Орищенко Виталий

@Vitosh992

Группа 1

**Архитектор ИИ**

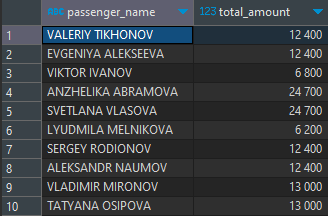
**Промежуточная аттестация 2**

1. Используя SQL язык и произвольные две таблицы из модели данных необходимо объединить их различными способами (UNION, JOIN)

Список пассажиров и их плата за билеты:

**select** t.passenger\_name, b.total\_amount **from** bookings.tickets t

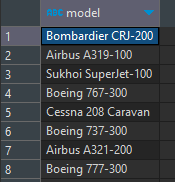
**join** bookings.bookings b **on** t.book\_ref = b.book\_ref



Список моделей самолетов, активных в таблице бронирований:

(**select** **distinct** a.model **from** bookings.aircrafts a

**inner** **join** bookings.flights f **on** f.aircraft\_code = a.aircraft\_code)



1. Используя SQL язык напишите запрос с любым фильтром WHERE к произвольной таблице и результат отсортируйте (ORDER BY) с ограничением вывода по количеству строк (LIMIT)

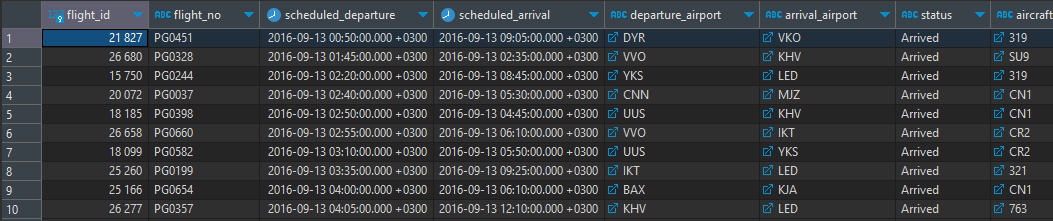
**select** \* **from** bookings.flights f

**where** (scheduled\_departure < '2016-10-01 00:30:00.000 +0000'::**timestamp**)

**and** (scheduled\_departure > '2016-08-31 00:00:00.000 +0300'::**timestamp**)

**order** **by** scheduled\_departure

**limit** 10



1. Используя SQL язык напишите OLAP-запрос к произвольной связке таблиц (в рамках JOIN оператора), используя оператор GROUP BY и любые агрегатные функции count, min, max, sum.

**select**

**case**

**when** res.total\_amount < 10000 **then** 'менее 10 тысяч'

**when** res.total\_amount < 20000 **and** res.total\_amount > 10000 **then** 'от 10 до 20 тысяч'

**when** res.total\_amount < 50000 **and** res.total\_amount > 20000 **then** 'от 20 до 50 тысяч'

**when** res.total\_amount > 50000 **then** 'более 50 тысяч'

**else** 'не определено'

**end** **as** цена\_билета,

**count**(res.passenger\_name) **as** "количество пассажиров"

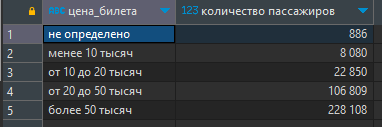
**from**

(**select** t.passenger\_name, b.total\_amount **from** bookings.tickets t

**join** bookings.bookings b **on** t.book\_ref = b.book\_ref) **as** res

**group** **by** цена\_билета

**order** **by** "количество пассажиров"

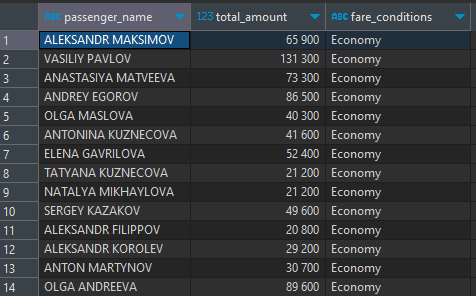


1. Используя SQL язык примените JOIN операторы (INNER, LEFT, RIGHT) для более чем двух таблиц из модели данных.

**select** t.passenger\_name, b.total\_amount, tf.fare\_conditions **from** bookings.tickets t

**join** bookings.bookings b **on** t.book\_ref = b.book\_ref

**join** bookings.ticket\_flights tf **on** tf.ticket\_no = t.ticket\_no

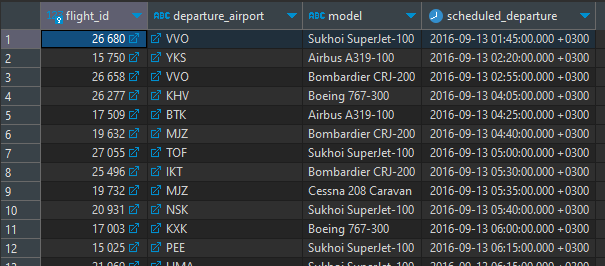


**select** **distinct** tf.flight\_id, f.departure\_airport, a.model, f.scheduled\_departure **from** bookings.flights f

**right** **join** bookings.aircrafts a **on** a.aircraft\_code = f.aircraft\_code

**right** **join** bookings.ticket\_flights tf **on** tf.flight\_id = f.flight\_id

**order** **by** f.scheduled\_departure



1. Создайте виртуальную таблицу VIEW с произвольным именем для SQL запроса из задания 2.

**create** **view** flights\_10m\_2016y **as**

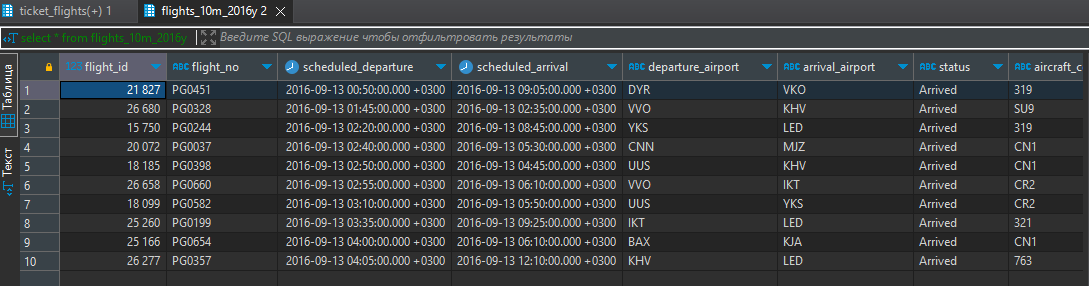
**select** \* **from** bookings.flights f

**where** (scheduled\_departure < '2016-10-01 00:30:00.000 +0000'::**timestamp**)

**and** (scheduled\_departure > '2016-08-31 00:00:00.000 +0300'::**timestamp**)

**order** **by** scheduled\_departure

**limit** 10



1. Используя язык программирования Python, создайте скрипт для вывода данных на экран (в консоль или IDE) из задания 5.

import numpy as np

import pandas as pd

import psycopg2

db\_host= 'localhost'

db\_port= 5432

db\_name= 'demo'

db\_user= 'postgres'

db\_password= '123'

if \_\_name\_\_ == '\_\_main\_\_':

try:

connection = psycopg2.connect(host=db\_host,port=db\_port,database=db\_name,user=db\_user,password=db\_password)

cursor = connection.cursor()

cursor.execute("SELECT \* FROM flights\_10m\_2016y")

result = cursor.fetchall()

print(result)

cursor.close()

connection.close()

except psycopg2.Error as e:(

print("Error connecting to PostgreSQL:", e))

df = pd.DataFrame(result)

