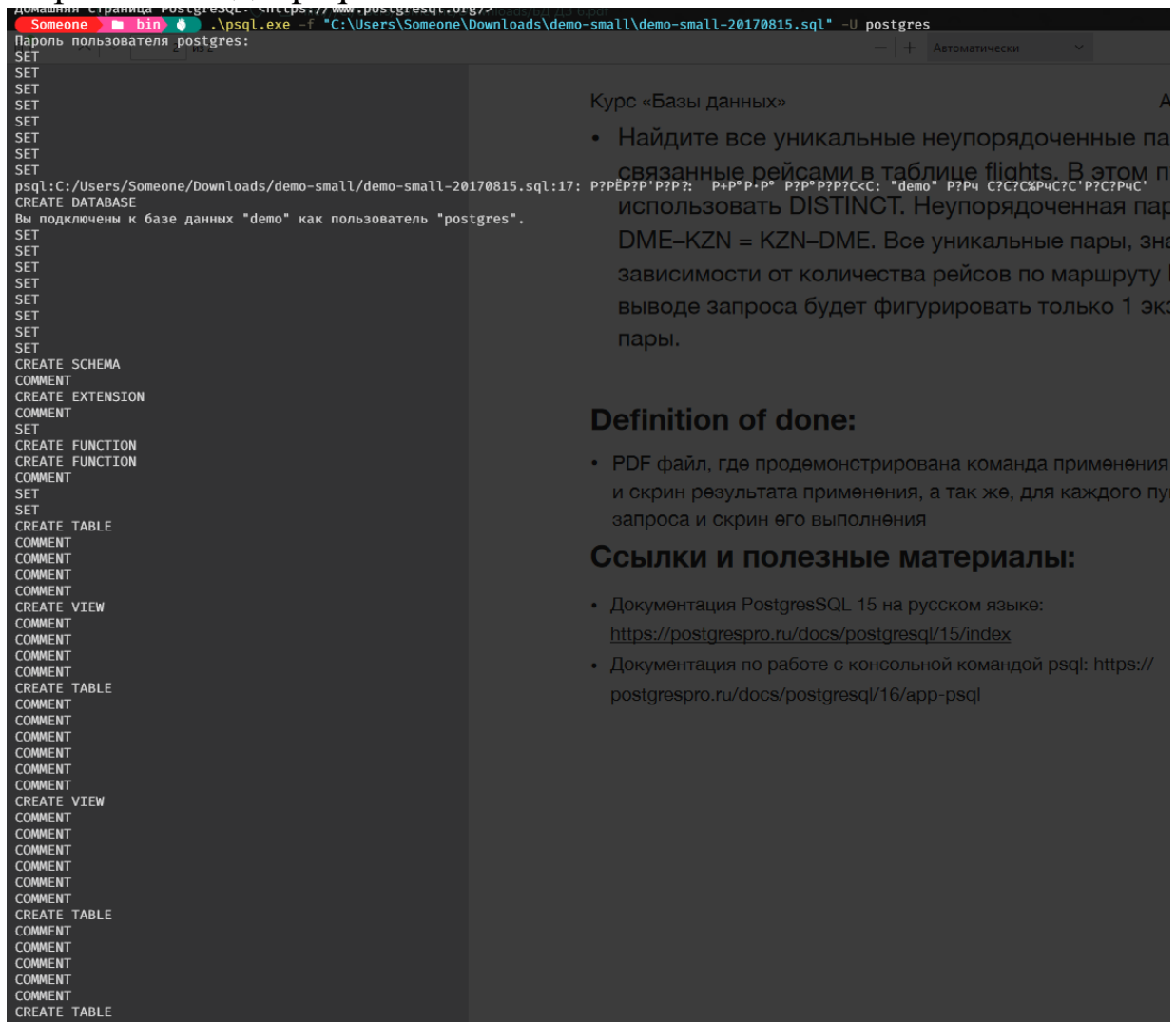
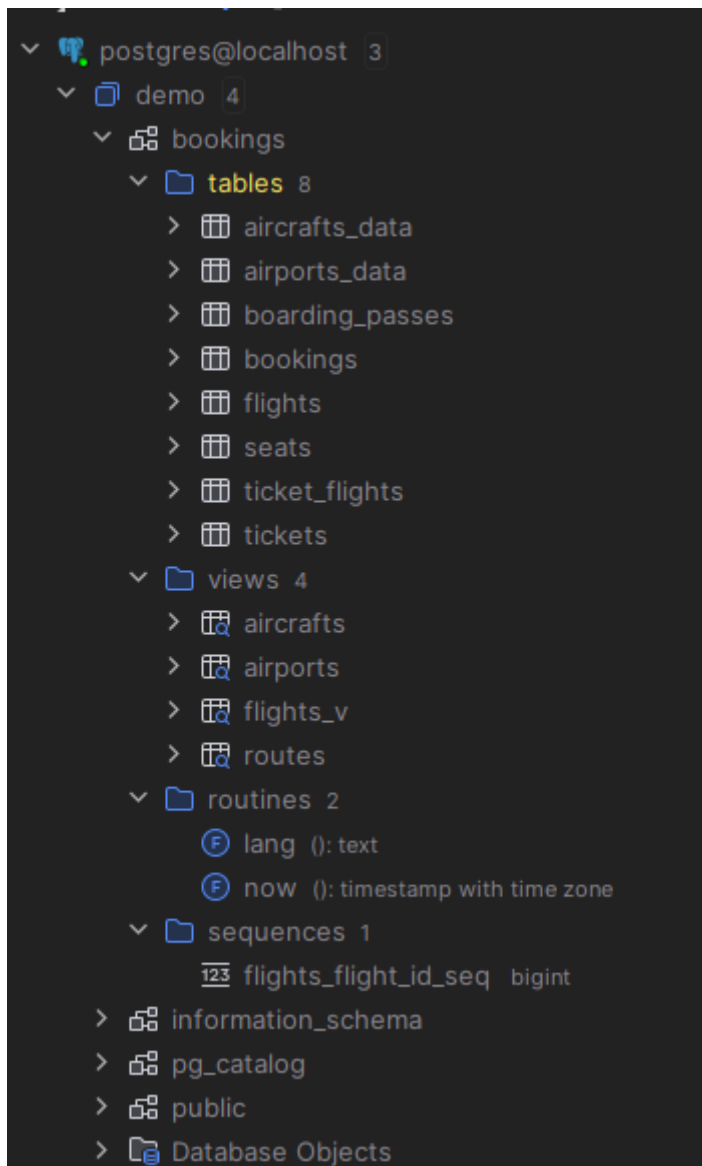


## Homework 6

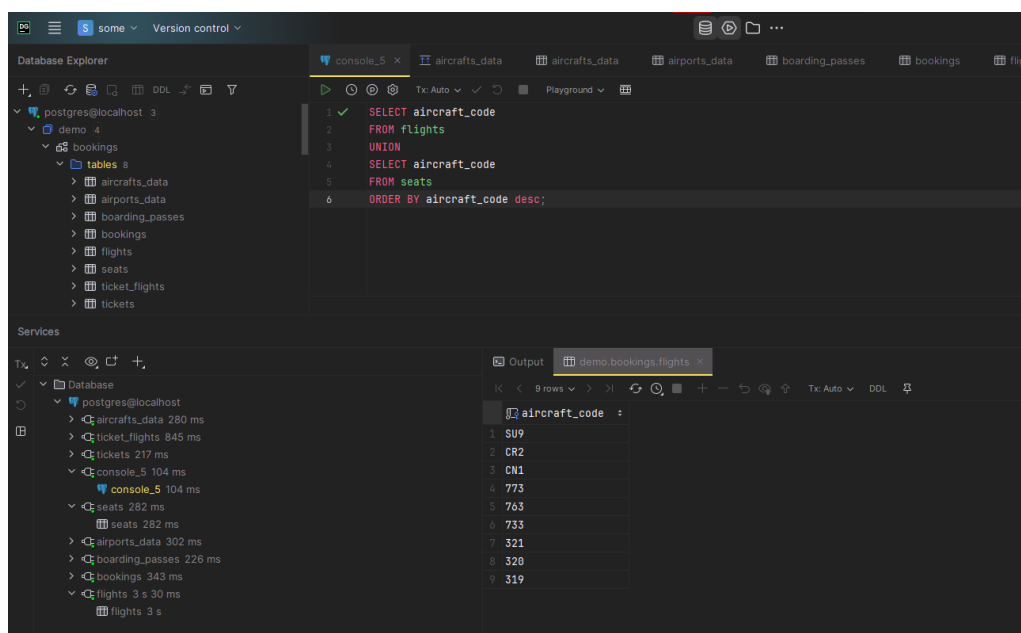
## 1. Скрин команды psql:



После выполнения команды `psql` в DataGrip появилась таблица “demo”, дамپ которой и производился ранее



2. Объединение данных (aircraft\_code) из таблиц flights и seats дало следующий результат:



### 3. Запрос с использованием WHERE и LIMIT:

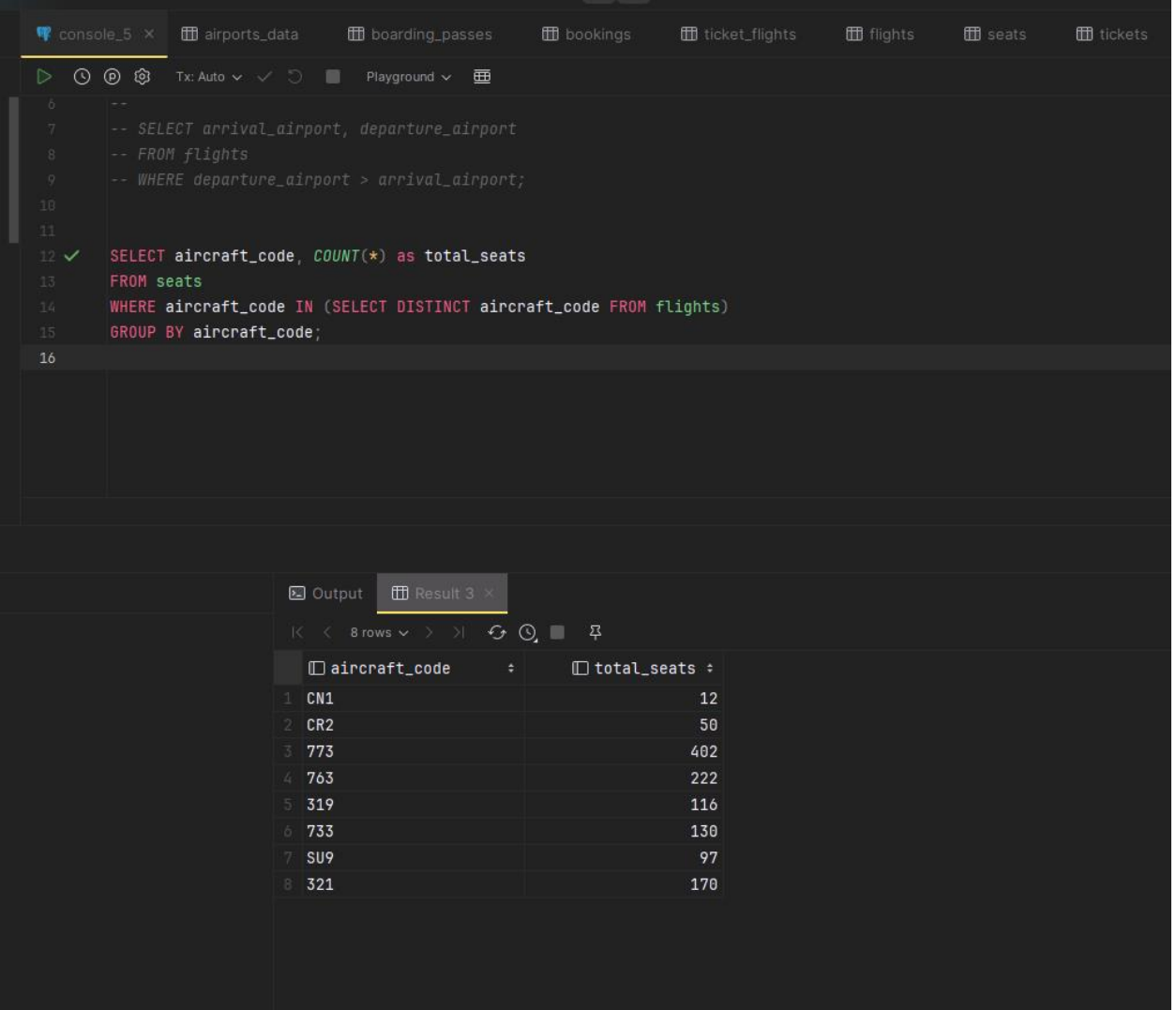
The screenshot shows a database client interface with a dark theme. The top panel displays a SQL query in the 'console\_5' tab:

```
1 SELECT flight_id
2 FROM ticket_flights
3 WHERE amount > 20000
4 LIMIT 30;
```

The bottom panel shows the 'Output' tab for the query, displaying 30 rows of results. The first column is labeled 'flight\_id'.

flight_id
300625
30625
30625
30625
30625
30625
30625
30625
30625
30625
30625
30625
30625
19092
19092
3993
3993
3993
3993
22080
6547
6547
1654
1654
1654

#### 4. Аналитический запрос, с помощью которого мы узнаём общее количество мест в каждом самолёте:



The screenshot shows a SQL playground interface with a dark theme. At the top, there are tabs for different database tables: `airports_data`, `boarding_passes`, `bookings`, `ticket_flights`, `flights`, `seats`, and `tickets`. The `console_5` tab is active, displaying a SQL query. The query is as follows:

```
--  
-- SELECT arrival_airport, departure_airport  
-- FROM flights  
-- WHERE departure_airport > arrival_airport;  
  
SELECT aircraft_code, COUNT(*) as total_seats  
FROM seats  
WHERE aircraft_code IN (SELECT DISTINCT aircraft_code FROM flights)  
GROUP BY aircraft_code;
```

Below the query editor, the results are displayed in a table. The table has two columns: `aircraft_code` and `total_seats`. There are 8 rows of data.

aircraft_code	total_seats
CN1	12
CR2	50
773	402
763	222
319	116
733	130
SU9	97
321	170

## 5. Уникальные неупорядоченные пары:

The screenshot shows a SQL playground interface with a query editor and an output table.

**Query Editor:**

```
8 FROM flights
9 -- WHERE EXISTS (SELECT seat_no FROM seats WHERE seats.aircraft_code = flights.aircraft_code)
10 -- AND flights.status IN ('Delayed', 'Cancelled', 'Departed')
11 -- GROUP BY flight_no;
12
13 ✓ SELECT departure_airport, arrival_airport
14 FROM flights
15 WHERE departure_airport <= arrival_airport
16
17 UNION
18
19 SELECT arrival_airport, departure_airport
20 FROM flights
21 WHERE departure_airport > arrival_airport;
22
```

**Output:**

Output table showing unique unordered airport pairs (HAVING COUNT(\*) = 2):

	departure_airport	arrival_airport
1	GDZ	ROV
2	DME	ULV
3	CEK	SGC
4	NBC	OMS
5	NJC	OVS
6	PES	TJM
7	IWA	SVX
8	ARH	NNM
9	KXK	SVX
10	KJA	NOZ
11	DME	MCX
12	OMS	VKO
13	DME	SCW
14	LED	OVS
15	ASF	DME
16	KVX	KZN
17	KYZ	SVX
18	IKT	LED
19	OMH	DME