

**ИНФОРМАЦИЯ, НЕОБХОДИМАЯ ДЛЯ ЭКСПЛУАТАЦИИ
ЭКЗЕМПЛЯРА МОДУЛЯ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ PMCONTROLLING: PMDATABASE**

Дата: 05.06.2023
Версия: 1.0

2023 г.

СОДЕРЖАНИЕ

1. ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	3
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ	3
3. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ МОДУЛЯ PMDATABASE.....	3
2.1. Подготовка рабочего пространства.....	3
2.2. Создание пользовательской инфраструктуры модуля	4
2.3. Описание конфигурации ETL	5
2.4. Описание конфигурации OLAP и конфигурации таблиц (TABLE).....	6
2.5. Описание конфигурации DASHBOARD	7
2.6. Описание элементов конфигурации DASHBOARD	8
2.7. Описание основного файла запуска скриптов	10
2.8. Запуск скрипта создания дашбордов.....	10
2.9. Открытие созданных дашбордов	11
2.10. Просмотр OLAP кубов	12
2.11. Просмотр настроек ETL.....	13
2.12. Просмотр процессов ETL.....	14

1. ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Сокращение	Расшифровка
Модуль	Модуль программного обеспечения PMControlling: PMDataBase
Система	Программное обеспечение PMControlling
OLAP	Online Analytical Processing (интерактивная аналитическая обработка). Представляет собой технологию для анализа и извлечения данных из больших наборов информации.
ETL	Extract, Transform, Load (извлечение, преобразование, загрузка). Общий термин для всех процессов миграции данных из одного источника в другой.
ELT	Extract, Load, Transform (извлечение, загрузка, преобразование). Общий термин для всех процессов миграции данных из одного источника в другой. Применяется при обработке больших объемов данных для целей бизнес-аналитики
SQL	Structured Query Language (язык структурированных запросов). декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

Модуль программного обеспечения PMControlling: PMDataBase является программным обеспечением, представляющим собой структуры хранения данных и выполнения серверных функций и скриптов баз данных, а также утилиты для работы с используемыми базами данных PMControlling. Модуль содержит также расширение программного обеспечения ClickHouse упрощающее конфигурирование и настройку ClickHouse для создания структуры БД, OLAP кубов, ETL и ELT процессов, а также формирования метаинформации подсистемы корпоративной отчетности, используемой во всех модулях программного обеспечения PMControlling.

Настоящий документ содержит требования по работе с модулем PMDataBase.

3. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ МОДУЛЯ PMDATABASE

2.1. Подготовка рабочего пространства

Необходимо установить Node.js версии 16.x.x, скачав установочный файл по ссылке <https://nodejs.org/en/about/previous-releases> для своей операционной системы.

Далее необходимо скачать zip-архив с утилитой dashboar-creator-xxxx-xx-xx.zip. Распаковать его содержимое в папку с наименованием этого архива.

В терминале перейти в папку с распакованным содержимым:
cd dashboar-creator-xxxx-xx-xx/

Выполнить команду:

npm install

Эта команда установит необходимые зависимости в `dashboar-creator-xxxx-xx-xx/node_modules/`.

2.2. Создание пользовательской инфраструктуры модуля

Перейти в папку `dashboar-creator-xxxx-xx-xx/examples/`.

Создать в ней папку с наименованием проекта в рамках, которого создаются скрипты.

Перейти в эту папку `dashboar-creator-xxxx-xx-xx/examples/name_some_project/`.

Создать в этой папке папку с наименованием схемы `dashboar-creator-xxxx-xx-xx/examples/name_some_project/name_some_schema/`. Использование такого подхода не является обязательным, но позволяет удобно структурировать информацию.

Схема – условная объединяющая сущность, содержащая все конфигурации и настройки в рамках одного источника ETL. В дальнейшем она участвует в формировании пути в URL-адресе дашборда и фигурирует в скриптах, как `schemaId`. Из названия схемы должно быть понятно, какого рода информацию содержат дашборды внутри неё.

Внутри папки с наименованием схемы создаём файлы и папки по аналогии с рекомендованной структурой в папке `dashboar-creator-xxxx-xx-xx/examples/base/`.

Примерная рекомендованная структура папок и файлов представлена на рисунке 2.1.

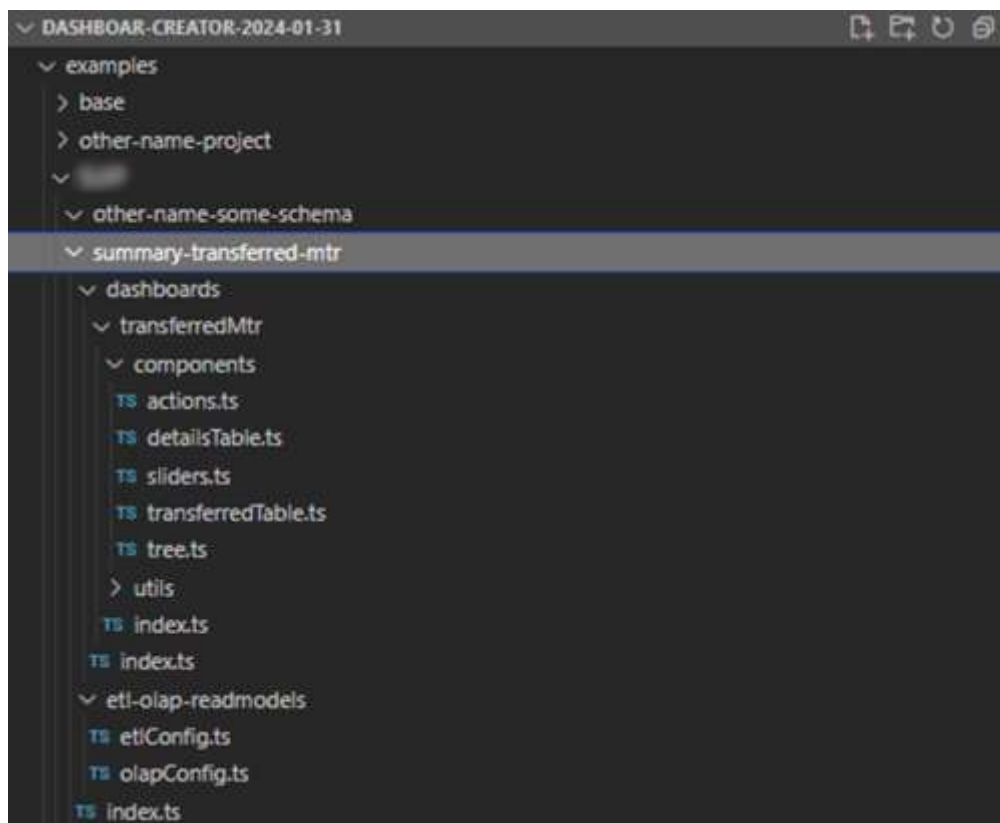


Рисунок 2.1. Примерная структура папок и файлов для схемы.

2.3. Описание конфигурации ETL

Для создания конфигурации ETL необходимо указать расписание CRON в пятизначном формате, источник данных с описанием параметров подключения к базе данных, последовательные операции ETL-процесса (jobs - шаги, которые будут выполнены при получении данных из источника, их преобразовании и загрузке в базу назначения).

Пример заполнения конфигурации ETL представлен на рисунке 2.2.

```
TS etlConfig.ts X
examples : summary-transferred-mtr > etl-olap-readmodels > TS etlConfig.ts > [e] etlConfig
1  import { ETLConfig } from "@pmc-bi/dashboard-creator";
2
3  export const etlConfig: ETLConfig = {
4    cron: "0 0 15 7 *",
5    sources: {
6      postgresql: {
7        type: "PGSQL",
8        payload: {
9          user: "db_user_name",
10         password: "db_user_pass",
11         database: "db_name",
12         host: "db_ip_host",
13         port: 5432,
14       },
15     },
16   },
17   jobs: [
18     {
19       type: "PGSQL.EXTRACT_FROM_CUSTOM_EXECUTE_QUERY",
20       payload: {
21         sourceName: "pgsql",
22         destinationTableName: "mtr_table",
23         sql: `
24           SELECT * FROM dev.db_table_name
25         `,
26       },
27     },
28     {
29       type: "Warehouse.EXECUTE_SQL",
30       payload: {
31         sql: `
32           INSERT INTO ${olapSchemaName~}.mtr_table
33             (id, project_id, stage_name, object_name, quantity)
34           SELECT
35             ROW_NUMBER() OVER () AS id, project_id, stage_name, object_name, quantity
36           FROM ${etlSchemaName~}.mtr_table
37         `,
38       },
39     },
40   ],
41 };
```

Рисунок 2.2. Пример заполнения конфигурации ETL.

При выполнении операции «EXTRACT_FROM_CUSTOM_EXECUTE_QUERY» данные извлекаются из источника и помещаются в таблицу destinationTableName внутри схемы БД с наименованием __ETL_SCHEMA__name_schema.

Операция с типом «Warehouse.EXECUTE_SQL» используется для преобразования данных. При выполнении этой операции данные выбираются из таблицы в схеме БД __ETL_SCHEMA__schema_name, преобразуются внутри SELECT и помещаются в таблицу в схеме БД __OLAP_SCHEMA__schema_name в указанные в INSERT-е поля.

Параметры \$(etlSchemaName~), \$(olapSchemaName~) используются для динамического формирования наименования схем БД на основе наименования схемы дашбордов.

Промежуточные схемы БД, формируемые в результате выполнения процесса ETL, представлены на рисунке 2.3.

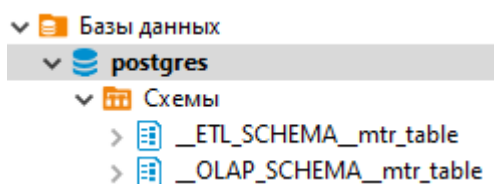


Рисунок 2.3. Промежуточные схемы БД при выполнении ETL-процесса.

2.4. Описание конфигурации OLAP и конфигурации таблиц (TABLE)

Эти конфиги предназначены для описания таблиц назначения, в которые будут загружены данные в результате выполнения процесса ETL для формирования на их основе дашбордов.

Конфигурация OLAP содержит в своём описании условное деление полей на измерения и факты. Конфигурация TABLE содержит просто описание полей таблицы, создаваемой в базе назначения.

Примеры заполнения конфигурации OLAP и конфигурации TABLE представлены на рисунке 2.4 и рисунке 2.5.

```

TS olapConfig.ts X
examples > > summary-transferred-mtr > etl-olap-readmodels > TS olapConfig.ts > olapConfig
1 import { OlapConfig, TableConfig } from "@pmc-bi/dashboard-creator";
2
3 export const olapConfig: OlapConfig = {
4   primaryKeyName: "id",
5   primaryKeyType: "bigint",
6   dimensionKeyPostfix: "_id",
7   dimensions: [
8     {
9       tableName: "mtr_table",
10      parentTablesNames: [],
11      fields: [
12        { dimensionName: "project_id", fieldName: "project_id", type: "integer" },
13        { dimensionName: "stage_name", fieldName: "stage_name", type: "varchar" },
14        { dimensionName: "object_name", fieldName: "object_name", type: "varchar" },
15      ],
16    },
17  ],
18  facts: [
19    {
20      tableName: "mtr_table",
21      fields: [
22        { fieldName: "quantity", type: "double" },
23      ],
24    },
25  ],
26 };

```

Рисунок 2.4. Пример заполнения конфигурации OLAP.

```

TS olapConfig.ts X
examples > SUIP > summary-transferred-mtr > etl-olap-readmodels > TS olapConfig.ts > tableConfigs
1 import { OlapConfig, TableConfig } from "@pmc-bi/dashboard-creator";
2
3 export const tableConfigs: TableConfig[] = [
4   {
5     name: "mtr_table",
6     columns: [
7       { name: "id", type: "bigint" },
8       { name: "project_id", type: "integer" },
9       { name: "construction_name", type: "varchar" },
10      { name: "stage_name", type: "varchar" },
11      { name: "object_name", type: "varchar" },
12      { name: "quantity", type: "double" },
13    ],
14     primaryKey: "id",
15   },
16 ];

```

Рисунок 2.5. Пример заполнения конфигурации TABLE.

2.5. Описание конфигурации DASHBOARD

Эта конфигурация предназначена для описания визуальной части дашборда. Она содержит в себе наименование дашборда, наименования и описание виджетов, описание размещения элементов дашборда, обработчики различных событий.

Пример заполнения конфигурации DASHBOARD представлен на рисунке 2.6.

```

index.ts x
examples > > summary-transferred-mtr > dashboards > transferredMtr > index.ts > dashboardTransferredMtrs

1 import { transferredTable } from "../components/transferredTable";
2 import { actions } from "../components/actions";
3 import { slideMenuFilters } from "../components/sliders";
4
5 const containerStyle = {
6   // необходимые свойства CSS в camelCase
7 }
8
9 export const dashboardTransferredMtrs: any = {
10  displayName: "Сводная информация по переданным в монтаж МТР",
11  pages: {
12    index: {
13      globalStyles: `
14      .PMCBi-slide-menu__children {
15        height: 100%;
16      }
17      `,
18      content: [
19        {
20          type: "container",
21          payload: {
22            style: {
23              // необходимые свойства CSS в camelCase
24            },
25            content: [
26              {
27                type: "container",
28                payload: {
29                  style: {
30                    ...containerStyle,
31                    flexDirection: "column",
32                  },
33                  content: [
34                    actions,
35                    transferredTable,
36                  ],
37                },
38              },
39            ],
40          },
41        },
42        slideMenuFilters,
43      ],
44    },
45  },
46 };

```

Рисунок 2.6. Пример заполнения конфигурации DASHBOARD.

2.6. Описание элементов конфигурации DASHBOARD

Существуют следующие основные виджеты Composed Chart, Gauge, Kpi, Pie Chart, Table2, Timeline, Tree, компоненты Slide Menu, Filters и другие элементы дашборда (панели, кнопки и т.п.). Удобнее всего каждый виджет описывать в отдельном файле для декомпозиции кода. А затем из этих файлов импортировать эти элементы в файл с конфигурацией DASHBOARDа и использовать в описании визуального представления дашборда.

Пример описания виджета table2 представлен на рисунке 2.7.

```
transferedTable.ts X
examples > > summary-transfered-mtr > dashboards > transferedMtr > components > transferedTable.ts

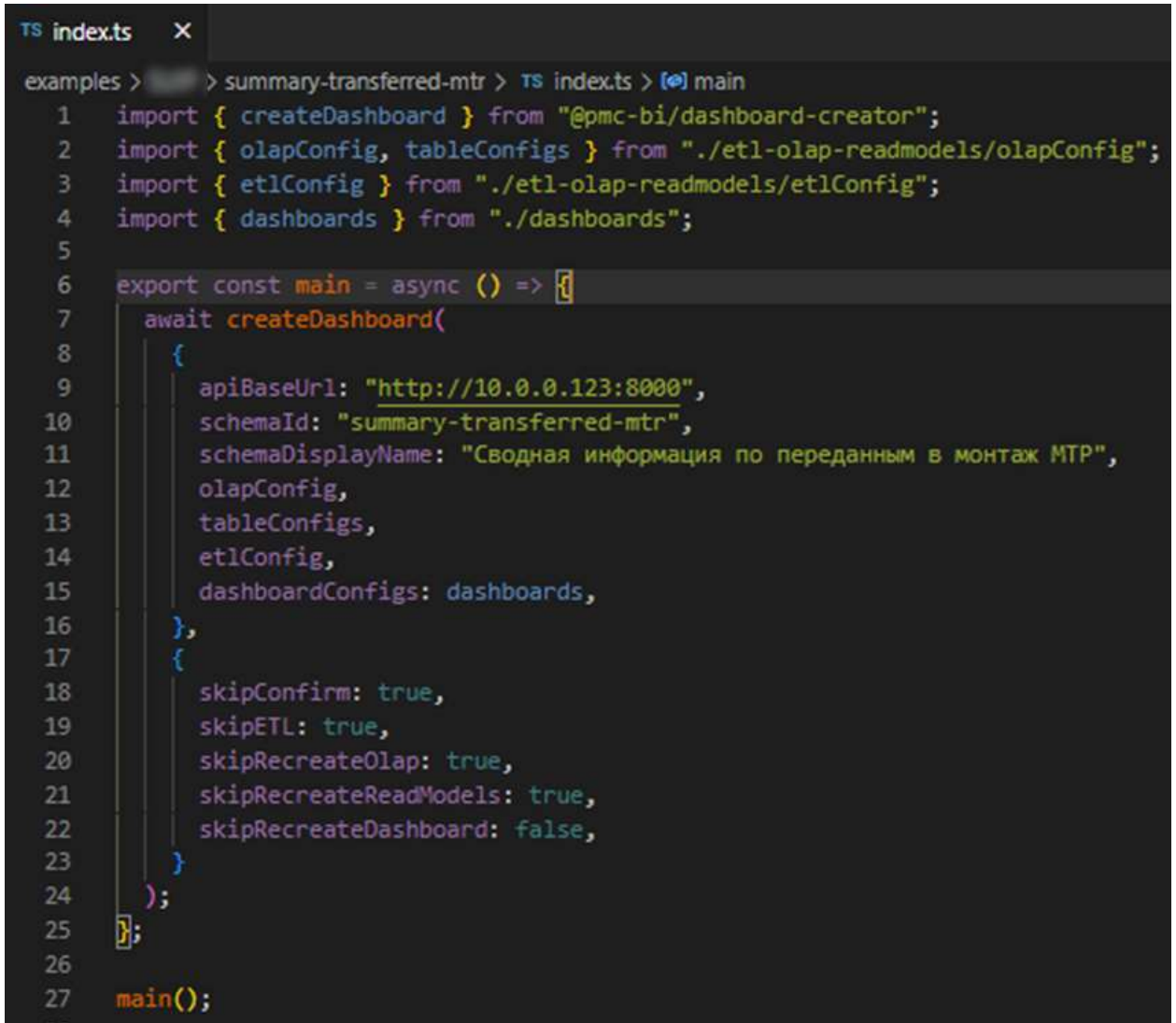
1 import { IDashboardTable } from "@pmc-bi/dashboard-creator";
2 import {
3   fnWhereIdMbs,
4   rowFormatter,
5   onRowClick,
6 } from "../utils/helpers";
7
8 export const transferedTable: IDashboardTable = {
9   name: "transferedTable",
10  type: "table2",
11  rowFormatter: rowFormatter.toString(),
12  onRowClick: onRowClick.toString(),
13  rowGroups: [
14    {
15      sourceTable: "mtr_table",
16      groupBy: "type_supply, mtr_group",
17      where: [
18        {
19          type: "selected",
20          formula: fnWhereIdMbs.toString(),
21        },
22      ],
23      cols: [
24        {
25          formula: "mtr_group",
26          tree: true,
27          displayName: "Группа МТР",
28          alias: "mtr_group",
29        },
30        {
31          formula: "sum(plan_engagement)",
32          displayName: "План вовлечения",
33          alias: "plan_engagement",
34        },
35        {
36          type: "meta",
37          formula: "type_supply",
38          alias: "type_supply",
39        }
40      ],
41    },
42  ],
43 };
```

Рисунок 2.7. Пример описания виджета table2.

2.7. Описание основного файла запуска скриптов

В корне папки с наименованием рассматриваемой схемы дашбордов должен располагаться основной файл, который будет запускаться в первую очередь при выполнении скриптов, так называемая точка входа. Это может быть файл `index.ts`.

Пример такого файла представлен на рисунке 2.8.



```
TS index.ts  X
examples > > summary-transferred-mtr > TS index.ts > [⌘] main
1  import { createDashboard } from "@pmc-bi/dashboard-creator";
2  import { olapConfig, tableConfigs } from "./etl-olap-readmodels/olapConfig";
3  import { etlConfig } from "./etl-olap-readmodels/etlConfig";
4  import { dashboards } from "./dashboards";
5
6  export const main = async () => {
7    await createDashboard(
8      {
9        apiUrl: "http://10.0.0.123:8000",
10       schemaId: "summary-transferred-mtr",
11       schemaDisplayName: "Сводная информация по переданным в монтаж МТР",
12       olapConfig,
13       tableConfigs,
14       etlConfig,
15       dashboardConfigs: dashboards,
16     },
17     {
18       skipConfirm: true,
19       skipETL: true,
20       skipRecreateOlap: true,
21       skipRecreateReadModels: true,
22       skipRecreateDashboard: false,
23     }
24   );
25 }
26
27 main();
```

Рисунок 2.8. Пример содержания основного файла.

В этом файле задается:

- `apiBaseUrl` – адрес бэкенда PMControlling Analytics
- `schemaId` – наименование схемы с дашбордами

Импортируются все конфигурации, определенные ранее. А также импортируется и вызывается функция, которая отправляет все эти конфиги в бэкенд Системы.

2.8. Запуск скрипта создания дашбордов

В терминале внутри папки `dashboar-creator-xxxx-xx-xx/` запускаем основной файл, используя `ts-node`.

Выполняем команду:

```
npx ts-node .\examples\project-name\dashboards-schema-name.ts
```

Например:

```
npx ts-node .\examples\internal\summary-transferred-mtr\index.ts
```

Для пересоздания только определенных конфигов после указания основного файла в этой команде запуска можно указывать флаги через пробел --olap, --etl, --dash, каждый в отдельности или все сразу. Будут пересозданы только указанные конфиги.

2.9. Открытие созданных дашбордов

Созданные дашборды будут доступны в браузере по адресу следующего вида:

```
http://bi-frontend-ip:bi-frontend-port/#/schema/dashboards-schema-name
```

Например:

```
http://10.0.0.123:6202/#/schema/summary-transferred-mtr
```

Перейдя по такому адресу, на закладке «Дашборды» выбрать интересующий дашборд и нажать кнопку «Открыть».

Созданные дашборды будут доступны в PMControlling Analytics в браузере по адресу следующего вида:

```
http://bi-frontend-ip:bi-frontend-port/#/schema/dashboards-schema-name
```

Например:

```
http://10.0.0.123:6202/#/schema/summary-transferred-mtr
```

Перейдя по такому адресу, в PMControlling Analytics на закладке «Дашборды» выбрать интересующий дашборд и нажать кнопку «Открыть».

Пример открывшегося дашборда PMControlling Analytics представлен на рисунке 2.9.



Рисунок 2.9. Пример дашборда.

2.10. Просмотр OLAP кубов

Для просмотра кубов после открытия главной страницы необходимо нажать на плитку «Список OLAP-кубов». При этом откроется список всех OLAP-кубов, присутствующих на рассматриваемом сервере.

Кликнув по конкретному кубу, можно открыть список дашбордов, созданных на его основе, а также настройки ETL и прочие настройки.

Пример списка OLAP-кубов представлен на рисунке 2.10.

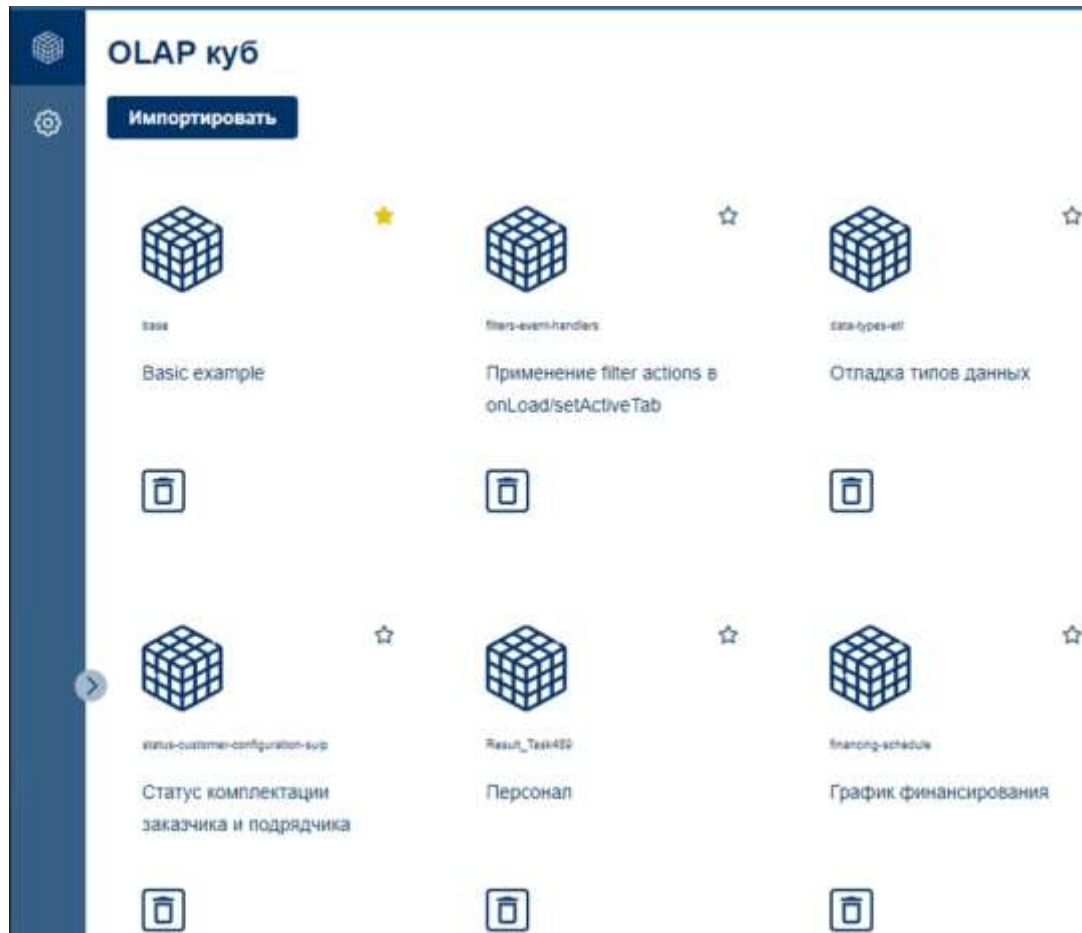


Рисунок 2.10. Пример списка OLAP-кубов

2.11. Просмотр настроек ETL

Выбрав конкретный куб и перейдя на вкладку «ETL конфиг», можно увидеть настройки ETL для этого куба.

Пример настроек ETL представлен на рисунке 2.11.

Дашборды
ETL конфиг
ETL процессы
Импорт / Экспорт
Системный дашборд
Окружение

cron:

Источники

id: pgsql

Тип:

user: password: port:

host: database:

Сохранить
Сбросить

Задачи

type:

sourceName: destinationTableName:

sql:

✖

type:

sql:

Сохранить
Сбросить

Рисунок 2.11. Пример списка настроек ETL.

2.12. Просмотр процессов ETL

Для просмотра информации о выполненных процессах ETL и состоянии, в котором они находятся, необходимо перейти на вкладку «ETL процессы».

Пример списка процессов ETL представлен на рисунке 2.12.

Дашборды
ETL конфиг
ETL процессы
Импорт / Экспорт
Системный дашборд
Окружение

История ETL-процессов

Запустить пайплайна

Статус	ID Пайплайна	Старт	Продолжительность	Тип запуска
passed	03304ca3-ad07-42ce-a1fa-a31f93ccc5fc	06.02.2024 13:41:38	1 секунда	manual
passed	38e4f7c1-9240-4b0f-a275-2e3e3c17c8a2	05.02.2024 19:06:27	1 секунда	cron

Рисунок 2.12. Пример списка процессов ETL.