

ИТОГОВАЯ РАБОТА

**Проектная работа по модулю
«SQL и получение данных»**

Выполнил:
Сава Данил Андреевич

Содержание:

1. Общие данные.....	3
2. ER-диаграмма.....	4
3. Краткое описание базы данных.....	5
4. Развёрнутый анализ базы данных.....	6
4.1 Общие данные о базе данных <i>bookings</i>	6
4.2 Таблица <i>aircrafts_data</i>	7
4.3 Таблица <i>airports_data</i>	7
4.4 Таблица <i>boarding_passes</i>	8
4.5 Таблица <i>bookings</i>	9
4.6 Таблица <i>flights</i>	9
4.7 Таблица <i>seats</i>	11
4.8 Таблица <i>ticket_flights</i>	11
4.9 Таблица <i>tickets</i>	12
4.10 Представление <i>flights_v</i>	12
4.11 Представление <i>routes</i>	13
4.12 Представление <i>aircrafts</i>	14
4.13 Представление <i>airports</i>	14
4.14 Бизнес задачи, которые можно решить, используя БД.....	14
5. Решение заданий из практической части	15
5.1 Задание I.....	15
5.2 Задание II.....	16
5.3 Задание III	17
5.4 Задание IV.....	18
5.5 Задание V	19
5.6 Задание VI.....	21
5.7 Задание VII.....	22
5.8 Задание VIII.....	23
5.9 Задание IX.....	24

1. Общие данные

В работе использовался облачный тип подключения.

Со следующими настройками соединения:

Хост

Порт

База данных

Пользователь

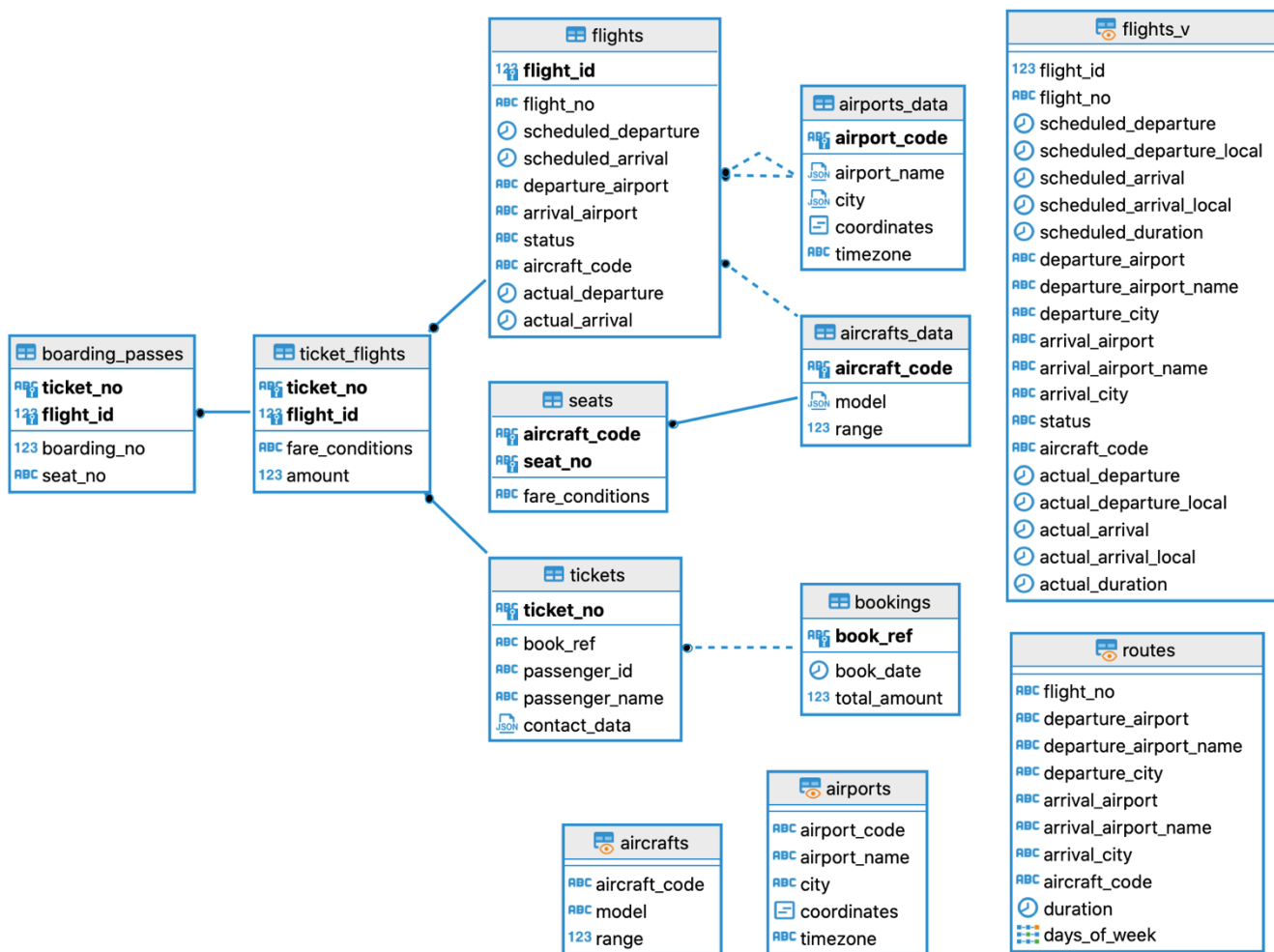
Пароль

Название базы

Схема

2. ER-диаграмма

Скриншот ER-диаграммы из DBeaver



3. Краткое описание базы данных

Краткое описание БД - из каких таблиц и представлений состоит.

Список отношений

Имя	Тип	Описание
aircrafts_data	таблица	Самолеты
airports_data	таблица	Аэропорты
boarding_passes	таблица	Посадочные талоны
bookings	таблица	Бронирования
flights	таблица	Рейсы
seats	таблица	Места
ticket_flights	таблица	Перелеты
tickets	таблица	Билеты
flights_v	представление	Рейсы
routes	представление	Маршруты
aircrafts	представление	Самолеты
airports	представление	Аэропорты

4. Развёрнутый анализ базы данных

4.1 Общие данные о базе данных *bookings*

Все объекты демонстрационной базы данных находятся в схеме `bookings`. Это означает, что при обращении к объектам необходимо либо явно указывать имя схемы (например: `bookings.flights`), либо предварительно изменить конфигурационный параметр `search_path` (например: `SET search_path = bookings, demo;`).

Однако для функции `bookings.now` в любом случае необходимо явно указывать схему, чтобы отличать ее от стандартной функции `now`.

Функция `now`

Демонстрационная база содержит временной «срез» данных — так, как будто в некоторый момент была сделана резервная копия реальной системы. Например, если некоторый рейс имеет статус `Departed`, это означает, что в момент резервного копирования самолет вылетел и находился в воздухе.

Позиция «среза» сохранена в функции `bookings.now()`. Ей можно пользоваться в запросах там, где в обычной жизни использовалась бы функция `now()`.

Кроме того, значение этой функции определяет версию демонстрационной базы данных.

Актуальная версия на текущий момент — от 13.10.2016.

Описание схемы

Основной сущностью является бронирование (`bookings`).

В одно бронирование можно включить несколько пассажиров, каждому из которых выписывается отдельный билет (`tickets`). Билет имеет уникальный номер и содержит информацию о пассажире. Как таковой пассажир не является отдельной сущностью. Как имя, так и номер документа пассажира могут меняться с течением времени, так что невозможно однозначно найти все билеты одного человека; для простоты можно считать, что все пассажиры уникальны.

Билет включает один или несколько перелетов (`ticket_flights`). Несколько перелетов могут включаться в билет в случаях, когда нет прямого рейса, соединяющего пункты отправления и назначения (полет с пересадками), либо когда билет взят «туда и обратно».

В схеме данных нет жесткого ограничения, но предполагается, что все билеты в одном бронировании имеют одинаковый набор перелетов.

Каждый рейс (`flights`) следует из одного аэропорта (`airports`) в другой. Рейсы с одним номером имеют одинаковые пункты вылета и назначения, но будут отличаться датой отправления.

При регистрации на рейс пассажиру выдается посадочный талон (boarding_passes), в котором указано место в самолете. Пассажир может зарегистрироваться только на тот рейс, который есть у него в билете. Комбинация рейса и места в самолете должна быть уникальной, чтобы не допустить выдачу двух посадочных талонов на одно место.

Количество мест (seats) в самолете и их распределение по классам обслуживания зависит от модели самолета (aircrafts), выполняющего рейс. Предполагается, что каждая модель самолета имеет только одну компоновку салона. Схема данных не контролирует, что места в посадочных талонах соответствуют имеющимся в самолете (такая проверка может быть сделана с использованием табличных триггеров или в приложении).

4.2 Таблица *aircrafts_data*

Каждая модель воздушного судна идентифицируется своим трехзначным кодом (aircraft_code). Указывается также название модели (model) и максимальная дальность полета в километрах (range).

Столбец	Тип	Модификаторы	Описание
aircraft_code	char(3)	NOT NULL	Код самолета, IATA
model	text	NOT NULL	Модель самолета
range	integer	NOT NULL	Максимальная дальность полета, км

Индексы:

PRIMARY KEY, btree (aircraft_code)

Ограничения-проверки:

CHECK (range > 0)

Ссылки извне:

TABLE "flights" FOREIGN KEY (aircraft_code)

REFERENCES aircrafts(aircraft_code)

TABLE "seats" FOREIGN KEY (aircraft_code)

REFERENCES aircrafts(aircraft_code) ON DELETE CASCADE

4.3 Таблица *airports_data*

Аэропорт идентифицируется трехбуквенным кодом (airport_code) и имеет свое имя (airport_name). Для города не предусмотрено отдельной сущности, но название (city) указывается и может служить для того, чтобы определить аэропорты одного города. Также указывается широта (longitude), долгота (latitude) и часовой пояс (timezone).

Столбец	Тип	Модификаторы	Описание
airport_code	char(3)	NOT NULL	Код аэропорта
airport_name	text	NOT NULL	Название аэропорта
city	text	NOT NULL	Город
coordinates	point	NOT NULL	Координаты аэропорта (Долгота, Широта)
timezone	text	NOT NULL	Временная зона аэропорта

Индексы:

PRIMARY KEY, btree (airport_code)

Ссылки извне:

TABLE "flights" FOREIGN KEY (arrival_airport)

REFERENCES airports(airport_code)

TABLE "flights" FOREIGN KEY (departure_airport)

REFERENCES airports(airport_code)

4.4 Таблица *boarding_passes*

При регистрации на рейс, которая возможна за сутки до плановой даты отправления, пассажиру выдается посадочный талон. Он идентифицируется также, как и перелет — номером билета и номером рейса.

Посадочным талонам присваиваются последовательные номера (boarding_no) в порядке регистрации пассажиров на рейс (этот номер будет уникальным только в пределах данного рейса). В посадочном талоне указывается номер места (seat_no).

Столбец	Тип	Модификаторы	Описание
ticket_no	char(13)	NOT NULL	Номер билета
flight_id	integer	NOT NULL	Идентификатор рейса
boarding_no	integer	NOT NULL	Номер посадочного талона
seat_no	varchar(4)	NOT NULL	Номер места

Индексы:

PRIMARY KEY, btree (ticket_no, flight_id)

UNIQUE CONSTRAINT, btree (flight_id, boarding_no)

UNIQUE CONSTRAINT, btree (flight_id, seat_no)

Ограничения внешнего ключа:

FOREIGN KEY (ticket_no, flight_id)

REFERENCES ticket_flights(ticket_no, flight_id)

4.5 Таблица *bookings*

Пассажир заранее (book_date, максимум за месяц до рейса) бронирует билет себе и, возможно, нескольким другим пассажирам. Бронирование идентифицируется номером (book_ref, шестизначная комбинация букв и цифр).

Поле total_amount хранит общую стоимость включенных в бронирование перелетов всех пассажиров.

Столбец	Тип	Модификаторы	Описание
book_ref	char(6)	NOT NULL	Номер бронирования
book_date	timestamptz	NOT NULL	Дата бронирования
total_amount	numeric(10,2)	NOT NULL	Полная сумма бронирования

Индексы:

PRIMARY KEY, btree (book_ref)

Ссылки извне:

TABLE "tickets" FOREIGN KEY (book_ref) REFERENCES bookings(book_ref)

4.6 Таблица *flights*

Естественный ключ таблицы рейсов состоит из двух полей — номера рейса (flight_no) и даты отправления (scheduled_departure). Чтобы сделать внешние ключи на эту таблицу компактнее, в качестве первичного используется суррогатный ключ (flight_id).

Рейс всегда соединяет две точки — аэропорты вылета (departure_airport) и прибытия (arrival_airport). Такое понятие, как «рейс с пересадками» отсутствует: если из одного аэропорта до другого нет прямого рейса, в билет просто включаются несколько необходимых рейсов.

У каждого рейса есть запланированные дата и время вылета (scheduled_departure) и прибытия (scheduled_arrival). Реальные время вылета (actual_departure) и прибытия (actual_arrival) могут отличаться: обычно не сильно, но иногда и на несколько часов, если рейс задержан.

Статус рейса (status) может принимать одно из следующих значений:

- **Scheduled** - Рейс доступен для бронирования. Это происходит за месяц до плановой даты вылета; до этого запись о рейсе не существует в базе данных.
- **On Time** - Рейс доступен для регистрации (за сутки до плановой даты вылета) и не задержан.
- **Delayed** - Рейс доступен для регистрации (за сутки до плановой даты вылета), но задержан.

- **Departed** - Самолет уже вылетел и находится в воздухе.
- **Arrived** - Самолет прибыл в пункт назначения.
- **Cancelled** - Рейс отменен.

Столбец	Тип	Модификаторы	Описание
flight_id	serial	NOT NULL	Идентификатор рейса
flight_no	char(6)	NOT NULL	Номер рейса
scheduled_departure	timestampz	NOT NULL	Время вылета по расписанию
scheduled_arrival	timestampz	NOT NULL	Время прилёта по расписанию
departure_airport	char(3)	NOT NULL	Аэропорт отправления
arrival_airport	char(3)	NOT NULL	Аэропорт прибытия
status	varchar(20)	NOT NULL	Статус рейса
aircraft_code	char(3)	NOT NULL	Код самолета, IATA
actual_departure	timestampz		Фактическое время вылета
actual_arrival	timestampz		Фактическое время прилёта

Индексы:

PRIMARY KEY, btree (flight_id)

UNIQUE CONSTRAINT, btree (flight_no, scheduled_departure)

Ограничения-проверки:

CHECK (scheduled_arrival > scheduled_departure)

CHECK ((actual_arrival IS NULL)

OR ((actual_departure IS NOT NULL AND actual_arrival IS NOT NULL)

AND (actual_arrival > actual_departure)))

CHECK (status IN ('On Time', 'Delayed', 'Departed', 'Arrived', 'Scheduled', 'Cancelled'))

Ограничения внешнего ключа:

FOREIGN KEY (aircraft_code)

REFERENCES aircrafts(aircraft_code)

FOREIGN KEY (arrival_airport)

REFERENCES airports(airport_code)

FOREIGN KEY (departure_airport)

REFERENCES airports(airport_code)

Ссылки извне:

TABLE "ticket_flights" FOREIGN KEY (flight_id)

REFERENCES flights(flight_id)

4.7 Таблица *seats*

Места определяют схему салона каждой модели. Каждое место определяется своим номером (*seat_no*) и имеет закрепленный за ним класс обслуживания (*fare_conditions*) — Economy, Comfort или Business.

Столбец	Тип	Модификаторы	Описание
aircraft_code	char(3)	NOT NULL	Код самолета, IATA
seat_no	varchar(4)	NOT NULL	Номер места
fare_conditions	varchar(10)	NOT NULL	Класс обслуживания

Индексы:

PRIMARY KEY, btree (*aircraft_code*, *seat_no*)

Ограничения-проверки:

CHECK (*fare_conditions* IN ('Economy', 'Comfort', 'Business'))

Ограничения внешнего ключа:

FOREIGN KEY (*aircraft_code*)

REFERENCES aircrafts(*aircraft_code*) ON DELETE CASCADE

4.8 Таблица *ticket_flights*

Перелет соединяет билет с рейсом и идентифицируется их номерами. Для каждого перелета указываются его стоимость (*amount*) и класс обслуживания (*fare_conditions*).

Столбец	Тип	Модификаторы	Описание
ticket_no	char(13)	NOT NULL	Номер билета
flight_id	integer	NOT NULL	Идентификатор рейса
fare_conditions	varchar(10)	NOT NULL	Класс обслуживания
amount	numeric(10,2)	NOT NULL	Стоимость перелета

Индексы:

PRIMARY KEY, btree (*ticket_no*, *flight_id*)

Ограничения-проверки:

CHECK (*amount* >= 0)

CHECK (*fare_conditions* IN ('Economy', 'Comfort', 'Business'))

Ограничения внешнего ключа:

FOREIGN KEY (*flight_id*) REFERENCES flights(*flight_id*)

FOREIGN KEY (*ticket_no*) REFERENCES tickets(*ticket_no*)

Ссылки извне:

TABLE "boarding_passes" FOREIGN KEY (ticket_no, flight_id)

REFERENCES ticket_flights(ticket_no, flight_id)

4.9 Таблица tickets

Билет имеет уникальный номер (ticket_no), состоящий из 13 цифр.

Билет содержит идентификатор пассажира (passenger_id) — номер документа, удостоверяющего личность, — его фамилию и имя (passenger_name) и контактную информацию (contact_data).

Ни идентификатор пассажира, ни имя не являются постоянными (можно поменять паспорт, можно сменить фамилию), поэтому однозначно найти все билеты одного и того же пассажира невозможно.

Столбец	Тип	Модификаторы	Описание
ticket_no	char(13)	NOT NULL	Номер билета
book_ref	char(6)	NOT NULL	Номер бронирования
passenger_id	varchar(20)	NOT NULL	Идентификатор пассажира
passenger_name	text	NOT NULL	Имя пассажира
contact_data	jsonb		Контактные данные пассажира

Индексы:

PRIMARY KEY, btree (ticket_no)

Ограничения внешнего ключа:

FOREIGN KEY (book_ref) REFERENCES bookings(book_ref)

Ссылки извне:

TABLE "ticket_flights" FOREIGN KEY (ticket_no) REFERENCES tickets(ticket_no)

4.10 Представление flights_v

Над таблицей flights создано представление flights_v, содержащее дополнительную информацию:

- расшифровку данных об аэропорте вылета
(departure_airport, departure_airport_name, departure_city),
- расшифровку данных об аэропорте прибытия
(arrival_airport, arrival_airport_name, arrival_city),
- местное время вылета
(scheduled_departure_local, actual_departure_local),

- местное время прибытия
(scheduled_arrival_local, actual_arrival_local),
- продолжительность полета
(scheduled_duration, actual_duration).

Столбец	Тип	Описание
flight_id	integer	Идентификатор рейса
flight_no	char(6)	Номер рейса
scheduled_departure	timestamp	Время вылета по расписанию
scheduled_departure_local	timestamp	Время вылета по расписанию, местное время в пункте отправления
scheduled_arrival	timestamp	Время прилёта по расписанию
scheduled_arrival_local	timestamp	Время прилёта по расписанию, местное время в пункте прибытия
scheduled_duration	interval	Планируемая продолжительность полета
departure_airport	char(3)	Код аэропорта отправления
departure_airport_name	text	Название аэропорта отправления
departure_city	text	Город отправления
arrival_airport	char(3)	Код аэропорта прибытия
arrival_airport_name	text	Название аэропорта прибытия
arrival_city	text	Город прибытия
status	varchar(20)	Статус рейса
aircraft_code	char(3)	Код самолета, IATA
actual_departure	timestamp	Фактическое время вылета
actual_departure_local	timestamp	Фактическое время вылета, местное время в пункте отправления
actual_arrival	timestamp	Фактическое время прилёта
actual_arrival_local	timestamp	Фактическое время прилёта, местное время в пункте прибытия
actual_duration	interval	Фактическая продолжительность полета

4.11 Представление routes

Таблица рейсов содержит избыточность: из нее можно было бы выделить информацию о маршруте (номер рейса, аэропорты отправления и назначения), которая не зависит от конкретных дат рейсов.

Именно такая информация и составляет материализованное представление routes.

Столбец	Тип	Описание
flight_no	char(6)	Номер рейса
departure_airport	char(3)	Код аэропорта отправления
departure_airport_name	text	Название аэропорта отправления
departure_city	text	Город отправления
arrival_airport	char(3)	Код аэропорта прибытия
arrival_airport_name	text	Название аэропорта прибытия
arrival_city	text	Город прибытия
aircraft_code	char(3)	Код самолета, IATA
duration	interval	Продолжительность полета
days_of_week	integer[]	Дни недели, когда выполняются рейсы

4.12 Представление *aircrafts*

Представляет тоже самое, что таблица `aircrafts_data`, только в представлении указано название модели самолета на русском языке.

4.13 Представление *airports*

Представляет тоже самое, что таблица `airports_data`, только в представлении указано названия аэропорта и города на русском языке.

4.14 Бизнес задачи, которые можно решить, используя БД

Используя данную базу данных, можно решить ряд бизнес-задач, например:

- Можно проанализировать рейсы с низкой заполняемостью и оптимизировать их частоту повторения;
- Произвести анализ самолетов на количество посадочных мест и их заполняемость, при низкой заполняемости можно использовать самолеты с меньшим количеством посадочных мест;
- Произвести анализ билетов на количество рейсов с пересадкой, если в одинаковых направлениях много пассажиров берут рейсы с пересадкой, то можно ввести дополнительные прямые рейсы, тем самым разгрузить уже имеющиеся направления;
- Можно проанализировать количество перелетов по дням недели и времени вылета \ прилета, на рейсах где наблюдается повышенный спрос можно увеличить стоимость перелета.

5. Решение заданий из практической части

5.1 Задание I

Вопрос:

В каких городах больше одного аэропорта?

Дополнительные условия:

Отсутствуют

SQL запрос:

```
select
    a.city                                as "Город",
    count (a.airport_code)               as "Кол-во аэропортов"
from
    airports a
group by
    a.city
having
    count (a.airport_code) > 1
order by
    a.city;
```

Решение:

Город	Кол-во аэропортов
Ульяновск	2
Москва	3

5.2 Задание II

Вопрос:

В каких аэропортах есть рейсы, выполняемые самолетом с максимальной дальностью перелета?

Дополнительные условия:

- Подзапрос

SQL запрос:

```
select
distinct fv.departure_airport as "Аэропорты"
from (
    select
        ad.aircraft_code,
        dense_rank () over(order by ad.range desc) as "dr"
    from
        aircrafts_data ad
    ) as ac
join flights_v fv using (aircraft_code)
where ac.dr = 1;
```

Решение:

Аэропорты

SVX

DME

PEE

VKO

OVB

SVO

AER

5.3 Задание III

Вопрос:

Вывести 10 рейсов с максимальным временем задержки вылета

Дополнительные условия:

- Оператор LIMIT

SQL запрос:

```
select
    f.flight_no                as "Номер рейса",
    f.scheduled_departure      as "Время вылета по расписанию",
    f.actual_departure         as "Фактическое время вылета",
    age(f.actual_departure, f.scheduled_departure) as "Задержка"
from
    flights f
where
    age(f.actual_departure, f.scheduled_departure) is not null
order by
    "Задержка" desc
limit
    10;
```

Решение:

Номер рейса	Время вылета по расписанию	Фактическое время вылета	Задержка
PG0132	2017-05-22 12:25:00.000 +0300	2017-05-22 17:06:00.000 +0300	04:41:00
PG0531	2017-05-17 09:15:00.000 +0300	2017-05-17 13:56:00.000 +0300	04:41:00
PG0589	2017-07-29 15:30:00.000 +0300	2017-07-29 20:07:00.000 +0300	04:37:00
PG0164	2017-07-29 15:25:00.000 +0300	2017-07-29 19:53:00.000 +0300	04:28:00
PG0544	2017-07-06 09:55:00.000 +0300	2017-07-06 14:23:00.000 +0300	04:28:00
PG0364	2017-07-19 11:45:00.000 +0300	2017-07-19 16:12:00.000 +0300	04:27:00
PG0151	2017-07-03 14:00:00.000 +0300	2017-07-03 18:27:00.000 +0300	04:27:00
PG0337	2017-06-01 12:45:00.000 +0300	2017-06-01 17:10:00.000 +0300	04:25:00
PG0354	2017-07-03 08:45:00.000 +0300	2017-07-03 13:10:00.000 +0300	04:25:00
PG0444	2017-05-25 12:30:00.000 +0300	2017-05-25 16:53:00.000 +0300	04:23:00

5.4 Задание IV

Вопрос:

Были ли брони, по которым не были получены посадочные талоны?

Дополнительные условия:

- Верный тип JOIN

SQL запрос:

```
select
    b.book_ref      as "Брони без посадочных",
    t.ticket_no     as "Билеты без посадочных"
from
    bookings b
left join tickets t      using (book_ref)
left join boarding_passes bp using (ticket_no)
where bp.boarding_no isnull;
```

Решение:

Брони без посадочных | Билеты без посадочных

-----+-----	
000068	0005432293273
000181	0005435545944
000181	0005435545945
0002D8	0005435767874
0002DB	0005433986734
0002DB	0005433986733
00034E	0005435653907
000374	0005433785757
00044D	0005432106361
0004E1	0005433869065
0004E1	0005433869064
00053F	0005432106386
0005F4	0005432531067
0005F4	0005432531066
0006C3	0005432329492
0006C3	0005432329491

5.5 Задание V

Вопрос:

Найдите количество свободных мест для каждого рейса, их % отношение к общему количеству мест в самолете.

Добавьте столбец с накопительным итогом - суммарное накопление количества вывезенных пассажиров из каждого аэропорта на каждый день. Т.е. в этом столбце должна отражаться накопительная сумма - сколько человек уже вылетело из данного аэропорта на этом или более ранних рейсах в течении дня.

Дополнительные условия:

- Оконная функция
- Подзапросы или/и сte

SQL запрос:

```
select
f.flight_id                                as "ID рейса",
f.flight_no                                as "№ рейса",
ss."Всего мест",
bb."Занято мест",
ss."Всего мест" - bb."Занято мест"          as "Свободно мест",
round (((ss."Всего мест" - bb."Занято мест")::numeric / ss."Всего мест"):: numeric * 100 ,
2)                                           as "% свободных мест",
f.departure_airport                        as "Аэропорт вылета",
f.actual_departure ::date                  as "Актуальный день вылета",
sum (bb."Занято мест") over (partition by f.actual_departure::date , f.departure_airport
order by f.actual_departure)               as "Кол-во вывезенных пас."
from
flights f
left join (
select
s.aircraft_code ,
count(s.seat_no)                                as "Всего мест"
from
seats s
group by s.aircraft_code
) as ss using (aircraft_code)
left join (
select
bp.flight_id ,
count (bp.seat_no)                                as "Занято мест"
from
boarding_passes bp
group by bp.flight_id
) as bb using (flight_id);
```

Решение:

ID рейса	№ рейса	Всего мест	Занято мест	Своб. мест	% свободных мест	Аэропорт вылета	Актуальный день вылета	Кол-во вывезен. пас.
41784	PG0480	97	4	93	95.88	AAQ	2017-05-17	4
41701	PG0252	130	49	81	62.31	AAQ	2017-05-17	53
36203	PG0520	116	6	110	94.83	ABA	2017-05-17	6
36394	PG0585	12	2	10	83.33	ABA	2017-05-17	8

5.6 Задание VI

Вопрос:

Найдите процентное соотношение перелетов по типам самолетов от общего количества.

Дополнительные условия:

- Подзапрос или окно
- Оператор ROUND

SQL запрос:

```
select
"Код самолета",
"Модель самолета",
"Кол-во полетов",
round(("Кол-во полетов"::numeric) / (sum("Кол-во полетов"::numeric) over ()) * 100 , 2)
as "% от всех полетов"
from
(select
f.aircraft_code as "Код самолета",
a.model as "Модель самолета",
count(f.flight_id) as "Кол-во полетов"
from
flights f
join aircrafts a using (aircraft_code)
group by f.aircraft_code , a.model
) as ff;
```

Решение:

Код самолета	Модель самолета	Кол-во полетов	% от всех полетов
773	Боинг 777-300	1210	1.84
733	Боинг 737-300	2522	3.84
SU9	Сухой Суперджет-100	16870	25.69
CN1	Сессна 208 Караван	18394	28.01
CR2	Бомбардье CRJ-200	17920	27.29
321	Аэробус A321-200	3872	5.90
319	Аэробус A319-100	2455	3.74
763	Боинг 767-300	2421	3.69

5.7 Задание VII

Вопрос:

Были ли города, в которые можно добраться бизнес - классом дешевле, чем эконом-классом в рамках перелета?

Дополнительные условия:

- CTE

SQL запрос:

```
with cte_001 AS (  
select  
"Аэропорты перелета",  
max ("11") as "MAX цена Economy"  
from  
(select  
f.departure_airport || ' ==> ' || f.arrival_airport as "Аэропорты перелета",  
tf.amount as "11"  
from  
flights f  
left join ticket_flights tf using (flight_id)  
where tf.fare_conditions = 'Economy') as "1"  
group by "Аэропорты перелета" ),  
cte_002 AS (  
select  
"Аэропорты перелета",  
min ("22") as "MIN цена Business"  
from  
(select  
f.departure_airport || ' ==> ' || f.arrival_airport as "Аэропорты перелета",  
tf.amount as "22"  
from  
flights f  
left join ticket_flights tf using (flight_id)  
where tf.fare_conditions = 'Business') as "2"  
group by "Аэропорты перелета")  
select  
*  
from  
cte_001  
join cte_002 using ("Аэропорты перелета")  
where "MAX цена Economy" > "MIN цена Business"
```

Решение:

Аэропорты перелета	MAX цена Economy	MIN цена Business
--------------------	------------------	-------------------

-----+-----+-----

Таких перелетов нет.

5.8 Задание VIII

Вопрос:

Между какими городами нет прямых рейсов?

Дополнительные условия:

- Декартово произведение в предложении FROM
- Самостоятельно созданные представления (если облачное подключение, то без представления)
- - Оператор EXCEPT

SQL запрос:

```
with cte_003 AS (  
select  
v.departure_city,  
v.arrival_city  
from flights_v v)  
select distinct  
a.city as "Город отправления",  
a1.city as "Город прибытия"  
from airports a  
cross join airports a1  
where a.city != a1.city  
except  
select  
departure_city,  
arrival_city  
from cte_003
```

Решение:

Город отправления	Город прибытия
Новосибирск	Йошкар-Ола
Норильск	Череповец
Нижневартовск	Архангельск
Новокузнецк	Омск
Новокузнецк	Ханты-Мансийск
Анапа	Ставрополь
Калининград	Иваново
Новосибирск	Чебоксары
Чита	Москва
Хабаровск	Архангельск

5.9 Задание IX

Вопрос:

Вычислите расстояние между аэропортами, связанными прямыми рейсами, сравните с допустимой максимальной дальностью перелетов в самолетах, обслуживающих эти рейсы

В облачной базе координаты находятся в столбце airports_data.coordinates - работаете, как с массивом.

В локальной базе координаты находятся в столбцах airports.longitude и airports.latitude.

Кратчайшее расстояние между двумя точками A и B на земной поверхности (если принять ее за сферу) определяется зависимостью:

$d = \arccos \{ \sin(\text{latitude_a}) \cdot \sin(\text{latitude_b}) + \cos(\text{latitude_a}) \cdot \cos(\text{latitude_b}) \cdot \cos(\text{longitude_a} - \text{longitude_b}) \}$, где latitude_a и latitude_b — широты, longitude_a, longitude_b — долготы данных пунктов, d — расстояние между пунктами измеряется в радианах длиной дуги большого круга земного шара.

Расстояние между пунктами, измеряемое в километрах, определяется по формуле:

$L = d \cdot R$, где $R = 6371$ км — средний радиус земного шара.

Дополнительные условия:

- - Оператор RADIANS или использование sind/cosd
- - CASE

SQL запрос:

```
select
ff.flight_id                as "ID рейса",
ff.departure_airport as "Аэропорт отправления",
ff.arrival_airport        as "Аэропорт прибытия",
ff."Расстояние" ,
ff.aircraft_code          as "Код самолета",
ad.range                  as "Дальность полета",
(case
when ad.range > ff."Расстояние"
then 'Все нормально, долетим :)'
else 'Я не полечу на этом рейсе :('
end)                      as "Результат"
from
(
select
f.flight_id ,
f.aircraft_code ,
```



```

f.departure_airport ,
ad1.coordinates [0]      as "Долгота 1",
ad1.coordinates [1]      as "Широта 1",
f.arrival_airport ,
ad2.coordinates [0]      as "Долгота 2",
ad2.coordinates [1]      as "Широта 2",
round((((acos((sind(ad1.coordinates [1])*sind(ad2.coordinates [1]) + cosd(ad1.coordinates
[1]) * cosd(ad2.coordinates [1]) * cosd((ad1.coordinates [0] - ad2.coordinates [0]))) *
6371)::numeric, 2)
                                as "Расстояние"

from
flights f,
airports_data ad1,
airports_data ad2
where f.departure_airport = ad1.airport_code
and   f.arrival_airport = ad2.airport_code
) as ff
left join aircrafts_data ad using (aircraft_code);

```

Решение:

ID	Аэропорт	Аэропорт	Расстояние	Код	Дальность	Результат
рейса	отправления	прибытия		сам.	полета	
-----+-----+-----+-----+-----+-----+-----						
182	DME	LED	666.88	321	5600	Все нормально, долетим :)
1996	DME	JOK	636.59	CN1	1200	Все нормально, долетим :)
5979	DME	BZK	343.55	SU9	3000	Все нормально, долетим :)
8136	VKO	RTW	731.79	CR2	2700	Все нормально, долетим :)
10455	SVO	OVV	2800.37	773	11100	Все нормально, долетим :)
11531	SVO	PES	683.33	CN1	1200	Все нормально, долетим :)
13774	SVO	ULV	712.31	SU9	3000	Все нормально, долетим :)
16088	LED	DME	666.88	321	5600	Все нормально, долетим :)
16181	LED	VKO	625.45	321	5600	Все нормально, долетим :)
19239	LED	CEE	440.73	CN1	1200	Все нормально, долетим :)