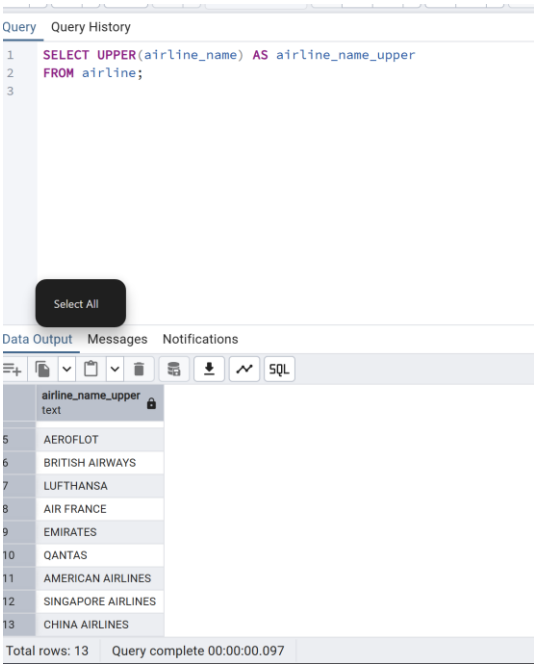
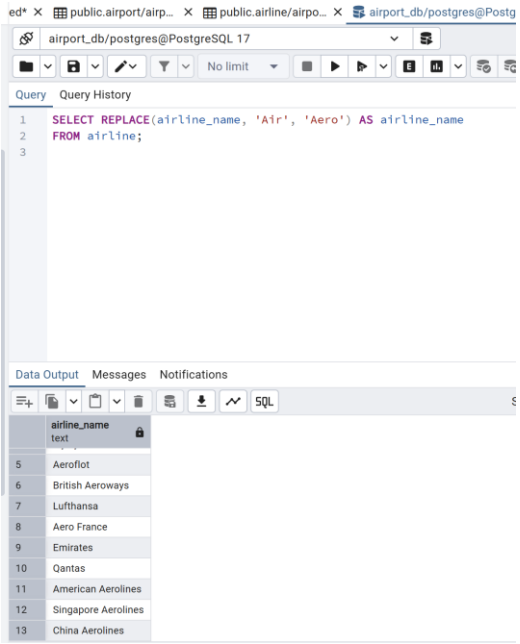


1)SELECT UPPER(airline_name) AS airline_name_upper

FROM airline;

2) SELECT REPLACE(airline_name, 'Air', 'Aero') AS modified_airline_name

FROM airline;



3)

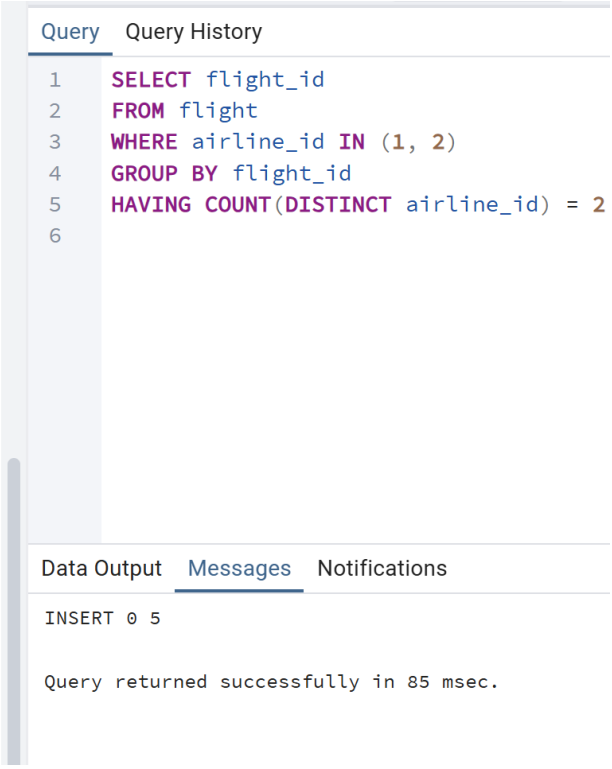
SELECT flight_id

FROM flight

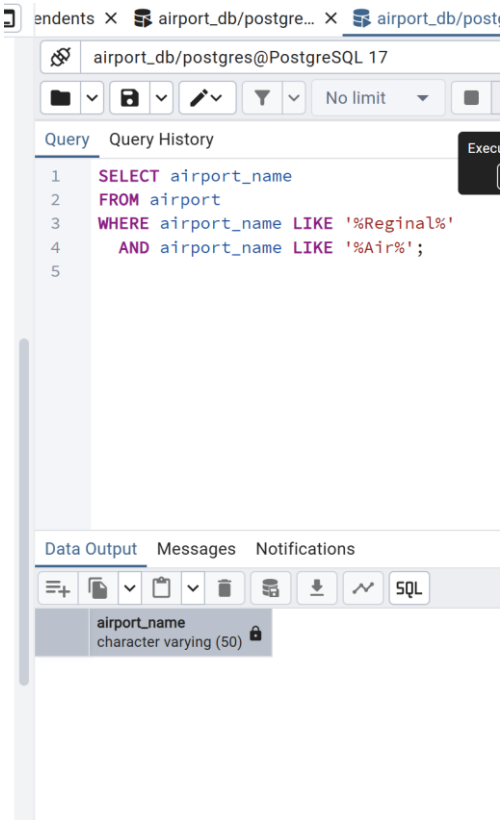
WHERE airline_id IN (1, 2)

GROUP BY flight_id

HAVING COUNT(DISTINCT airline_id) = 2;



4)



SELECT airport_name

FROM airport

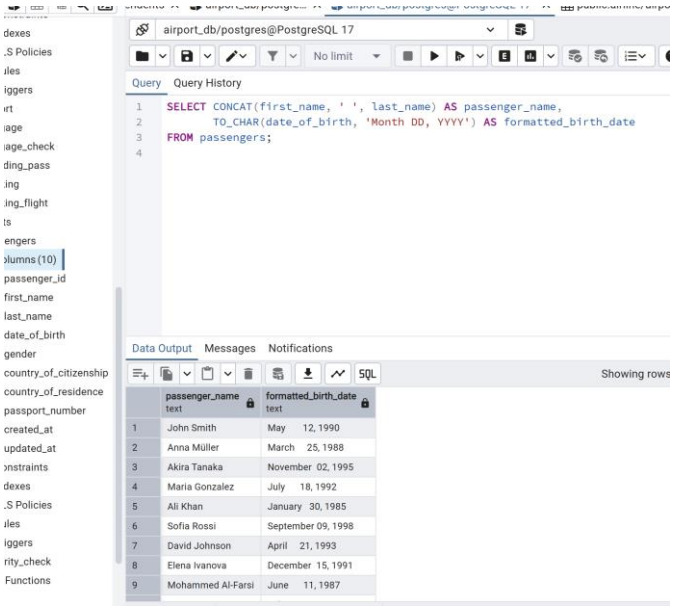
WHERE airport_name LIKE '%Reginal%' AND airport_name LIKE '%Air%';

5)

SELECT CONCAT(first_name, ' ', last_name) AS passenger_name,

TO_CHAR(date_of_birth, 'Month DD, YYYY') AS formatted_birth_date

FROM passengers;

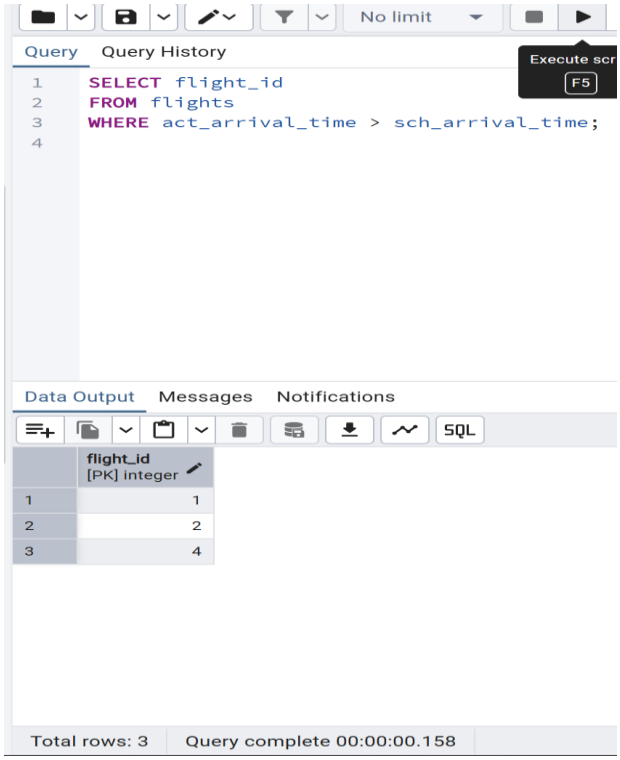


6)

SELECT flight_id

FROM flights

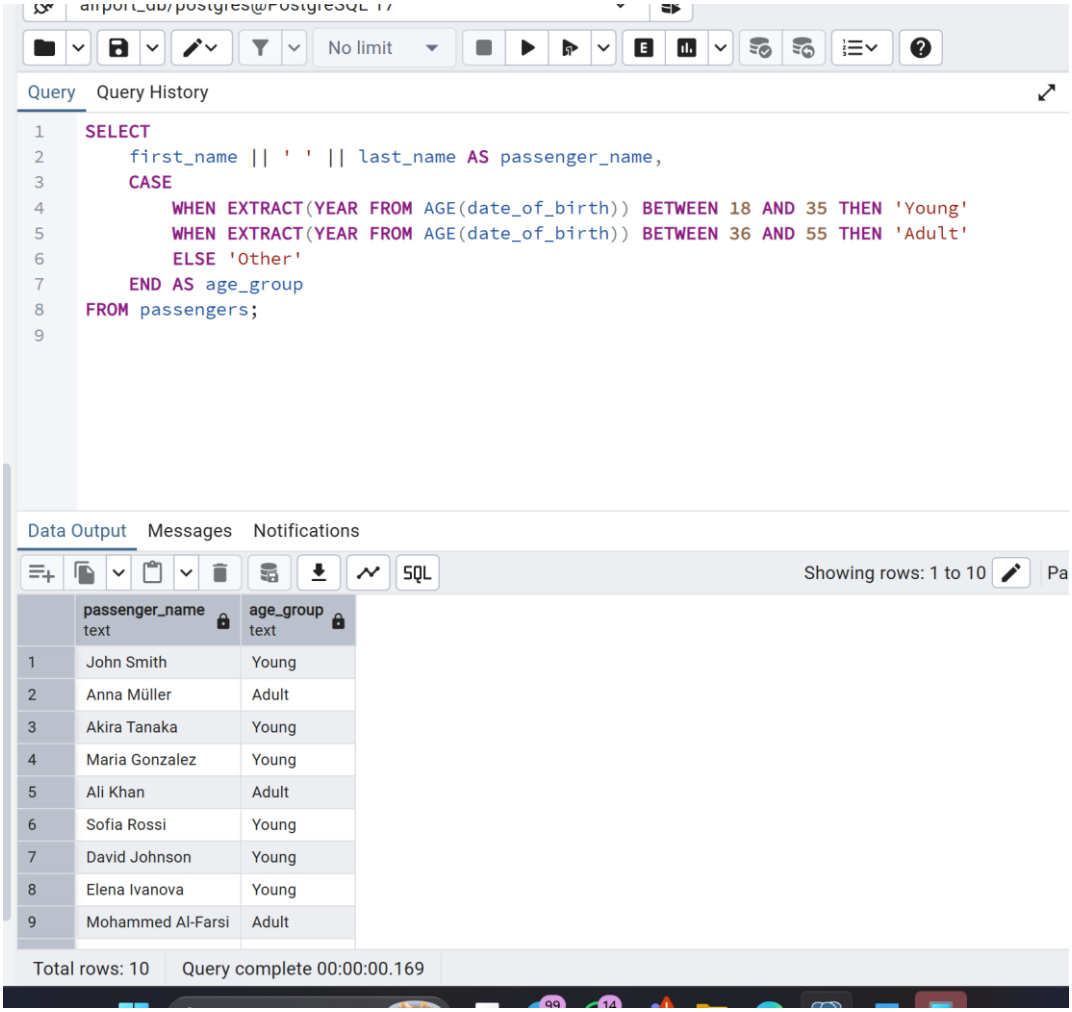
WHERE act_arrival_time > sch_arrival_time;



7)

```
SELECT
  first_name || ' ' || last_name AS passenger_name,
  CASE
    WHEN EXTRACT(YEAR FROM AGE(date_of_birth)) BETWEEN 18 AND 35 THEN 'Young'
    WHEN EXTRACT(YEAR FROM AGE(date_of_birth)) BETWEEN 36 AND 55 THEN 'Adult'
  END AS age_group
FROM passengers;
```

(|| – in PostgreSQL, this is the string concatenation operator.That is, first_name || ' ' || last_name takes the first name (first_name), adds a space (' '), and then the last name (last_name).For example: "Ivan" || ' ' || "Petrov" → "Ivan Petrov".AS passenger_name – this is a column alias.It sets the name of the resulting column.)

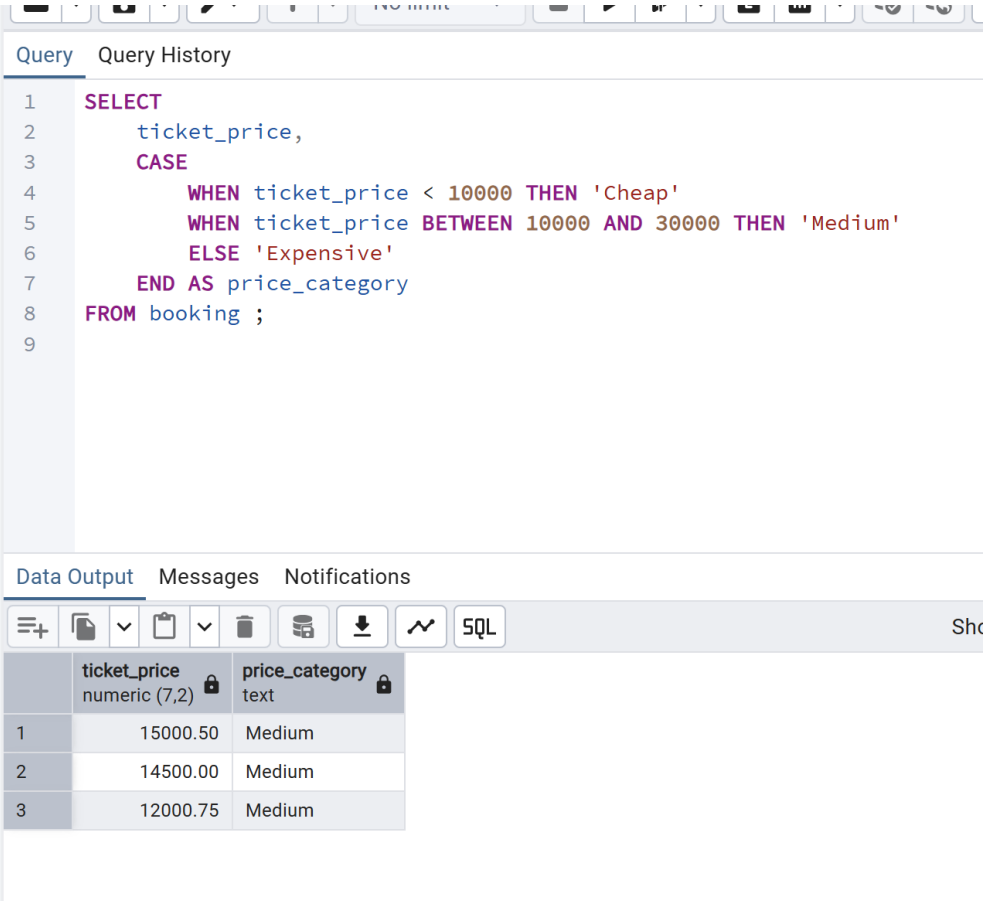


	passenger_name text	age_group text
1	John Smith	Young
2	Anna Müller	Adult
3	Akira Tanaka	Young
4	Maria Gonzalez	Young
5	Ali Khan	Adult
6	Sofia Rossi	Young
7	David Johnson	Young
8	Elena Ivanova	Young
9	Mohammed Al-Farsi	Adult

8)

```
SELECT
  ticket_price,
```

```
  CASE
    WHEN ticket_price < 10000 THEN 'Cheap'
    WHEN ticket_price BETWEEN 10000 AND 30000 THEN 'Medium'
    ELSE 'Expensive'
  END AS price_category
FROM booking ;
```



9)

```
SELECT airline_country, COUNT(*) AS airline_count
FROM airline
GROUP BY airline_country;
```

“ COUNT(*) is an aggregate function that counts the number of rows.* means "count all rows."If you just write COUNT(*), it counts all rows in each group (or in the entire table if there is no GROUP BY).Example: if there are 5 airlines from Kazakhstan in a table, then COUNT(*) for the "Kazakhstan" group will return 5. ”

Query Query History

```
SELECT airline_country, COUNT(*) AS airline_count
FROM airline
GROUP BY airline_country;
```

Data Output Messages Notifications

	airline_country character varying (50)	airline_count bigint
1	Singapore	1
2	France	2
3	United States	1
4	Turkey	1
5	China	1
6	Australia	1
7	United Arab Emirates	1
8	United Kingdom	1
9	Germany	1

Total rows: 12 Query complete 00:00:00.199

10)

SELECT flight_id, sch_arrival_time, act_arrival_time

FROM flights

WHERE act_arrival_time > sch_arrival_time;

Query Query History

```
SELECT flight_id, sch_arrival_time, act_arrival_time
FROM flights
WHERE act_arrival_time > sch_arrival_time;
```

Data Output Messages Notifications

Show

	flight_id [PK] integer	sch_arrival_time timestamp without time zone	act_arrival_time timestamp without time zone
1	1	2025-10-05 11:00:00	2025-10-05 11:20:00
2	2	2025-10-06 18:00:00	2025-10-06 18:05:00
3	4	2025-10-09 00:30:00	2025-10-09 00:45:00

Filtering: we display only those flights where the actual arrival time is later than the scheduled time (that is, the flight arrived behind schedule).