**Промежуточная аттестация 2**

1. **Форма контроля:** проверка выполненных заданий на образовательной платформе в текстовых файлах.
2. **Диагностические инструменты:**

Описание задания:

1. Скачайте и разверните [модель](https://drive.google.com/file/d/1Qtms55MSY7hAR-OnAJQzIEVcmkYWRUBq/view?usp=share_link) [данных](https://drive.google.com/file/d/1Qtms55MSY7hAR-OnAJQzIEVcmkYWRUBq/view?usp=share_link) на вашем установленной БД PostgreSQL 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)

**select**

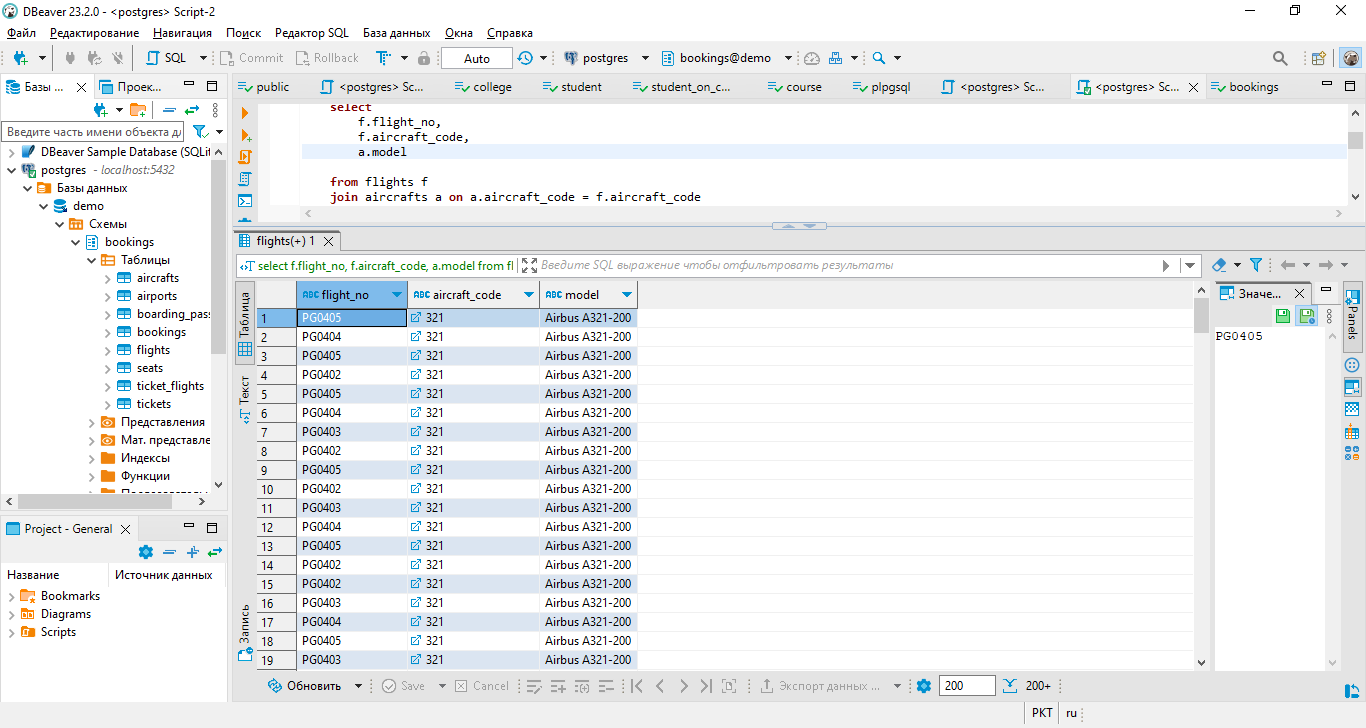
f.flight\_no,

f.aircraft\_code,

a.model

**from** flights f

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

**with** find\_model **as** (

**select**

f.flight\_no,

f.aircraft\_code,

a.model

**from** flights f

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

)

**SELECT** \*

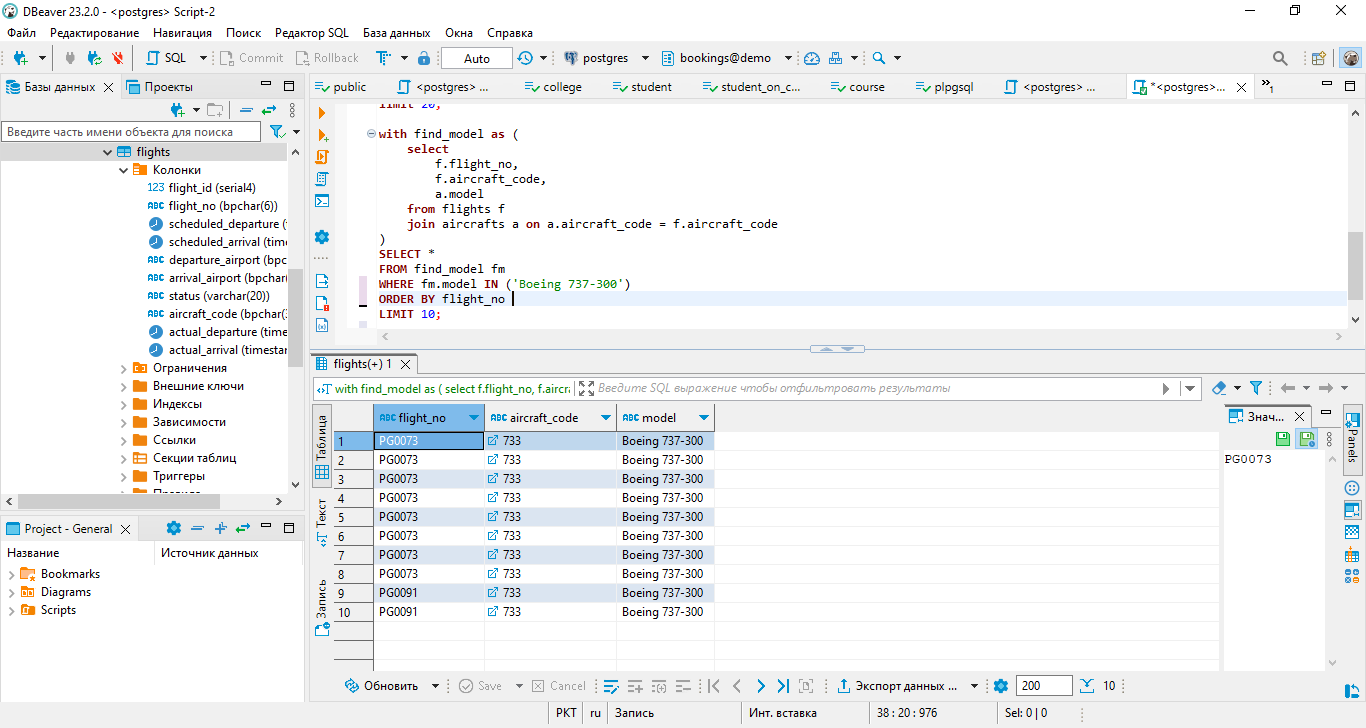
**FROM** find\_model fm

**WHERE** fm.model **IN** ('Boeing 737-300')

**ORDER** **BY** flight\_no

**ASC**

**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.

**with** find\_model **as** (

**select**

f.flight\_no,

f.aircraft\_code,

a.model

**from** flights f

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

)

**select** fv.flight\_no **as** "Номер рейса",

fm.model **as** "Тип самолета",

fv.scheduled\_duration **as** "Длительность полета",

fv.departure\_city **as** "Город отправления",

fv.departure\_airport **as** "Аэропорт",

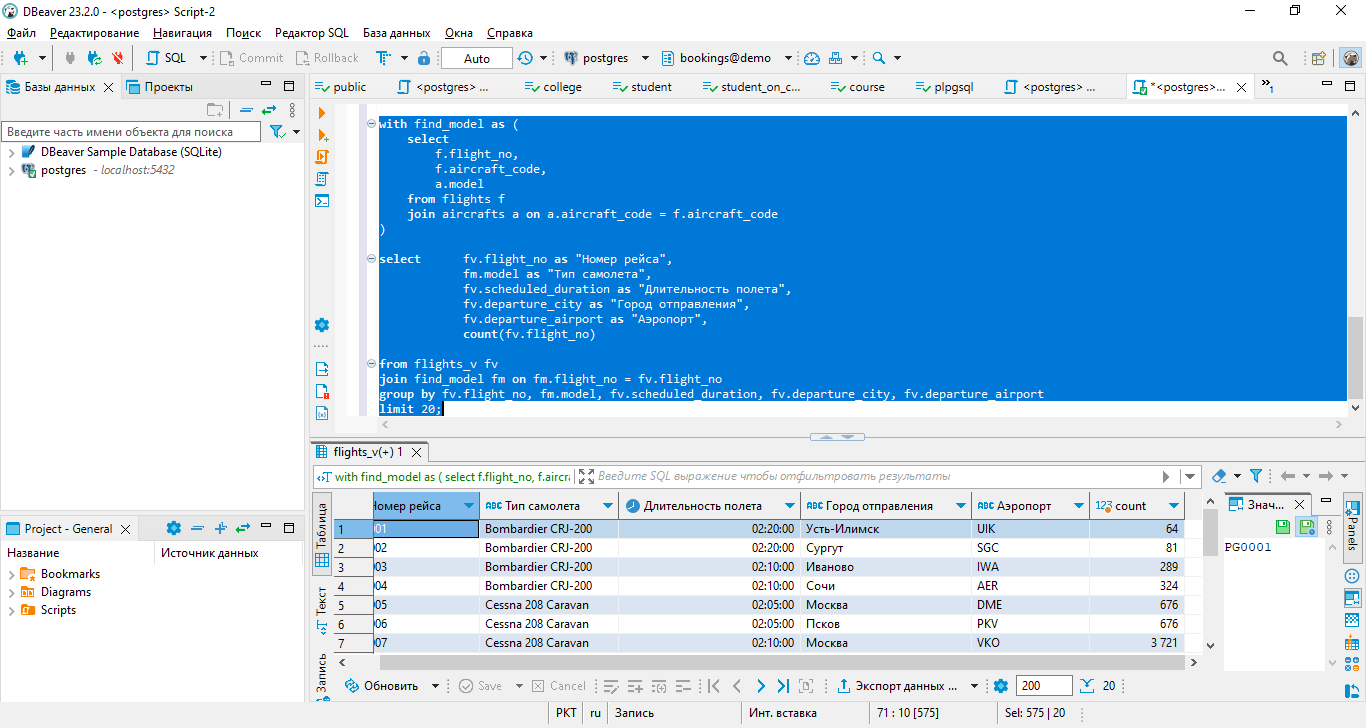
**count**(fv.flight\_no)

**from** flights\_v fv

**join** find\_model fm **on** fm.flight\_no = fv.flight\_no

**group** **by** fv.flight\_no, fm.model, fv.scheduled\_duration, fv.departure\_city, fv.departure\_airport

**limit** 10;



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

**with** find\_model **as** (

**select**

f.flight\_id,

f.flight\_no,

f.aircraft\_code,

a.model

**from** flights f

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

)

**SELECT** \*

**FROM** find\_model fm

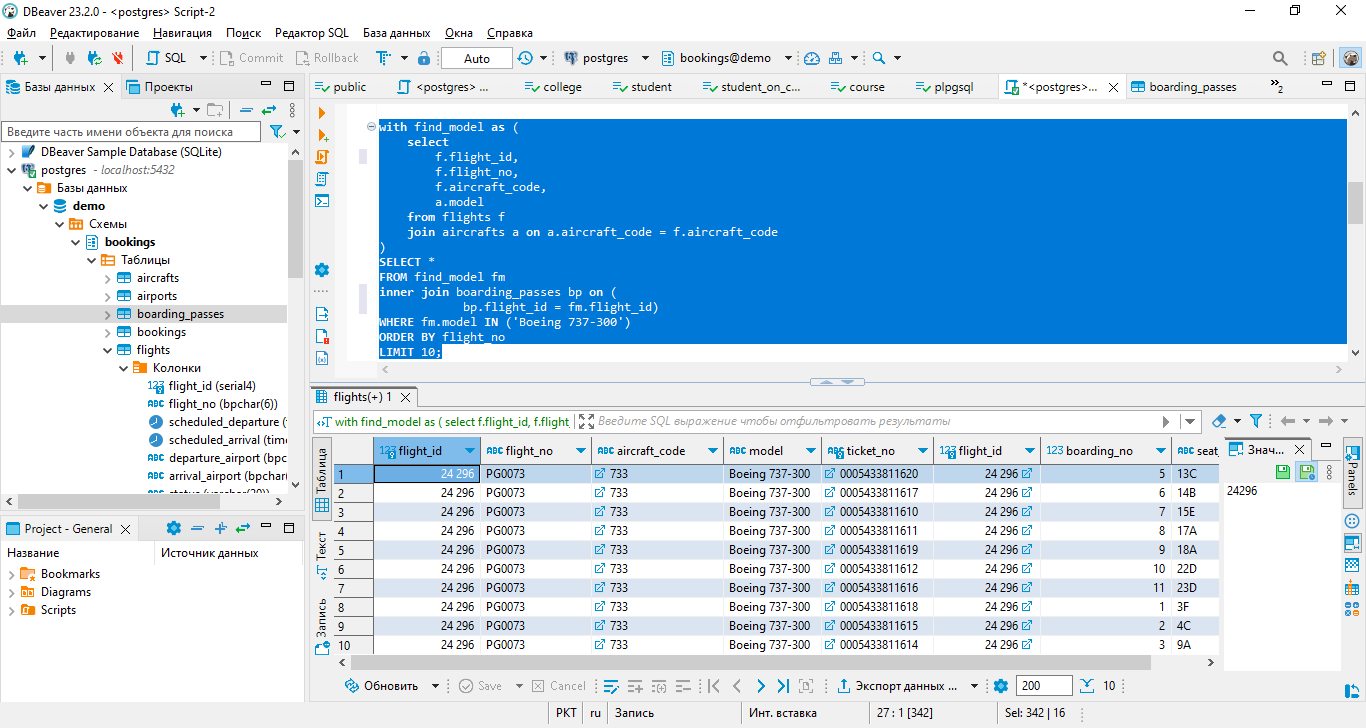
**inner** **join** boarding\_passes bp **on** (

bp.flight\_id = fm.flight\_id)

**WHERE** fm.model **IN** ('Boeing 737-300')

**ORDER** **BY** flight\_no

**LIMIT** 10;



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

**CREATE** **VIEW** my\_view1 **AS**

**select**

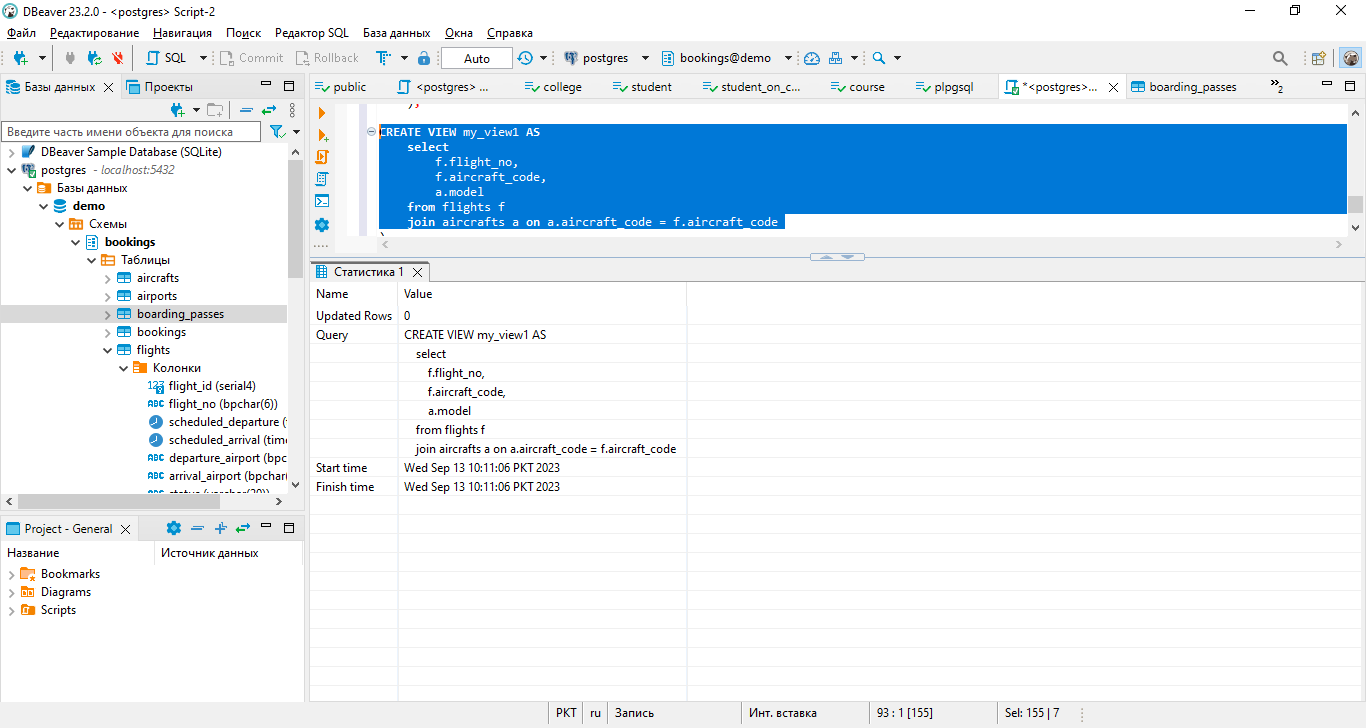
f.flight\_no,

f.aircraft\_code,

a.model

**from** flights f

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



1. используя язык программирования Python, создайте Python-скрипт для вывода данных на экран (в консоль или IDE) из задания 5)
2. import psycopg2  
   from psycopg2 import Error, sql  
     
   DATABASE = "demo"  
   USER = "postgres"  
   PASSWORD = "Salavat\_2014"  
   LOCALHOST = "127.0.0.1"  
   PORT = "5432"  
     
   SET\_PATH = '''SET search\_path = bookings;'''  
     
   READ\_DEMO\_BOARDING\_INFO = '''  
   select  
    'Без посадочного талона' as "Статус брони",  
    COUNT(b.book\_ref) AS "Количество броней"  
   FROM bookings b  
   JOIN tickets t ON t.book\_ref = b.book\_ref  
   LEFT JOIN boarding\_passes bp ON bp.ticket\_no = t.ticket\_no  
   WHERE bp.boarding\_no IS null;  
   '''  
     
   READ\_DEMO\_BOARDING\_INFO\_FULL = '''  
   with find\_model as (  
    select   
    f.flight\_no,   
    f.aircraft\_code,   
    a.model  
    from flights f  
    join aircrafts a on a.aircraft\_code = f.aircraft\_code  
   )  
     
   select fv.flight\_no as "Номер рейса",  
    fm.model as "Тип самолета",  
    fv.scheduled\_departure::date as "Плановая дата",  
    fv.scheduled\_departure::time as "Плановое время",  
    fv.scheduled\_arrival::date as "Плановая дата вылета",  
    fv.scheduled\_arrival::time as "Плановое время вылета",  
    fv.scheduled\_duration as "Длительность полета",   
    fv.departure\_city as "Город отправления",  
    fv.departure\_airport as "Аэропорт"  
     
   from flights\_v fv  
   join find\_model fm on fm.flight\_no = fv.flight\_no  
   limit 20;  
   '''  
     
   try:  
    connection = psycopg2.connect(dbname=DATABASE, user=USER, password=PASSWORD, host=LOCALHOST, port=PORT)  
    curs = connection.cursor()  
     
    print("Информация по подключению")  
    print(connection.get\_dsn\_parameters())  
     
    curs.execute(SET\_PATH)  
    curs.execute(READ\_DEMO\_BOARDING\_INFO\_FULL)  
     
    record = curs.fetchone()   
     
    print(f"Текущая запись {record}") # что выгрузилось  
     
     
   except(Exception, Error) as error:  
    print("Возникло исключение при работе с Postgres", error)

**Форма работы:** индивидуальная **Набор технологий:**

* Редактор кода
* PostgreSQL сервер
* Клиент PostgreSQL

1. **Показатели и критерии оценивания:**

Показатели и критерии оценивания проектов в области искусственного интеллекта включают в себя производительность, надежность, эффективность, скорость выполнения и простоту использования. На основании этих показателей можно оценивать качество и эффективность реализуемых проектов и принимать решения о их дальнейшем развитии.

Управление проектами в IT-индустрии в области искусственного интеллекта требует использования специальных форм контроля, диагностических инструментов, набора технологий и показателей и критериев оценивания.

1. **Шкала оценивания**

**Оценки:** зачтено/не зачтено

* + зачтено заслуживает работа, отвечающая требованиям. Работа может содержать незначительные ошибки. Отвечает всем всем критериям оценивания.
  + не зачтено заслуживает работа, в которой присутствуют низкие показатели по критериям оценивания.