Курсовая DWH

1. SQL-скрипт создания таблицы фактов и всех справочников

**CREATE** **TABLE** bookings.aircrafts\_data (

aircraft\_code **bpchar**(3) **NOT** **NULL**,

model **jsonb** **NOT** **NULL**,

"range" **int4** **NOT** **NULL**,

**CONSTRAINT** aircrafts\_pkey **PRIMARY** **KEY** (aircraft\_code),

**CONSTRAINT** aircrafts\_range\_check **CHECK** ((**range** > 0))

);

**CREATE** **TABLE** bookings.airports\_data (

airport\_code **bpchar**(3) **NOT** **NULL**,

airport\_name **jsonb** **NOT** **NULL**,

city **jsonb** **NOT** **NULL**,

coordinates point **NOT** **NULL**,

timezone **text** **NOT** **NULL**,

**CONSTRAINT** airports\_data\_pkey **PRIMARY** **KEY** (airport\_code)

);

**CREATE** **TABLE** bookings.boarding\_passes (

ticket\_no **bpchar**(13) **NOT** **NULL**,

flight\_id **int4** **NOT** **NULL**,

boarding\_no **int4** **NOT** **NULL**,

seat\_no **varchar**(4) **NOT** **NULL**,

**CONSTRAINT** boarding\_passes\_flight\_id\_boarding\_no\_key **UNIQUE** (flight\_id, boarding\_no),

**CONSTRAINT** boarding\_passes\_flight\_id\_seat\_no\_key **UNIQUE** (flight\_id, seat\_no),

**CONSTRAINT** boarding\_passes\_pkey **PRIMARY** **KEY** (ticket\_no, flight\_id),

**CONSTRAINT** boarding\_passes\_ticket\_no\_fkey **FOREIGN** **KEY** (ticket\_no,flight\_id) **REFERENCES** bookings.ticket\_flights(ticket\_no,flight\_id)

);

**CREATE** **TABLE** bookings.bookings (

book\_ref **bpchar**(6) **NOT** **NULL**,

book\_date **timestamptz** **NOT** **NULL**,

total\_amount **numeric**(10, 2) **NOT** **NULL**,

**CONSTRAINT** bookings\_pkey **PRIMARY** **KEY** (book\_ref)

);

**CREATE** **TABLE** bookings.flights (

flight\_id serial4 **NOT** **NULL**,

flight\_no **bpchar**(6) **NOT** **NULL**,

scheduled\_departure **timestamptz** **NOT** **NULL**,

scheduled\_arrival **timestamptz** **NOT** **NULL**,

departure\_airport **bpchar**(3) **NOT** **NULL**,

arrival\_airport **bpchar**(3) **NOT** **NULL**,

status **varchar**(20) **NOT** **NULL**,

aircraft\_code **bpchar**(3) **NOT** **NULL**,

actual\_departure **timestamptz** **NULL**,

actual\_arrival **timestamptz** **NULL**,

**CONSTRAINT** flights\_check **CHECK** ((scheduled\_arrival > scheduled\_departure)),

**CONSTRAINT** flights\_check1 **CHECK** (((actual\_arrival **IS** **NULL**) **OR** ((actual\_departure **IS** **NOT** **NULL**) **AND** (actual\_arrival **IS** **NOT** **NULL**) **AND** (actual\_arrival > actual\_departure)))),

**CONSTRAINT** flights\_flight\_no\_scheduled\_departure\_key **UNIQUE** (flight\_no, scheduled\_departure),

**CONSTRAINT** flights\_pkey **PRIMARY** **KEY** (flight\_id),

**CONSTRAINT** flights\_status\_check **CHECK** (((status)::**text** = **ANY** (**ARRAY**[('On Time'::**character** **varying**)::**text**, ('Delayed'::**character** **varying**)::**text**, ('Departed'::**character** **varying**)::**text**, ('Arrived'::**character** **varying**)::**text**, ('Scheduled'::**character** **varying**)::**text**, ('Cancelled'::**character** **varying**)::**text**]))),

**CONSTRAINT** flights\_aircraft\_code\_fkey **FOREIGN** **KEY** (aircraft\_code) **REFERENCES** bookings.aircrafts\_data(aircraft\_code),

**CONSTRAINT** flights\_arrival\_airport\_fkey **FOREIGN** **KEY** (arrival\_airport) **REFERENCES** bookings.airports\_data(airport\_code),

**CONSTRAINT** flights\_departure\_airport\_fkey **FOREIGN** **KEY** (departure\_airport) **REFERENCES** bookings.airports\_data(airport\_code)

);

**CREATE** **TABLE** bookings.seats (

aircraft\_code **bpchar**(3) **NOT** **NULL**,

seat\_no **varchar**(4) **NOT** **NULL**,

fare\_conditions **varchar**(10) **NOT** **NULL**,

**CONSTRAINT** seats\_fare\_conditions\_check **CHECK** (((fare\_conditions)::**text** = **ANY** (**ARRAY**[('Economy'::**character** **varying**)::**text**, ('Comfort'::**character** **varying**)::**text**, ('Business'::**character** **varying**)::**text**]))),

**CONSTRAINT** seats\_pkey **PRIMARY** **KEY** (aircraft\_code, seat\_no),

**CONSTRAINT** seats\_aircraft\_code\_fkey **FOREIGN** **KEY** (aircraft\_code) **REFERENCES** bookings.aircrafts\_data(aircraft\_code) **ON** **DELETE** **CASCADE**

);

**CREATE** **TABLE** bookings.ticket\_flights (

ticket\_no **bpchar**(13) **NOT** **NULL**,

flight\_id **int4** **NOT** **NULL**,

fare\_conditions **varchar**(10) **NOT** **NULL**,

amount **numeric**(10, 2) **NOT** **NULL**,

**CONSTRAINT** ticket\_flights\_amount\_check **CHECK** ((amount >= (0)::**numeric**)),

**CONSTRAINT** ticket\_flights\_fare\_conditions\_check **CHECK** (((fare\_conditions)::**text** = **ANY** (**ARRAY**[('Economy'::**character** **varying**)::**text**, ('Comfort'::**character** **varying**)::**text**, ('Business'::**character** **varying**)::**text**]))),

**CONSTRAINT** ticket\_flights\_pkey **PRIMARY** **KEY** (ticket\_no, flight\_id),

**CONSTRAINT** ticket\_flights\_flight\_id\_fkey **FOREIGN** **KEY** (flight\_id) **REFERENCES** bookings.flights(flight\_id),

**CONSTRAINT** ticket\_flights\_ticket\_no\_fkey **FOREIGN** **KEY** (ticket\_no) **REFERENCES** bookings.tickets(ticket\_no)

);

**CREATE** **TABLE** bookings.tickets (

ticket\_no **bpchar**(13) **NOT** **NULL**,

book\_ref **bpchar**(6) **NOT** **NULL**,

passenger\_id **varchar**(20) **NOT** **NULL**,

passenger\_name **text** **NOT** **NULL**,

contact\_data **jsonb** **NULL**,

**CONSTRAINT** tickets\_pkey **PRIMARY** **KEY** (ticket\_no),

**CONSTRAINT** tickets\_book\_ref\_fkey **FOREIGN** **KEY** (book\_ref) **REFERENCES** bookings.bookings(book\_ref)

);

**CREATE** **TABLE** bookings.dim\_calendar (

"date" **date** **NOT** **NULL**

);

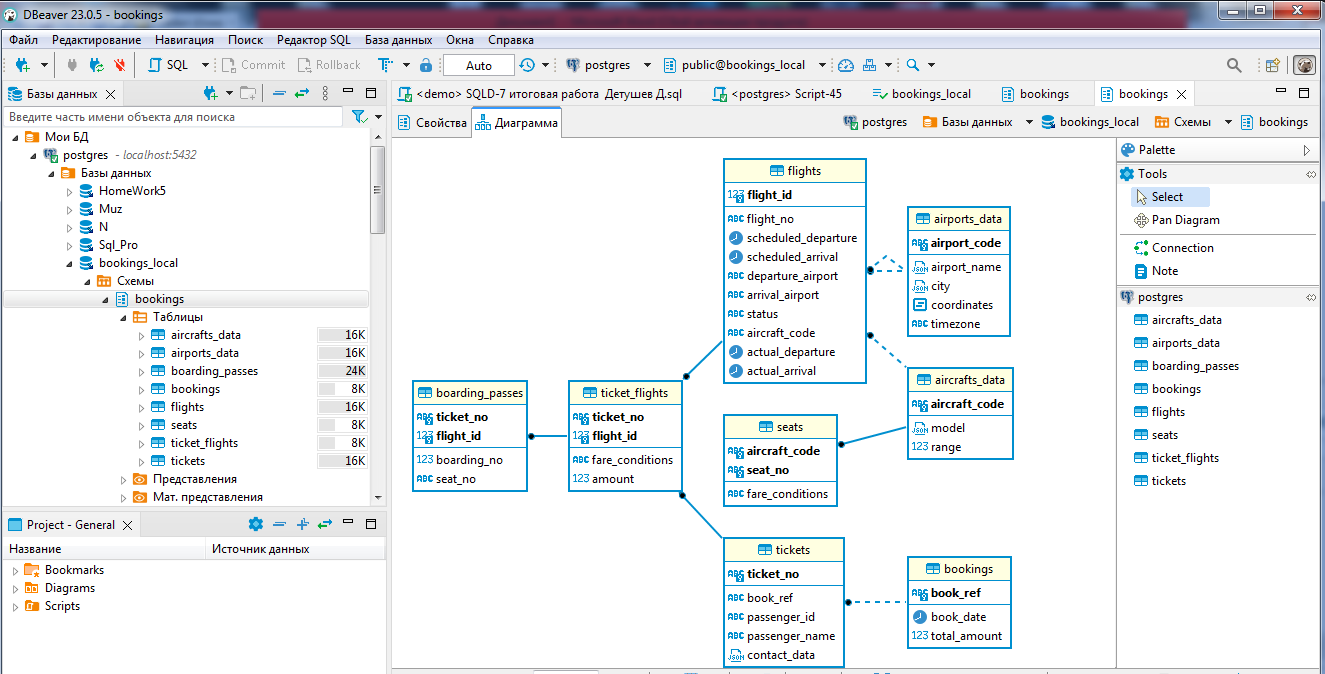


Диаграмма базы данных

--для создания колендаря дат создадим процедуру которая будет генерировать таблицу от заданной даты до текущей.

**create** **or** **replace** **procedure** insert\_dim\_calendar (date1 **date**) **as** **$$**

**declare**

i record;

date2 **date**;

**begin**

**for** i **in** (**select** **generate\_series**(

(date1)

,('now ()')

, '1 day'::**interval**)::**date** **as** **date**)

**loop**

date2 = i.**date**;

**insert** **into** dim\_calendar (**date**) **values** (date2);

**end** **loop**;

**end**;

**$$** **language** plpgsql

**call** insert\_dim\_calendar ('2020-01-01')

**CREATE** **TABLE** bookings.dim\_passengers (

passenger\_id **varchar**(30) **NOT** **NULL**,

passenger\_name **varchar**(300) **NOT** **NULL**,

contact\_data **varchar**(300) **NOT** **NULL**,

**CONSTRAINT** dim\_passengers\_pkey **PRIMARY** **KEY** (passenger\_id)

);

**CREATE** **TABLE** bookings.dim\_aircrafts (

aircraft\_code **varchar**(3) **NOT** **NULL**,

model **varchar**(100) **NOT** **NULL**,

"range" **int4** **NOT** **NULL**,

**CONSTRAINT** dim\_aircrafts\_model\_key **UNIQUE** (model),

**CONSTRAINT** dim\_aircrafts\_pkey **PRIMARY** **KEY** (aircraft\_code),

**CONSTRAINT** dim\_aircrafts\_range\_check **CHECK** ((**range** > 0))

);

**CREATE** **TABLE** bookings.dim\_airports (

airport\_code **varchar**(3) **NOT** **NULL**,

airport\_name **varchar**(100) **NOT** **NULL**,

city **varchar**(100) **NOT** **NULL**,

timezone **varchar**(100) **NOT** **NULL**,

**CONSTRAINT** dim\_airports\_airport\_name\_key **UNIQUE** (airport\_name),

**CONSTRAINT** dim\_airports\_pkey **PRIMARY** **KEY** (airport\_code)

);

**CREATE** **TABLE** bookings.fact\_flights (

passenger **varchar**(150) **NULL**,

date\_dep **timestamp** **NULL**,

date\_arr **timestamp** **NULL**,

diff\_dep **varchar**(20) **NULL**,

diff\_arr **varchar**(20) **NULL**,

model **text** **NULL**,

airport\_dep **varchar**(50) **NULL**,

airport\_arr **varchar**(50) **NULL**,

fare\_condition **varchar**(20) **NULL**,

amount **numeric**(10, 2) **NULL**

);

**CREATE** **TABLE** bookings.dim\_tariff (

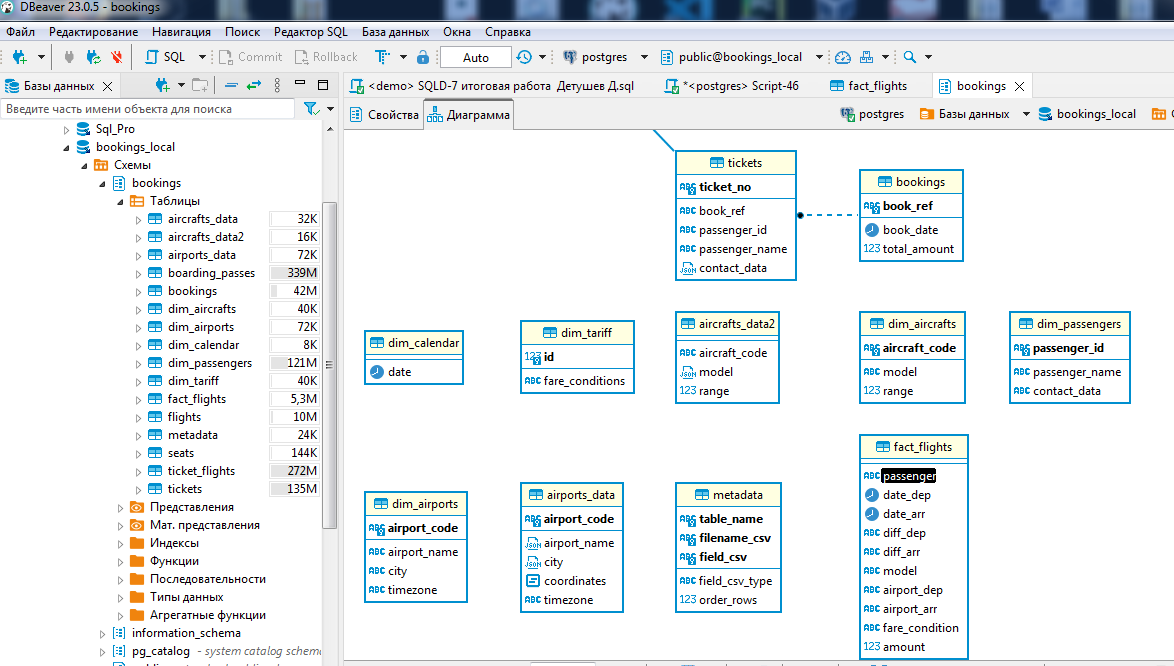
id serial4 **NOT** **NULL**,

fare\_conditions **varchar**(50) **NOT** **NULL**,

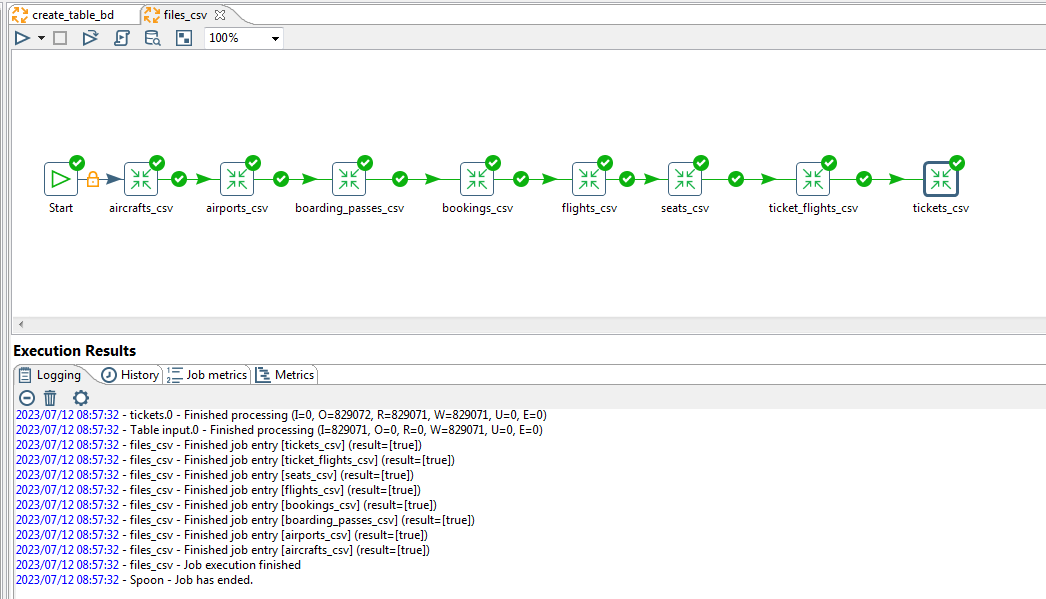
**CONSTRAINT** dim\_tariff\_fare\_conditions\_key **UNIQUE** (fare\_conditions),

**CONSTRAINT** dim\_tariff\_pkey **PRIMARY** **KEY** (id)

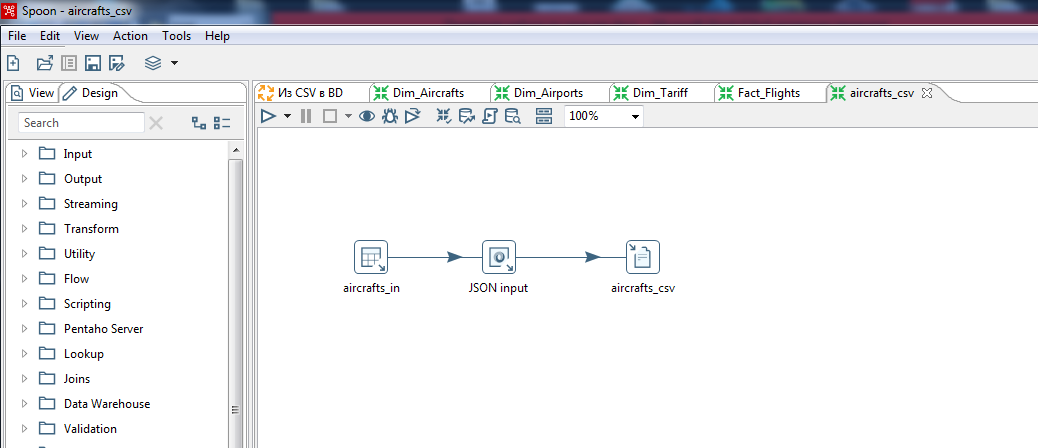
);

Таблицы фактов и всех справочников

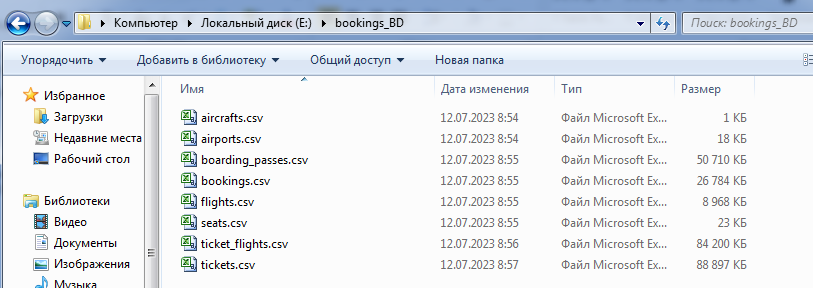
1. Трансформации



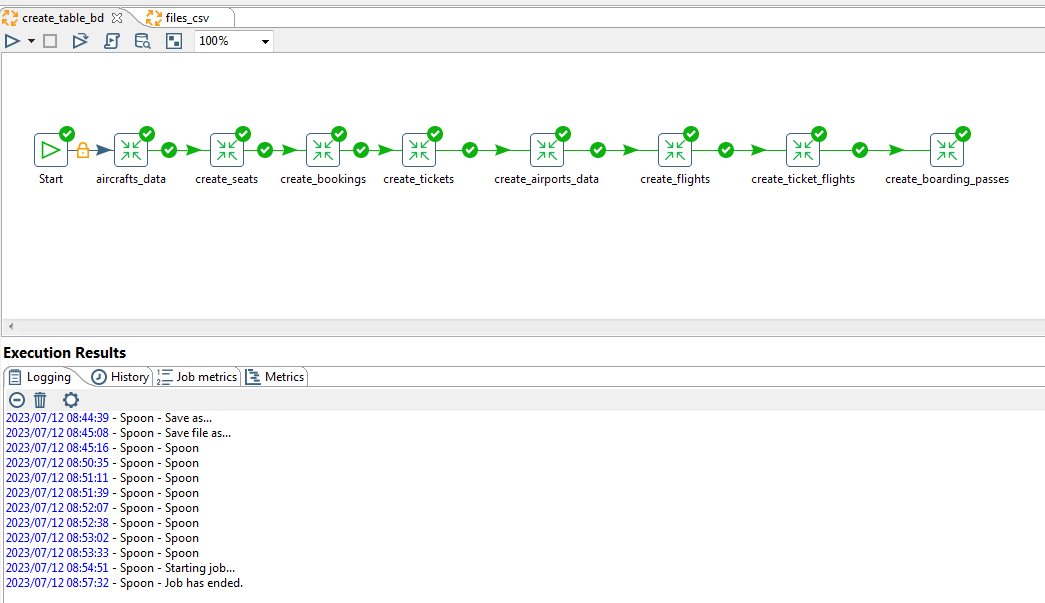
Работа с трансформациями сохранения таблиц с фактами и размерностями в csv файлы



Трансформация на примере сохранения данных в файл aircrafts\_csv



csv файлы



Работа с трансформациями создания таблиц базы данных

Для автоматизации процесса воспользуемся методом работы с метаданными. Для этого создадим необходимую таблицу

--Создание таблицы с метаданными

**CREATE** **TABLE** bookings."metadata" (

table\_name **varchar**(20) **NOT** **NULL**,

filename\_csv **varchar**(50) **NOT** **NULL**,

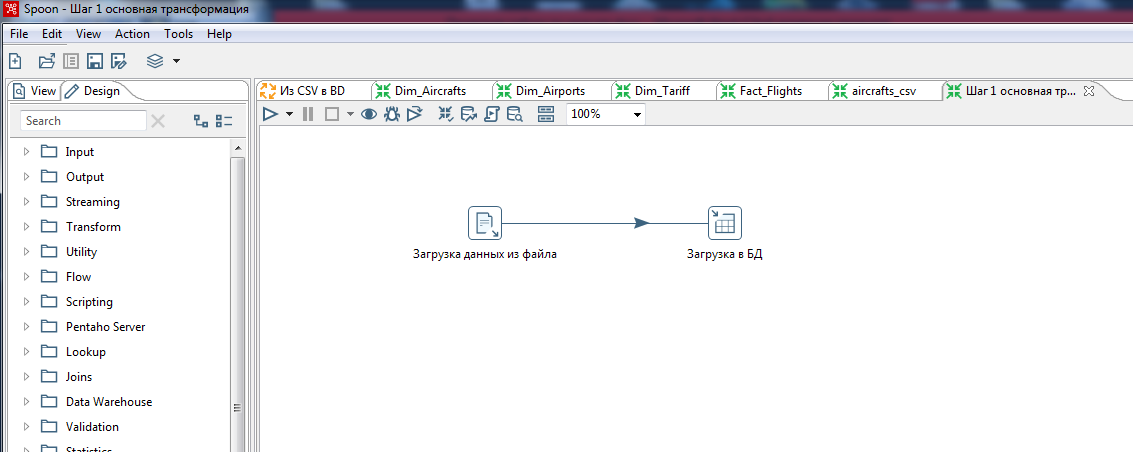
field\_csv **varchar**(50) **NULL**,

field\_csv\_type **varchar**(20) **NULL**,

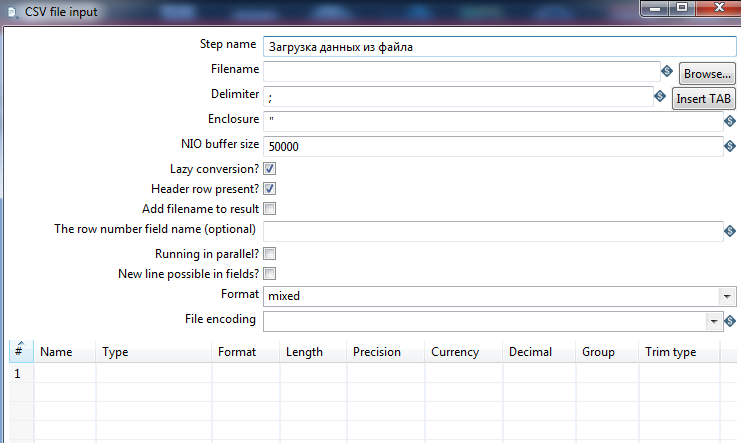
order\_rows **int2** **NULL**,

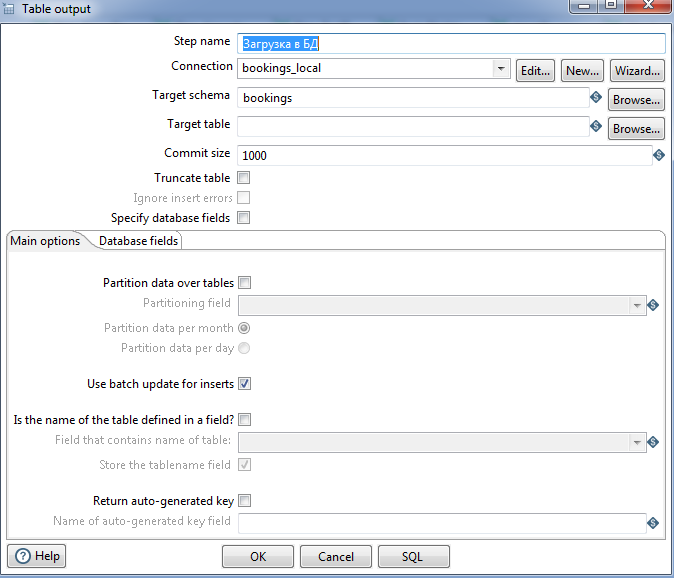
**CONSTRAINT** csv\_pkey **PRIMARY** **KEY** (table\_name, filename\_csv, field\_csv)

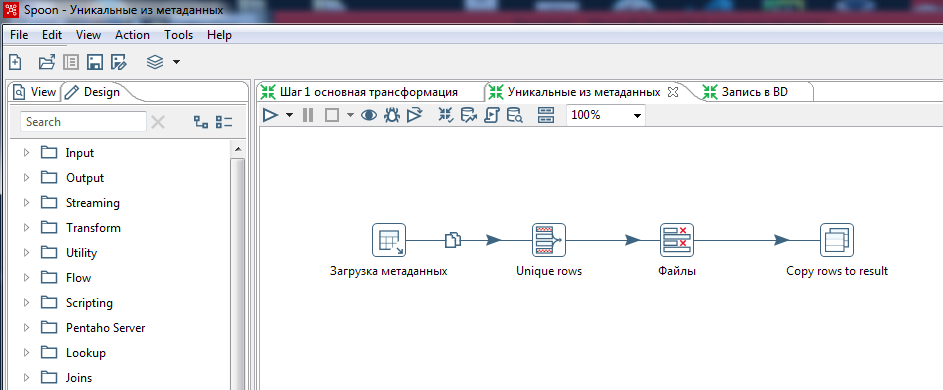
);



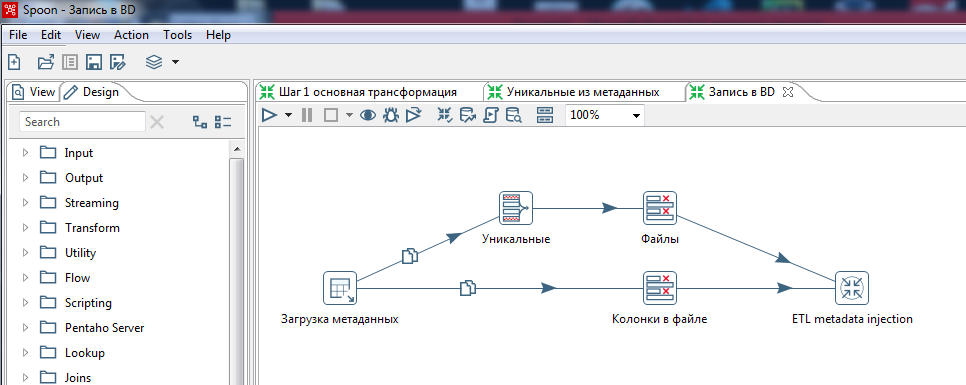
Основная трансформация



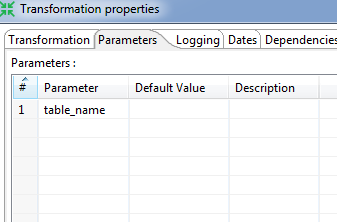




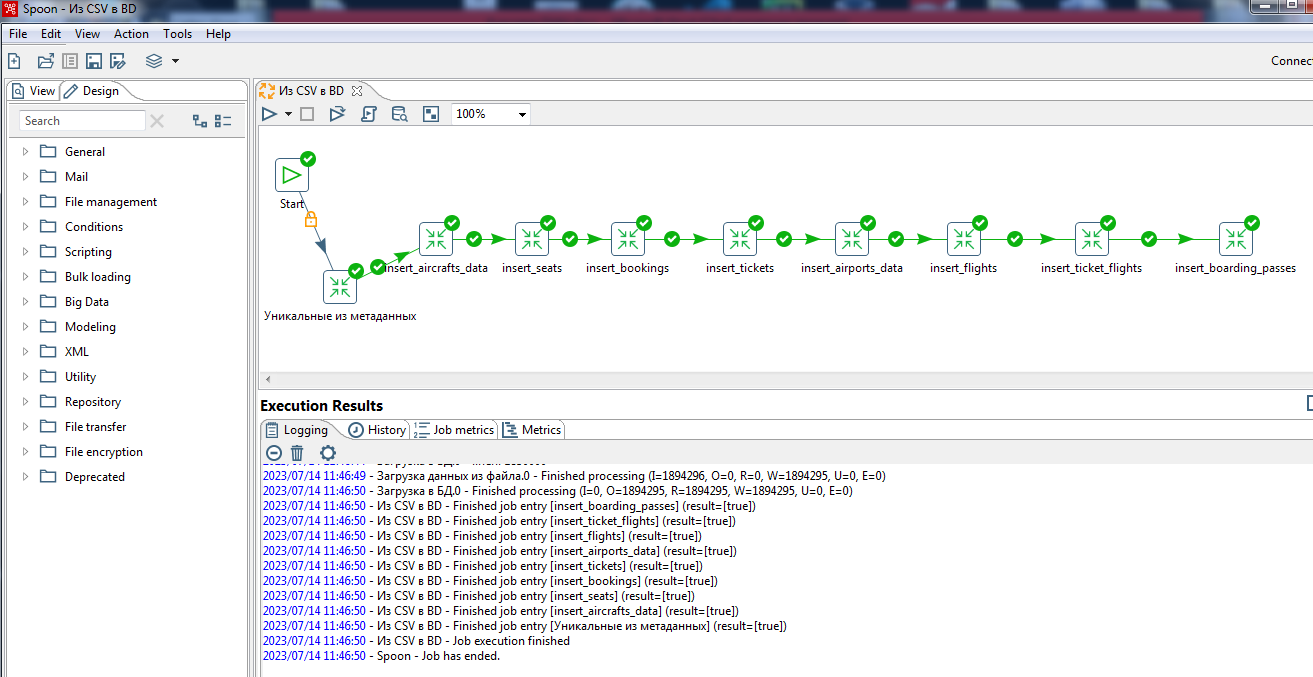
Уникальные из метаданных



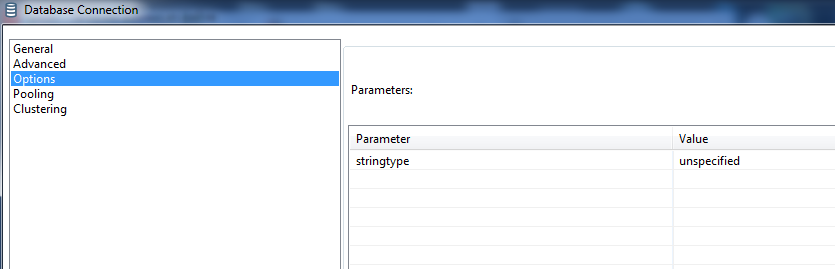
Запись в BD



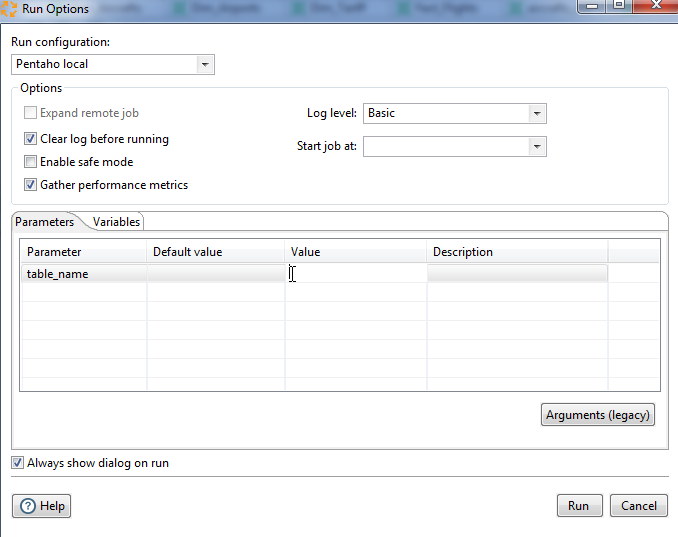
Для всех трансформаций созадим переменную table\_name



Работа с трансформациями сохранения данных в таблицы базы данных

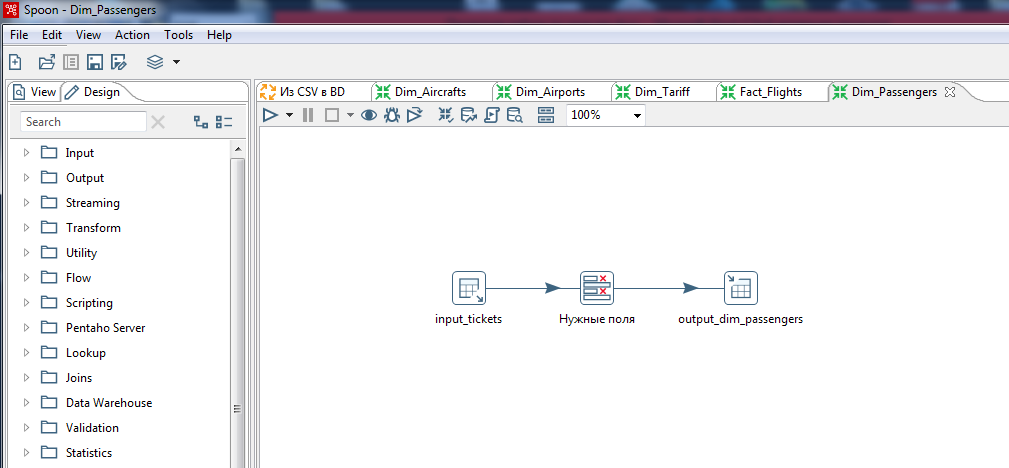


Для конвертации данных в тип JSON во вкладке опции установим параметр

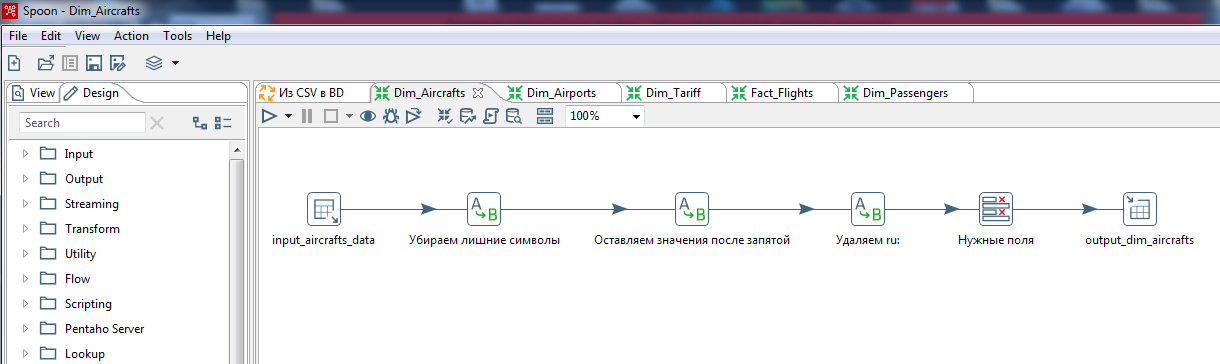


Теперь всё что нам нужно это при запуске работы записать наименование таблицы в базе данных в соответствии с очерёдностью

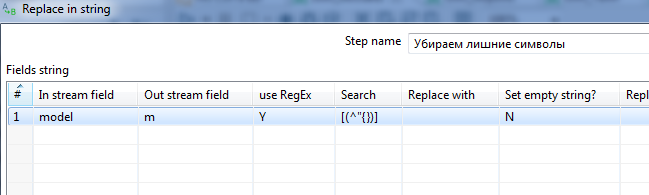
1. Трансформации сохранения данных в справочники

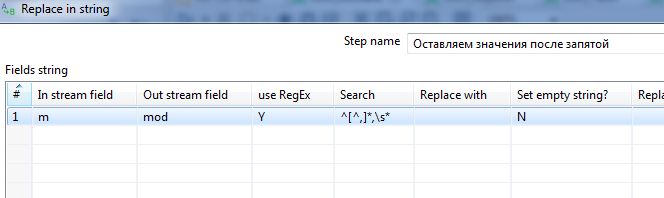


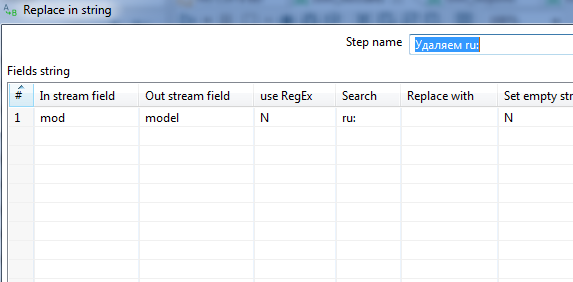
Dim\_Passengers

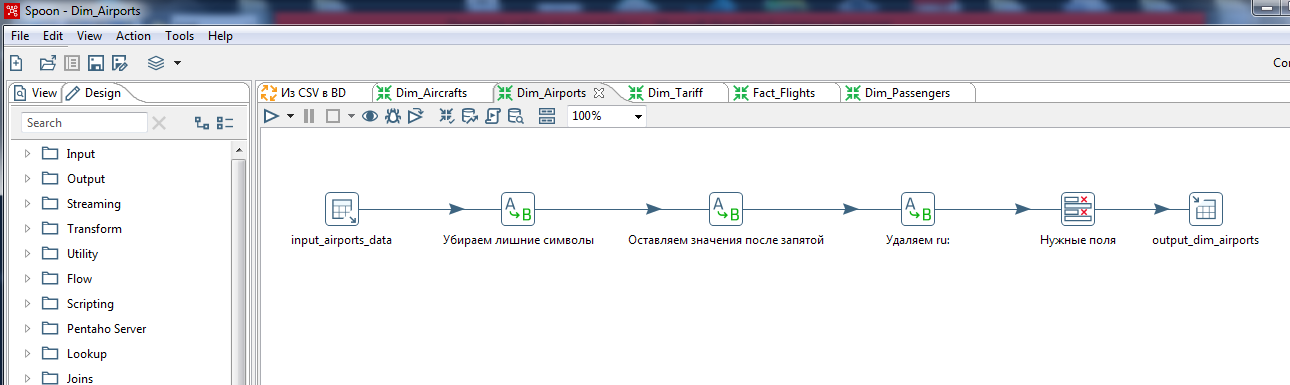


Dim\_Aircrafts

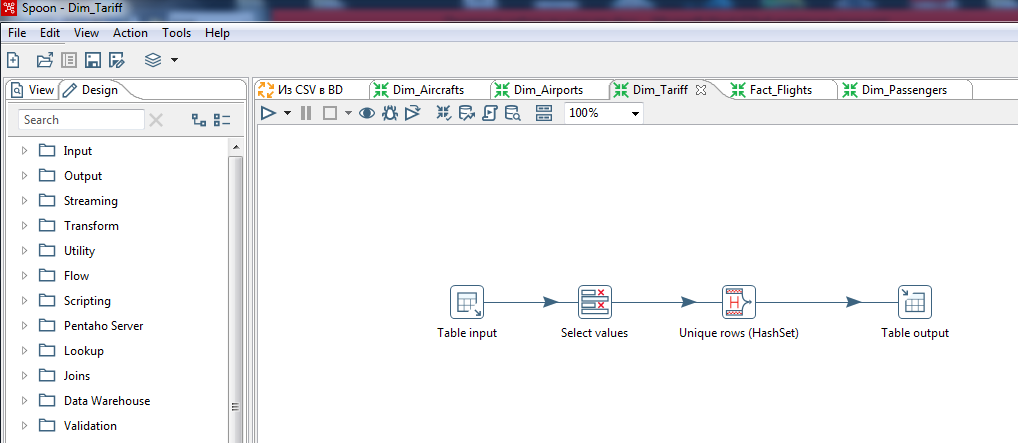




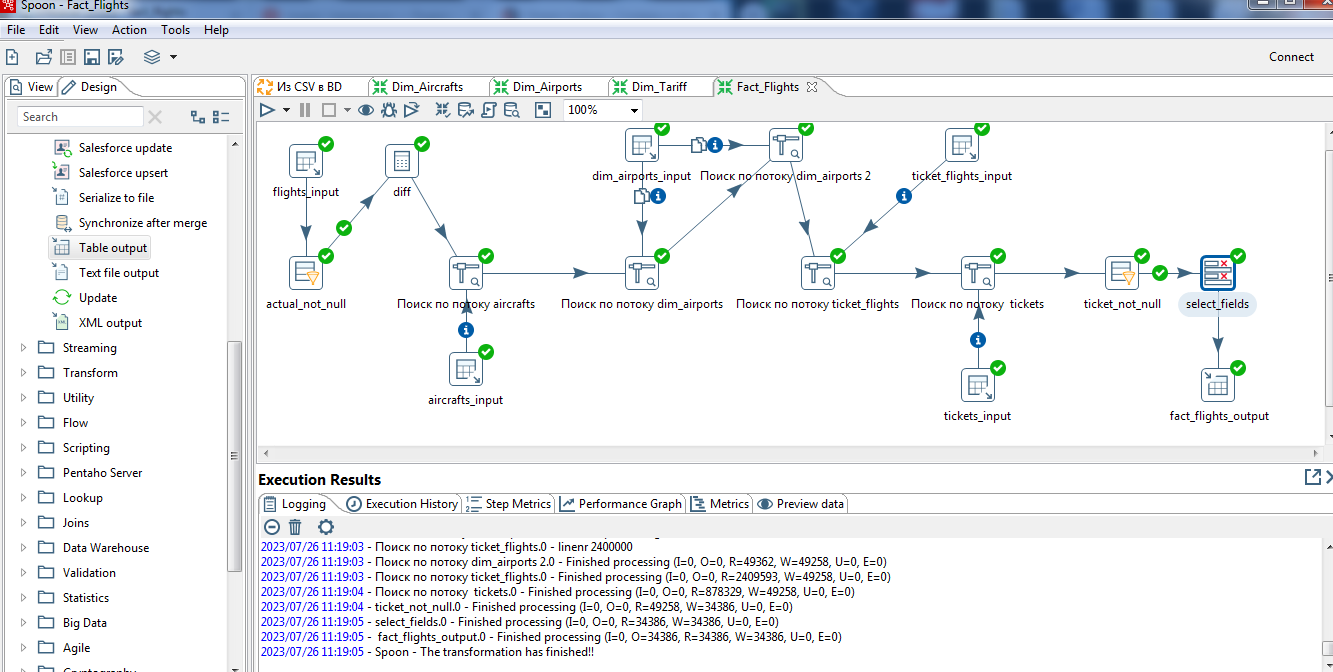




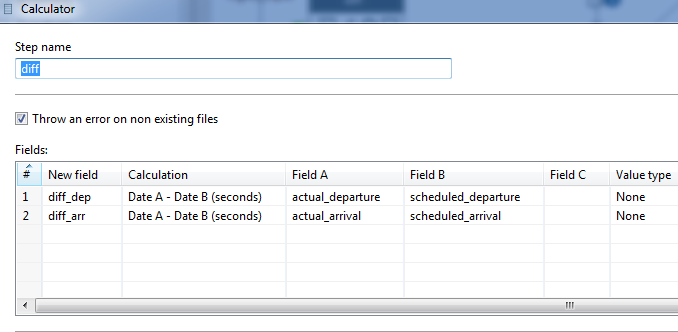
Dim\_Airports

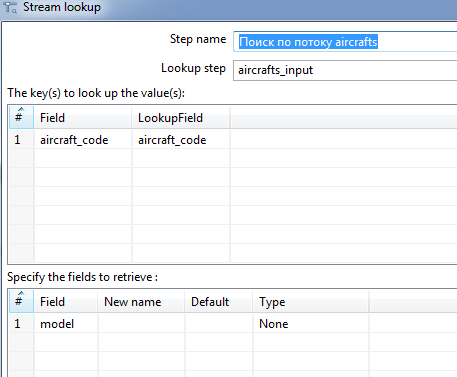


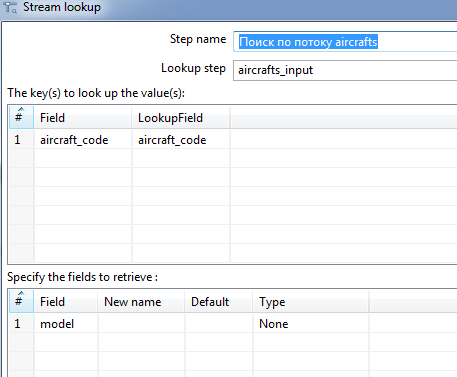
Dim\_Tariff

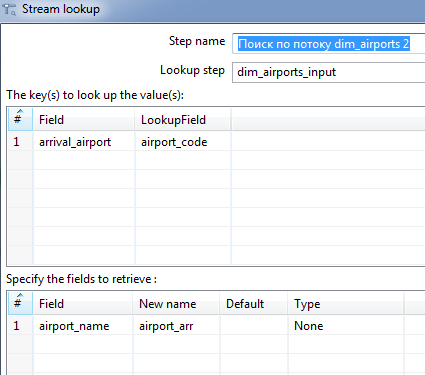


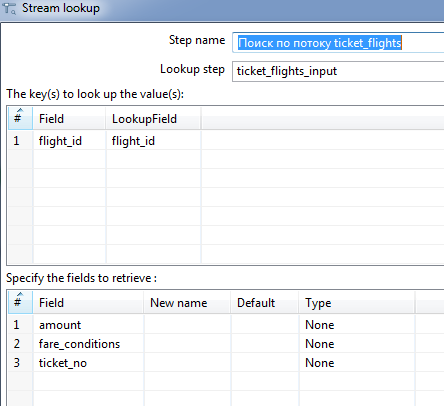
Fact\_Flights

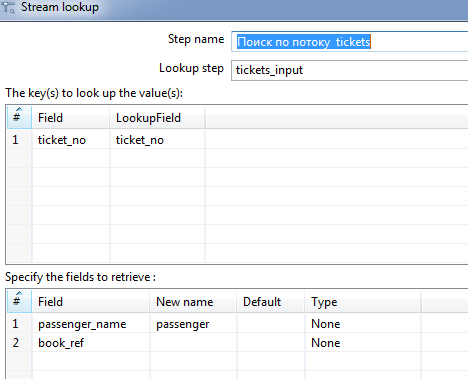


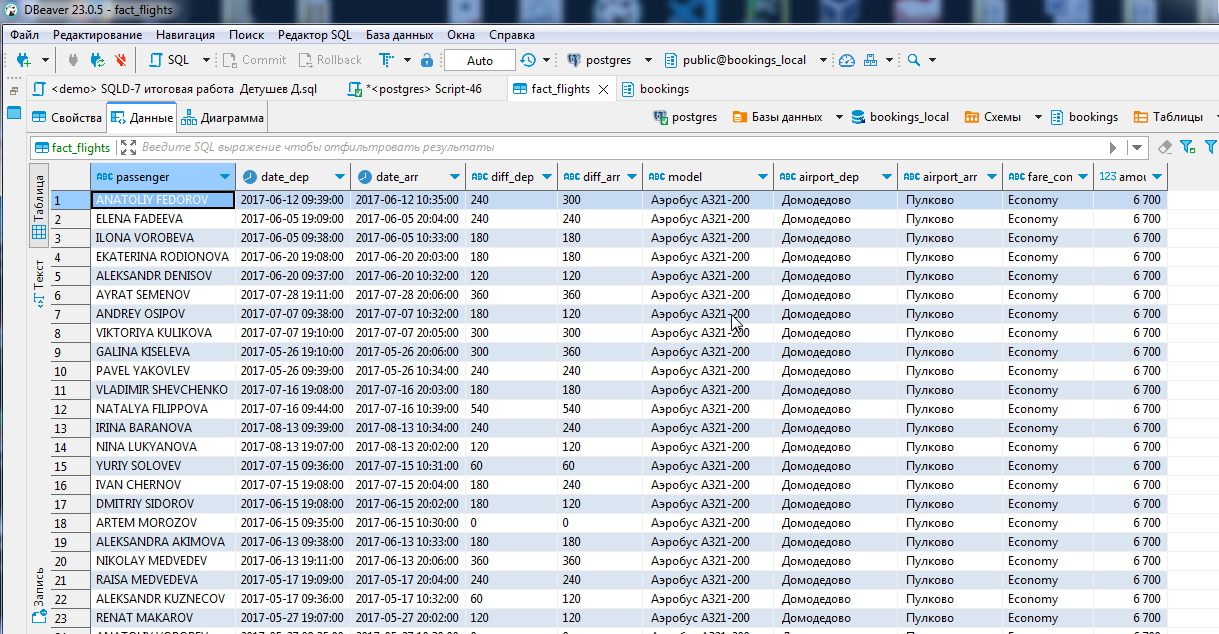












Данные в таблице Fact\_Flights