name (Имя профиля) нажатием на «+». Появится список категорий, принадлежащих определенному профилю. Напротив названия профиля в графе Send (Отправлено) должен отражаться статус False/True (не отправлено/отправлено). Название категории следует смотреть в поле Category (Категория). При нажатии двойным щелчком левой кнопки мыши на наименование категории откроется диалоговое окно.

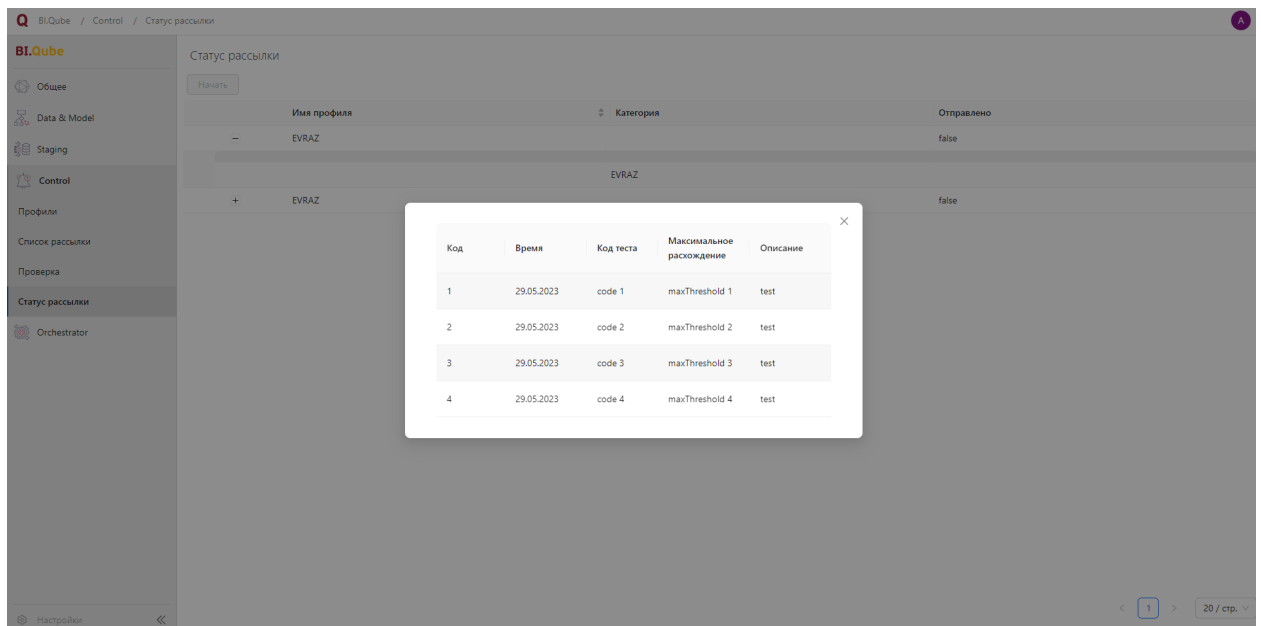


Рисунок 68. Диалоговое окно на странице Mailing status (Статус рассылки)

В данном окне представлены поля: Код (Code), Время (Time), Код теста (Test code), Максимальное расхождение (Max threshold) и Описание (Description).