

1. ВВЕДЕНИЕ	2
2. ОРГАНИЗАЦИЯ РАБОТЫ С BI.QUBE 2.0	3
3. МЕТАCOMMON	7
3.1 ПРОФИЛИ	8
3.2 ДАННЫЕ	11
3.3 ПОДКЛЮЧЕНИЯ	12
3.3.1 Файловые сервисы	15
3.3.1.1 SMB	16
3.3.1.2 S3 (Simple Storage Service)	19
3.3.2 СУБД	22
3.3.2.1 Microsoft SQL Server	23
3.3.2.2 MySQL	26
3.3.2.3 Oracle	28
3.3.2.4 PostgreSQL	30
3.3.2.5 SQL Server	32
3.3.2.6 SAP Hana	34
3.3.3 Веб-сервисы	35
3.3.3.1 Apache Kafka	36
3.3.3.2 RestAPI	38
3.3.4 1С Предприятие	41
3.3.4.1 1С на базе Microsoft SQL Server	42
3.3.4.2 1С на базе PostgreSQL	44
3.4 ПАРАМЕТРЫ	46
4. МЕТАTAGING	48
4.1 ПРОСМОТР ПРОФИЛЕЙ	49
4.2 СОЗДАНИЕ И РЕДАКТИРОВАНИЕ КОМАНД ДЛЯ ЗАГРУЗКИ ДАННЫХ	50
4.2.1 Запрос извлечения файлов с компьютера пользователя	53
4.2.2 Запрос извлечения данных из 1С Предприятие	55
4.2.3 Запрос извлечения данных из СУБД	57
4.2.4 Запрос извлечения данных из веб-сервисов REST API	58
4.2.5 Запрос извлечения данных из файловых хранилищ S3 и SMB	60
4.3 ТИПЫ ЗАГРУЗКИ	61
4.3.1 Полная загрузка	62
4.3.2 Полная загрузка с сохранением истории	63
4.3.3 Инкрементальная загрузка	64
4.3.4 Секции	65
4.4 ЗАПУСК НА ВЫПОЛНЕНИЕ	68
4.5 СЕССИИ	69
4.6 ДАННЫЕ МЕТАTAGING	71
5. DATA & MODEL	72
5.1 ПРОФИЛЬ DATA & MODEL	73
5.2 ДОМЕН	75
5.3 СОЗДАНИЕ МОДЕЛИ	76
5.3.1 Создание сущности в модели	77
5.3.1.1 Создание сущности без данных	79
5.3.1.2 Создание сущности на основе данных БД	80
5.3.2 Просмотр и редактирование данных	82
5.3.3 Создание связей между сущностями	88
5.3.4 Сборка сущности	90
5.3.5 Работа с моделью в графическом режиме	91
6. МЕТАCONTROL	95
6.1 ПРОФИЛЬ METACONTROL	96
6.2 СПИСОК РАССЫЛКИ	98
6.3 ПРОВЕРКА	99
6.4 СТАТУС РАССЫЛКИ	100

# ВВЕДЕНИЕ

**BI.Qube 2.0** (далее **BI.Qube**) – платформа (фреймворк, набор инструментов) предназначена для комплексного анализа данных и метаданных начиная от извлечения их из источников данных (учетных систем, веб-сервисов, баз данных и так далее) до построения масштабируемой модели данных для хранения и использования в BI аналитики, с возможностью обогащения не системными данными, осуществления контроля за качеством данных, с организацией представления ролевого доступа к реализованной модели данных.

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 по code/ low code.

**BI.Qube** включает в себя ряд компонентов, позволяющих полноценно решать определенный круг задач, появляющихся при построении КХД, не зависимо друг от друга, с другой стороны, каждый компонент предоставляет полноценный интерфейс доступа к данным о своей деятельности, что компонентов. 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 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 интеграции 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 по code/ low code и предназначенный для выполнения всех необходимых настроек, которые в последующем используют все остальные компоненты;
- **MetaStaging** – компонент имеющий развитый визуальный интерфейс, реализующий работу в режиме по 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 по code/ low code, предназначенный для создания аналитической модели данных, работает со справочниками, обогащения данных. Компонент работает с данными, доставляемыми с использованием компонента MetaStaging. Включает в себя компоненты MetaVault и MetaMasterData;
  - **MetaVault** – компонент имеющий развитый визуальный интерфейс, реализующий работу в режиме по 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 по 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 по 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 2.0

## ОБЩИЕ СВЕДЕНИЯ

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
- **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
- **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
- **Control:**
  - Profiles (Профили) – страница предназначена для просмотра группировки команд в исполняемый объект
  - Mailing list (Список рассылки) – страница предназначена для создания списков рассылки
  - Validation (Проверка) – страница предназначена для создания настройки бизнес-правил
  - Mailing status (Статус рассылки) – страница предназначена для просмотра результатов рассылок.

## ОПИСАНИЕ ВЕБ-ИНТЕРФЕЙСА

Все страницы системы BI.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. Макет типовой страницы).

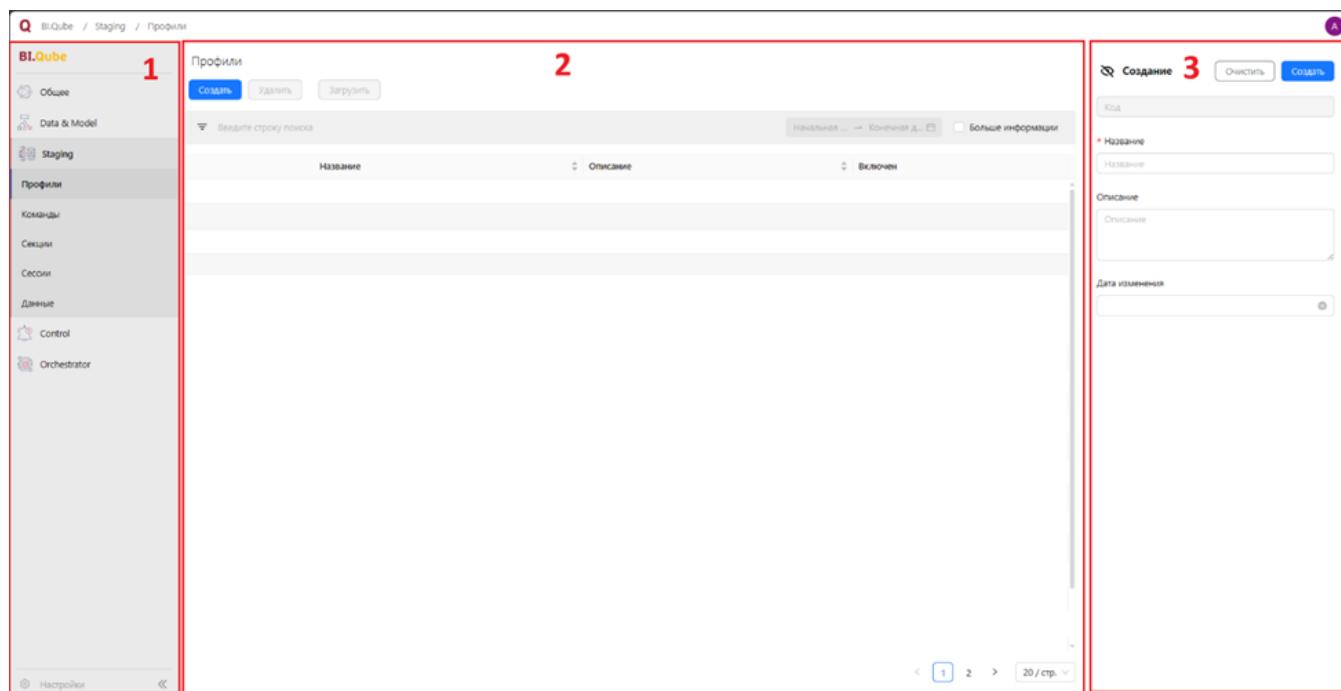


Рисунок. Макет типовой страницы

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

В процессе работы с системой могут возникать непредвиденные ошибки. Действия, которые не могут быть обработаны системой, генерируют ошибку. Текст ошибки отображается во всплывающем окне и дополнительно фиксируется в центре уведомлений. Центр уведомлений доступен на любой странице. Вызов

осуществляется с помощью нажатия на иконку "звонка" (  ) в правом верхнем углу (Рисунок. Центр уведомлений).

Кол-во непрочитанных уведомления указывается рядом с иконкой "звонка" цифрами. Пометить всё как прочитанное можно нажатием на иконку с двойной

галочкой (  ), очистить все уведомления - нажатием на иконку "кисти" (  ). При наведении курсора на иконку всплывает подсказка о её назначении. Для просмотра списка уведомлений необходимо воспользоваться колёсиком мыши или нажатием на серую полоску - бегунок справа окна свойств. Сортировка осуществляется с указанием даты и времени появления уведомления (Рисунок. Центр уведомлений).

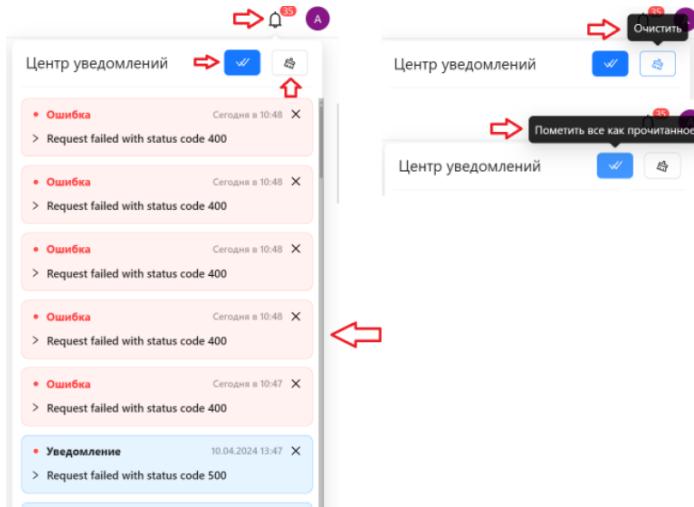


Рисунок. Центр уведомлений

Для лучшего визуального представления выделенные объекты таблицы подсвечиваются цветом и шрифт меняет свой стиль на жирный (Рисунок. Выделение таблиц - визуальное отображение).

Рисунок. Выделение таблиц (сущностей) - визуальное отображение

Для удобства в интерфейсе есть функция для сворачивания бокового меню слева и окна свойств справа, реализованная с помощью двух кнопок указанных стрелками на рисунке ниже. Для возврата в исходное состояние необходимо снова нажать на указанные кнопки.

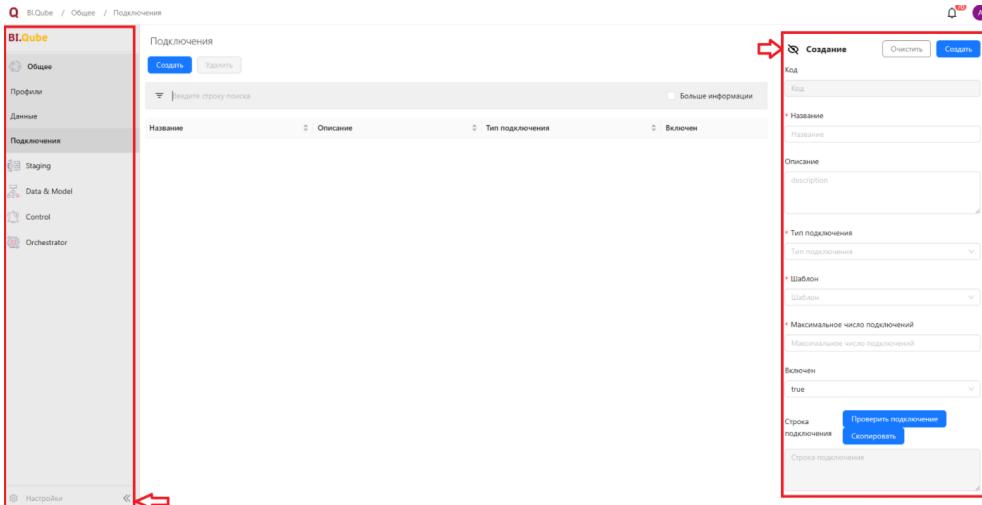


Рисунок. Отмеченные красным области можно свернуть

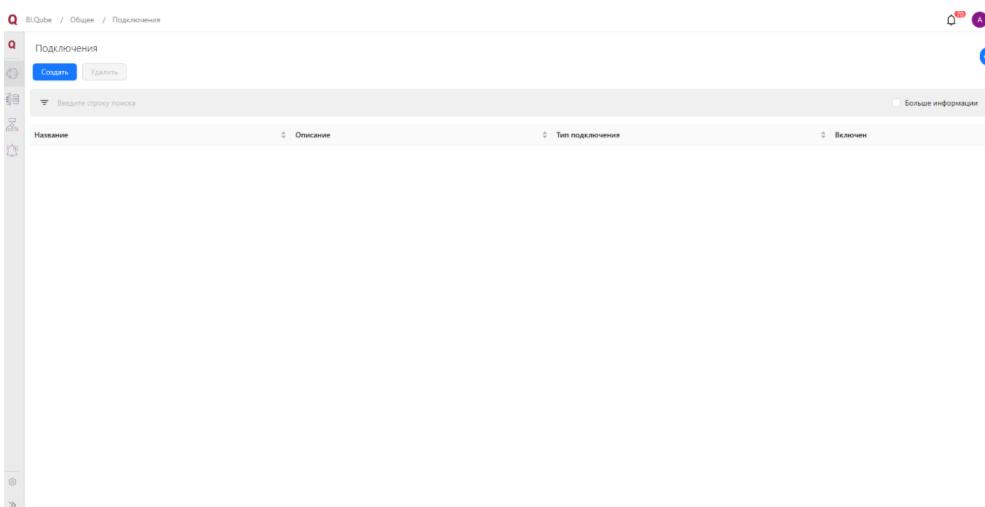


Рисунок. Отображение интерфейса при свёрнутых боковом меню и окне свойств

В разделе Data&Model - Models (Модель) при открытии выбранной модели данных, можно очистить все данные выбранной сущности (таблицы) с помощью

**Очистить**

. При нажатии на кнопку Clear (Очистить) появляется диалоговое окно, в котором предлагается подтвердить удаление данных в выбранной сущности (Рисунок. Диалоговое окно очистки выбранной сущности).

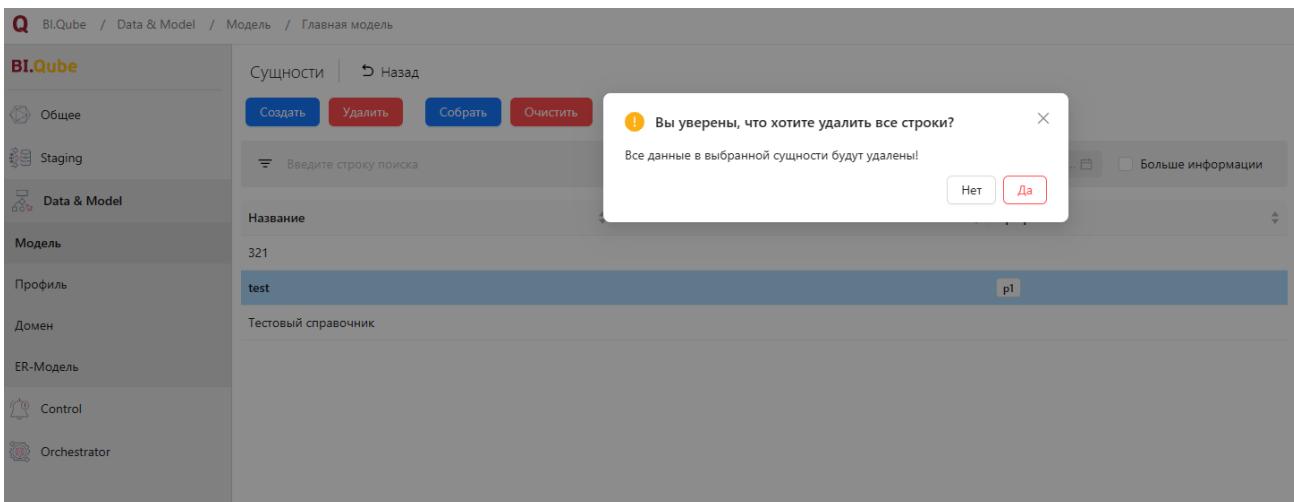


Рисунок. Диалоговое окно очистки выбранной сущности (таблицы)

\*При поиске в поле Name (Имя)

Во вкладках: General → Parameters (Общие → Параметры) и Staging → Commands (Staging → Команды) для удобства пользователя создана кнопка Copy (Скопировать). Данная кнопка создаёт копию выделенной строки.

The screenshot shows the BI.Qube application interface. The left sidebar has sections like 'Общее' (General), 'Staging', 'Команды' (Commands), 'Профили' (Profiles), 'Секции' (Sections), 'Сессии' (Sessions), and 'Данные' (Data). The 'Staging' section is selected. The main area is titled 'Команды' (Commands) and contains a table with the following data:

Имя	Описание	Запрос	Профили	Источник	Целевая система
Факты	Загрузка таблиц вебинара	SELECT * FROM BIQube_Template.src.Факты	Вебинар	SQLServer	PostgreWebinar
тест условий		SELECT * FROM "1c"."1c_test" where ОтработаноДней= ... <a href="#">подробнее</a>		DWH	DWH
Справочник ФизЛиц	Загрузка таблиц вебинара	SELECT * FROM src.СправочникФизЛиц	Вебинар	SQLServer	PostgreWebinar
Справочник СостоянияРемонта	Загрузка таблиц вебинара	SELECT * FROM src.СправочникСостоянияРемонта	Вебинар	SQLServer	PostgreWebinar

Рисунок. Кнопка Copy (Скопировать) во вкладке Staging → Commands (Staging → Команды)

# METACOMMON

## Авторизация

Для входа в систему пользователь должен быть зарегистрирован в сервисе Keycloak

**Keycloak** — это решение для управления идентификацией и доступом с открытым исходным кодом, предназначенное для использования в ИС где могут использоваться паттерны микросервисной архитектуры.

В инфраструктуре пользователя сервис должен быть развернут, должны быть занесены пользователи и для пользователей BI.Qube должна быть выполнены соответствующие настройки.

## Общие

Компонент предназначен для концентрации в одном месте всех общих настроек, параметров системы, которые в своей работе используют все остальные компоненты. В данном разделе доступны следующие страницы:

- Домены - на данной странице создаются и настраиваются имена будущих подмножеств объектов системы, к которым будет предоставлен ролевой доступ. Например, доступ к подели данных, доступ к командам и так далее
- Пользователи - на данной странице отображаются данные текущего пользователя, если пользователь имеет роль администратора, то ему доступен для просмотра полный перечень пользователей системы, сгруппированные по ролям, данные на эту страницу попадают из Keycloak.
- Роли - на данной странице перечисляются роли, созданные в Keycloak, которым доступна система BI.qube. Здесь осуществляется настройка доступа ролей к доменам, объектам доменов, а также разрешения внутри доступа (чтение, запись, удаление).
- Подключения - страница предназначена для создания подключений к источникам и получателям данных.
- Профили - страница предназначена для создания объектов типа "Профиль", в которых группируются команды компонентов BI.qube. Использование контейнеров позволяет группировать задачи и запускать их на выполнение в режиме параллельной обработки
- **Параметры** -
- **Данные** -

# ПРОФИЛИ

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 настройках каждого компонента. Удаление профиля доступно только на странице "Профили" в разделе "Общие".

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 из веб интерфейса.

Name	Author	Description	Schedule	Start date	Enabled	Catch up
p1	admin				<input checked="" type="checkbox"/>	<input type="checkbox"/>
ExcelTest	mstgng_owner	Тестирование различных файлов Excel	None	01.04.2024 12:28:22	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
APITest	mstgng_owner	Получать курс доллара	None	01.04.2024 11:52:56	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
PrimerModel_mstg	mstgng_owner	загрузка и создание модели из Excel	None	01.04.2024 15:18:21	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
PrimerModel	admin	загрузка Excel			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
КЛАДР	mstgng_owner	Загрузка данных КЛАДР	None	05.04.2024 14:22:10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SapHANA	mstgng_owner		None	14.04.2024 18:06:39	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sap_HANA	mstgng_owner		None	14.04.2024 18:06:41	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
KLADRprofile	admin	Профиль для работы с адресами			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
company_n_salesanalytics	mstgng_owner	Загрузка данных аналитики продаж для компании <N> ... <a href="#">подробнее</a>	None	15.04.2024 16:01:11	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
OracleTest	mstgng_owner	test	None	19.04.2024 16:08:08	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ЦБ	mstgng_owner	протестировать endpoint Rest API для загрузки данн ... <a href="#">подробнее</a>	None	22.04.2024 14:27:49	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
OzonTest	mstgng_owner	протестировать endpoint Rest API загрузки данных о ... <a href="#">подробнее</a>	None	23.04.2024 15:52:58	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рисунок. Страница Profiles (Профили)

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

- Name (Название) – уникальное имя профиля, позволяющее отличать один профиль от другого, как правило, даётся осмысленное имя, поясняющее назначение профиля, не должно содержать пробелов;
- Author (Автор) – заполняется автоматически.
- Description (Описание) – расширенное описание назначения профиля (необязательное поле, введено для удобства пользователей);
- Расписание (Schedule) - это планировщик в формате, приемлемом для Airflow (когда запускать профиль);
- Включен (Enabled) - включён профиль;
- Catch up - планировщик по умолчанию запускает запуск DAG для любого интервала данных, который не запускался с момента последнего интервала данных (или был очищен). Эта концепция и называется Catch up;
- Start date (Дата изменения) – автоматически создаваемое поле, содержит дату создания поля, при необходимости дата может быть отредактирована.



A

## Создание

[Очистить](#)[Создать](#)

Код

\* Название

Название

Автор

Автор

Описание

Описание

Расписание

Расписание

Дата начала работы

Дата начала работы



Включен

true



Catch up

Catch up



Рисунок. 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. Дополнительную информацию можно увидеть, нажав на кнопку More info (Больше информации), в строке фильтров над таблицей, в основной части экрана (Рисунок. Пример заполненной страницы Profiles (Профили) с включённым фильтром More info (больше информации)).

The screenshot shows the BIQube interface with the 'Profiles' tab selected. On the left, there's a sidebar with categories like 'Общее' (General), 'Профили' (Profiles), 'Параметры' (Parameters), 'Данные' (Data), and 'Подключения' (Connections). The main area displays a table of profiles with columns: Код (Code), Название (Name), Автор (Author), Описание (Description), Расписание (Schedule), Дата начала работы (Start Date), Включен (Enabled), Catch up (Catch up), Дата создания (Creation Date), Автор создания (Creator), Дата редактирования (Last Edit Date), and Автор редактирования (Last Editor). A search bar at the top says 'Введите строку поиска' (Enter search string). To the right of the table is a detailed view panel for a selected profile, showing fields for Название (Name), Автор (Author), Описание (Description), Расписание (Schedule), Дата начала работы (Start Date), Включен (Enabled), and Catch up (Catch up). Buttons for 'Создание' (Create), 'Очистить' (Clear), and 'Создать' (Create) are also visible.

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

После заполнения всех свойств необходимо нажать кнопку Create (Создать), и только что созданный профиль отобразится в таблице. Для удаления профиля необходимо нажать на кнопку Delete (Удалить) (Рисунок. Команды создания/удаления данных профиля).

## Профили

[Создать](#)

[Удалить](#)

Рисунок. Команды создания/удаления данных профиля

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. Команда обновления данных о профиле).



Рисунок. Команда обновления данных о профиле

# 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. Просмотр загруженных данных).

The screenshot shows the BIQube interface with the 'Data' tab selected in the sidebar. The main area displays a table of loaded data with columns: ID Номенклатура, Номенклатура, and Класс товара. The data includes rows for Джинсы, Юбка, Рубашка, and Брюки. To the right of the table is a query editor window titled 'СправочникНоменклатуры' containing the SQL query: 'SELECT \* FROM PrimerModel.СправочникНоменклатуры'. Below the query is a button labeled 'Запустить скрипт' (Run script). To the right of the editor is a sidebar titled 'Иерархия' (Hierarchy) with a dropdown menu set to 'DWH (postgresql)'. A red arrow points from the text 'В разработке. Здесь же есть возможность создавать хранимые процедуры и другие объекты базы данных необходимые для поддержки работы хранилища' to this dropdown menu. The sidebar also lists other databases like 'mdm\_dwh', 'auth\_test', and 'public', along with their schemas and tables.

Рисунок. Просмотр загруженных данных

*В разработке. Здесь же есть возможность создавать хранимые процедуры и другие объекты базы данных необходимые для поддержки работы хранилища*

# ПОДКЛЮЧЕНИЯ

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

При развертывании BI.Qube создается подключение «Локальные файлы» и «Хранилище данных». Первое подключение используется для поддержки алгоритма загрузки файлов с локальных компьютеров пользователей. Все файлы с компьютера пользователя всегда загружаются в промежуточное хранилище и хранятся там для возможности просмотра истории загрузок, после чего могут быть загружены непосредственно в хранилище. Для этого используется второе подключение, созданное при развертывании. Кроме этого, именно в этом подключении будет строится хранилище данных.

The screenshot shows the BI.Qube interface with the 'Endpoints' section selected in the sidebar. The main area displays a table of existing connections, including 'Локальные файлы' (Type: s3, Enabled: checked) and 'DWH' (Type: Хранилище данных, Connection: postgresql, Enabled: checked). A search bar and a 'Create' button are visible above the table. To the right, there's a detailed form for creating a new connection, with fields for 'Code' (Code), 'Name' (Name), 'Description' (Description), 'Endpoint Type' (Type), 'Template' (Template), 'Max connections' (Max connections), 'Enabled' (Enabled), and a 'Connection string' (Строка подключения) field containing a placeholder 'Скрипт подключения'. Buttons for 'TestConnection' (Проверить подключение) and 'Copy' (Скопировать) are also present. The bottom right corner shows a pagination indicator '1 / 20 / страниц'.

Рисунок. Страница создания/ редактирования подключений

Создание нового подключения осуществляется нажатием кнопки «Создать». Справой стороны экрана в окне свойств необходимо заполнить следующие поля (поля, указанные со звёздочкой, обязательны для заполнения.) (Рисунок. Страница создания/ редактирования подключений):

- Code (Код) – уникальный идентификатор записи в базе данных, заполняется автоматически;
- Name (Название) – имя подключения, вводится без пробелов;
- Description (Описание) – бизнес описание подключения;
- Endpoint Type (Тип подключения) – источник данных СУБД, веб-сервис или другая система, к которой настроен коннектор в системе BI.Qube. BI.Qube содержит большой перечень коннекторов к различным источникам и типов источников;
- Template (Шаблон) – шаблон строки подключения, для выбранного типа 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 (Рисунок. Поля для настройки подключений).

Код

\* Название

Описание

\* Тип подключения

\* Шаблон

\* Максимальное число подключений

Включен

Строка подключения

Проверить подключение
Скопировать

Строка подключения

Рисунок. Поля для настройки подключений

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.

Название	Описание
test	decr

Рисунок. Удаление подключения

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

Каждому типу подключения (endpoints) соответствует один или более шаблон настроек. В зависимости от доступных пользователю данных для авторизации на стороне подключения (endpoints) выбирается подходящий шаблон (Рисунок. Группировка по типам Endpoint (Подключений) в поле Endpoint Type (Тип подключения)).

В окне свойств справа в поле Endpoint Type (Тип подключения) для удобства осуществлена группировка по типам Endpoint (Подключений).

\* Тип подключения

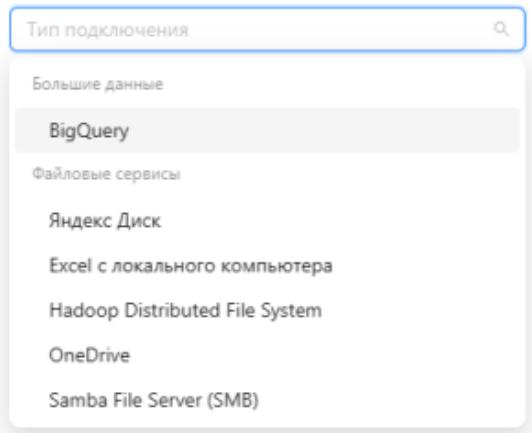


Рисунок. Группировка по типам Endpoint (Подключений) в поле Endpoint Type (Тип подключения)

# Файловые сервисы

**Файловый сервер (ФС)** — это выделенный компьютер или устройство в сети, которое предоставляет централизованное хранилище и файловые службы другим устройствам в такой сети. Основное назначение файлового сервера — хранение и защита информации, авторизация доступа и совместное использование файлов между несколькими клиентами по сети. Наличие ФС устраняет необходимость в отдельном локальном хранилище файлов на каждом компьютере.

Простыми словами, файловый сервер — это один или несколько физических компьютеров, ресурсы которых выделены для хранения файлов.

# SMB

**SMB** (сокр. от англ. Server Message Block) — сетевой протокол прикладного уровня для удалённого доступа к файлам, принтерам и другим сетевым ресурсам, а также для межпроцессного взаимодействия (для Windows). А SMB (сокр. от англ. Samba File Server) — это набор софта для Linux. Эти два понятия синонимы, различия только в том, на какой операционной системе они работают.

Для типа подключения **SMB** доступен один вариант строки подключения (шаблонов). В интерфейсе выбор нужного шаблона осуществляется в поле Template (Шаблон) (Рисунок. Вариант шаблона для типа подключения SMB.).

\* Тип подключения

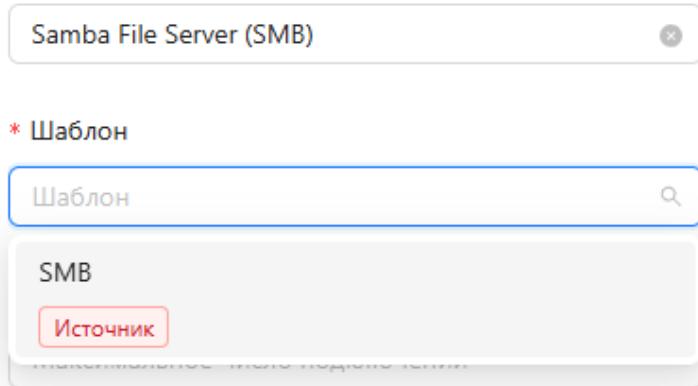


Рисунок. Вариант шаблона для типа подключения SMB.

99+ PA

**Редактор** Сбросить Обновить

Описание

description

Тип подключения: Samba File Server (SMB)

Шаблон: SMB

Максимальное число подключений: 100

Включен: true

Адрес: msk-ssbi.com

Общая папка: Evraz

Относительный путь: SED CSV

Домен: msk

Логин: MSK\_9365\_cdszcsd

Пароль: \*\*\*\*

Тип транспорта для SMB: DirectTCPTTransport

Строка подключения: Host=msk-ssbi.com; Folder=Evraz;

Проверить подключение Скопировать

Рисунок. Строки заполнения шаблона

При выборе данного шаблона в окне свойств появляются следующие строки для заполнения (Рисунок. Строки заполнения шаблона):

- Endpoint Type (Тип подключения) – SMB;
- Template (Шаблон) – smb (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Host (Адрес) – указывается сетевой адрес, где размещен endpoint;
- Relative path (Относительный путь) – путь относительно общей папки. Если нужен корень - не указывать ничего. В конце добавлять слэш;
- Folder (Общая папка) - папка с общим доступом (та, к которой предоставлен общий доступ, указывать без слешей);
- Domain (Домен) - нужно указывать, если в имени пользователя, которому доступны данные в endpoint, указан домен. В других случаях это поле заполнять не нужно;
- Login (Логин) - указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) – пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- SMB transport type (Тип транспорта для SMB) - можно выбрать из выпадающего списка.
  - NetBIOS over TCP/IP (NetBT) – это адаптация протокола NetBIOS для работы в сетях TCP/IP. Эта адаптация позволяет использовать услуги NetBIOS в сетях IP, расширяя функциональность NetBIOS за пределы локальных сетевых границ.
  - (Рекомендуется для современных систем) - Direct TCP Transport - обладает более совершенными механизмами безопасности, такие как сквозное шифрование и алгоритм Advanced Encryption Standard (AES) (Рисунок. Тип транспорта для SMB).

\* Тип транспорта для SMB

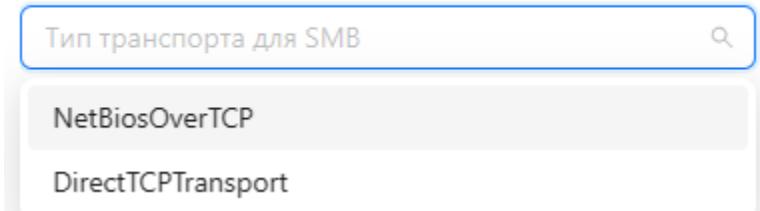


Рисунок. Тип транспорта для SMB

- Connection string (Строка подключения) - здесь создаётся строка подключения из заполненных полей выше: Host=/{Host}\*/; Folder=/{Folder}\*/; Domain=/{Domain}\*/; Login=postgres; Password=\*\*\*; SmbTransportType=/{SmbTransportType}\*;/.

Поля со звёздочкой обязательны для заполнения.

## S3 (Simple Storage Service)

**S3 (Simple Storage Service)** - сервис для хранения цифровых данных большого объёма. Работает по одноимённому протоколу.

Это вариант «плоского» (не иерархического) хранилища. С точки зрения системы все объекты равнозначны, поэтому в S3-хранилище удобно долго хранить разнородную информацию и быстро получать к ней доступ.

Для данного типа источника доступные шаблоны подключения приведены в одноименном поле. После выбора подходящего шаблона необходимо заполнить появившиеся поля (Рисунок. Endpoint Type (Тип подключения) - S3):

## Создание

Очистить

Создать

\* Название

Описание

\* Тип подключения

\* Шаблон

\* Максимальное число подключений

Включен

\* Ключ доступа

\* Секретный ключ

\* Имя бакета данных

\* Адрес сервера

Регион

Строка  
подключения

Проверить подключение

Скопировать

```
Key=/{Key}*/;
Secret=/{Secret}*/;
BucketName=/{BucketName}*/;
```

Рисунок. Endpoint Type (Тип подключения) - S3

- 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.

После создания подключения можно переходить к следующему шагу, например просмотру данных доступных в источнике или перейти к другим компонентам системы.

Поля со звёздочкой обязательны для заполнения.

# СУБД

**СУБД** (система управления базами данных) — это комплекс программ, позволяющих создать базу данных и манипулировать данными (вставлять, обновлять, удалять и выбирать).

Система обеспечивает быстродействие, безопасность данных, простоту получения и обновления данных, многопользовательский доступ и способность хранить большое количество данных, а также предоставляет средства для администрирования БД (базой данных).

Бывают реляционные, нереляционные, графовые, документоориентированные, колоночные (столбцовые), базы данных key-value, сетевые и иерархические.

Примеры СУБД: **Oracle, MySQL, Microsoft SQL Server, PostgreSQL**.

Основные функции СУБД:

- управление данными во внешней памяти (на дисках);
- управление данными в оперативной памяти с использованием дискового кэша;
- журнализация изменений (сохранение истории), резервное копирование и восстановление базы данных после сбоев;
- поддержка языков БД (язык определения данных, язык манипулирования данными).

Каждая СУБД основывается на какой-либо модели данных, это является одним из признаков классификации.

# Microsoft SQL Server

**Microsoft SQL Server (MSSQL)** - это система управления реляционными базами данных (СУБД), используемая для хранения и извлечения данных из других программных приложений.

Для данного типа источника доступные шаблоны подключения приведены в одноименном поле (Рисунок. Варианты шаблонов). После выбора подходящего шаблона необходимо заполнить появившиеся поля (Рисунок. Endpoint Type (Тип подключения) - Microsoft SQL Server (MSSQL)):

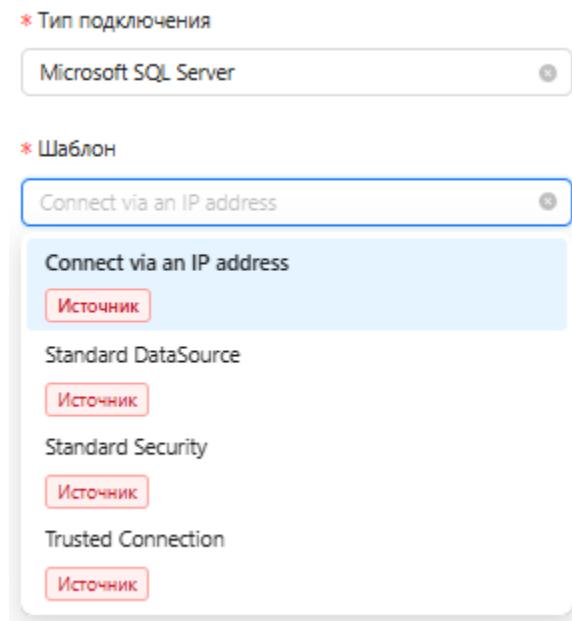


Рисунок. Варианты шаблонов

## Описание

description

## \* Тип подключения

Microsoft SQL Server

## \* Шаблон

Connect via an IP address

## \* Максимальное число подключений

Максимальное число подключений

## Включен

true

## \* Источник данных

Источник данных

## \* База данных

База данных

## \* Пользователь

Пользователь

## \* Пароль

Пароль

## Кодировка

Кодировка

## \* Сетевой протокол ①

Сетевой протокол

Строка  
подключения

Проверить подключение

Скопировать

Data Source=/{Data Source}\*/;

Рисунок. Endpoint Type (Тип подключения) - Microsoft SQL Server (MSSQL)

- Endpoint Type (Тип подключения) – Microsoft SQL Server (MSSQL);

- Template (Шаблон);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Data Source (источник данных);
- Initial Catalog (база данных);
- User (Пользователь) – указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) – пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- TrustServerCertificate (кодировка);
- Network Protocol (сетевой протокол);
- Connection string (Строка подключения) - здесь создаётся строка подключения из заполненных полей выше.

Поля со звёздочкой обязательны для заполнения.

# MySQL

MySQL - это реляционная система управления базами данных (СУБД) с открытым исходным кодом, позволяющая хранить, организовывать большие объёмы данных, и манипулировать ими. Использует стандартный язык SQL для обработки данных (Рисунок. Endpoint Type (Тип подключения) - MySQL).

Для типа подключения MySQL доступны несколько вариантов строк подключения (шаблонов). В интерфейсе выбор нужного шаблона осуществляется в поле Template (Шаблон) (Рисунок. Варианты шаблонов для подключения mysql).

Создание      Очистить      Создать

Код

Код

\* Название

Название

Описание

description

\* Тип подключения

MySQL

\* Шаблон

Шаблон

\* Максимальное число подключений

Максимальное число подключений

Включен

true

Строка подключения

Проверить подключение

Скопировать

Строка подключения

Рисунок. Endpoint Type (Тип подключения) - MySQL

## \* Шаблон

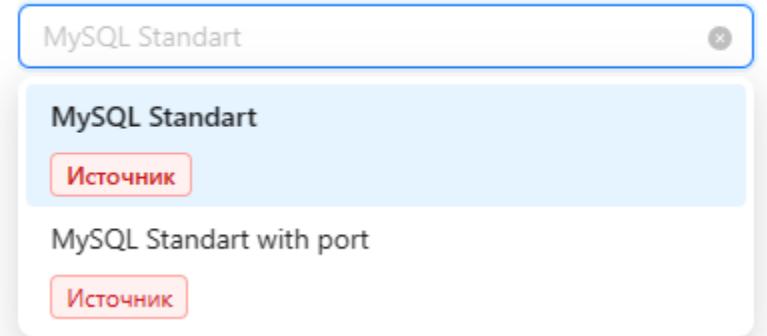


Рисунок. Варианты шаблонов для подключения mysql.

Примеры шаблонов:

### *MySQL Standart*

- Endpoint Type (Тип подключения) – mysql;
- Template (Шаблон) – MySQL Standart (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Server (Адрес сервера) – указывается сетевой адрес, где размещен endpoint;
- Database (База данных) – указывается имя базы данных;
- Uid (Пользователь) – указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) – пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- Connection string (Строка подключения) - здесь создаётся строка подключения из заполненных полей выше: Server=localhost; Database=/\*{Database}\*/; Uid=postgres; Pwd=\*\*\*;.

### *MySQL Standart with port*

- Endpoint Type (Тип подключения) – mysql;
- Template (Шаблон) – MySQL Standart with port (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Server (Адрес сервера) – указывается сетевой адрес, где размещен endpoint;
- Port (Порт);
- 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);
- Connection string (Строка подключения) - здесь создаётся строка подключения из заполненных полей выше: Server=localhost; Database=/\*{Database}\*/; Uid=postgres; Pwd=\*\*\*;.

Поля со звёздочкой обязательны для заполнения.

# Oracle

Oracle Database — это объектно-реляционная система управления базами данных (СУБД) от компании Oracle. Она используется для создания структуры новой базы, её наполнения, редактирования содержимого и отображения информации. Подходит для работы с высоконагруженными проектами, которые обрабатывают запросы миллионов пользователей (Рисунок. Endpoint Type (Тип подключения) - Oracle).

Создание

Очистить

Создать

Код

Название

Описание

description

\* Тип подключения

Oracle

\* Шаблон

Шаблон

\* Максимальное число подключений

Максимальное число подключений

Включен

true

Строка подключения

Проверить подключение

Скопировать

Строка подключения

Рисунок. Endpoint Type (Тип подключения) - Oracle

Для типа подключения **Oracle** доступны несколько вариантов строк подключения (шаблонов). В интерфейсе выбор нужного шаблона осуществляется в поле `Template` (Шаблон) (Рисунок. Виды шаблонов для oracle).

## \* Шаблон

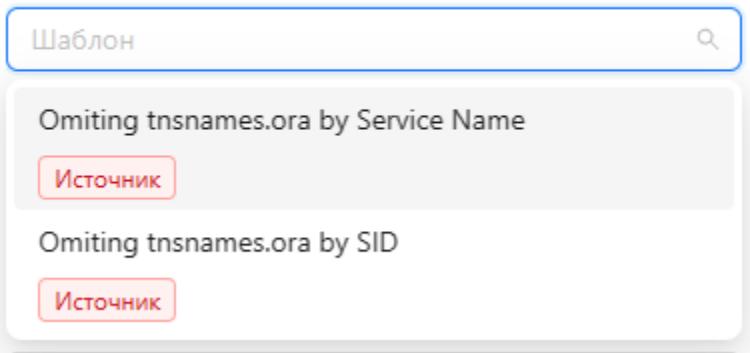


Рисунок. Виды шаблонов для oracle

Примеры шаблонов:

### Omiting tnsnames.ora by Service Name

- Endpoint Type (Тип подключения) – oracle;
- Template (Шаблон) – Omiting tnsnames.ora by Service Name (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Host (Адрес сервера) – указывается сетевой адрес, где размещен endpoint;
- Port (Порт) – указывается порт, по которому доступен endpoint;
- Service Name (Имя целевой службы);
- User id (Пользователь) – указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) – пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- Connection Protocol (Протокол подключения);
- Connection string (Строка подключения) - здесь создается строка подключения из заполненных полей выше: Host=localhost; Port=1521; SERVICE\_NAME=/{SERVICE\_NAME}\*/; User Id=postgres; Password=\*\*\*; PROTOCOL=TCP..

### Omiting tnsnames.ora by SID

- Endpoint Type (Тип подключения) – oracle;
- Template (Шаблон) – Omiting tnsnames.ora by SID (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Host (Адрес сервера) – указывается сетевой адрес, где размещен endpoint;
- Port (Порт) – указывается порт, по которому доступен endpoint;
- Service Name (Имя целевой службы);
- User id (Пользователь) – указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) – пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- Connection Protocol (Протокол подключения);
- Connection string (Строка подключения) - здесь создается строка подключения из заполненных полей выше: Host=localhost; Port=1521; SERVICE\_NAME=/{SERVICE\_NAME}\*/; User Id=postgres; Password=\*\*\*; PROTOCOL=TCP..

Поля со звездочкой обязательны для заполнения.

# PostgreSQL

Postgre — это свободно распространяемая объектно-реляционная система управления базами данных (СУБД) с открытым исходным кодом, написанном на языке С.

Для типа подключения PostgreSQL доступны несколько вариантов строк подключения (шаблонов). В интерфейсе выбор нужного шаблона осуществляется в поле Template (Шаблон).

## \* Шаблон

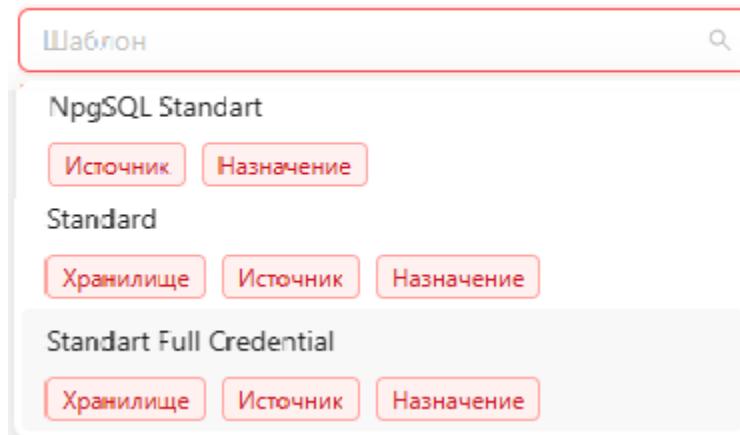


Рисунок. Варианты шаблонов для подключения PostgreSQL.

Примеры шаблонов:

### Npgsql Standart

- Endpoint Type (Тип подключения) – PostgreSQL;
- Template (Шаблон) – Npgsql Standart (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- 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;
- Connection string (Строка подключения) - здесь создаётся строка подключения из заполненных полей выше: Server=192.168.128.1; Port=5432; Database=postgres; User Id=postgres; Password=\*\*\*; Include Error Detail=false.

### Standart

- Endpoint Type (Тип подключения) – PostgreSQL;
- Template (Шаблон) – Standart (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- 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;
- Connection string (Строка подключения) - здесь создаётся строка подключения из заполненных полей выше: Server=192.168.128.1; Port=5432; Database=postgres; User Id=postgres; Password=\*\*\*; Include Error Detail=false.

### Standart Full Credential

- Endpoint Type (Тип подключения) – PostgreSQL;
- Template (Шаблон) – Standart Full Credential (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- 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;
- Connection string (Строка подключения) - здесь создаётся строка подключения из заполненных полей выше: Server=192.168.128.1; Port=5432; Database=postgres; User Id=postgres; Password=\*\*\*; Include Error Detail=false.

Поля со звёздочкой обязательны для заполнения.

# SQL Server

**Microsoft SQL Server** (MSSQL) – это система управления реляционными базами данных (СУБД), используемая для хранения и извлечения данных из других программных приложений. Microsoft разработала это программное обеспечение для управления информацией на нескольких компьютерах в одной сети. Используя язык программирования SQL (Structured Query Language – «язык структурированных запросов»), SQL Server может выполнять аналитику и обработку транзакций, а также работу с информацией.

Для типа подключения **SQL Server** доступны несколько вариантов строк подключения (шаблонов). В интерфейсе выбор нужного шаблона осуществляется в поле **Template (Шаблон)**.

## \* Шаблон

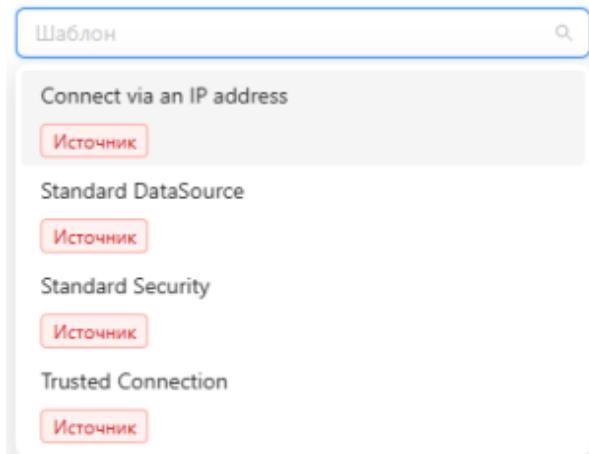


Рисунок. Тип подключения SQL Server и шаблоны

Примеры шаблонов:

### Connect via an IP address

- Endpoint Type (Тип подключения) – sqlserver;
- Template (Шаблон) – Connect via an IP address (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Data Source (Источник данных) - откуда берутся загружаемые данные;
- Initial Catalog (База данных) – указывается база данных;
- User id (Пользователь) – указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) – пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- TrustServerCertificate (Кодировка);
- Network Protocol (Сетевой протокол) выбирается из выпадающего списка;
- Connection string (Строка подключения) – здесь создаётся строка подключения из заполненных полей выше. Пример: User id=sqlserver; Password=\*\*\*; TrustServerCertificate=/{TrustServerCertificate}\*/; Data Source=192,168,128,1; Initial Catalog=sqlserver;

## \* Сетевой протокол ?

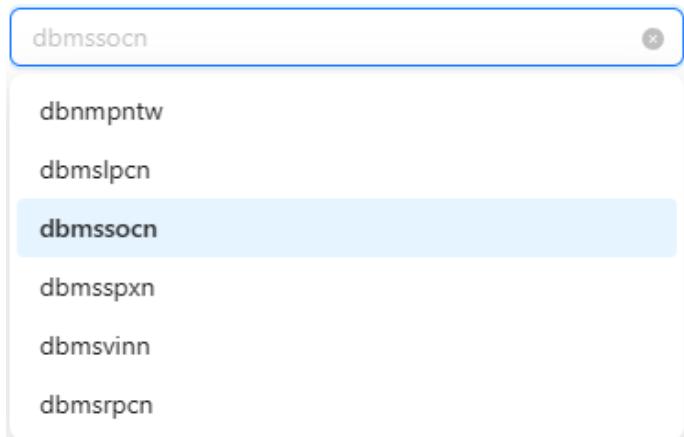


Рисунок. Выдающий список поля Network Protocol (Сетевой протокол)

*Standart DataSource*

- Endpoint Type (Тип подключения) – sqlserver;
- Template (Шаблон) – Standart DataSource (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Data Source (Источник данных) - откуда берутся загружаемые данные;
- Initial Catalog (База данных) – указывается база данных;
- User id (Пользователь) – указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) – пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- TrustServerCertificate (Доверять сертификату сервера);
- Connection string (Строка подключения) - здесь создаётся строка подключения из заполненных полей выше. Пример: User id=sqlserver; Password=\*\*\*; TrustServerCertificate=/\*{TrustServerCertificate}\*/; Server=localhost; Database=/\*{Database}\*/;.

#### *Standart Security*

- Endpoint Type (Тип подключения) – sqlserver;
- Template (Шаблон) – Standart Security (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Server (Источник данных) - откуда берётся данные;
- Database (База данных) – указывается база данных;
- User id (Пользователь) – указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) – пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- Trust Server Certificate (Доверенный сертификат);
- Connection string (Строка подключения) - здесь создаётся строка подключения из заполненных полей выше. Пример: User id=sqlserver; Password=\*\*\*; TrustServerCertificate=/\*{TrustServerCertificate}\*/; Data Source=192,168,128,1; Initial Catalog=sqlserver;.

#### *Trusted Connection*

- Endpoint Type (Тип подключения) – sqlserver;
- Template (Шаблон) – Trusted Connection (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Server (Источник данных) - откуда берётся данные;
- Database (База данных) – указывается база данных;
- Connection string (Строка подключения) - здесь создаётся строка подключения из заполненных полей выше. Пример: User id=sqlserver; Password=\*\*\*; TrustServerCertificate=/\*{TrustServerCertificate}\*/; Server=localhost; Database=/\*{Database}\*/;.

Поля со звёздочкой обязательны для заполнения.

# SAP Hana

**SAP HANA** (High-performance ANalytic Appliance) — это многомодельная база данных, в которой данные хранятся в памяти, а не на диске.

Для типа подключения **Saphana** доступны несколько вариантов строк подключения (шаблонов). В интерфейсе выбор нужного шаблона осуществляется в поле **Template** (Шаблон).

## \* Шаблон

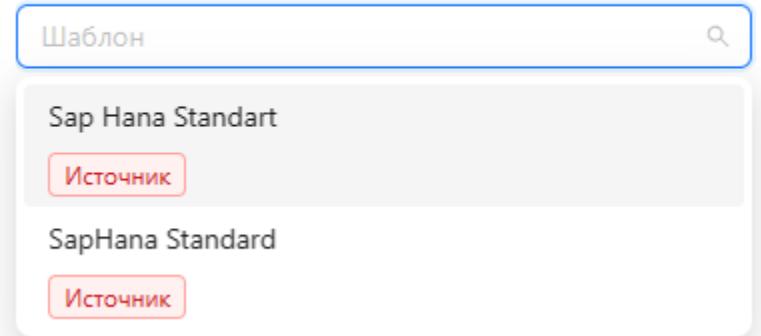


Рисунок. Варианты шаблонов для подключения saphana.

### Sap Hana Standart

- Endpoint Type (Тип подключения) – saphana;
- Template (Шаблон) – Sap Hana Standart (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Host (Адрес сервера) – указывается сетевой адрес, где размещен endpoint;
- User (Пользователь) – указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) – пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- Connection string (Строка подключения) - здесь создается строка подключения из заполненных полей выше: Host=hxehost:39015; User Id=SYSTEM; Password=\*\*\*;.

### SapHana Standard

- Endpoint Type (Тип подключения) – saphana;
- Template (Шаблон) – Sap Hana Standart (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Host (Адрес сервера) – указывается сетевой адрес, где размещен endpoint;
- User (Пользователь) – указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) – пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- Keep-Alive query (Keep-Alive запрос) - тип запроса, который делает проверку подключения;
- Connection string (Строка подключения) - здесь создается строка подключения из заполненных полей выше: Host=hxehost:39015; User Id=SYSTEM; Password=\*\*\*;.

Поля со звёздочкой обязательны для заполнения.

# Веб-сервисы

**Веб-сервисы (или веб-службы)** — это технология, позволяющая системам обмениваться данными друг с другом через сетевое подключение. Обычно веб-сервисы работают поверх протокола HTTP или протокола более высокого уровня. Веб-сервис — просто адрес, ссылка, обращение к которому позволяет получить данные или выполнить действие.

# Apache Kafka

Apache **Kafka** — это распределённая система, предназначенная для обработки потоков данных в режиме реального времени. Её можно сравнить с почтой — одни сервисы передают туда сообщения-письма, а другие — получают. Apache Kafka называют брокером сообщений, потому что она выступает в качестве посредника.

Для типа подключения **Kafka** доступны несколько вариантов строк подключения (шаблонов). В интерфейсе выбор нужного шаблона осуществляется в поле **Template (Шаблон)**.

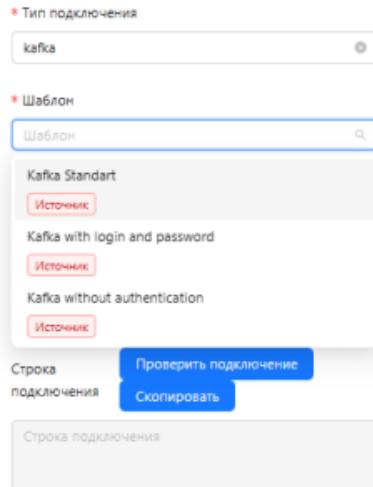


Рисунок. Варианты шаблонов для типа подключения kafka.

Примеры шаблонов:

## *Kafka Standart*

- Endpoint Type (Тип подключения) – kafka;
- Template (Шаблон) – Kafka Standart (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Bootstrap Servers (Адрес кластера);
- User (Имя пользователя) – указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) – пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- SslCaPem (Строка сертификата CA);
- Certificate File Path (Путь к файлу с SslCaPem);
- Content Type (Тип возвращаемых данных);
- Connection string (Строка подключения) – здесь создаётся строка подключения из заполненных полей выше. Пример: BootstrapServers=198.250.56.15; ContentType=json; User=postgres; Password=\*\*\*; SslCaPem=\*\*\*; CertificateFilePath=/\*{CertificateFilePath}\*/;

## *Kafka Standart with login and password*

- Endpoint Type (Тип подключения) – kafka;
- Template (Шаблон) – Kafka Standart (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Sasl mechanism (Механизм Sasl);
- Security protocol (Протокол безопасности);
- Bootstrap Servers (Адрес кластера);
- User (Имя пользователя) – указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) – пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- Content Type (Тип возвращаемых данных);
- Connection string (Строка подключения) – здесь создаётся строка подключения из заполненных полей выше. Пример: BootstrapServers=198.250.56.15; ContentType=json; User=postgres; Password=\*\*\*; SaslMechanism=Plain; SecurityProtocol=SaslPlaintext;.

## *Kafka Standart without authentication*

- Endpoint Type (Тип подключения) – kafka;
- Template (Шаблон) – Kafka Standart (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Bootstrap Servers (Адрес кластера);
- Sasl mechanism (Механизм Sasl);
- Security protocol (Протокол безопасности);
- Content Type (Тип возвращаемых данных);

- Connection string (Строка подключения) - здесь создаётся строка подключения из заполненных полей выше. Пример: BootstrapServers=198.250.56.15; ContentType=json; SaslMechanism=Plain; SecurityProtocol=SaslPlaintext;.

Веб-сервисы (или веб-службы) — это **технология, позволяющая системам обмениваться данными друг с другом через сетевое подключение**. Обычно веб-сервисы работают поверх протокола HTTP или протокола более высокого уровня. Веб-сервис — просто адрес, ссылка, обращение к которому позволяет получить данные или выполнить действие.

Поля со звёздочкой обязательны для заполнения.

# RestAPI

REST (Representational State Transfer) API — это архитектурный стиль для разработки веб-сервисов, основанный на стандартных HTTP-методах и ресурсоориентированном подходе. Формат данных в REST API может быть разнообразным, включая JSON, XML и другие. REST API широко применяется в веб-приложениях и мобильных приложениях для обеспечения межсистемного взаимодействия и интеграции с различными сервисами и платформами.

Для типа подключения **RestAPI** доступны несколько вариантов строк подключения (шаблонов). В интерфейсе выбор нужного шаблона осуществляется в поле **Template (Шаблон)**.

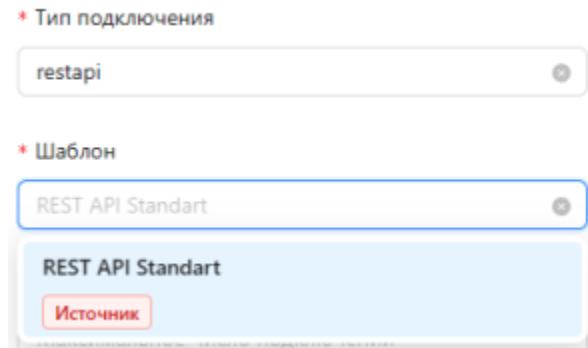


Рисунок. Пример шаблона REST API Standart для Endpoint Type (Тип подключения) - restapi.

Рассмотрим пример заполнения шаблона REST API Standart.

- Endpoint Type (Тип подключения) – restapi;
- Template (Шаблон) – REST API Standart (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- User (Пользователь) – указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) – пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- AuthType (Тип аутентификации) - выбирается из выпадающего списка;

Рисунок. Выпадающий список поля AuthType (Тип аутентификации)

- ContentType (Тип возвращаемого контента) - выбирается из выпадающего списка для файлов типа csv, xml, json;

## \* Тип возвращаемого контента

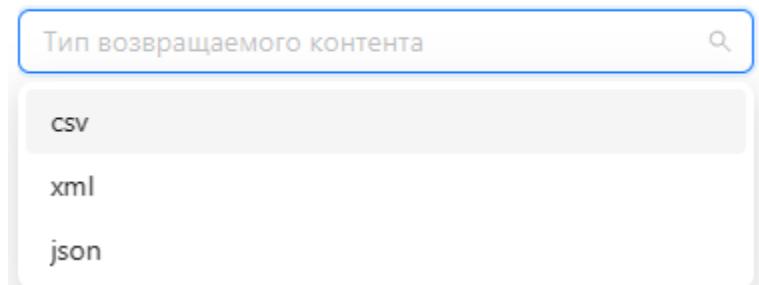


Рисунок. Выпадающий список поля ContentType (Тип возвращаемого контента)

- Encoding (Кодировка);
- AcceptEncoding (Тип декомпрессии);
- Start (Стартовое значение диапазона);
- Connection string (Строка подключения) - здесь создается строка подключения из заполненных полей выше: User=postgres; Password=\*\*\*; AuthType=1; ContentType=csv; Encoding=/\*{Encoding}\*/; AcceptEncoding=/\*{AcceptEncoding}\*/; Start=/\*{Start}\*/;

## Настройка команды загрузки данных по протоколу Rest API с источником json

После создания и заполнения Endpoint (подключения) на странице General (Общее), мы можем во вкладке Staging, Commands (Команды), в окне свойств справа в поле Source (Источник) выбрать созданный Endpoint (Подключение) Rest API и нажать на кнопку Create (Создать).

The screenshot shows the BI.Qube application interface. On the left, there's a sidebar with sections like 'Общее', 'Staging', and 'Команды'. The 'Команды' section is highlighted with a red box. In the main area, there's a table of commands with columns: Имя, Описание, Запрос, Профили, Источник, Целевая система, and Время. Two rows are visible: 'testCreate53\_ test' and 'testCreate53\_ test Copy\_1\_Copy'. To the right, a modal window titled 'Создание' is open. It has a search bar with 'RestApi (restapi)' selected, a dropdown for 'Целевая система' set to 'Локальные файлы', and a 'Запрос' input field containing 'SELECT \* FROM excel Моя файлы/Ном11111/с43б695e9\_подробнее'. A large red box highlights the 'Создать' button at the bottom right of the modal.

Рисунок. Выбор Endpoint (подключения) Rest API

В появившемся диалоговом окне в поле Enter file address (Введите адрес файла) ввести адрес файла и нажать на иконку ( ). В окне справа отобразится сформированный код файла.

На нажатии на кнопку From a query (Сформировать запрос) - генерируется предварительно простой запрос на выборку данных. Запрос по умолчанию формируется на SQL. Нажатие на кнопку Run Query (Выполнить запрос) соответственно выполняет сформированный запрос. Для дальнейшей работы с данными нажать кнопку OK.

This screenshot shows the BI.Qube interface with a 'Run Query' dialog open. The dialog has several numbered steps: 1. 'Введите адрес файла' (Enter file address) with the URL 'https://api.spacexdata.com/v3/capsules'; 2. 'OK' button; 3. 'SQL' tab; 4. 'Сформировать запрос' (Generate query) button; 5. 'Выполнить запрос' (Execute query) button. Below the dialog, there's a JSON preview panel showing a list of capsules with their details. A large red box highlights the JSON code in the preview panel, and a red arrow points from the 'OK' button in the dialog to the 'OK' button in the JSON preview panel.

Рисунок. Диалоговое окно загрузки данных из Rest API

- В случае, если запрос не выполнен, то система сообщит об ошибке. Запрос может быть не выполнен, если загружаемый JSON (файл) имеет сложную структуру. Для того чтобы это исправить в левом верхнем углу необходимо из выпадающего списка выбрать JOLT. В поле для ввода необходимо ввести код JOLT, который преобразует исходный JSON (файл) к более простому виду. Далее нажать кнопку Run Query (Выполнить запрос) - исходный файл преобразуется в соответствии с полученными инструкциями JOLT к новому виду и выводится во вкладке JSON (файл) в левом нижнем углу окна. Если всё успешно, то появятся вкладки с таблицами, которые будут созданы на основании нового созданного JSON (файла). В случае, если новый созданный JSON (файл) алгоритмы системы не смогли обработать, то вкладки не появятся, но преобразованный с помощью JOLT исходный JSON (файл) выведется на экран, чтобы можно было ознакомиться с результатом преобразования. Это позволит подкорректировать текст кода JOLT и выполнить процедуру загрузки и преобразования данных повторно.
- Jolt — это инструмент для выполнения задач, предназначенный для разработки программного обеспечения. Задачи определяются в скриптах Python. <https://jolt.readthedocs.io/en/latest/> - данный сайт может помочь в написании JOLT./

### Настройка команды загрузки данных по протоколу Rest API с источником xml/csv

Загрузка файлов по протоколу Rest API с источником xml/csv полностью повторяет уже описанный алгоритм действий для настройки команды загрузки данных по протоколу Rest API с источником json. За исключением того, что для источников xml/csv JOLT не используется.

Поля со звёздочкой обязательны для заполнения.

# **1С Предприятие**

**1С:Предприятие** - это полнотекстовая малокодовая платформа, предоставляющая готовую к использованию инфраструктуру и инструменты для быстрой разработки бизнес-приложений, таких как ERP, POS, WMS или другое индивидуальное корпоративное программное обеспечение.

Может быть развёрнута на СУБД: MS SQL Server и PostgreSQL.

# 1С на базе Microsoft SQL Server

**SQLServer1C** – серверное программное обеспечение, для работы с базами данных «1С» в клиент-серверном режиме (СУБД). СУБД обрабатывает запрос, который пришел от сервера «1С» и отправляет данные обратно на сервер «1С». Подключение SQL необходимо при работе в 1С в клиент-серверном режиме, это позволяет оптимизировать работу большого количества пользователей с большим объемом информации, за счет переноса ресурсоёмких операций на сервер.

Для типа подключения **SQLServer1C** доступны несколько вариантов строк подключения (шаблонов). В интерфейсе выбор нужного шаблона осуществляется в поле "Шаблон" (Template).

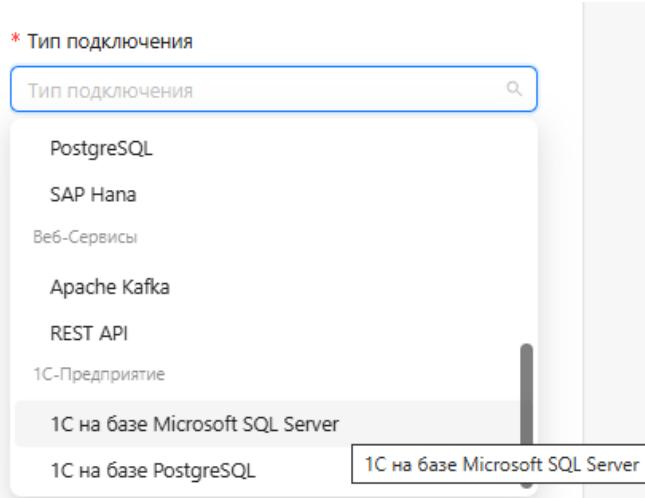


Рисунок. Тип подключения "1С на базе Microsoft SQL Server"

## \* Шаблон

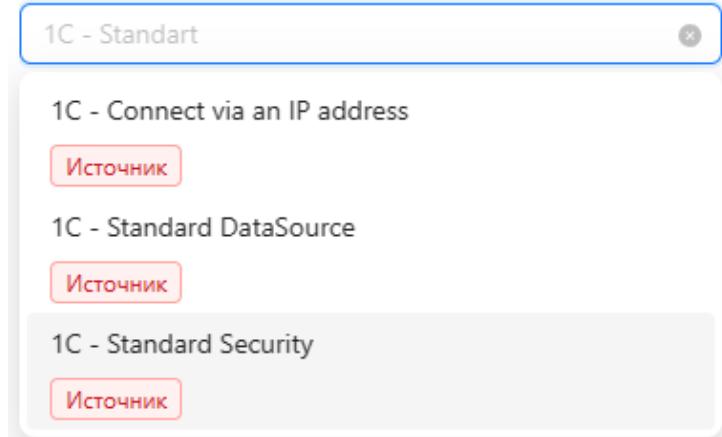


Рисунок. Варианты шаблонов

Примеры шаблонов:

### 1C - Connect via an IP address

- Endpoint Type (Тип подключения) – sqlserver1c;
- Template (Шаблон) – 1C - Connect via an IP address (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false – определяет доступность, создаваемого подключения в командах, если поле имеет состояние false то команду с этим endpointом создать будет нельзя;
- Type Work 1C Translator (Тип режима работы 1С) представлен двумя позициями: canvas/hard;
- Base Key 1C (Название БД 1С);
- Data Source (Сервер);
- Initial Catalog (База данных);
- User Id (Пользователь) - указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) - пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- Trust Server Certificate (Доверять сертификату сервера);
- Connection string (Строка подключения) - здесь создаётся строка подключения из заполненных полей выше: TypeWork1CTranslator=canvas; BaseKey1C=default; Data Source=localhost; Initial Catalog=/\*{Initial Catalog}\*/; User Id=root; Password=\*\*\*; TrustServerCertificate=True;

#### *1C - Standart DataSource*

- Endpoint Type (Тип подключения) – sqlserver1c;
- Template (Шаблон) – 1C - Standart DataSource (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Type Work 1C Translator (Тип режима работы 1С) представлен двумя позициями: canvas/hard;
- Base Key 1C (Название БД 1С);
- Data Source (Сервер);
- Initial Catalog (База данных);
- User Id (Пользователь) - указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) - пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- Trust Server Certificate (Доверять сертификату сервера);
- Connection string (Строка подключения) - здесь создаётся строка подключения из заполненных полей выше: TypeWork1CTranslator=canvas; BaseKey1C=default; Data Source=localhost; Initial Catalog=/\*{Initial Catalog}\*/; User Id=root; Password=\*\*\*; TrustServerCertificate=Yes;.

#### *1C - Standart Security*

- Endpoint Type (Тип подключения) – sqlserver1c;
- Template (Шаблон) – 1C - Standart Security (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Type Work 1C Translator (Тип режима работы 1С) представлен двумя позициями: canvas/hard;
- Base Key 1C (Название БД 1С);
- Data Source (Сервер);
- Initial Catalog (База данных);
- User Id (Пользователь) - указывается имя пользователя, которому доступны данные в endpoint;
- Password (Пароль) - пароль пользователя (сохраняется в базе данных системы и защищен от злоумышленников);
- Trust Server Certificate (Доверять сертификату сервера);
- Connection string (Строка подключения) - здесь создаётся строка подключения из заполненных полей выше: TypeWork1CTranslator=canvas; BaseKey1C=default; Data Source=localhost; Initial Catalog=/\*{Initial Catalog}\*/; User Id=root; Password=\*\*\*; TrustServerCertificate=Yes;.

# 1С на базе PostgreSQL

**PostgreSQL 1C** — одна из систем управления базами данных, которую поддерживает платформа в клиент-серверном варианте работы. Включает патчи с оптимизациями, выполненными разработчиками платформы 1С:Предприятия, которые учитывают особенности работы платформы 1С:Предприятие и типовых решений фирмы «1С». Используется «1С» в высоконагруженных коммерческих проектах, например, 1С:Fresh.

Для типа подключения **PostgreSQL 1C** доступны несколько вариантов строк подключения (шаблонов). В интерфейсе выбор нужного шаблона осуществляется в поле Template (Шаблон).

## \* Тип подключения

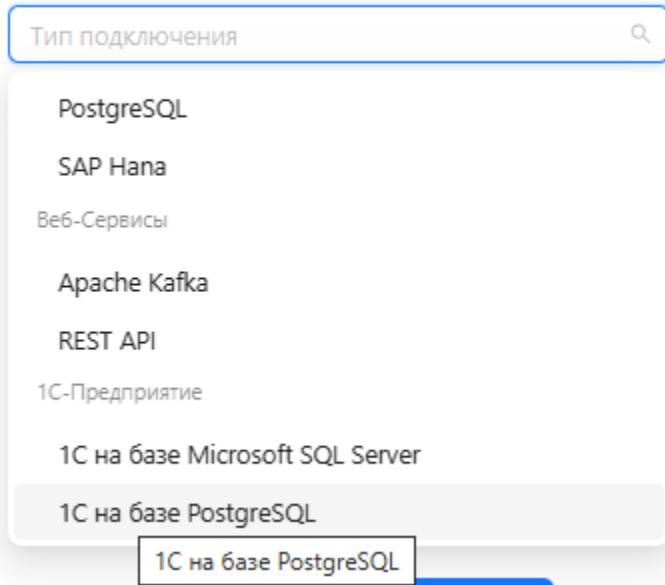


Рисунок. Выбор типа подключения

## \* Шаблон

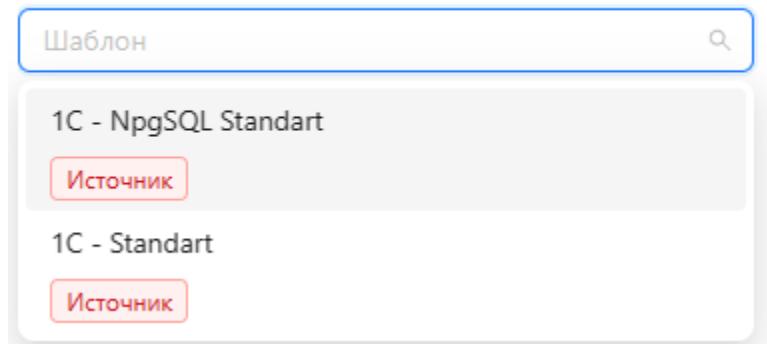


Рисунок. Варианты шаблонов для подключения PostgreSQL1C.

Примеры шаблонов:

*1C - Npgsql Standart*

- Endpoint Type (Тип подключения) – postgresql1c;
- Template (Шаблон) – 1С - Npgsql Standart (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Type Work 1C Translator (Тип режима работы 1С) представлен двумя позициями: canvas/hard;
- Base Key 1С (Название БД 1С);
- Server (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;

- Connection string (Строка подключения) – здесь создаётся строка подключения из заполненных полей выше: TypeWork1CTranslator=canvas; BaseKey1C=default; Server=localhost; Port=3306; Database=postgres; User Id=postgres; Password=\*\*\*; Include Error Detail=false;

#### *1C - Standart*

- Endpoint Type (Тип подключения) – postgresql1c;
- Template (Шаблон) – 1C - Standart (выбирается в зависимости от настроек endpoint);
- Max connections (Максимальное число подключений);
- Поле Enabled (Включён) представлено в выпадающем списке двумя позициями: true/false;
- Type Work 1C Translator (Тип режима работы 1С) представлен двумя позициями: canvas/hard;
- Base Key 1C (Название БД 1С);
- 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;
- Connection string (Строка подключения) – здесь создаётся строка подключения из заполненных полей выше: TypeWork1CTranslator=canvas; BaseKey1C=default; Server=localhost; Port=3306; Database=postgres; User Id=postgres; Password=\*\*\*; Include Error Detail=false;

# ПАРАМЕТРЫ

Параметры описать страницу параметры

Для создания/удаления/копирования параметра необходимо использовать кнопки ( **Создать** / **Удалить** / **Копировать** )

The screenshot shows the BI.Qube interface with the 'Parameters' section selected. On the left, there's a sidebar with navigation links like 'Общее', 'Профили', 'Параметры', 'Данные', 'Подключения', 'Staging', 'Data & Model', 'Control', and 'Orchestrator'. The main area has a header 'Параметры' with buttons 'Создать', 'Удалить', and 'Копировать'. A search bar says 'Введите строку поиска' and a 'Больше информации' link. The table lists parameters with columns: 'Наименование', 'Тип', 'Значение', and 'Описание'. One row is highlighted for 'param5'. To the right, a modal window titled 'Редактор' shows the details for 'param5': 'Код параметра' (5), 'Наименование' (param5), 'Тип' (SQL), 'Значение' (select \* from meta.endpoint\_template), and 'Описание' (Тестирование модального окна создания запроса). There are also 'Обновить' and 'Сохранить' buttons.

Рисунок. Страница Parameters (Параметры)

This is a detailed view of the 'Создание' (Create) dialog for a new parameter. It includes fields for 'Код параметра' (5), 'Наименование' (param5), 'Тип' (SQL), 'Значение' (select \* from meta.endpoint\_template), 'Описание' (Тестирование модального окна создания запроса), and an optional 'Сконструировать' (Construct) button. The 'Сконструировать' button is highlighted in blue.

Рисунок. Окно свойств справа заполненной/ незаполненное (пример)

Окно свойств справа содержит следующие поля для заполнения:

- Code (Код параметра)

- Name (Наименование)
- Type (Тип)
- Value (Значение)
- Description (Описание)

# METASTAGING

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, при этом в этом слое данные обычно хранятся в том виде, как они представлены в источнике. MetaStaging поддерживает достаточно сложные сценарии создания детального слоя:

1. детальный слой формируется полностью в реляционном слое – наиболее распространенный подход к организации подготовки детального слоя;
2. формируется озеро данных (data lake) – файловое хранилище, файлы представлены в формате \*.parquet, с автоматическим формированием в реляционном слое объектов External Table с возможностью материализации;
3. дополнительно к первым двум сценариям есть возможность сохранения истории загрузок данных, в оригинальном формате, представленных на источниках в форматах xls, xlsx, csv, xml, json.

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.

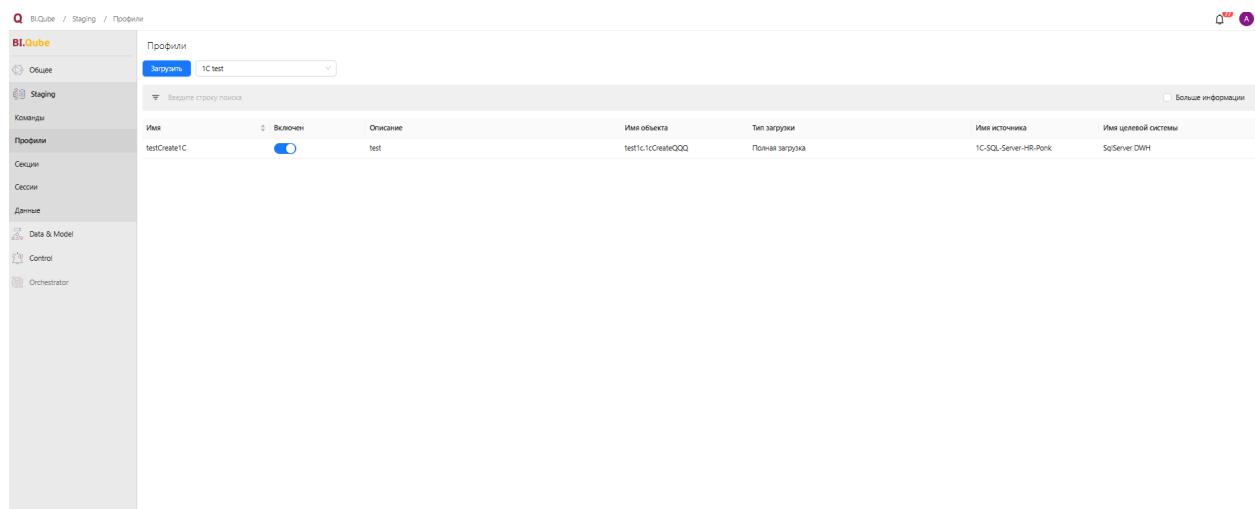
Система поддерживает два режима выполнения команд:

- с использованием веб-интерфейса
- с использованием планировщика (оркестратора), рекомендуется применять Airflow.

В текущем руководстве рассмотрена работа в режиме веб-интерфейса.

# ПРОСМОТР ПРОФИЛЕЙ

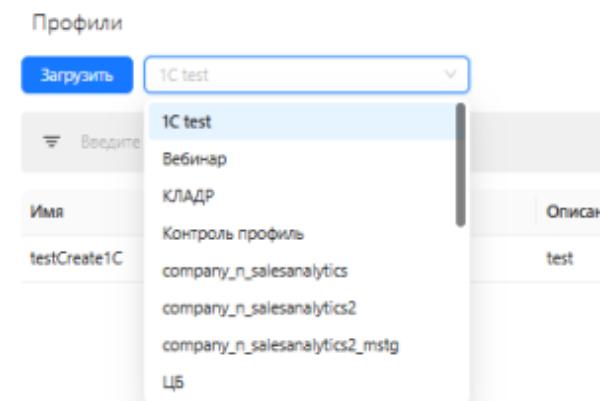
Для просмотра созданных профилей необходимо зайти в раздел Стейджинг (Staging) во вкладку Профили (Profiles ).



The screenshot shows the BIQube application interface. On the left, there is a sidebar with various navigation items: Общие, Staging (which is selected), Команды, Профили (selected), Сессии, Данные, Data & Model, Control, and Orchestrator. The main area is titled "Профили" (Profiles). A search bar at the top has the text "1C test". Below it is a table with columns: Имя (Name), Включен (Enabled), Описание (Description), Имя объекта (Object Name), Тип загрузки (Load Type), Имя источника (Source Name), and Имя целевой системы (Target System Name). There is one row in the table: "testCreate1C" with "Включен" set to "Yes", "Описание" as "test", "Имя объекта" as "test1c.1cCreateQQQ", "Тип загрузки" as "Полная загрузка" (Full Load), "Имя источника" as "1C-SQL-Server-HR-Polk", and "Имя целевой системы" as "SqlServer DWH". A checkbox labeled "Больше информации" (More info) is located at the bottom right of the table.

Рисунок. Пример созданного профиля

Для просмотра и выбора, необходимо выбрать нужный профиль в выпадающем списке. (Рисунок. Выбор профиля).



The screenshot shows the BIQube interface with the "Profiles" section selected. A dropdown menu is open under the "Загрузить" (Load) button, which contains the text "1C test". The dropdown also lists other profiles: "Вебинар", "КЛАДР", "Контроль профиль", "company\_n\_salesanalytics", "company\_n\_salesanalytics2", "company\_n\_salesanalytics2\_mstg", and "ЦБ". To the right of the dropdown, there is a tooltip with the text "Описан" (Described) and the value "test".

Рисунок. Выбор профиля

После выбора интересующего имени профиля на экране появится перечень команд включенных в этот профиль. Более детально про работу с профилями будет рассказано в разделе "Запуск на выполнение"

Для просмотра дополнительной информации по сущностям (таблицам) необходимо поставить галочку More info (Больше информации).

# СОЗДАНИЕ И РЕДАКТИРОВАНИЕ КОМАНД ДЛЯ ЗАГРУЗКИ ДАННЫХ

- Общие параметры
- Команда «Создать»

## Общие параметры

При создании команды для загрузки данных из источника в точку назначения потребуется некоторый опыт работы с СУБД и начальное понимание SQL-запросов. Создание команды выполняется на странице "Команды" (Command)

The screenshot shows the BI.Qube interface with the 'Staging' tab selected. On the left, there's a sidebar with sections like 'Общее' (General), 'Staging', 'Команды' (Commands), 'Профили' (Profiles), 'Сессии' (Sessions), 'Данные' (Data), 'Data & Model', 'Control', and 'Orchestrator'. The main area is titled 'Команды' (Commands) and contains a table with columns: 'Имя' (Name), 'Описание' (Description), 'Запрос' (Query), 'Профили' (Profiles), 'Источник' (Source), 'Целевая система' (Target System), and 'Виртуальное хранилище' (Virtual Warehouse). There are 8 rows in the table, each showing a different command name, its description, a complex SQL query, and its target system ('Бизнес'). To the right of the table is a large 'Создание' (Create) form. It includes fields for 'Код' (Code), 'Имя' (Name), 'Описание' (Description), 'Профили' (Profiles), 'Источник' (Source), 'Целевая система' (Target System), 'Использовать промежуточное хранилище' (Use intermediate storage), 'Промежуточное хранилище DataLake' (Intermediate storage DataLake), 'Материализовать данные в конечной точке' (Materialize data in final destination), and 'Запрос' (Query). At the bottom of the page, there are navigation buttons (1, 2, 3, 4, 5, 20/стр.) and a 'Создать' (Create) button.

Рисунок. Страница работы с командами

Для создания новой команды необходимо нажать кнопку "Создать" (Create), после чего появится возможность заполнить все необходимые свойства создаваемой команды. Если необходимо создать команду с настройками, которые были созданы ранее в других таблицах можно нажать кнопку "Скопировать" (Copy) и все свойства для новой команды будут скопированы из той, которая была выбрана в таблице основной части экрана. Здесь нужно быть внимательным и обязательно проверить правильность заполнения всех полей. Команды с одним именем не допускаются.

**Рисунок. 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 (для удобства пользователя осуществлена группировка по типам источников);
5. Destination (Целевая система) – система, в которую планируется загрузить данные из источников (для удобства пользователя осуществлена группировка по типам целевой системы);
6. Use intermediate storage (Использовать промежуточное хранилище) – иногда при построении хранилищ требуется использовать дополнительное промежуточное 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 (Создать) – 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 сохранятся 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.

После заполнения всех полей ввода необходимо нажать на кнопку Create (Создать) в верхней части меню свойств. В строках «Команды» появится таблица с создаваемым именем.

*Возможность выбора промежуточного хранилища и выбора опции сохранения истории доступны не всем endpoints!*

## Команда «Создать»

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) – подключение «Локальные файлы» создается при развертывании;

- запрос извлечения данных из 1С Предприятие (на основе MS SQL Server, PostgreSQL) – если тип подключения соответствует выбранному типу подключения 1С;
- запрос извлечения данных из СУБД (MS SQL Server, Oracle, MySQL, PG, GP) – если тип подключения соответствует чему-то из перечисленного;
- запрос извлечения данных из веб-сервисов по протоколу REST API (JSON, XML, CSV) – если выбран тип подключения Rest API;
- запрос извлечения данных из общих каталогов windows (xls,xlsx, csv) – если тип подключения соответствует протоколу SMB.

Тип нужного диалогового окна определяется автоматически, на основе выбранного endpoint, используемого в качестве источника данных. Диалект SQL запроса зависит от типа источника данных, запрос будет выполняться на стороне источника.

Для удаления или копирования уже созданной команды, необходимо нажать на соответствующие кнопки (Рисунок. Кнопки создания/удаления/копирования команды).



Рисунок. Кнопки создания/удаления/копирования команды

# 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 щелкнуть по кнопке «OK». Выбранный файл автоматически загрузится в хранилище и подсветится в структуре каталогов. С этого момента файл доступен для анализа.

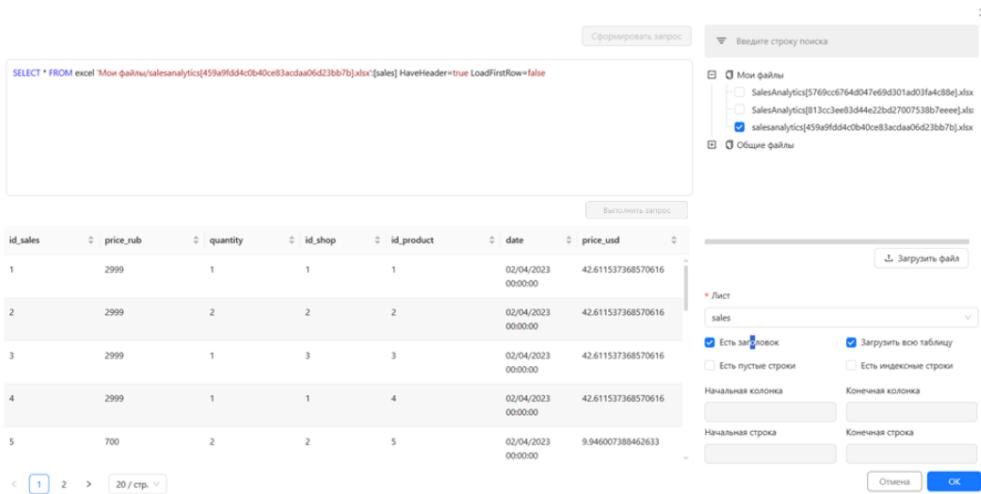


Рисунок. Загрузка файла с компьютера пользователя

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 результатирующего запроса и результатов его работы необходимо нажать на кнопку Form a query (Сформировать запрос) - программа сформирует простой SQL запрос, а затем необходимо нажать на кнопку Run query (Выполнить запрос), сформируется текст запроса и в зоне предварительного просмотра появятся результаты выполнения этого запроса.

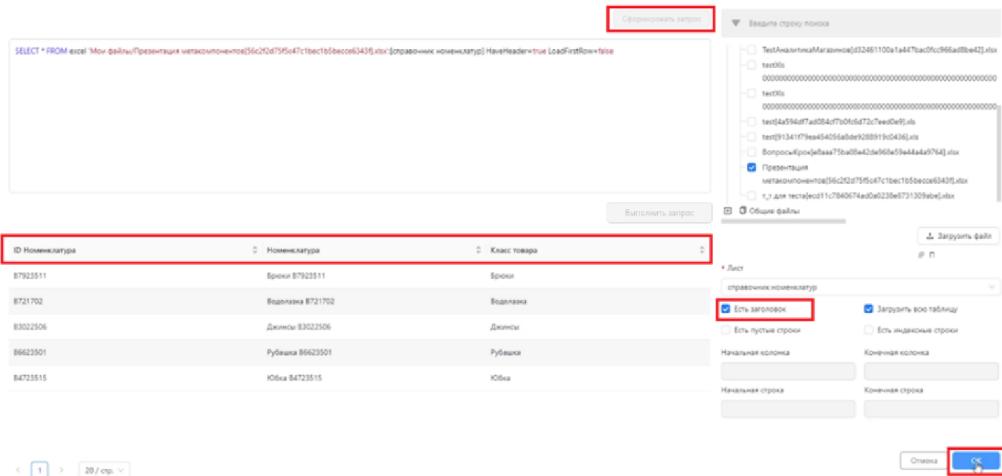


Рисунок. Сверка загруженного файла и добавление заголовков в таблицу

После окончания настройки запроса следует нажать кнопку «OK», 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.

Выбрать alias.ДатаРождения КАК ДатаРождения, alias.Пол КАК Пол, alias.ИИН КАК ИИН, alias.СтраховойНомерПФР КАК СтраховойНомерПФР, alias.МестоРождения КАК МестоРождения, alias.ГруппаДоступа КАК ГруппаДоступа, alias.ИмеетНаучныеТруды КАК ИмеетНаучныеТруды, alias.ИмеетИзобретения КАК ИмеетИзобретения, alias.ФИО КАК ФИО, alias.УточнениеНаименования КАК УточнениеНаименования, alias.ДатаРегистрации КАК ДатаРегистрации, alias.НаименованиеСлужебное КАК НаименованиеСлужебное, alias.ЛьготаПриНачисленииПособий КАК ЛьготаПриНачисленииПособий, alias.ПостоянноПроживалВКрыму18Марта2014Года КАК ПостоянноПроживалВКрыму18Марта2014Года, alias.Фамилия КАК Фамилия, alias.Имя КАК Имя, alias.Отчество КАК Отчество, alias.УдалитьИнициалыИмена КАК УдалитьИнициалыИмена, alias.Инициалы КАК Инициалы, alias.ОтношениеКОсновнойЗоне КАК ОтношениеКОсновнойЗоне, alias.ОбластьДанныхОсновныеДанные КАК ОбластьДанныхОсновныеДанные, alias.Ссылка КАК Ссылка, alias.ПометкаУдаления КАК ПометкаУдаления, alias.Родитель КАК Родитель, alias.Код КАК Код, alias.Наименование КАК Наименование, alias.Предопределенный КАК Предопределенный, alias.ВерсияДанных КАК ВерсияДанных ИС СправочникФизическиеЛица КАК alias

Рисунок. Пример отображения данных в 1С запросе

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

SQL

Сформировать запрос

Рисунок . 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 графическим 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.

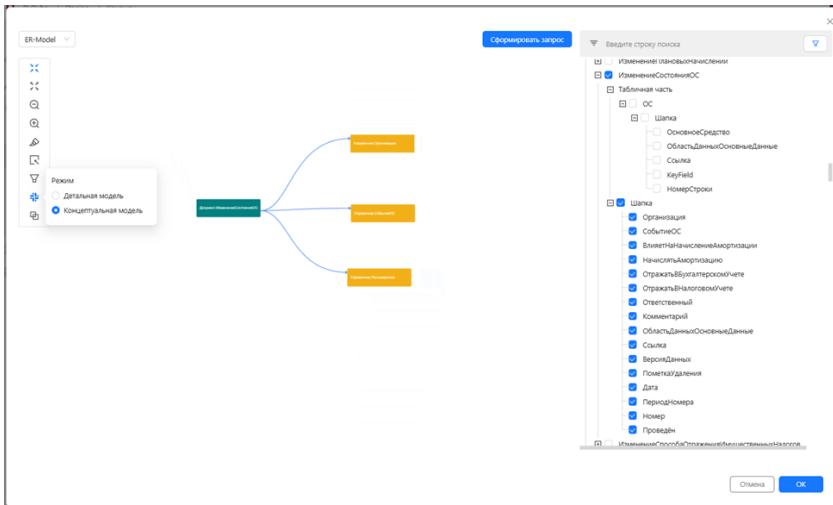


Рисунок . Концептуальная ER - модель отображения данных

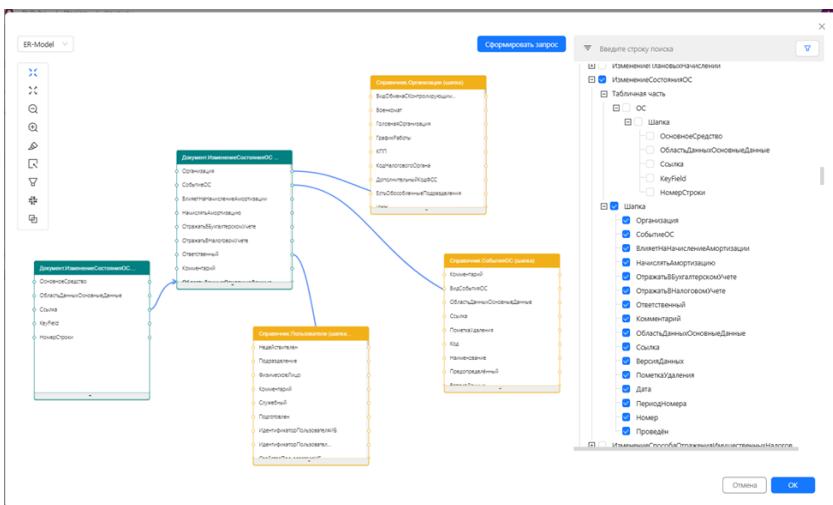


Рисунок. Детальная ER – модель отображения данных

Работа с окном заканчивается после того, как настроен запрос и нажата кнопка «OK» 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) - программа сформирует простой запрос на выборку всех данных, данный запрос можно редактировать, при этом, следует помнить, что нотация SQL запроса зависит от выбранного endpoint., а затем необходимо нажать на кнопку Run query (Выполнить запрос), сформируется текст запроса и в зоне предварительного просмотра появятся результаты выполнения этого запроса.

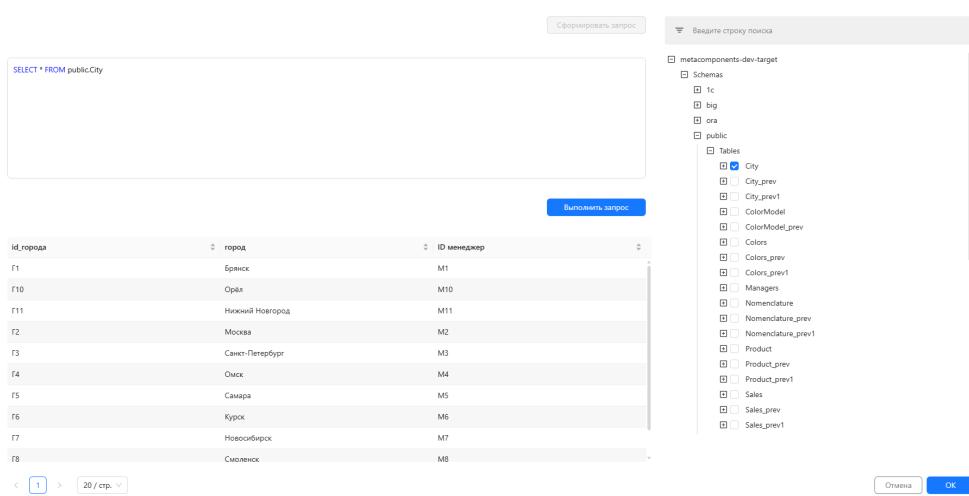


Рисунок. Пример запуска скрипта извлечённых данных из СУБД

После окончания настройки запроса необходимо нажать кнопку «OK» 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 в окне свойств справа нажать Create (Создать) в поле Query (Запрос). В строку Enter file address (Введите адрес файла) необходимо ввести адрес файла на веб-сервере, данные из которого необходимо загружать (под цифрой 1 на рисунке). Выбрать Method (метод загрузки) (под цифрой 2 на рисунке). Поддерживается 2 метода загрузки: POST, GET. После чего нажать кнопку Form a query (Сформировать запрос). Затем на кнопку Check request (Проверить запрос) (под цифрой 5 на рисунке). В поле под номером 6 на рисунке появится строка сформированного запроса - значит запрос проверен успешно. Далее нажать Run query (Выполнить запрос) (под цифрой 4 на рисунке).

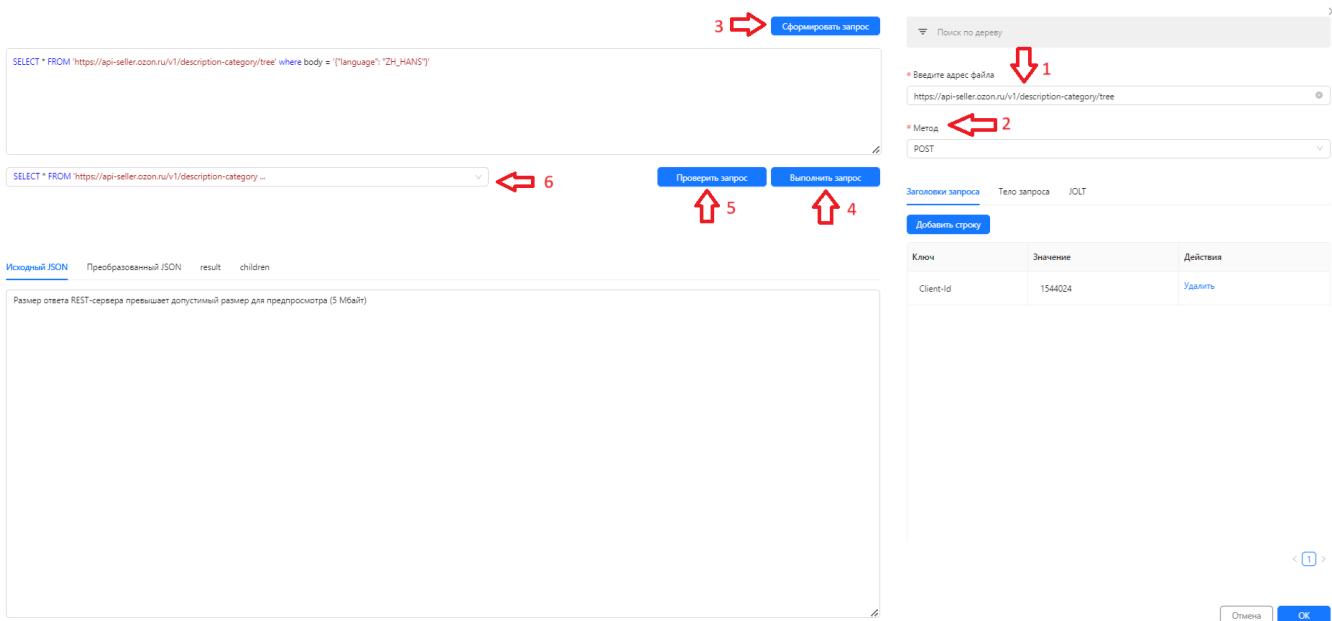


Рисунок. Диалоговое окно для формирования запроса Rest Api

Данные из файла отобразятся в поле просмотра ниже. Доступно 3 варианта отображения:

- Original JSON (Исходный JSON) - В случае, если размер ответа REST-сервера превышает допустимый размер для предпросмотра (5 Мбайт) отобразится соответствующее сообщение;
- Converted JSON (Преобразованный JSON) - В случае, если размер ответа REST-сервера превышает допустимый размер для предпросмотра (5 Мбайт) отобразится соответствующее сообщение;
- Табличный – данные будут разложены в нормализованные табличные структуры. Не всегда простой запрос может полностью в автоматическом режиме выполнить эту операцию. На рисунке отображен пример преобразованного исходного JSON к табличному виду в виде: result, children - эти вкладки отображают загружаемые данные в виде таблицы.

Исходный JSON	Преобразованный JSON	result	children
62573858	积木玩具套装	disabled	type_name type_id
		false	拼图书 97250
		false	塑料构造器 92952
		false	婴儿益智玩具 970956618
		false	粘粘积木 97711
		false	宝宝拼图 92951
		false	木制积木 92939
		false	纸质积木 97669
		false	金属构造器 92944
		false	磁性构造函数 92943
		false	儿童砖积木 97200

Рисунок. Пример преобразованного исходного JSON к табличному виду: result и children.

В окне справа во вкладке Request headers (Заголовки запроса) есть кнопка Add a row (Добавить строку) при нажатии на которую добавляется строка с редактируемыми полями: ключ, значение, действия.

Заголовки запроса	Тело запроса	JOLT
Добавить строку		
Ключ	Значение	Действия
Client-Id	1544024	<a href="#">Удалить</a>

Рисунок. Вкладка Request headers (Заголовки запроса).

Во вкладке Request body (Тело запроса) отображается само тело сформированного запроса.

Заголовки запроса	Тело запроса	JOLT
{ "language": "ZH_HANS" }		

Рисунок. Вкладка Request body (Тело запроса).

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

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

# Запрос извлечения данных из файловых хранилищ S3 и SMB

Добавить описание описание кнопки, разбивку, Excel

CSV

Сформировать запрос

Пожалуйста, введите запрос:

Проверить запрос Выполнить запрос

Виджет строки поиска

- Testentry.parquet
- identity.parquet
- hive.parquet
- increment
- increment@2022104001\_movies\_appear.parquet
- increment@2022104002\_movies\_appear.parquet
- increment@2022104003\_movies\_appear.parquet
- increment@2022105001\_movie\_tickets.parquet
- increment@2023010002\_movie\_tickets.parquet
- increment@2023010003\_movie\_tickets.parquet
- increment@2023010004\_movie\_tickets.parquet
- increment@2023010005\_movie\_tickets.parquet

Загрузить файл

Лист

Нет данных

Начальная колонка Конечная колонка

Начальная строка Конечная строка

OK Отмена

## 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. 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. 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.

\* Тип загрузки  
Инкрементальная загрузка

Размер пакета данных  
1000

Схема секционирования  
Схема секционирования

Поле секционирования  
Поле секционирования

Условия для секции  
Условия для секции

Рисунок. 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. 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. Параметры секционирования

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

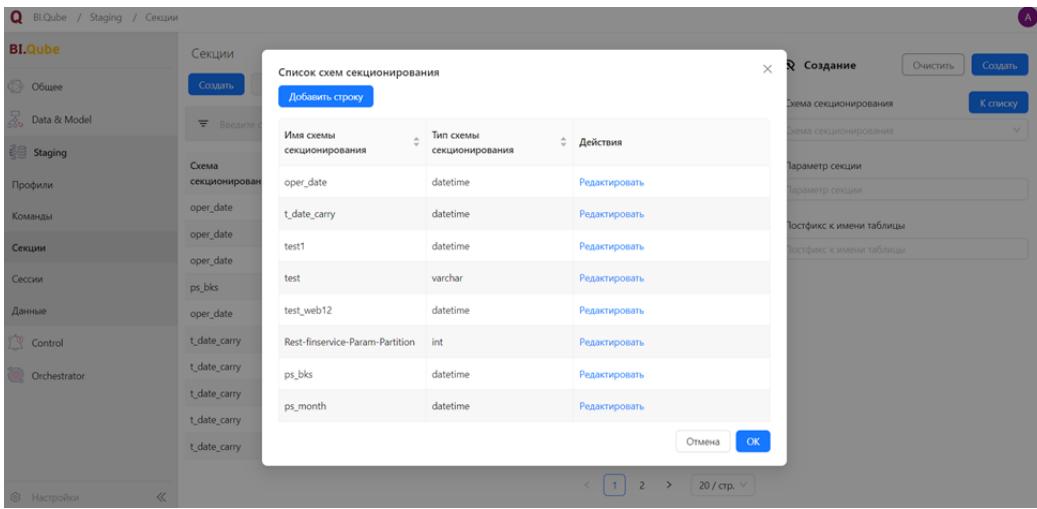


Рисунок. Модальное окно «Список схем секционирования»

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

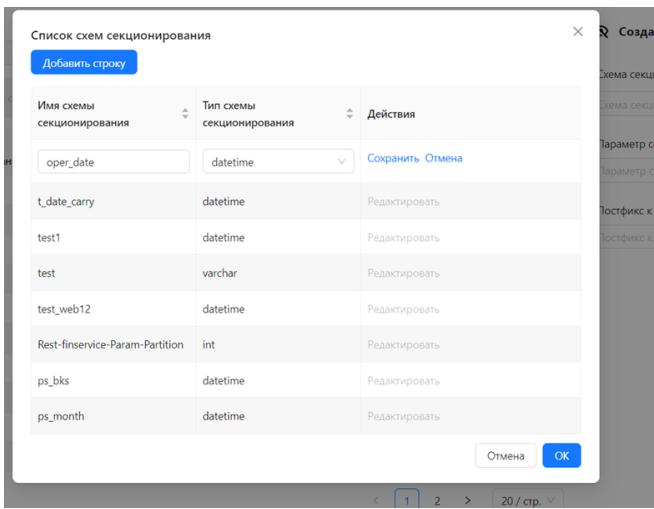


Рисунок. Список схем секционирования. Редактирование

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

Список схем секционирования		
Добавить строку		
Имя схемы секционирования	Тип схемы секционирования	Действия
oper_date	datetime	Редактировать
t_date_carry	datetime	Редактировать
test1	datetime	Редактировать
test	varchar	Редактировать
test_web12	datetime	Редактировать
Rest-finservice-Param-Partition	int	Редактировать
pt_bks	datetime	Редактировать
pt_month	datetime	Редактировать

Рисунок. 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.

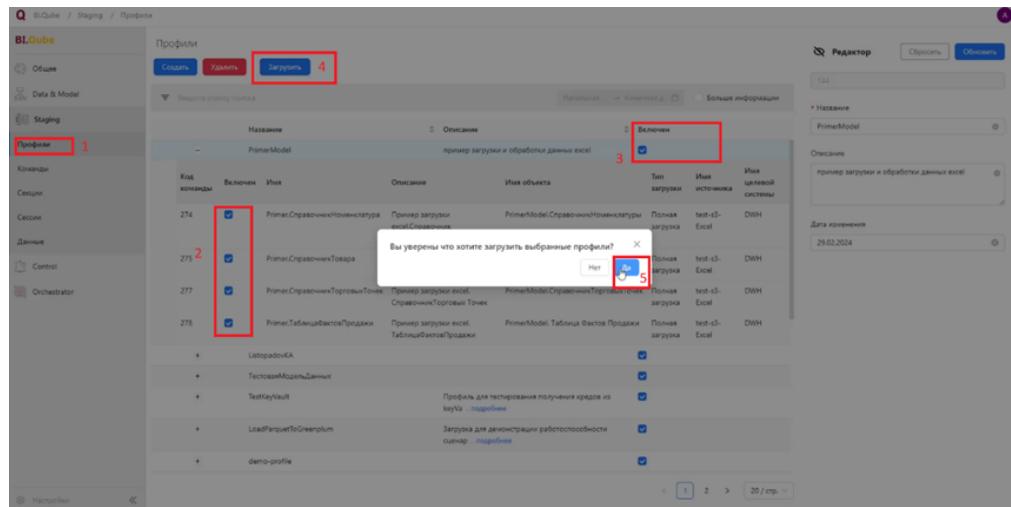


Рисунок. Выбор загруженных данных

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.

The screenshot shows the BIQube interface with the 'Sessions' tab selected. A search bar at the top right contains the placeholder 'Введите строку поиска'. Below it is a table with columns: 'Старт' (Start), 'Профиль' (Profile), and 'Команды' (Commands). The table lists multiple sessions, each with a '+' sign indicating expandable details. The first session, 'Start', is expanded, showing a detailed log of commands run, including 'LoadParquetToGreenplum' and 'PrimerModel'. At the bottom of the table, there are navigation buttons for pages 1 through 44, with '20 / стр.' (20 / page) indicated.

Рисунок Страница Session (Сессии)

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

Рисунок. Состав сессии

The screenshot shows a modal window titled 'Сведения о выполнении команды' (Information about command execution) with the sub-section 'Сводка о выполнении команды' (Summary of command execution). It displays a table of command executions with columns: 'Код' (Code), 'Время старта' (Start time), 'Сообщение стадийника' (Stage message), 'Сообщение ошибки' (Error message), 'Трассировка стека' (Stack trace), 'Команда для целевой системы' (Command for target system), and 'Статус' (Status). The table lists several entries, each with a status indicator: 'Старт' (Start), 'Успех' (Success), or 'Ошибка' (Error). One entry shows an error message related to schema creation. At the bottom of the modal, there are 'Отмена' (Cancel) and 'OK' buttons.

Рисунок. Окно, демонстрирующее детальные сведения о выполнении команды

## Статусы выполнения команд

Статусы проставляются в таблицу в соответствии с перечислением:

- Skipped (0) - команда не была выполнена из-за завершения сессии, т.е. выполнение команды даже не началось. По завершению сессии все команды со статусом Queued переводятся в Skipped через вызов XPI при использовании оркестратора, а при запуске через Backend через ProfileController;
- Success (1) - команда отработала без ошибок, данные загружены;
- Running (2) - команда в процессе загрузки (нельзя запускать данную команду в других профилях);
- Failed (3) - команда отработала с ошибками (см подробные логи);
- Queued (4) - Команда в очереди на загрузку (нельзя запускать данную команду в других профилях);
- Debug (5) - отладка команды (для внутренних задач, в т.ч. значение по умолчанию в БД);
- NoData (6) - команда отработала без ошибок, но данные из источника не загружены (возможно их нет в источнике).

Все статусы, кроме Skipped проставляются внутри экстрактора.

## **Статусы сессий загрузки**

Статусы сессия не хранятся в БД, а являются вычисляемыми.

- **Success** - Ни одна команда в сессии не имеет статус Running, Failed, Queued, Skipped;
- **Running** - Как минимум одна команда имеет статус Running;
- **Failed** - Как минимум одна команда имеет статус Failed.

# ДАННЫЕ METASTAGING

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.

The screenshot shows the BI.Cube interface with the 'Data' (Данные) tab selected. On the left, a sidebar lists various sections: Общее, Data & Model, Staging, Профили, Команды, Секции, Сессии, Данные, Control, and Orchestrator. The 'Данные' section is currently active. In the main area, there's a search bar with the placeholder 'Введите строку поиска' and a dropdown menu titled 'Назначение' containing 'DWH (postgresql)' which is highlighted with a red box. Below the search bar is a table titled 'СправочникНоменклатуры' with the following data:

ID Номенклатура	Номенклатура	Класс товара
83022506	Джинсы 81022506	Джинсы
84723515	Юбка 84723515	Юбка
86623501	Рубашка 86623501	Рубашка
8721702	Водолазка 8721702	Водолазка
87923511	Брюки 87923511	Брюки

Below the table is a button labeled 'Запустить скрипты'. To the right of the table is a tree view of database objects under 'Назначение':

- Схемы:
  - ExcelModel
  - PrimeModel
  - Tables
    - Таблица Фактов Продажи
    - СправочникНоменклатуры
    - СправочникТовара
    - СправочникТорговыхПунктов
- auth\_test
- public
- snapshot/ExcelModel
- stg
- test
- testexcel
- vault
- vault\_v

A red arrow points from the text 'Здесь же есть возможность создавать хранимые процедуры и другие объекты базы данных необходимые для поддержки работы хранилища.' to the 'Tables' node in the tree view.

Рисунок. Просмотр загруженных данных

*В разработке. Здесь же есть возможность создавать хранимые процедуры и другие объекты базы данных необходимые для поддержки работы хранилища.*

# DATA & MODEL

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 DATA & MODEL

Для просмотра созданных профилей необходимо зайти в "DATA & MODEL" во вкладку Profiles (Профили). (Рисунок. Пример созданного профиля)

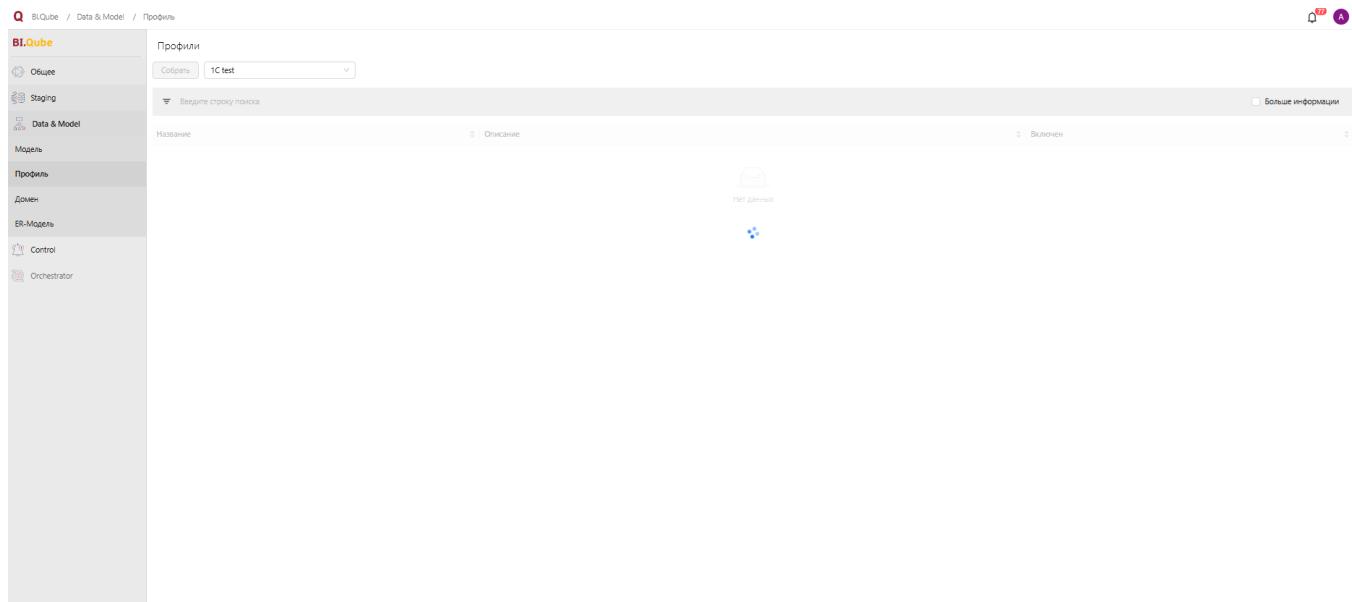


Рисунок. Пример созданного профиля

Для просмотра и выбора, необходимо выбрать нужный профиль в выпадающем списке. (Рисунок. Выбор профиля).

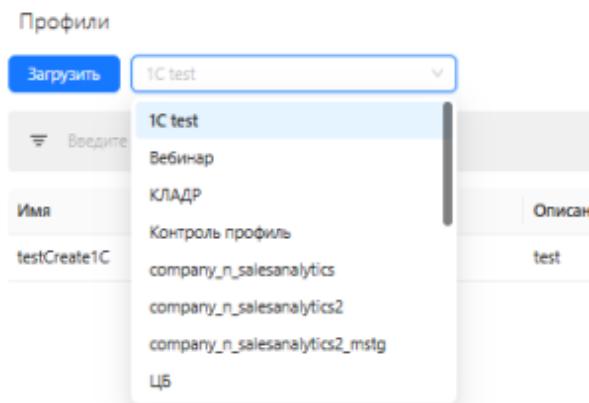


Рисунок. Выбор профиля

Для того, чтобы загрузить необходимые сущности (таблицы) необходимо выделить их и нажать на кнопку Load (Загрузить). В появившемся диалоговом окне нажать на кнопку Yes (Да). (Рисунок. Выбор и загрузка сущностей (таблиц)).

Имя Включен Описание Имя объекта Тип загрузки Имя источника Имя целевой системы

Факты	<input checked="" type="checkbox"/>	Загрузка таблиц вебинара	stg_Факты	Полная загрузка	SQLServer	PostgreWebinar
СправочникДело	<input checked="" type="checkbox"/>	Загрузка таблиц вебинара	stg_Дело	Полная загрузка	SQLServer	PostgreWebinar
СправочникКодРемонта	<input checked="" type="checkbox"/>	Загрузка таблиц вебинара	stg_КодРемонта	Полная загрузка	SQLServer	PostgreWebinar
СправочникФизлиц	<input checked="" type="checkbox"/>	Загрузка таблиц вебинара	stg_Физлица	Полная загрузка	SQLServer	PostgreWebinar
сводПрчинПерепростоя	<input checked="" type="checkbox"/>	Загрузка таблиц вебинара	stg_ПрчинныПерепростоя	Полная загрузка	SQLServer	PostgreWebinar
СправочникСостоинияРемонта	<input checked="" type="checkbox"/>	Загрузка таблиц вебинара	stg_СостоинияРемонта	Полная загрузка	SQLServer	PostgreWebinar
СправочникВидыРабот	<input checked="" type="checkbox"/>	Загрузка таблиц вебинара	stg_ВидыРабот	Полная загрузка	SQLServer	PostgreWebinar
СправочникМестоРемонта	<input checked="" type="checkbox"/>	Загрузка таблиц вебинара	stg_МестоРемонта	Полная загрузка	SQLServer	PostgreWebinar
СправочникВидРемонта	<input checked="" type="checkbox"/>	Загрузка таблиц вебинара	stg_ВидРемонта	Полная загрузка	SQLServer	PostgreWebinar
СправочникЛокомотивов	<input checked="" type="checkbox"/>	Загрузка таблиц вебинара	stg_Локомотивов	Полная загрузка	SQLServer	PostgreWebinar
СправочникОтветстваяСторонаР	<input checked="" type="checkbox"/>	Загрузка таблиц вебинара	stg_ОтветстваяСторонаР	Полная загрузка	SQLServer	PostgreWebinar
Вебинар_АктО выполнении Этапа Раб от Загрузки	<input checked="" type="checkbox"/>		stg_Акт_Затраты	Полная загрузка	1C_вебинар	PostgreWebinar
TimeLog_вебинар	<input checked="" type="checkbox"/>	Выгружаем запрос из девопса	stg_devops	Полная загрузка	DevOps_вебинар	PostgreWebinar
Вебинар_АктО выполнении Этапа Раб от Шапки	<input checked="" type="checkbox"/>		stg_Акт_Шапка	Полная загрузка	1C_вебинар	PostgreWebinar
Area_Devops_вебинар	<input checked="" type="checkbox"/>		stg_devops_area	Полная загрузка	DevOps_вебинар	PostgreWebinar

Вы уверены что хотите загрузить выбранные профили?

Рисунок. Выбор и загрузка сущностей (таблиц)

Для просмотра дополнительной информации по сущностям (таблицам) необходимо поставить галочку More info (Больше информации)

# 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).

The screenshot displays two screenshots of a web-based domain management system. The top screenshot shows the 'Create' screen for a new domain. It features a 'Name' field containing 'testDomain' and a 'Description' field containing 'test create1'. The bottom screenshot shows the 'Edit' screen for the same domain ('testDomain'). It includes a detailed view with columns for 'Code' (525), 'Name' (testDomain), 'Description' (test create1), 'Author' (admin), and 'Modified date' (14.02.2024). Both screenshots have red boxes highlighting the 'Create' button, the 'More info' link, and the 'Edit' button.

Рисунок. 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.

The screenshot shows the BI.Qube interface with the 'Data & Model' section selected. In the center, there's a table of models. One model, 'тестовая модель данных', is selected. To its right, a modal window titled 'Creator' is open, allowing the user to input a new model's name and description. The interface includes standard navigation elements like a search bar, filters, and a page number indicator.

Рисунок. Создание модели данных

Редактирование имени и описание выполняется аналогичным образом, щелкнуть левой кнопкой 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.

This screenshot shows the 'Entities' section for the 'тестовая модель данных'. It lists several entities: 'валидатор', 'Номенклатура...', 'Справочник Города...', 'Справочник менеджеров', 'Справочник товара', 'Справочник торговые точки', 'Справочник цвет', 'Справочник цветомодель', and 'Таблица фактов продажи'. Each entity has a description and a profile listed. A modal window on the right provides a detailed view of an entity, including fields for 'Name', 'Domain', 'Profiles', 'Description', 'Source table', and 'Keys'.

Рисунок. 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;
- Domen (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;
- Materialize historical view (историчное представление) – материализация данных изменений 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. Свойства сущности

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

BI.Qube / Data & Model / Модель / тестовая модель данных

**Сущности** | Назад

**Редактор** Сбросить Обновить

Код 2717

Ведите строку поиска Все домены Начальная Конечная Больше информации

Название	Описание	Профили	Домен
вальяжива		тестовый профиль	ПустаяСущность
Номенклатура...	справочных номенклатур, загружен из Excel "имя файла .логофоне"	тестовый профиль	
ПустаяСущность	Пример создания пустой сущности	тестовый профиль	
Справочник Города...	Справочник Города, из excel TestАналитика магазин .логофоне	тестовый профиль	
Справочник менеджеров	Справочник менеджеров, загружен из excel TestДанные .логофоне	тестовый профиль	
Справочник товара	Справочник товара, загружен из excel TestАналитика .логофоне	тестовый профиль	
Справочник торговые точки	Справочник торговые точки, загружен из excel TestA .логофоне	тестовый профиль	
Справочник цвет	Справочник цвет, загружен из excel TestАналитикаMa .логофоне	тестовый профиль	Таблица-источник
Справочник цветомодель	Справочник цветомодель, загружен из excel TestАнал .логофоне	тестовый профиль	Целевая таблица-источник Выбрать
Таблица фактов продажи	Таблица фактов, TestАналитикаМагазинов	тестовый профиль	Ключи
		code	Число Удалить Добавить

Настройки <<

Рисунок. Созданная сущность

# Создание сущности без данных

в том числе заполнение данными вручную

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.

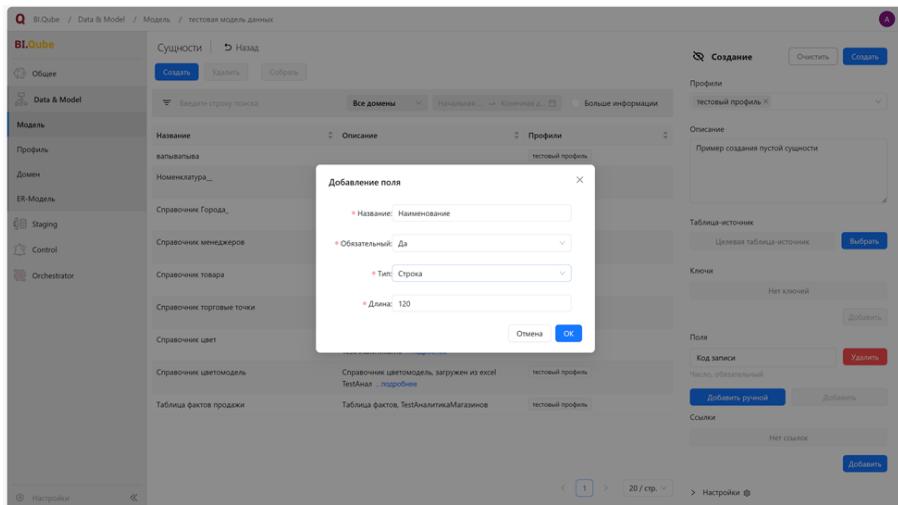


Рисунок. Создание атрибутов сущности

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

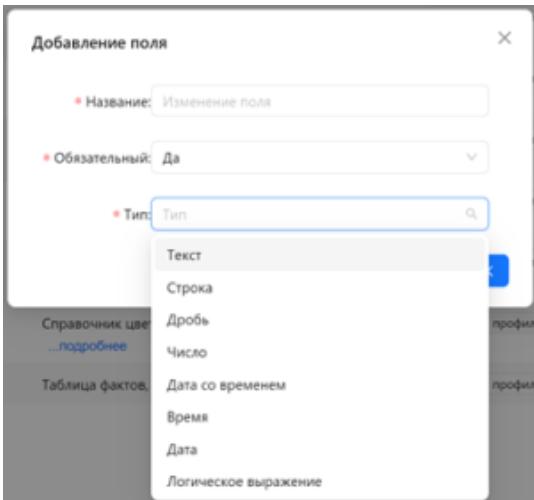


Рисунок. Доступные типы данных

# Создание сущности на основе данных БД

Для создания сущности, которая может быть заполнена данными из источника данных необходимо указать источник данных (Таблица-источник) для этого 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).

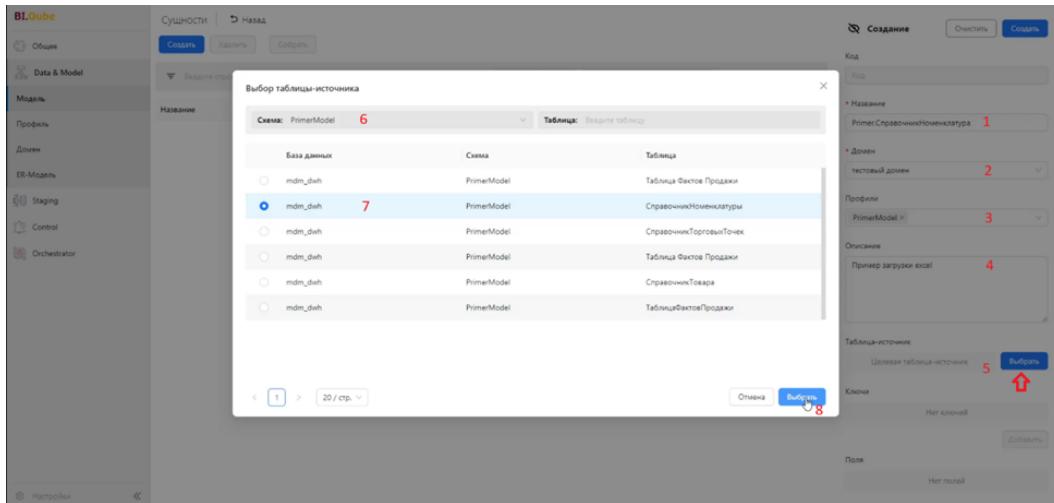


Рисунок. Описание заполнения сущности

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).

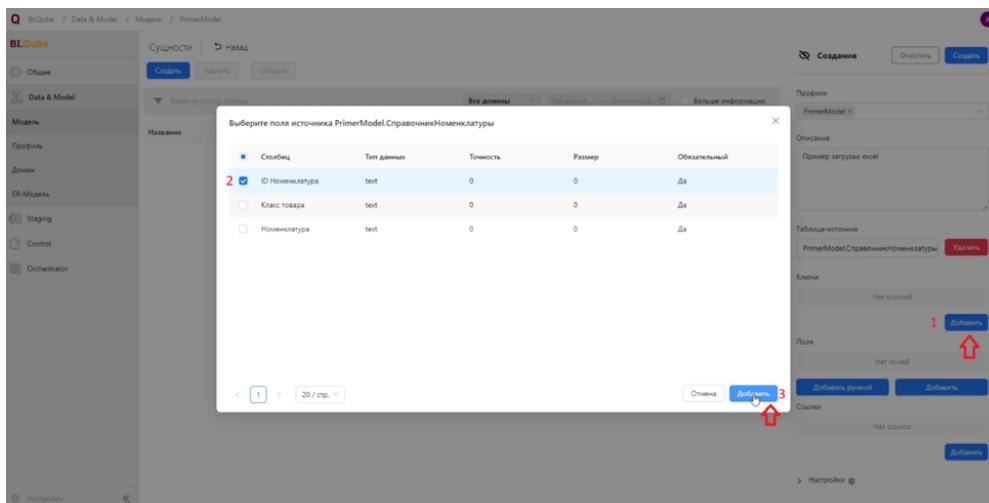


Рисунок. Заполнение поля «Ключи»

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.

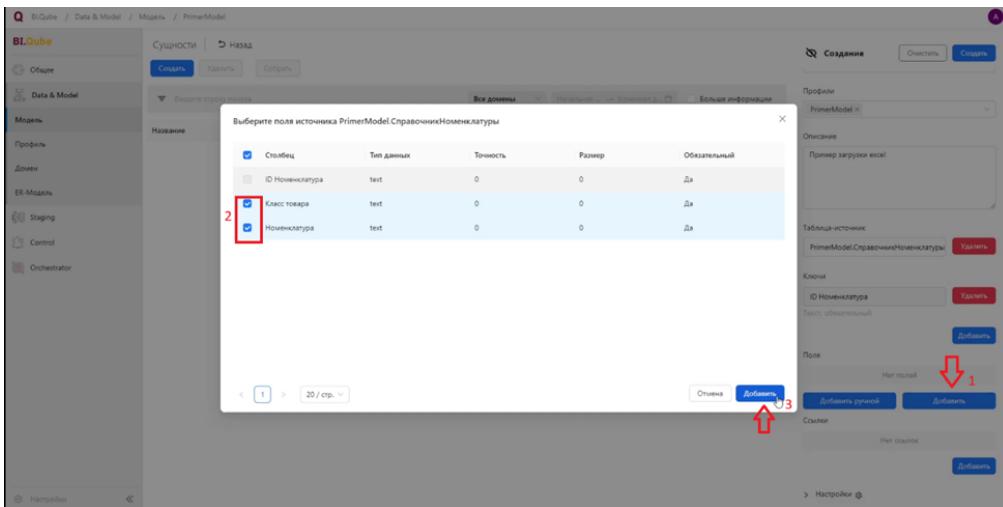


Рисунок. Поэтапное добавление полей

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

# Просмотр и редактирование данных

- Добавление и редактирование данных
- Фильтрация данных
- Режимы массового редактирования сущности (таблицы)
- Настройка режимов отображения данных

## Добавление 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).

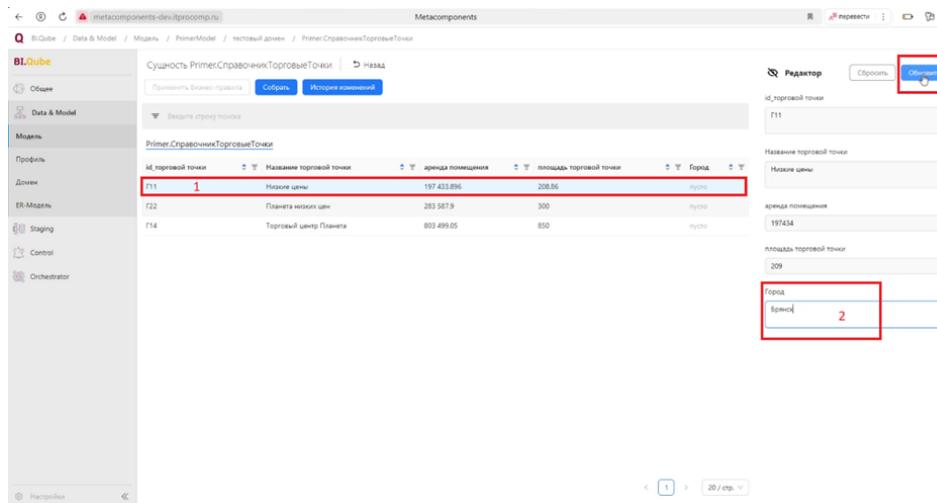


Рисунок. Заполнение добавленного поля в сущности (таблице)

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	2001-01-01T00:00:00Z	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	2024-03-14T12:34:25	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, выфиммав, empty)	admin	2024-03-14T12:36:56	2024-03-14T12:36:56Z	2024-03-14T12:39:34Z	false
2	4	пусто	(2, 4, выфиммав, 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, выфиммав, empty)	admin	2024-03-14T12:55:39	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	выфиммав	пусто	admin	2024-03-18T15:27:31	2024-03-18T15:27:32Z	9999-01-01T00:00:00Z	false

Рисунок. Просмотр истории изменения записей

При выборе любой сущности (таблицы) можно выбирать любой атрибут для редактирования щёлкнув по нему в окне свойств справа. В открывшемся диалоговом окне в поле (Описание) пользователь может задать любое описание атрибута и изменять его название в интерфейсе. Это возможно для всех атрибутов.

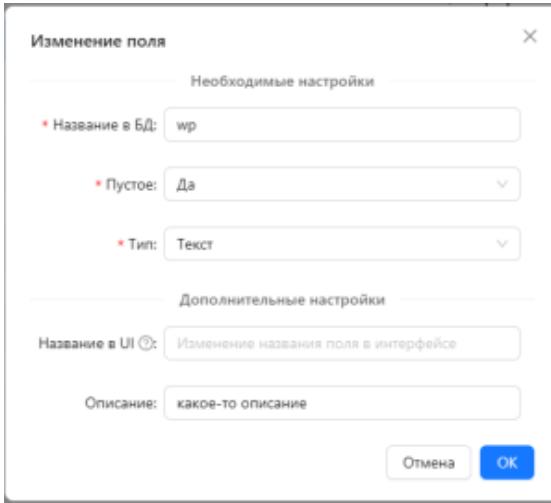


Рисунок. Диалоговое окно для внесения изменений в выбранный атрибут

## Фильтрация данных

При необходимости данные могут быть отфильтрованы. Окно настройки фильтра вызывается нажатием на иконку фильтр (  ), расположенный в заголовке каждой колонки таблицы.

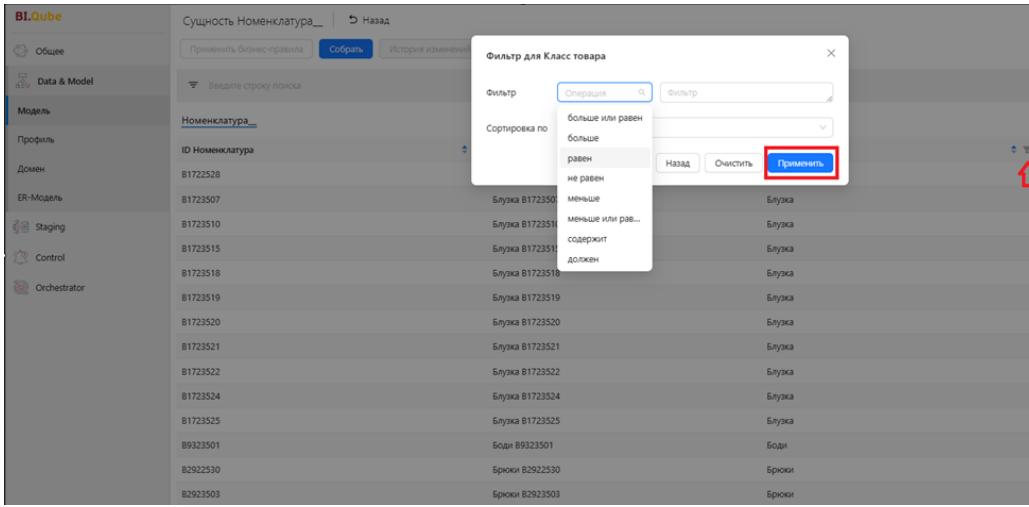


Рисунок. Окно настройки фильтра

Диалоговое окно настройки фильтра позволяет для выбранного поля Filter (Фильтр) использовать следующие операции:

- больше или равен;
- больше;
- равен;
- не равен;
- меньше;
- меньше или равен;
- содержит;
- должен.

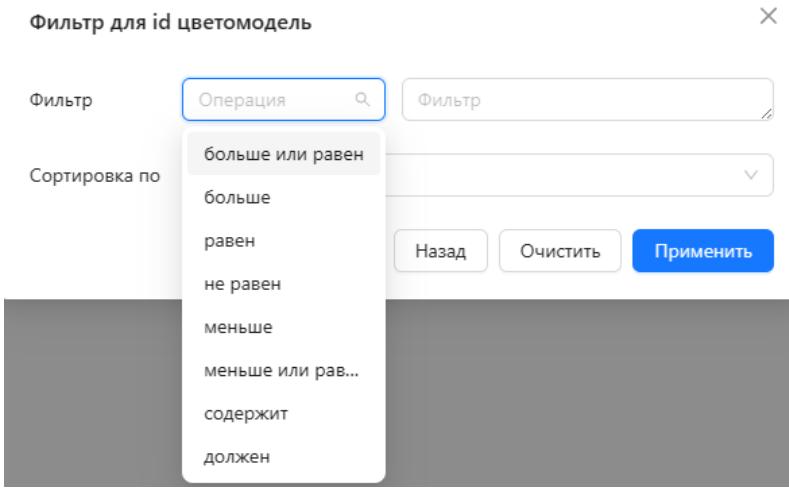


Рисунок. Выпадающий список в диалоговом окне фильтра по полю Filter (Фильтр)

Диалоговое окно настройки фильтра позволяет для выбранного поля Sorting by (Сортировка по) использовать следующие операции:

- по возрастанию;
- по убыванию.

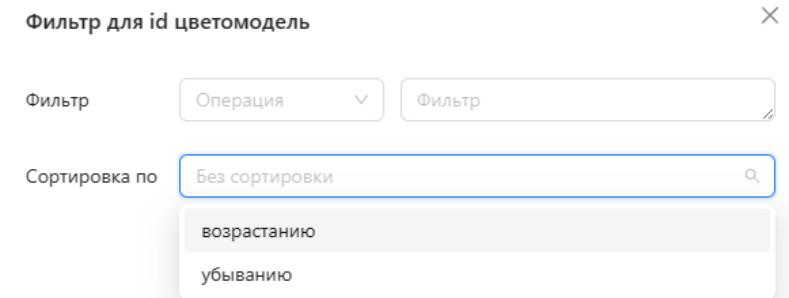


Рисунок. Выпадающий список в диалоговом окне фильтра по полю Sorting by (Сортировка по)

Важно! Поиск по словам осуществляется с учётом регистра.

### Режимы массового редактирования сущности (таблицы)

1. Для заполнения нескольких полей одновременно одинаковыми значениями можно выделить несколько строк в сущности (таблице). В окне свойств в нужное поле внести необходимые изменения, затем нажать кнопку Update (Обновить). И одновременно во всех выделенных столбцах появятся внесённые изменения, при этом следует помнить, что если в каких-то строках, в этом поле были данные, то они будут заменены новыми.

The screenshot shows a 'Mass Edit' interface for a table. On the left is a sidebar with navigation links: BI.Qube, Общее, Staging, Data & Model, Модель, Профиль, Домен, ER-Модель, Control, and Orchestrator. The main area shows a table with 12 rows and 17 columns. Row 5 is selected, highlighted in blue. To the right of the table is a 'Редактор' (Editor) panel with fields for columns 11 through 16. A message at the top of the editor panel says: 'Изменения будут применены для всех выбранных записей'. The editor panel contains fields for 'Текст' (Text) and 'Файл' (File).

Рисунок. Редактирование нескольких записей одновременно в сущности (таблице)

2. При включении фильтра вносимые изменения применяются ко всей отфильтрованной выборке по установленным параметрам. После нажатия на кнопку Apply (Применить), фильтр произведёт фильтрацию всей сущности (таблицы). В окне свойств справа появится уведомление о вносимых изменениях. После нажатия на кнопку Update (Обновить) появится диалоговое окно Confirmation of the update (Подтверждение обновления), в котором указаны те записи, для которых будут произведены изменения. Для подтверждения необходимо нажать на кнопку "OK".

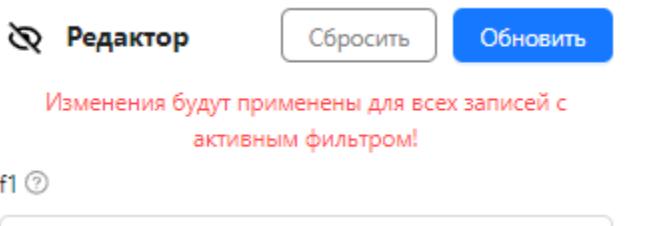


Рисунок. Предупреждение о вносимых изменениях в окне свойств

The screenshot shows the BI Suite interface. On the left, there's a confirmation dialog titled 'Подтверждение обновления' (Confirmation of update) with the message: 'Для всех записей будут применены следующие изменения:' (Changes will be applied to all records with the following changes:). It lists: '• Атрибут 'f1' будет изменен на '4вроло''. Below it, 'Примененные фильтры:' (Applied filters:) shows: '• Для атрибута 'code' применен фильтр 'равен' со значением ''4вроло''. On the right, the properties dialog shows a table with a filter applied to column f1, showing only rows where f1 contains '4вроло'. A red box highlights the filter input field.

Рисунок. Диалоговое окно подтверждения обновления.

Если необходимо очистить данные в каком-то поле сразу для нескольких строк, то можно выделить эти строки или воспользоваться фильтром, затем в окне свойств, в интересующем поле нажать на иконку "кисточка" ( ), после чего нажать кнопку Update (Обновить). Данные в этом поле для всех выделенных строк будут удалены.

### Настройка режимов отображения данных

С помощью иконки в форме шестерёнки( ) можно выбрать колонки для закрепления или поменять колонки местами. С помощью ползунка внизу можно листать таблицу.

The screenshot shows the BI Suite interface under the 'Data & Model' tab. A modal dialog titled 'Выберите зафиксированные колонки:' (Select frozen columns:) is open over a table. The table has several columns labeled f1 through f8. A red box highlights the first column 'code'. A red arrow points to the bottom of the table, indicating scrollable content. The modal dialog lists 'code' as a selected frozen column. Other columns like f1, f2, f3, f4, f5, f6, f7, f8 are shown as unfrozen options.

Рисунок. Выбор зафиксированных колонок

Очистить

) можно

При выборе зафиксированных колонок, данные колонки также фиксируются в окне свойств справа. С помощью кнопки ( ) можно очистить все выделенные колонки сразу.

The screenshot shows the BI.Qube interface for creating a new entity named 'f5'. A table is displayed with several columns. The first two columns ('поле1' and 'wp') are highlighted with a red border and marked as 'fixed' in the column header. To the right of the table, a 'Properties' panel is open, showing fields for 'name1' (with value 'Текст'), '2' (with value '2'), and '3'. Below these are dropdown menus for 'link', 'name1', and '2', each with a 'Clear' button. A modal window titled 'Выберите зафиксированные колонки:' lists 'ID', 'поле1' (checked), 'wp' (checked), 'link', 'name1', and '2'. The 'Clear' button for this modal is also highlighted with a red box.

Рисунок. Зафиксированные колонки

В режиме администратора могут создаваться предустановленные фильтры, что облегчает работу пользователя с большими таблицами. По умолчанию уже созданы три фильтра: главные атрибуты, вспомогательные атрибуты и линки.

The screenshot shows the BI.Qube interface for a large entity with many attributes. A table is displayed with columns labeled f1 through f15. A context menu is open over the 'f14' column, showing options: 'Главные атрибуты', 'Вспомогательные атрибуты', and 'Линки'. To the right, a 'Редактор' (Editor) panel is open for attribute 'code', showing its value '3'. Below it are panels for attributes f1, f2, f3, f4, f5, and f6, each containing their respective values: 'createMassEdit', 'Text', '1111111111', 'шев', 'Text', and 'Text' respectively.

Рисунок. Фильтрация

При наведении курсора мыши на поле любого из линков, появляется информация о его названии в БД (базе данных). При наведении курсора мыши на иконку ( ), появляется его описание. Тот же функционал доступен в окне свойств справа.

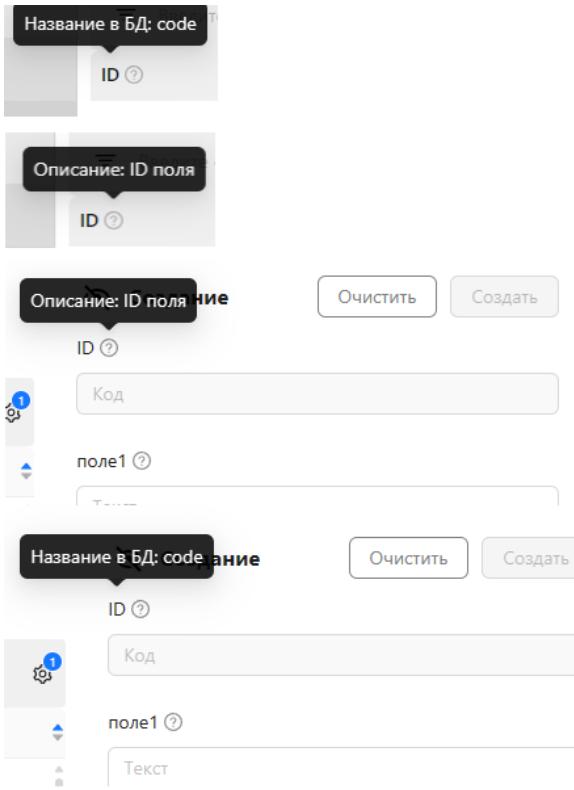


Рисунок. Всплывающие подсказки

Над колонками сущности есть кнопка "Проверка на дубли". При нажатии на данную кнопку система находит дублирующиеся значения и выделяет их цветом. Это удобно, если нужно проверить, какие данные в таблице повторяются и можно ли их удалять. Пока в реализации

поле1	wp	link	name1	2	3	4
testtest	dfgh		пусто	пусто	пусто	пусто
testtest	value		пусто	пусто	пусто	пусто
testtest	safad		{ testtest, safad }	2	пусто	пусто
testtest	safad		{ testtest, safad }	пусто	пусто	пусто
testtest	safad		пусто	пусто	пусто	пусто
testtest	safad		пусто	пусто	пусто	пусто

# 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.

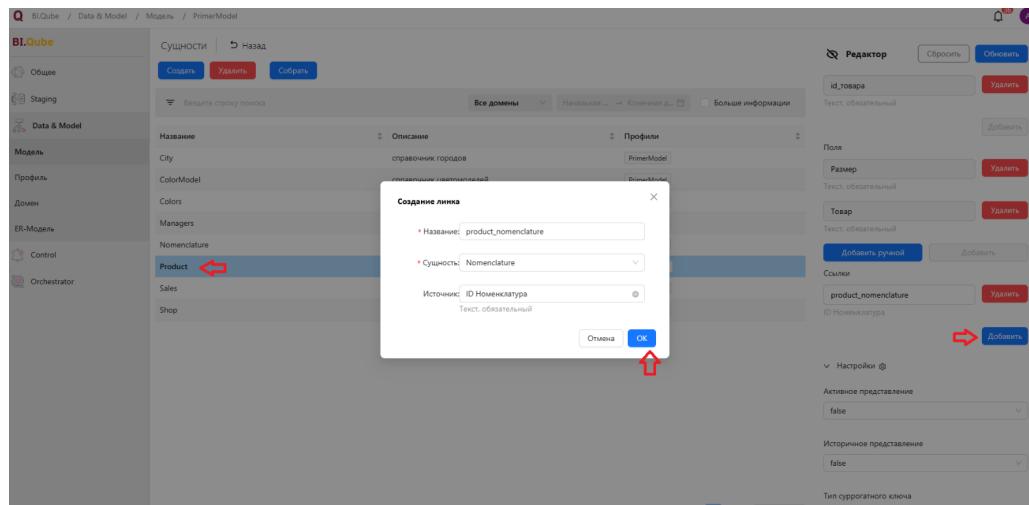


Рисунок. Выбор сущности для создания связи и диалоговое окно создания линка (связи)

Следует указать имя связи, выбрать связываемый объект, и выбрать атрибут текущей сущности, к которому привязываются данные сторонней сущности. Следует отметить, что связываемая сущность (таблица) связывается с текущей по бизнес ключу об этом, следует помнить при создании бизнес-ключей. В поле "Сущность" можно выбирать нужное из выпадающего списка, либо не выбирать ничего, и оно будет заполнено автоматически. Далее необходимо перейти в настройки отображения линка.

Для просмотра дополнительных настроек для линка необходимо нажать на иконку в форме шестерёнки. Появятся дополнительные поля:

- Активное представление представлено в выпадающем списке двумя значениями: true/false;
- Историчное представление представлено в выпадающем списке двумя значениями: true/false;
- Тип суррогатного ключа;
- Атрибуты в связанных сущностях можно выбирать по своему усмотрению из выпадающего списка.

This screenshot shows the 'Настройки' (Settings) section for a link. It includes fields for 'Активное представление' (Active representation) set to 'false', 'Историчное представление' (Historical representation) set to 'false', 'Тип суррогатного ключа' (Surrogate key type) set to 'Int32', and a 'Атрибуты в связанных сущностях' (Attributes in associated entities) dropdown containing 'Размер' (Size), 'Товар' (Product), and 'id\_товара' (Product ID).

Рисунок. Выбрана все атрибуты

▼ Настройки 

Активное представление



Историчное представление



Тип суррогатного ключа

Размер	
Товар	
id_товара	



Рисунок. Выбрана только часть атрибутов

# 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.

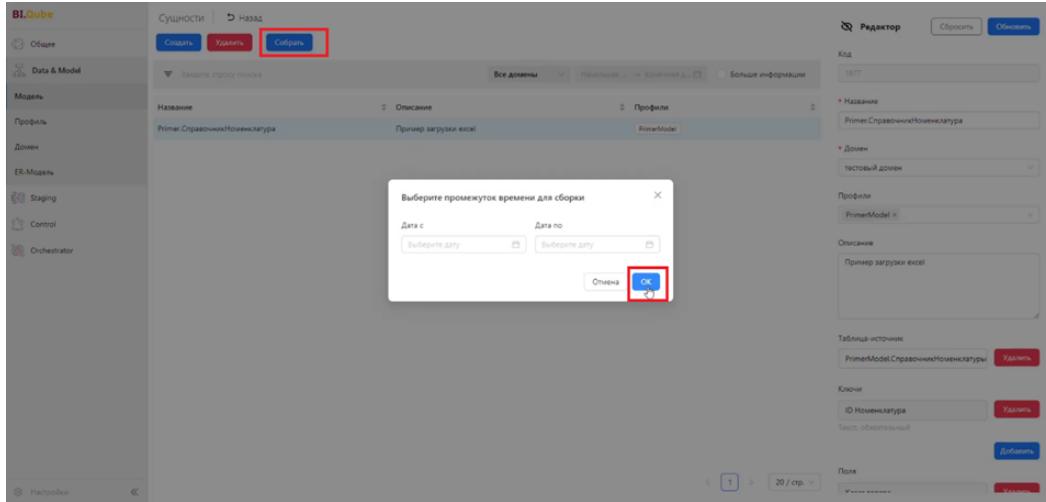


Рисунок. Сборка сущности

В основном окне нажать кнопку «Собрать», затем в появившемся диалоговом окне нажать «OK». 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 визуально 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. Выбор ER – модели в BI.Qube

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

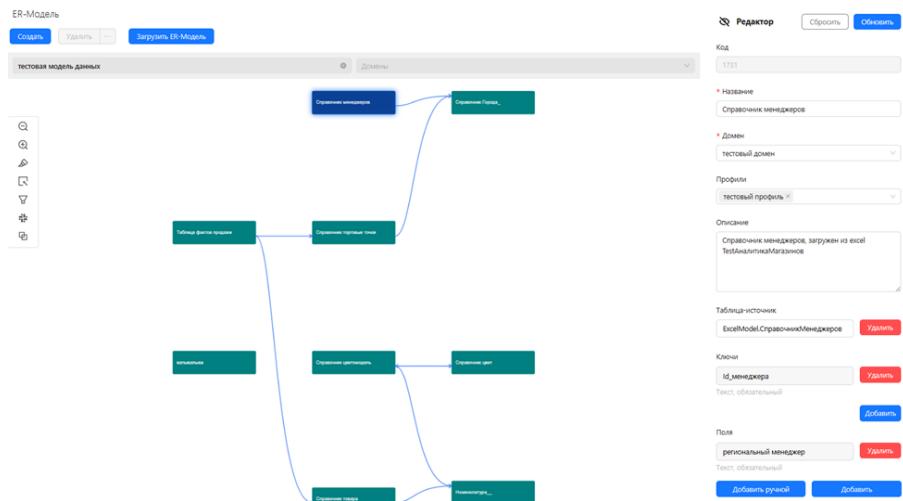


Рисунок. Редактирование таблицы в ER – модели

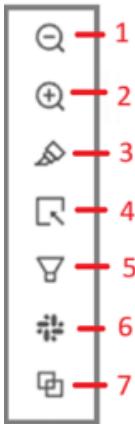


Рисунок. Панель инструментов (1 и 2 – лупа (уменьшить/увеличить); 3 – подсветка связей; 4 – фильтр слоёв; 5 – фильтр связей; 6 – режим отрисовки ER – модели; 7 – макет)

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).

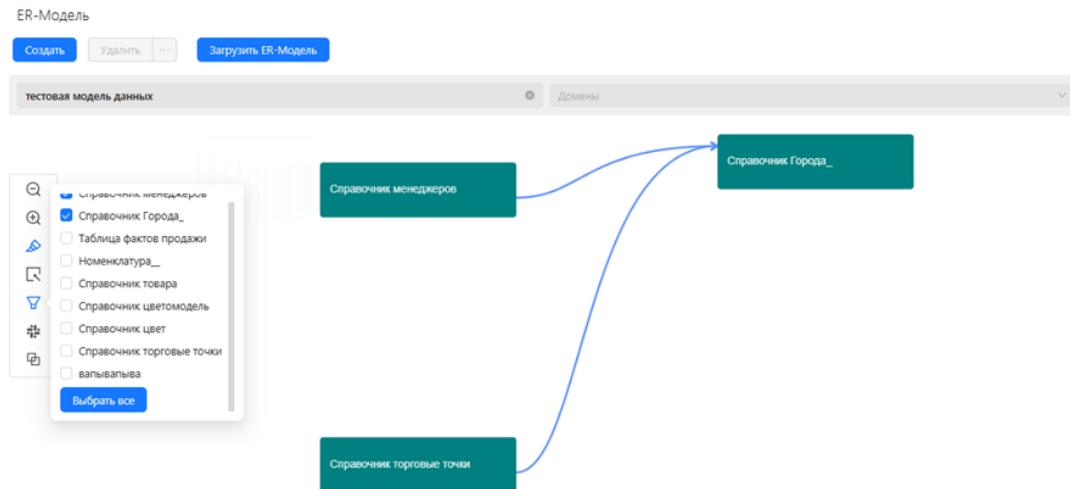


Рисунок. Работа с фильтром связей в режиме концептуальная модель

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.

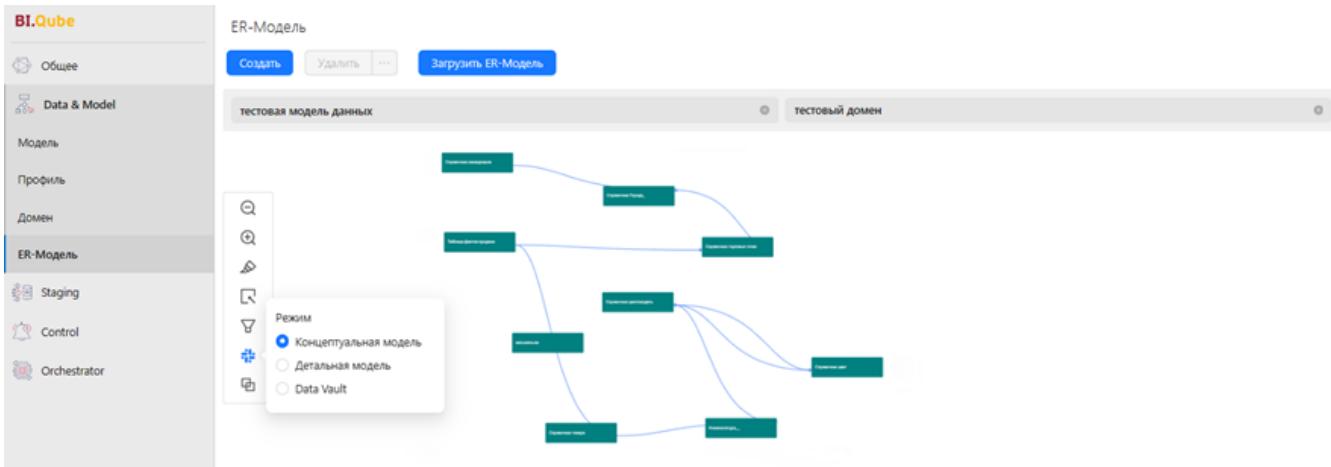


Рисунок. Пример отображения связей в концептуальной модели

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.

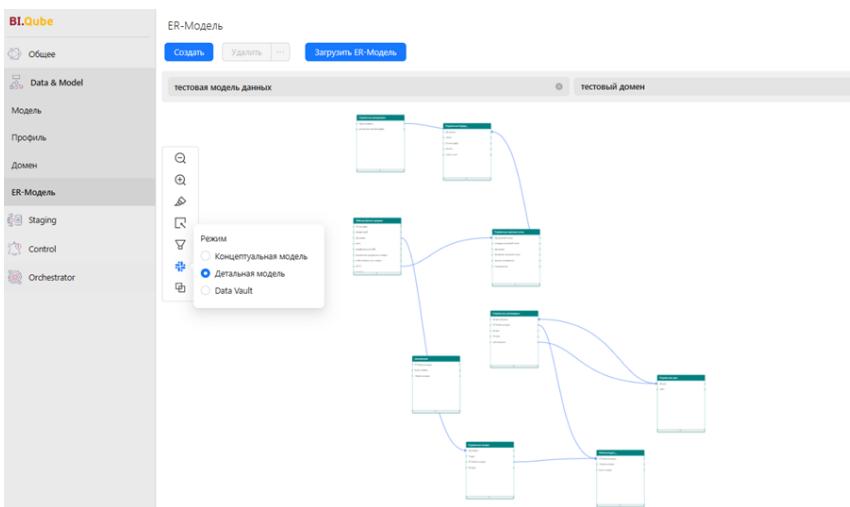


Рисунок. Пример отображения связей в детальной модели

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

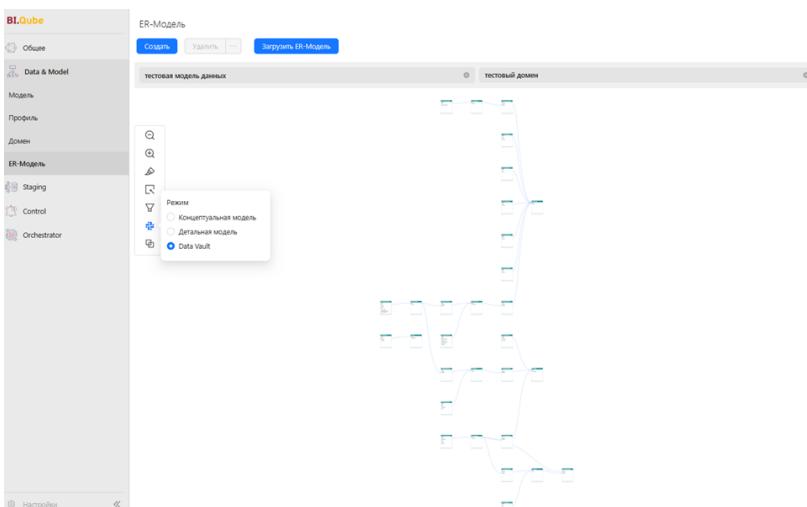


Рисунок. Пример отображения DataVault

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

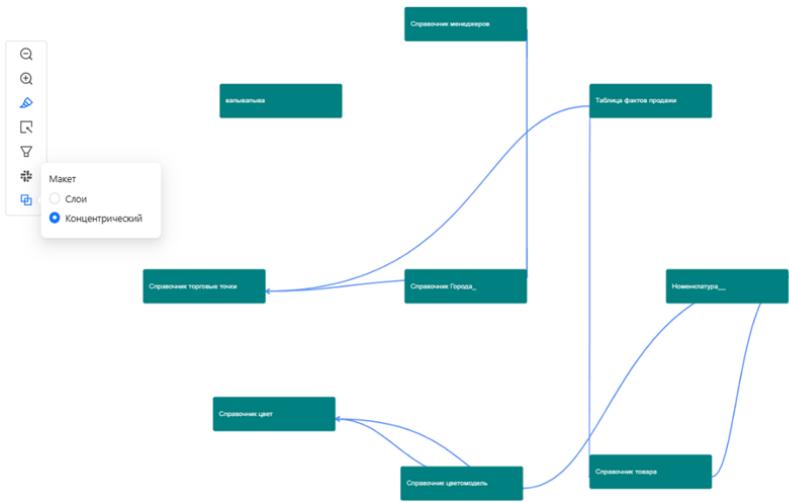


Рисунок. Концентрический макет

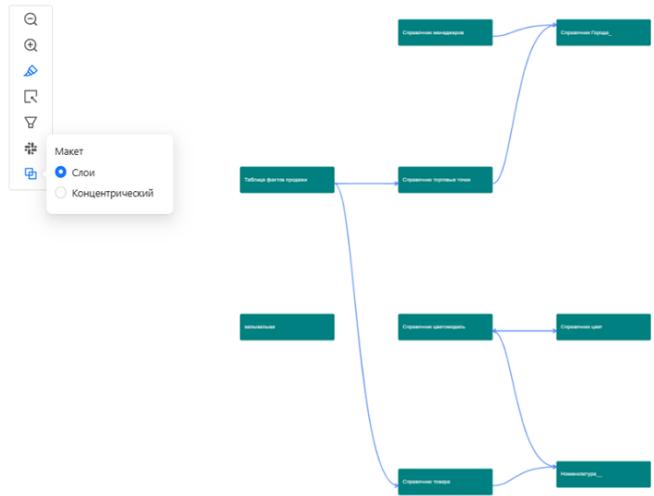


Рисунок. Макет в режиме «слои»

# METACONTROL

Цель компонента MetaControl заключается в своевременном уведомлении пользователей о статусе событий, происходящих с данными на основе правил, создаваемых пользователями.

Уведомления MetaControl отправляются путем рассылки о произведенных контролях с использованием почтового сервера и telegram-канала.

Компонент MetaControl входит в состав системы BI.Qube и может эксплуатироваться как отдельный компонент, так и в составе системы, так и под управлением компонента MetaOrchestrator, в такой конфигурации использование компонента является наиболее эффективной.

# ПРОФИЛЬ МЕТАКОНТРОЛ

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.

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

The screenshot shows the BIQube Control interface. On the left is a sidebar with navigation links: Общие, Data & Model, Staging, Control, Профили, Список рассылок, Проверка, Статус рассылок, and Orchestrator. The main area has a title 'Профили' with buttons 'Создать' (Create) and 'Удалить' (Delete). A search bar says 'Введите строку поиска'. Below is a table with columns 'Название' (Name), 'Описание' (Description), and 'Включен' (Enabled). Two rows are shown: 'EVRAZ' with 'Test1' in Description and 'Enabled' checked, and 'EVRAZ' with 'Test2 (сверка)' in Description and 'Enabled' checked. To the right is a 'Создание' (Creation) form with fields: 'Код' (Code) with placeholder 'Код', 'Название' (Name) with placeholder 'Название', 'Описание' (Description) with placeholder 'Описание', 'Дата создания' (Creation Date) with placeholder 'Дата создания' and a calendar icon, and 'Включен' (Enabled) with a dropdown menu. At the bottom are navigation buttons and a page number '20 / стр.'.

Рисунок. 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.

The screenshot shows the 'Создание' (Creation) form for Profiles. It includes a 'Создать' (Create) button and an 'Очистить' (Clear) button. The form fields are: 'Код' (Code) with placeholder 'Код'; 'Название' (Name) with placeholder 'Название'; 'Описание' (Description) with placeholder 'Описание'; 'Дата создания' (Creation Date) with placeholder 'Дата создания' and a calendar icon; and 'Включен' (Enabled) with a dropdown menu.

Рисунок. Окно свойств для создания Profiles (Профиля)

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

BIQube / Control / Профили

Создать Характеристики

Ведите строку поиска

Начальная дата Конечная дата  Больше информации

Код	Название	Описание	Автор	Дата создания	Включен
37	EVRAZ	Test1		22.01.2024	<input type="checkbox"/>
38	EVRAZ	Test2 (сверка)		14.02.2024	<input checked="" type="checkbox"/>

Настройки <>

1 20 / стр.

**Создание**

Название

Описание

Дата создания

Включен

Рисунок. 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

Имя списка рассылки	Категория	Список email для рассылки	Список для рассылки в телеграмм
EVRAZ	evraz	FrantsevSD@biqube.ru,Fokin_A5@biqube.ru,gol_omazdin_лодарине	
asdasd	asdasd	kva@mail.ru	
est	test	tes@mail.ru	
efd	dfd	tes@mail.ru	

Рисунок. 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 телеграмм) 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. 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.

# СТАТУС 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

Имя профиля	Категория	Отправлено
EVRAZ	false	false
EVRAZ	false	false

Рисунок. 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.

Код	Время	Код теста	Максимальное расхождение	Описание
1	29.05.2023	code 1	maxThreshold 1	test
2	29.05.2023	code 2	maxThreshold 2	test
3	29.05.2023	code 3	maxThreshold 3	test
4	29.05.2023	code 4	maxThreshold 4	test

Рисунок. 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).