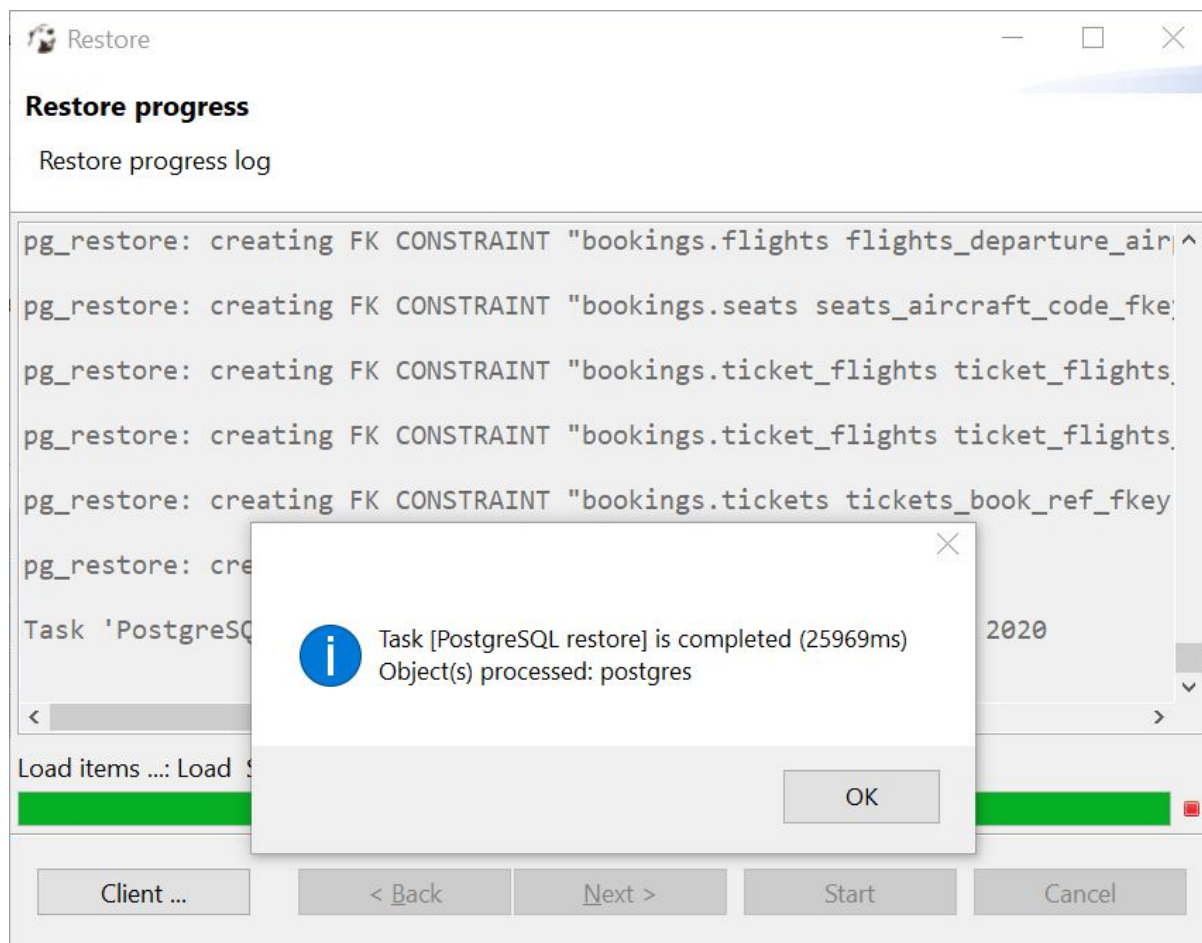
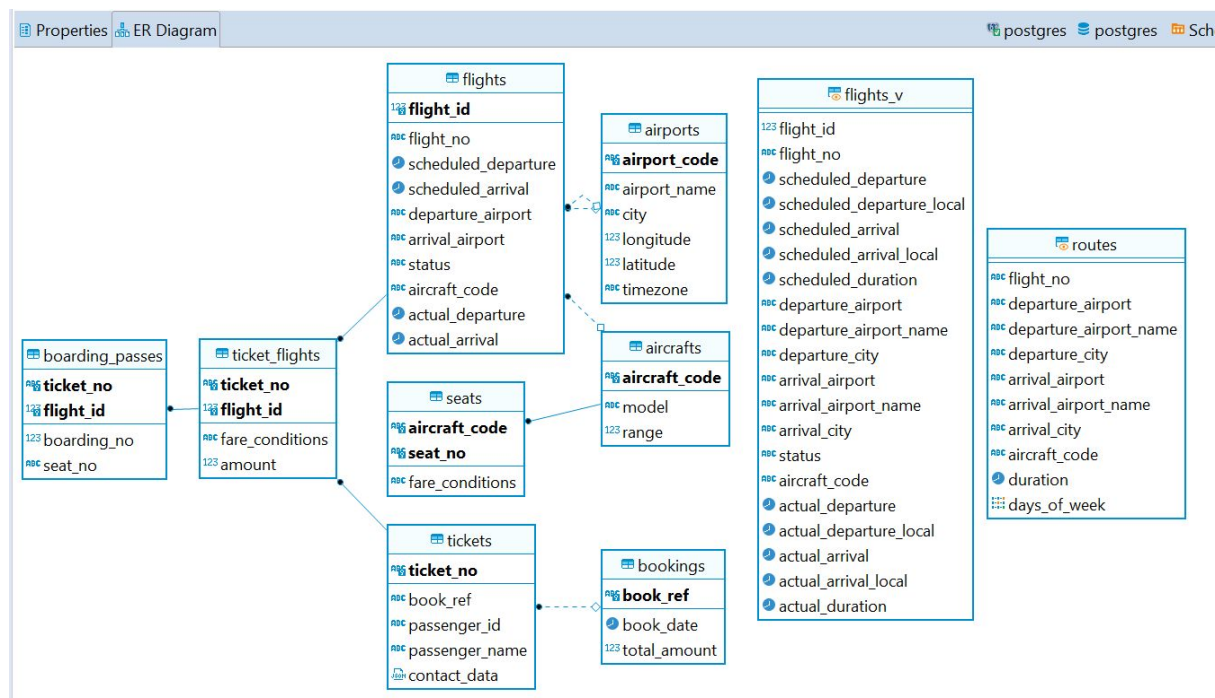


1. В работе использовался локальный тип подключения.  
База была развернута из \*.backup файла (avia.backup):



2. Скриншот ER-диаграммы из DBeaver'a:



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

В БД всего 8 таблиц: 'bookings', 'tickets', 'ticket\_flights', 'boarding\_passes', 'airports', 'flights', 'aircrafts', 'seats'. И 2 представления: 'flights\_v' и 'routes'.

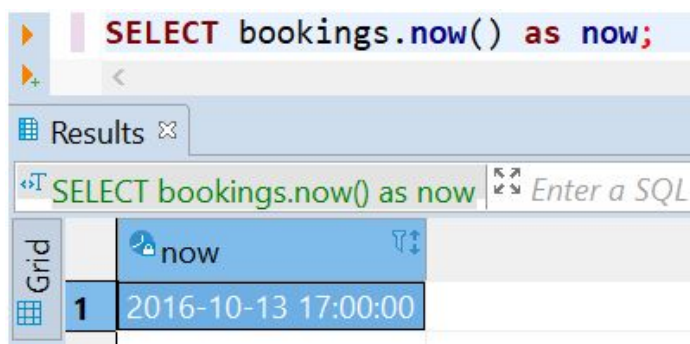
4. Театр начинается с вешалки, а полеты начинаются с бронирования.

В одном бронировании ('bookings') могут быть указаны несколько человек, каждый из которых получит свой отдельный билет ('tickets'), причем один билет может включать несколько перелетов ('ticket\_flights'), если например нет прямого рейса и/или билет сразу 'туда и обратно'.

Рейс ('flights') следует из пункта А в пункт В (airports). Также есть расширенное представление рейсов ('flights\_v') с информацией о наименованиях аэропортов, продолжительностью полета итд. и материализованное представление, где выделена информация только о уникальных маршрутах ('routes'), независимо от дат рейсов.

При регистрации на рейс выдается посадочный талон ('boarding\_passes'), в котором указано место в самолете. Также за каждым рейсом закреплен самолет ('aircrafts'). Количество мест и класс их обслуживания в самолетах хранятся в таблице ('seats').

Сама база представляет собой срез на определенную дату - 13 октября 2016 года:



Results	
SELECT bookings.now() as now	
Grid	now
1	2016-10-13 17:00:00

И можно посмотреть например что на этот момент за месяц прибыло 16.707 рейсов и еще на месяц вперед запланировано 15.383 рейса:

	status	count	min_scheduled_departure	max_scheduled_departure
1	Arrived	16,707	2016-09-13 00:50:00	2016-10-13 16:25:00
2	Cancelled	414	2016-09-16 10:35:00	2016-11-12 19:55:00
3	Departed	58	2016-10-13 08:55:00	2016-10-13 16:50:00
4	Delayed	41	2016-10-13 14:15:00	2016-10-14 16:25:00
5	On Time	518	2016-10-13 16:55:00	2016-10-14 17:00:00
6	Scheduled	15,383	2016-10-14 17:05:00	2016-11-12 19:40:00

5. Список SQL запросов с описанием логики их выполнения:

1) В каких городах больше одного аэропорта?

```
select city, count(city) as num
from airports a
group by city
having count(city) > 1
order by count(city) desc;
```

airports

```
select city, count(city) as num from airports a
```

	city	num
1	Москва	3
2	Ульяновск	2

2) В каких аэропортах есть рейсы, выполняемые самолетом с максимальной дальностью перелета? (- Подзапрос)

```
select distinct departure_airport_name as airport -- выбираем уникальные аэропорты
from routes r -- из маршрутов
where aircraft_code = ( -- где код самолета равен
    select aircraft_code -- коду самолета
    from aircrafts a
    where "range" = (select max("range") from aircrafts))
-- у которого дальность равна максимальной
```

routes

```
select distinct departure_airport_name as airport
```

	airport
1	Внуково
2	Пермь
3	Толмачёво
4	Шереметьево
5	Кольцово
6	Сочи
7	Домодедово

3) Вывести 10 рейсов с максимальным временем задержки вылета (- Оператор LIMIT)

```
select flight_id, departure_airport_name, arrival_airport_name,  
       (actual_departure - scheduled_departure) as delay  
-- доп.столбец - разность актуального и запланированного времени вылета  
from flights_v fv  
where actual_departure notnull -- отсекаем невылетевшие рейсы  
order by delay desc -- сортируем по убыванию задержки  
limit 10; -- оставляем топ-10
```

	123 flight_id	abc departure_airport_name	abc arrival_airport_name	delay
1	14,750	Пермь	Кольцово	04:37:00
2	1,408	Домодедово	Новый Уренгой	04:28:00
3	24,253	Краснодар	Баратаевка	04:27:00
4	22,778	Советский	Уфа	04:20:00
5	2,852	Домодедово	Курск-Восточный	04:18:00
6	21,684	Петрозаводск	Домодедово	04:18:00
7	11,426	Кольцово	Магнитогорск	04:16:00
8	9,891	Толмачёво	Шереметьево	04:16:00
9	13,645	Уфа	Ухта	04:14:00
10	4,781	Внуково	Курган	04:08:00

4) Были ли брони, по которым не были получены посадочные талоны?  
(- Верный тип JOIN)

```
select count(book_ref)  
from tickets t  
left join ticket_flights tf on t.ticket_no = tf.ticket_no  
left join boarding_passes bp on (tf.ticket_no, tf.flight_id) = (bp.ticket_no, bp.flight_id)  
left join flights f on tf.flight_id = f.flight_id  
where (seat_no isnull) and (actual_departure notnull);  
-- соединяем билеты с перелетами и талонами и рейсами  
-- и смотрим, чтобы одновременно не было места (не выдан талон) и рейс уже улетел.
```

	123 count
1	0



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 (- Оконная функция) (- Подзапрос)

```
select *,
sum("non-empty seats") over (partition by airport, dep_day order by actual_departure) as "cumulative"
from (
select distinct flight_id, airport, dep_day, actual_departure,
"non-empty seats", ("all seats" - "non-empty seats") as "empty seats",
round(100.0 * ("all seats" - "non-empty seats") / "all seats", 2) as "% empty seats"
from (
select f.flight_id, f.departure_airport as "airport", f.aircraft_code,
f.actual_departure, f.actual_departure::date as dep_day,
s.seat_no, bp.boarding_no, bp.seat_no,
count(bp.seat_no) over (partition by f.flight_id) as "non-empty seats",
count(s.seat_no) over (partition by f.flight_id) as "all seats"
from flights f
join seats s on f.aircraft_code = s.aircraft_code
left join boarding_passes bp on (f.flight_id, s.seat_no) = (bp.flight_id, bp.seat_no)
where actual_departure notnull
) as slc1
order by airport, dep_day, actual_departure
) as slc2
```

	flight_id	airport	dep_day	actual_departure	non-emp	empty seats	% empty seats	cumulative
1	21,103	AAQ	2016-09-13	2016-09-13 11:35:00	3	94	96.91	3
2	20,993	AAQ	2016-09-13	2016-09-13 12:08:00	51	79	60.77	54
3	21,096	AAQ	2016-09-14	2016-09-14 11:29:00	3	94	96.91	3
4	20,982	AAQ	2016-09-14	2016-09-14 12:07:00	50	80	61.54	53
5	21,049	AAQ	2016-09-15	2016-09-15 08:14:00	0	130	100	0
6	21,084	AAQ	2016-09-15	2016-09-15 11:26:00	5	92	94.85	5
7	21,041	AAQ	2016-09-15	2016-09-15 12:09:00	50	80	61.54	55

с комментариями:

```
select *,
sum("non-empty seats") over (partition by airport, dep_day order by actual_departure) as "cumulative"
-- добавляем сверху окно над (аэропортом + датой)
-- и считаем накопительную сумму (количество непустых мест) в порядке вылета внутри окна
from (
select distinct flight_id, airport, dep_day, actual_departure,
-- оставляем только по одной стрке на один рейс
"non-empty seats", ("all seats" - "non-empty seats") as "empty seats",
round(100.0 * ("all seats" - "non-empty seats") / "all seats", 2) as "% empty seats"
-- считаем отношение пустых мест к общему количеству мест и переводим это в процент
from (
select f.flight_id, f.departure_airport as "airport", f.aircraft_code,
f.actual_departure, f.actual_departure::date as dep_day, -- выделяем день из timestamp
s.seat_no, bp.boarding_no, bp.seat_no,
count(bp.seat_no) over (partition by f.flight_id) as "non-empty seats",
count(s.seat_no) over (partition by f.flight_id) as "all seats"
-- с помощью окна над рейсами считаем количество всего мест и мест по посадочным талонам
from flights f
join seats s on f.aircraft_code = s.aircraft_code
left join boarding_passes bp on (f.flight_id, s.seat_no) = (bp.flight_id, bp.seat_no)
-- присоединяем к местам в самолете занятые места по 'ключу' (flight_id + seat_no)
where actual_departure notnull
) as slc1
order by airport, dep_day, actual_departure
) as slc2
```

6) Найдите процентное соотношение перелетов по типам самолетов от общего количества. (- Подзапрос) (- Оператор ROUND)

```

select *, round(100.0 * "num tf" / sum("num tf") over (), 2) as "% tf"
-- доля количества перелетов по модели от общего количества
from (
  select a.model, count(tf.ticket_no) as "num tf"
  -- количество билетов(перелетов) по всем рейсам модели
  from flights f
  join aircrafts a on f.aircraft_code = a.aircraft_code
  join ticket_flights tf on f.flight_id = tf.flight_id
  group by a.model
) as slc

```

	model	num tf	% tf
1	Airbus A319-100	52,853	5.05
2	Airbus A321-200	107,129	10.24
3	Boeing 737-300	86,102	8.23
4	Boeing 767-300	124,774	11.93
5	Boeing 777-300	144,376	13.81
6	Bombardier CRJ-200	150,122	14.36
7	Cessna 208 Caravan	14,672	1.4
8	Sukhoi SuperJet-100	365,698	34.97

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

```

with flight_fare_conditions_amount_cte as (
  select tf.flight_id, tf.fare_conditions,
  (case when fare_conditions = 'Business' then min(tf.amount) else max(tf.amount) end) as amount
  -- для каждого рейса выберем строку с минимальной стоимостью бизнес-класса
  -- и строки с максимальной стоимостью для других классов (эконом, премиум..)
  from ticket_flights tf
  group by tf.flight_id, tf.fare_conditions)

```

	flight_id	fare_conditions	amount
1	1	Business	20,000
2	1	Economy	7,300
3	2	Business	20,000
4	2	Economy	7,300

далее

```

select slc.flight_id, "business", "economy", fv.arrival_city
from (
  select flight_id,
  max(amount) filter (where fare_conditions = 'Business') as "business",
  max(amount) filter (where fare_conditions = 'Economy') as "economy"
  -- преобразуем строки 'Business' и 'Economy' в столбцы
  from flight_fare_conditions_amount_cte
  group by flight_id) as slc
join flights_v fv on slc.flight_id = fv.flight_id
-- добавим информацию по рейсам для извлечения города назначения
where economy > business
-- ищем строки где бизнес-класс дешевле эконома - таких нет :)

```

	flight_id	business	economy	arrival_city



- 8) Между какими городами нет прямых рейсов?  
 (- Декартово произведение в предложении FROM)  
 (- Представления) (- Оператор EXCEPT)

```
create view all_cities_1 as (select distinct a.city from airports a order by a.city)
create view all_cities_2 as (select distinct a.city from airports a order by a.city)
-- всего 101 город
select * from all_cities_1 c1 cross join all_cities_2 c2
-- декартово произведение списка уникальных городов (по сути самого на себя)
where c1.city != c2.city
-- исключая перелеты внутри города :) дает 10.100 уникальных упорядоченных пар
except (select distinct fv.departure_city, fv.arrival_city from flights_v fv)
-- исключаем 516 уникальных действующих маршрутов
-- и получаем 9.584 несуществующих маршрута
```

Results

select \* from all\_cities\_1 c1 cross join all\_cities\_2 c2

	city	city
1	Новосибирск	Йошкар-Ола
2	Норильск	Череповец
3	Нижевартовск	Архангельск
4	Новокузнецк	Омск
5	Новокузнецк	Ханты-Мансийск

- 9) Вычислите расстояние между аэропортами, связанными прямыми рейсами, сравните с допустимой максимальной дальностью перелетов в самолетах, обслуживающих эти рейсы \* (- Оператор RADIANS или использование sind/cosd)

```
with connected_airports_cte as (
  select distinct f.departure_airport as departure,
  f.arrival_airport as arrival, f.aircraft_code as aircraft, ac."range"
  -- из таблицы рейсов выбираем уникальные маршруты
  from flights f
  join aircrafts ac on f.aircraft_code = ac.aircraft_code)
  -- добавляем максимальную дальность полета в километрах "range" из aircrafts
```

	departure	arrival	aircraft	range
1	OVS	NJC	CR2	2,700
2	NNM	ARH	CR2	2,700
3	SVO	KGD	733	4,200

```
select d_ap as departure, a_ap as arrival, "L km",
"range" as "range km", "range" - "L km" as "rest km"
-- добавляем столбик "остаток" (запас хода) в километрах
from (
  select *, round(6371.0 * acos(sind(a_lat)*sind(d_lat) +
  cosd(a_lat)*cosd(d_lat)*cosd(a_long - d_long))::numeric, 2) as "L km"
  -- вычисляем по формуле расстояние в радианах и переводим в километры
  from (
    select ca.departure, da.latitude as d_lat, da.longitude as d_long,
    da.airport_name as d_ap, ca.arrival, aa.latitude as a_lat,
    aa.longitude as a_long, aa.airport_name as a_ap, ca.aircraft, ca."range"
    from connected_airports_cte ca
    join airports da on ca.departure = da.airport_code
    join airports aa on ca.arrival = aa.airport_code) as sc1) as sc2
  -- к связанным аэропортам добавляем их широту и долготу, а также названия
order by "rest km"
```

	departure	arrival	L km	range km	rest km
1	Шереметьево	Кемерово	2,994.44	3,000	5.56
2	Кемерово	Шереметьево	2,994.44	3,000	5.56
3	Домодедово	Кемерово	2,986.35	3,000	13.65
4	Кемерово	Домодедово	2,986.35	3,000	13.65