

-- 1 В каких городах больше одного аэропорта?

```
select *
from (select
      city->>'ru' as city,
      count(airport_name) airports
      from airports_data ad
      group by city
      order by airports desc
      ) c_a
where c_a.airports > 1
```

Комментарий:

Вывожу список всех аэропортов и сортирую по городам, далее оставляю только те где больше одного аэропорта

№ 2

Вопрос : В каких аэропортах есть рейсы, выполняемые самолетом с максимальной дальностью перелета?
В решении обязательно должно быть использовано -
Подзапрос

```
select *
from (select max(range) as distance,
      airport
      from (select f.departure_airport ,
                  f.aircraft_code ,
                  ad.airport_name->> 'ru' as airport,
                  acd.range
                  from flights f
                  join airports_data ad on ad.airport_code = f.departure_airport
                  join aircrafts_data acd on acd.aircraft_code = f.aircraft_code) as info
      group by info.airport
      order by distance desc) foo
where distance = any(select range
                     from (select range,
```

```

row_number()over(order by range desc) rating
from aircrafts_data ad) max_range
where max_range.rating = 1)

```

Код может показаться сложным на первый взгляд наверное, но если посмотреть на последнюю вложенность

```

(select range,
row_number()over(order by range desc) rating
from aircrafts_data ad) max_range
where max_range.rating = 1) - где я вывожу цифру

```

самой дальней дистанции и сравниваю с основным запросом, то все вроде бы просто и логично. Хотя, уверен, можно решить задачу множеством способом. Я выбрал данный.

Вот еще вариант через **rank()** функцию:

```

select departure_airport,
       airport,
       range
from (select f.departure_airport ,
            f.aircraft_code ,
            ad.airport_name->> 'ru' as airport,
            acd.range,
            rank()over(order by acd.range desc)
from flights f
join airports_data ad on ad.airport_code = f.departure_airport
join aircrafts_data acd on acd.aircraft_code = f.aircraft_code) foo
where rank = 1
group by foo.departure_airport, foo.aircraft_code, foo.airport, foo.range, foo.rank

```

- Все также работает.

№ -3

Вопрос : Вывести 10 рейсов с максимальным временем задержки вылета

В решении обязательно должно быть использовано - Оператор LIMIT

```

select flight_id ,
       flight_no,
       departure_airport,
       arrival_airport,
       departures as delays_in_departure
from (select *,
       rank()over(order by departures desc)
       from(select flight_id ,
                  flight_no ,
                  departure_airport ,
                  arrival_airport ,
                  (actual_departure - scheduled_departure) as departures
                  from flights f) delays) delays
where rank > 1
limit 10

```

Комментарий:

С помощью функции rank() я отфильтровал значение NULL которое всегда выползало вверху при убывающем упорядовании order by desc. Т.е. Те рейсы, время которых еще не подошло. Отсортировал по убывающей разницу в (actual_departure - scheduled_departure) as departures и лимитом оставил только 10

№ 4

Вопрос : Были ли брони, по которым не были получены посадочные талоны?

В решении обязательно должно быть использовано - Верный тип JOIN */

```

with b_w_b as (
    select max(rank) booking_who_boarded -- 502 052
    from (select --(book_ref),
              rank()over(order by book_ref)

```

```

from (select (t.ticket_no) ,
            (b.book_ref) ,
            bp.flight_id ,
            (bp.boarding_no)
from bookings b
inner join tickets t on b.book_ref = t.book_ref
inner join boarding_passes bp on bp.ticket_no = t.ticket_no
--group by bp.flight_id
order by bp.flight_id ) foo
group by book_ref) foo
)
select (count(book_ref) - b_w_b.booking_who_boarded) nr_of_bookings_not_boarded
from b_w_b, bookings b
group by b_w_b.booking_who_boarded

```

Комментарий:

В данном задании за счет Inner Join я отсекаю все Bookings которые не соответствуют посадочным талонам и данную сумму отнял от общего количества Bookings, получился результат 91 381.

№ 5

Вопрос : 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.

В решении обязательно должно быть использовано :

- Оконная функция
- Подзапросы или cte

Свободные места и процент к общему

```
with cte as (
    select f.departure_airport ,
           ad.city->> 'ru' as departure_city,
           f.arrival_airport,
           f.flight_id,
           f.aircraft_code ,
           count(bp.seat_no) seats_ocupied
    from flights f
    inner join boarding_passes bp on bp.flight_id = f.flight_id
    join airports_data ad ON ad.airport_code = f.departure_airport
    group by f.departure_airport ,
             f.arrival_airport,
             f.flight_id,
             f.flight_no ,
             f.aircraft_code,
             ad.city
)
select foo.aircraft_code,
       foo.seats_amount,
       cte.seats_ocupied,
       (foo.seats_amount - cte.seats_ocupied) amount_of_empty_seat_per_flight,
       round(((foo.seats_amount - cte.seats_ocupied) / foo.seats_amount ::
numeric * 100), 2) percents_of_empty_seats,
       cte.departure_airport ,
       cte.departure_city,
       cte.arrival_airport,
       cte.flight_id
from (select aircraft_code ,
             count(seat_no) seats_amount
      from seats s
      group by aircraft_code) foo
join cte on cte.aircraft_code = foo.aircraft_code
```

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.

```

with cte as (select f.departure_airport ,
                  f.arrival_airport,
                  f.flight_id,
                  f.flight_no ,
                  count(bp.seat_no) seats_occupied,
                  sum(count(bp.seat_no))over(partition by flight_no
order by f.flight_no, f.scheduled_departure desc),
                  f.scheduled_departure
from flights f
inner join boarding_passes bp on bp.flight_id = f.flight_id
join airports_data ad ON ad.airport_code = f.departure_airport
group by f.departure_airport ,
         f.arrival_airport,
         f.flight_id,
         f.flight_no
         )

select *
from cte

```

№ 6

Вопрос : Найдите процентное соотношение перелетов по типам самолетов от общего количества.

В решении обязательно должно быть использовано - Подзапрос

- Оператор ROUND

```

with cte as (select -- count(flight_id) -- 65664
                  ad.aircraft_code,
                  count(f.flight_id) as nr_of_flights
from flights f
join aircrafts_data ad on ad.aircraft_code = f.aircraft_code

```

```

group by ad.aircraft_code)
select * ,
        round(nr_of_flights / (select sum(nr_of_flights) :: numeric from cte) * 100, 2) as
percent
from cte

```

Взял данные по всем перелетам из таблицы flight и сгруппировал по каждому типу самолета затем высчитал процентное соотношение из общей суммы

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

```

with cte as(select tf1.amount as price_business_class,
                  tf2.amount as price_economy_class,
                  (tf1.amount - tf2.amount) difference_business_economy,
                  tf1.flight_id
from(select tf.amount,
            tf.flight_id
      --concat(f.departure_airport, ' ',
ad.city->>'ru') departure_city,
      --concat(f.arrival_airport, ' ',
ad.city->>'ru') arrival_city
      from ticket_flights tf
      join flights f on f.flight_id = tf.flight_id
      --join airports_data ad on ad.airport_code =
f.departure_airport
      where tf.fare_conditions = 'Business') as tf1
join ticket_flights tf2 on tf2.flight_id = tf1.flight_id
where tf2.fare_conditions = 'Economy'
order by flight_id, difference_business_economy
)

```

```

select cte.difference_business_economy,
       cte.price_business_class,
       cte.price_economy_class,
       cte.flight_id,
       case
         when cte.difference_business_economy < 0 then 'билеты_бизнес_дешевле'
         else 'билеты_эконом_дешевле'
       end
from cte
group   by      cte.difference_business_economy,      cte.price_business_class,
cte.price_economy_class, cte.flight_id
order by cte.flight_id

```

Комментарий:

В 7-м задании я разделил билеты на бизнес и эконом по каждому рейсу и вычел одно из другого. Дешевого бизнес класса не оказалось.

№ 8

Вопрос : Между какими городами нет прямых рейсов?

В решении обязательно должно быть использовано : -
Декартово произведение в предложении FROM

- Самостоятельно созданные представления (если облачное подключение, то без представления)

- Оператор EXCEPT

```

with cte as(select *
              from (select *
                    from (select ad.airport_code as city_1,
                                ad2.airport_code as city_2
                           from airports_data ad , airports_data ad2)
                    dekart_cities
                    where city_1 != city_2) foo
              except
              select departure_airport ,

```



```

                                arrival_airport
                                from flights f)
select concat(city_1 , ' ', ad1.airport_name ->>'ru') as "С этого аэропорта не летают
В - ",
       concat(ad2.airport_name ->>'ru' , ' ', city_2) as "данные города "
from cte
left join airports_data ad1 on ad1.airport_code = cte.city_1
left join airports_data ad2 on ad2.airport_code = cte.city_2
order by city_1, city_2

```

С помощью декартова произведения вывел все возможные комбинации перелетов по городам/аэропортам, далее с помощью `except` вычел действующие перелеты из всех комбинаций, остались только те, между которыми нет прямых перелетов.

№ 9

Вопрос : Вычислите расстояние между аэропортами, связанными прямыми рейсами, сравните с допустимой максимальной дальностью перелетов в самолетах, обслуживающих эти рейсы *

В решении обязательно должно быть использовано :: -
Оператор RADIANS или использование sind/cosd
- CASE

```

select t."город вылета",
       t."город прилета",
       t.distance_in_km,
       t.aircraft_code,
       acd.range as "дальность перелета самолета"
from (with cte as(select concat(t.departure_airport , ' ',
                                ad1.city->>'ru') "город вылета" ,

```

```

        ad1.coordinates[0] Latitude_depart,
        ad1.coordinates[1] Longitude_depart ,
        concat(t.arrival_airport ,
        ad2.city->> 'ru') "город прилета",
        ad2.coordinates[0] Latitude_arrive,
        ad2.coordinates[1] Longitude_arrive ,
        t.aircraft_code
    from(select departure_airport ,
            arrival_airport ,
            aircraft_code
        from flights f
        group by f.departure_airport , f.arrival_airport, f.aircraft_code
        order by departure_airport, arrival_airport) t
    join airports_data ad1 on t.departure_airport = ad1.airport_code
    join airports_data ad2 on t.arrival_airport = ad2.airport_code)
select * ,
        round(111.111 *
        DEGREES(ACOS(LEAST(1.0, COS(RADIANS(cte.Latitude_depart))
        * COS(RADIANS(cte.Latitude_arrive))
        * COS(RADIANS(cte.Longitude_depart - cte.Longitude_arrive))
        + SIN(RADIANS(cte.Latitude_depart))
        * SIN(RADIANS(cte.Latitude_arrive)))))):: numeric) AS distance_in_km
    from cte )t
join aircrafts_data acd on acd.aircraft_code = t.aircraft_code

```

/* Обратите внимание, что постоянная 111,1111 - это количество километров на градус широты, основанное на старом наполеоновском определении метра как одной десяти тысячной расстояния от экватора до полюса. Это определение достаточно близко для работы по поиску местоположения. */