



Александр Сербул

Head of Data Science,
ООО «IC-Битрикс»

Apache Superset и свой плагин к Trino

для эффективной и неограниченно
масштабируемой BI-аналитики

О чем поговорим



1

Архитектура REST ВІ интерфейса
Битрикс24 и данных сущностей

Упрощаем и денормализуем

О чем поговорим



1

**Архитектура REST ВІ интерфейса
Битрикс24 и данных сущностей**

Упрощаем и денормализуем

2

**Новые подходы к аналитике
больших данных**

Google Looker Studio, Trino

О чем поговорим



1

**Архитектура REST BI интерфейса
Битрикс24 и данных сущностей**

Упрощаем и денормализуем

2

**Новые подходы к аналитике
больших данных**

Google Looker Studio, Trino

3

**Как написать свой коннектор
к Trino за два дня**

О чем поговорим



1

**Архитектура REST BI интерфейса
Битрикс24 и данных сущностей**

Упрощаем и денормализуем

2

**Новые подходы к аналитике
больших данных**

Google Looker Studio, Trino

3

**Как написать свой коннектор
к Trino за два дня**

4

**Как поднять качественный, бесплатный
и масштабируемый стек для BI-аналитики**

в Apache Superset и Trino для компании любого размера

О чем поговорим



1

**Архитектура REST BI интерфейса
Битрикс24 и данных сущностей**

Упрощаем и денормализуем

2

**Новые подходы к аналитике
больших данных**

Google Looker Studio, Trino

3

**Как написать свой коннектор
к Trino за два дня**

4

**Как поднять качественный, бесплатный
и масштабируемый стек для BI-аналитики**

в Apache Superset и Trino для компании любого размера

5

Как подключить Trino к DBeaver

Коды и описания колонок В1-сущностей Битрикс24

Русскоязычная
документация по ключам
и по полям сущностей

<https://helpdesk.bitrix24.ru/open/16398726/>



- + crm_deal – сделки
- + crm_lead – лиды
- + crm_deal_uf – пользовательские поля сделок
- + crm_lead_uf – пользовательские поля лидов
- + crm_deal_stage_history – история сделок
- + crm_lead_status_history – история лидов
- + telephony_call – звонки
- + user – пользователи
- + crm_company – компании
- + crm_company_uf – пользовательские поля компании
- + crm_contact – контакты
- + crm_contact_uf – пользовательские поля контакта
- + socialnetwork_group – проект
- + crm_activity – дело
- + crm_lead_product_row – товарные позиции в лиде
- + crm_deal_product_row – товарные позиции в сделке
- + crm_dynamic_items – смарт-процесс

Источники данных в маркетплейсе продукта

CRM-аналитика ЗАКАЗАТЬ ВНЕДРЕНИЕ



Подключение Google Looker Studio

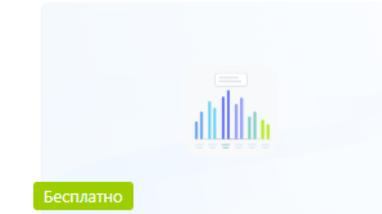
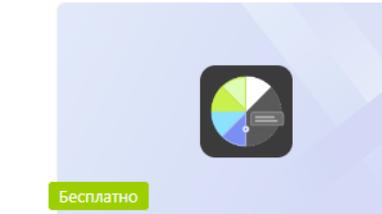
Создавайте наглядные интерактивные отчёты и делитесь ими с коллегами. Превратите данные из вашего Битрикс24 в дашборды, графики и диаграммы с помощью инструментов Google Looker Studio.

[Инструкция по подключению](#)

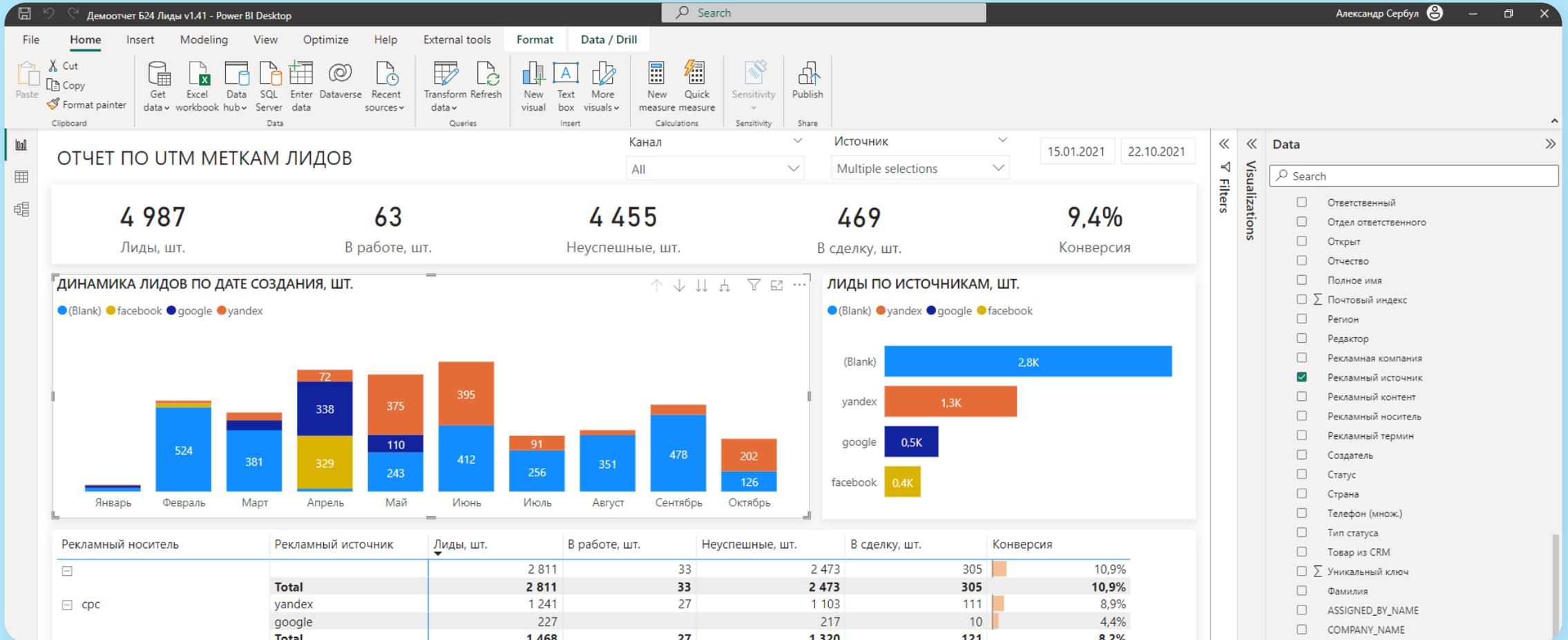
ПОДКЛЮЧИТЬ GOOGLE LOOKER STUDIO

Бесплатно bi-аналитика продажи лиды сделки эффективность звонки дела

Сортировать: Популярные Сначала новые Высокий рейтинг Больше установок

 <p>Бесплатно</p> <p>Универсальный источник данных для Google Looker Studio</p> <p>Вы сможете выбрать нужный источник данных в настройках Google Looker Studio</p> <p>☆☆☆☆☆ 📄 309</p>	 <p>Бесплатно</p> <p>Звонки</p> <p>Звонки CRM в виде источника Google Looker Studio</p> <p>☆☆☆☆☆ 📄 52</p>	 <p>Бесплатно</p> <p>Лиды</p> <p>Лиды в виде источника Google Looker Studio</p> <p>☆☆☆☆☆ 📄 1498</p>	 <p>Бесплатно</p> <p>Сделки</p> <p>Сделки в виде источника Google Looker Studio</p> <p>☆☆☆☆☆ 📄 3011</p>
---	---	---	---

Данные для дашборда



```
bx24_load_entity ←  
  
let  
    //bx24_entity_name: "crm_lead" | "crm_deal"  
    //https://docs.microsoft.com/en-us/powerquery-m/web-contents  
    func = (bx24_entity_name as text) as table =>  
        let  
            response = Web.Contents(  
                "https://" & #"Адрес сервера", ←  
                [  
                    RelativePath = "bitrix/tools/biconnector/pbi.php",  
                    Query = [  
                        //token = BITRIX24_BI_TOKEN, ←  
                        table = bx24_entity_name  
                    ],  
                    Content = Json.FromValue([dateRange = [  
                        startDate = Date.AddYears(Date.From(DateTime.LocalNow()),-5),  
                        endDate = Date.From(DateTime.LocalNow()), ←  
                        key = #"Секретный ключ" ←  
                    ]]  
                )  
            ),  
            jd = Json.Document(response)  
        in  
            Table.FromRows(  
                List.Skip(jd, //data: >=1 row (index 0 based)  
                List.First(jd)//header: 0 row (index 0 based)  
            )  
        in  
            func
```

✓ No syntax errors have been detected.

Функция загрузки JSON-данных

Банально загружаем из портала
Битрикс24 в формате таблицы
JSON сырых данных «Лидов»
или другую BI-сущность.

Вызов функции загрузки JSON-данных

Код BI-сущности

Лиды

```
let
raw_t = bx24_load_entity("crm_lead"),
fixed_t = Table.TransformColumnTypes(raw_t,
{
{"ID", Int64.Type}, //Уникальный ключ
{"DATE_MODIFY", type datetime}, //Время изменения
{"DATE_CREATE", type datetime}, //Время создания
{"CREATED_BY_ID", Int64.Type}, //Идентификатор создателя
{"CREATED_BY", type text}, //Создатель
{"MODIFY_BY_ID", Int64.Type}, //Идентификатор редактора
{"MODIFIED_BY", type text}, //Редактор
{"ASSIGNED_BY_ID", Int64.Type}, //Идентификатор ответственного
{"ASSIGNED_BY", type text}, //Ответственный
{"ASSIGNED_BY_DEPARTMENT", type text}, //Отдел ответственного
{"OPENED", type text}, //Открыт
{"COMPANY_ID", Int64.Type}, //Идентификатор компании
{"COMPANY", type text}, //Компания
{"CONTACT_ID", Int64.Type}, //Идентификатор контакта
{"CONTACT", type text}, //Контакт
{"STATUS_ID", type text}, //Идентификатор статуса в воронке
{"STATUS", type text}, //Статус
{"STATUS_DESCRIPTION", type text}, //Описание статуса
{"STATUS_SEMANTIC_ID", type text}, //Идентификатор типа статуса
{"STATUS_SEMANTIC", type text}, //Тип статуса
{"CRM_PRODUCT", type text}, //Товар из CRM
{"OPPORTUNITY", type number}, //Ожидаемая сумма
{"CURRENCY_ID", type text}, //Валюта
{"SOURCE_ID", type text}, //Идентификатор источника
{"SOURCE", type text}, //Источник
{"SOURCE_DESCRIPTION", type text}, //Описание источника
{"TITLE", type text}, //Заголовок
{"FULL_NAME", type text}, //Полное имя
{"NAME", type text}, //Имя
{"LAST_NAME", type text}, //Фамилия
{"MIDDLE_NAME", type text}, //Отчество
{"COMPANY_TITLE", type text}, //Название компании
{"POST", type text}, //Должность
{"ADDRESS_1", type text}, //Адрес 1
{"ADDRESS_2", type text}, //Адрес 2
{"ADDRESS_CITY", type text}, //Город
{"ADDRESS_POSTAL_CODE", Int64.Type}, //Почтовый индекс
{"ADDRESS_REGION", type text}, //Регион
}
```

Назначаем типы колонкам

Назначаем имена колонкам

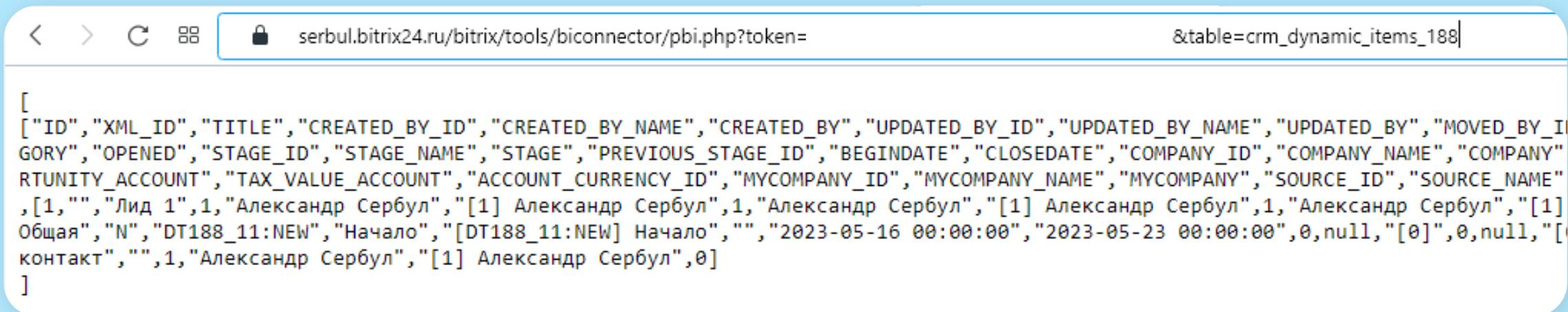
```
),
renamed_t = Table.RenameColumns(
fixed_t,
{
{"ID", "Уникальный ключ"},
{"DATE_MODIFY", "Время изменения"},
{"DATE_CREATE", "Время создания"},
{"CREATED_BY_ID", "Идентификатор создателя"},
{"CREATED_BY", "Создатель"},
{"MODIFY_BY_ID", "Идентификатор редактора"},
{"MODIFIED_BY", "Редактор"},
{"ASSIGNED_BY_ID", "Идентификатор ответственного"},
{"ASSIGNED_BY", "Ответственный"},
{"ASSIGNED_BY_DEPARTMENT", "Отдел ответственного"},
{"OPENED", "Открыт"},
{"COMPANY_ID", "Идентификатор компании"},
{"COMPANY", "Компания"},
{"CONTACT_ID", "Идентификатор контакта"},
{"CONTACT", "Контакт"},
{"STATUS_ID", "Идентификатор статуса в воронке"},
{"STATUS", "Статус"},
{"STATUS_DESCRIPTION", "Описание статуса"},
{"STATUS_SEMANTIC_ID", "Идентификатор типа статуса"},
{"STATUS_SEMANTIC", "Тип статуса"},
{"CRM_PRODUCT", "Товар из CRM"},
{"OPPORTUNITY", "Ожидаемая сумма"},
{"CURRENCY_ID", "Валюта"},
{"SOURCE_ID", "Идентификатор источника"},
{"SOURCE", "Источник"},
{"SOURCE_DESCRIPTION", "Описание источника"},
{"TITLE", "Заголовок"},
{"FULL_NAME", "Полное имя"},
{"NAME", "Имя"},
{"LAST_NAME", "Фамилия"},
{"MIDDLE_NAME", "Отчество"},
{"COMPANY_TITLE", "Название компании"},
{"POST", "Должность"},
{"ADDRESS_1", "Адрес 1"},
{"ADDRESS_2", "Адрес 2"},
{"ADDRESS_CITY", "Город"},
{"ADDRESS_POSTAL_CODE", "Почтовый индекс"},
}
```

✓ No syntax errors have been detected.

Универсальный источник данных для Power BI в маркетплейсе

Данные отдаются BI-коннектором в виде списка списков JSON.

Первая строка списка — список заголовков колонок. Вторая и последующие строки — данные. Данные, как правило, строки. В редких случаях используются числа и другие типы данных.



```
[
  [
    "ID", "XML_ID", "TITLE", "CREATED_BY_ID", "CREATED_BY_NAME", "CREATED_BY", "UPDATED_BY_ID", "UPDATED_BY_NAME", "UPDATED_BY", "MOVED_BY_ID", "MOVED_BY_NAME", "MOVED_BY", "GORY", "OPENED", "STAGE_ID", "STAGE_NAME", "STAGE", "PREVIOUS_STAGE_ID", "BEGINDATE", "CLOSEDATE", "COMPANY_ID", "COMPANY_NAME", "COMPANY_RTUNITY_ACCOUNT", "TAX_VALUE_ACCOUNT", "ACCOUNT_CURRENCY_ID", "MYCOMPANY_ID", "MYCOMPANY_NAME", "MYCOMPANY", "SOURCE_ID", "SOURCE_NAME", "1", "", "Лид 1", 1, "Александр Сербул", "[1] Александр Сербул", 1, "Александр Сербул", "[1] Александр Сербул", 1, "Александр Сербул", "[1] Общая", "N", "DT188_11:NEW", "Начало", "[DT188_11:NEW] Начало", "", "2023-05-16 00:00:00", "2023-05-23 00:00:00", 0, null, "[0]", 0, null, "[1] контакт", "", 1, "Александр Сербул", "[1] Александр Сербул", 0
  ]
]
```

Универсальный источник данных для Power BI в маркетплейсе

The screenshot displays the Power Query Editor interface. The ribbon at the top includes tabs for File, Home, Transform, Add Column, View, Tools, and Help. The main workspace shows a data table with the following columns and data:

ID	XML_ID	TITLE	CREATED_BY_ID	CREATED_BY_NAME	CREATED_BY
1	1	Лид 1	1	Александр Сербул	[1] Александр Сербул

The left-hand pane shows a list of queries under 'Settings [4]' and 'Other Queries [17]'. The 'Other Queries' list includes: crm_dynamic_items, user, crm_contact_uf, crm_company_uf, crm_deal_product_row, crm_lead_product_row, crm_activity, socialnetwork_group, crm_contact, crm_company, and telephony_call. An arrow points from the 'bx24_load_entity' query in the 'Settings' section to the 'crm_dynamic_items' query in the 'Other Queries' section. Another arrow points from the 'crm_dynamic_items' query to the data table in the main workspace.

Протокол работы ВІ-коннектора Битрикс24



Сказать пару слов про изменение трендов в архитектуре Data Warehouses, Data Lakes, Data Lakehouse и т.п. в сторону гетерогенного зондирования данных.

Архитектура протокола

Простая и гибкая интеграция BI REST API с любыми современными BI-системами

Выходной формат – JSON (список-списков).

Первая строка – названия колонок. Остальные - значения колонок.

При потоковой отдаче BI элементов из Битрикс24 агрегации, как правило, не выполняются. Это эффективно потому умеют делать BI-движки.

Поддерживаемые BI REST API операции реляционной алгебры:

- Projection: позволяет выбирать только необходимые колонки BI-сущности, что резко увеличивает скорость запросов.
- Selection: позволяет гибко передавать условия для фильтрации элементов BI-сущности (predicate pushing). Сильно увеличивает скорость запросов.

Есть возможность выбирать колонки для сортировки по дате-времени.

Активно работаем с коннекторами к Trino/Presto и другим хранилищам для Apache Superset, Metabase, Redash.

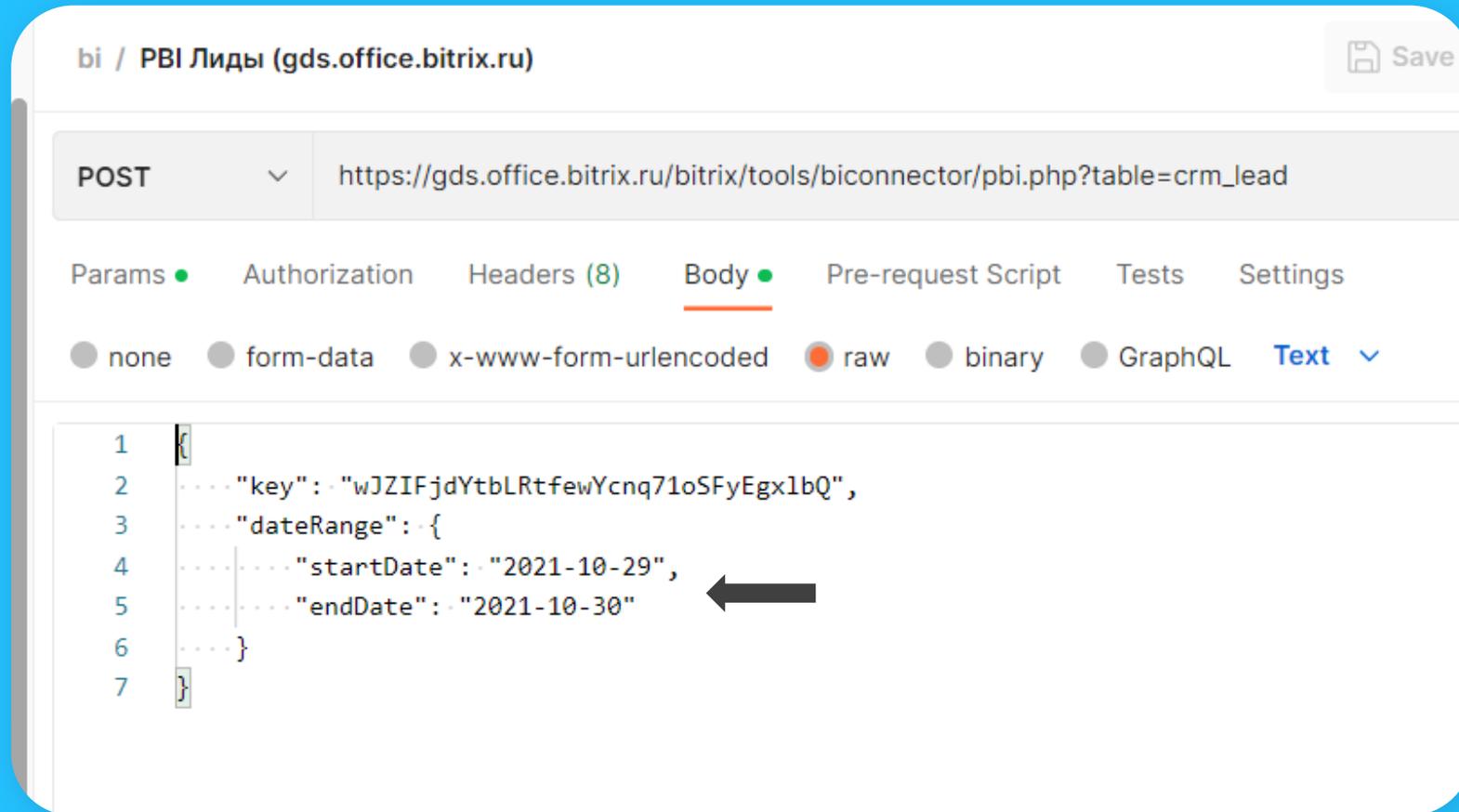
Подробнее про BI REST API Битрикс24:

<https://developers.google.com/looker-studio/connector/reference>



Фильтрация (selection) по вилке дат

Поле даты для фильтрации по умолчанию — первое поле типа дата-время BI-сущности

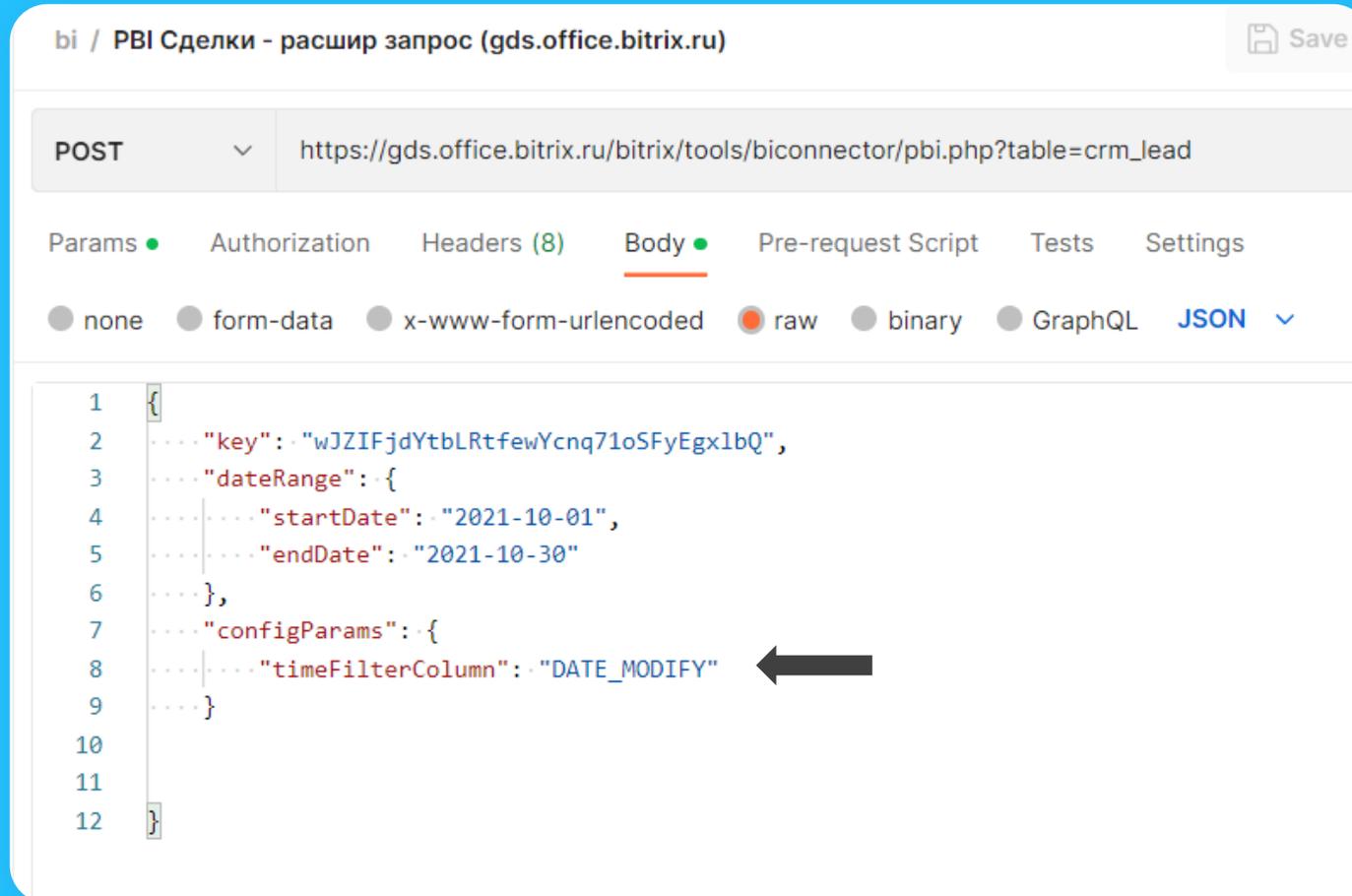


The screenshot shows a REST client interface for a POST request to `https://gds.office.bitrix.ru/bitrix/tools/biconnector/pbi.php?table=crm_lead`. The request body is in raw text format and contains a JSON object with a `dateRange` property. A black arrow points to the `startDate` field within the `dateRange` object.

```
1 {
2   ... "key": "wJZIFjdYtbLRtfewYcnq71oSFyEgx1bQ",
3   ... "dateRange": {
4     ... "startDate": "2021-10-29",
5     ... "endDate": "2021-10-30"
6   ... }
7 }
```

Фильтрация (selection) по вилке дат

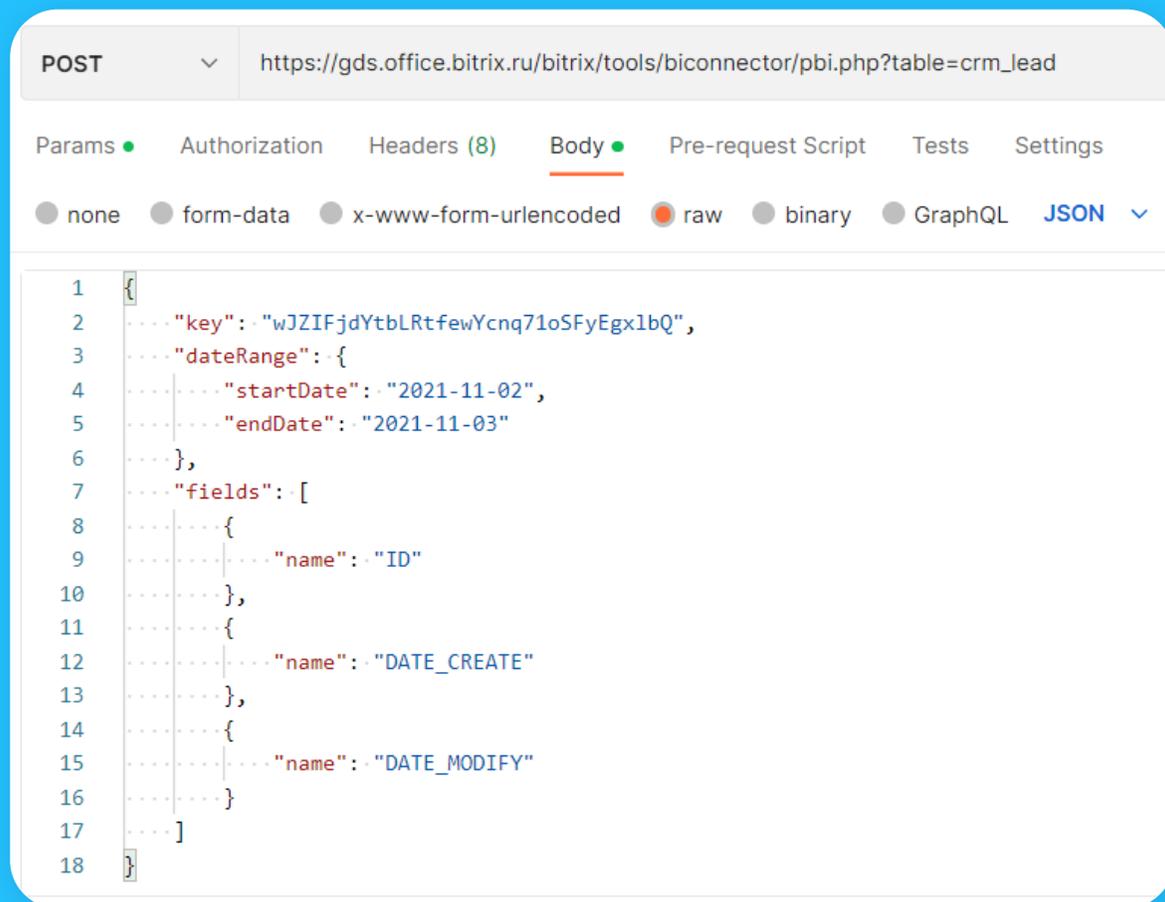
Поле даты для фильтрации можно выбрать.



The screenshot shows a REST client interface for a POST request to `https://gds.office.bitrix.ru/bitrix/tools/biconnector/pbi.php?table=crm_lead`. The request body is in JSON format and includes a `dateRange` object with `startDate` and `endDate` fields, and a `configParams` object with a `timeFilterColumn` field. A black arrow points to the `timeFilterColumn` field, which is set to `"DATE_MODIFY"`.

```
1 {
2   ... "key": "wJZIFjdYtbLRtfewYcnq71oSFyEgxlBQ",
3   ... "dateRange": {
4     ... "startDate": "2021-10-01",
5     ... "endDate": "2021-10-30"
6   },
7   ... "configParams": {
8     ... "timeFilterColumn": "DATE_MODIFY" ←
9   }
10 }
11
12 }
```

Выбор только нужных колонок (projection)



```
POST https://gds.office.bitrix.ru/bitrix/tools/biconnector/pbi.php?table=crm_lead

Params • Authorization Headers (8) Body • Pre-request Script Tests Settings
● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ▾

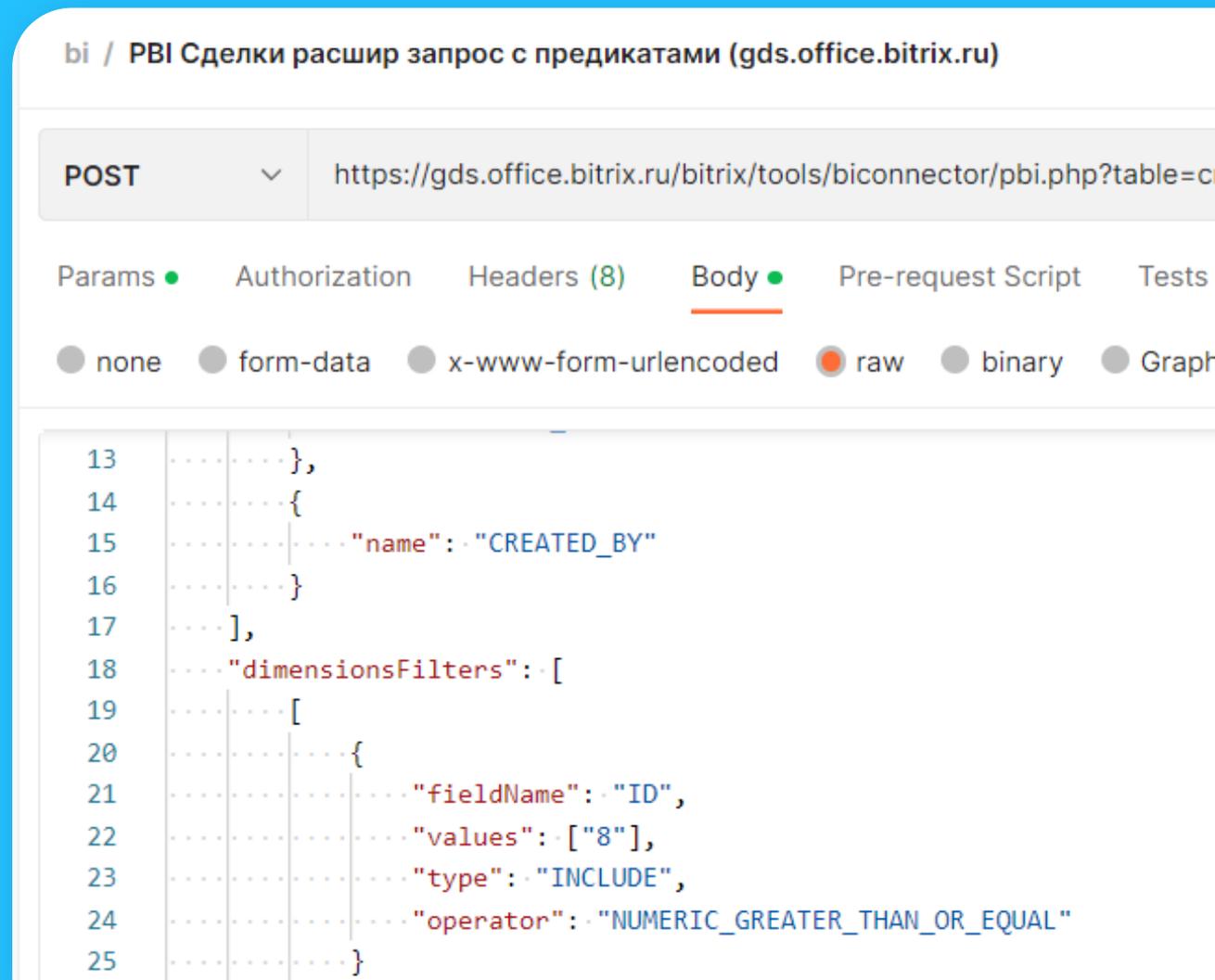
1 {
2   ... "key": "wJZIFjdYtbLRtfewYcnq71oSFyEgx1bQ",
3   ... "dateRange": {
4     ... "startDate": "2021-11-02",
5     ... "endDate": "2021-11-03"
6   },
7   ... "fields": [
8     ... {
9       ... "name": "ID"
10    },
11    ... {
12      ... "name": "DATE_CREATE"
13    },
14    ... {
15      ... "name": "DATE_MODIFY"
16    }
17  ]
18 }
```

Фильтрация (selection) через «predicate pushing»

Поддерживается ограниченный набор предикатов для обеспечения работы по протоколу Google Data Studio.

Но их можно использовать с любыми BI-продуктами: Power BI, Yandex Datalens и т.д.

Подробнее на YouTube-канале.



bi / PBI Сделки расшир запрос с предикатами (gds.office.bitrix.ru)

POST ▼ https://gds.office.bitrix.ru/bitrix/tools/biconnector/pbi.php?table=c

Params ● Authorization Headers (8) Body ● Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary Graph

```
13 ..... },
14 ..... {
15 .....     "name": "CREATED_BY"
16 ..... }
17 ..... ],
18 ..... "dimensionsFilters": [
19 ..... [
20 .....     {
21 .....         "fieldName": "ID",
22 .....         "values": ["8"],
23 .....         "type": "INCLUDE",
24 .....         "operator": "NUMERIC_GREATER_THAN_OR_EQUAL"
25 .....     }
```

Собственный плагин для Trino (PrestoSQL)



History of Trino

2008

Hive
(Facebook)

2013

Presto
(Facebook)

2015

Presto in EMR
(Amazon Web
Services)

2016

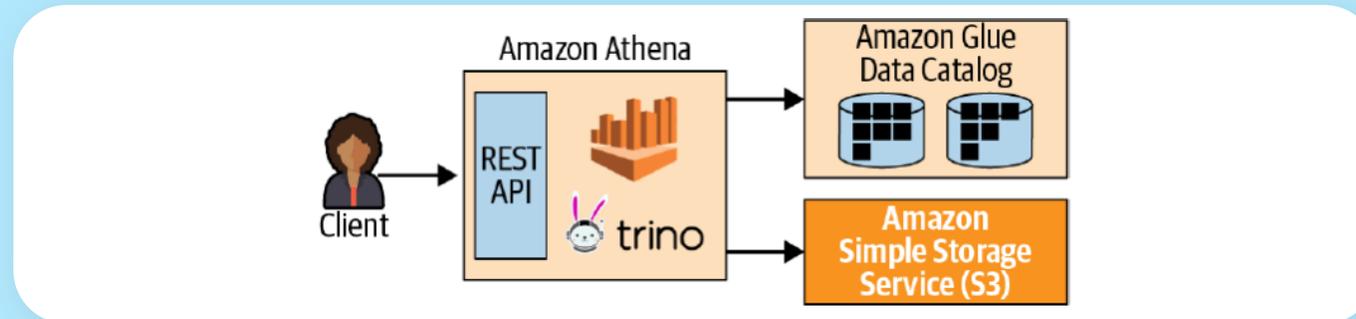
Athena/Presto
(Amazon Web
Services)

2019

Presto SQL Presto
DB

2020

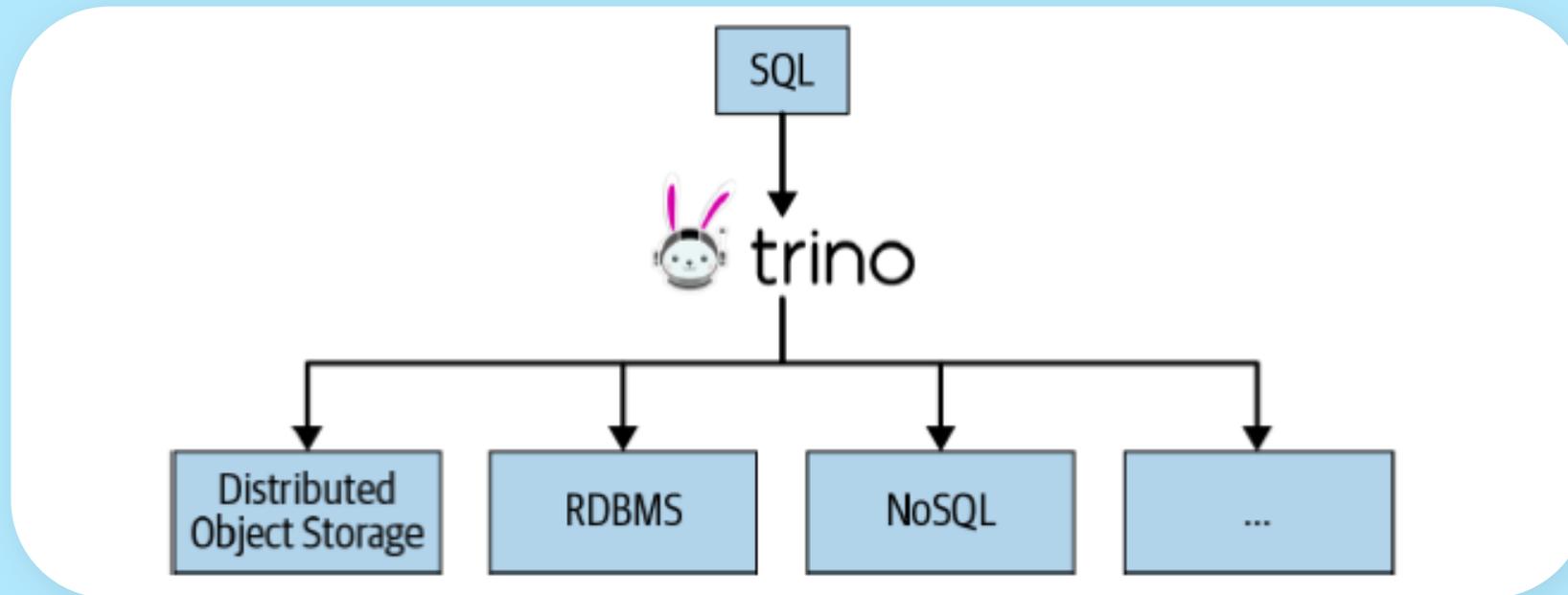
Trino



«In January 2019, the original creators of Presto, Martin Traverso, Dain Sundstrom, and David Phillips, created a fork of the Presto project. They initially kept the name Presto and used the PrestoSQL web handle to distinguish it from the original PrestoDB project. Simultaneously, they announced the Presto Software Foundation. The foundation is a not-for-profit organization dedicated to the advancement of the Presto open source distributed SQL query engine.[7][8]

In December 2020, PrestoSQL was rebranded as Trino. The Trino Software Foundation, code base, and all other PrestoSQL assets were renamed as part of the rebrand.[9]»

Архитектура Trino — «SQL on anything»



Документация к Trino



Trino 418 Documentation

Trino 418 Documentation

- Overview
- Installation
- Clients
- Security
- Administration
- Event listeners
- Query optimizer
- Connectors
- Functions and operators
- SQL language
- SQL statement syntax
- Developer guide
- Glossary
- Appendix
- Release notes

Trino documentation

Overview

[Use cases](#)

[Trino concepts](#)

Installation

[Deploying Trino](#)

[Trino in a Docker container](#)

[Trino on Kubernetes with Helm](#)

[RPM package](#)

[Improve query processing resilience](#)

Clients

[Command line interface](#)

[JDBC driver](#)

Security

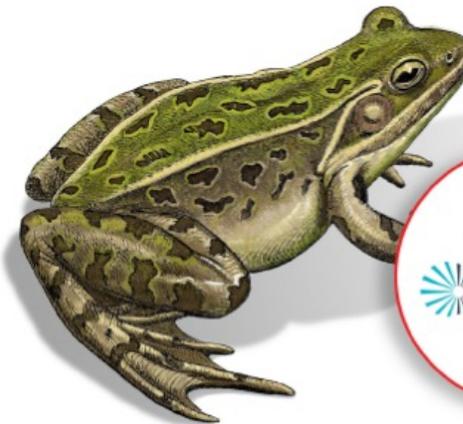
O'REILLY®

Second
Edition

Trino

The Definitive Guide

SQL at Any Scale, on Any Storage,
in Any Environment



Compliments of

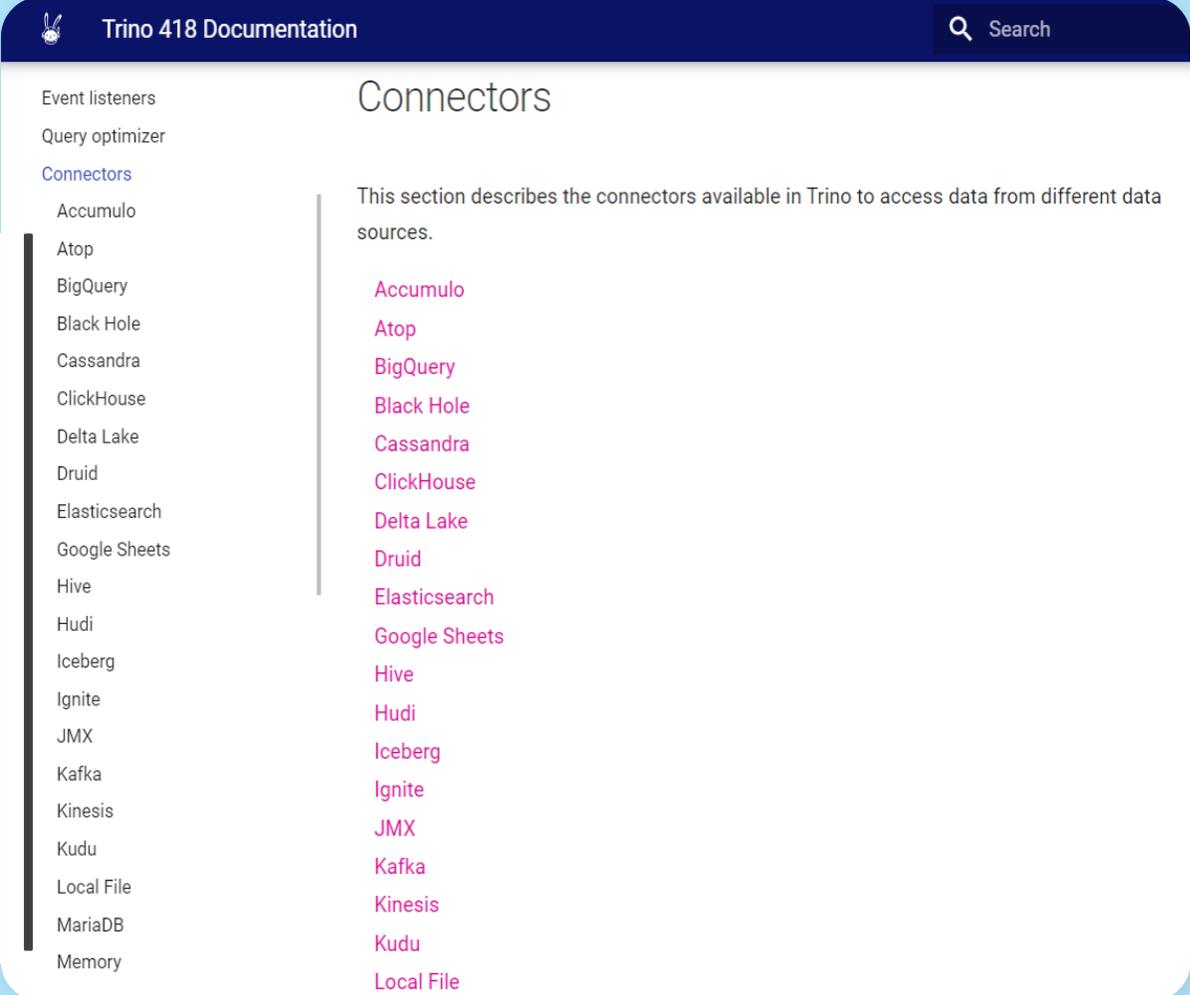


Starburst

Matt Fuller, Manfred Moser
& Martin Traverso

Коннекторы в Trino

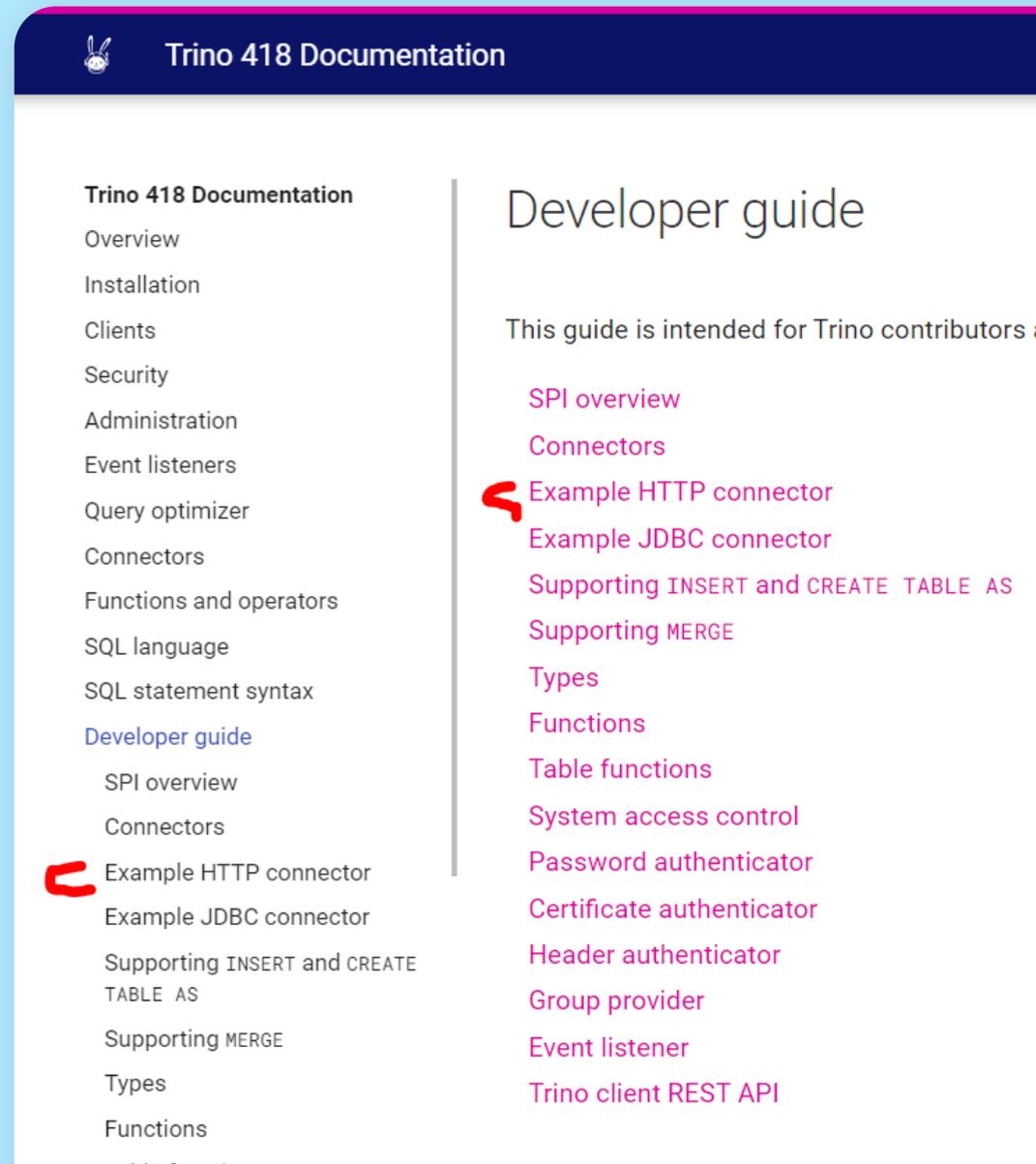
- Из коробки поддерживаются самые популярные хранилища
- Коннекторы лежат в понятных исходниках на github



The screenshot shows the Trino 418 Documentation website. The page title is "Trino 418 Documentation" and the main heading is "Connectors". A search bar is visible in the top right corner. The left sidebar contains a list of navigation items: Event listeners, Query optimizer, Connectors (highlighted), Accumulo, Atop, BigQuery, Black Hole, Cassandra, ClickHouse, Delta Lake, Druid, Elasticsearch, Google Sheets, Hive, Hudi, Iceberg, Ignite, JMX, Kafka, Kinesis, Kudu, Local File, MariaDB, and Memory. The main content area contains the text: "This section describes the connectors available in Trino to access data from different data sources." Below this text is a list of connector names: Accumulo, Atop, BigQuery, Black Hole, Cassandra, ClickHouse, Delta Lake, Druid, Elasticsearch, Google Sheets, Hive, Hudi, Iceberg, Ignite, JMX, Kafka, Kinesis, Kudu, and Local File.

Демонстрационный HTTP-коннектор

- Документации — не хватает
- Выручают — понятные исходники
- От простого — к сложному



The screenshot shows the Trino 418 Documentation website. The top navigation bar is dark blue with a white rabbit icon and the text "Trino 418 Documentation". Below the navigation bar is a white content area. On the left, there is a vertical navigation menu with the following items: "Trino 418 Documentation", "Overview", "Installation", "Clients", "Security", "Administration", "Event listeners", "Query optimizer", "Connectors", "Functions and operators", "SQL language", "SQL statement syntax", "Developer guide", "SPI overview", "Connectors", "Example HTTP connector", "Example JDBC connector", "Supporting INSERT and CREATE TABLE AS", "Supporting MERGE", "Types", and "Functions". The "Example HTTP connector" item is highlighted with a red bracket. On the right, the "Developer guide" page is displayed, with the text "This guide is intended for Trino contributors". Below this text is a list of links: "SPI overview", "Connectors", "Example HTTP connector", "Example JDBC connector", "Supporting INSERT and CREATE TABLE AS", "Supporting MERGE", "Types", "Functions", "Table functions", "System access control", "Password authenticator", "Certificate authenticator", "Header authenticator", "Group provider", "Event listener", and "Trino client REST API". The "Example HTTP connector" link is highlighted with a red bracket.

Trino 418 Documentation

Trino 418 Documentation

Overview

Installation

Clients

Security

Administration

Event listeners

Query optimizer

Connectors

Functions and operators

SQL language

SQL statement syntax

Developer guide

SPI overview

Connectors

Example HTTP connector

Example JDBC connector

Supporting INSERT and CREATE TABLE AS

Supporting MERGE

Types

Functions

Developer guide

This guide is intended for Trino contributors

SPI overview

Connectors

Example HTTP connector

Example JDBC connector

Supporting INSERT and CREATE TABLE AS

Supporting MERGE

Types

Functions

Table functions

System access control

Password authenticator

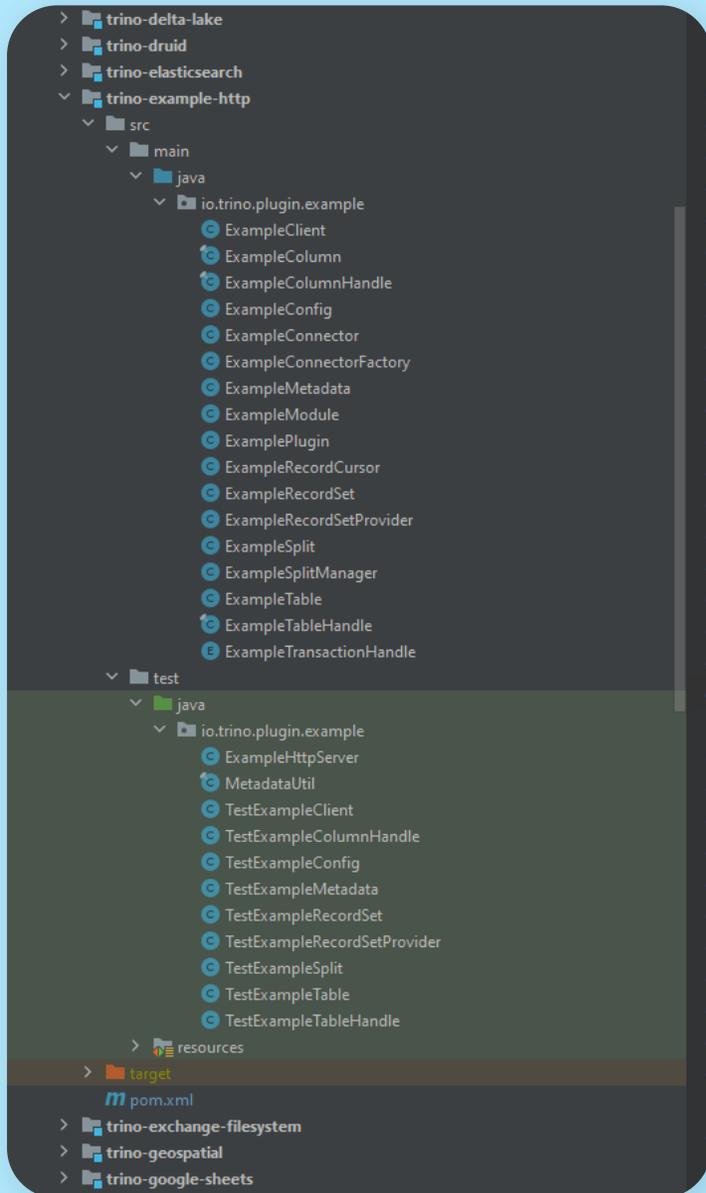
Certificate authenticator

Header authenticator

Group provider

Event listener

Trino client REST API

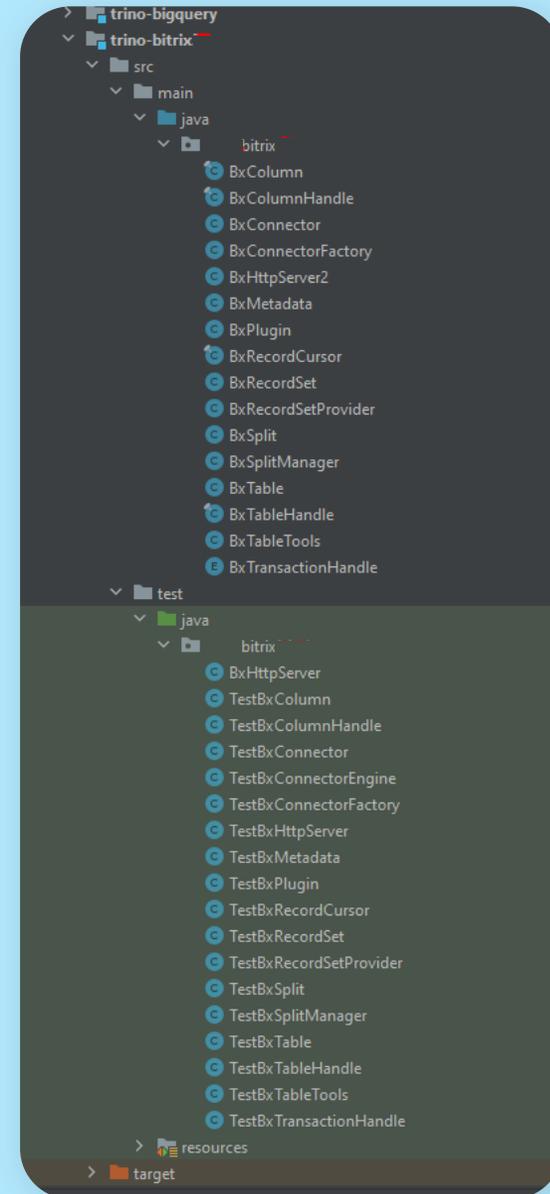


Демонстрационный HTTP-коннектор — структура файлов

- Пара десятков простых классов
- Убираем «аннотации», другую магию и упрощаем 😊
- Обязательно покрываем все тестами как можно полнее: unit и интеграционными с движком Trino

Демонстрационный HTTP-коннектор — структура файлов

- Пара десятков простых классов
- Убираем «аннотации», другую магию и упрощаем 😊
- Обязательно покрываем все тестами как можно полнее: unit и интеграционными с движком Trino



Свой коннектор — структура «БД» и данные

```
@Override
public boolean getBoolean(final int field)
{
    checkArgument( expression: field >= 0 && field < columnHandlesSize, errorMessage: "Invalid field index");
    return (Boolean) getFieldValue(field);
}

@Override
public long getLong(final int field)
{
    checkArgument( expression: field >= 0 && field < columnHandlesSize, errorMessage: "Invalid field index");
    return (Long) getFieldValue(field);
}

@Override
public double getDouble(final int field)
{
    checkArgument( expression: field >= 0 && field < columnHandlesSize, errorMessage: "Invalid field index");
    return (Double) getFieldValue(field);
}

@Override
public Slice getSlice(final int field)
{
    checkArgument( expression: field >= 0 && field < columnHandlesSize, errorMessage: "Invalid field index");
    return Slices.utf8Slice((String) getFieldValue(field));
}
```

1. Расширяем ConnectorMetadata

2. Переопределяем:

- listSchemaNames,
- listTables
- getColumnMetadata
- getTableMetadata
- listTableColumns

1. Расширяем/impleментируем

- ConnectorRecordSetProvider
- RecordSet
- RecordCursor

Свой коннектор — интеграционные тесты

- Через движок тестируем:
- список таблиц с тестового внутреннего http-сервера
- описание таблицы
- данные таблицы
- SQL запросы к данным: выборки, агрегации, сортировки и т.п.
- запросы к удаленным тестовым серверам

```
no usages
780 @Test
781 public void testQueryConfigColTypesDateTime()
782     throws Exception
783     {
784     final Session lsession = testSessionBuilder()
785         .setCatalog(testCatalogName)
786         .setSchema(BxMetadata.SCHEMA)
787         .build();
788
789     try (final DistributedQueryRunner lqueryRunner = DistributedQueryRunner.builder(lsession).build()) {
790         final String restUrl = bxHttpServer.resolve("/").toString();
791
792         Map<String, String> catProps = Map.of(
793             BxConnectorFactory.CNFSERVER, restUrl,
794             BxConnectorFactory.CNFSKEY, v2: "1",
795             k3: "crm_lead.ID", v3: "integer",
796             k4: "crm_lead.DATE_MODIFY", v4: "timestamp",
797             k5: "crm_lead.CREATED_BY", v5: "bigint",
798             k6: "crm_lead.TITLE", v6: "varchar"
799         ); //any will work
800
801         lqueryRunner.installPlugin(new BxPlugin());
802
803         lqueryRunner.createCatalog(testCatalogName, BxConnectorFactory.PLNAME, catProps);
804
805         List<MaterializedRow> matRows = lqueryRunner.getClient().execute(sql: "SHOW CREATE TABLE crm_lead")
806             .getResult().getMaterializedRows();
807
808         assertTrue(matRows.get(0).getField(0).toString().contains("id integer"));
809         assertTrue(matRows.get(0).getField(0).toString().contains("date_modify timestamp")); //timestamp(0)
810         assertTrue(matRows.get(0).getField(0).toString().contains("created_by bigint"));
811         assertTrue(matRows.get(0).getField(0).toString().contains("title varchar"));
812
813         matRows = lqueryRunner.getClient().execute(sql: "SELECT date_modify FROM crm_lead ORDER BY date_modify DESC")
```

Свой коннектор — Dockerfile

```
1 FROM maven:3.9-eclipse-temurin-17
2
3 RUN apt-get update && apt-get install -y mercurial && apt-get install -y git
4
5 RUN mkdir -p /opt/plugin
6 WORKDIR /opt/plugin
7
8
9 RUN git clone -b 396 --single-branch https://github.com/trinodb/trino.git trino-396
10 RUN mkdir trino-396/plugin/trino-bitrix24
11 WORKDIR trino-396/plugin/trino-bitrix24
12 RUN hg init
13
14 CMD bash
15
16 #docker build -t trino_builder .
17 #docker run -it trino_builder
18 #hg pull -u https://a.serbul:\*\*\*@bitrix24.com trino_b24bi_plugin
19 #mvn test -DexcludedGroups=remote
20 #mvn install -DskipTests=true
21 #ls -al target/
22 #trino-bitrix24-396.zip <-- built plugin, take it!
```

BI-аналитика в Open Source продуктах с Trino



BI-лаборатория: экспериментальный плагин к Trino (PrestoSQL) — конфиг

- 1 Скачивается бесплатный сервер Trino.
- 2 В папку конфигов сервера кладется конфиг.
- 3 В папку плагинов распаковывается кастомный плагин.

trino-bitrix24-396.zip

20.04.2023 18:34

Сжатая ZIP-папка

16 320 КБ

Портал Битрикс24
и BI-токен.

Время кэширования
списка сущностей,
колонок сущности
и данных сущности.

Типы полей для каждой
BI-сущности Битрикс24.
Если не указать,
используется
дефолтный тип строка.
Если парсинг
неуспешный – будет
NULL в значении.

```
connector.name=bitrix24-bi

server_url=https://qds.office.bi
secret_key=wJZIFjdYtbLRtfewYcnq7

cacheListBiTablesMills=3600000
cacheDescribeBiTableMills=3600000
cacheBiTableDataMills=3600000

crm_lead.ID=bigint
crm_lead.DATE_MODIFY=timestamp
crm_lead.DATE_CREATE=timestamp
crm_lead.CREATED_BY_ID=bigint
crm_lead.CREATED_BY=varchar
crm_lead.MODIFY_BY_ID=bigint
crm_lead.MODIFIED_BY=varchar
crm_lead.ASSIGNED_BY_ID=bigint
crm_lead.ASSIGNED_BY=varchar
crm_lead.ASSIGNED_BY_DEPARTMENT=
crm_lead.OPENED=varchar
crm_lead.COMPANY_ID=bigint
crm_lead.COMPANY=varchar
crm_lead.CONTACT_ID=bigint
crm_lead.CONTACT=varchar
crm_lead.STATUS_ID=varchar
crm_lead.STATUS=varchar
crm_lead.STATUS_DESCRIPTION=varc
crm_lead.STATUS_SEMANTIC_ID=varc
crm_lead.STATUS_SEMANTIC=varchar
crm_lead.CRM_PRODUCT=varchar
```

BI-лаборатория: экспериментальный плагин к Trino (PrestoSQL) — конфиг

Каждая BI-сущность
Битрикс24 видна как
отдельная SQL-таблица.



Можно делать аналитические
SQL-запросы к BI-таблицам,
соединять их как угодно.



```
trino:bitrix24> use bi.bitrix24;
USE
trino:bitrix24> show tables;
Table
-----
crm_activity
crm_company
crm_contact
crm_deal
crm_deal_product_row
crm_deal_stage_history
crm_deal_uf
crm_lead
crm_lead_product_row
crm_lead_status_history
crm_lead_uf
socialnetwork_group
telephony_call
test
(14 rows)

Query 20221102_161133_00020_zwuvr, FINISHED, 1 node
Splits: 4 total, 4 done (100.00%)
0.24 [14 rows, 446B] [59 rows/s, 1.85KB/s]

trino:bitrix24> select * from crm_lead_product_row;
id | lead_id | lead_date_modify | lead_date_create | lead_date_closed |
-----+-----+-----+-----+-----+
 1 |      1 | 2022-02-08 12:14:32 | 2021-09-13 07:25:53 | NULL | [2
 8 |      2 | 2022-08-09 15:34:03 | 2021-09-13 07:25:54 | NULL | [4
 9 |      2 | 2022-08-09 15:34:03 | 2021-09-13 07:25:54 | NULL | [4
(3 rows)
```

BI-лаборатория: экспериментальный плагин к Trino (PrestoSQL) — конфиг

Каждая BI-сущность имеет разный набор колонок разных типов. Типы можно менять в конфиге.

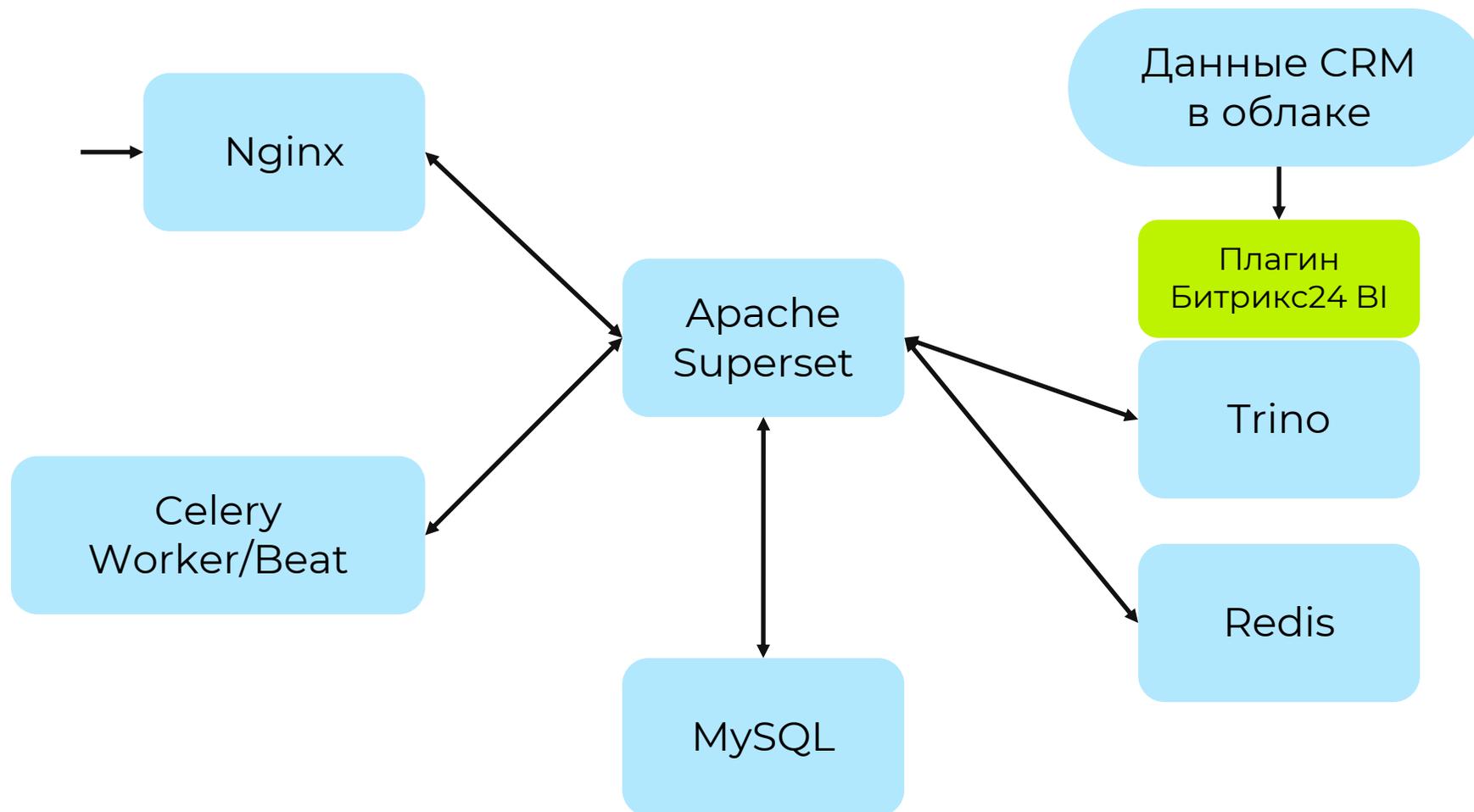


```
trino:bitrix24> show create table crm_lead_uf;
Create Table
-----
CREATE TABLE bi.bitrix24.crm_lead_uf (
  lead_id bigint,
  date_modify timestamp(0),
  date_create timestamp(0),
  date_closed timestamp(0),
  uf_crm_1638790150858 varchar,
  uf_crm_1638790177370 varchar,
  uf_crm_1638868487702 varchar,
  uf_crm_1639131105359 varchar,
  uf_crm_1639131129521 varchar,
  uf_crm_1639556890415 varchar,
  uf_crm_1639556926473 varchar,
  uf_crm_1639557709875 varchar,
  uf_crm_1639558003990 varchar,
  uf_crm_1639558482832 varchar,
  uf_crm_1639558559731 varchar,
  uf_crm_1639559566867 varchar,
  uf_crm_1639560168337 varchar,
  uf_crm_1639560187915 varchar,
  uf_crm_1639560310067 varchar,
  uf_crm_1639560328397 varchar,
  uf_crm_1639560431748 varchar,
  uf_crm_1639560447029 varchar,
  uf_crm_1639561771519 varchar,
  uf_crm_1639561813989 varchar,
  uf_crm_1639562516437 varchar,
  uf_crm_1639562682658 varchar,
  uf_crm_1639562707157 varchar,
  uf_crm_16395628333 varchar
)
(1 row)

Query 20221102_132430_00003_2eshn, FINISHED, 1 node
Splits: 1 total, 1 done (100.00%)
0.70 [0 rows, 0B] [0 rows/s, 0B/s]

trino:bitrix24> select * from crm_lead_uf limit 1;
lead_id | date_modify | date create
-----
1 | 2022-02-08 12:14:32 | 2021-09-13 07:25:53
(1 row)
```

Архитектура контейнеров для BI



Docker-compose c Apache Superset + Trino

Containers [Give feedback](#)

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)



Only show running containers

<input type="checkbox"/>	Name	Image	Status	Port(s)
<input type="checkbox"/>	fullfunct2products	-	Running (7/7)	
<input type="checkbox"/>	ss_celery_worker-1 f39fb839471c	fullfunct2products-ss_celery_worker	Running	
<input type="checkbox"/>	ss_nginx-1 a77fd5fe6845	fullfunct2products-ss_nginx	Running	1443:1443
<input type="checkbox"/>	ss_celery_beat-1 95972c0ed44e	fullfunct2products-ss_celery_beat	Running	
<input type="checkbox"/>	ss_superset-1 68b29f2681fa	fullfunct2products-ss_superset	Running	
<input type="checkbox"/>	ss_trino-1 f3ab546a7574	fullfunct2products-ss_trino	Running	
<input type="checkbox"/>	ss_redis-1 102c4813c14d	fullfunct2products-ss_redis	Running	
<input type="checkbox"/>	ss_mysql-1 b92396ea5970	fullfunct2products-ss_mysql	Running	

Nginx- настройки для Trino

```
1 #trino
2 server {
3     listen      443 ssl http2;
4     listen      [::]:443 ssl http2;
5     server_name _;
6
7     ssl_certificate "/etc/nginx/nginx-selfsigned.crt";
8     ssl_certificate_key "/etc/nginx/nginx-selfsigned.key";
9     ssl_session_cache shared:SSL:1m;
10    ssl_session_timeout 10m;
11    ssl_ciphers HIGH:!aNULL:!MD5;
12    ssl_prefer_server_ciphers on;
13
14    location / {
15        proxy_pass http://ss_trino:8080; # container "ss_trino"
16        proxy_redirect http:// https://;
17        proxy_set_header X-Forwarded-Proto https;
18        proxy_set_header Host $host;
19        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
20        proxy_set_header X-Real-IP $remote_addr;
21    }
22
23 }
```

Nginx – настройки для Apache Superset

```
1 #superset
2 server {
3     listen      1443 ssl http2;
4     listen      [::]:1443 ssl http2;
5     server_name  _;
6
7     ssl_certificate "/etc/nginx/nginx-selfsigned.crt";
8     ssl_certificate_key "/etc/nginx/nginx-selfsigned.key";
9     ssl_session_cache shared:SSL:1m;
10    ssl_session_timeout 10m;
11    ssl_ciphers HIGH:!aNULL:!MD5;
12    ssl_prefer_server_ciphers on;
13
14    location / {
15        proxy_pass http://ss_superset:8088; # container "superset"
16        proxy_set_header X-Forwarded-Proto https;
17        proxy_set_header X-Forwarded-Port 1443;
18
19        proxy_set_header Host $host;
20        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
21        proxy_set_header X-Real-IP $remote_addr;
22    }
23
```

Dockerfile Trino

```
1 FROM alpine:latest as unzipper
2
3 RUN apk add --no-cache zip
4 RUN mkdir -p /opt
5 WORKDIR /opt
6 COPY trino-bitrix24-396.zip .
7 RUN unzip trino-bitrix24-396.zip
8 RUN rm -f trino-bitrix24-396.zip
9
10 FROM trinodb/trino:396
11
12 RUN mkdir -p /usr/lib/trino/plugin/trino-bitrix24-396
13 COPY --from=unzipper /opt/ /usr/lib/trino/plugin/
14
15 WORKDIR /etc/trino
16
17 RUN rm -f catalog/*
18
19 #ignored in container, also set in compose, here for docs only
20 COPY etc/security/limits.d/trino.conf /etc/security/limits.d/trino.conf
21
22 COPY catalog/bi.properties catalog/bi.properties
23 COPY jvm.config jvm.config
24 COPY config.properties config.properties
25 COPY log.properties log.properties
26 COPY node.properties node.properties
27 COPY access-control.properties access-control.properties
28 COPY password-authenticator.properties password-authenticator.properties
29 COPY rules.json rules.json
30 COPY password.db password.db
31
32 EXPOSE 8080
```

Конфигурации сервера Trino – config.properties

```
1 coordinator=true
2 node-scheduler.include-coordinator=true
3 http-server.http.port=8080
4 discovery.uri=http://localhost:8080
5
6 http-server.process-forwarded=true
7 http-server.authentication.type=PASSWORD
8 internal-communication.shared-secret=
9
```

Конфигурации сервера Trino — безопасность, multitenancy

access-control.properties

```
1 access-control.name=file
2 security.config-file=etc/rules.json
3 security.refresh-period=10s
```

password-authenticator.properties

```
1 password-authenticator.name=file
2 file.password-file=etc/password.db
3
```

rules.json

```
1 {
2
3   "queries": [
4     {
5       "user": "admin",
6       "allow": ["execute", "kill", "view"]
7     },
8     {
9       "allow": ["execute", "kill"]
10    }
11  ],
12
13  "system_information": [
14    {
15      "user": "admin",
16      "allow": ["read", "write"]
17    }
18  ]
19 }
```

```
20 "tables": [
21
22   {
23     "user": "admin",
24     "privileges": ["SELECT", "INSERT", "DELETE", "UPDATE", "OWNERSHIP"]
25   },
26
27   {
28     "catalog": "bi",
29     "privileges": ["SELECT"]
30   },
31
32   {
33     "catalog": "system",
34     "schema": "runtime",
35     "table": "queries",
36     "privileges": ["SELECT"]
37   },
38
39   {
40     "catalog": "system",
41     "schema": "metadata",
42     "privileges": ["SELECT"]
43   }
44 ]
```

Консольная настройка Apache Superset с Trino

```
1  #!/bin/bash
2
3  sleep 5
4  superset fab create-admin --username admin --firstname Superset --lastname Admin \
5      --email admin@superset.com --password admin
6
7  sleep 5
8  superset db upgrade
9
10 sleep 5
11 superset init
12
13 ### Add default database, change portal and bi-token here
14 sleep 5
15 superset set-database-uri -d 'client_demo' \
16     -u 'trino://client_demo:wJZIFjdYtbLRtfewYcnq71oSfYegxlbQ@ss_nginx:443/bi?http_scheme=https&
17     session_properties=%7B%22bi.server_url%22%3A%22https%3A%2F%2Fgds.office.bitrix.ru%22%2C+++%22
18     bi.secret_key%22%3A%22wJZIFjdYtbLRtfewYcnq71oSfYegxlbQ%22%7D&source=client_demo&verify=false'
19
20 sleep 5
21 superset import-dashboards -u admin -p leads.json
22
23 sleep 5
24 superset import-dashboards -u admin -p deals.json
25
```

Dockerfile Apache Superset

```
1 FROM apache/superset:2.1.0
2 #2.1.0, 2.0.1, 1.5.3
3
4 USER superset
5
6 RUN pip install trino
7 RUN pip install mysql-connector-python
8
9 WORKDIR "/app"
10 COPY superset_config.py pythonpath/superset_config.py
11 COPY bx_init.sh bx_init.sh
12
13 COPY demo_dashbs/leads.json leads.json
14 COPY demo_dashbs/deals.json deals.json
15
16 USER root
17 RUN chmod a+x bx_init.sh
18
19 USER superset
```

superset_config.py — ключевые части

```
2 SECRET_KEY = '!!!CHANGE_ME_ON_PROD!!!'
3
4 ### Custom metadata backend
5 SQLALCHEMY_DATABASE_URI = "mysql://root:123456@ss_mysql/superset"
6
7 FEATURE_FLAGS = {
8     'ENABLE_TEMPLATE_PROCESSING': True,
9     "ENABLE_EXPLORE_DRAG_AND_DROP": True,
10    "DASHBOARD_CROSS_FILTERS": True,
11    "EMBEDDED_SUPERSET": True
12 }
13
14 #JINJA_CONTEXT_ADDONS = {
15 #     'portal_url': 'https://someportal.bitrix24.ru',
16 #     'user_detail_url': '/company/personal/user/'
17 # }
```

```
116 ENABLE_PROXY_FIX = True
117
118 GUEST_TOKEN_JWT_EXP_SECONDS = 864000000
119 GUEST_ROLE_NAME = "Gamma"
120 GUEST_TOKEN_JWT_SECRET = "!!!CHANGE_ME_ON_PROD!!!"
121 JWT_ACCESS_TOKEN_EXPIRES = 86400
```

superset_config.py — ключевые части

```
20 FILTER_STATE_CACHE_CONFIG = {
21     "CACHE_TYPE": "RedisCache",
22     "CACHE_KEY_PREFIX": "superset_filter_state", # make sure
23     "CACHE_DEFAULT_TIMEOUT": 86400, # 60 seconds * 60
24     "CACHE_REDIS_DB": 0,
25     "CACHE_REDIS_HOST": "ss_redis"
26 }
27
28 #
29 EXPLORE_FORM_DATA_CACHE_CONFIG = {
30     "CACHE_TYPE": "RedisCache",
31     "CACHE_KEY_PREFIX": "superset_explore_form_data",
32     "CACHE_DEFAULT_TIMEOUT": 86400, # 60 seconds * 60
33     "CACHE_REDIS_DB": 0,
34     "CACHE_REDIS_HOST": "ss_redis"
35 }
36
37 #
38 DATA_CACHE_CONFIG = {
39     "CACHE_TYPE": "RedisCache",
40     "CACHE_KEY_PREFIX": "superset_data", # make sure
41     "CACHE_DEFAULT_TIMEOUT": 86400, # 60 seconds * 60
42     "CACHE_REDIS_DB": 0,
43     "CACHE_REDIS_HOST": "ss_redis"
44 }
45
46 #
47 CACHE_CONFIG = {
48     "CACHE_TYPE": "RedisCache",
49     "CACHE_KEY_PREFIX": "superset_cache", # make sure
50     "CACHE_DEFAULT_TIMEOUT": 86400, # 60 seconds * 60
51     "CACHE_REDIS_DB": 0,
52     "CACHE_REDIS_HOST": "ss_redis"
53 }
```

```
55 ### Async jobs
56 from cachelib.redis import RedisCache
57
58 RESULTS_BACKEND = RedisCache(
59     host='ss_redis', port=6379, key_prefix='superset_results', db=0)
60
61
62 from celery.schedules import crontab
63
64 class CeleryConfig(object):
65     broker_url = 'redis://ss_redis:6379/0'
66     imports = (
67         'superset.sql_lab'
68     )
69     result_backend = 'redis://ss_redis:6379/0'
70     worker_log_level = 'DEBUG'
71     worker_prefetch_multiplier = 1
72     task_acks_late = False
73     task_annotations = {
74         'sql_lab.get_sql_results': {
75             'rate_limit': '100/s',
76         }
77     }
78     beat_schedule = {
79         "reports.scheduler": {
80             "task": "reports.scheduler",
81             "schedule": crontab(minute="*", hour="*"),
82         },
83         "reports.prune_log": {
84             "task": "reports.prune_log",
85             "schedule": crontab(minute=10, hour=0),
86         },
87     },
88
89
90 CELERY_CONFIG = CeleryConfig
```

MySQL — настройка

```
1 [mysqld]
2
3 #Increase on prod!
4 innodb_buffer_pool_size = 256M
5
6 character-set-server = utf8
7 collation-server = utf8_unicode_ci
8
9 init_connect='SET NAMES utf8; SET collation_connection = utf8_unicode_ci'
10 skip-character-set-client-handshake
11
12 [client]
13 default-character-set = utf8
```

Общий Docker compose, обзор

```
2 services:
3
4   ss_nginx:
5     build: nginx
6     ports:
7       - "1443:1443"
8     depends_on:
9       - ss_superset
10
11   ss_trino:
12     build: trino
13     ulimits:
14       nproc: 131072
15       nofile:
16         soft: 131072
17         hard: 131072
18
19   ss_redis:
20     build: redis
21     volumes:
22       - ss_redis:/data
23
```

```
24   ss_mysql:
25     build: mysql
26     environment:
27       - MYSQL_ROOT_PASSWORD=123456
28       - MYSQL_DATABASE=superset
29     volumes:
30       - ss_mysql:/var/lib/mysql
31     healthcheck:
32       test: ["CMD", "mysqladmin", "ping", "-h", "localhost", "-p123456"]
33       timeout: 20s
34       start_period: 5s
35       interval: 15s
36       retries: 20
37
38   ss_superset:
39     build: superset
40     depends_on:
41       ss_mysql:
42         condition: service_healthy
```

```
44   ss_celery_worker:
45     build: superset
46     depends_on:
47       - ss_superset
48     command: celery --app=superset.tasks.celery_app:app worker -O fair --pool=prefork -c 4 -l INFO
49     restart: on-failure
50
51   ss_celery_beat:
52     build: superset
53     depends_on:
54       - ss_superset
55     command: ["celery", "--app=superset.tasks.celery_app:app",
56             "beat", "--pidfile", "", "-l", "DEBUG", "-s", "/app/superset_home/celerybeat-schedule"]
57     restart: on-failure
58
59 volumes:
60   ss_mysql:
61   ss_redis:
```

Apache Superset + Trino (PrestoSQL)

The screenshot displays the Apache Superset web interface. At the top, navigation tabs include 'Dashboards', 'Charts', 'SQL Lab', and 'Data'. The main area is titled 'Untitled Query 1'. On the left sidebar, the 'DATABASE' is set to 'trino' and the 'SCHEMA' is 'bitrix24'. Under 'SEE TABLE SCHEMA', a list of tables is shown, with 'crm_lead' selected and highlighted. The central SQL editor contains the query: `SELECT * FROM crm_activity WHERE type_name = 'Звонок'`. Below the editor, a 'RUN' button is visible along with 'LIMIT: 1 000' and a timer '00:00:00.25'. The 'RESULTS' tab is active, showing a table with 5 rows returned. The table columns are 'id', 'type_id', 'type_name', 'owner_id', 'owner_type_id', 'owner_type_name', and 'associated_entity_id'. A vertical arrow points to the 'associated_entity_id' column.

1 -- Note: Unless you save your query, these tabs will NOT persist if you clear your cookies or change browser
2
3 `SELECT * FROM crm_activity WHERE type_name = 'Звонок'`

RUN LIMIT: 1 000 00:00:00.25

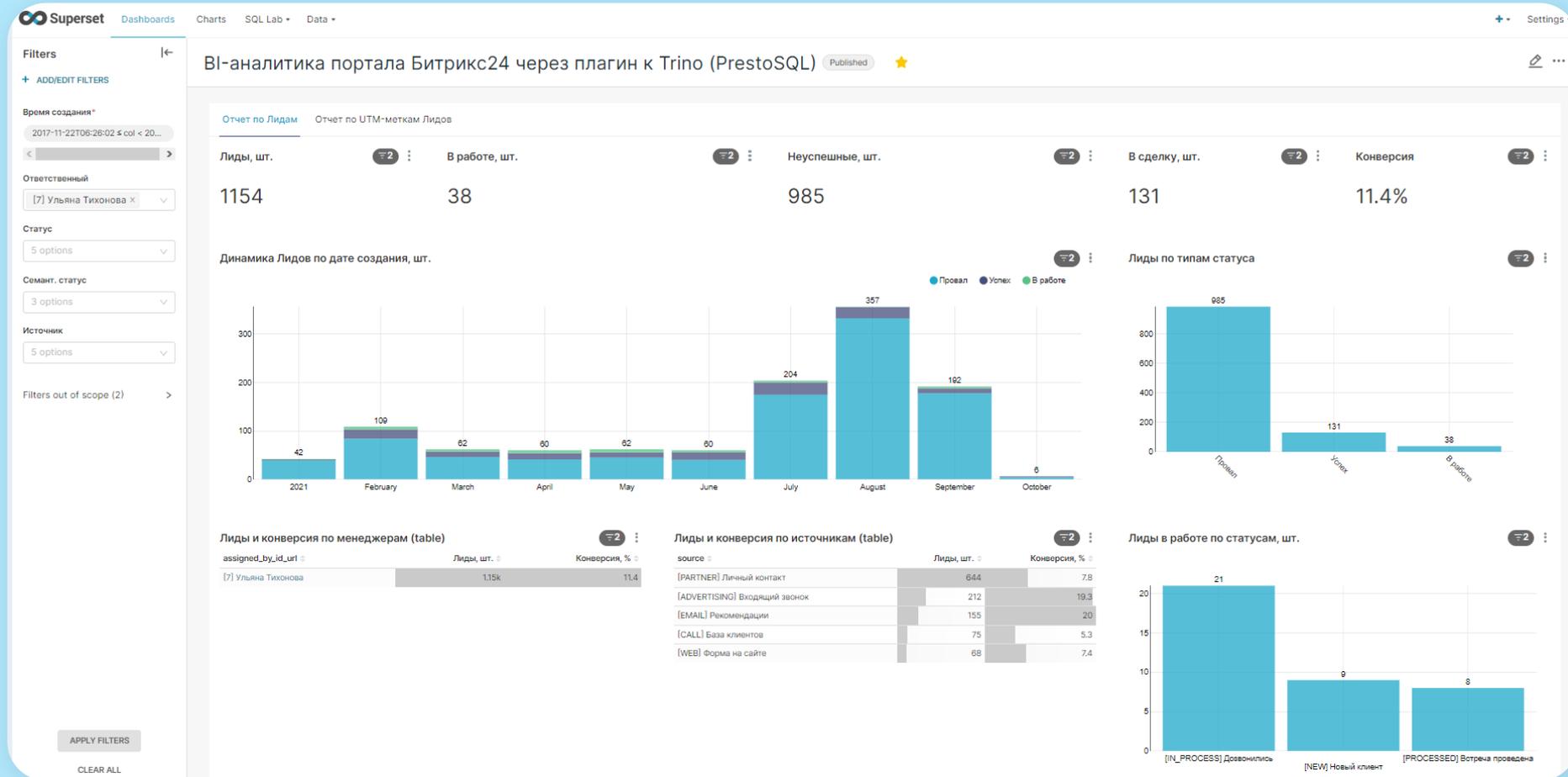
RESULTS QUERY HISTORY PREVIEW: 'CRM_ACTIVITY' PREVIEW: 'CRM_LEAD'

EXPLORE DOWNLOAD TO CSV COPY TO CLIPBOARD Filter results

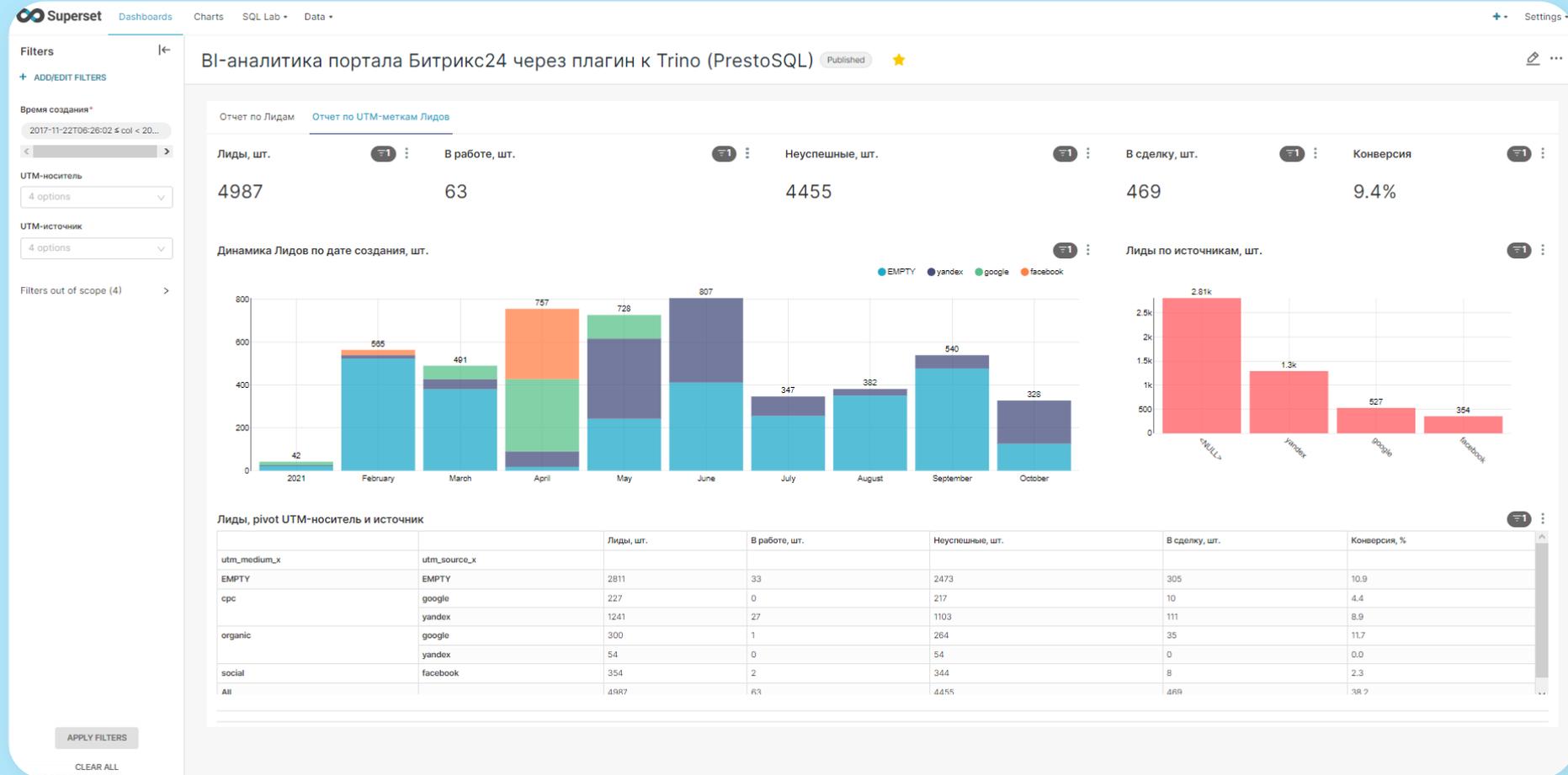
5 rows returned

id	type_id	type_name	owner_id	owner_type_id	owner_type_name	associated_entity_id
1	2	Звонок	2	2	DEAL	0
3	2	Звонок	3	3	CONTACT	0
6	2	Звонок	7	2	DEAL	0
8	2	Звонок	3	2	DEAL	0
16	2	Звонок	14	2	DEAL	0

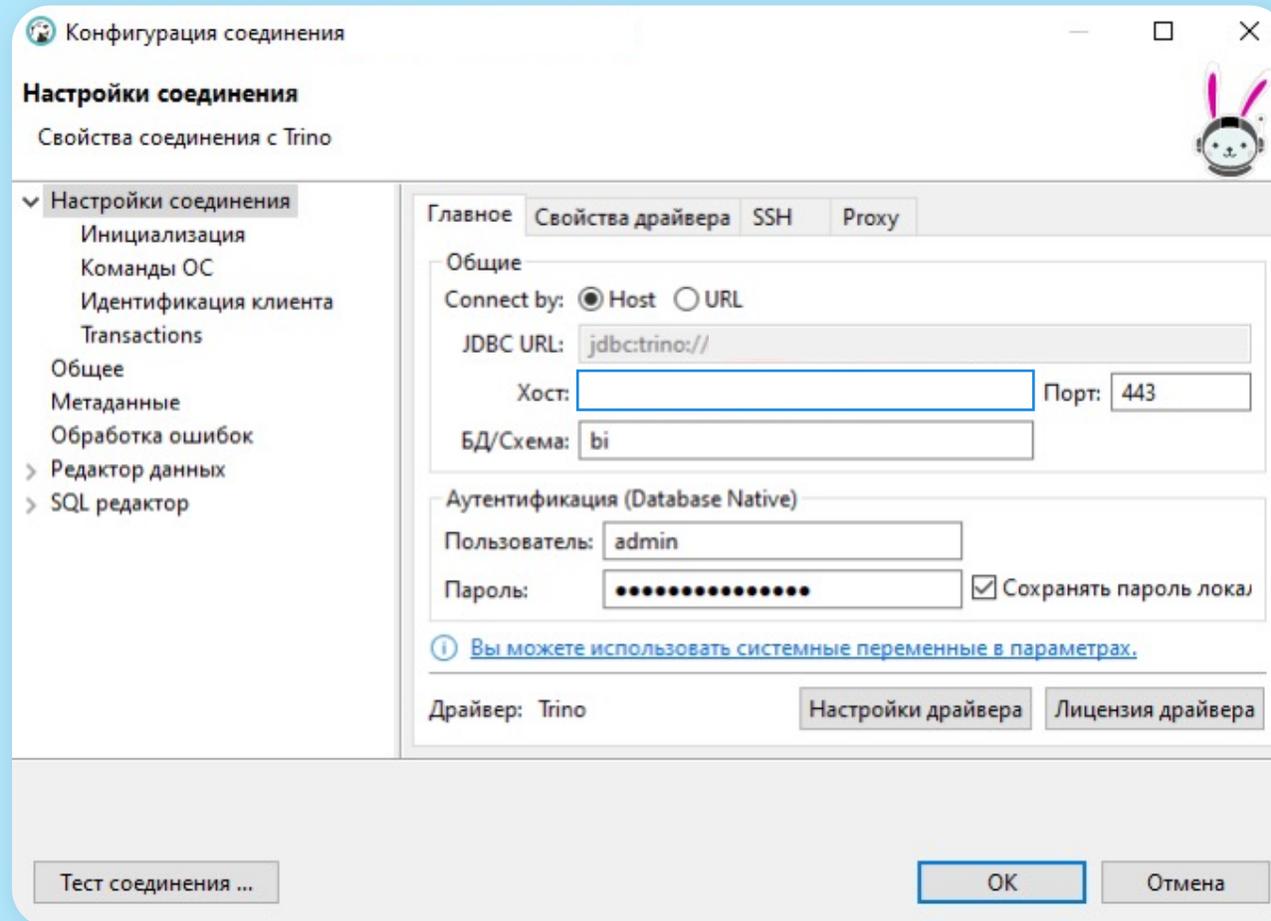
Apache Superset + Trino (PrestoSQL)



Apache Superset + Trino (PrestoSQL)



Trino (PrestoSQL) + Dbeaver — конфигурация JDBC-драйвера



Trino (PrestoSQL) + Dbeaver — BI-таблицы портала Битрикс24

DBeaver 23.0.3 - crm_deal

Файл Редактирование Навигация Поиск Редактор SQL База данных Окна Справка

Auto

Базы данных Проекты SQL Commit Rollback

Исследование демоданных Битрикс24

Connections

bitrix24

Таблицы

crm_activity

crm_company

crm_company_uf

crm_contact

crm_contact_uf

crm_deal

crm_deal_product_row

crm_deal_stage_history

crm_deal_uf

crm_dynamic_items_170

crm_dynamic_items_190

crm_dynamic_items_31

crm_dynamic_items_36

crm_lead

crm_lead_product_row

crm_lead_status_history

crm_lead_uf

socialnetwork_group

telephony_call

test

user

Представления

Индексы

Процедуры

Типы данных

information_schema

system

crm_lead

crm_deal

Свойства Данные Диаграмма

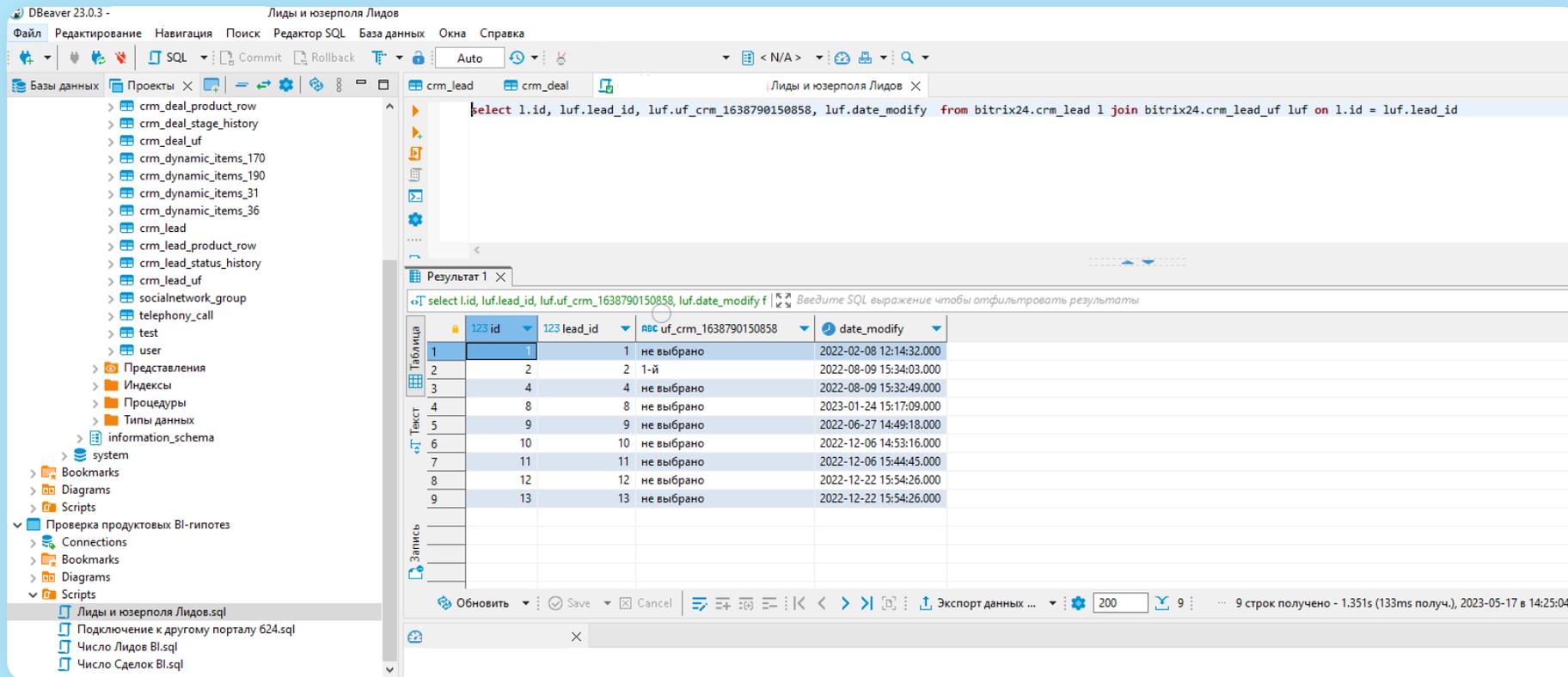
crm_deal

Введите SQL выражение чтобы отфильтровать результаты

	category_id	category_name	category	stage_id	stage_name	stage	stage_semantic_id	stage_semantic	is_new	is_recurring
1	0	Общая	[0] Общая	WON	Сделка успешна	[WON] Сделка успешна	S	Успех	N	N
2	0	Общая	[0] Общая	NEW	Новая заявка	[NEW] Новая заявка	P	В работе	Y	N
3	0	Общая	[0] Общая	PREPARATION	Звонок по монтажу	[PREPARATION] Звонок по монтажу	P	В работе	N	N
4	0	Общая	[0] Общая	PREPARATION	Звонок по монтажу	[PREPARATION] Звонок по монтажу	P	В работе	N	N
5	0	Общая	[0] Общая	PREPARATION	Звонок по монтажу	[PREPARATION] Звонок по монтажу	P	В работе	N	N
6	0	Общая	[0] Общая	PREPARATION	Звонок по монтажу	[PREPARATION] Звонок по монтажу	P	В работе	N	N
7	0	Общая	[0] Общая	PREPARATION	Звонок по монтажу	[PREPARATION] Звонок по монтажу	P	В работе	N	N
8	0	Общая	[0] Общая	LOSE	Дорого	[LOSE] Дорого	F	Провал	N	N
9	0	Общая	[0] Общая	PREPARATION	Звонок по монтажу	[PREPARATION] Звонок по монтажу	P	В работе	N	N
10	0	Общая	[0] Общая	PREPARATION	Звонок по монтажу	[PREPARATION] Звонок по монтажу	P	В работе	N	N
11	0	Общая	[0] Общая	APOLOGY	Дубль	[APOLOGY] Дубль	F	Провал	N	N
12	0	Общая	[0] Общая	PREPARATION	Звонок по монтажу	[PREPARATION] Звонок по монтажу	P	В работе	N	N
13	0	Общая	[0] Общая	6	Спам	[6] Спам	F	Провал	N	N
14	0	Общая	[0] Общая	6	Спам	[6] Спам	F	Провал	N	N
15	0	Общая	[0] Общая	6	Спам	[6] Спам	F	Провал	N	N
16	0	Общая	[0] Общая	7	Передумал / Отменил ремонт	[7] Передумал / Отменил ремонт	F	Провал	N	N
17	0	Общая	[0] Общая	6	Спам	[6] Спам	F	Провал	N	N
18	0	Общая	[0] Общая	PREPARATION	Звонок по монтажу	[PREPARATION] Звонок по монтажу	P	В работе	N	N
19	0	Общая	[0] Общая	PREPARATION	Звонок по монтажу	[PREPARATION] Звонок по монтажу	P	В работе	N	N
20	0	Общая	[0] Общая	6	Спам	[6] Спам	F	Провал	N	N
21	0	Общая	[0] Общая	PREPARATION	Звонок по монтажу	[PREPARATION] Звонок по монтажу	P	В работе	N	N

Обновить Save Cancel Экспорт данных ... 200 200 строк получено - 5.109s (3.574s получ.), 2023-05-17 в 14:20:04

Trino (PrestoSQL) + Dbeaver — соединение Лидов и юзерполей Лидов портала Битрикс24



The screenshot shows the DBeaver 23.0.3 interface. The main window displays a SQL query in the editor:

```
select l.id, luf.lead_id, luf.uf_crm_1638790150858, luf.date_modify from bitrix24.crm_lead l join bitrix24.crm_lead_uf luf on l.id = luf.lead_id
```

Below the editor, the results are displayed in a table with the following columns: id, lead_id, abc_uf_crm_1638790150858, and date_modify. The table contains 9 rows of data.

id	lead_id	abc_uf_crm_1638790150858	date_modify
1	1	не выбрано	2022-02-08 12:14:32.000
2	2	1-й	2022-08-09 15:34:03.000
3	4	не выбрано	2022-08-09 15:32:49.000
4	8	не выбрано	2023-01-24 15:17:09.000
5	9	не выбрано	2022-06-27 14:49:18.000
6	10	не выбрано	2022-12-06 14:53:16.000
7	11	не выбрано	2022-12-06 15:44:45.000
8	12	не выбрано	2022-12-22 15:54:26.000
9	13	не выбрано	2022-12-22 15:54:26.000

The status bar at the bottom indicates that 9 rows were retrieved in 1.351s (133ms per row) on 2023-05-17 at 14:25:04.

Мой youtube-канал с видеоуроками про VI-коннектор Битрикс24

На русском:



На английском:



Спасибо за внимание!

Александр Сербул

Head of Data Science,
ООО «1С-Битрикс»

Tw — @AlexSerbul
Fb — Alexandr Serbul
Email — serbul@1c-bitrix.ru

