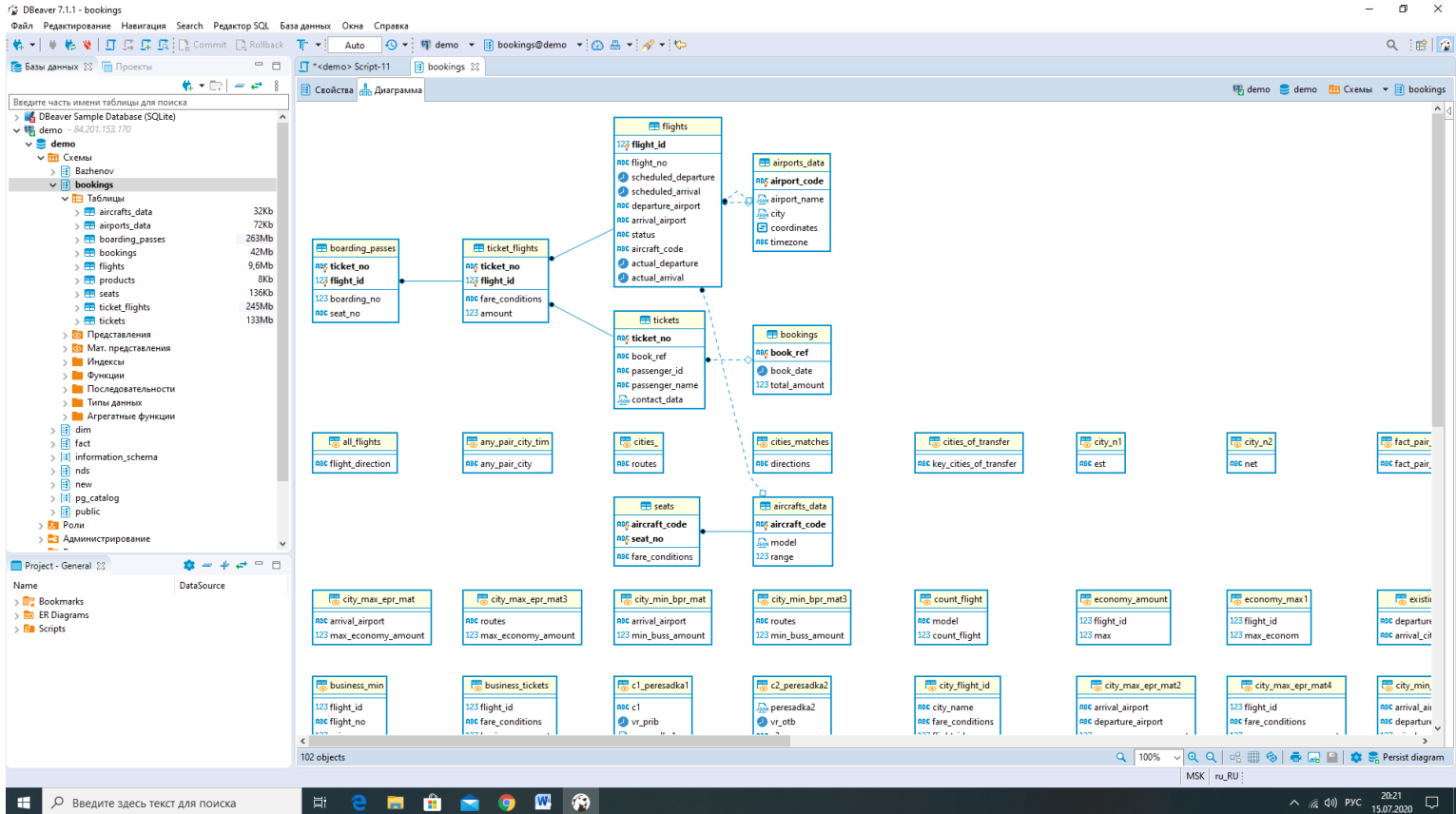


Проектная работа по модулю “SQL и получение данных”

Итоговая работа

1. В работе использовался _облачный_ тип подключения.
2. Скриншот ER-диаграммы из DBeaver`а.



3. Краткое описание БД - из каких таблиц и представлений состоит.

В качестве предметной области выбраны авиаперевозки по России. База данных состоит из 9-ти таблиц:

- aircrafts_data (описание самолетов)
- airports_data (описание аэропортов и направления рейсов)
- boarding_passes (посадочные талоны)
- bookings (бронирования)
- flights (полеты)
- products (товары)
- seats (места)
- ticket_flights (билеты на рейсы)
- tickets (билеты с описанием пассажиров)

Представление "bookings.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

Материализованное представление bookings.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. Развернутый анализ БД - описание таблиц, логики, связей и бизнес области (частично можно взять из описания базы данных, оформленной в виде анализа базы данных)

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).

Список отношений

Имя	Тип	Small	Medium	Big	Описание
aircrafts	таблица	16 kB	16 kB	16 kB	Самолеты
airports	таблица	48 kB	48 kB	48 kB	Аэропорты
boarding_passes	таблица	31 MB	102 MB	427 MB	Посадочные талоны
bookings	таблица	13 MB	30 MB	105 MB	Бронирования
flights	таблица	3 MB	6 MB	19 MB	Рейсы
flights_v	представление	0 kb	0 kb	0 kb	Рейсы
routes	мат. предст.	136 kB	136 kB	136 kB	Маршруты
seats	таблица	88 kB	88 kB	88 kB	Места
ticket_flights	таблица	64 MB	145 MB	516 MB	Перелеты
tickets	таблица	47 MB	107 MB	381 MB	Билеты

Таблица bookings.aircrafts

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

Таблица bookings.airports

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
longitude	float	NOT NULL	Координаты аэропорта: долгота
latitude	float	NOT NULL	Координаты аэропорта: 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)

Таблица bookings.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)

Таблица bookings.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	timestampz	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)

Таблица bookings.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)
```

Таблица bookings.products

123 id	name	count	price	Значение	Метаданные	Назв. каталога	Назв. схемы	Назв. таблицы	Макс. длина	Точность	Масштаб	Тип JDBC	Not Null	Авто
1	Стиральная машина	5	670											
2	Холодильник	0	540											
3	Микроволновка	3	460											
4	Пылесос	2	799											
5	Вентилятор	0	480											
6	Телевизор	7	650											
7	Тостер	2	460											
8	Принтер	4	460											

В данной таблице, видимо, содержатся товары, которыми можно воспользоваться на борту самолетов и цена пользования за каждый из них.

Таблица bookings.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
```

Таблица bookings.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)

Таблица bookings.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)

5. Список SQL запросов с описанием логики их выполнения

Запрос	Описание
1. В каких городах больше одного аэропорта?	
Вариант №1: select ad.city, count(ad.airport_name) from airports_data ad group by ad.city having count(ad.airport_name) > 1 Вариант №2: SELECT a.airport_code as code, a.airport_name, a.city, a.timezone FROM airports a WHERE a.city IN (SELECT aa.city FROM airports aa GROUP BY aa.city HAVING COUNT(*) > 1) ORDER BY a.city, a.airport_code;	Обратились к таблице «airports_data». Больше одного аэропорта имеют города Москва и Ульяновск.
2. В каких аэропортах есть рейсы, выполняемые самолетом с максимальной дальностью перелета?	
select distinct a.airport_code, a.airport_name, a.city, ar.model, ar.range from airports a join flights f on f.departure_airport = a.airport_code join aircrafts ar on ar.aircraft_code = f.aircraft_code where ar.range = (select max(ar.range) from aircrafts ar);	Берем за основу таблицу airports. Присоединяем таблицу aircrafts через таблицу flights для получения данных о дальности полёта. Так как функция max не может использоваться в where напрямую из-за этапа выполнения запросов, то оборачиваем в подзапрос.
3. Вывести 10 рейсов с максимальным временем задержки вылета	
select f.flight_id, f.flight_no, f.scheduled_departure, f.actual_departure, f.actual_departure - f.scheduled_departure as time_fail from flights f where f.actual_departure > f.scheduled_departure order by time_fail desc limit 10;	Мы обратились к таблице flights и выбрали те строки, где фактическое время вылета больше, чем время вылета по расписанию. Также мы создали новый столбец time_fail, в котором вычислили время задержки вылетов самолетов. Ранжировали все по новому столбцу и вывели первые 10 полетов с максимальным временем задержки.
4. Были ли брони, по которым не были получены посадочные талоны?	
SELECT count(*) FROM (tickets t2 JOIN boarding_passes bp ON t2.ticket_no = bp.ticket_no) LEFT OUTER JOIN tickets t3 ON t2.ticket_no = bp.ticket_no WHERE t2.book_ref IS NOT NULL AND bp.boarding_no is NULL;	
5. Найдите свободные места для каждого рейса, их % отношение к общему количеству мест в самолете. Добавьте столбец с накопительным итогом - суммарное количество вывезенных пассажиров из аэропорта за день. Т.е. в этом столбце должна отражаться сумма - сколько человек уже вылетело из данного аэропорта на этом или более ранних рейсах	

за сегодняшний день.	
SELECT ts.flight_id, ts.flight_no, ts.scheduled_departure_local, ts.departure_city, ts.arrival_city, a.model, ts.fact_passengers, ts.total_seats, round(ts.fact_passengers::numeric / ts.total_seats::numeric, 2) AS fraction FROM (SELECT f.flight_id, f.flight_no, f.scheduled_departure_local, f.departure_city, f.arrival_city, f.aircraft_code, count(tf.ticket_no) AS fact_passengers, (SELECT count(s.seat_no) FROM seats s WHERE s.aircraft_code = f.aircraft_code) AS total_seats FROM flights_v f JOIN ticket_flights tf ON f.flight_id = tf.flight_id WHERE f.status = 'Arrived' GROUP BY 1, 2, 3, 4, 5, 6) AS ts JOIN aircrafts AS a ON ts.aircraft_code = a.aircraft_code ORDER BY ts.scheduled_departure_local;	<p>С помощью данного запроса мы выяснили степень заполнения самолетов на всех рейсах.</p> <p>Запрос выдает нам не только число билетов, проданных на данный рейс, и общее число мест в самолете, но также вычисляет отношение этих двух показателей.</p> <p>Самый внутренний подзапрос — total_seats — выдает общее число мест в самолете. Этот подзапрос выполняется для каждой строки, обрабатываемой во внешнем подзапросе, то есть для каждой модели самолета. Для подсчета числа проданных билетов использовалось соединение представления flights_v с таблицей ticket_flights с последующей группировкой строк и вызовом функции count.</p>
6.Найдите процентное соотношение перелетов по типам самолетов от общего количества.	
Select aircraft_code, round (count (flight_no)*100./(select count (flight_no) from flights), 1) From flights Group by aircraft_code	<p>Мы обратились к таблице flights. Вычислили процентное соотношение перелетов от общего количества с помощью функции round.</p> <p>И сгруппировали все по типам самолетов.</p>
7.Были ли города, в которые можно добраться бизнес - классом дешевле, чем эконом-классом в рамках перелета?	
with min_busin as (select ticf.fare_conditions ,min(amount) as business_min from ticket_flights ticf join flights_v f on f.flight_id =ticf.flight_id where ticf.fare_conditions='Business' group by ticf.flight_id,ticf.fare_conditions order by ticf.flight_id asc), max_econ as (select ticf.fare_conditions ,max(amount)) as max_econ from ticket_flights ticf join flights_v f on f.flight_id =ticf.flight_id where ticf.fare_conditions='Economy' group by ticf.flight_id,ticf.fare_conditions order by ticf.flight_id asc) select business_min, max_econ from min_busin join max_econ on ticf.flight_id = f.flight_id where business_min < max_econ order by business_min asc	<p>С помощью CTE мы создали новые столбцы на основе таблицы ticket_flights, чтобы нам вывелись значения по каждому полету – максимальная цена по эконому и минимальная цена по бизнесу. Далее мы сравнили построчно эти два новых столбца, прописав условие, где максимальный эконом будет больше минимального бизнеса. Далее к сожалению не смогла прописать запрос..</p>
8.Между какими городами нет прямых рейсов?	

<pre> with p_air as (select a.city->>'ru' as c_from, b.city->>'ru' as c_to from airports_data a, airports_data b where a.city <> b.city) select distinct * from p_air except select fl.city_dp, fl.city_ar from fl_city fl; </pre>	<p>В запросе было использовано декартово произведение, представления и оператор Ехерт (выводит только те данные из первого набора строк, которых нет во втором наборе).</p>
<p>9.Вычислите расстояние между аэропортами, связанными прямыми рейсами, сравните с допустимой максимальной дальностью перелетов в самолетах, обслуживающих эти рейсы.</p>	
<pre> create view d_c as(select flight_id , flight_no , departure_airport , departure_city , aircraft_code , a.latitude ,a.longitude from flights_v fv join airports a on fv.departure_airport = a.airport_code) create view a_c as (select flight_id , flight_no , arrival_airport , arrival_city , aircraft_code , a.latitude ,a.longitude from flights_v fv join airports a on fv.arrival_airport = a.airport_code) create view calculation as(select d_c.flight_id, departure_city, d_c.latitude as dLatitude, d_c.longitude as dLogitude, arrival_city, a_c.latitude as aLatitude, a_c.longitude as aLongitude, d_c.aircraft_code from d_c join a_c on d_c.flight_id = a_c.flight_id) with final as(select distinct c.departure_city, c.arrival_city, c.aircraft_code, radians(acosd(sind(dLatitude)*sind(alatitude) + cosd(dlatitude)*cosd(alatitude)*cosd(dlogitude - alongitude))) * 6371 as distance_km, a."range" as max_aircraft_distance from calculation c join aircrafts a on c.aircraft_code = a.aircraft_code order by departure_city) select *, (case when distance_km < max_aircraft_distance then 'flight will be great' when distance_km > max_aircraft_distance then 'flight will stop' end) from final </pre>	<p>Добавляю данные о широте и долготе аэропорта вылета Добавляю данные о широте и долготе аэропорта прилета создаю представление, где объединены начальный и конечный город и их координаты Рассчитываю дистанцию между аэропортами и добавляю информацию о максимальной дальности полета самолета Сравниваю расстояние между аэропортами и максимальную дальность полета самолета</p>