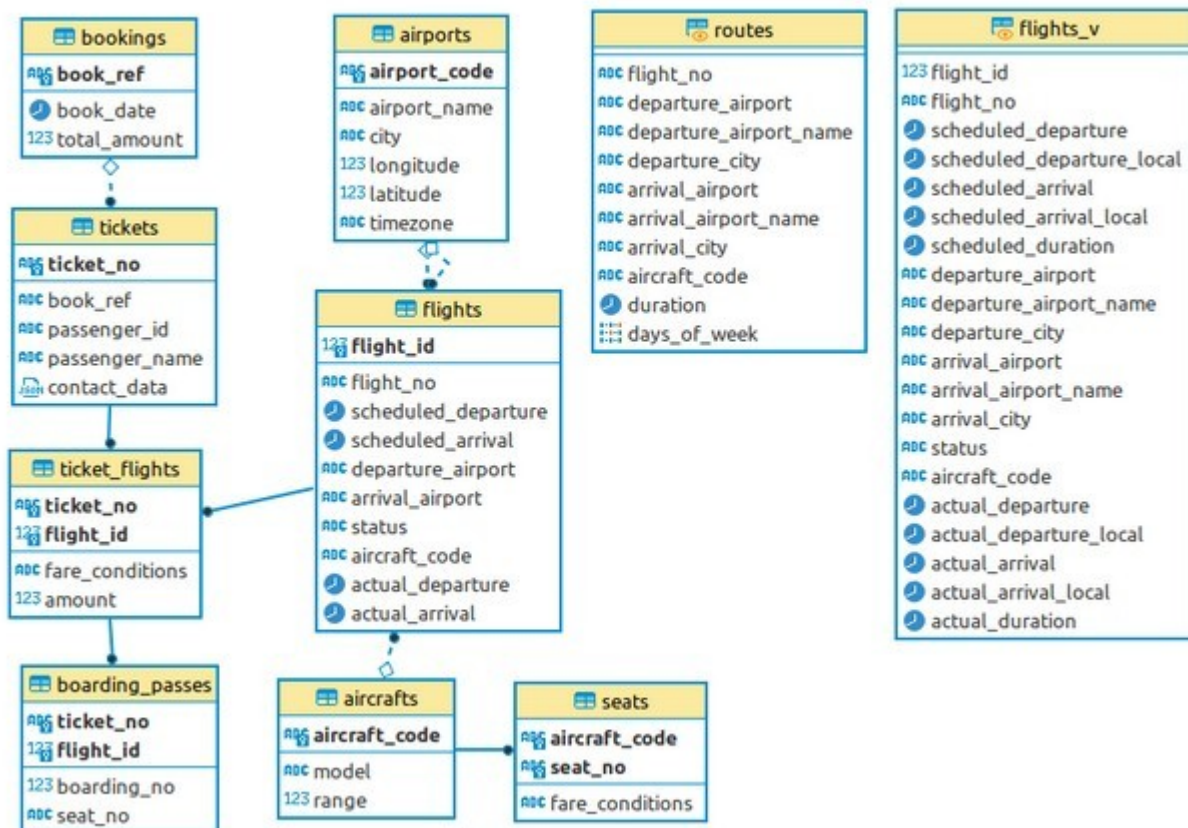


## 1.Скриншот ER-диаграммы из DBeaver



## 2.Краткое описание БД - из каких таблиц и представлений состоит?

База данных содержит 8 таблиц

	Название таблицы	Содержание
1	bookings	Бронирования
2	tickets	Билеты
3	ticket_flights	Перелеты
4	boarding_passes	Посадочные талоны
5	flights	Рейсы
6	airports	Аэропорты
7	aircrafts	Самолеты
8	seats	Места

и 2 представления

	Название представления	Содержание
1	flights_v	Рейсы
2	routes	Маршруты

## Таблицы, ограничения и ключевые поля

	Название схемы	Название таблицы	Ограничение	position	Ключевое поле
1	bookings	aircrafts	aircrafts_pkey	1	aircraft_code
2	bookings	airports	airports_pkey	1	airport_code
3	bookings	boarding_passes	boarding_passes_pkey	1	ticket_no
4	bookings	boarding_passes	boarding_passes_pkey	2	flight_id
5	bookings	bookings	bookings_pkey	1	book_ref
6	bookings	flights	flights_pkey	1	flight_id
7	bookings	seats	seats_pkey	1	aircraft_code
8	bookings	seats	seats_pkey	2	seat_no
9	bookings	ticket_flights	ticket_flights_pkey	1	ticket_no
10	bookings	ticket_flights	ticket_flights_pkey	2	flight_id
11	bookings	tickets	tickets_pkey	1	ticket_no

## Внешние связи

	Название схемы	Название таблицы	Внешний ключ	position	Ключевое поле
1	bookings	boarding_passes	boarding_passes_ticket_no_fkey	1	ticket_no
2	bookings	boarding_passes	boarding_passes_ticket_no_fkey	2	flight_id
3	bookings	flights	flights_aircraft_code_fkey	1	aircraft_code
4	bookings	flights	flights_arrival_airport_fkey	1	arrival_airport
5	bookings	flights	flights_departure_airport_fkey	1	departure_airport
6	bookings	seats	seats_aircraft_code_fkey	1	aircraft_code
7	bookings	ticket_flights	ticket_flights_flight_id_fkey	1	flight_id
8	bookings	ticket_flights	ticket_flights_ticket_no_fkey	1	ticket_no
9	bookings	tickets	tickets_book_ref_fkey	1	book_ref

## 3. Развернутый анализ БД - описание таблиц, логики, связей и бизнес области. Бизнес задачи, которые можно решить, используя БД.

### 3.1 Развернутый анализ

Основной сущностью является бронирование **bookings**. Таблица содержит номер и дату бронирования, полную стоимость билетов.

#### Таблица bookings

	название поля	тип данных	содержание	ключи
1	book_ref	char(6) Not Null	Номер бронирования	PK
2	book_date	timestampz Not Null	Дата бронирования	
3	total_amount	numeric(10,2) Not Null	Полная сумма бронирования	



В одно бронирование **bookings** можно включить несколько пассажиров, каждому из которых выписывается отдельный билет **tickets**. Билет имеет уникальный номер и содержит информацию о пассажире. Таблица связана с таблицей **bookings** через номер бронирования, также содержит имя пассажира и произвольные контактные данные в формате json. Идентификатор пассажира не связан ни с какой таблицей.

#### Таблица tickets

	название поля	тип данных	содержание	ключи	связь
1	ticket_no	char(13) Not Null	Номер билета	PK	
2	book_ref	char(6) Not Null	Номер бронирования	FK	bookings
3	passenger_id	varchar(20) Not Null	Идентификатор пассажира		
4	passenger_name	text Not Null	Имя пассажира		
5	contact_data	jsonb	Контактные данные пассажира		

Билет **tickets** 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». Таблица содержит информацию о классе обслуживания (Эконом, Комфорт, Бизнес) и стоимости перелета.

#### Таблица ticket\_flights

	название поля	тип данных	содержание	ключи	связь	ограничения
1	ticket_no	char(13) Not Null	Номер билета	PK FK	tickets	
2	flight_id	int4 Not Null	Идентификатор рейса	PK FK	flights	
3	fare_conditions	varchar(10) Not Null	Класс обслуживания			'Economy','Comfort','Business'
4	amount	numeric(10, 2) Not Null	Стоимость перелета			amount > 0

Каждый рейс **flights** следует из одного аэропорта **airports** в другой.

Таблица **flights** содержит информацию об аэропортах отправления и прибытия, времени по расписанию и фактического времени вылета и прилета, а также состояние рейса:

- 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 отменён.

Таблица flights

	название поля	тип данных	содержание	ключи	связь	ограничения
1	flight_id	serial4 Not Null	Идентификатор рейса	PK		
2	flight_no	char(6) Not Null	Номер рейса	UK		
3	scheduled_departure	timestampz Not Null	Время вылета по расписанию	UK		< scheduled_arrival
4	scheduled_arrival	timestampz Not Null	Время прилёта по расписанию			> scheduled_departure
5	departure_airport	char(3) Not Null	Аэропорт отправления	FK	airports	
6	arrival_airport	char(3) Not Null	Аэропорт прибытия	FK	airports	
7	status	varchar(20) Not Null	Статус рейса			'On Time', 'Delayed', 'Departed', 'Arrived', 'Scheduled', 'Cancelled'
8	aircraft_code	char(3) Not Null	Код самолета, IATA	FK	aircrafts	
9	actual_departure	timestampz	Фактическое время вылета			
10	actual_arrival	timestampz	Фактическое время прилёта			

Таблица **airports** содержит название аэропорта и города, координаты и часовой пояс.

Таблица airports

	название поля	тип данных	содержание	ключи	связь	ограничения
1	airport_code	char(3) Not Null	Код аэропорта	PK		
2	airport_name	text Not Null	Название аэропорта			
3	city	text Not Null	Город			
4	longitude	float8 Not Null	Координаты аэропорта: долгота			
5	latitude	float8 Not Null	Координаты аэропорта: широта			
6	timezone	text Not Null	Временная зона аэропорта			

При регистрации на рейс пассажиру выдаётся посадочный талон **boarding\_passes**, в котором указано место в самолете, соответствующее определенному классу обслуживания. Таблица содержит номер билета, идентификатор рейса, и уникальные в пределах рейса номер талона и номер посадочного места.

Таблица boarding\_passes

	название поля	тип данных	содержание	ключи	связь	ограничения
1	ticket_no	char(13) Not Null	Номер билета	PK FK	ticket_flights	
2	flight_id	int4 Not Null	Идентификатор рейса	PK UK_1 UK_2 FK	ticket_flights	
3	boarding_no	int4 Not Null	Номер посадочного талона	UK_1		
4	seat_no	varchar(4) Not Null	Номер места	UK_2		



Количество мест **seats** в самолете и их распределение по классам обслуживания зависит от модели самолета **aircrafts**, выполняющего рейс. Таблица **aircrafts** содержит код самолета, информацию о его модели и максимальной дальности полетов.

Таблица **aircrafts**

	название поля	тип данных	содержание	ключи	связь	ограничения
1	aircraft_code	char(3) Not Null	Код самолета, IATA			PK
2	model	text Not Null	Модель самолета			
3	range	int4 Not Null	Максимальная дальность полета, км			range > 0

Таблица **seats** содержит информацию о распределении мест и классов обслуживания для каждого самолета.

Таблица **seats**

	название поля	тип данных	содержание	ключи	связь	ограничения
1	aircraft_code	char(3) Not Null	Код самолета, IATA	PK FK	aircrafts	
2	seat_no	varchar(4) Not Null	Номер места	PK		
3	fare_conditions	varchar(10) Not Null	Класс обслуживания			

Часто используемая информация представлена в двух представлениях **routes** и **flights\_v**. Первое содержит маршруты, названия аэропортов и городов, время в полете и дни вылетов. Второе содержит информацию о рейсах, наименования аэропортов и городов, фактическое и время по расписанию вылетов, прилетов и нахождения в полете. Также содержит информацию о статусе рейса и код самолета.

Представление **routes**

	название поля	тип данных	содержание
1	flight_no	char(6)	Номер рейса
2	departure_airport	char(3)	Код аэропорта отправления
3	departure_airport_name	text	Название аэропорта отправления
4	departure_city	text	Город отправления
5	arrival_airport	char(3)	Код аэропорта прибытия
6	arrival_airport_name	text	Название аэропорта прибытия
7	arrival_city	text	Город прибытия
8	aircraft_code	char(3)	Код самолета, IATA
9	duration	interval	Фактическая продолжительность полета
10	days_of_week	{int4}	Дни недели, когда выполняются рейсы

## Представление flights\_v

	название поля	тип данных	содержание
1	flight_id	int4	Идентификатор рейса
2	flight_no	char(6)	Номер рейса
3	scheduled_departure	timestampz	Время вылета по расписанию
4	scheduled_departure_local	timestamp	Время вылета по расписанию, местное время в пункте отправления
5	scheduled_arrival	timestampz	Время прилёта по расписанию
6	scheduled_arrival_local	timestamp	Время прилёта по расписанию, местное время в пункте прибытия
7	scheduled_duration	interval	Планируемая продолжительность полета
8	departure_airport	char(3)	Код аэропорта отправления
9	departure_airport_name	text	Название аэропорта отправления
10	departure_city	text	Город отправления
11	arrival_airport	char(3)	Код аэропорта прибытия
12	arrival_airport_name	text	Название аэропорта прибытия
13	arrival_city	text	Город прибытия
14	status	varchar(20)	Статус рейса
15	aircraft_code	char(3)	Код самолета, IATA
16	actual_departure	timestampz	Фактическое время вылета
17	actual_departure_local	timestamp	Фактическое время вылета, местное время в пункте отправления
18	actual_arrival	timestampz	Фактическое время прилёта
19	actual_arrival_local	timestamp	Фактическое время прилёта, местное время в пункте прибытия
20	actual_duration	interval	Фактическая продолжительность полета

## 3.2 Бизнес задачи

Анализируя базу данных можно получить разнообразную информацию.

Например:

- определить наиболее востребованные направления, для увеличения количества рейсов
- оценить спрос по разным классам обслуживания пассажиров, для возможной замены борта с другого направления
- определить клиентов, постоянно пользующихся услугами аиакомпании, для предоставления бонусов
- определить часы налета по самолетам для планирования технического обслуживания наземными службами
- определить требуемое количество комплектов питания для пассажиров в зависимости от продолжительности полета и нагрузки рейса
- определить малонагруженные рейсы для планирования маркетинговых акции

## 4. Написать SQL запросы с описанием логики их выполнения и представить скрины их выполнения

### 4.1 В каких городах больше одного аэропорта?

- обратимся к таблице **airports** и выберем все города
- сгруппируем список по названию города
- выведем название города и количество элементов в группе, имеющей более 1 записи

```
query(''  
SET search_path TO bookings;  
  
SELECT city AS Город, COUNT(*) AS аэропортов  
FROM airports a  
GROUP BY city  
HAVING count(*) > 1;  
'')
```

	Город	аэропортов
0	Ульяновск	2
1	Москва	3



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

- соединим таблицы **airports flights aircrafts** через INNER JOIN
- сделаем подзапрос к таблице **aircrafts** и найдем значение максимальной дальности полета
- найдем аэропорты назначения и отправления, у которых есть рейсы, выполняемые самолетами с такой дальностью полета

```
query(''  
SET search_path TO bookings;  
  
SELECT DISTINCT airport_name AS аэропорт  
FROM airports a  
      INNER JOIN flights f ON a.airport_code = f.departure_airport  
                           OR a.airport_code = f.arrival_airport  
      INNER JOIN aircrafts a2 ON a2.aircraft_code = f.aircraft_code  
WHERE a2.range = (SELECT MAX(a3.range) FROM aircrafts a3 )  
ORDER BY airport_name;  
'' )
```

аэропорт	
0	Внуково
1	Домодедово
2	Кольцово
3	Пермь
4	Сочи
5	Толмачёво
6	Шереметьево



### 4.3 Вывести 10 рейсов с максимальным временем задержки вылета

- обратимся к представлению **flights\_v**
- выберем вылетевшие рейсы, задав условие ненулевого значения запланированного и фактического времени вылета
- создадим вычисляемое поле - разность между фактическим и запланированным временем вылета
- отсортируем запрос в порядке убывания вычисленного значения
- ограничим вывод 10-ю строками

```
query(''  
SET search_path TO bookings;  
  
SELECT f.flight_no AS Рейс,  
       f.departure_airport_name AS Вылет,  
       f.arrival_airport_name AS Прилет,  
       f.actual_departure - f.scheduled_departure AS Задержка_вылета  
FROM flights_v f  
WHERE f.actual_departure IS NOT NULL  
      AND f.scheduled_departure IS NOT NULL  
ORDER BY Задержка_вылета DESC  
LIMIT 10;  
''')
```

	Рейс	Вылет	Прилет	Задержка_вылета
0	PG0589	Пермь	Кольцово	0 days 04:37:00
1	PG0164	Домодедово	Новый Уренгой	0 days 04:28:00
2	PG0364	Краснодар	Баратаевка	0 days 04:27:00
3	PG0568	Советский	Уфа	0 days 04:20:00
4	PG0454	Домодедово	Курск-Восточный	0 days 04:18:00
5	PG0096	Петрозаводск	Домодедово	0 days 04:18:00
6	PG0166	Кольцово	Магнитогорск	0 days 04:16:00
7	PG0278	Толмачёво	Шереметьево	0 days 04:16:00
8	PG0564	Уфа	Ухта	0 days 04:14:00
9	PG0669	Внуково	Курган	0 days 04:08:00

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

Бронирование начинается за месяц до даты рейса, а посадочные талоны выдаются начиная за день до рейса. Поэтому следует разделить рейсы на группы:

- которые вылетели или отменены, регистрация по ним завершена
- на которые открыта регистрация, но рейсы задержаны
- на которые открыта регистрация и ещё не завершена

В общем случае логика запроса аналогична для всех групп:

- соединим таблицы **tickets** - **ticket\_flights** - **flights** и найдем все номера бронирований, соответствующие одной из вышеуказанных групп по статусу рейса
- соединим таблицы **tickets** - **boarding\_passes** и найдем все номера бронирований, по которым получены посадочные талоны
- из первого множества исключим номера второго множества
- подсчитаем количество оставшихся строк и сделаем вывод на основании результата

### 4.4.1 по вылетевшим, завершенным и отмененным рейсам

```
: query(''  
SET search_path TO bookings;  
  
SELECT  
    CASE  
        WHEN COUNT(*) > 0  
        THEN CONCAT('Не получено талонов: ', COUNT(*))  
        ELSE 'Неполученных талонов нет'  
    END Ответ  
  
FROM (  
    SELECT t.book_ref  
    FROM tickets t  
        INNER JOIN ticket_flights tf USING (ticket_no)  
        INNER JOIN flights f USING(flight_id)  
    WHERE status IN ('Departed', 'Arrived', 'Cancelled')  
    EXCEPT  
    SELECT t2.book_ref  
        FROM tickets t2  
        INNER JOIN boarding_passes bp USING (ticket_no)  
    GROUP BY t2.book_ref  
) t;  
''
```

:

Ответ

0 Неполученных талонов нет



#### 4.4.2 по задержанным рейсам

```
query(''  
SET search_path TO bookings;  
  
SELECT  
    CASE  
        WHEN COUNT(*) > 0  
        THEN CONCAT('Не получено талонов: ', COUNT(*))  
        ELSE 'Неполученных талонов нет'  
    END Ответ  
  
FROM (  
    SELECT t.book_ref  
    FROM tickets t  
        INNER JOIN ticket_flights tf USING (ticket_no)  
        INNER JOIN flights f USING(flight_id)  
    WHERE status IN ('Delayed')  
    EXCEPT  
    SELECT t2.book_ref  
        FROM tickets t2  
        INNER JOIN boarding_passes bp USING (ticket_no)  
    GROUP BY t2.book_ref  
) t;  
''')
```

Ответ

0 Не получено талонов: 392

#### 4.4.3 по открытым для регистрации и не задержанным рейсам

```
query(''  
SET search_path TO bookings;  
  
SELECT  
    CASE  
        WHEN COUNT(*) > 0  
        THEN CONCAT('Не получено талонов: ', COUNT(*))  
        ELSE 'Неполученных талонов нет'  
    END Ответ  
  
FROM (  
    SELECT t.book_ref  
    FROM tickets t  
        INNER JOIN ticket_flights tf USING (ticket_no)  
        INNER JOIN flights f USING(flight_id)  
    WHERE status IN ('On Time')  
    EXCEPT  
    SELECT t2.book_ref  
        FROM tickets t2  
        INNER JOIN boarding_passes bp USING (ticket_no)  
    GROUP BY t2.book_ref  
) t;  
''')
```

Ответ

0 Не получено талонов: 4613

## 4.5 Найдите свободные места 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 за день.

- сформируем для объединения две таблицы:
  - запрос из таблицы **boarding\_passes** - количество выданных посадочных талонов по каждому рейсу
  - запрос из таблицы **seats** - общее количество мест по типу самолета
- соединим с таблицами **flights - airports - aircrafts**
- добавим вычисляемое поле - процент свободных мест
- добавим оконную функцию для вычисления суммы выданных посадочных талонов с группировкой по аэропорту вылета и дате вылета, с сортировкой по времени вылета

```
query(''
SET search_path TO bookings;

SELECT f.flight_no AS рейс,
       a.airport_name AS аропорт,
       a2.model AS самолет,
       t1.col_pass AS пасс,
       t2.max_seats AS мест,
       (t2.max_seats - t1.col_pass) * 100 / t2.max_seats AS проц_своб,
       SUM(t1.col_pass)
         OVER(PARTITION BY f.departure_airport, f.actual_departure::date
              ORDER BY f.actual_departure) AS вылетело,
       f.actual_departure::date AS дата
FROM flights f
  INNER JOIN (
    SELECT bp.flight_id, MAX(bp.boarding_no) AS col_pass
    FROM boarding_passes bp
    GROUP BY bp.flight_id) t1 USING(flight_id)
  INNER JOIN (
    SELECT aircraft_code, COUNT(*) AS max_seats
    FROM seats
    GROUP BY aircraft_code
    ) t2 USING(aircraft_code)
  INNER JOIN airports a ON f.departure_airport = a.airport_code
  INNER JOIN aircrafts a2 USING(aircraft_code)
WHERE actual_departure IS NOT NULL ;
'')
```



	рейс	аэропорт	самолет	пасс	мест	проц_своб	вылетело	дата
0	PG0480	Витязево	Sukhoi SuperJet-100	3	97	96	3	2016-09-13
1	PG0252	Витязево	Boeing 737-300	51	130	60	54	2016-09-13
2	PG0480	Витязево	Sukhoi SuperJet-100	3	97	96	3	2016-09-14
3	PG0252	Витязево	Boeing 737-300	50	130	61	53	2016-09-14
4	PG0480	Витязево	Sukhoi SuperJet-100	5	97	94	5	2016-09-15
...	...	...	...	...	...	...	...	...
11473	PG0699	Якутск	Bombardier CRJ-200	35	50	30	48	2016-10-11
11474	PG0244	Якутск	Airbus A319-100	9	116	92	9	2016-10-12
11475	PG0699	Якутск	Bombardier CRJ-200	40	50	20	49	2016-10-12
11476	PG0244	Якутск	Airbus A319-100	19	116	83	19	2016-10-13
11477	PG0699	Якутск	Bombardier CRJ-200	40	50	20	59	2016-10-13

11478 rows × 8 columns

#### 4.6 Найдите процентное соотношение перелетов по типам самолетов от общего количества.

- соединим таблицы **flights** - **aircrafts**
- добавим вычисляемое поле на основе оконных функций:
  - количество строк в группах по кодам самолета
  - общее количество строк
  - рассчитаем процентное соотношение
- выведем название модели и процентное соотношение

```
query(''
SET search_path TO bookings;

SELECT DISTINCT model AS Тип_самолета,
       ROUND(
         COUNT(*) OVER(PARTITION BY aircraft_code) * 100.00 /
         COUNT(*) OVER() ,2 ) AS процент
FROM flights
     INNER JOIN aircrafts USING(aircraft_code);
'')
```

	Тип_самолета	процент
0	Boeing 737-300	3.85
1	Airbus A321-200	5.89
2	Cessna 208 Caravan	28.00
3	Boeing 767-300	3.69
4	Bombardier CRJ-200	27.32
5	Boeing 777-300	1.84
6	Sukhoi SuperJet-100	25.68
7	Airbus A319-100	3.74

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

- создадим таблицу с данными: город вылета - город назначения - **минимальная** стоимость билета **бизнес** класса, соединив таблицы **flights** - **airports** - **ticket\_flights**
- создадим таблицу с данными: город вылета - город назначения - **максимальная** стоимость билета **эконом** класса, соединив таблицы **flights** - **airports** - **ticket\_flights**
- выберем города назначения из созданных таблиц с условием одинаковых городов вылета, городов назначения и найдем билеты бизнес класса с ценой ниже эконом класса.

```
query(''
SET search_path TO bookings;

WITH cte_b AS (
-- минимальная стоимость перелетов Бизнес классом
SELECT a.city AS city_out, b.city AS city_in, MIN(tf.amount)
FROM flights f
      INNER JOIN airports a ON f.departure_airport = a.airport_code
      INNER JOIN airports b ON f.arrival_airport = b.airport_code
      INNER JOIN ticket_flights tf USING(flight_id)
WHERE fare_conditions = 'Business'
GROUP BY city_out, city_in
),
cte_e AS (
-- максимальная стоимость перелетов Эконом классом
SELECT a.city AS city_out, b.city AS city_in, MAX(tf.amount)
FROM flights f
      INNER JOIN airports a ON f.departure_airport = a.airport_code
      INNER JOIN airports b ON f.arrival_airport = b.airport_code
      INNER JOIN ticket_flights tf USING(flight_id)
WHERE fare_conditions = 'Economy'
GROUP BY city_out, city_in
)
SELECT b.city_in AS Город, COUNT(*) AS количество
FROM cte_b b, cte_e e
WHERE b.city_in = e.city_in
      AND b.city_out = e.city_out
      AND b.min < e.max
GROUP BY b.city_in;
')
```

Город	количество
-------	------------

Таких городов не найдено



## 4.8 Между какими городами нет прямых рейсов?

- создадим множество всех возможных комбинаций городов с аэропортами используя таблицу **airports** и CROSS JOIN
- найдем множество существующих пар городов из таблицы рейсов **flights**
- исключим из множества всех пересечений - множество существующих маршрутов

```
query(''  
SET search_path TO bookings;  
  
SELECT a.city AS город_1, b.city AS город_2  
FROM (SELECT DISTINCT city FROM airports) a  
      CROSS JOIN (SELECT DISTINCT city FROM airports) b  
WHERE a.city != b.city  
EXCEPT  
SELECT a.city, b.city  
FROM flights f  
      INNER JOIN airports a ON f.departure_airport = a.airport_code  
      INNER JOIN airports b ON f.arrival_airport = b.airport_code  
GROUP BY a.city, b.city  
'')
```

	город_1	город_2
0	Новосибирск	Йошкар-Ола
1	Норильск	Череповец
2	Нижневартовск	Архангельск
3	Новокузнецк	Омск
4	Новокузнецк	Ханты-Мансийск
...	...	...
9579	Орск	Мирный
9580	Мирный	Курган
9581	Курск	Нижневартовск
9582	Казань	Усинск
9583	Калуга	Ижевск

9584 rows × 2 columns

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

- для определения расстояния по координатам используем функцию gc\_dist, допускаем при этом что поверхность земли - сферическая
- сформируем CTE:
  - соединим таблицы **flights - airports - aircrafts**
  - получим номер рейса, названия аэропортов, их координаты, тип и дальность полета самолета
  - добавим вычисляемое поле расстояния между аэропортами
- используя CTE выведем информацию по рейсам и сравним расстояние между аэропортами и дальность полета используемого типа самолета

```

query(''
SET search_path TO bookings;

CREATE OR REPLACE FUNCTION gc_dist(
    lat1 double precision, lon1 double precision,
    lat2 double precision, lon2 double precision
) RETURNS double precision
    LANGUAGE plpgsql
AS $$
    -- https://en.wikipedia.org/wiki/Haversine_formula
    -- http://www.movable-type.co.uk/scripts/latlong.html
    DECLARE R INT = 6371; -- km, https://en.wikipedia.org/wiki/Earth_radius
    DECLARE dLat double precision = (lat2-lat1)*PI()/180;
    DECLARE dLon double precision = (lon2-lon1)*PI()/180;
    DECLARE a double precision = sin(dLat/2) * sin(dLat/2) +
        cos(lat1*PI()/180) * cos(lat2*PI()/180) *
        sin(dLon/2) * sin(dLon/2);
    DECLARE c double precision = 2 * asin(sqrt(a));
BEGIN
    RETURN R * c;
EXCEPTION
    -- если координаты совпадают, то получим исключение, а падать нельзя
WHEN numeric_value_out_of_range
    THEN RETURN 0;
END;
$$;

WITH cte AS (
SELECT DISTINCT flight_no,
    a.airport_name AS air_out,
    b.airport_name AS air_in,
    a.latitude, a.longitude, b.latitude, b.longitude,
    ROUND(gc_dist(a.latitude, a.longitude, b.latitude, b.longitude)) AS dist,
    model, range
FROM flights f
    INNER JOIN airports a ON f.departure_airport = a.airport_code
    INNER JOIN airports b ON f.arrival_airport = b.airport_code
    INNER JOIN aircrafts USING(aircraft_code)
)
SELECT flight_no AS рейс,
    air_out AS вылет,
    air_in AS прилет,
    dist AS расстояние,
    model AS тип_самолета,
    range AS дальность,
    CASE
        WHEN range < dist
        THEN CONCAT('превышено на ', dist - range, ' км')
        ELSE CONCAT('норма, запас ', range - dist, ' км')
    END сравнение
FROM cte
''')

```



	рейс	вылет	прилет	расстояние	тип_самолета	дальность	сравнение
0	PG0001	Усть-Илимск	Сургут	1658.0	Bombardier CRJ-200	2700	норма, запас 1042 км
1	PG0002	Сургут	Усть-Илимск	1658.0	Bombardier CRJ-200	2700	норма, запас 1042 км
2	PG0003	Иваново-Южный	Сочи	1502.0	Bombardier CRJ-200	2700	норма, запас 1198 км
3	PG0004	Сочи	Иваново-Южный	1502.0	Bombardier CRJ-200	2700	норма, запас 1198 км
4	PG0005	Домодедово	Псков	639.0	Cessna 208 Caravan	1200	норма, запас 561 км
...	...	...	...	...	...	...	...
705	PG0706	Магадан	Сыктывкар	4881.0	Boeing 767-300	7900	норма, запас 3019 км
706	PG0707	Советский	Сургут	522.0	Boeing 737-300	4200	норма, запас 3678 км
707	PG0708	Сургут	Советский	522.0	Boeing 737-300	4200	норма, запас 3678 км
708	PG0709	Домодедово	Чульман	5015.0	Airbus A319-100	6700	норма, запас 1685 км
709	PG0710	Чульман	Домодедово	5015.0	Airbus A319-100	6700	норма, запас 1685 км

710 rows × 7 columns