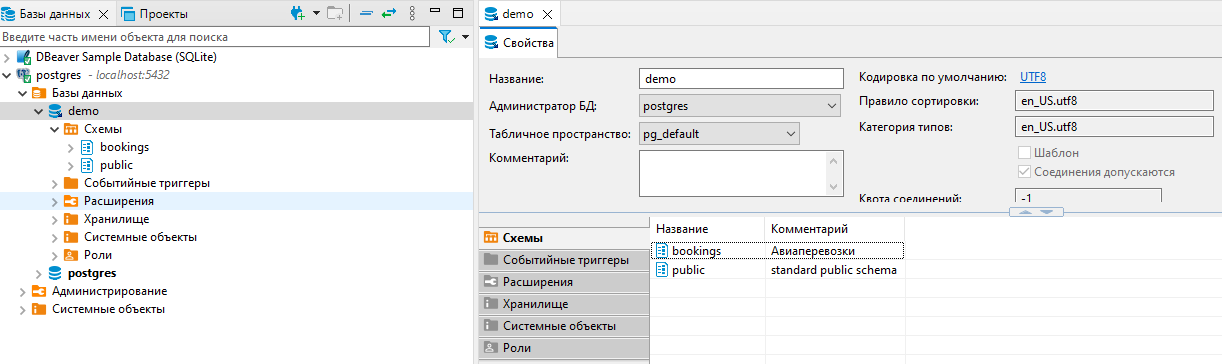
**Промежуточная аттестация №2**

1. **Форма контроля:** проверка выполненных заданий на образовательной платформе в текстовых файлах
2. **Диагностические инструменты:**
   1. Описание задания: скачайте и разверните [модель данных](https://drive.google.com/file/d/1Qtms55MSY7hAR-OnAJQzIEVcmkYWRUBq/view) на вашем установленной БД PostreeSQL. Модель данных развернется в базу данных demo.
   2. Используя SQL язык и произвольные две таблицы из модели данных необходимо объединить их различными способами (UNION, JOIN)
   3. Используя SQL язык напишите запрос с любым фильтром WHERE к произвольной таблице и результат отсортируйте (ORDERBY) с ограничением вывода по количеству строк (LIMIT)
   4. 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.
   5. Используя SQL язык примените JOIN операторы (INNER, LEFT, RIGHT) для более чем двух таблиц из модели данных.
   6. Создайте виртуальную таблицу VIEW с произвольным именем для SQL запроса из задания 2
   7. Используя язык программирования Python, создайте Python-скрипт для вывода данных на экран (в консоль или IDE ) из задания 5
3. **Разворачиваем модель данных в БД postgres, в развернутый в Docker контейнер my-postgres, используя команду docker exec -i my-postgres psql postgres postgres < air.sql.**



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

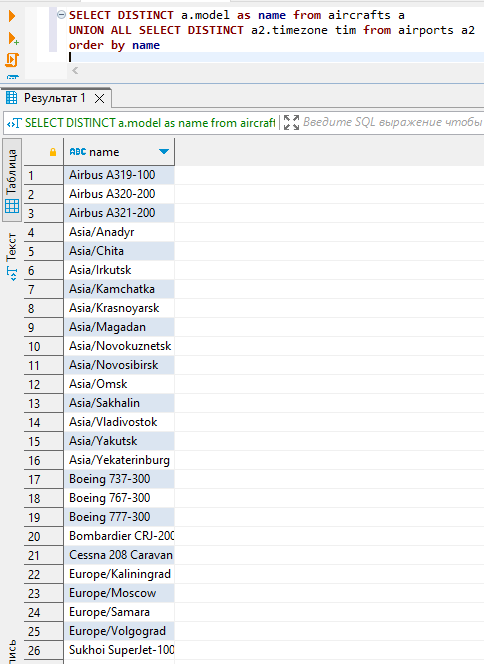
**Решение:**

Выводим в одном столбце уникальные значения из таблиц aircrafts (столбец model) и airports (столбец city), упорядоченные по возрастанию.

**SELECT** **DISTINCT** a.model **as** **name** **from** aircrafts a

**UNION** **ALL** **SELECT** **DISTINCT** a2.city tim **from** airports a2

**order** **by** **name**



Выводим номера рейсов и соответствующие им модели самолетов, вылетающие из аэропорта Домодедово, упорядоченные по убыванию номера рейса.

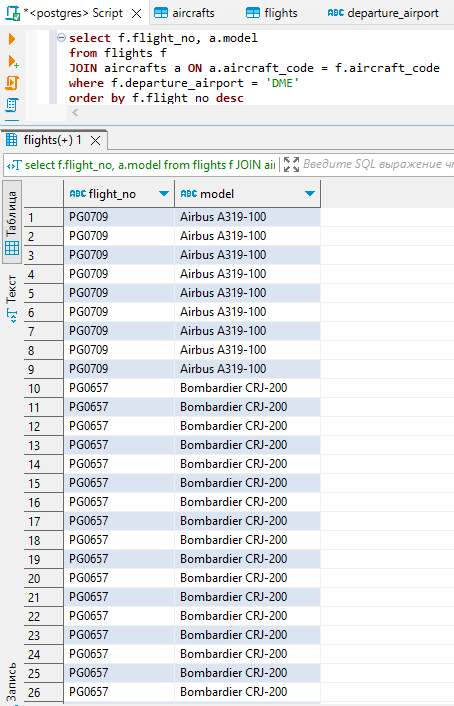
**select** f.flight\_no, a.model

**from** flights f

**JOIN** aircrafts a **ON** a.aircraft\_code = f.aircraft\_code

**where** f.departure\_airport = 'DME'

**order** **by** f.flight\_no **desc**



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

**Решение:**

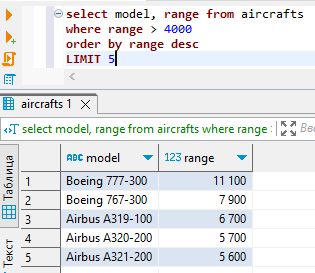
Выводим 5 самолётов с максимальной дальность полета более 4000 км, упорядоченные по убыванию дальности полёта.

**select** model, **range** **from** aircrafts

**where** **range** > 4000

**order** **by** **range** **desc**

**LIMIT** 5



* 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.**

**Решение:**

Выводим модели самолётов с соответствующим им количеством билетов, максимально й и минимальной ценой билета, упорядоченные по возрастанию модели самолёта.

**select** a.model **as** "Модель самолета", **count**(tf.ticket\_no) **as** "количество билетов",

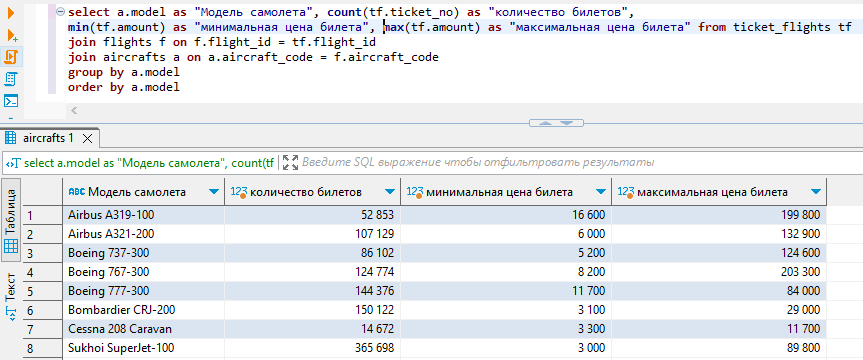
**min**(tf.amount) **as** "минимальная цена билета", **max**(tf.amount) **as** "максимальная цена билета" **from** ticket\_flights tf

**join** flights f **on** f.flight\_id = tf.flight\_id

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

**group** **by** a.model

**order** **by** a.model



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

**Решение:**

**select** \* **from** aircrafts a

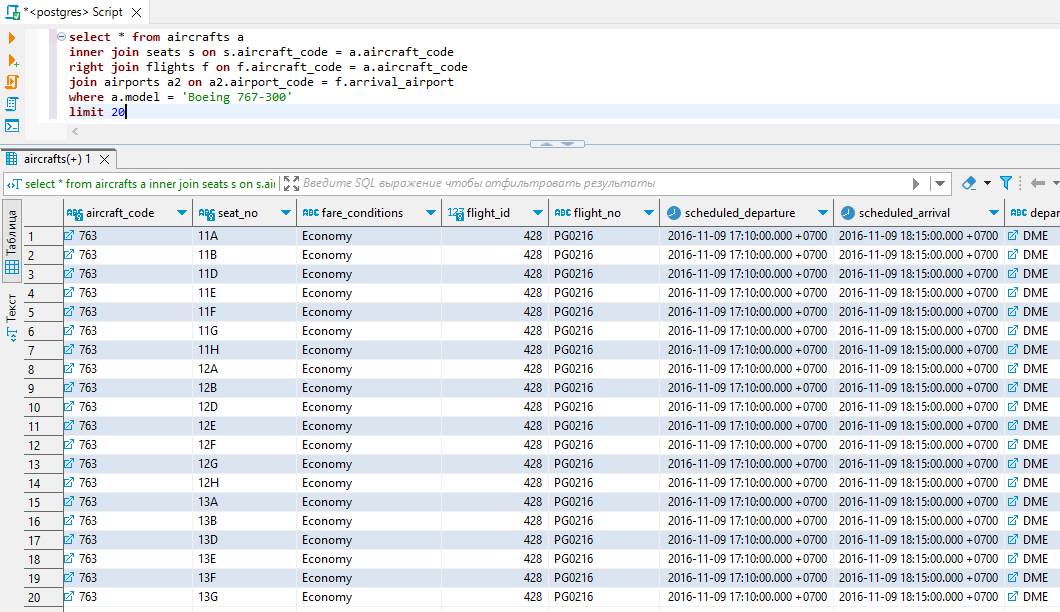
**inner** **join** seats s **on** s.aircraft\_code = a.aircraft\_code

**right** **join** flights f **on** f.aircraft\_code = a.aircraft\_code

**join** airports a2 **on** a2.airport\_code = f.arrival\_airport

**where** a.model = 'Boeing 767-300'

**limit** 20



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

**Решение:**

**CREATE** **OR** **REPLACE** **VIEW** bookings.aircrafts\_with\_max\_range **AS**

**SELECT** aircrafts.model,

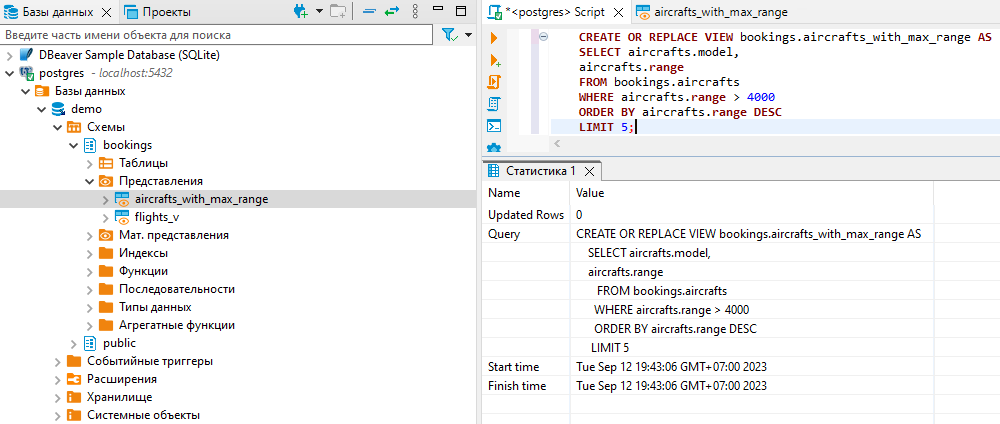
aircrafts.**range**

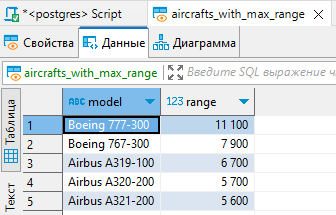
**FROM** bookings.aircrafts

**WHERE** aircrafts.**range** > 4000

**ORDER** **BY** aircrafts.**range** **DESC**

**LIMIT** 5;





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

**Решение:**

import sqlalchemy as sql  
import psycopg2  
# Database configuration  
db\_host = 'localhost' # Replace with your Docker container IP or hostname  
db\_port = 5432 # Port you mapped to the container (5432 by default)  
db\_name = 'demo' # Replace with your actual database name  
db\_user = 'postgres' # PostgreSQL username  
db\_password = 'mysecretpassword' # PostgreSQL password  
if \_\_name\_\_ == '\_\_main\_\_':  
 try:  
 # Connect to the database  
 connection = psycopg2.connect(host=db\_host, port=db\_port, database=db\_name, user=db\_user, password=db\_password)  
 cursor = connection.cursor()  
 # Execute a sample query  
 cursor.execute("select \* from demo.bookings.aircrafts\_with\_max\_range")  
 result = cursor.fetchall()  
 print(result)  
 # Close the cursor and connection  
 cursor.close()  
 connection.close()  
 except psycopg2.Error as e:  
 print("Error connecting to PostgreSQL:", e)

