**Центр онлайн-обучения «Нетология-групп»**

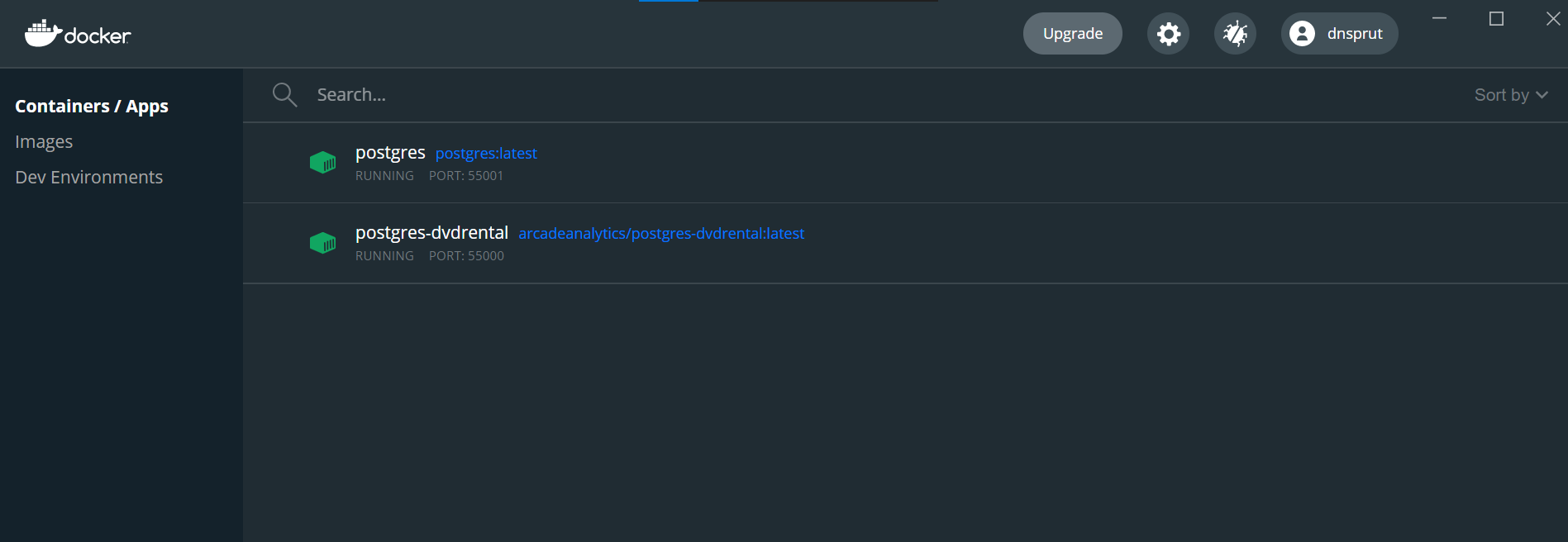
**Проектная работа по модулю**

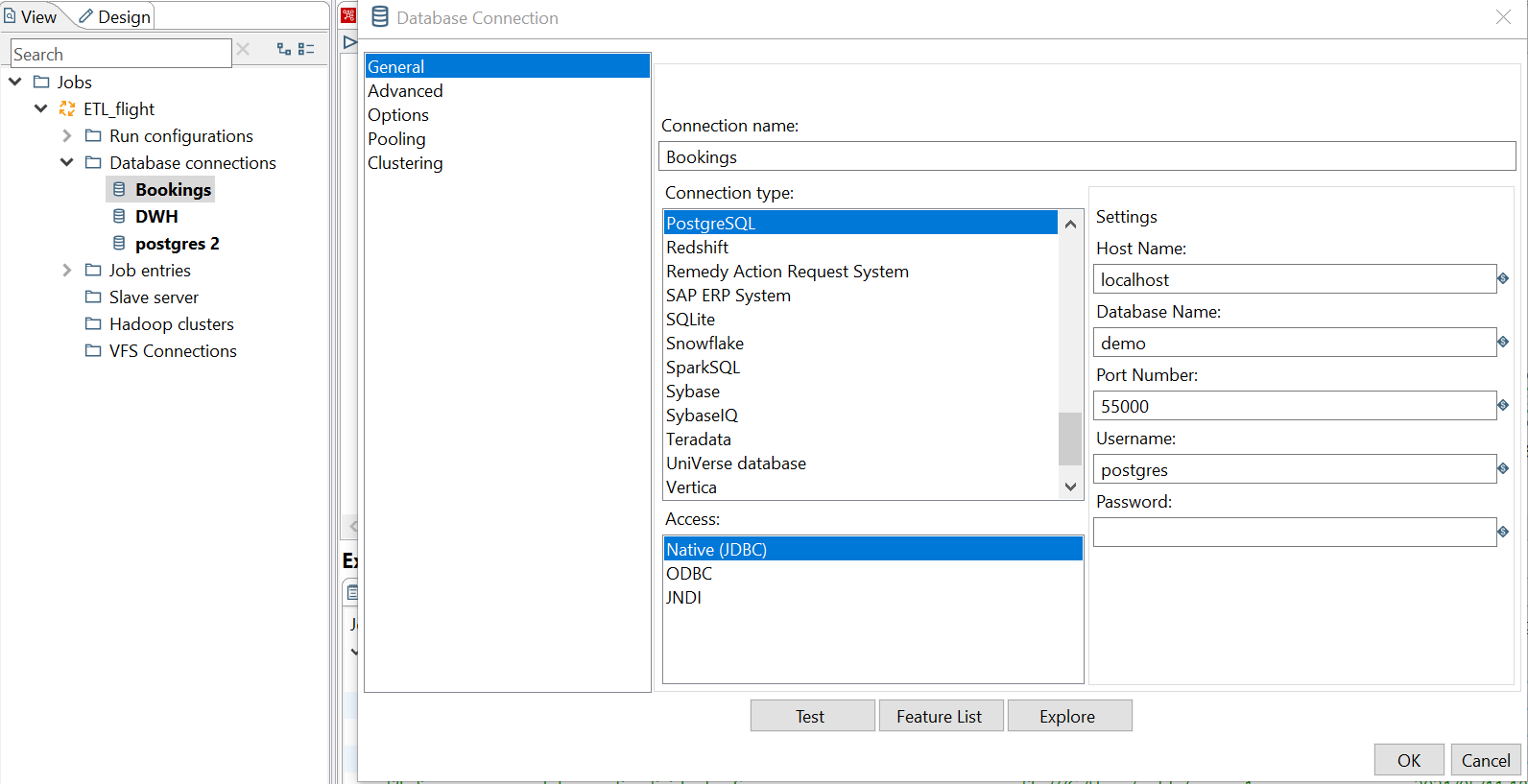
**“DWH ”**

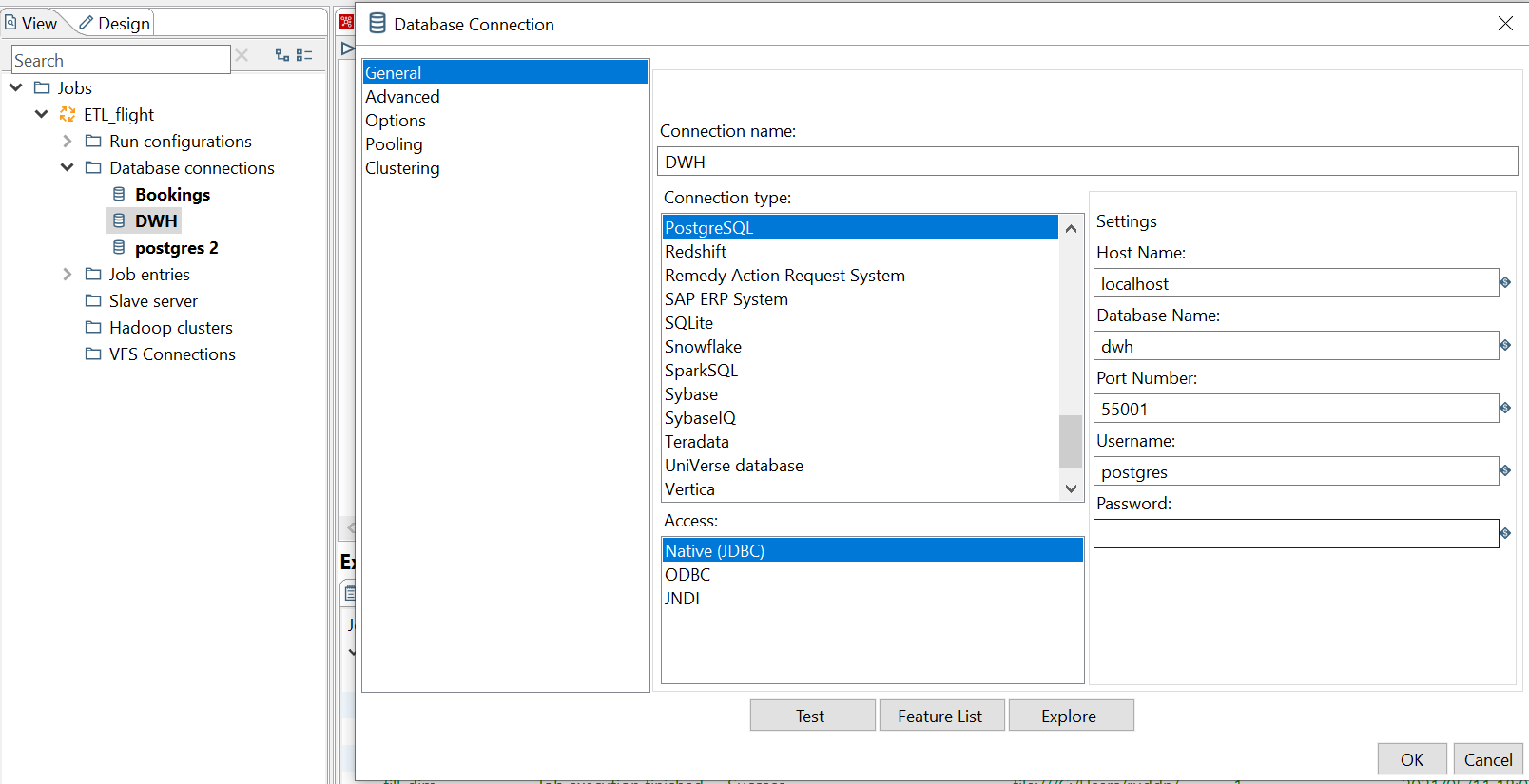
Выполнил: Руди Д.Н

май, 2021

1. Для подключения к серверу БД использовались контейнеры docker.
2. В работе использовалось два подключения к серверу БД – одно для доступа к базе данных **bookings**, содержащей источники данных, второе для подключения к хранилищу данных **dwh**







1. Краткое описание БД

Для реализации хранилища было создано три схемы:

* dds – хранилище данных (таблица фактов и справочники - DWH)
* rejected – хранилище для данных, не прошедших проверку качества (DQ)
* stage – таблицы для хранениях «очищенных» (прошедших DQ) данных, предназначенных для загрузки в хранилище (DWH)

Схема dds содержит:

* 1. Таблица фактов dds.Fact\_Flights

id serial **not** **null** **primary** **key**,

flight\_id **char**(6) **not** **null**,

pass\_id **int** **not** **null** **references** dds.Dim\_Passengers (id),

date\_departure\_id **int** **not** **null** **references** dds.Dim\_Calendar (id),

date\_arrival\_id **int** **not** **null** **references** dds.Dim\_Calendar (id),

actual\_departure **timestamp**,

actual\_arrival **timestamp**,

delay\_departure **int**, --timestamp,

delay\_arrival **int**, --timestamp,

arrival\_airport\_id **int** **not** **null** **references** dds.Dim\_Airports (airport\_id),

departure\_airport\_id **int** **not** **null** **references** dds.Dim\_Airports (airport\_id),

aircraft\_id **int** **not** **null** **references** dds.Dim\_Aircrafts (aircraft\_id),

tariff **int** **not** **null** **references** dds.Dim\_Tariff(id),

price **float8**

* 1. Справочник dds.Dim\_Passengers

id serial **not** **null** **primary** **key**,

passenger\_id **varchar**(20) **not** **null**,

passenger\_name **varchar**(20) **not** **null**,

ticket\_no **bpchar**

* 1. Справочник dds.Dim\_Aircrafts

aircraft\_id serial **primary** **key**,

aircraft\_code **varchar**(4) **not** **null**,

model **varchar**(50) **not** **null**,

aircraft\_range **int** **not** **null**

* 1. Справочник dds.Dim\_Airports

airport\_id serial **primary** **key**,

airport\_code **varchar** (5) **not** **null**,

airport\_name **varchar**(50) **not** **null**,

city **varchar**(100) **not** **null**

* 1. Справочник dds.Dim\_Tariff

id serial **not** **null** **primary** **key**,

fare\_conditions **varchar**(10) **not** **nul**

* 1. Справочник dds.Dim\_Calendar

id **int**,

dt **date**,

ansi\_date **varchar**,

**day int**,

week\_number **int**,

**month int**,

**year int** ,

week\_day **int**

ER – диаграмма схемы dds:

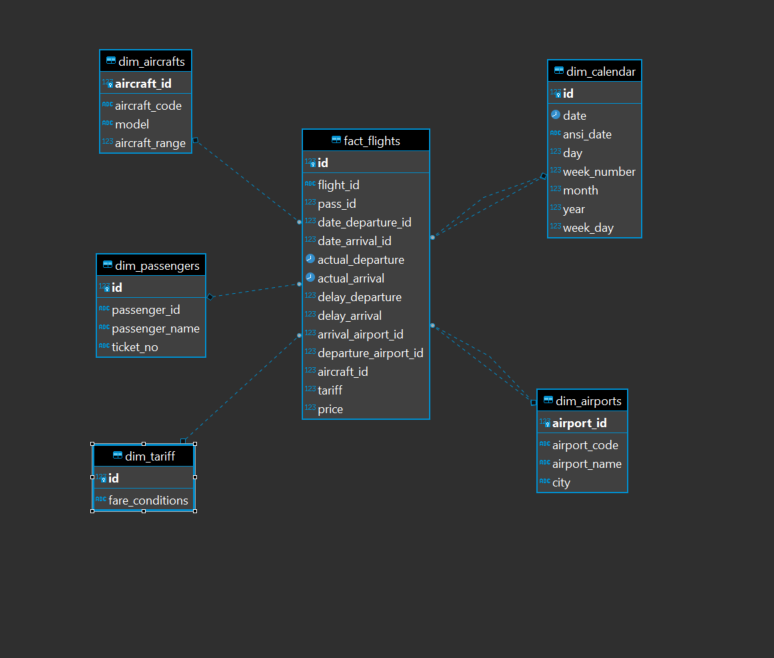


Схема rejected содержит:

* 1. Таблица rejected.Passengers

passenger\_id **varchar**,

passenger\_name **varchar**,

ticket\_no **bpchar**

* 1. Таблица rejected.Dim\_Aircrafts

aircraft\_code **varchar**,

model **varchar**,

aircraft\_range **int**

* 1. Таблица rejected.Dim\_Airports

airport\_code **varchar**,

airport\_name **varchar**,

city **varchar**

Схема stage содержит:

* 1. Справочник stage.Dim\_Passengers

passenger\_id **varchar**(20) **not** **null**,

passenger\_name **varchar**(20) **not** **null**,

ticket\_no **bpchar**

* 1. Справочник stage.Dim\_Aircrafts

aircraft\_code **varchar**(4) **not** **null**,

model **varchar**(50) **not** **null**,

aircraft\_range **int** **not** **null**

* 1. Справочник stage.Dim\_Airports

airport\_code **varchar** (5) **not** **null**,

airport\_name **varchar**(50) **not** **null**,

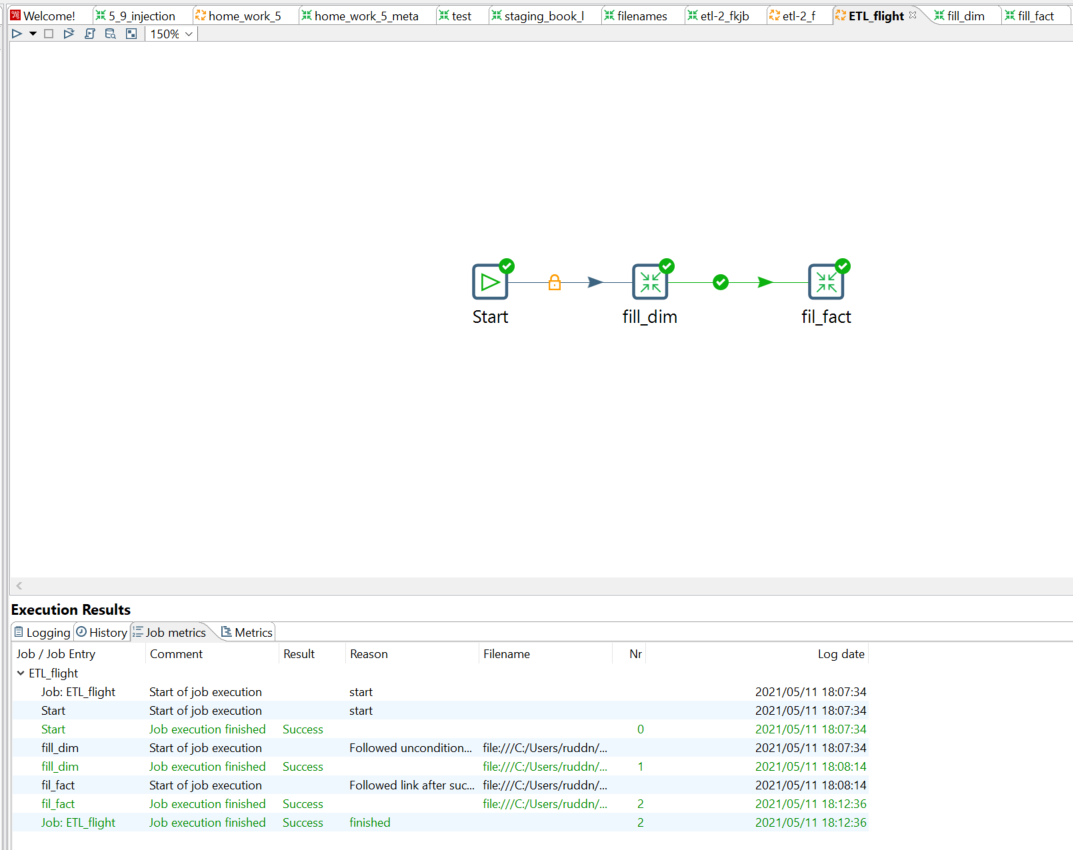
city **varchar**(100) **not** **null**

* 1. Справочник stage.Dim\_Tariff

fare\_conditions **varchar**(10) **not** **nul**

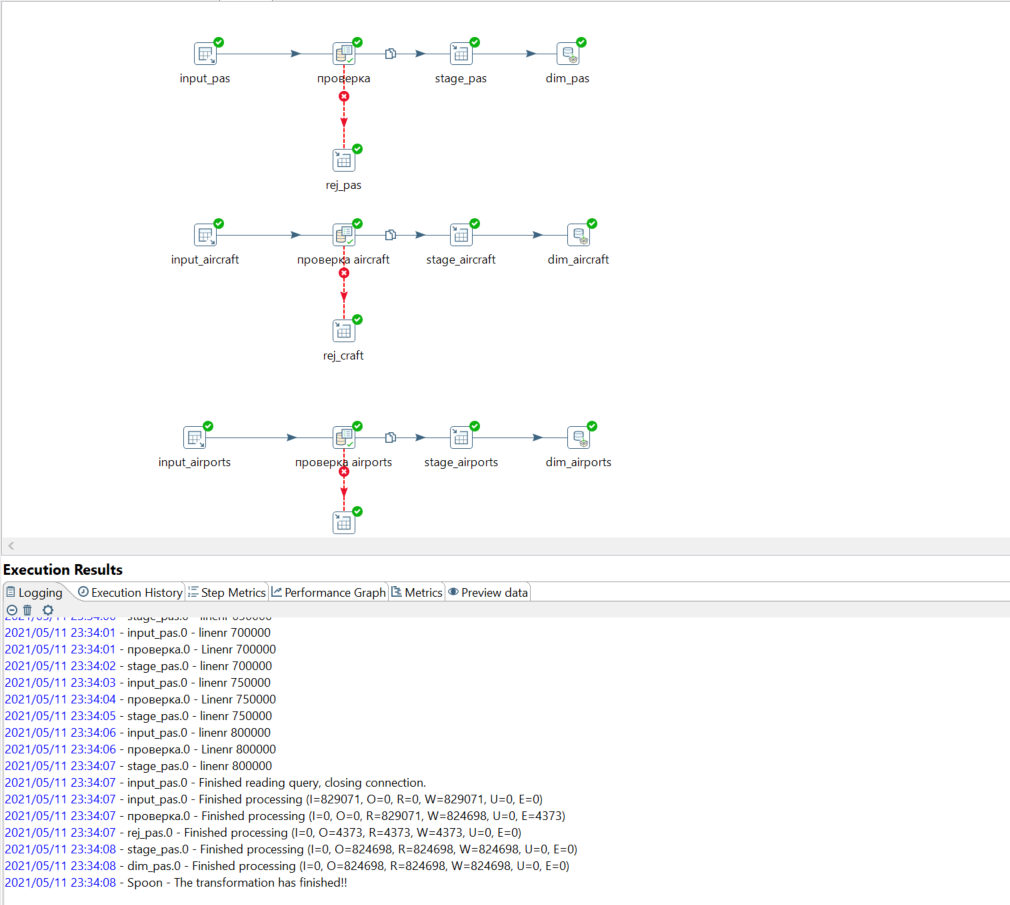
Реализация скрипта создания перечисленных таблиц приведена в приложении к данной проектной работе в виде SQL-скрипта «Итоговая\_ETL\_DDS.sql»

1. Краткое описание ETL процесса



Процесс загрузки данных в хранилище представляет из себя шаги из двух трансформаций, содержащих под трансформации, запускаемые при выполнении «работы».

* 1. Первая трансформация предназначена для загрузки справочников-измерений dds.Dim\_Passengers , dds.Dim\_Aircrafts, dds.Dim\_Airports , dds.Dim\_Tariff и содержит следующие шаги:

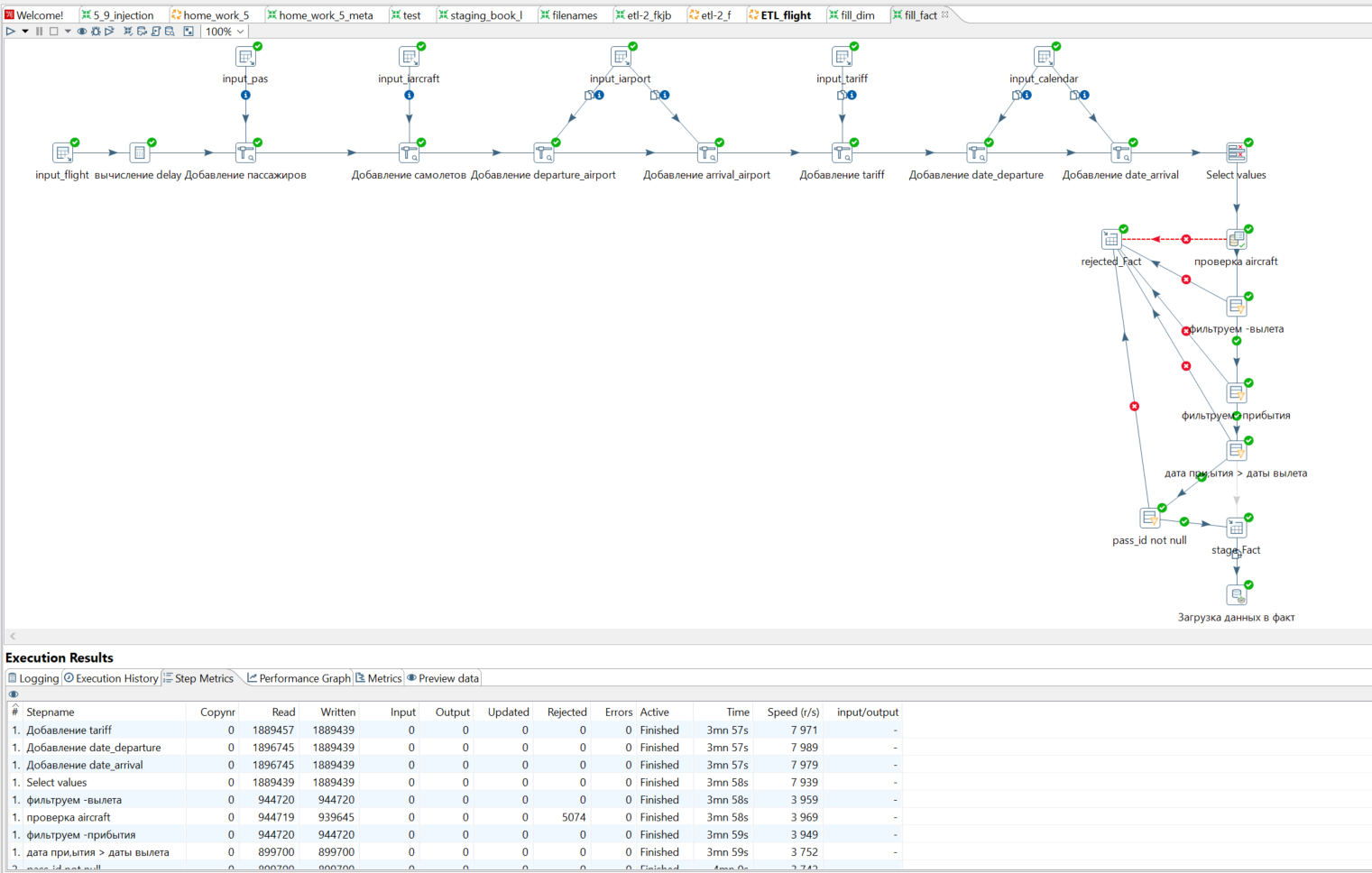


* + 1. Первый шаг - загрузка данных из таблиц источника
    2. Второй – реализует проверку на заполнение обязательных полей, записи, содержащие не заполненные поля отклоняются и сохраняются в специальные таблице в хранилище отклонённых данных rejected, для последующей их обработки специалистами
    3. Валидные данные сохраняются в хранилище валидных данных stage с помощью шага TableOutput. В хранилище данные сохраняются без ограничений первичных ключей, внешних ссылок, индексов.
    4. На последнем шаге валидные данные загружаются из хранилища stage в витрину данных dwh с помощью шага PostgreSQL bulk loader без проверок.

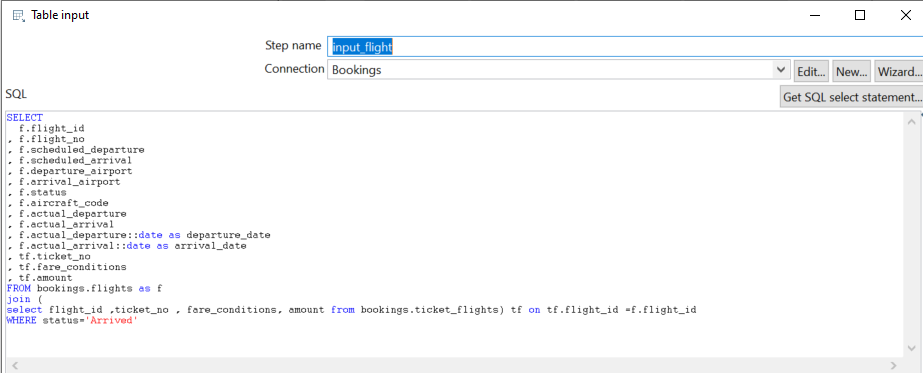
Третий и четвёртый шаги предназначены для ускорения загрузки данных в хранилище.

Загрузка справочников реализована в виде отдельных pipelines, которые запускаются и выполняются параллельно, для ускорения работы трансформации.

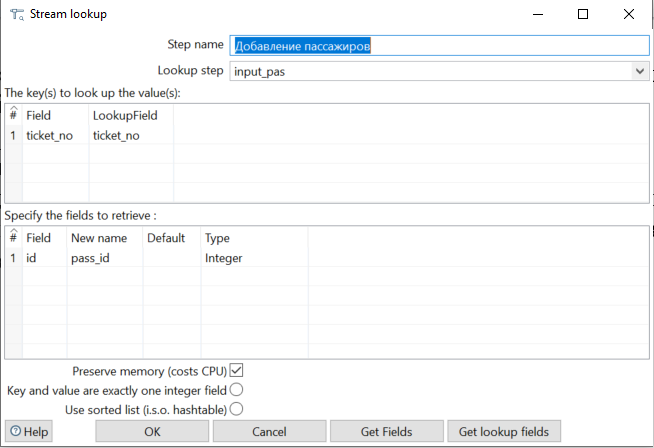
* 1. Следующий шаг – загрузка данных в таблица фактов:

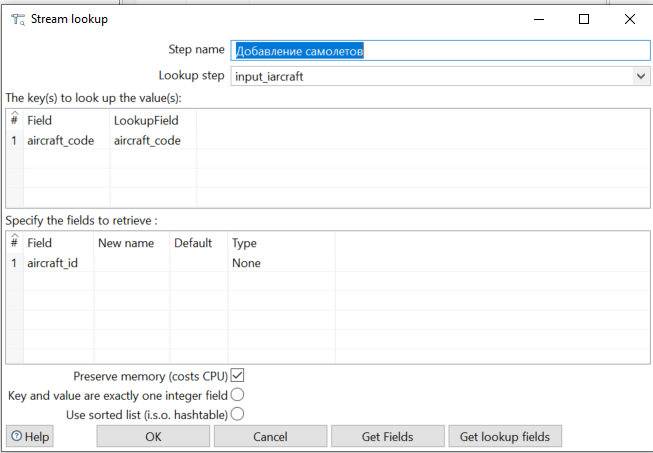


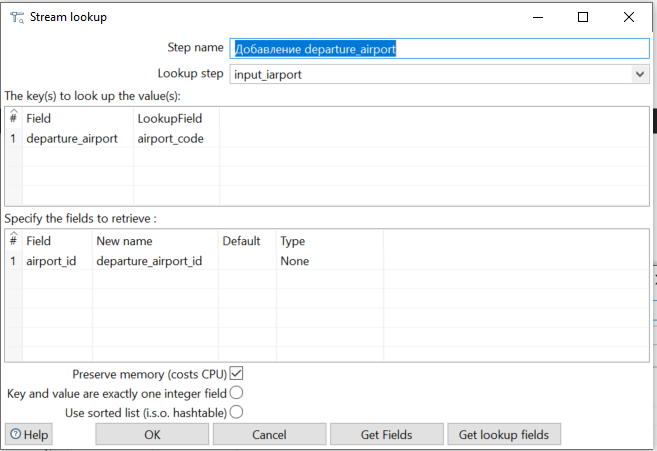
* + 1. Загрузка «обогащенных» данных из таблицы источника Flights объединенной с таблицей Ticket\_ Flights, т.к. таблица-источник не содержит прямой ссылки на пассажиров (passenger\_id).

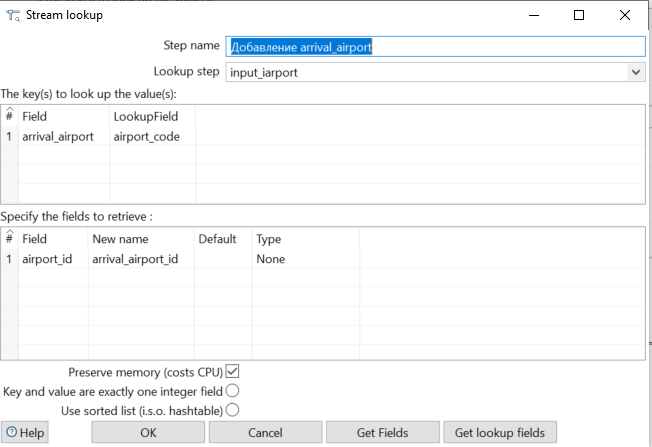


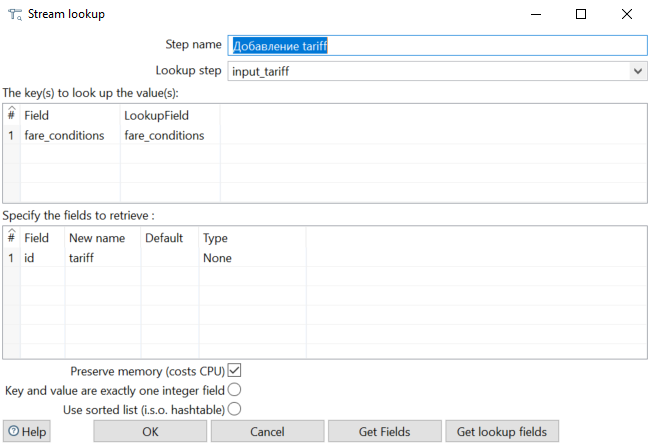
* + 1. На втором шаге вычисляются задержки вылета и прилета рейсов
    2. На последующих шагах последовательно осуществляется добавление полей, содержащих ID - ссылки на измерения в будущую таблицу фактов с помощью шагов Stream Lookup

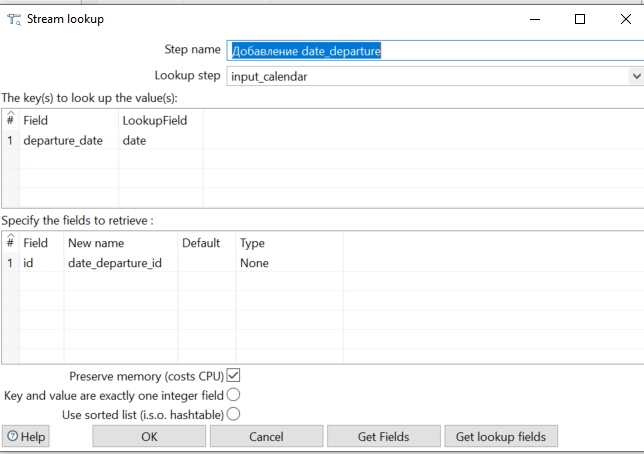


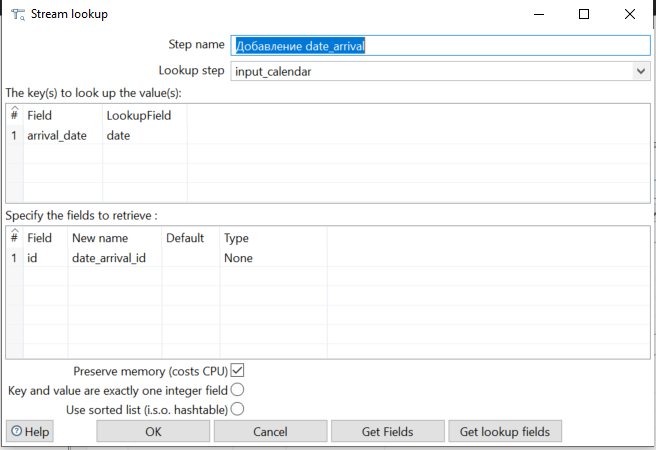




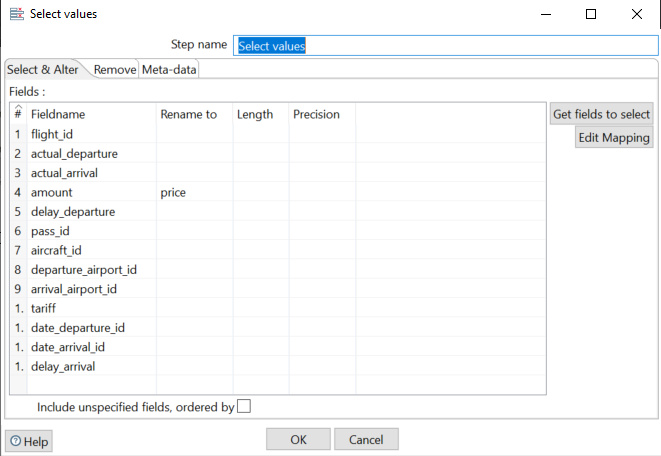








* + 1. Далее выполняется отбор и маппинг полей для записи в БД



* + 1. Последующие шаги предназначены для проверки качества данных в потоке. Шаги проверки:
  + на заполнение обязательных полей
  + задержка вылета отрицательная
  + задержка прибытия отрицательная
  + дата прибытия больше даты вылета

Данные, прошедшие проверку, загружаются в хранилище stage, данные не прошедшие DQ загружаются в хранилище rejected

* + 1. Валидные данные загружаются в таблицу фактов dds.Fact\_Flights

1. К отчету прилагаются следующие файлы:

|  |  |
| --- | --- |
| **Файл** | **Описание** |
| «Итоговая ETL DDS\_2.sql» | Скрипт создания схем и таблиц из пункта 3 |
| ETL\_flight.kjb | Файл, содержащий job, запускающий всю трансформацию |
| fill\_dim.ktr | Трансформация для загрузки справочников |
| fill\_fact.ktr | Трансформация для загрузки таблицы фактов |
| ER-диаграмма.png | Скрин ER-диаграммы схемы хранилища |
| Общая схема процесса.png | Скрин общей схемы процесса |
| Загрузка справочников.png | Скрин схемы загрузки справочников |
| Загрузка факта.png | Скрин схемы загрузки таблицы фактов |