Оглавление

[1. Создание таблиц справочников и фактов с экспортом данных в файлы формата “CSV” 2](#_Toc124369012)

[1.1 Основное задание 2](#_Toc124369013)

[1.2 Трансформация формирования таблиц справочников 2](#_Toc124369014)

[1.3 Трансформация формирования таблицы фактов 3](#_Toc124369015)

[1.4 Трансформация записи таблиц справочников в файл «CSV» 3](#_Toc124369016)

[1.5 Трансформации записи таблицы фактов в файл «CSV» 5](#_Toc124369017)

[1.5.1 Трансформация получения данных фактов из БД 5](#_Toc124369018)

[1.5.2 Трансформация записи данных фактов в файл «CSV» 6](#_Toc124369019)

[2. SQL-скрипты для работы трансформаций 9](#_Toc124369020)

[2.1 Трансформация формирования таблиц справочников 9](#_Toc124369021)

[2.2 Трансформация формирования таблицы фактов 10](#_Toc124369022)

[2.3 Трансформация записи таблиц справочников в файл «CSV» 10](#_Toc124369023)

[2.4 Трансформация получения данных фактов из БД 11](#_Toc124369024)

1. **Создание таблиц справочников и фактов с экспортом данных в файлы формата “CSV”**

Для выполнения задачи предусматривается задание («Project\_J1») с последовательным выполнением трансформаций для реализации требуемой задачи.

* 1. **Основное задание**

Задание состоит из начального ETL-элемента «Start» и ряда трансформаций, последовательно соединенных друг с другом (Рисунок 1). Результатом выполнения задания является:

* создание таблиц справочников и фактов в схеме «dwh\_1» на основе данных, полученных из БД «Bookings».
* Создание «выгрузки» данных из созданных таблиц справочников и фактов схемы «dwh\_1» в файлы формата «csv».

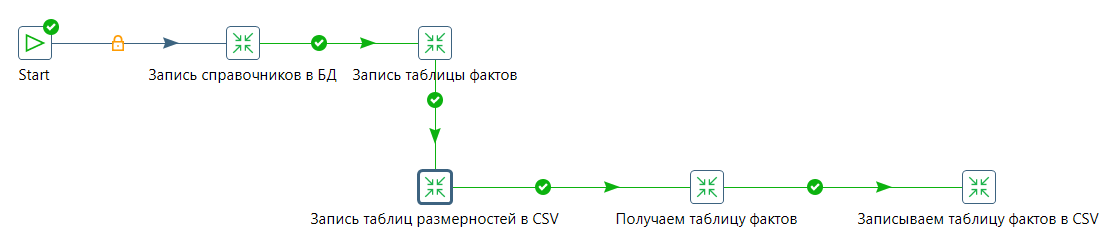
****

Рисунок 1 – Основное задание

* 1. **Трансформация формирования таблиц справочников**

Трансформация формирования таблиц справочников состоит из пяти элементов одинаковых ETL-элементов «Execute SQL script» (Рисунок 2), вызываемых одновременно после записи в лог сообщения «Create tables» (ETL-элемент «Write to log»).

ETL-элементы «Execute SQL script» содержат вызываемые SQL-скрипты для создания таблиц в схеме «dwh\_1» с необходимыми проверками, применяющимися при наполнении данных. SQL-крипты для каждого справочника приведены в разделе 2.

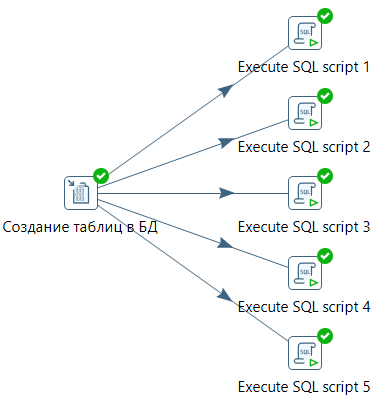


Рисунок 2 – Трансформация формирования справочников в БД

* 1. **Трансформация формирования таблицы фактов**

Трансформация формирования таблицы фактов состоит из одного ETL-элемента «Execute SQL script», вызывающего SQL-скрипт для создания и наполнения таблицы фактов в схеме «dwh\_1».

После того как справочники созданы в схеме «dwh\_1» выполняется создание таблицы фактов с указанием ограничений, ссылающихся на таблицы фактов, ранее созданных в этой же схеме.

Скрипт для данной трансформации приведен в разделе 2.

* 1. **Трансформация записи таблиц справочников в файл «CSV»**

Трансформация формирования справочников состоит из пяти одинаковых цепочек ETL-элементов (Рисунок 4), результатом которых является создание в БД таблиц справочников. Описание элементов приведено в таблице ниже.

Таблица 1 – Описание элементов трансформации формирования справочников

|  |  |
| --- | --- |
| **ETL-элемент** | **Описание работы элемента** |
| Table input | Элемент выполняет выборку из SQL-таблицы и передает выбранный DataSet в поток для дальнейшего использования |
| Select values | Элемент выполняет выборку из входного потока полей с данными для обработки и дальнейшей передачи |
| Text file output | Элемент выполняет запись полученных из потока данных в файл с задаваемым в настройках расширением, разделителем, форматом переноса строки, кодировкой и др.. Для текущей трансформации используется расширение «csv» |

Для ETL-элемента «Table input» используется подключение к БД с настройками, в которых указываются глобальные переменные (Рисунок 3) Pentaho DI, конфигурируемые в файле «kettle.properties».

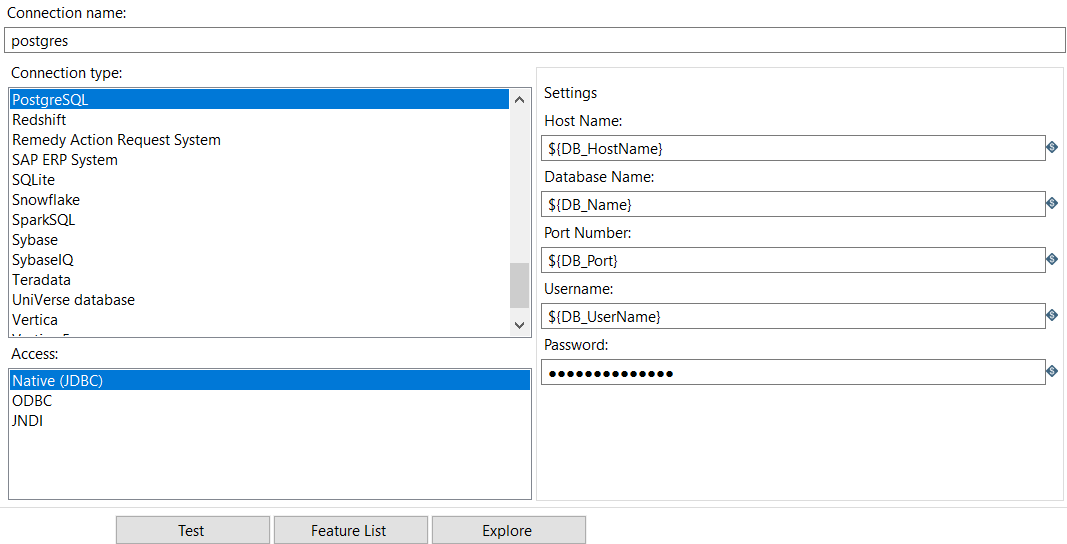


Рисунок 3 – Настройки подключения к БД

Скрипты выборки для данной трансформации приведен в разделе 2.

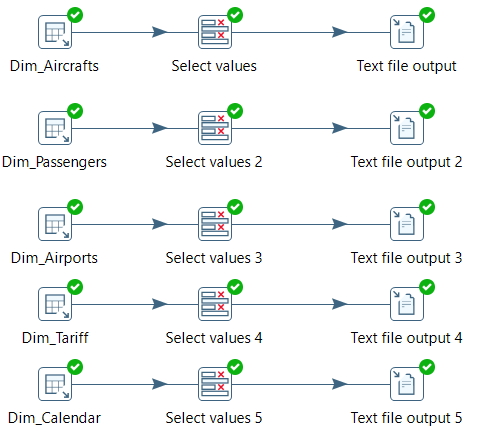


Рисунок 4 – Трансформация формирования справочников

* 1. **Трансформации записи таблицы фактов в файл «CSV»**

Запись таблицы фактов состоит из двух трансформаций, т.к. данная таблица имеет достаточно большое количество данных, что может приводить к ошибке переполнения буфера данных для операций при записи в файл.

Первая трансформация получает данные из таблицы SQL и передает их в поток Pentaho DI. Вторая получает данные из потока в виде параметров каждой строки и записывает их в файл.

* + 1. **Трансформация получения данных фактов из БД**

Трансформация состоит из ETL-элементов «Table Input», «Select values» и «Copy rows to result»(Рисунок 5).

Описание работы первых двух элементов представлено в Таблице 1. Элемент «Copy rows to result» выполняет копирование данных потока в результат выполнения трансформации, поток которого получает основное задание в этом шаге трансформации.

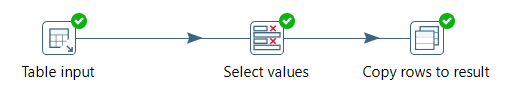
****

Рисунок 5 – Трансформация получения данных фактов из БД

На уровне задания полученные данные копируются в аргументы с соответствующими полями выбранной таблицы (Рисунок 6).

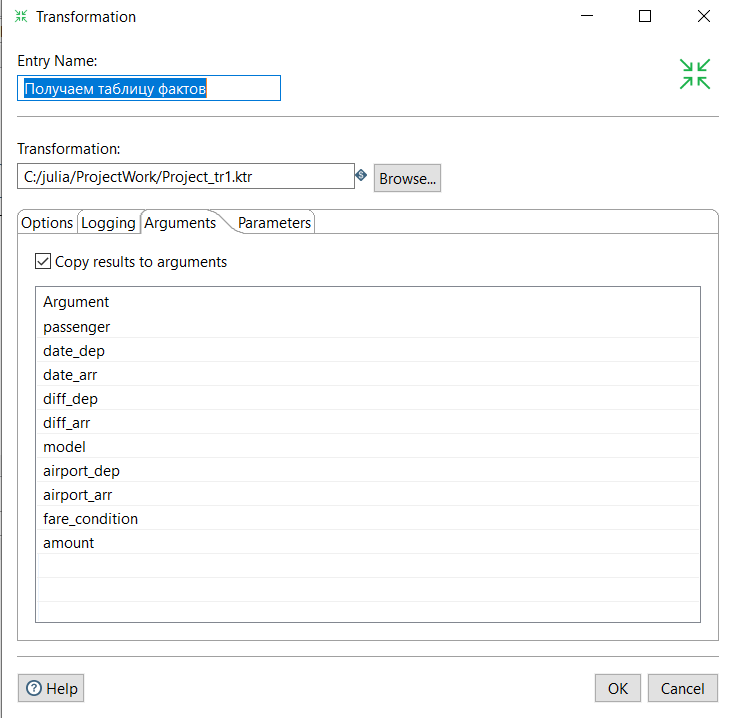


Рисунок 6 – Настройки аргументов в трансформации на уровне задания

* + 1. **Трансформация записи данных фактов в файл «CSV»**

В трансформацию передаем данные, полученные на предыдущем шаге задания. На уровне задания выполняется настройка вызова трансформации для каждой строки, полученной из входного потока(Рисунок 7, Рисунок 8).

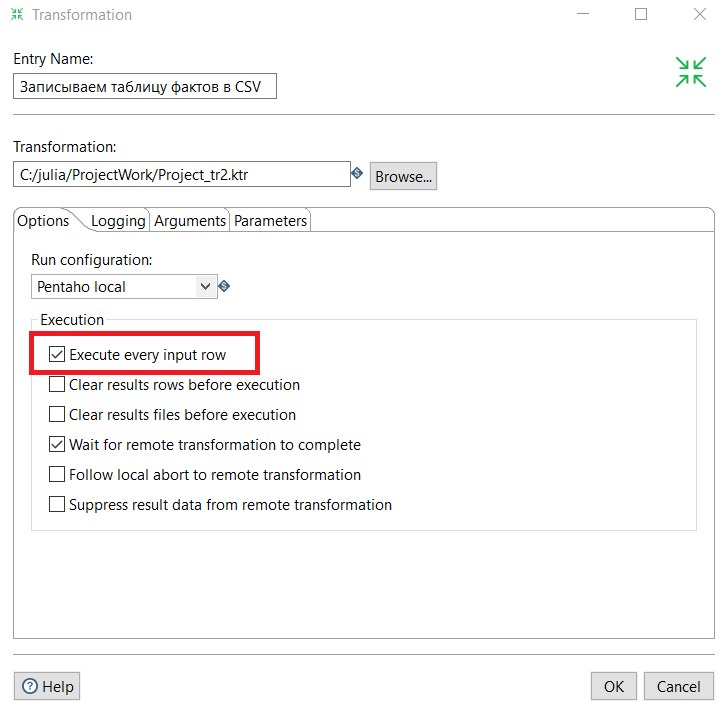
****

Рисунок 7 –Настройки трансформации на уровне задания, вкладка «Options»

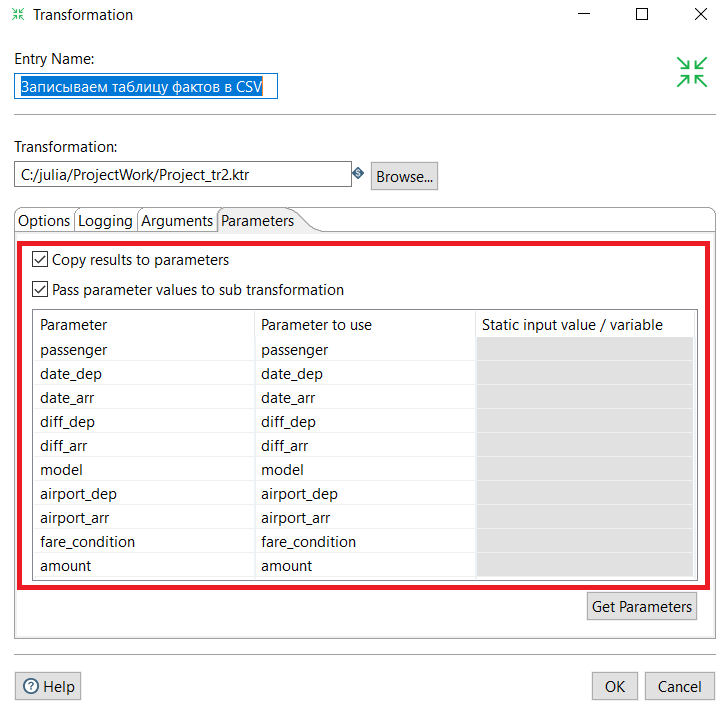
****

Рисунок 8 –Настройки трансформации на уровне задания, вкладка «Parameters»

Трансформация состоит из трех ETL-элементов «Get Variables», «Select values» и «Text file output». Описание элементов представлено в таблице ниже.

Таблица 2 – Описание элементов трансформации записи данных фактов в файл «CSV»

| **ETL-элемент** | **Описание работы элемента** |
| --- | --- |
| Get variable | Элемент получает значения переменных трансформации и передает их в поля для выходного потока. |
| Select values | Элемент выполняет выборку из входного потока полей с данными для обработки и дальнейшей передачи |
| Text file output | Элемент выполняет запись полученных из потока данных в файл с задаваемым в настройках расширением, разделителем, форматом переноса строки, кодировкой и др.. Для текущей трансформации используется расширение «csv».  Т.к. трансформация выполняется для каждой строки, в настроках во вкладке «Content» выставлена опция «Append», позволяющая добавлять новые данные в конец файла, а при отсутствии файла- создавать его. |

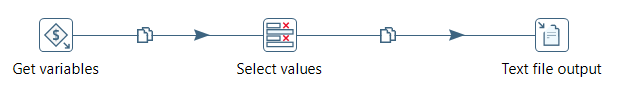
****

Рисунок 7 – Трансформация записи данных фактов в файл «CSV»

1. **SQL-скрипты для работы трансформаций**
   1. **Трансформация формирования таблиц справочников**

Таблица 3 – SQL-скрипты для трансформации формирования таблиц справочников

| **Описание скрипта** | **SQL-скрипт** |
| --- | --- |
| Создание и наполнение справочника «dim\_tariff» | create table dwh\_1.dim\_tariff ( --создаю Dim\_Tariff - справочник тарифов  id serial primary key, -- serial отвечает за гарантированное наполнение атрибута уникальными значениями, не равными null  fare\_conditions varchar(50) not null unique --проверка значения должны быть уникальными, не должно быть пустых значений  );  insert into dwh\_1.dim\_tariff (fare\_conditions) --заполняю справочник уникальными значениями атрибута fare\_condition из отношения fact\_flights схемы bookings  select distinct fare\_condition  from fact\_flights ff; |
| Создание и наполнение справочника «dim\_calendar» | create table dwh\_1.Dim\_Calendar ( --создаю Dim\_Calendar - справочник дат  "date" date not null);  insert into dwh\_1.Dim\_Calendar --заполняю справочник данными  select \* from generate\_series ('2017-01-01'::timestamp without time zone, '2025-01-31', interval '1 day') |
| Создание и наполнение справочника «dim\_airports» | create table dwh\_1.Dim\_Airports ( --создаю Dim\_Airports - справочник аэропортов  airport\_code varchar (3) primary key,  airport\_name varchar (100) not null unique,  city varchar (100) not null,  timezone varchar (100) not null  );  insert into dwh\_1.Dim\_Airports -- заполняю справочник данными из таблицы airports\_data схемы bookings  select airport\_code,  airport\_name ->> 'ru' as airport\_name,  city ->> 'ru' as city,  timezone  from bookings.airports\_data |
| Создание и наполнение справочника «dim\_passengers» | create table dwh\_1.Dim\_Passengers ( --создаю справочник пассажиров Dim\_Passengers  passenger\_id varchar(30) primary key, -- значение должно быть обязательно заполнено и оно должно быть уникальным  passenger\_name varchar (300) not null, -- значение должно быть обязательно заполнено  contact\_data varchar(300) not null -- значение должно быть обязательно заполнено  );  insert into dwh\_1.Dim\_Passengers -- заполняю справочник данными из таблицы tickets схемы bookings  select passenger\_id, passenger\_name, contact\_data  from tickets t |
| Создание и наполнение справочника «dim\_aircrafts» | create table dwh\_1.Dim\_Aircrafts ( --создаю таблицу Dim\_Aircrafts - справочник самолетов  aircraft\_code varchar(3) primary key, --primary key гарантирует уникальность значений и обязательность их заполнения  model varchar (100) not null unique, -- значение не может быть равно null и должно быть уникальным  "range" int not null check ("range" > 0) --значение не может быть равно null и должно быть больше 0  );  insert into dwh\_1.Dim\_Aircrafts (aircraft\_code, model, "range") -- наполняю таблицу "Dim\_Aircrafts - справочник самолетов" данными  select  aircraft\_code,  model::json->>'ru' as model,  range  from bookings.aircrafts\_data ad |

* 1. **Трансформация формирования таблицы фактов**

Таблица 4 – SQL-скрипт для трансформации формирования таблицы фактов

| **Описание скрипта** | **SQL-скрипт** |
| --- | --- |
| Создание и наполнение таблицы фактов «fact\_flights» | create table dwh\_1.fact\_flights ( --создаю в схеме dwh\_1 таблицу fact\_flights  passenger int4 not null,  date\_dep timestamp check (date\_dep <= current\_date), --проверка: дата вылета(факт) не может быть больше текущей даты  date\_arr timestamp check (date\_arr > date\_dep),  diff\_dep varchar(20),  diff\_arr varchar(20),  model varchar (100) not null references dwh\_1.Dim\_Aircrafts (model), --ограничение на произвольное заполнение, значения доступны только из таблицы Dim\_Aircrafts, значение не может быть нулевым  airport\_dep varchar(50) not null references dwh\_1.Dim\_Airports (airport\_name), --ограничение на произвольное заполнение, значения доступны только из таблицы Dim\_Airports, значение не может быть нулевым  airport\_arr varchar(50) not null references dwh\_1.Dim\_Airports (airport\_name) check (airport\_arr != airport\_dep), --ограничение на произвольное заполнение,  --значения доступны только из таблицы Dim\_Airports, значение не может быть нулевым, аэропорт прилета не может быть равен аэропорту вылета  fare\_condition varchar(20) not null references dwh\_1.Dim\_Tariff (fare\_conditions), --ограничение на произвольное заполнение, значения доступны только из таблицы Dim\_Tariff, значение не может быть нулевым  amount numeric(10, 2) not null --значение не может быть нулевым  ); |

* 1. **Трансформация записи таблиц справочников в файл «CSV»**

Таблица 5 – SQL-скрипт для трансформации записи таблиц справочников в файл «CSV»

| **Описание скрипта** | **SQL-скрипт** |
| --- | --- |
| Выборка данных для справочника «dim\_aircrafts» | select  aircraft\_code,  model::json->>'ru' as model,  range  from bookings.aircrafts\_data ad |
| Выборка данных для справочника «dim\_passengers» | select passenger\_id, passenger\_name, contact\_data  from bookings.tickets |
| Выборка данных для справочника «dim\_airports» | select airport\_code,  airport\_name ->> 'ru' as airport\_name,  city ->> 'ru' as city,  timezone  from bookings.airports\_data |
| Выборка данных для справочника «dim\_tariff» | select distinct fare\_condition  from bookings.fact\_flights |
| Выборка данных для справочника «dim\_calendar» | select \* from generate\_series ('2017-01-01'::timestamp without time zone, '2025-01-31', interval '1 day') limit 2 |

* 1. **Трансформация получения данных фактов из БД**

Для выборки данных из таблицы фактов используется SQL-запрос:

select \* from bookings.fact\_flights;