Итоговая работа

по курсу

**SQL и получение данных**

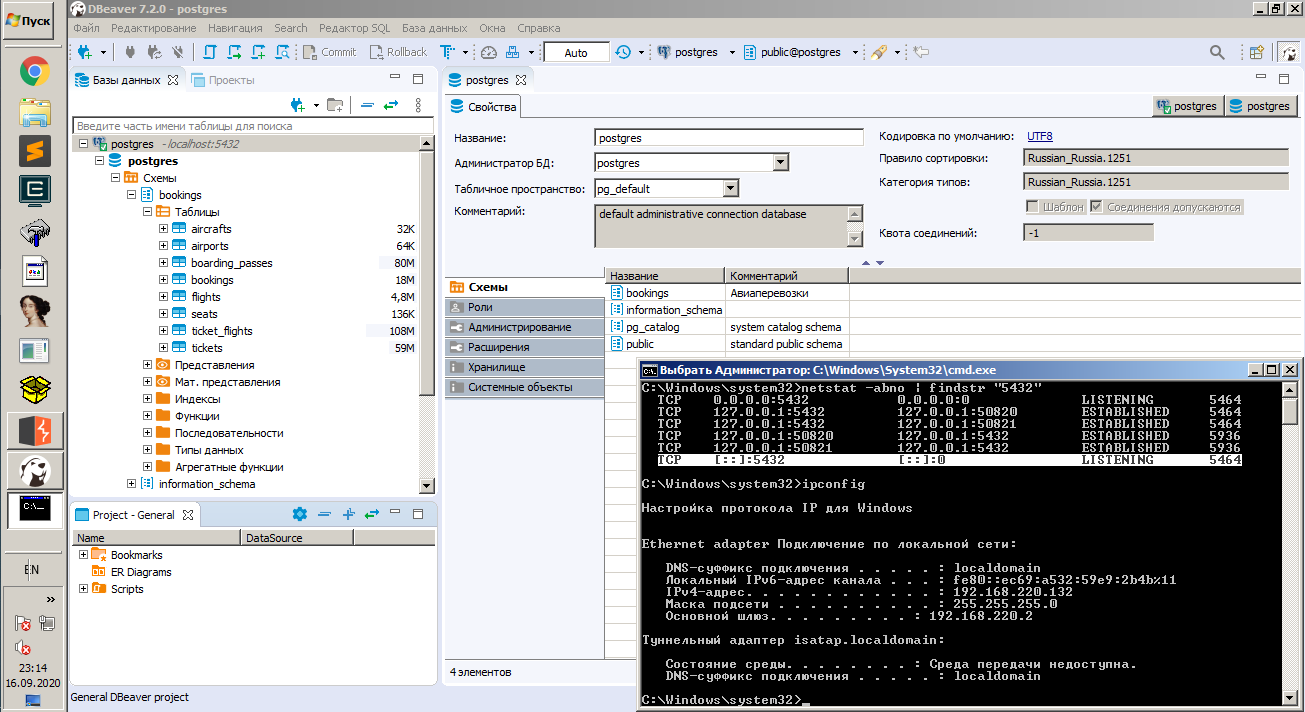
Михаил Гудырин

Прага, 2020, netology.ru

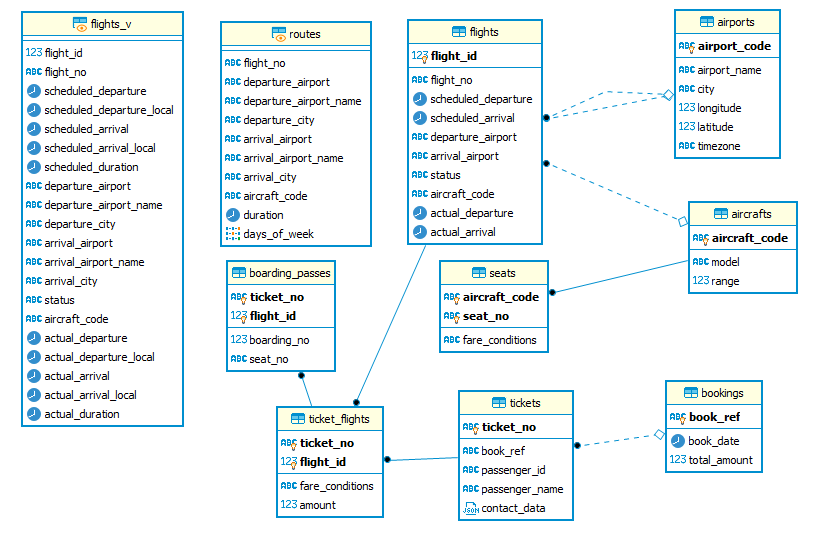
1. В работе использовался **локальный** тип подключения, в частности использовалась ВМ-ка с Win 7.

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

Скриншот:



2. Скриншот ER-диаграммы из DBeaver`a 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 по-больше (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-техлонологического.

- bookings – таблица броней, соединена с таблицей билетов

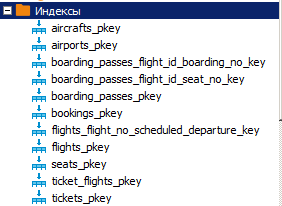
- tickets – таблица билетов, соединена с таблицами броней и посадочных талонов через соединительную таблицу ticket\_flights

- 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)

)

**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;