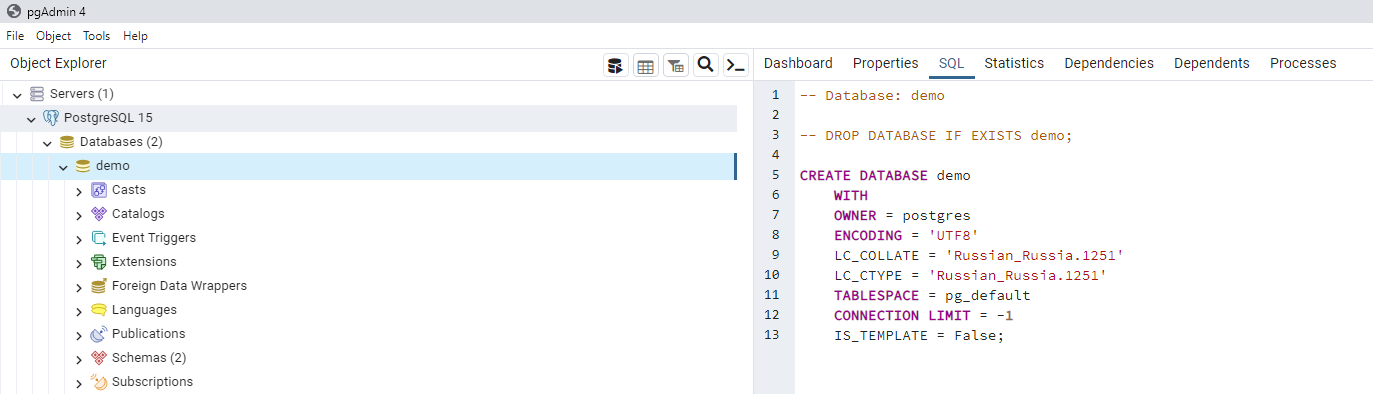


1. Развернем базу данных в БД PostgreSQL. Модель развернулась в базу данных demo 
2. Используя SQL язык и произвольные две таблицы из модели данных, объединим их различными способами

выбрать все места, предусмотренные компоновкой салона самолета Cessna 208 Caravan.SELECT a.aircraft\_code, a.model, s.seat\_no, s.fare\_conditions

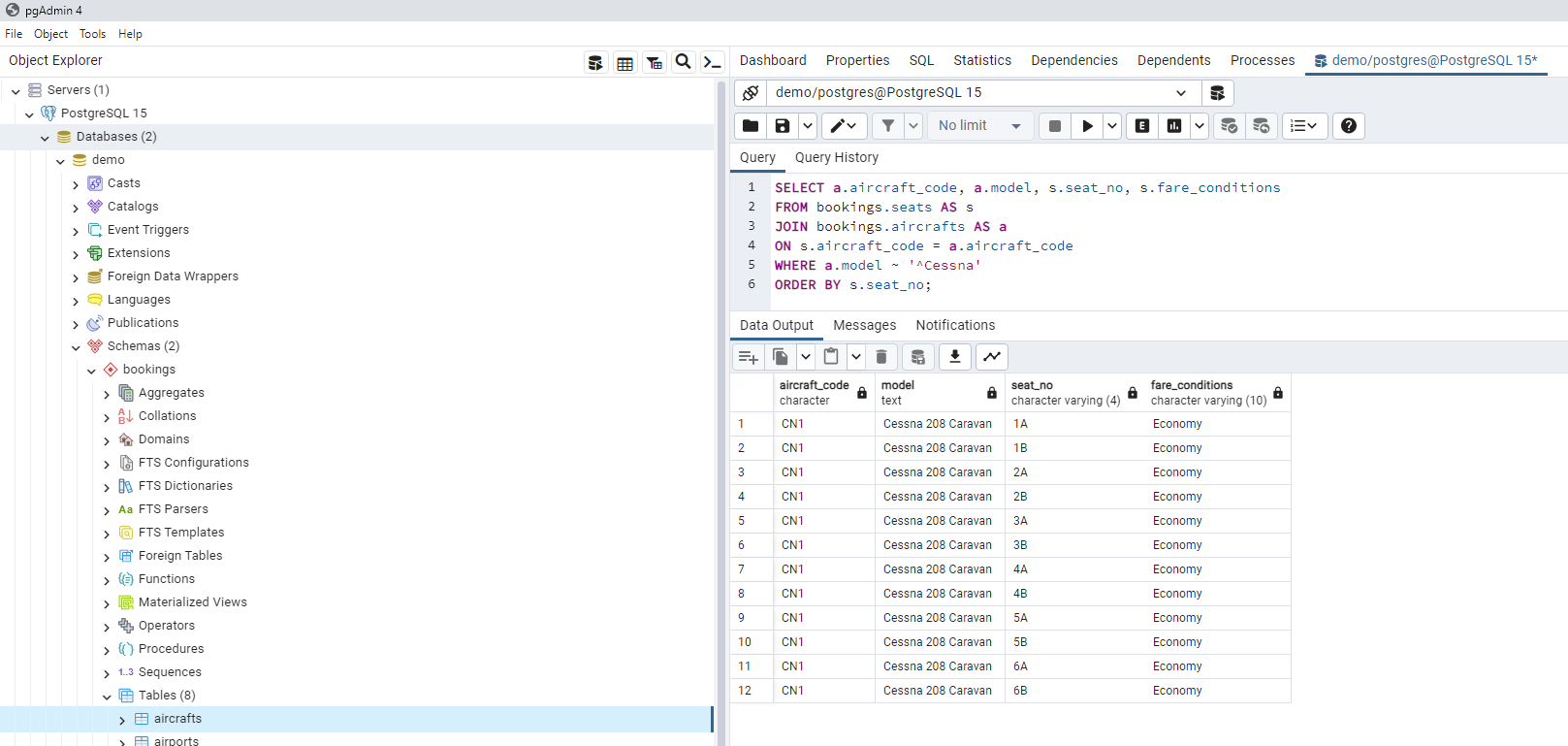
FROM bookings.seats AS s

JOIN bookings.aircrafts AS a

ON s.aircraft\_code = a.aircraft\_code

WHERE a.model ~ '^Cessna'

ORDER BY s.seat\_no;



в какие города можно улететь либо из Казани, либо из Сочи?

SELECT arrival\_city FROM routes

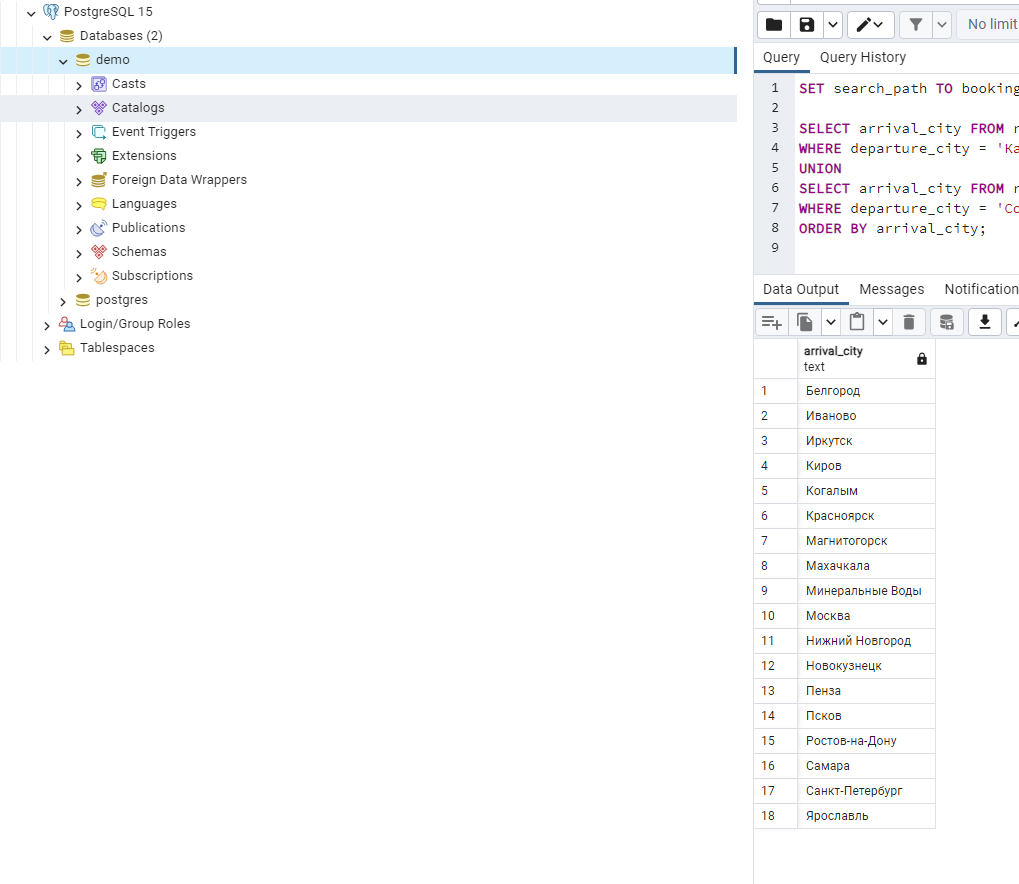
WHERE departure\_city = 'Казань'

UNION

SELECT arrival\_city FROM routes

WHERE departure\_city = 'Сочи'

ORDER BY arrival\_city;



SELECT a.aircraft\_code, a.model,

left( a.model, strpos( a.model, ' ' ) - 1 ) AS company,

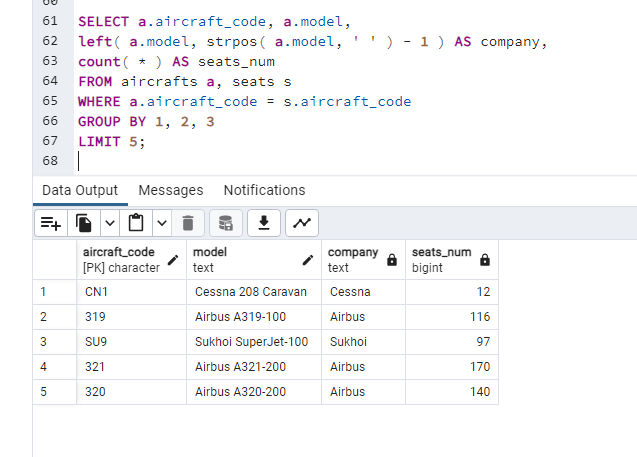
count( \* ) AS seats\_num

FROM aircrafts a, seats s

WHERE a.aircraft\_code = s.aircraft\_code

GROUP BY 1, 2, 3

LIMIT 5;



1. Найти распределение количества бронирований по диапазонам сумм с шагом в 50 тысяч рублей. Максимальная сумма в одном бронировании составляет 800000 рублей.

SELECT r.min\_sum, r.max\_sum, count( b.\* )

FROM bookings b

RIGHT OUTER JOIN

( VALUES ( 0, 50000 ), ( 50000, 100000 ),

( 100000, 150000 ), ( 150000, 200000 ),

( 200000, 250000 ), ( 250000, 300000 ),

( 300000, 350000 ), ( 350000, 400000 ),

( 400000, 450000 ), ( 450000, 500000 ),

( 500000, 550000 ), ( 600000, 650000 ),

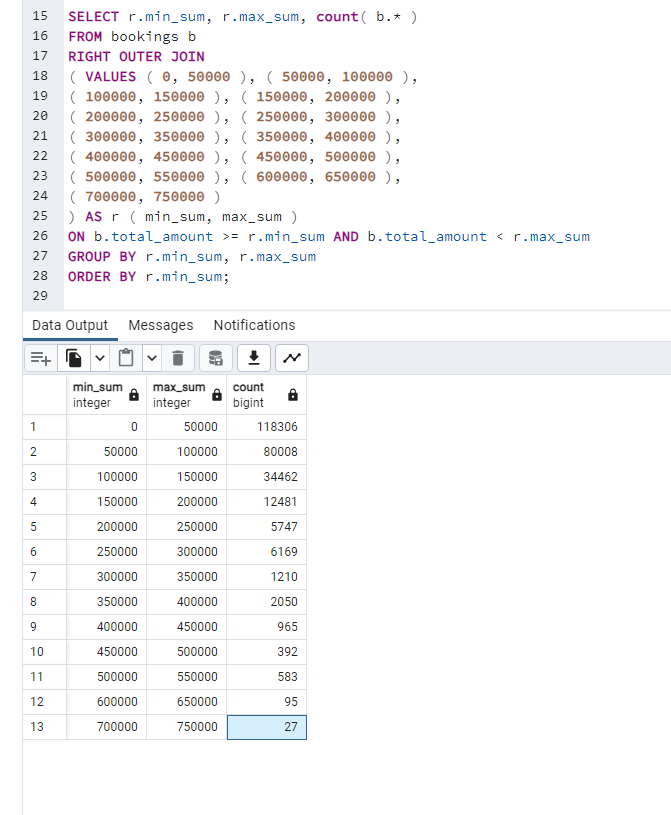
( 700000, 750000 )

) AS r ( min\_sum, max\_sum )

ON b.total\_amount >= r.min\_sum AND b.total\_amount < r.max\_sum

GROUP BY r.min\_sum, r.max\_sum

ORDER BY r.min\_sum;



1. отсортировать места в порядке их расположения в салоне самолета и вывести также адреса электронной почты пассажиров

SELECT s.seat\_no, p.passenger\_name, p.email

FROM seats s

LEFT OUTER JOIN (

SELECT t.passenger\_name, b.seat\_no,

t.contact\_data->'email' AS email

FROM (

ticket\_flights tf

JOIN tickets t ON tf.ticket\_no = t.ticket\_no

)

JOIN boarding\_passes b

ON tf.ticket\_no = b.ticket\_no

AND tf.flight\_id = b.flight\_id

WHERE tf.flight\_id = 27584

) AS p

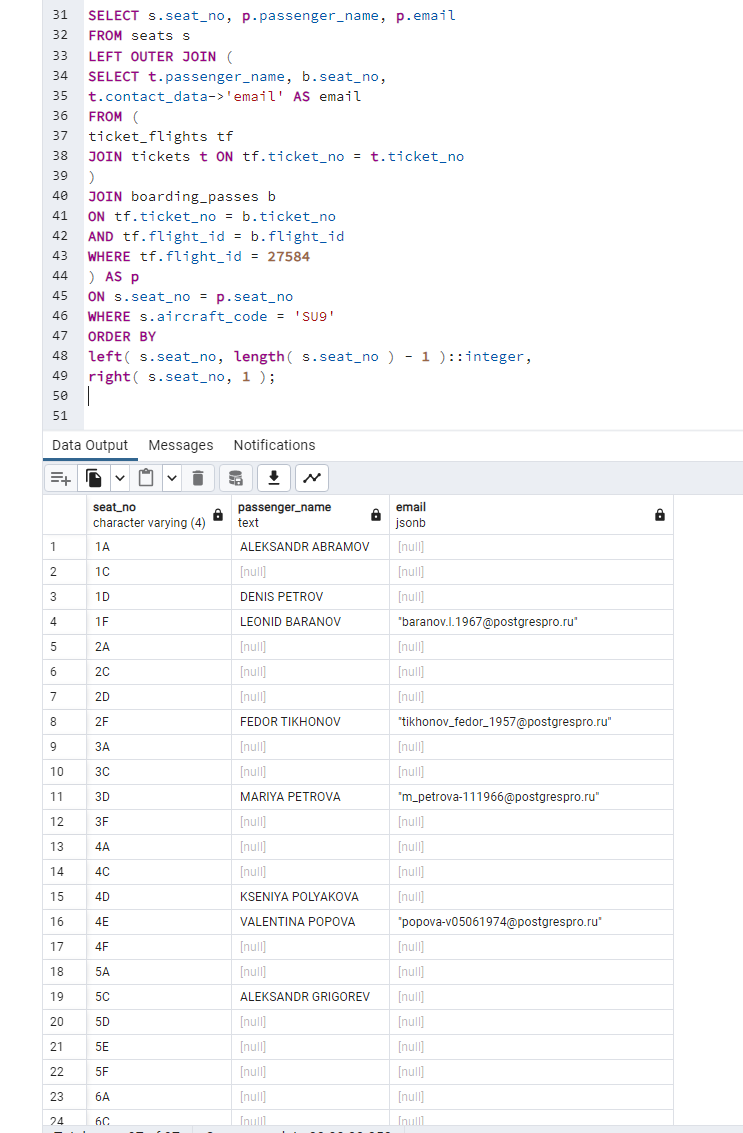
ON s.seat\_no = p.seat\_no

WHERE s.aircraft\_code = 'SU9'

ORDER BY

left( s.seat\_no, length( s.seat\_no ) - 1 )::integer,

right( s.seat\_no, 1 );



1. CREATE VIEW aircrafts\_seats AS

( SELECT a.aircraft\_code, a.model,

left( a.model, strpos( a.model, ' ' ) - 1 ) AS company,

count( \* ) AS seats\_num

FROM aircrafts a, seats s

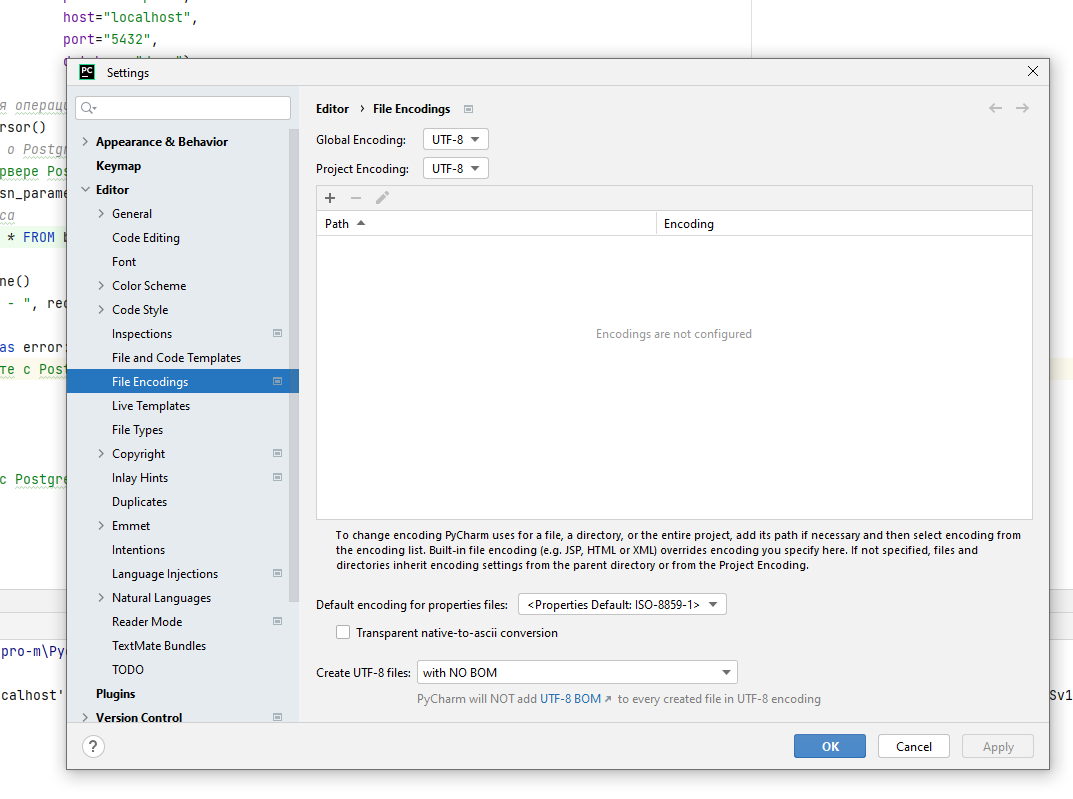
WHERE a.aircraft\_code = s.aircraft\_code

GROUP BY 1, 2, 3

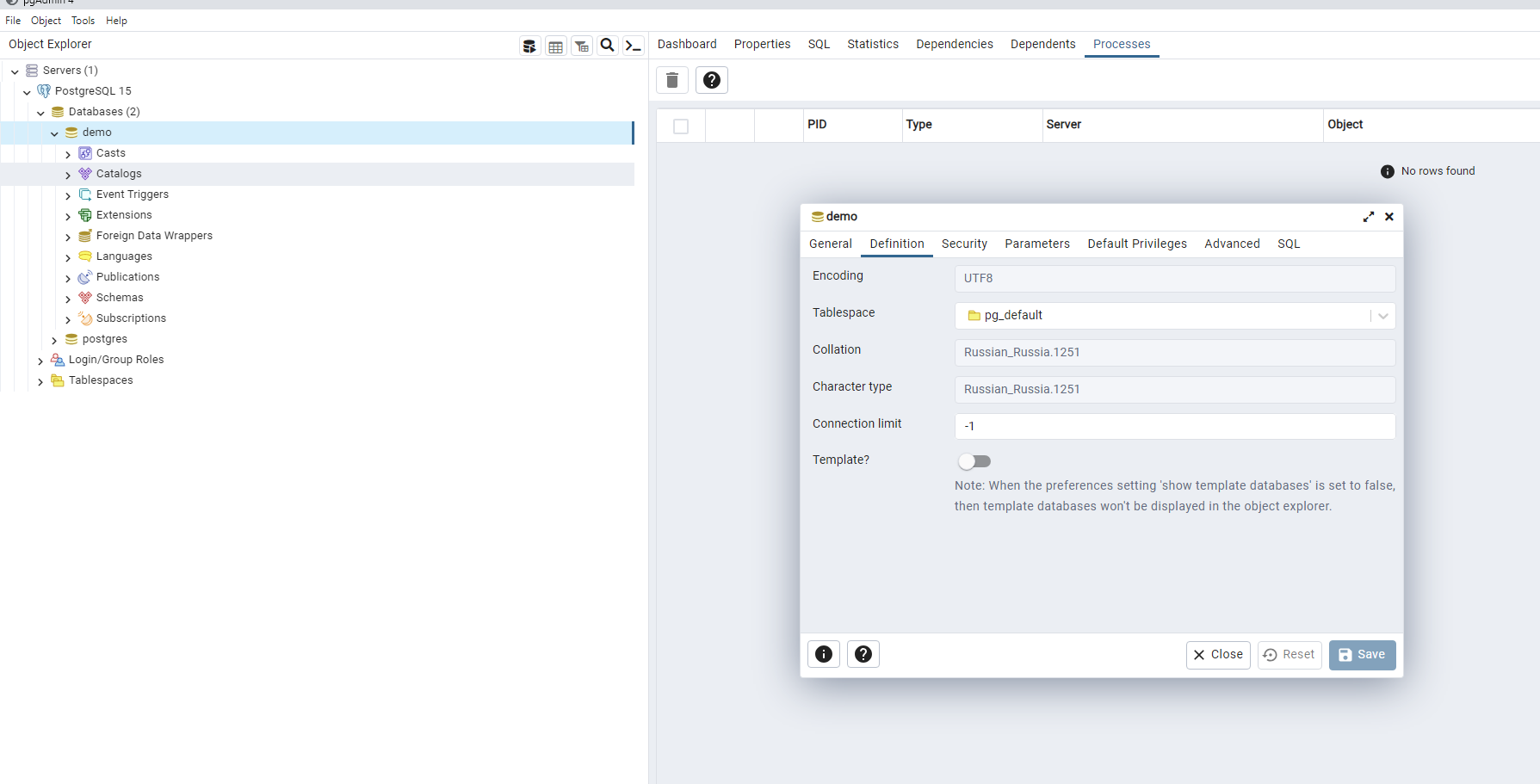
LIMIT 5

);

1. Для выполнения этого задания необходимо было поменять кодировку в настройках PyCharm, а именно

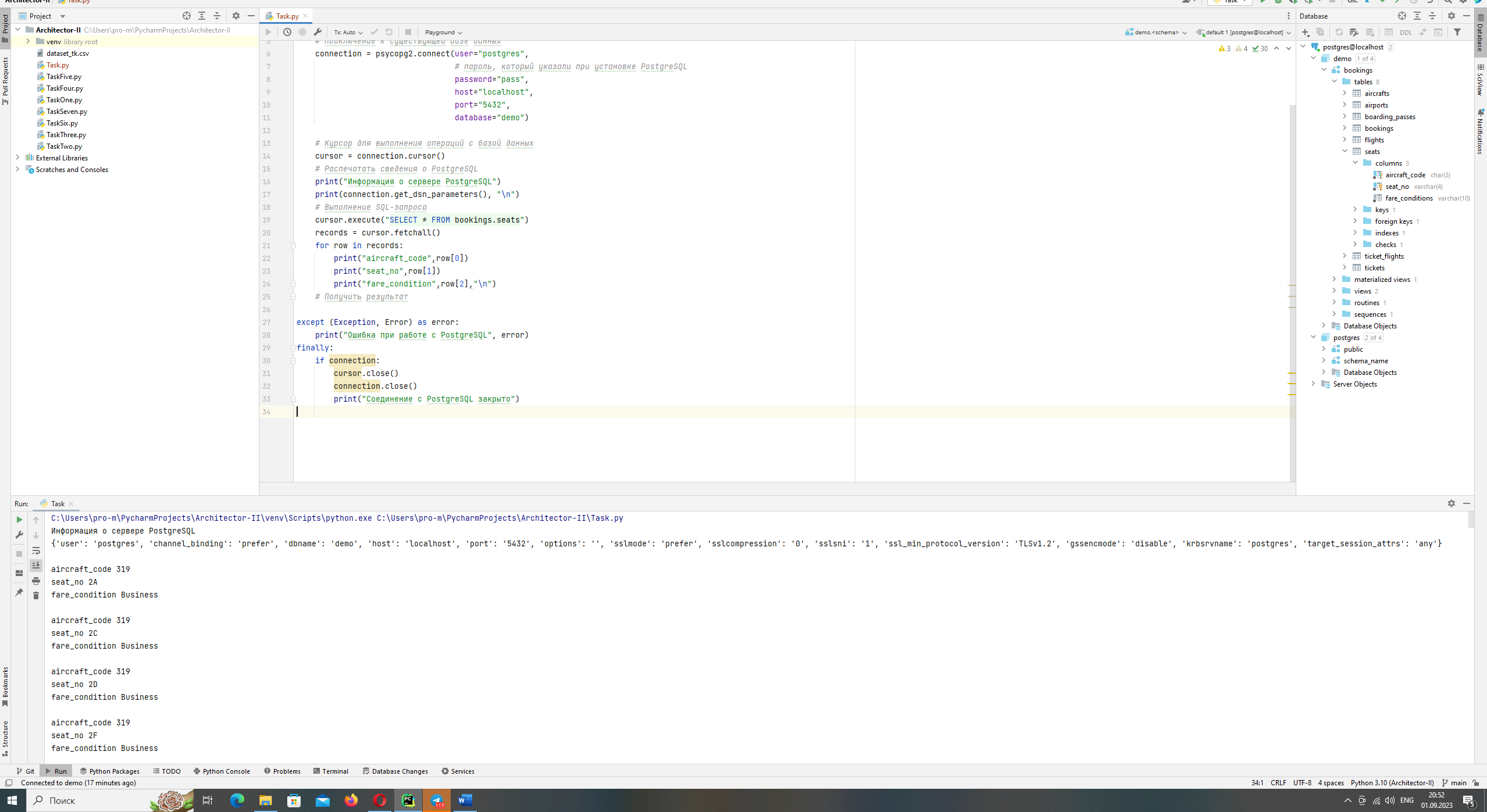


и проверить, какая в PgAdmin

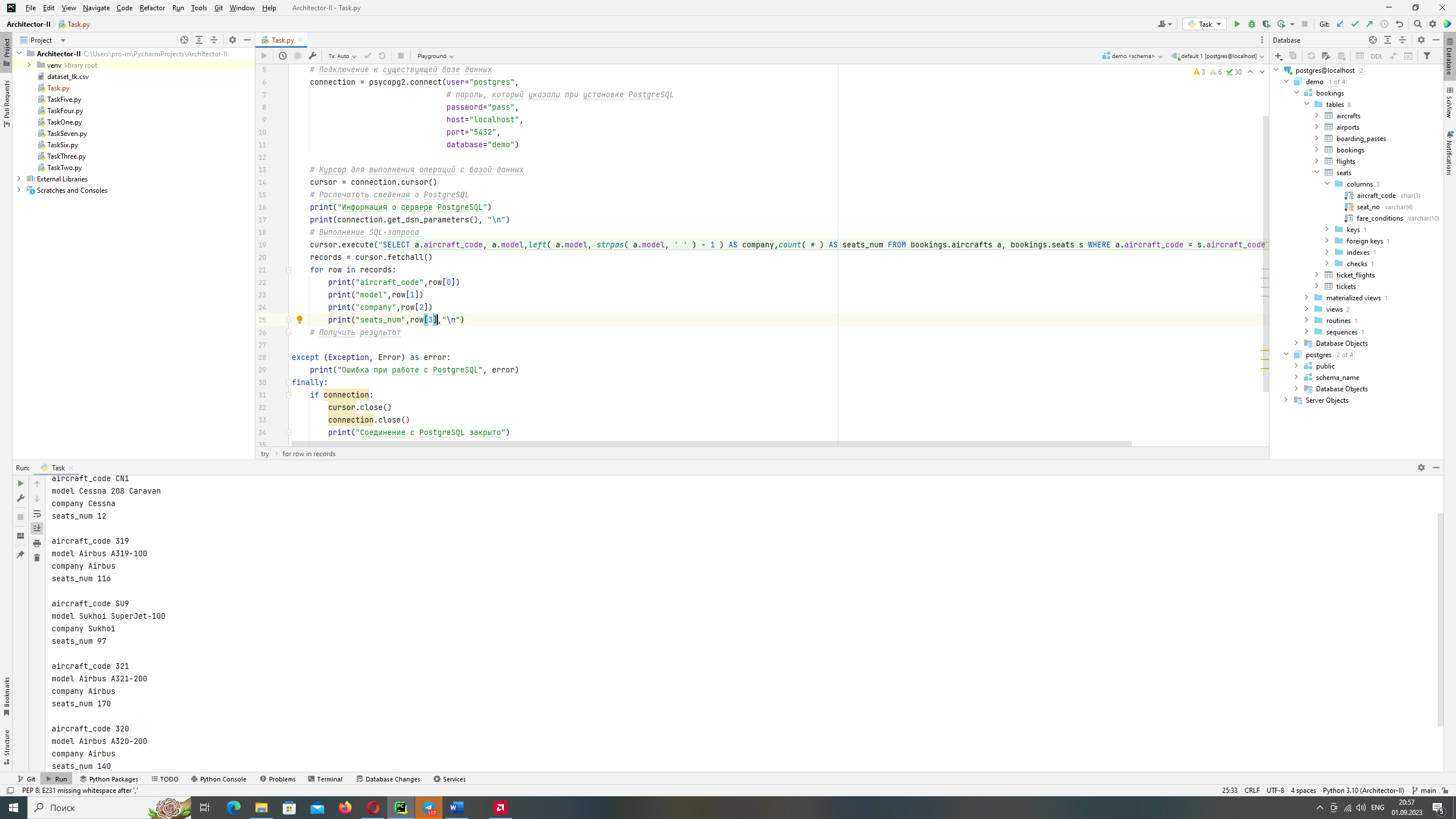


Отлично, подготовительный этап выполнили.

Установим pusorc2, напишем небольшой скрипт, чтобы проверить, работают ли запросы.



Отлично, запрос отработал. Теперь запустим view из 4 задания



Отлично, все так же, как и в pgAdmin

SET search\_path TO bookings;

SELECT arrival\_city FROM routes

WHERE departure\_city = 'Казань'

UNION

SELECT arrival\_city FROM routes

WHERE departure\_city = 'Сочи'

ORDER BY arrival\_city;

SELECT airport\_name, city, longitude

FROM airports

ORDER BY longitude DESC

LIMIT 5;

SELECT r.min\_sum, r.max\_sum, count( b.\* )

FROM bookings b

RIGHT OUTER JOIN

( VALUES ( 0, 50000 ), ( 50000, 100000 ),

( 100000, 150000 ), ( 150000, 200000 ),

( 200000, 250000 ), ( 250000, 300000 ),

( 300000, 350000 ), ( 350000, 400000 ),

( 400000, 450000 ), ( 450000, 500000 ),

( 500000, 550000 ), ( 600000, 650000 ),

( 700000, 750000 )

) AS r ( min\_sum, max\_sum )

ON b.total\_amount >= r.min\_sum AND b.total\_amount < r.max\_sum

GROUP BY r.min\_sum, r.max\_sum

ORDER BY r.min\_sum;

SELECT s.seat\_no, p.passenger\_name, p.email

FROM seats s

LEFT OUTER JOIN (

SELECT t.passenger\_name, b.seat\_no,

t.contact\_data->'email' AS email

FROM (

ticket\_flights tf

JOIN tickets t ON tf.ticket\_no = t.ticket\_no

)

JOIN boarding\_passes b

ON tf.ticket\_no = b.ticket\_no

AND tf.flight\_id = b.flight\_id

WHERE tf.flight\_id = 27584

) AS p

ON s.seat\_no = p.seat\_no

WHERE s.aircraft\_code = 'SU9'

ORDER BY

left( s.seat\_no, length( s.seat\_no ) - 1 )::integer,

right( s.seat\_no, 1 );

CREATE VIEW aircrafts\_seats AS

( SELECT a.aircraft\_code, a.model,

left( a.model, strpos( a.model, ' ' ) - 1 ) AS company,

count( \* ) AS seats\_num

FROM aircrafts a, seats s

WHERE a.aircraft\_code = s.aircraft\_code

GROUP BY 1, 2, 3

);

SELECT a.aircraft\_code, a.model,

left( a.model, strpos( a.model, ' ' ) - 1 ) AS company,

count( \* ) AS seats\_num

FROM aircrafts a, seats s

WHERE a.aircraft\_code = s.aircraft\_code

GROUP BY 1, 2, 3

LIMIT 5;

import psycopg2  
from psycopg2 import Error  
  
try:  
 *# Подключение к существующей базе данных* connection = psycopg2.connect(user="postgres",  
 *# пароль, который указали при установке PostgreSQL* password="pass",  
 host="localhost",  
 port="5432",  
 database="demo")  
  
 *# Курсор для выполнения операций с базой данных* cursor = connection.cursor()  
 *# Распечатать сведения о PostgreSQL* print("Информация о сервере PostgreSQL")  
 print(connection.get\_dsn\_parameters(), "\n")  
 *# Выполнение SQL-запроса* cursor.execute("SELECT a.aircraft\_code, a.model,left( a.model, strpos( a.model, ' ' ) - 1 ) AS company,count( \* ) AS seats\_num FROM bookings.aircrafts a, bookings.seats s WHERE a.aircraft\_code = s.aircraft\_code GROUP BY 1, 2, 3 LIMIT 5 ")  
 records = cursor.fetchall()  
 for row in records:  
 print("aircraft\_code",row[0])  
 print("model",row[1])  
 print("company",row[2])  
 print("seats\_num",row[3],"\n")  
 *# Получить результат*except (Exception, Error) as error:  
 print("Ошибка при работе с PostgreSQL", error)  
finally:  
 if connection:  
 cursor.close()  
 connection.close()  
 print("Соединение с PostgreSQL закрыто")