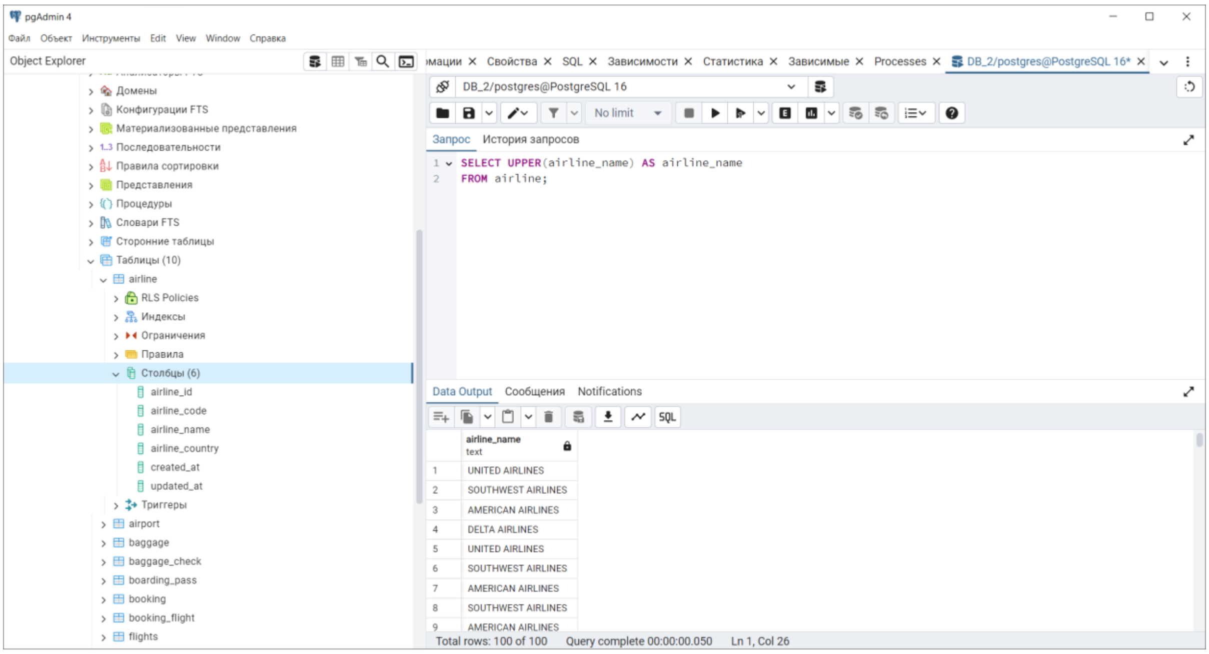
Laboratory work 4

Tasks:

1. Retrieve all airline names in uppercase.

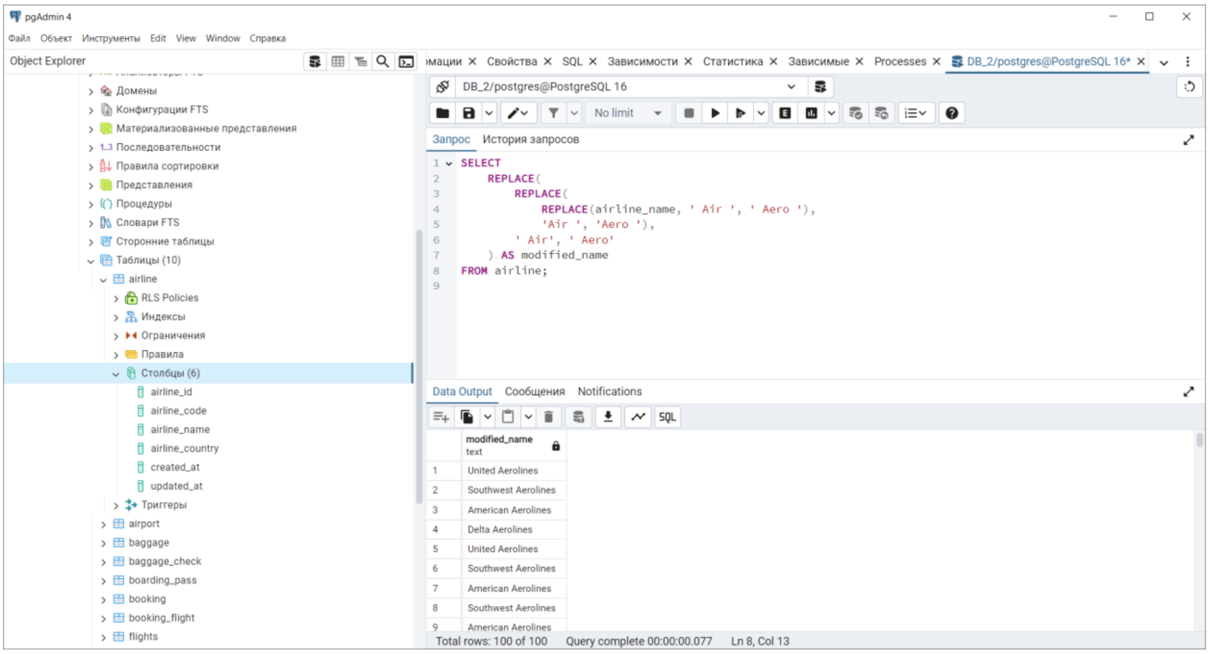


```sql
SELECT UPPER(airline_name) AS airline_name
FROM airline;
```

	airline_name text
1	UNITED AIRLINES
2	SOUTHWEST AIRLINES
3	AMERICAN AIRLINES
4	DELTA AIRLINES
5	UNITED AIRLINES
6	SOUTHWEST AIRLINES
7	AMERICAN AIRLINES
8	SOUTHWEST AIRLINES
9	AMERICAN AIRLINES

Total rows: 100 of 100 Query complete 00:00:00.050 Ln 1, Col 26

2. Replace any occurrence of the word "Air" in airline names with "Aero".

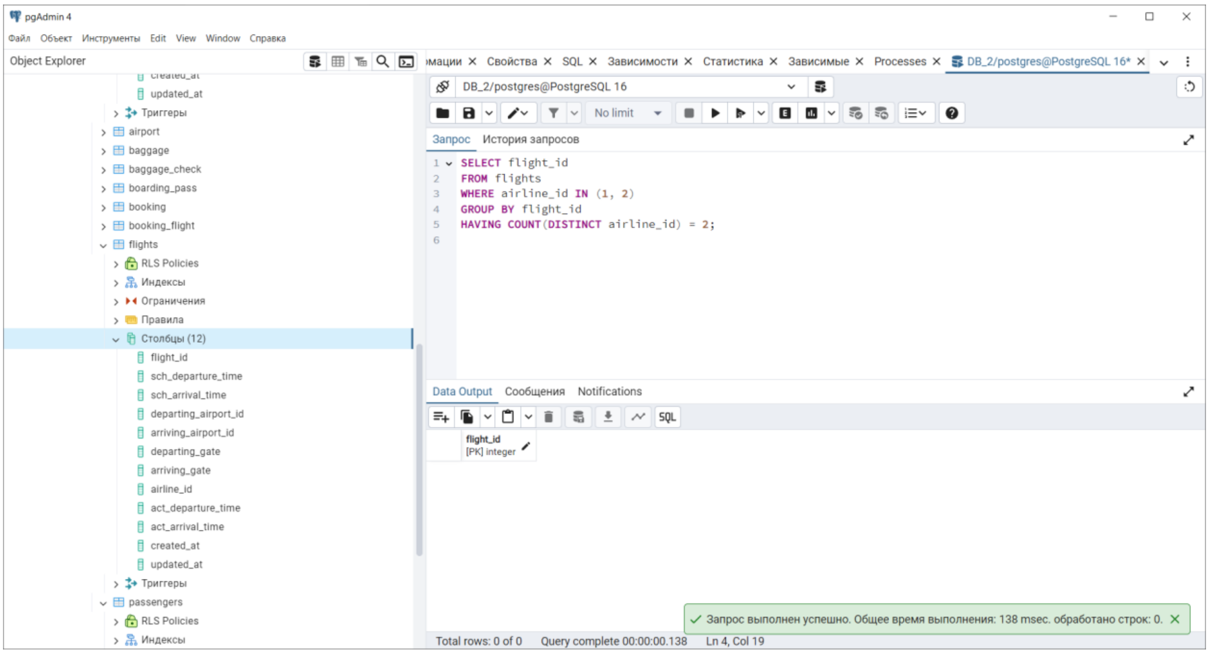


```sql
SELECT
    REPLACE(
        REPLACE(
            REPLACE(airline_name, ' Air ', ' Aero '),
            'Air ', 'Aero '),
        ' Air', ' Aero'
    ) AS modified_name
FROM airline;
```

	modified_name text
1	United Aerolines
2	Southwest Aerolines
3	American Aerolines
4	Delta Aerolines
5	United Aerolines
6	Southwest Aerolines
7	American Aerolines
8	Southwest Aerolines
9	American Aerolines

Total rows: 100 of 100 Query complete 00:00:00.077 Ln 8, Col 13

3. Fing all flight numbers that coordinates with both airline 1 and airline 2.



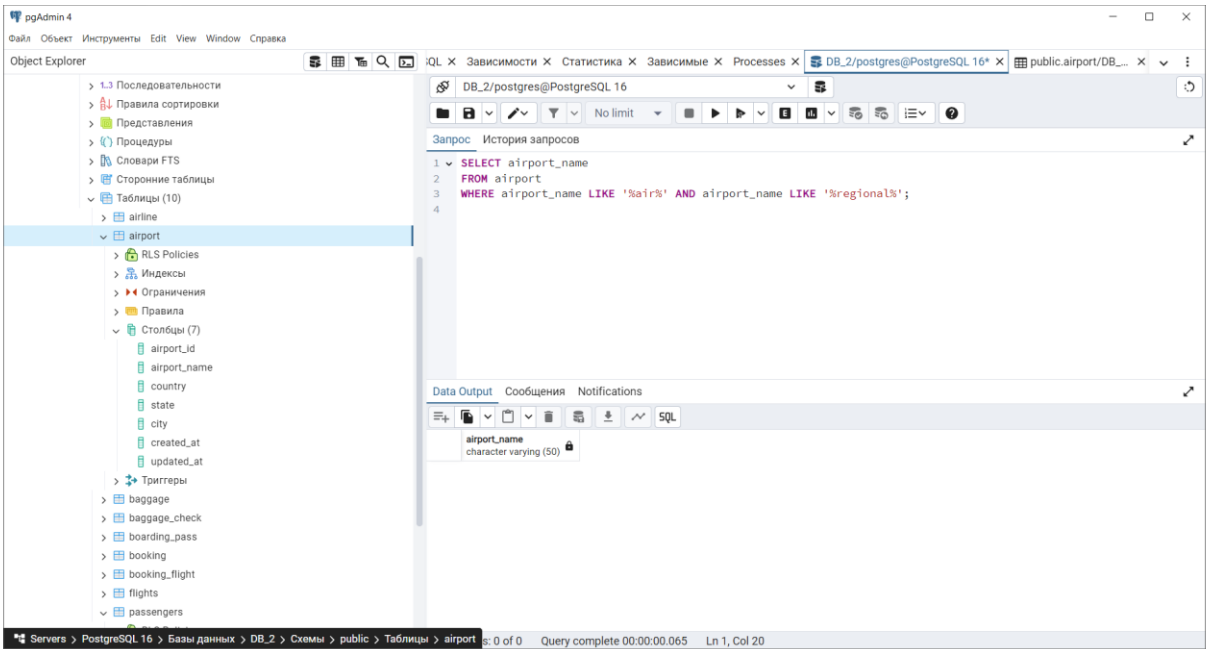
```
SELECT flight_id
FROM flights
WHERE airline_id IN (1, 2)
GROUP BY flight_id
HAVING COUNT(DISTINCT airline_id) = 2;
```

flight_id
[PK] integer

✓ Запрос выполнен успешно. Общее время выполнения: 138 msec. обработано строк: 0.

Total rows: 0 of 0   Query complete 00:00:00.138   Ln 4, Col 19

4. Retrieve airports that contain the word "Reginal" and "Air" in their names.



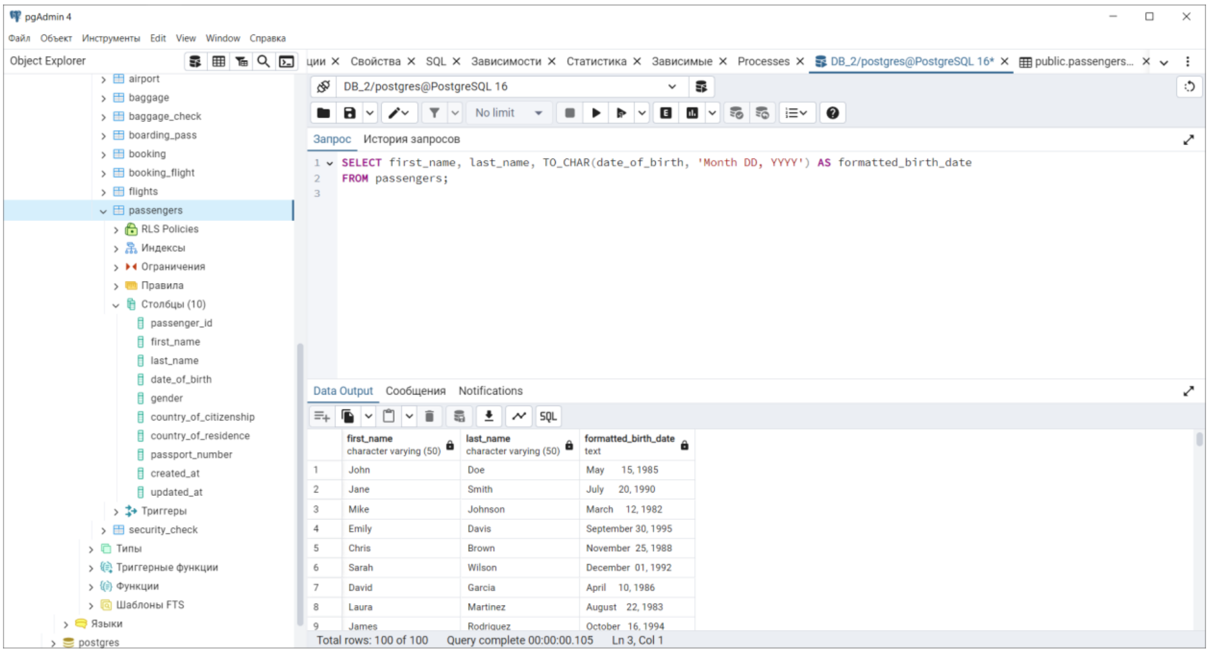
```
SELECT airport_name
FROM airport
WHERE airport_name LIKE '%air%' AND airport_name LIKE '%regional%';
```

airport_name
character varying (50)

Servers > PostgreSQL 16 > Базы данных > DB_2 > Схемы > public > Таблицы > airport

s: 0 of 0   Query complete 00:00:00.065   Ln 1, Col 20

5. Retrieve passenger names and format their birth dates as 'Month DD, YYYY'.

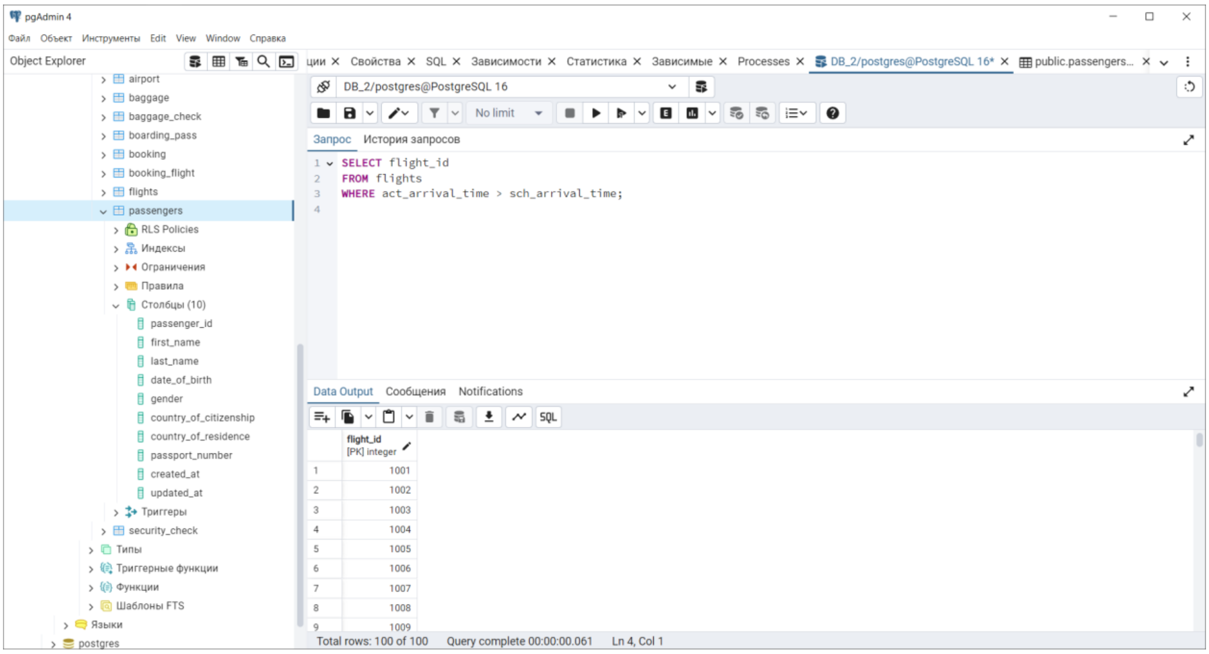


```
SELECT first_name, last_name, TO_CHAR(date_of_birth, 'Month DD, YYYY') AS formatted_birth_date
FROM passengers;
```

	first_name character varying (50)	last_name character varying (50)	formatted_birth_date text
1	John	Doe	May 15, 1985
2	Jane	Smith	July 20, 1990
3	Mike	Johnson	March 12, 1982
4	Emily	Davis	September 30, 1995
5	Chris	Brown	November 25, 1988
6	Sarah	Wilson	December 01, 1992
7	David	Garcia	April 10, 1986
8	Laura	Martinez	August 22, 1983
9	James	Rodriguez	October 16, 1994

Total rows: 100 of 100 Query complete 00:00:00.105 Ln 3, Col 1

6. Find flight numbers that have been delayed based on the actual arrival time.

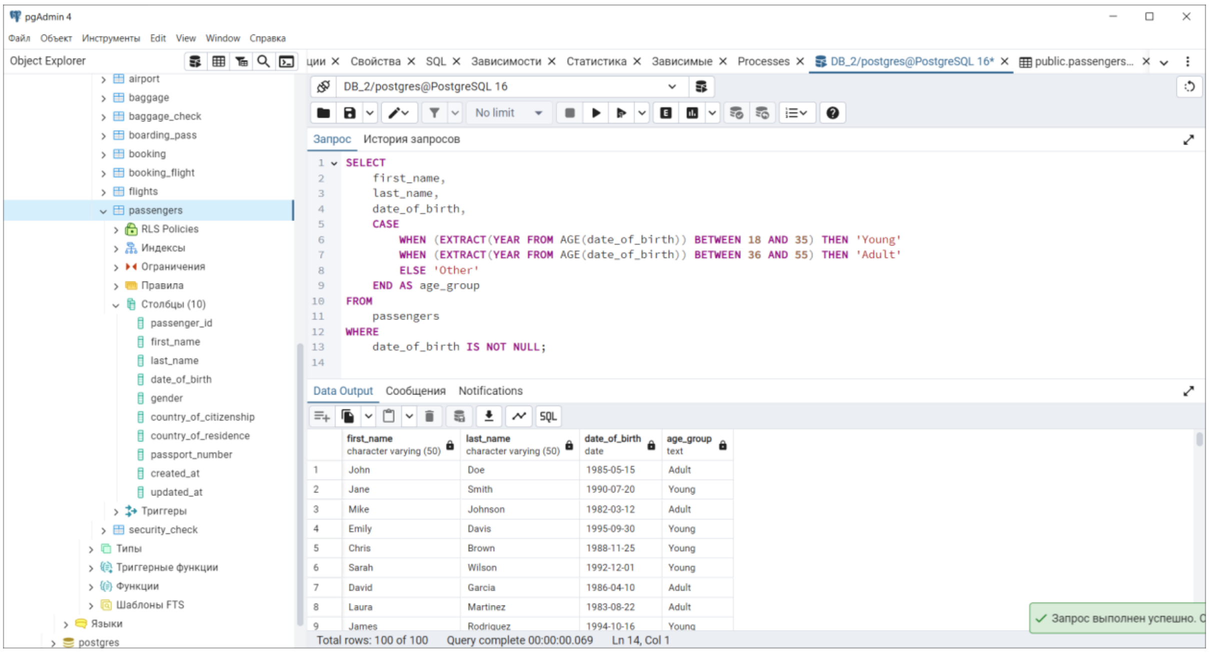


```
SELECT flight_id
FROM flights
WHERE act_arrival_time > sch_arrival_time;
```

	flight_id [PK] integer
1	1001
2	1002
3	1003
4	1004
5	1005
6	1006
7	1007
8	1008
9	1009

Total rows: 100 of 100 Query complete 00:00:00.061 Ln 4, Col 1

7. Create a query that divides passengers into age groups like ‘Young’ and ‘Adult’ based on their birth date. Young passengers age between 18 and 35, Adult passengers age between 36 and 55.



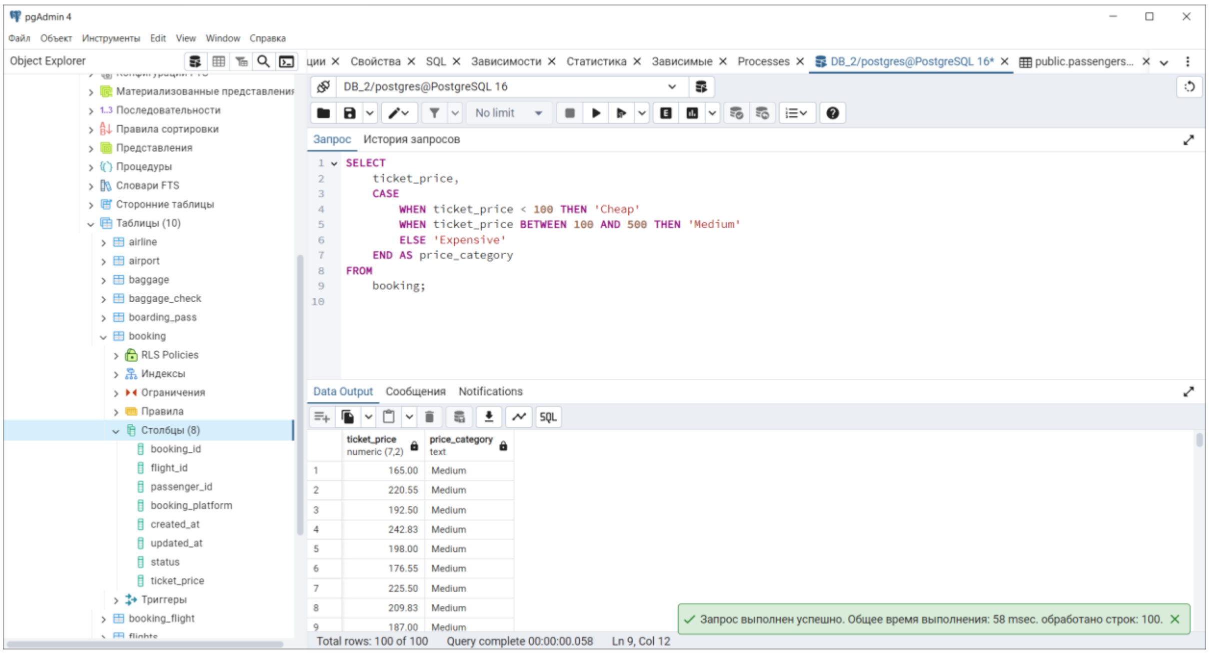
```
SELECT
    first_name,
    last_name,
    date_of_birth,
    CASE
        WHEN (EXTRACT(YEAR FROM AGE(date_of_birth)) BETWEEN 18 AND 35) THEN 'Young'
        WHEN (EXTRACT(YEAR FROM AGE(date_of_birth)) BETWEEN 36 AND 55) THEN 'Adult'
        ELSE 'Other'
    END AS age_group
FROM
    passengers
WHERE
    date_of_birth IS NOT NULL;
```

	first_name character varying (50)	last_name character varying (50)	date_of_birth date	age_group text
1	John	Doe	1985-05-15	Adult
2	Jane	Smith	1990-07-20	Young
3	Mike	Johnson	1982-03-12	Adult
4	Emily	Davis	1995-09-30	Young
5	Chris	Brown	1988-11-25	Young
6	Sarah	Wilson	1992-12-01	Young
7	David	Garcia	1986-04-10	Adult
8	Laura	Martinez	1983-08-22	Adult
9	James	Rodriguez	1994-10-16	Young

✓ Запрос выполнен успешно. О

Total rows: 100 of 100 Query complete 00:00:00.069 Ln 14, Col 1

8. Create a query that categorizes ticket prices based on their price as "Cheap," "Medium" or "Expensive."



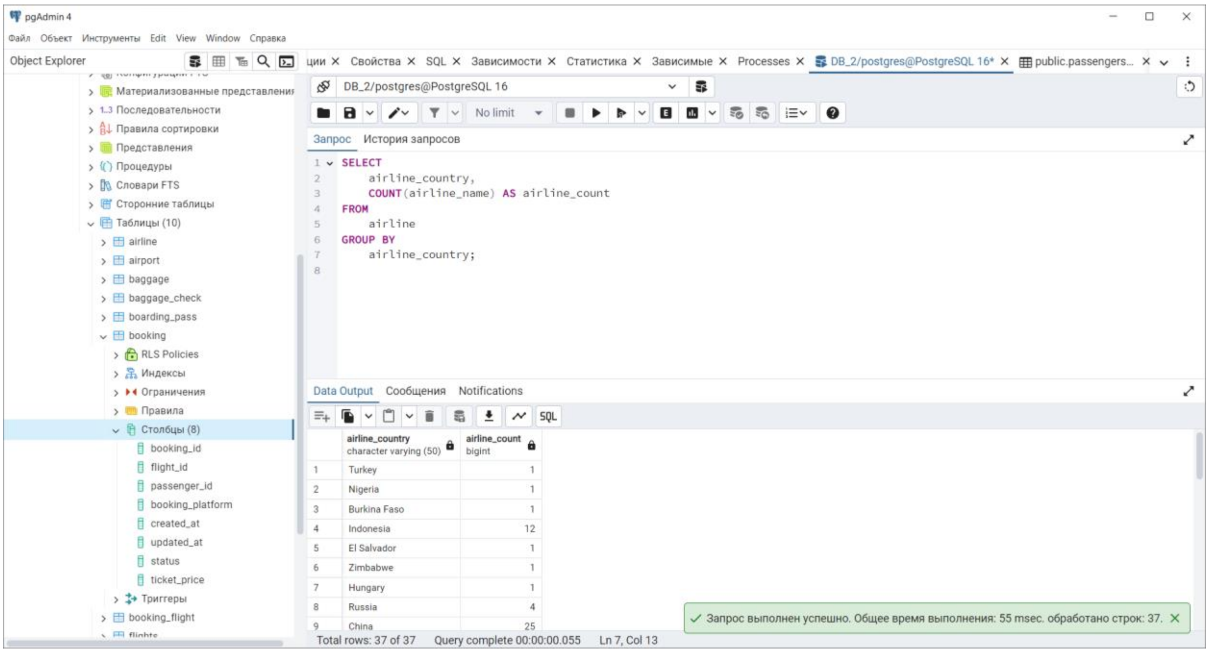
```
SELECT
    ticket_price,
    CASE
        WHEN ticket_price < 100 THEN 'Cheap'
        WHEN ticket_price BETWEEN 100 AND 500 THEN 'Medium'
        ELSE 'Expensive'
    END AS price_category
FROM
    booking;
```

	ticket_price numeric (7,2)	price_category text
1	165.00	Medium
2	220.55	Medium
3	192.50	Medium
4	242.83	Medium
5	198.00	Medium
6	176.55	Medium
7	225.50	Medium
8	209.83	Medium
9	187.00	Medium

✓ Запрос выполнен успешно. Общее время выполнения: 58 msec. обработано строк: 100.

Total rows: 100 of 100 Query complete 00:00:00.058 Ln 9, Col 12

9. Find number of airline names in each airline country.



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

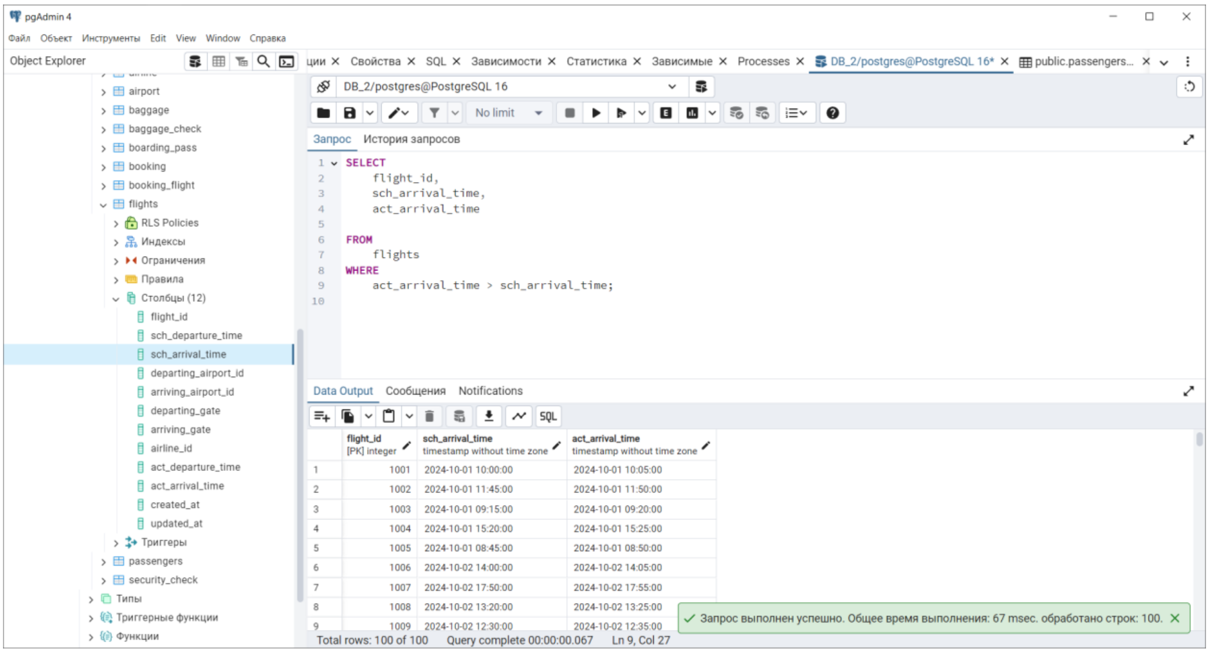
	airline_country character varying (50)	airline_count bigint
1	Turkey	1
2	Nigeria	1
3	Burkina Faso	1
4	Indonesia	12
5	El Salvador	1
6	Zimbabwe	1
7	Hungary	1
8	Russia	4
9	China	25

✓ Запрос выполнен успешно. Общее время выполнения: 55 msec. обработано строк: 37.

Total rows: 37 of 37 Query complete 00:00:00.055 Ln 7, Col 13

10. Find flights that arrived late according to their actual arrival time compared to the scheduled arrival time.

s



```
SELECT
    flight_id,
    sch_arrival_time,
    act_arrival_time

FROM
    flights
WHERE
    act_arrival_time > sch_arrival_time;
```

	flight_id [PK] integer	sch_arrival_time timestamp without time zone	act_arrival_time timestamp without time zone
1	1001	2024-10-01 10:00:00	2024-10-01 10:05:00
2	1002	2024-10-01 11:45:00	2024-10-01 11:50:00
3	1003	2024-10-01 09:15:00	2024-10-01 09:20:00
4	1004	2024-10-01 15:20:00	2024-10-01 15:25:00
5	1005	2024-10-01 08:45:00	2024-10-01 08:50:00
6	1006	2024-10-02 14:00:00	2024-10-02 14:05:00
7	1007	2024-10-02 17:50:00	2024-10-02 17:55:00
8	1008	2024-10-02 13:20:00	2024-10-02 13:25:00
9	1009	2024-10-02 12:30:00	2024-10-02 12:35:00

✓ Запрос выполнен успешно. Общее время выполнения: 67 msec. обработано строк: 100.

Total rows: 100 of 100 Query complete 00:00:00.067 Ln 9, Col 27