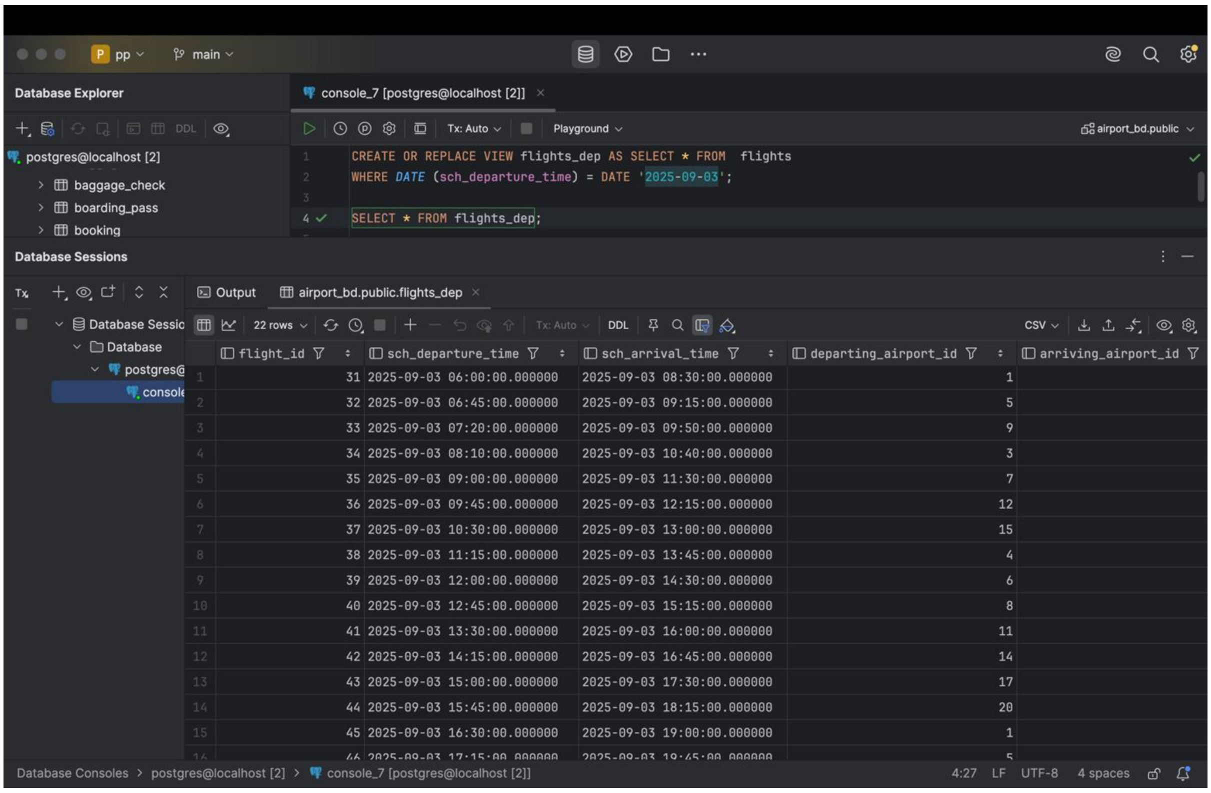
Laboratory work 8

1. Create a view to show details of all flights that are departing on a specific date.



```
CREATE OR REPLACE VIEW flights_dep AS SELECT * FROM  flights
WHERE DATE (sch_departure_time) = DATE '2025-09-03';

SELECT * FROM flights_dep;
```

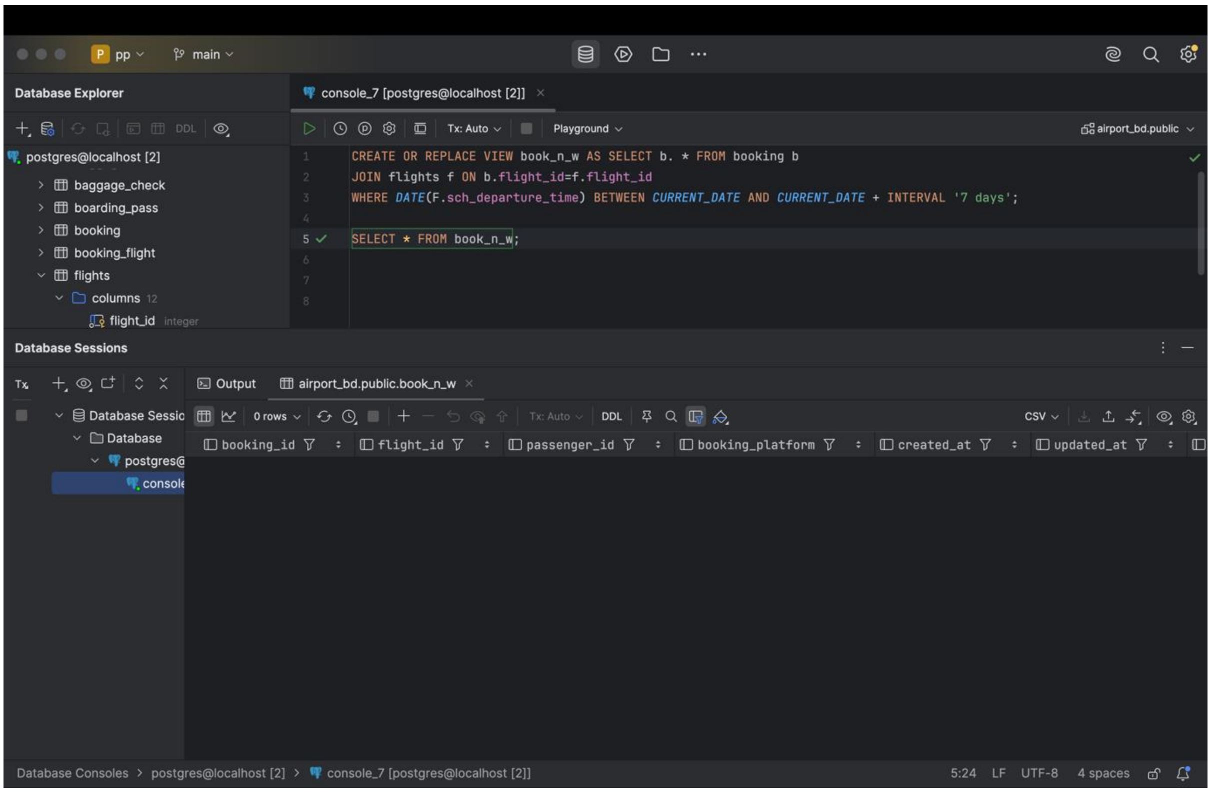
airport_bd.public.flights_dep — 22 rows

	flight_id	sch_departure_time	sch_arrival_time	departing_airport_id	arriving_airport_id
1	31	2025-09-03 06:00:00.000000	2025-09-03 08:30:00.000000	1	
2	32	2025-09-03 06:45:00.000000	2025-09-03 09:15:00.000000	5	
3	33	2025-09-03 07:20:00.000000	2025-09-03 09:50:00.000000	9	
4	34	2025-09-03 08:10:00.000000	2025-09-03 10:40:00.000000	3	
5	35	2025-09-03 09:00:00.000000	2025-09-03 11:30:00.000000	7	
6	36	2025-09-03 09:45:00.000000	2025-09-03 12:15:00.000000	12	
7	37	2025-09-03 10:30:00.000000	2025-09-03 13:00:00.000000	15	
8	38	2025-09-03 11:15:00.000000	2025-09-03 13:45:00.000000	4	
9	39	2025-09-03 12:00:00.000000	2025-09-03 14:30:00.000000	6	
10	40	2025-09-03 12:45:00.000000	2025-09-03 15:15:00.000000	8	
11	41	2025-09-03 13:30:00.000000	2025-09-03 16:00:00.000000	11	
12	42	2025-09-03 14:15:00.000000	2025-09-03 16:45:00.000000	14	
13	43	2025-09-03 15:00:00.000000	2025-09-03 17:30:00.000000	17	
14	44	2025-09-03 15:45:00.000000	2025-09-03 18:15:00.000000	20	
15	45	2025-09-03 16:30:00.000000	2025-09-03 19:00:00.000000	1	
16	46	2025-09-03 17:15:00.000000	2025-09-03 19:45:00.000000	5	

```
CREATE OR REPLACE VIEW flights_dep AS SELECT * FROM  flights
WHERE DATE (sch_departure_time) = DATE '2025-09-03';

SELECT * FROM flights_dep;
```

2. Create a view that shows bookings for flights scheduled to depart within the next week.

```
CREATE OR REPLACE VIEW book_n_w AS SELECT b. * FROM booking b
JOIN flights f ON b.flight_id=f.flight_id
WHERE DATE(F.sch_departure_time) BETWEEN CURRENT_DATE AND CURRENT_DATE +
INTERVAL '7 days';

SELECT * FROM book_n_w;
```

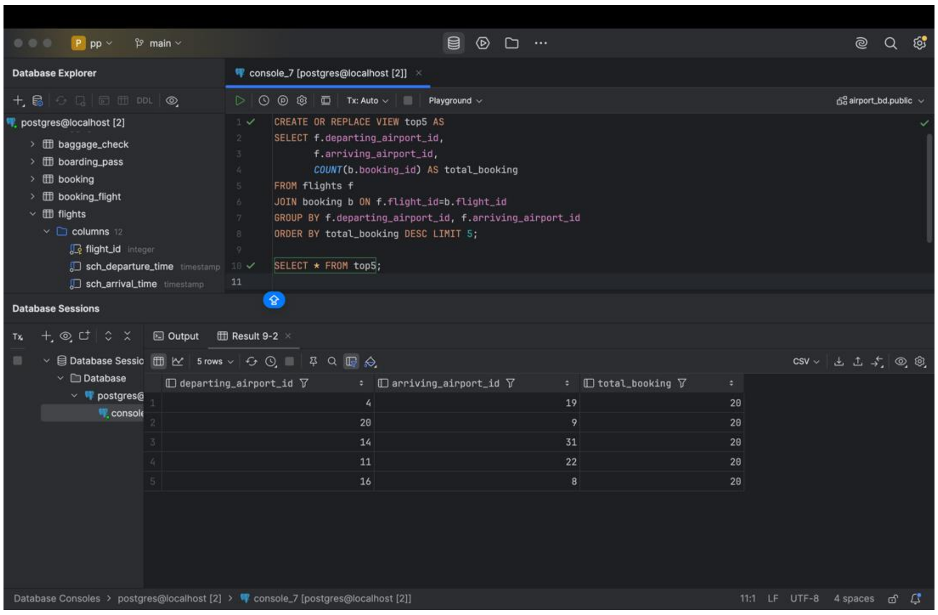


```
CREATE OR REPLACE VIEW book_n_w AS SELECT b. * FROM booking b
JOIN flights f ON b.flight_id=f.flight_id
WHERE DATE(F.sch_departure_time) BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '7 days';

SELECT * FROM book_n_w;
```

```
booking_id   flight_id   passenger_id   booking_platform   created_at   updated_at
```

3. Create a view to show the top 5 most popular flight routes based on the number of bookings.

```
CREATE OR REPLACE VIEW top5 AS
SELECT f.departing_airport_id,
       f.arriving_airport_id,
       COUNT(b.booking_id) AS total_booking
FROM flights f
JOIN booking b ON f.flight_id=b.flight_id
GROUP BY f.departing_airport_id, f.arriving_airport_id
ORDER BY total_booking DESC LIMIT 5;

SELECT * FROM top5;
```

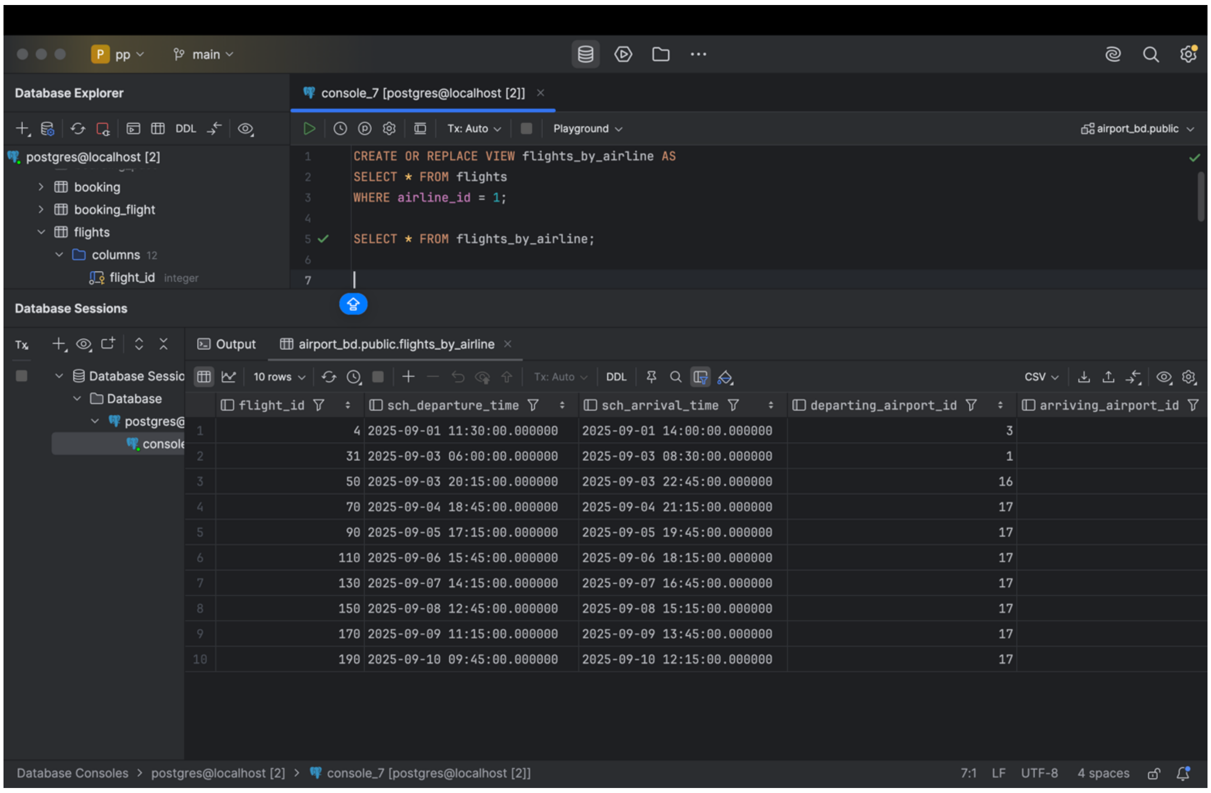


```
CREATE OR REPLACE VIEW top5 AS
SELECT f.departing_airport_id,
       f.arriving_airport_id,
       COUNT(b.booking_id) AS total_booking
FROM flights f
JOIN booking b ON f.flight_id=b.flight_id
GROUP BY f.departing_airport_id, f.arriving_airport_id
ORDER BY total_booking DESC LIMIT 5;

SELECT * FROM top5;
```

```
   departing_airport_id   arriving_airport_id   total_booking
1                     4                    19              20
2                    20                     9              20
3                    14                    31              20
4                    11                    22              20
5                    16                     8              20
```

4. Create a view that lists all flights for a specific airline.

```sql
CREATE OR REPLACE VIEW flights_by_airline AS
SELECT * FROM flights
WHERE airline_id = 1;

SELECT * FROM flights_by_airline;
```



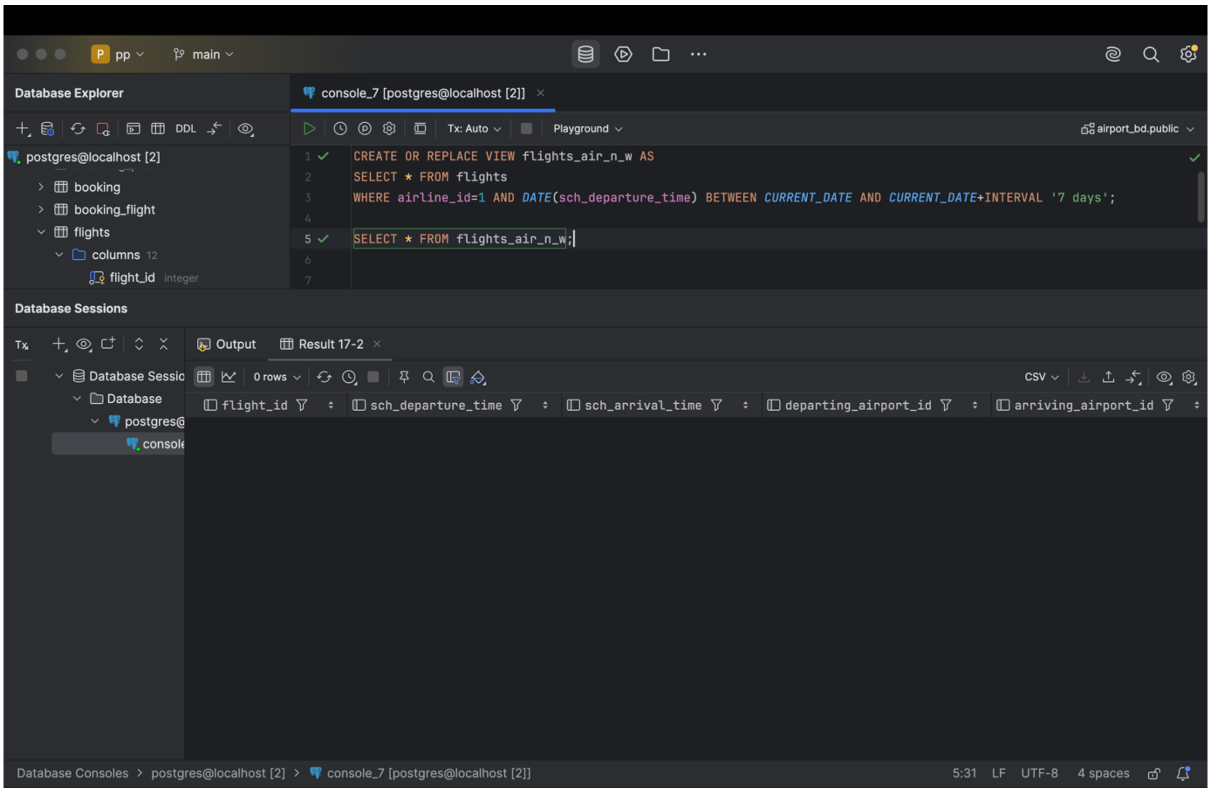
```sql
CREATE OR REPLACE VIEW flights_by_airline AS
SELECT * FROM flights
WHERE airline_id = 1;

SELECT * FROM flights_by_airline;
```

	flight_id	sch_departure_time	sch_arrival_time	departing_airport_id	arriving_airport_id
1	4	2025-09-01 11:30:00.000000	2025-09-01 14:00:00.000000	3	
2	31	2025-09-03 06:00:00.000000	2025-09-03 08:30:00.000000	1	
3	50	2025-09-03 20:15:00.000000	2025-09-03 22:45:00.000000	16	
4	70	2025-09-04 18:45:00.000000	2025-09-04 21:15:00.000000	17	
5	90	2025-09-05 17:15:00.000000	2025-09-05 19:45:00.000000	17	
6	110	2025-09-06 15:45:00.000000	2025-09-06 18:15:00.000000	17	
7	130	2025-09-07 14:15:00.000000	2025-09-07 16:45:00.000000	17	
8	150	2025-09-08 12:45:00.000000	2025-09-08 15:15:00.000000	17	
9	170	2025-09-09 11:15:00.000000	2025-09-09 13:45:00.000000	17	
10	190	2025-09-10 09:45:00.000000	2025-09-10 12:15:00.000000	17	

Database Consoles > postgres@localhost [2] > console_7 [postgres@localhost [2]]

5. Modify the view created in task 4 to show only flights departing within the next 7 days for a specific airline.

```sql
CREATE OR REPLACE VIEW flights_air_n_w AS
SELECT * FROM flights
WHERE airline_id=1 AND DATE(sch_departure_time) BETWEEN CURRENT_DATE
AND CURRENT_DATE+INTERVAL '7 days';

SELECT * FROM flights_air_n_w;
```

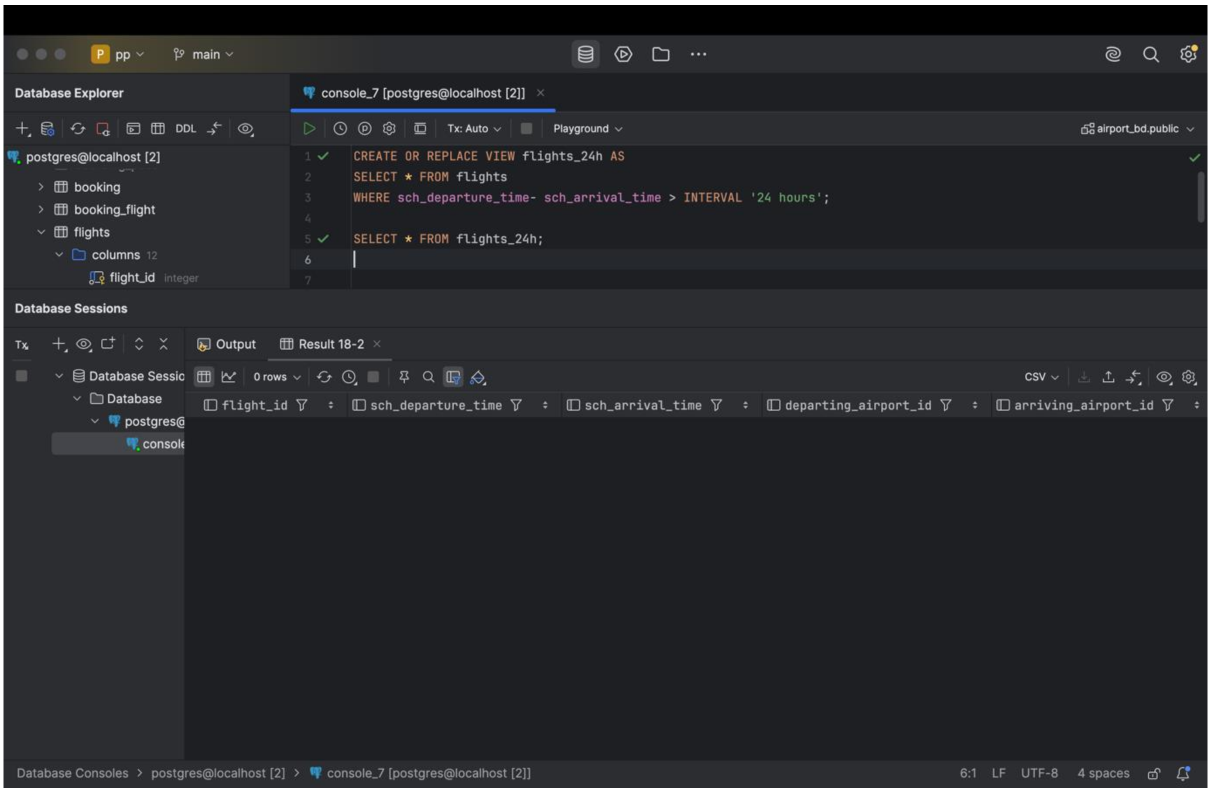


```sql
CREATE OR REPLACE VIEW flights_air_n_w AS
SELECT * FROM flights
WHERE airline_id=1 AND DATE(sch_departure_time) BETWEEN CURRENT_DATE AND CURRENT_DATE+INTERVAL '7 days';

SELECT * FROM flights_air_n_w;
```

```
flight_id | sch_departure_time | sch_arrival_time | departing_airport_id | arriving_airport_id
```

6. Create a view to show flights that are delayed by more than 24 hours.

```sql
CREATE OR REPLACE VIEW flights_24h AS
SELECT * FROM flights
WHERE sch_departure_time- sch_arrival_time > INTERVAL '24 hours';

SELECT * FROM flights_24h;
```

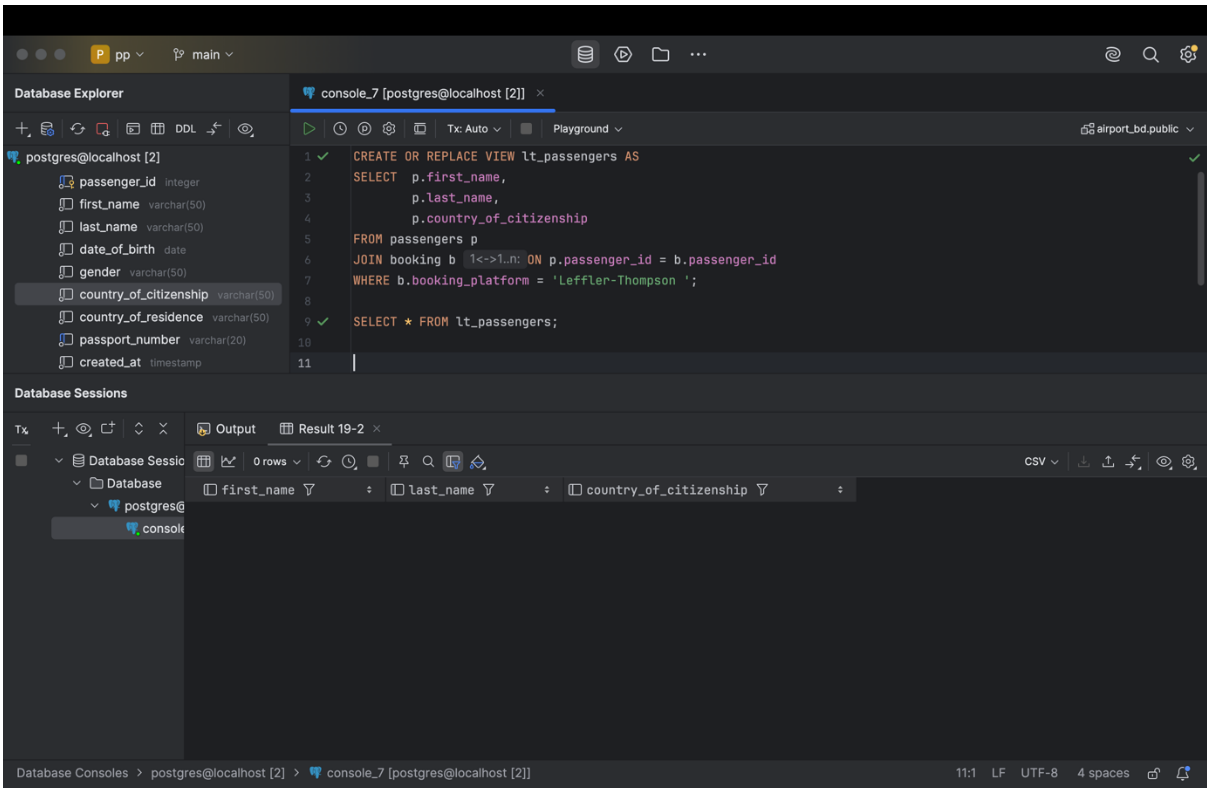


```sql
CREATE OR REPLACE VIEW flights_24h AS
SELECT * FROM flights
WHERE sch_departure_time- sch_arrival_time > INTERVAL '24 hours';

SELECT * FROM flights_24h;
```

```
flight_id | sch_departure_time | sch_arrival_time | departing_airport_id | arriving_airport_id
```

7. Create a view in which you can display the full name and country of origin of passengers who made bookings on Leffler-Thompson platform. Then show the list of that passengers.

```sql
CREATE OR REPLACE VIEW lt_passengers AS
SELECT  p.first_name,
        p.last_name,
        p.country_of_citizenship
FROM passengers p
JOIN booking b ON p.passenger_id = b.passenger_id
WHERE b.booking_platform = 'Leffler-Thompson ';

SELECT * FROM lt_passengers;
```



```
postgres@localhost [2]
    passenger_id  integer
    first_name  varchar(50)
    last_name  varchar(50)
    date_of_birth  date
    gender  varchar(50)
    country_of_citizenship  varchar(50)
    country_of_residence  varchar(50)
    passport_number  varchar(20)
    created_at  timestamp
```

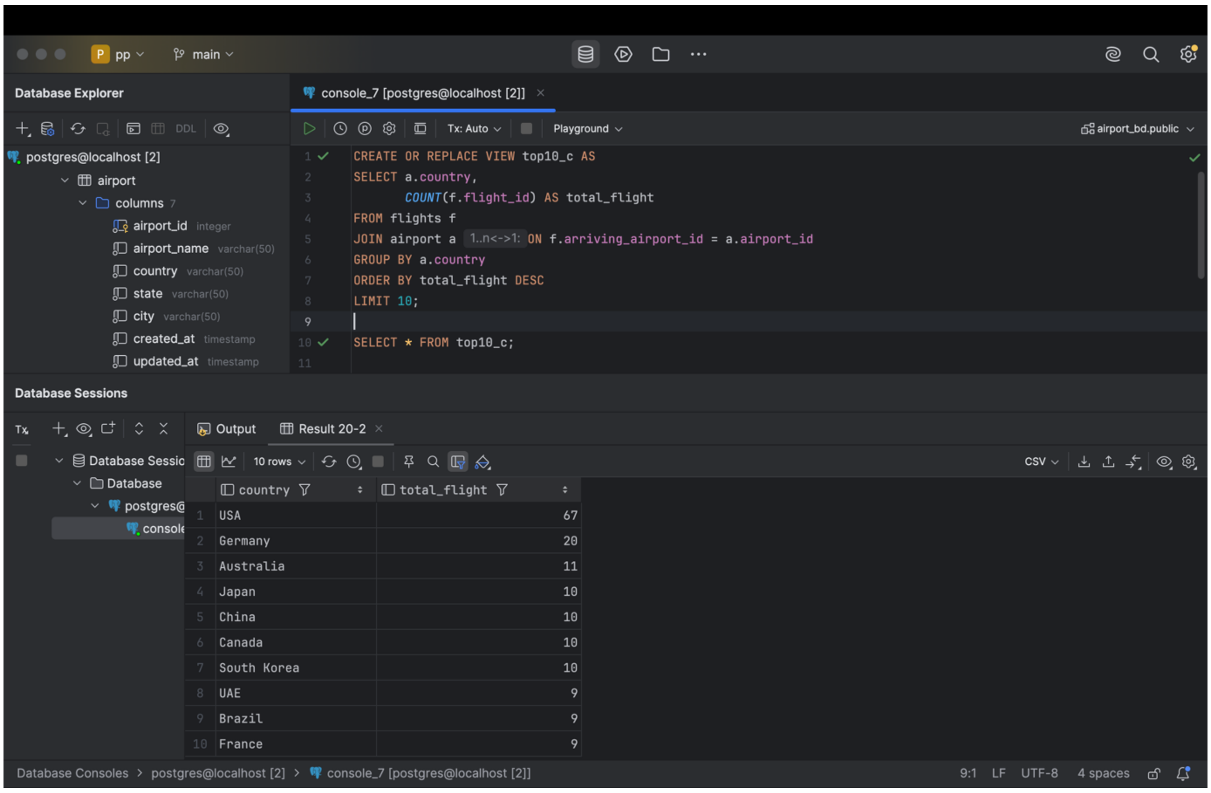
```sql
CREATE OR REPLACE VIEW lt_passengers AS
SELECT  p.first_name,
        p.last_name,
        p.country_of_citizenship
FROM passengers p
JOIN booking b 1<->1..n: ON p.passenger_id = b.passenger_id
WHERE b.booking_platform = 'Leffler-Thompson ';

SELECT * FROM lt_passengers;
```

```
first_name | last_name | country_of_citizenship
```

```
Database Consoles > postgres@localhost [2] > console_7 [postgres@localhost [2]]    11:1  LF  UTF-8  4 spaces
```

8. Create a view that shows top 10 most visited countries.

```sql
CREATE OR REPLACE VIEW top10_c AS
SELECT a.country,
       COUNT(f.flight_id) AS total_flight
FROM flights f
JOIN airport a ON f.arriving_airport_id = a.airport_id
GROUP BY a.country
ORDER BY total_flight DESC
LIMIT 10;

SELECT * FROM top10_c;
```

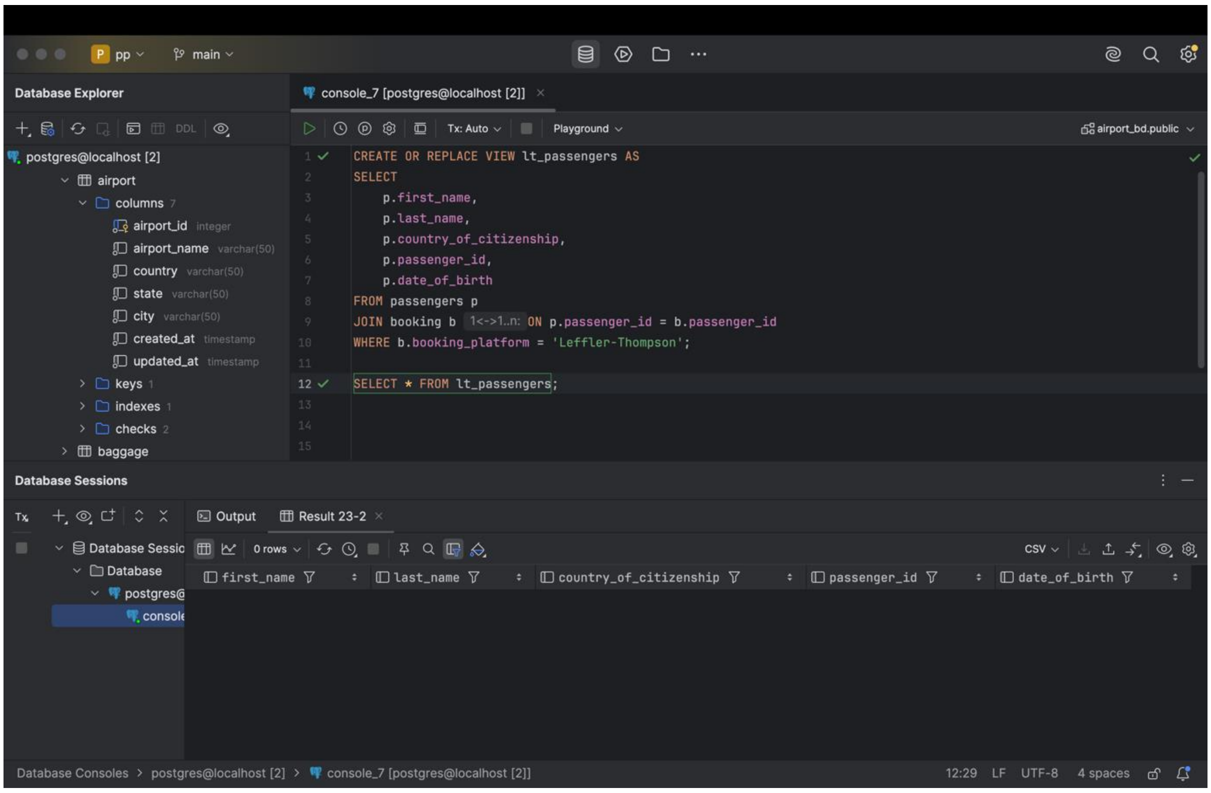


```
CREATE OR REPLACE VIEW top10_c AS
SELECT a.country,
       COUNT(f.flight_id) AS total_flight
FROM flights f
JOIN airport a 1..n<->1: ON f.arriving_airport_id = a.airport_id
GROUP BY a.country
ORDER BY total_flight DESC
LIMIT 10;

SELECT * FROM top10_c;
```

	country	total_flight
1	USA	67
2	Germany	20
3	Australia	11
4	Japan	10
5	China	10
6	Canada	10
7	South Korea	10
8	UAE	9
9	Brazil	9
10	France	9

9. Update any of the created views by adding new information in the view table. Show results.

```
CREATE OR REPLACE VIEW lt_passengers AS
SELECT
    p.first_name,
    p.last_name,
    p.country_of_citizenship,
    p.passenger_id,
    p.date_of_birth
FROM passengers p
JOIN booking b ON p.passenger_id = b.passenger_id
WHERE b.booking_platform = 'Leffler-Thompson';

SELECT * FROM lt_passengers;
```

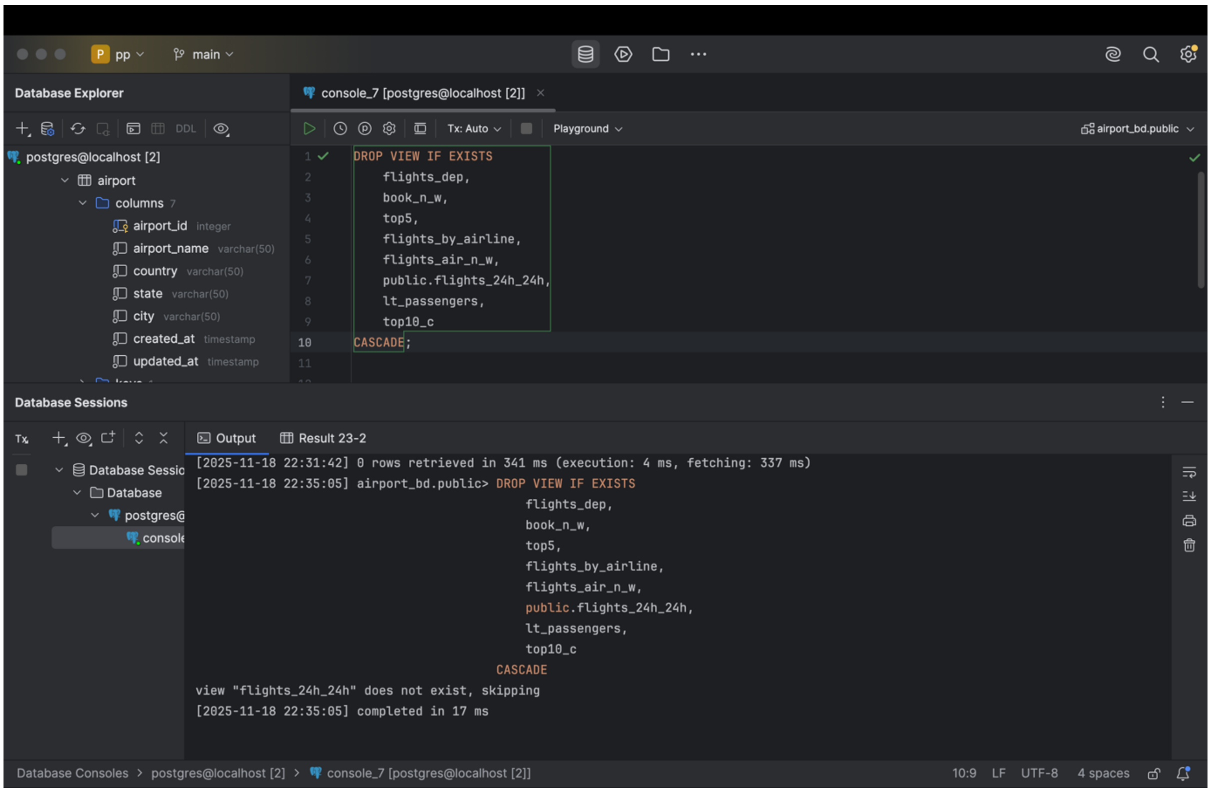


```sql
CREATE OR REPLACE VIEW lt_passengers AS
SELECT
    p.first_name,
    p.last_name,
    p.country_of_citizenship,
    p.passenger_id,
    p.date_of_birth
FROM passengers p
JOIN booking b ON p.passenger_id = b.passenger_id
WHERE b.booking_platform = 'Leffler-Thompson';

SELECT * FROM lt_passengers;
```

```
first_name | last_name | country_of_citizenship | passenger_id | date_of_birth
```

10. Drop all existing views.

```sql
DROP VIEW IF EXISTS
    flights_dep,
    book_n_w,
    top5,
    flights_by_airline,
    flights_air_n_w,
    public.flights_24h_24h,
    lt_passengers,
    top10_c
CASCADE;
```



```sql
DROP VIEW IF EXISTS
    flights_dep,
    book_n_w,
    top5,
    flights_by_airline,
    flights_air_n_w,
    public.flights_24h_24h,
    lt_passengers,
    top10_c
CASCADE;
```

```
[2025-11-18 22:31:42] 0 rows retrieved in 341 ms (execution: 4 ms, fetching: 337 ms)
[2025-11-18 22:35:05] airport_bd.public> DROP VIEW IF EXISTS
                                             flights_dep,
                                             book_n_w,
                                             top5,
                                             flights_by_airline,
                                             flights_air_n_w,
                                             public.flights_24h_24h,
                                             lt_passengers,
                                             top10_c
                                         CASCADE
view "flights_24h_24h" does not exist, skipping
[2025-11-18 22:35:05] completed in 17 ms
```