## Промежуточная аттестация 2

## Форма контроля:

проверка выполненных заданий на образовательной платформе в текстовых файлах.

## Диагностические инструменты:

Описание задания:

1. Скачайте и разверните модель данных 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)
3. Используя SQL язык напишите запрос с любым фильтром WHERE к произвольной таблице и результат отсортируйте (ORDER BY) 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

**Форма работы:** индивидуальная

**Набор технологий:**

* Редактор кода
* PostgreSQL сервер
* Клиент PostgreSQL

## 3) 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.

## 4) Шкала оценивания

**Оценки**: зачтено/не зачтено

* зачтено заслуживает работа, отвечающая требованиям. Работа может содержать незначительные ошибки. Отвечает всем всем критериям

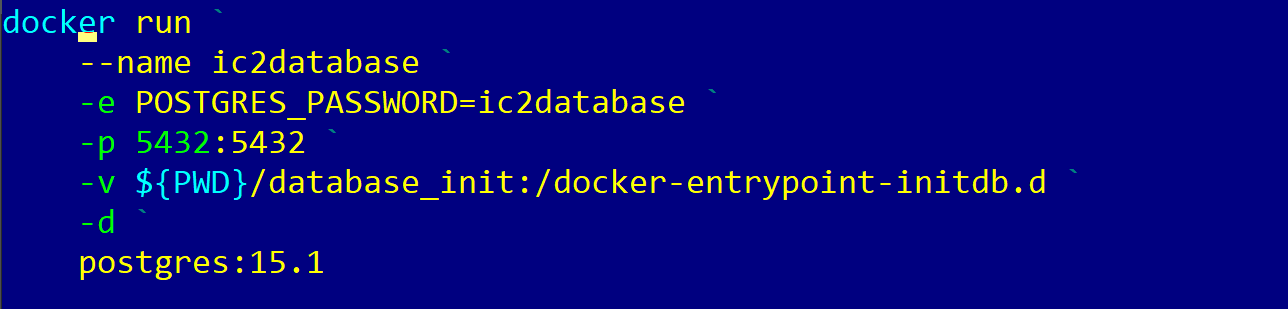
оценивания.

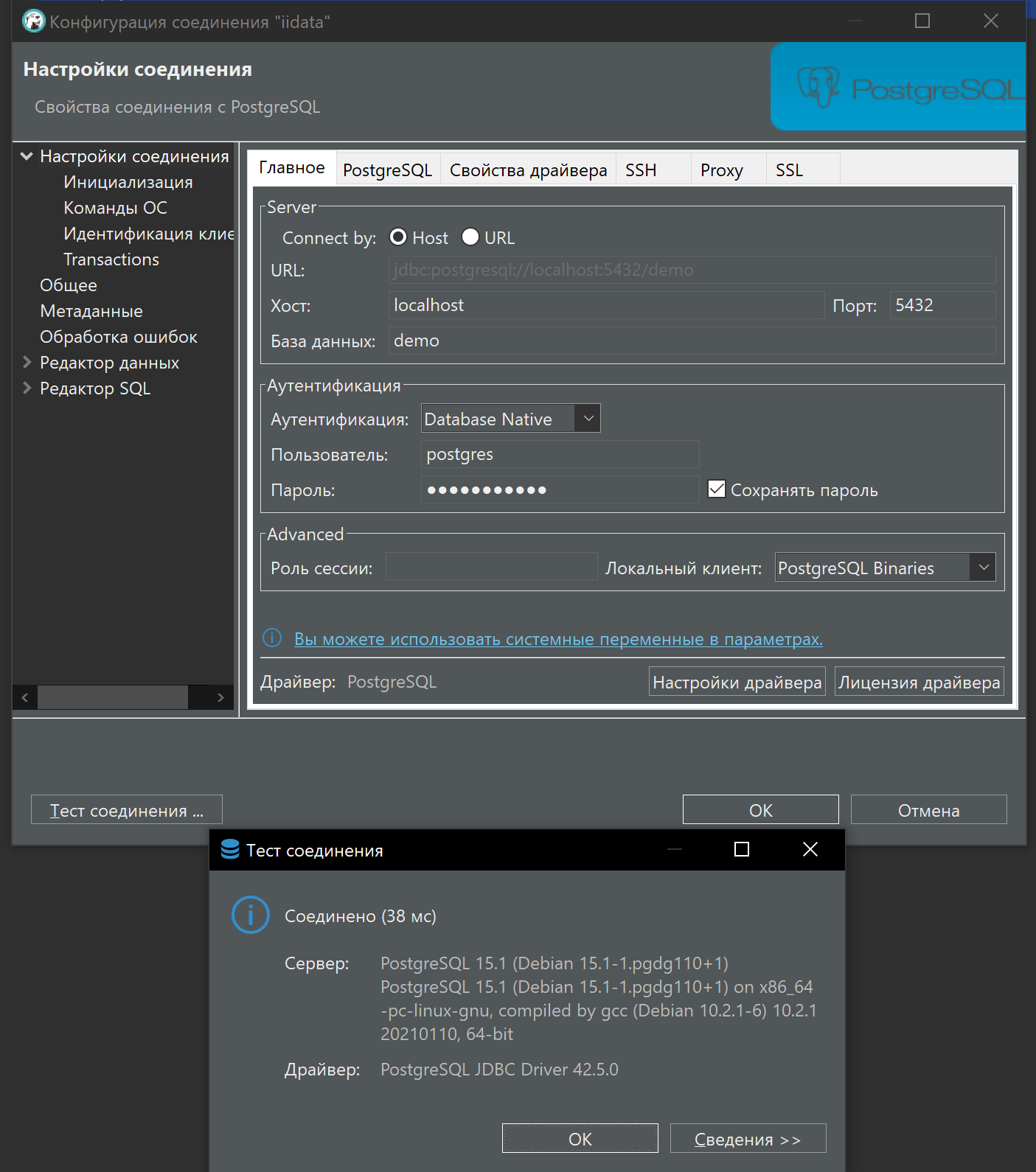
* не зачтено заслуживает работа, в которой присутствуют низкие показатели по критериям оценивания.

## Решение:

[Ссылка на проект.](https://github.com/keeper0null/hwII/tree/main/interimCertifications/ic2)

--0) Скачайте и разверните модель данных на вашем установленной БД PostgreSQL. Модель данных развернется в базу данных demo.





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

**select** a.airport\_name

,f.flight\_no

**from** bookings.airports a

**join** bookings.flights f **on**

f.departure\_airport = a.airport\_code

**union** **all**

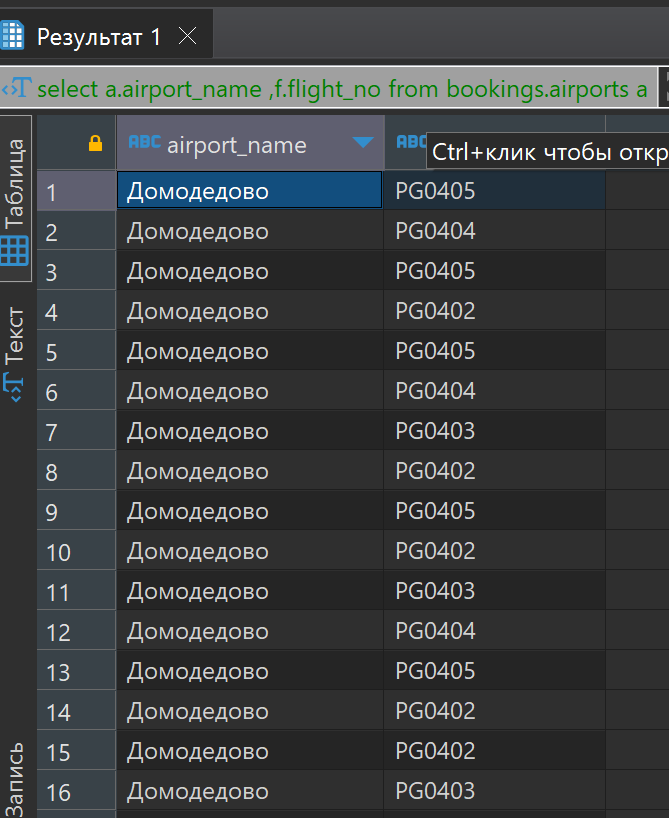
**select** a.airport\_name

,f.flight\_no

**from** bookings.airports a

**join** bookings.flights f **on**

f.arrival\_airport = a.airport\_code



--2) 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** f.flight\_id

,f.flight\_no

,f.scheduled\_departure

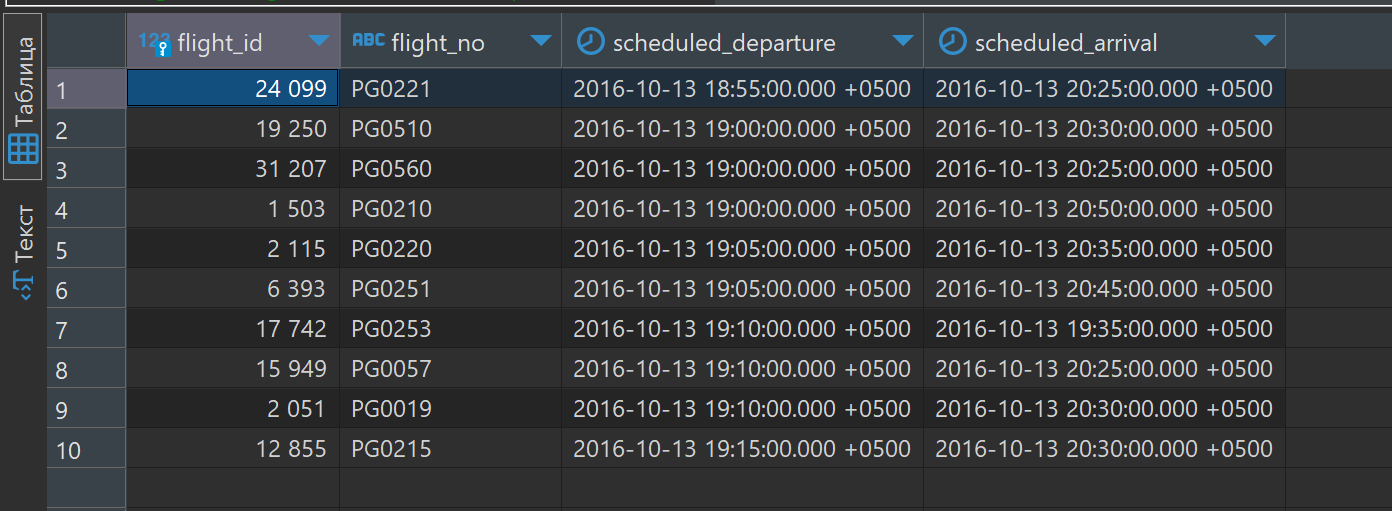
,f.scheduled\_arrival

**from** bookings.flights f

**where** f.status = 'On Time'

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

**limit** 10



--3) 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** a.city

,a.airport\_name

,**count**(**distinct** f.flight\_id) "Количество рейсов"

,**count**(**distinct** tf.ticket\_no) "Количество проданных билетов"

,**min**(tf.amount) "Минимальная цена билета"

,**max**(tf.amount) "Максимальаня цена билета"

,**sum**(tf.amount) "Вырученная сумма"

**from** bookings.flights f

**join** bookings.airports a **on**

a.airport\_code **in** (f.departure\_airport, f.arrival\_airport)

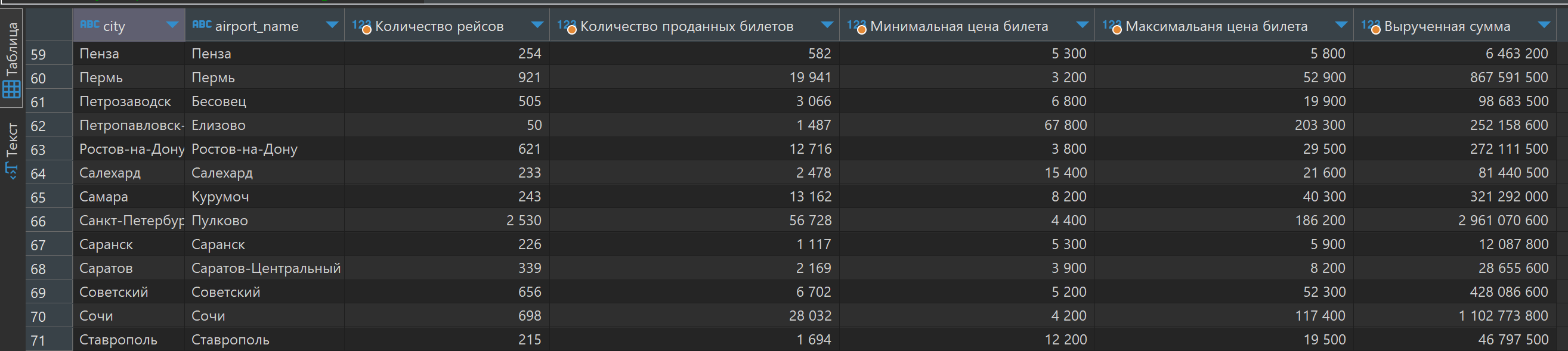
**join** bookings.ticket\_flights tf **on**

tf.flight\_id = f.flight\_id

**group** **by**

a.city

,a.airport\_name



--4) Используя SQL язык примените JOIN операторы (INNER, LEFT, RIGHT) для более чем двух таблиц из модели данных.

**select** ac.model

,s.seat\_no

,s.fare\_conditions

**from** bookings.aircrafts ac

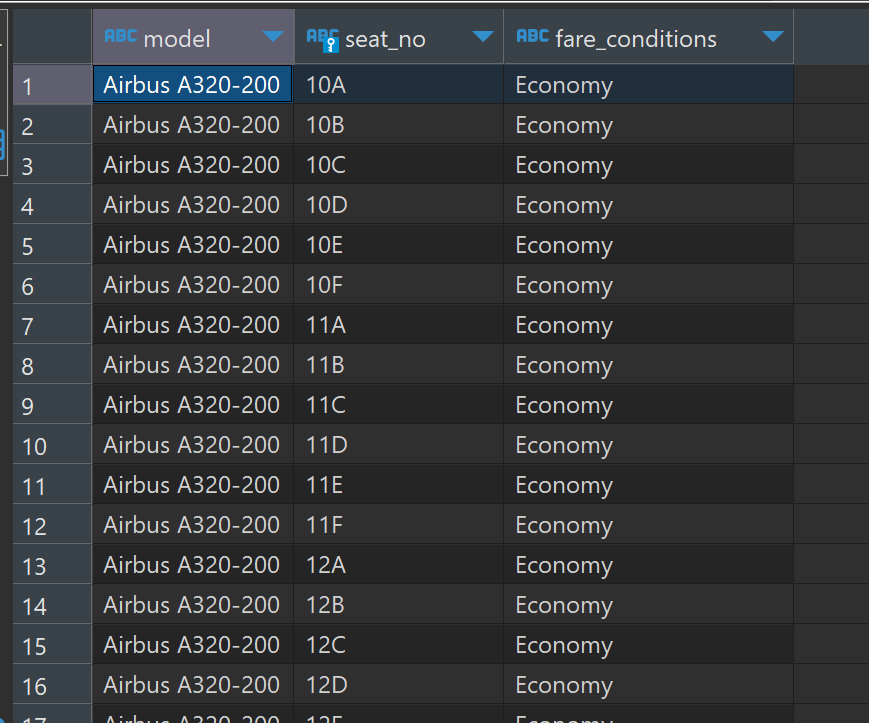
**inner** **join** bookings.seats s **on**

s.aircraft\_code = ac.aircraft\_code

**left** **join** bookings.flights f **on**

f.aircraft\_code = ac.aircraft\_code

**where** f.flight\_id **is** **null** -- места не использованные на рейсах



--5) Создайте виртуальную таблицу VIEW с произвольным именем для SQL запроса из задания 2

**create** **or** **replace** **view** v\_on\_time\_board

**as**

**select** f.flight\_id

,f.flight\_no

,f.scheduled\_departure

,f.scheduled\_arrival

**from** bookings.flights f

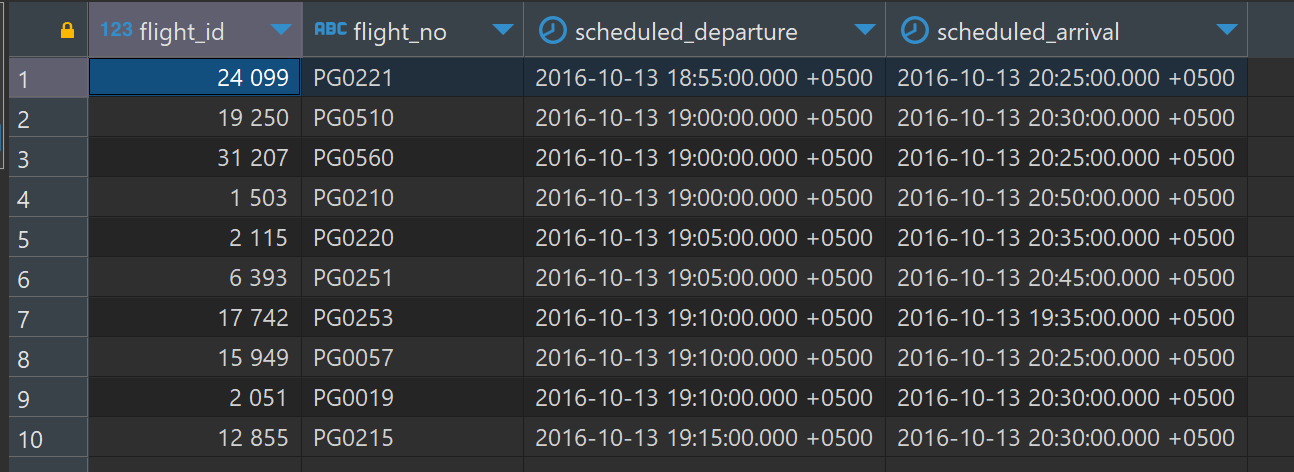
**where** f.status = 'On Time'

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

**limit** 10

**select** \*

**from** v\_on\_time\_board



--6) используя язык программирования Python, создайте Python-скрипт для вывода данных на экран (в консоль или IDE) из задания 5

import psycopg2

try:

    conn = psycopg2.connect("dbname='demo' user='postgres' host='localhost' password='ic2database'")

except:

    print("Нет подключения к БД")

with conn.cursor() as curs:

    try:

        curs.execute("""

select f.flight\_id

      ,f.flight\_no

      ,f.scheduled\_departure

      ,f.scheduled\_arrival

  from v\_on\_time\_board f""")

        many\_rows = curs.fetchmany(11)

        print(\*many\_rows, sep = "\n")

    except (Exception, psycopg2.DatabaseError) as error:

        print(error)

