DWH-6

2021

**Проектная работа по модулю DWH**

некляева а.н

Оглавление

[Источники данных 2](#_Toc75981599)

[SQL-скрипты создания таблицы фактов и всех справочников 3](#_Toc75981600)

[Скриншоты созданных таблицы фактов и всех справочников в БД dwh\_result 6](#_Toc75981601)

[ER-диаграмма созданной базы данных хранилища 9](#_Toc75981602)

[Описание ETL-процессов и проверок качества данных 10](#_Toc75981603)

[Dim\_passengers: 10](#_Toc75981604)

[Проверки dim\_passengers: 10](#_Toc75981605)

[Dim\_tariff: 11](#_Toc75981606)

[Проверки Dim\_tariff: 11](#_Toc75981607)

[Dim\_airports 12](#_Toc75981608)

[Проверки Dim\_Airports 12](#_Toc75981609)

[Dim\_aircrafts 13](#_Toc75981610)

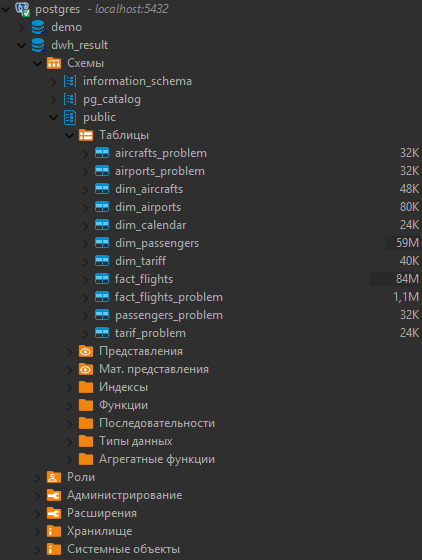
[Проверки Dim\_Aircrafts 13](#_Toc75981611)

[Fact\_Flights. 14](#_Toc75981612)

[Проверки Fact\_Flights 14](#_Toc75981613)

# Источники данных

1. После курса по SQL осталась развернутая база по авиаперевозкам на локальном компьютере - будем работать с ней.
2. Описание базы данных: https://edu.postgrespro.ru/bookings.pdf
3. Работаем в схеме bookings
4. Создана БД dwh\_result:



# SQL-скрипты создания таблицы фактов и всех справочников

Fact\_Flights – таблица фактов:

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

**PRIMARY** **KEY** (passenger\_id\_fk, actual\_departure, departure\_airport\_fk),

passenger\_id\_fk **int4** **NULL**,

actual\_departure **timestamp** **NULL**,

actual\_arrival **timestamp** **NULL**,

departure\_delay **float8** **NULL**,

arrival\_delay **float8** **NULL**,

aircraft\_code\_fk **int4** **NULL**,

departure\_airport\_fk **int4** **NULL**,

arrival\_airport\_fk **int4** **NULL**,

fare\_conditions\_fk **int4** **NULL**,

amount **numeric**(12,2) **NULL**,

actual\_departure\_date\_fk **int4** **NULL**,

actual\_arrival\_date\_fk **int4** **NULL**,

**CONSTRAINT** aircraft\_fk **FOREIGN** **KEY** (aircraft\_code\_fk) **REFERENCES** public.dim\_aircrafts(id),

**CONSTRAINT** arrival\_airport\_fk **FOREIGN** **KEY** (arrival\_airport\_fk) **REFERENCES** public.dim\_airports(id),

**CONSTRAINT** arrival\_date\_fk **FOREIGN** **KEY** (actual\_arrival\_date\_fk) **REFERENCES** public.dim\_calendar(id),

**CONSTRAINT** departure\_airport\_fk **FOREIGN** **KEY** (departure\_airport\_fk) **REFERENCES** public.dim\_airports(id),

**CONSTRAINT** departure\_date\_fk **FOREIGN** **KEY** (actual\_departure\_date\_fk) **REFERENCES** public.dim\_calendar(id),

**CONSTRAINT** fare\_conditions\_fk **FOREIGN** **KEY** (fare\_conditions\_fk) **REFERENCES** public.dim\_tariff(id),

**CONSTRAINT** passenger\_fk **FOREIGN** **KEY** (passenger\_id\_fk) **REFERENCES** public.dim\_passengers(id)

);

SQL скрипты справочников:

Dim\_Calendar - справочник дат:

*К сожалению, через запрос вида (SELECT MIN(scheduled\_departure) FROM demo.bookings.flights) сделать нельзя, т.к. итоговая работа выполняется в БД dwh\_result. Поэтому вводим результаты таких запросов руками:*

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

id serial **NOT** **NULL**,

calendar\_date **date** **NULL**,

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

);

**insert** **into** dim\_calendar(calendar\_date)

**select** **generate\_series**(**to\_date**('13.09.2016','dd.mm.yyyy'),**to\_date**('12.11.2016','dd.mm.yyyy'),'1 day');

Dim\_Passengers - справочник пассажиров:

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

id serial **NOT** **NULL**,

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

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

contact\_data **varchar**(400) **NULL**,

**CONSTRAINT** dim\_passengers\_pkey **PRIMARY** **KEY** (id),

**CONSTRAINT** unique\_passenger\_id **UNIQUE** (passenger\_id)

);

Dim\_Aircrafts - справочник самолетов:

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

id serial **NOT** **NULL**,

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

model **text** **NOT** **NULL**,

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

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

**CONSTRAINT** dim\_aircrafts\_pkey **PRIMARY** **KEY** (id),

**CONSTRAINT** unique\_aircraft\_code **UNIQUE** (aircraft\_code)

);

Dim\_Airports - справочник аэропортов:

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

id serial **NOT** **NULL**,

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

airport\_name **text** **NOT** **NULL**,

city **text** **NOT** **NULL**,

longitude **float8** **NOT** **NULL**,

latitude **float8** **NOT** **NULL**,

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

**CONSTRAINT** dim\_airports\_pkey **PRIMARY** **KEY** (id),

**CONSTRAINT** unique\_airport\_code **UNIQUE** (airport\_code)

);

Dim\_Tariff - справочник тарифов (Эконом/бизнес и тд):

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

id serial **NOT** **NULL**,

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

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

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

**CONSTRAINT** unique\_conditions **UNIQUE** (fare\_conditions)

);

SQL скрипты rejected-таблиц:

fact\_flights\_problem:

**CREATE** **TABLE** public.fact\_flights\_problem (

id serial **NOT** **NULL**,

passenger\_id **varchar**(64) **NULL**,

actual\_departure **timestamp** **NULL**,

actual\_arrival **timestamp** **NULL**,

actual\_departure\_date **timestamp** **NULL**,

actual\_arrival\_date **timestamp** **NULL**,

departure\_delay **float8** **NULL**,

arrival\_delay **float8** **NULL**,

aircraft\_code **varchar**(3) **NULL**,

departure\_airport **varchar**(64) **NULL**,

arrival\_airport **varchar**(64) **NULL**,

fare\_conditions **varchar**(64) **NULL**,

amount **numeric**(12,2) **NULL**,

passenger\_id\_fk **int4** **NULL**,

problem **varchar**(400) **NULL**,

**CONSTRAINT** fact\_flights\_problem\_pkey **PRIMARY** **KEY** (id)

);

passengers\_problem:

**CREATE** **TABLE** public.passengers\_problem (

id serial **NOT** **NULL**,

passenger\_id **text** **NULL**,

passenger\_name **text** **NULL**,

contact\_data **text** **NULL**,

book\_date **text** **NULL**,

problem **varchar**(400) **NULL**,

**CONSTRAINT** passengers\_problem\_pkey **PRIMARY** **KEY** (id)

);

aircrafts\_problem:

**CREATE** **TABLE** public.aircrafts\_problem (

id serial **NOT** **NULL**,

aircraft\_code **varchar**(15) **NULL**,

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

"range" **int4** **NULL**,

problem **varchar**(400) **NULL**,

**CONSTRAINT** aircrafts\_problem\_pkey **PRIMARY** **KEY** (id)

);

airports\_problem:

**CREATE** **TABLE** public.airports\_problem (

id serial **NOT** **NULL**,

airport\_code **text** **NULL**,

airport\_name **text** **NULL**,

city **text** **NULL**,

longitude **float8** **NULL**,

latitude **float8** **NULL**,

timezone **text** **NULL**,

problem **varchar**(400) **NULL**,

**CONSTRAINT** airports\_problem\_pkey **PRIMARY** **KEY** (id)

);

tarif\_problem:

**CREATE** **TABLE** public.tarif\_problem (

id serial **NOT** **NULL**,

fare\_conditions **varchar**(64) **NULL**,

problem **varchar**(400) **NULL**,

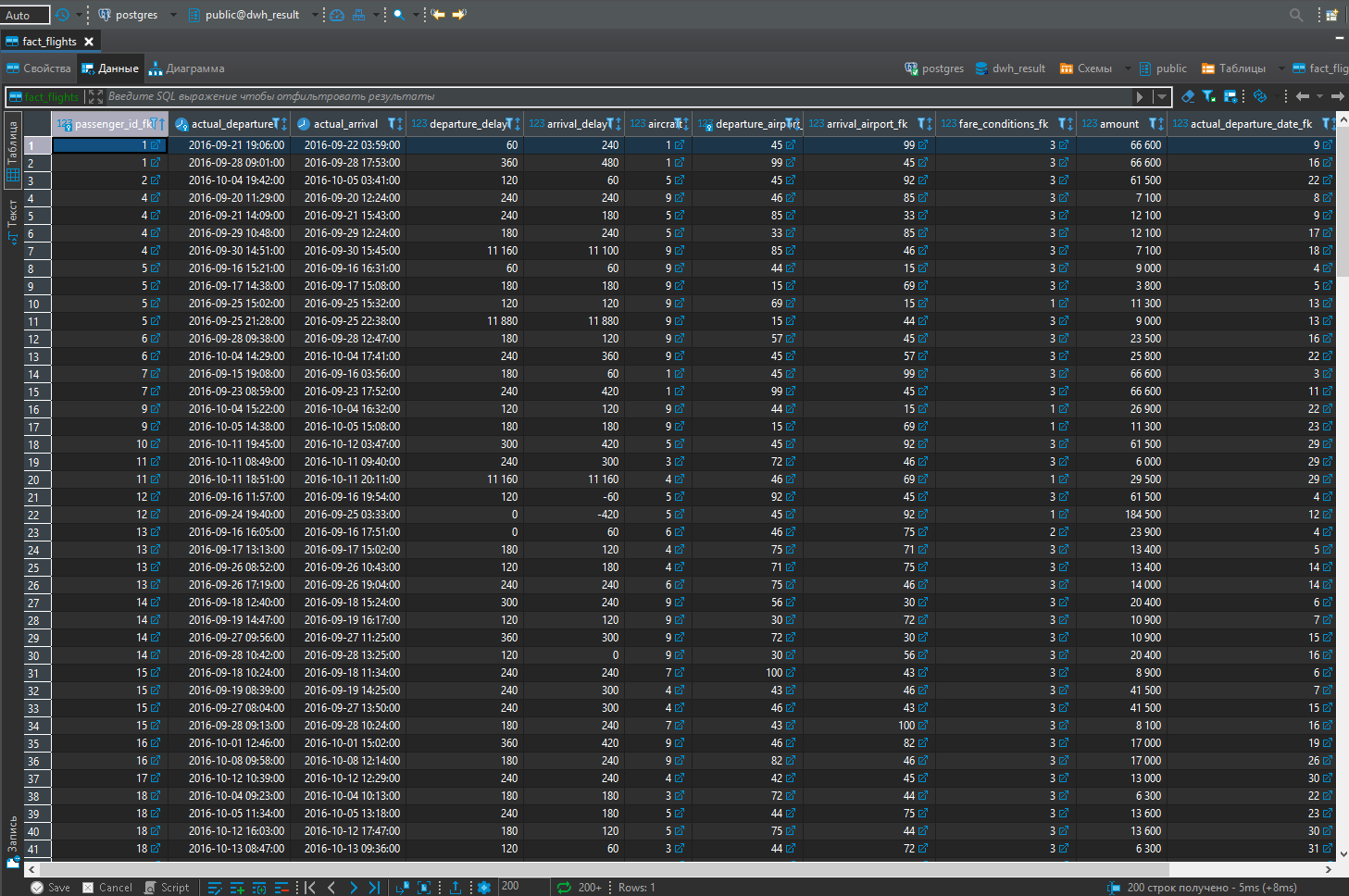
**CONSTRAINT** tarif\_problem\_pkey **PRIMARY** **KEY** (id)

);

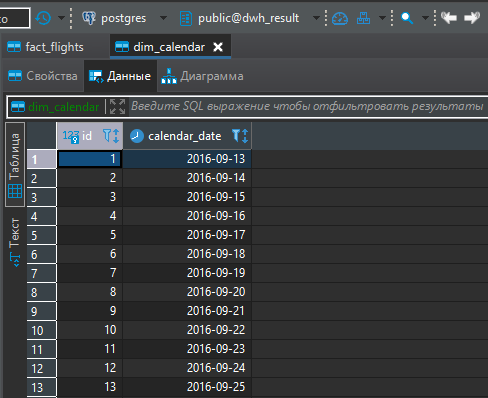
# Скриншоты созданных таблицы фактов и всех справочников в БД dwh\_result

Все созданные таблицы – содержат данные:

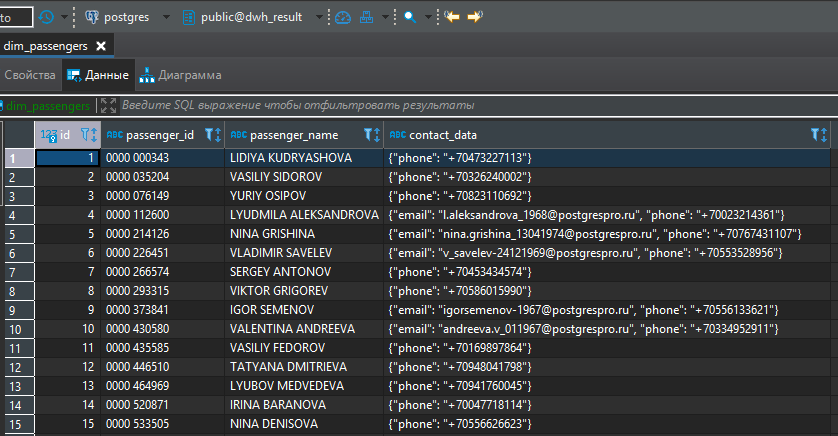
Fact\_Flights – таблица фактов: [ссылка на скриншот](http://joxi.ru/eAOwEwYSk1d4aA).

[](http://joxi.ru/eAOwEwYSk1d4aA)

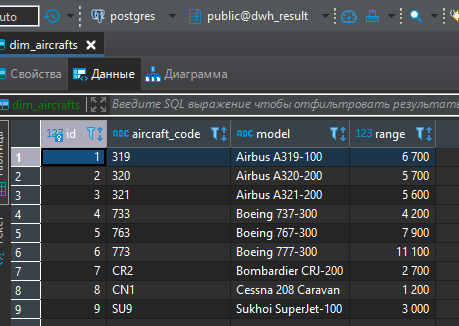
Dim\_Calendar - справочник дат: [ссылка на скриншот](http://joxi.ru/D2PwgwYSJDY10m)

[](http://joxi.ru/D2PwgwYSJDY10m)

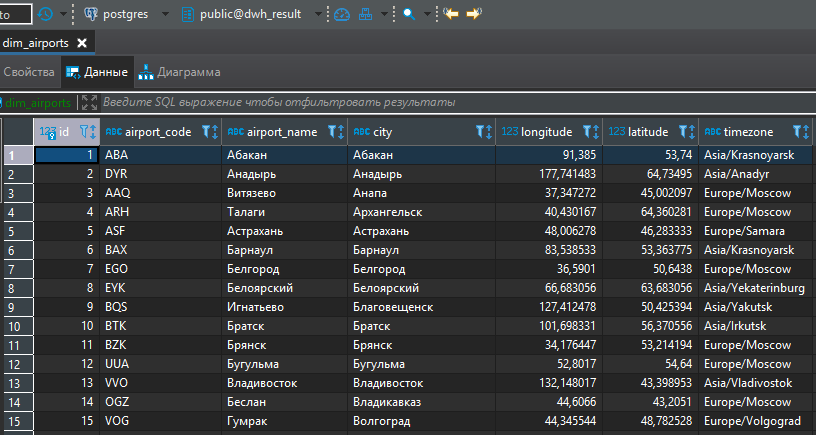
Dim\_Passengers - справочник пассажиров: [ссылка на скриншот](http://joxi.ru/4AkWLWoFkPYMbA)

[](http://joxi.ru/4AkWLWoFkPYMbA)

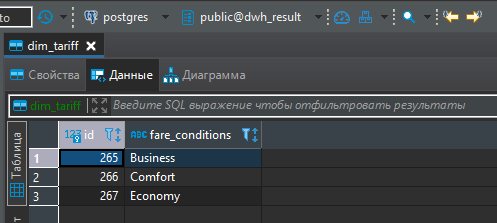
Dim\_Aircrafts - справочник самолетов: [ссылка на скриншот](http://joxi.ru/LmGwgweSlxvRGm)



Dim\_Airports - справочник аэропортов: [ссылка на скриншот](http://joxi.ru/E2pa3a1uGnwBQA)

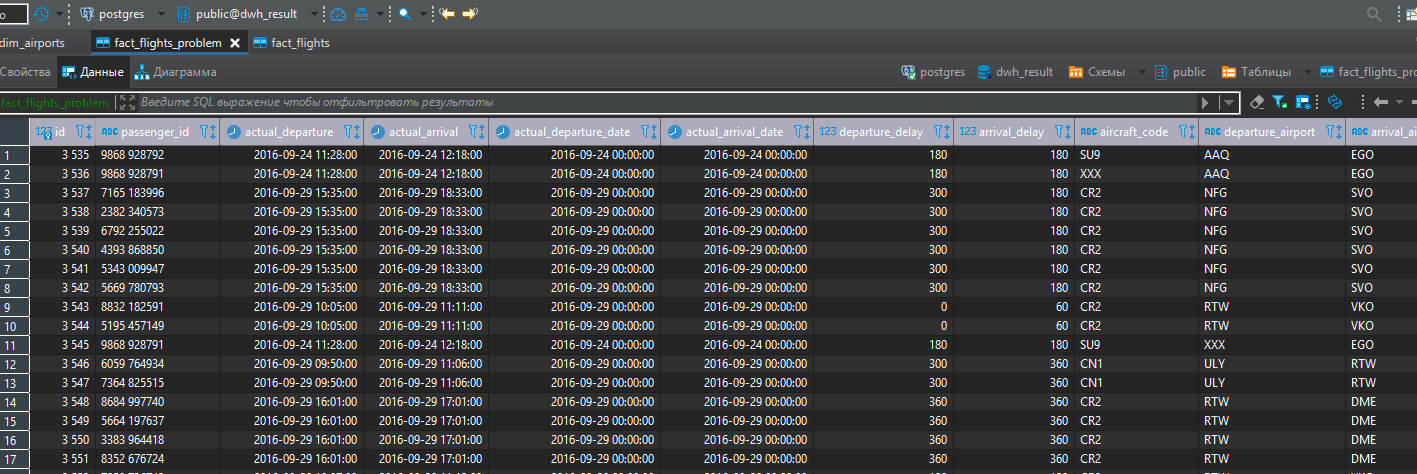
[](http://joxi.ru/E2pa3a1uGnwBQA)

Dim\_Tariff - справочник тарифов (Эконом/бизнес и тд): [ссылка на скриншот](http://joxi.ru/DrldKdocyWJzlr)

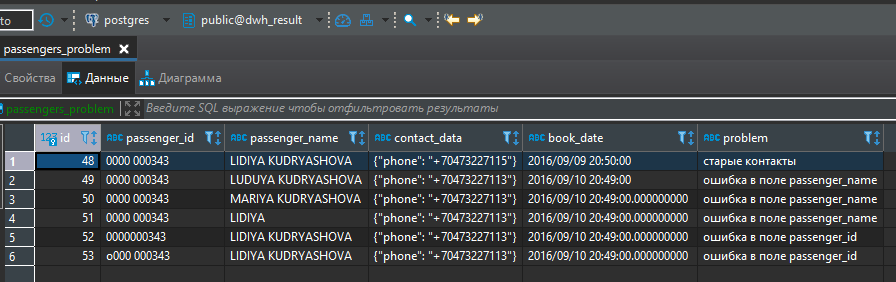


Rejected-таблиц:

fact\_flights\_problem: [ссылка на скриншот](http://joxi.ru/DmB9g9LH4Vvwxr)

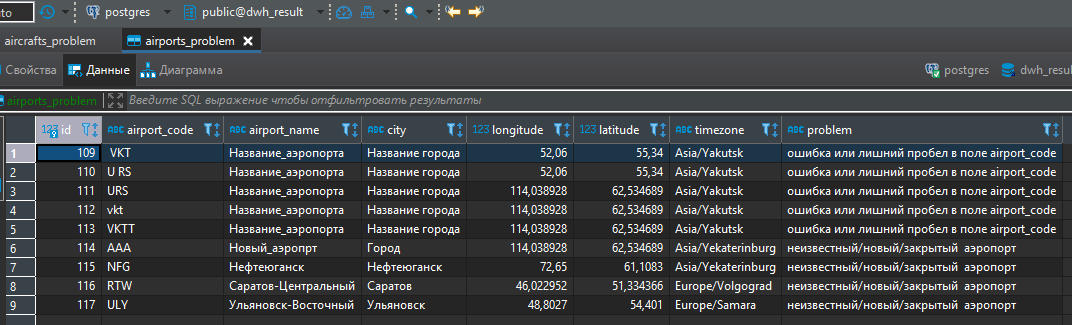
[](http://joxi.ru/DmB9g9LH4Vvwxr)

passengers\_problem: [ссылка на скриншот](http://joxi.ru/DmB9g9LH4VjaDr)



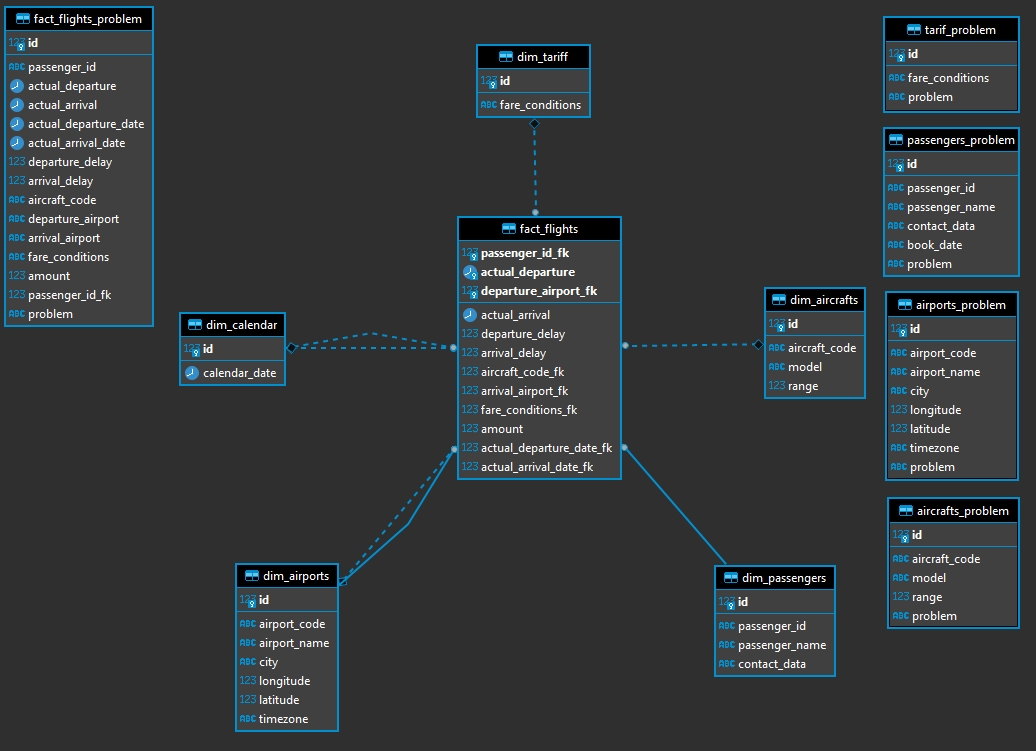
|  |  |
| --- | --- |
| aircrafts\_problem: [ссылка на скриншот](http://joxi.ru/a2XwlwZS4n7Y7m) | tarif\_problem: [ссылка на скриншот](http://joxi.ru/n2YwLwaSeLRJ0r) |
|  |  |

airports\_problem: [ссылка на скриншот](http://joxi.ru/KAgw5woSKzVLGA)



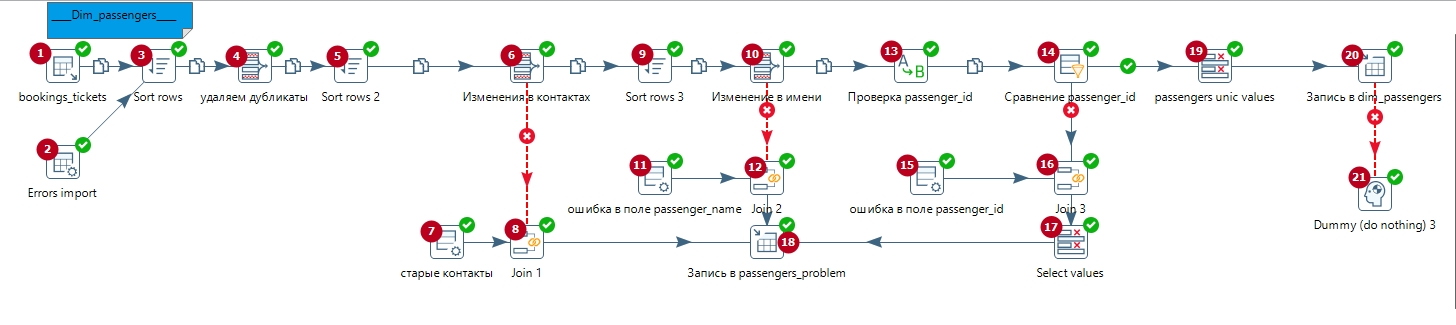
# ER-диаграмма созданной базы данных хранилища

[Ссылка на скриншот](http://joxi.ru/xAe4y4DTXz75N2)



# Описание ETL-процессов и проверок качества данных

## Dim\_passengers:

[](http://joxi.ru/8AnKaKoIyZXoRr)

* 1 -2-3-4-5 – импортируем данные, подмешиваем ошибки, сортируем, удаляем дубликаты
* 6-7-8 – проверка данных на наличие изменений в контактных данных
* 9-10-11-12 – проверка данных в поле passenger\_name
* 13-14-15-16 – проверка данных в поле passenger\_id
* 17-19 – поля в нужном порядке
* 18 – запись в таблицу ошибок passengers\_problem
* 20 – запись в таблицу dim\_passengers
* 21 – чтобы вся трансформация не останавливалась добавлен этот шаг

### Проверки dim\_passengers:

---Пассажиры---

1)Ошибка в букве в имени / Имя не полное (например имя без фамилии или когда имя заполнено не верно).

2)Изменение контактных данных.

4)passenger\_id состоит из набора символов, соответствующего паспорту.

**Оригинальная строка, которую будем делать ошибочной:**

**passenger\_id** | **passenger\_name**  | **contact\_data** | **book\_date**

0000 000343 | LIDIYA KUDRYASHOVA | {"phone": "+70473227113"} | 2016/09/10 20:49:00.000000000

Созданные для проверки ошибки:

1) 0000 000343 LIDIYA KUDRYASHOVA {"phone": "+70473227115"} 2016/09/09 20:49:00.000000000 - изменяем номер

телефона

2) 0000 000343 LUDUYA KUDRYASHOVA {"phone": "+70473227113"} 2016/09/10 20:49:00.000000000 - добавляем ошибки в имени

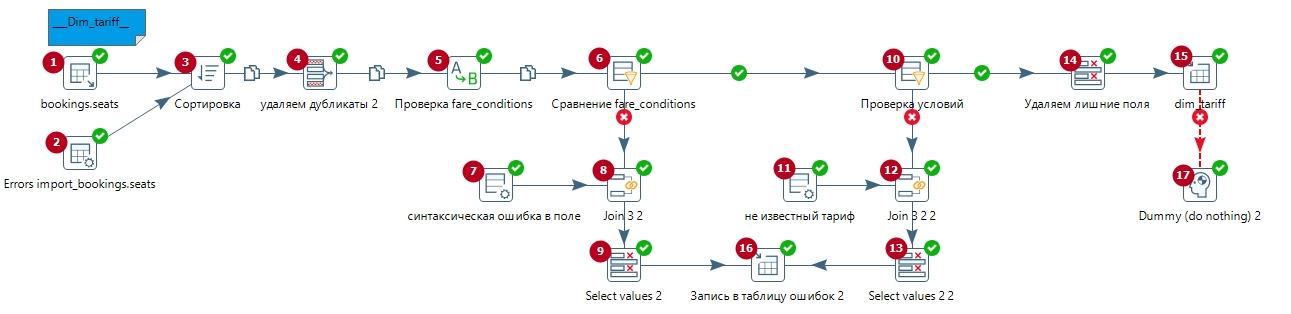
2.1)0000 000343 MARIYA KUDRYASHOVA {"phone": "+70473227113"} 2016/09/10 20:49:00.000000000 - изменяем имя на другое

2.2)0000 000343 LIDIYA {"phone": "+70473227113"} 2016/09/10 20:49:00.000000000 - имя без фамилии

3) 0000000343 LIDIYA KUDRYASHOVA {"phone": "+70473227113"} 2016/09/10 20:49:00.000000000 - в passenger\_id убираем пробел

3.1)o000000343 LIDIYA KUDRYASHOVA {"phone": "+70473227113"} 2016/09/10 20:49:00.000000000 - в passenger\_id заменяем первый символ 0 на букву "о"

## Dim\_tariff:

[](http://joxi.ru/D2PwgwYSJDlYPm)

* 1-2-3-4 – импортируем данные, подмешиваем ошибки, сортируем, удаляем дубликаты
* 5-6-7-8 – проверка данных в поле fare\_conditions на корректные символы
* 10-11-12 – проверка данных на наличие не известного тарифа
* 9-13-14 – поля в нужном порядке
* 16 - запись в таблицу ошибок tarif\_problem
* 15 – запись в таблицу dim\_tariff
* 17 – чтобы вся трансформация не останавливалась добавлен этот шаг

### Проверки Dim\_tariff:

-------Тарифы------

1. Тариф начинается с заглавной буквы, все остальные строчные

2.Тариф содержит только латинские символы и не содержит цифры

3.Тарифа всего три (Business,Economy,Comfort) и поле с тарифом не пустое

Сделанные для проверки ошибки:

Comfort123 -- наличие цифр

Lite ---- используется в качестве не известного тарифа помимо (Business,Economy,Comfort)

Comфort Lite -- наличие ошибок в слове

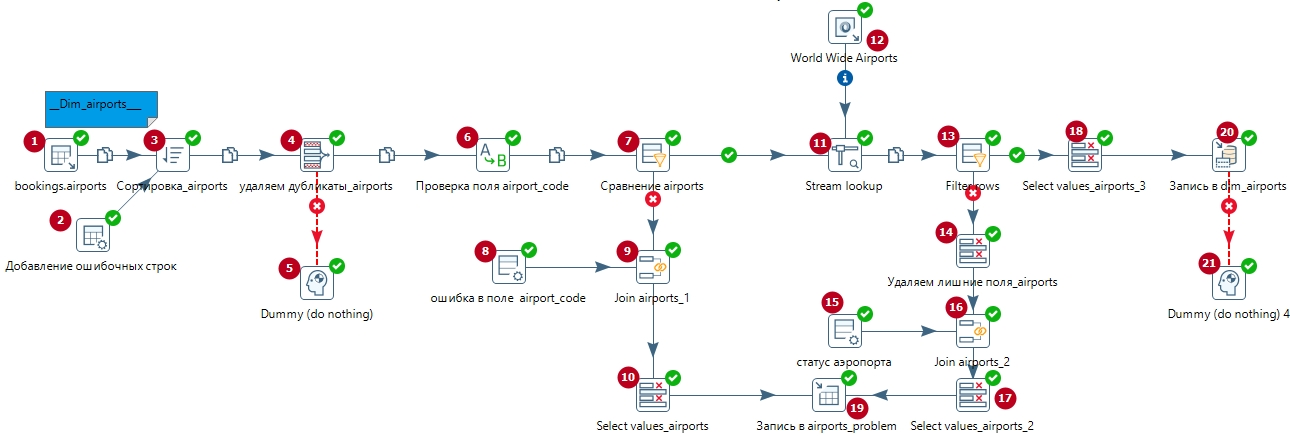
Com fort -- пробел в слове

BUSINES -- пробел в начале + все заглавные символы

BUSINES -- просто все заглавные буквы

Ecоnomy -- вторая "о" вставлена из русской раскладки

## Dim\_airports

[](http://joxi.ru/L21k5kzc07nxgr)

* 1-2-3-4 – импортируем данные, подмешиваем ошибки, сортируем, удаляем дубликаты
* 5 – здесь можно увидеть не уникальные строки
* 6-7-8-9 – проверка данных в поле airport\_code на корректные символы (ошибка или лишний пробел)
* 11-12 – здесь мы добавляем json-базу с данными по аэропортам (аэропорт закрытый или открытый)

[Ссылка на сайт](https://support.travelpayouts.com/hc/ru/articles/203956063-%D0%91%D0%B0%D0%B7%D1%8B-IATA) – проверка доступности аэропорта (отсюда взяты данные для блока 12)

* 13-14-15-16 – проверка данных на статус аэропорта (ищем новый/закрытый/ не известный базе аэропорт)
* 10-17-18 – поля в нужном порядке
* 19 - запись в таблицу ошибок airports\_problem
* 20 – запись в таблицу dim\_airports
* 21 – чтобы вся трансформация не останавливалась добавлен этот шаг

### Проверки Dim\_Airports

1) Код не пустой

2)Состоит из латинских символов

3)Символов должно быть три, согласно международным стандартам

4)Проверка статуса аэропорта (открытый/закрытый/новый)

Созданные для проверок ошибки:

1) vkt - строчные символы в airport\_code

2)VKTT - символов в коде больше трех

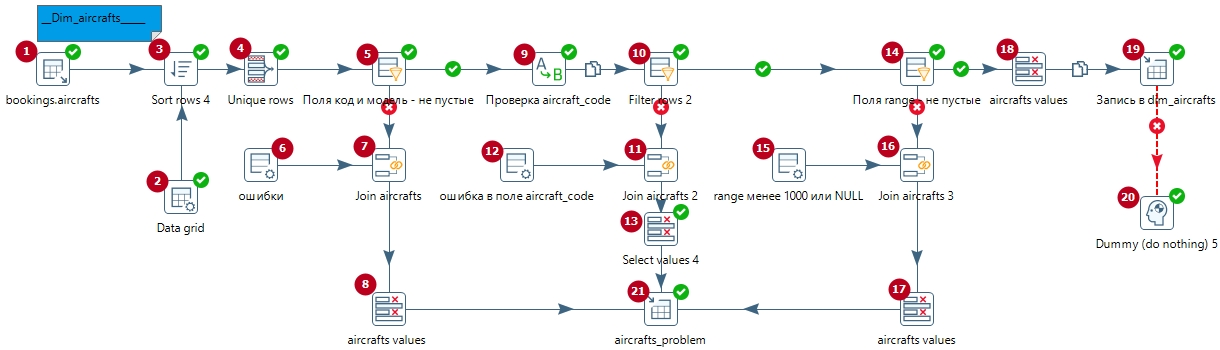
3)URS\_ - существующий код аэропорта, к которому добавлен в конце пробельный символ

4)\_VKT - пробельный символ добавлен в начале кода (на проверках пробельного кода не будет видно)

5)U RS - пробельный символ добавлен между буквами в коде

6)AAA - добавлена строка в качестве нового не известного аэропорта, этот аэропорт будем искать в базе международных аэропортов и смотреть его статус (открыт или закрыт)

## Dim\_aircrafts

[](http://joxi.ru/xAe4y4DTXz7yK2)

* 1-2-3-4 – импортируем данные, подмешиваем ошибки, сортируем, удаляем дубликаты
* *5-6-7 –*  проверка данных aircraft\_code и model не пустые
* 9-10-11-12 - проверка данных на корректные символы в поле
* 14-15-16 - проверка данных в поле range (range должен быть больше и равно 1000 км и не равен нулю)
* 8-13-17-18 – располагаем поля в нужном порядке перед записью в таблицу, удаляем лишние
* 19 – запись в таблицу dim*\_*aircrafts
* 20 – чтобы вся трансформация не останавливалась добавлен этот шаг
* 21 - запись в таблицу ошибок aircrafts\_problem

### Проверки Dim\_Aircrafts

-----Самолеты------

1)aircraft\_code - не пустой

2)aircraft\_code -содержит три символа, который может содержать латинские буквы и цифры

3)Поле "модель" - не пустое и содержит более одного символа латинские буквы и цифры

4)range должен быть больше и равно 1000 км и не равен нулю

Созданные ошибки для проверки:

aircraft\_code | model | range

пустое поле | Boeing | 1000 -- содержит пустое поле

AS 5 |Boeing 767-300 | 1000 -- содержит пробел в коде

АФК3 |Boeing 767-300 | 1000 -- содержит не латинские символы

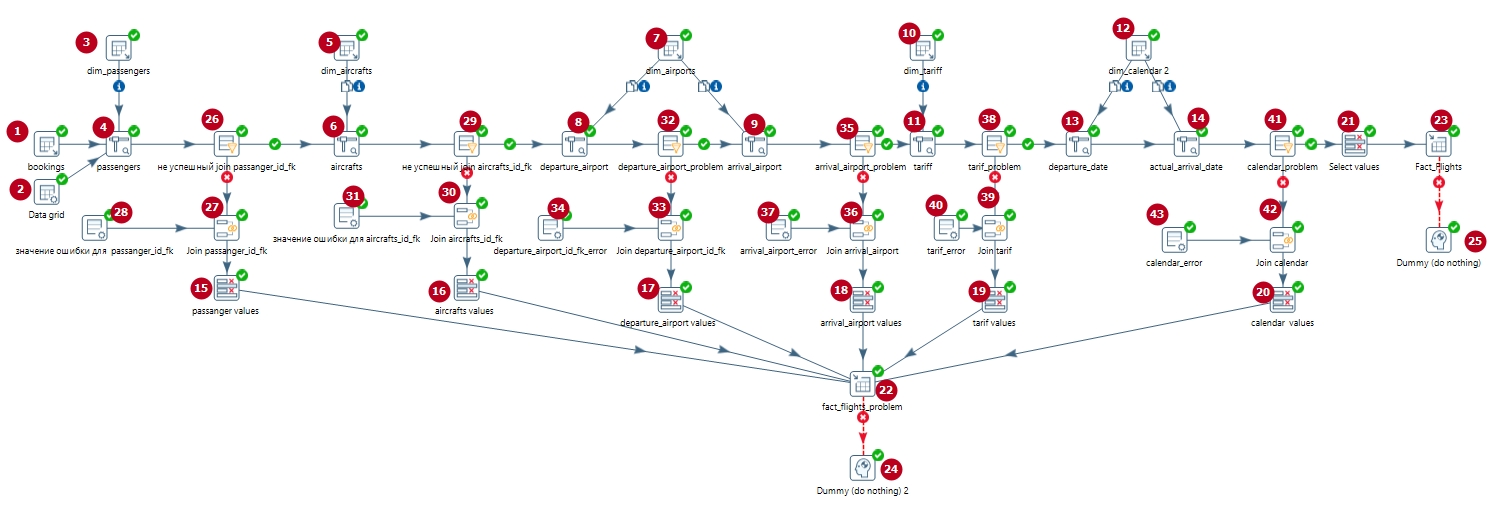
СR2 |Boeing 767-300 | 1000 -- в коде бука "С" - не из латинского алфавита

763 |пустое поле | 1000 -- пустое поле в model

111 |Boeing 111-111 | -- пустое поле в range

999 |Boeing 111-111 | 999 -- значение в поле range меньше 1000

## Fact\_Flights.



* 1-2- – импортируем данные, подмешиваем ошибки
* 3-4 – присоединяем таблицу dim\_passengers
* 5-6 – присоединяем таблицу dim\_aircrafts
* 7-8-9 – присоединяем таблицу dim\_airports
* 10-11 – присоединяем таблицу dim\_tariff
* 12-13-14 – присоединяем таблицу dim\_calendar
* 15-16-17-18-19-20-21 – выставляем последовательность столбцов перед записью в таблицу, удаляем лишние
* 22- запись в таблицу ошибок fact\_flights\_problem
* 23 – запись в таблицу fact\_flights
* 24-25 - чтобы вся трансформация не останавливалась добавлен этот шаг
* 26-27-28 – проверка данных успешный join passanger\_id\_fk
* 29-30-31 - проверка данных успешный join aircrafts\_id\_fk
* 32-33-34 - проверка данных успешный join departure\_airport\_id\_fk
* 35-36-37 - проверка данных успешный join arrival\_airport\_id\_fk
* 38-39-40 - проверка данных успешный join tarif\_id\_fk
* 41-42-43 - проверка данных дата вылета не может быть позже даты прилета

### Проверки Fact\_Flights

Так как качество текстовых данных мы проверяем на этапе заполнения справочников, на этапе ввода фактов достаточно проверить что данные корректно соединились:

1) успешный join passanger\_id\_fk

2) успешный join aircrafts\_id\_fk

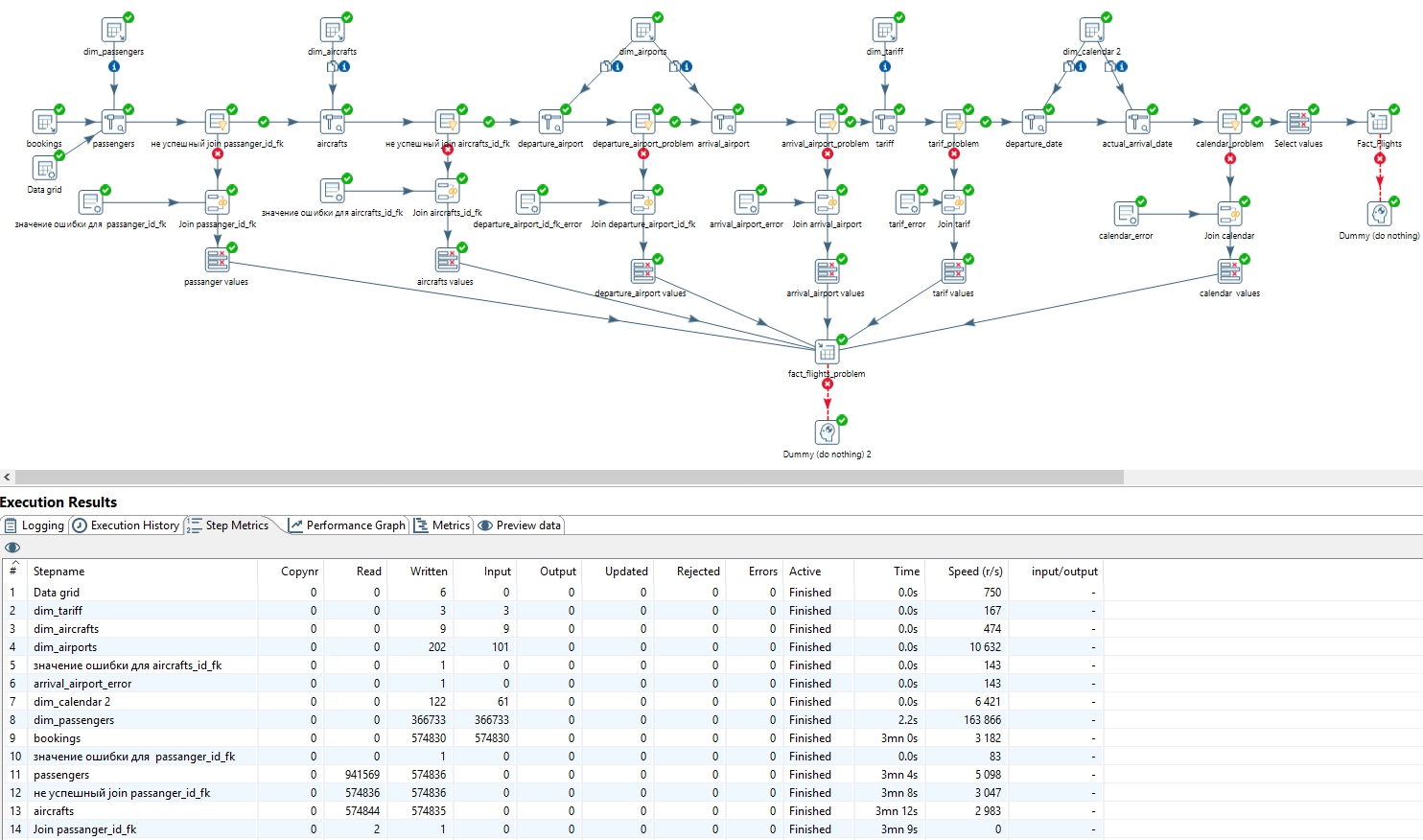
3) успешный join departure\_airport\_id\_fk

4) успешный join arrival\_airport\_id\_fk

5) успешный join tarif\_id\_fk

6)дата вылета не может быть позже даты прилета

Весь процесс. Step Metrics

[](http://joxi.ru/YmEwgwaSM4ewjA)