

Итоговая работа

по курсу

SQL и получение данных

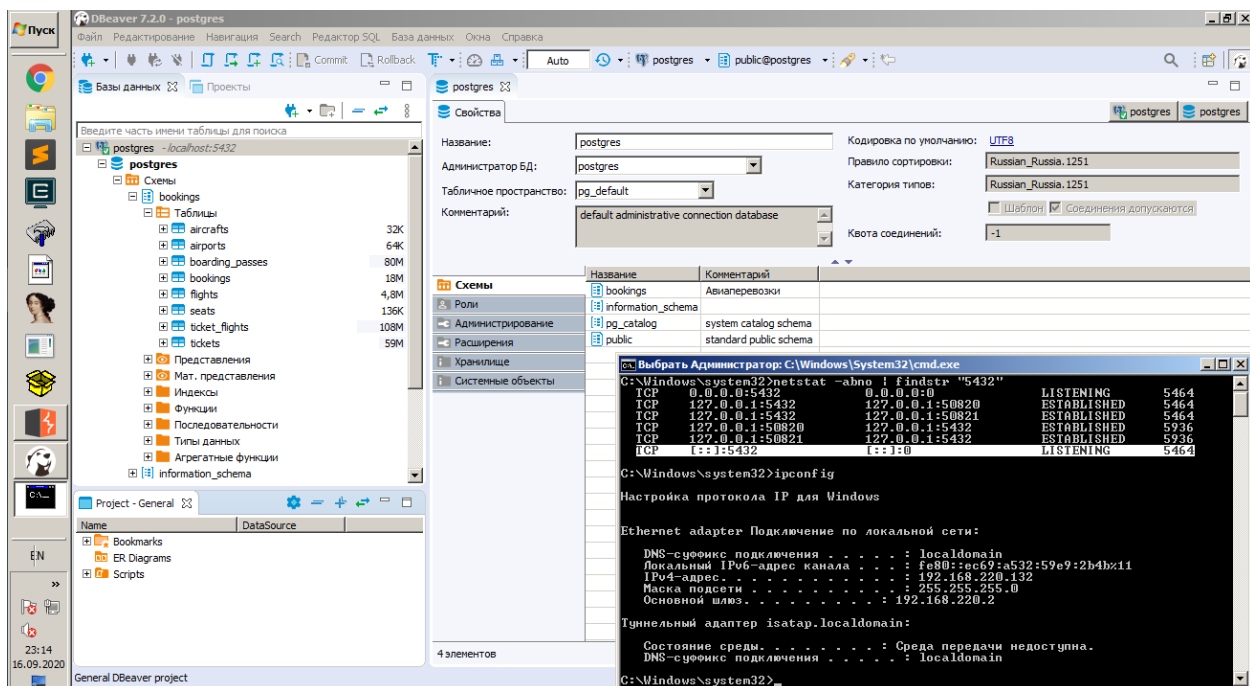
Михаил Гудырин

Прага, 2020, netology.ru

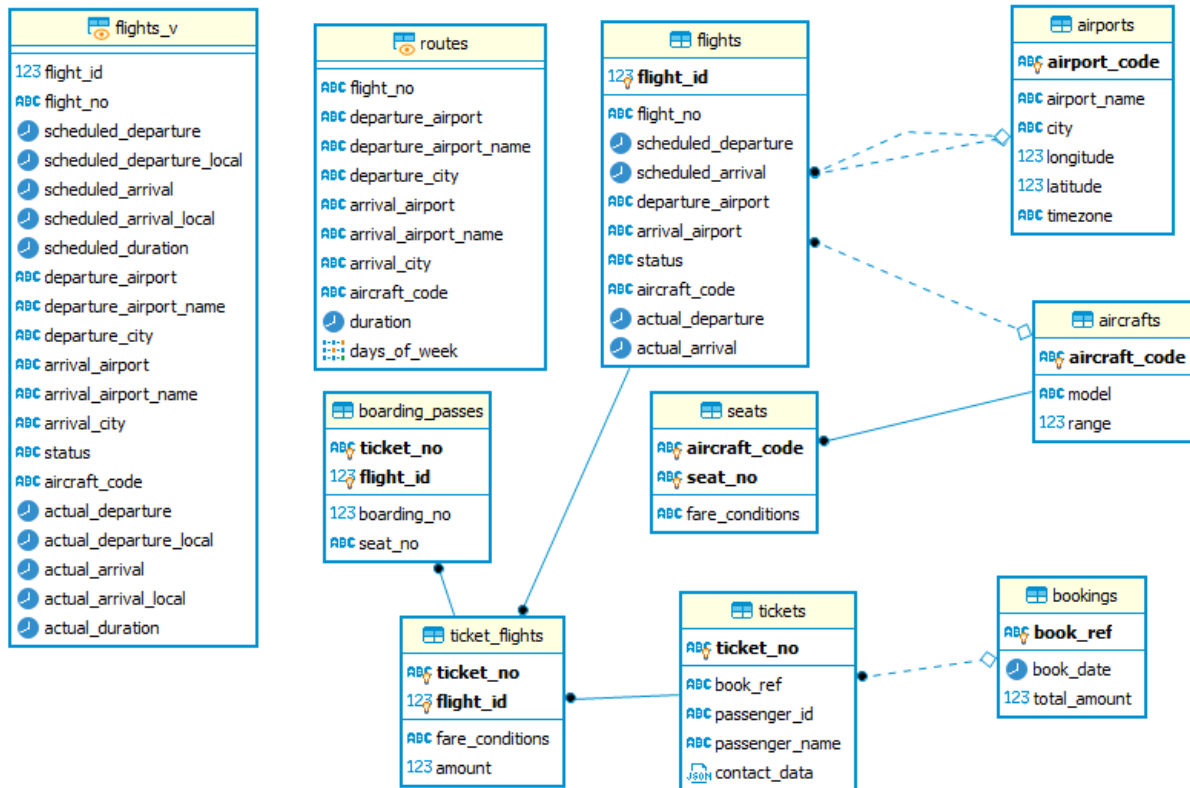
1. В работе использовался локальный тип подключения, в частности использовалась VM-ка с Win 7.

-если база была развернута из *.sql или *.backup файла, необходимо приложить скриншот успешного импорта или восстановления

Скриншот:



2. Скриншот ER-диаграммы из DBeaver`а согласно Вашего подключения:



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

Все таблицы видно на ER диаграмме, как они называются, так я и понимаю их смысл. Это БД авиакомпании либо компании продающей билеты. Учебной, т.к. у настоящей АК БД чуток побольше (я видел у Czech Airlines, когда они еще были самостоятельной компанией) 😊.

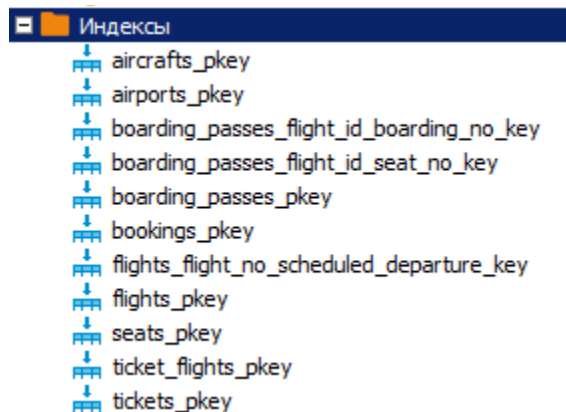
4. Развернутый анализ БД - описание таблиц, логики, связей и бизнес области (частично можно взять из описания базы данных, оформленной в виде анализа базы данных).

- большая таблица **flights**, собственно содержит информацию по рейсам, связана с таблицами аэропортов, самолетов и билетов через соединительную таблицу **ticket_flights**
- **airports** – таблица аэропортов, соединена с таблицей рейсов
- **aircrafts** – таблица самолетов, соединена с таблицами рейсов и таблицей мест в самолете
- **seats** – таблица мест в самолете. Она просто тут есть 😊, с ней не связано ничего научно-технологического.
- **bookings** – таблица брони, соединена с таблицей билетов
- **tickets** – таблица билетов, соединена с таблицами брони и посадочных талонов через соединительную таблицу **ticket_flights**
- **ticket_flights** – соединительная таблица между таблицами билетов и рейсов

- boarding_passes – таблица посадочных талонов, по сути добавляет информацию к записям из соединительной ticket_flights

Так-же имеем практичную вьюху flights_v, которая позволяет получить детальную инфу по рейсам, и материальную вьюху routes, содержащую информацию по маршрутам.

Информация по индексам:



5. Список SQL запросов с описанием логики их выполнения:

5.1 В каких городах больше одного аэропорта?

```
select city, count(city)
from airports
group by city
having count(city) > 1;
```

5.2 В каких аэропортах есть рейсы, выполняемые самолетом с максимальной дальностью перелета? (подзапрос)

```
select distinct departure_airport
from flights
where aircraft_code = (select aircraft_code from aircrafts order by range desc limit 1);
```

5.3 Вывести 10 рейсов с максимальным временем задержки вылета (оператор LIMIT).

```
select *, (actual_departure - scheduled_departure) as delay
from flights
where status = 'Arrived'
order by delay desc limit 10;
```

5.4 Были ли брони, по которым не были получены посадочные талоны? (верный тип JOIN)

```
select *
from bookings
left join tickets t using(book_ref)
full outer join boarding_passes bp using(ticket_no)
where bp.boarding_no is null and t.ticket_no is not null;
```

5.5 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 f.flight_id, f.aircraft_code, capacity.amount_of_places, passengers_onboard, f.actual_departure,
((100.0 * (capacity.amount_of_places - passengers_onboard)) / capacity.amount_of_places) as
percentage_free,
sum(passengers_onboard) over (partition by f.departure_airport, extract(day from f.actual_departure) order
by f.actual_departure) as dispatched_passengers
from flights f
left join (
    select f.flight_id, count(bp.boarding_no) as passengers_onboard -- количество пассажиров на рейсе
    from flights f
    left join boarding_passes bp using(flight_id)
    group by f.flight_id
) as passengers_onboard using(flight_id)
left join (
    select a.aircraft_code, count(s.seat_no) as amount_of_places -- количество мест в самолете
    from aircrafts a
    left join seats s using(aircraft_code)
    group by a.aircraft_code
) as capacity using(aircraft_code);
```

5.6 Найдите процентное соотношение перелетов по типам самолетов от общего количества (подзапрос, оператор ROUND).

```
select f.aircraft_code,
round(100.0 * count(f.flight_id) / ((select count(f.flight_id) from flights f))) as percent_flights
from flights f
group by f.aircraft_code;
```

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

```
with task_cte as (
    select a.city, econom_max.max as econom_max, business_min.min as business_min,
    econom_max.departure_airport as econom_dep_airport, econom_max.arrival_airport as
econom_arr_airport,
    business_min.departure_airport as business_dep_airport, business_min.arrival_airport as
business_arr_airport
    from airports a
    left join(
        select a.city, f.departure_airport, f.arrival_airport, tf.fare_conditions, max(tf.amount)
        -- максимальная стоимость в каждый город эконом классом
        from ticket_flights tf
        left join flights f using(flight_id)
        left join airports a on f.arrival_airport = a.airport_code
        where tf.fare_conditions = 'Economy'
        group by a.city, tf.fare_conditions, f.flight_id
    ) as econom_max using(city)
    left join(
        select a.city, f.departure_airport, f.arrival_airport, tf.fare_conditions, min(tf.amount)
        -- минимальная стоимость в каждый город бизнес классом
        from ticket_flights tf
        left join flights f using(flight_id)
        left join airports a on f.arrival_airport = a.airport_code
        where tf.fare_conditions = 'Business'
        group by a.city, tf.fare_conditions, f.flight_id
    ) as business_min using(city)
```

```

        ) as business_min using(city)
    )
select * from task_cte
where econom_max > business_min and econom_dep_airport = business_dep_airport;

```

5.8 Между какими городами нет прямых рейсов? (декартово произведение в предложении FROM, представления, оператор EXCEPT)

```

create view airports_1 as
select r.departure_airport
from routes r

create view airports_2 as
select r.arrival_airport
from routes r

with cte_task8 as (
select *
from airports_1, airports_2
where departure_airport != arrival_airport
except
select f.departure_airport, f.arrival_airport
from flights f
group by f.departure_airport, f.arrival_airport
)
select a1.city, a2.city
from cte_task8 c
inner join airports a1 on c.departure_airport = a1.airport_code
inner join airports a2 on c.arrival_airport = a2.airport_code;

```