Kazakh British Technical University

Databases

Laboratory work №4

Kislitsin Alexandr

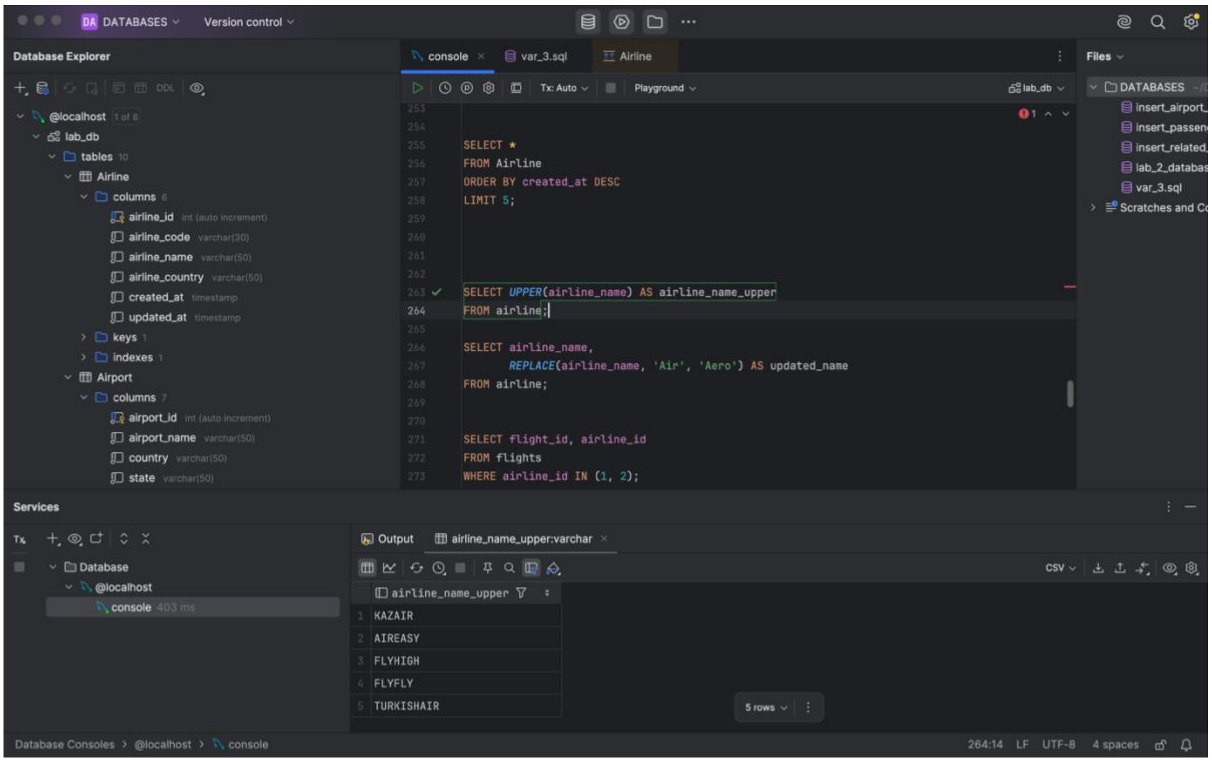
23B031861

Introduction

In this lab, we continued to study SQL queries using the airport database as an example.

The main goal is to learn how to use string functions, conditional expressions, and comparison operators, as well as apply data formatting, filtering, and grouping in queries.

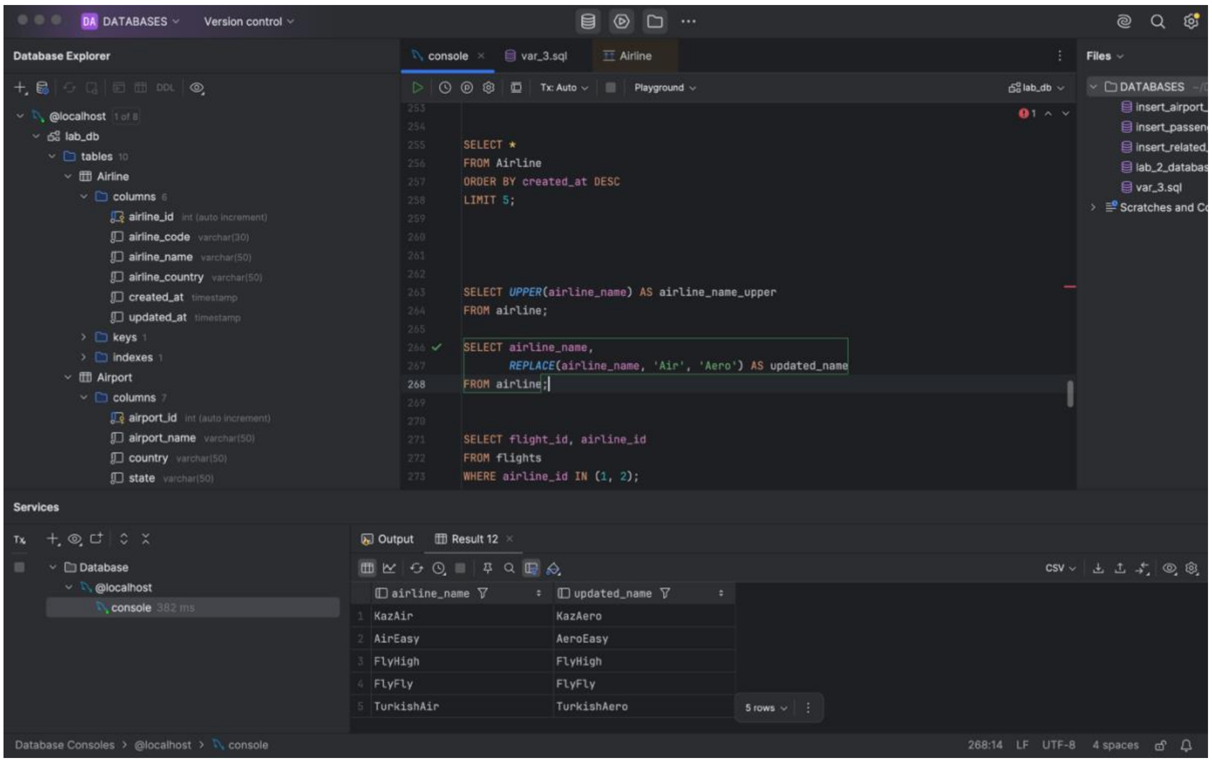
Task 1. Retrieve all airline names in uppercase



```sql
SELECT *
FROM Airline
ORDER BY created_at DESC
LIMIT 5;



SELECT UPPER(airline_name) AS airline_name_upper
FROM airline;

SELECT airline_name,
       REPLACE(airline_name, 'Air', 'Aero') AS updated_name
FROM airline;


SELECT flight_id, airline_id
FROM flights
WHERE airline_id IN (1, 2);
```

	airline_name_upper
1	KAZAIR
2	AIREASY
3	FLYHIGH
4	FLYFLY
5	TURKISHAIR

Task 2. Replace any occurrence of the word “Air” with “Aero”

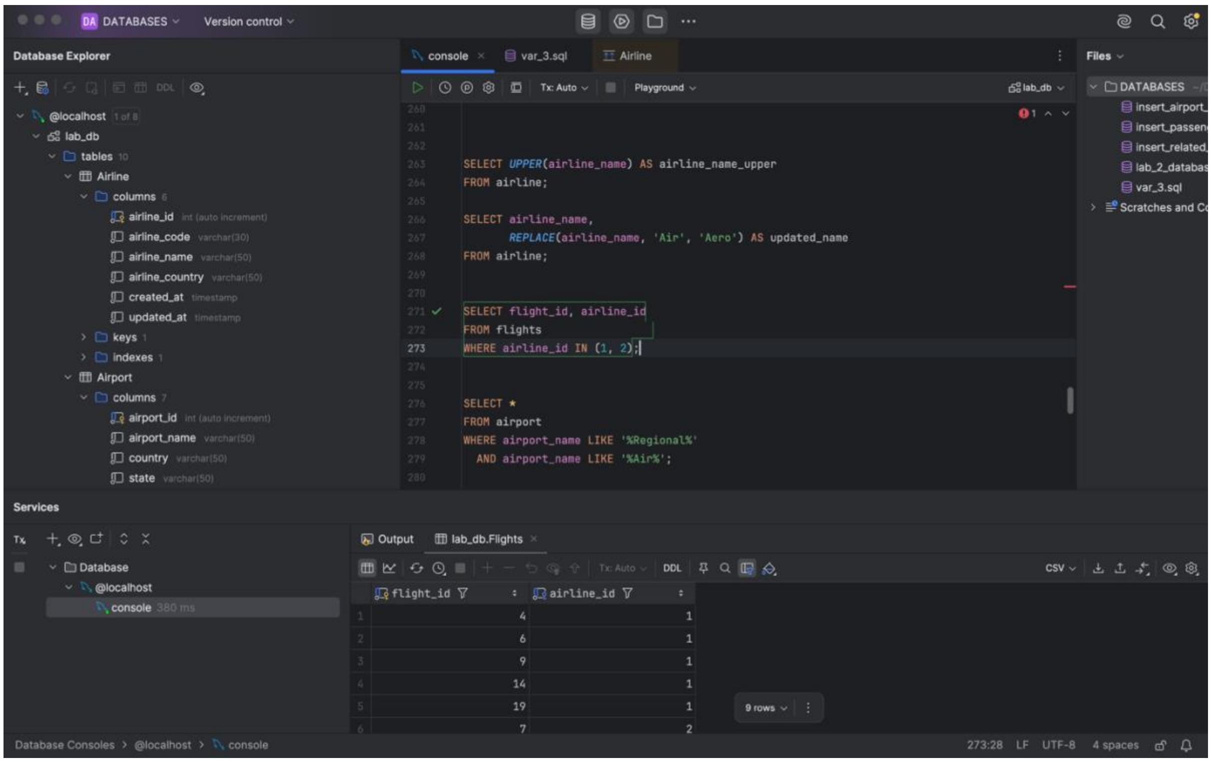


```sql
SELECT *
FROM Airline
ORDER BY created_at DESC
LIMIT 5;



SELECT UPPER(airline_name) AS airline_name_upper
FROM airline;

SELECT airline_name,
       REPLACE(airline_name, 'Air', 'Aero') AS updated_name
FROM airline;


SELECT flight_id, airline_id
FROM flights
WHERE airline_id IN (1, 2);
```

	airline_name	updated_name
1	KazAir	KazAero
2	AirEasy	AeroEasy
3	FlyHigh	FlyHigh
4	FlyFly	FlyFly
5	TurkishAir	TurkishAero

5 rows

Task 3. Find all flight numbers that coordinate with both airline 1 and airline

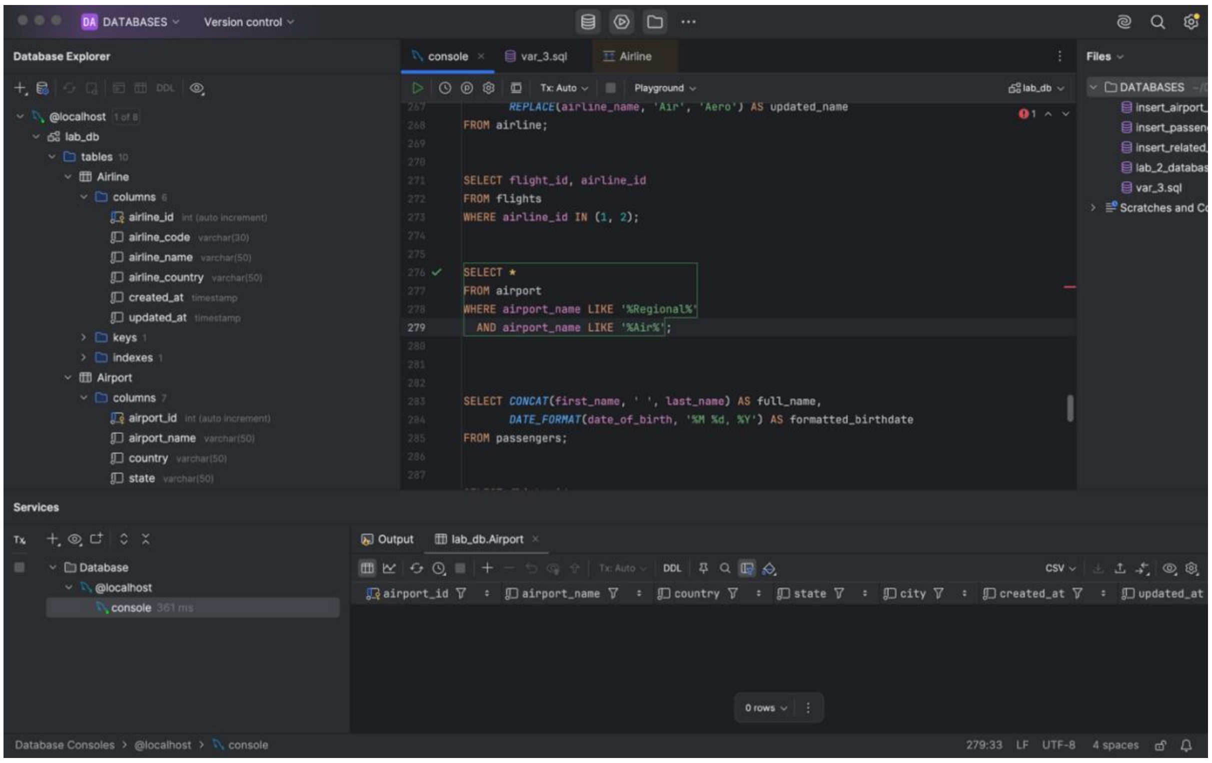


```sql
SELECT UPPER(airline_name) AS airline_name_upper
FROM airline;

SELECT airline_name,
       REPLACE(airline_name, 'Air', 'Aero') AS updated_name
FROM airline;


SELECT flight_id, airline_id
FROM flights
WHERE airline_id IN (1, 2);


SELECT *
FROM airport
WHERE airport_name LIKE '%Regional%'
  AND airport_name LIKE '%Air%';
```

```
   flight_id  airline_id
1          4           1
2          6           1
3          9           1
4         14           1
5         19           1
6          7           2
```

9 rows

Task 4. Retrieve airports that contain “Regional” and “Air” in their names

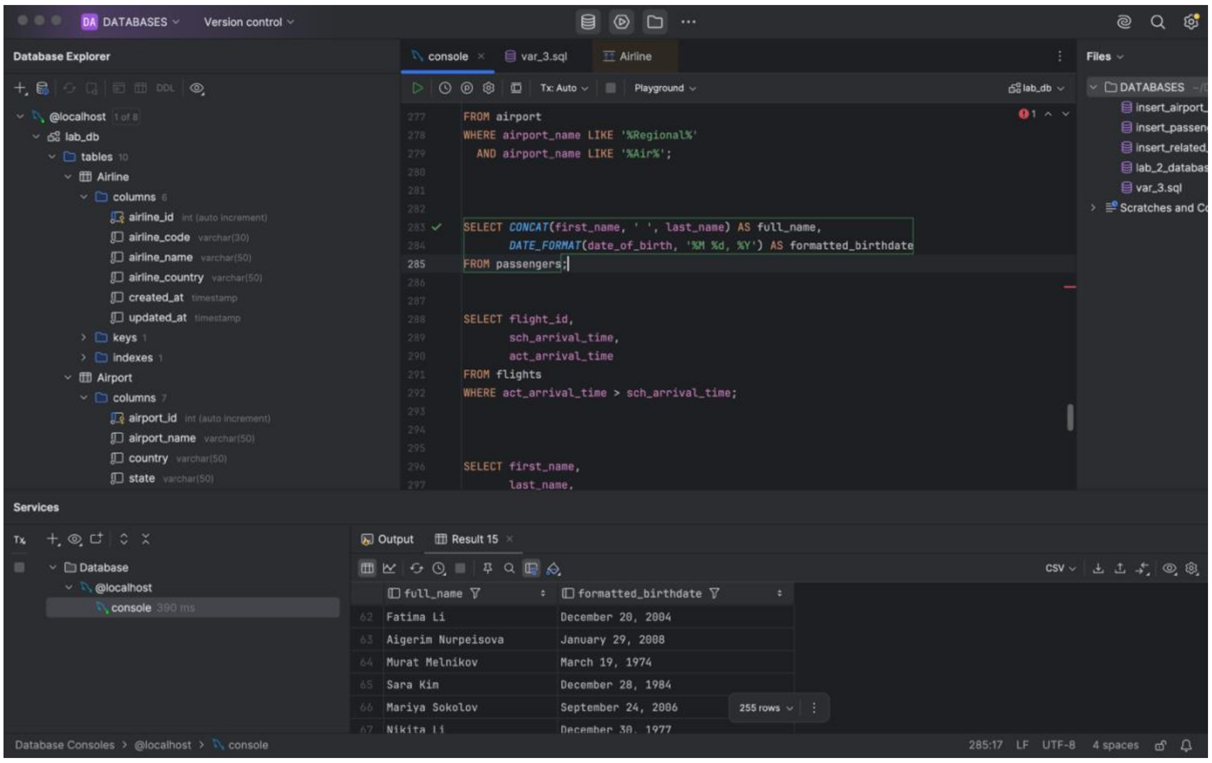


```sql
        REPLACE(airline_name, 'Air', 'Aero') AS updated_name
FROM airline;


SELECT flight_id, airline_id
FROM flights
WHERE airline_id IN (1, 2);


SELECT *
FROM airport
WHERE airport_name LIKE '%Regional%'
  AND airport_name LIKE '%Air%';



SELECT CONCAT(first_name, ' ', last_name) AS full_name,
       DATE_FORMAT(date_of_birth, '%M %d, %Y') AS formatted_birthdate
FROM passengers;
```

airport_id	airport_name	country	state	city	created_at	updated_at

0 rows

Task 5. Retrieve passenger names and format their birth dates as ‘Month DD, YYYY’

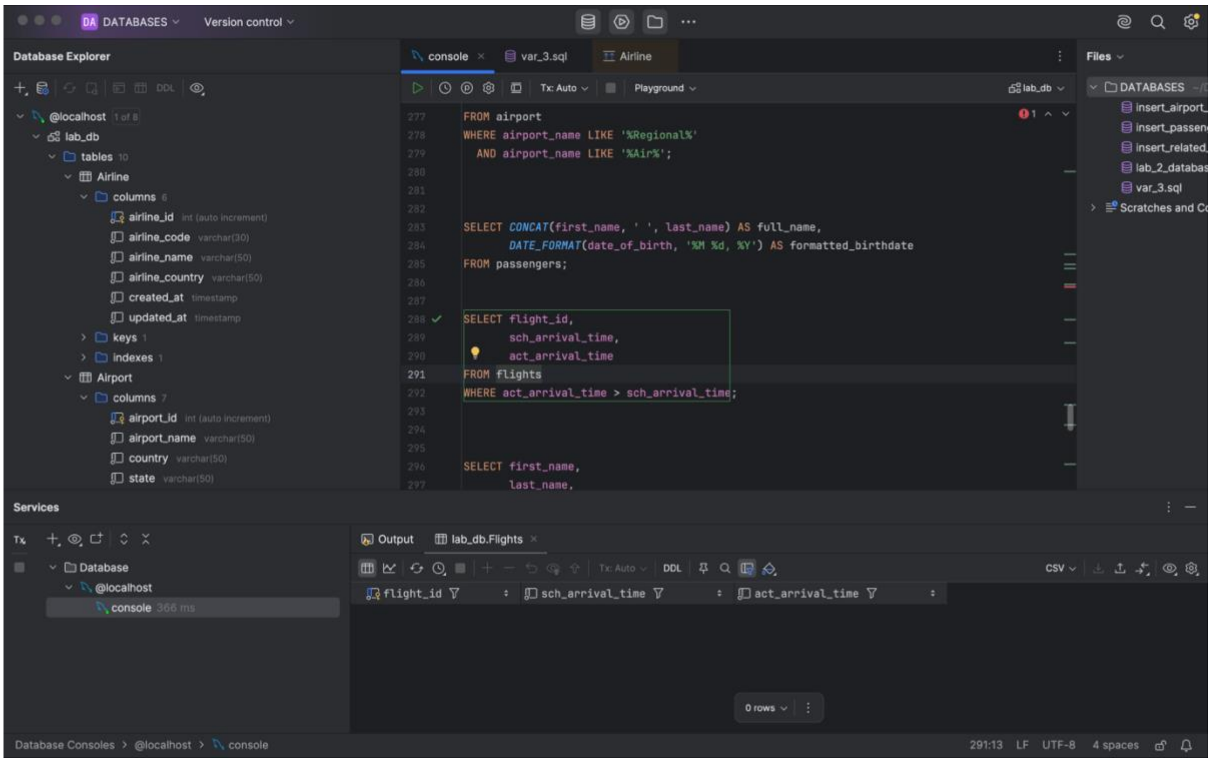


```sql
FROM airport
WHERE airport_name LIKE '%Regional%'
  AND airport_name LIKE '%Air%';


SELECT CONCAT(first_name, ' ', last_name) AS full_name,
       DATE_FORMAT(date_of_birth, '%M %d, %Y') AS formatted_birthdate
FROM passengers;


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



SELECT first_name,
       last_name,
```

	full_name	formatted_birthdate
62	Fatima Li	December 20, 2004
63	Aigerim Nurpeisova	January 29, 2008
64	Murat Melnikov	March 19, 1974
65	Sara Kim	December 28, 1984
66	Mariya Sokolov	September 24, 2006
67	Nikita Li	December 30, 1977

255 rows

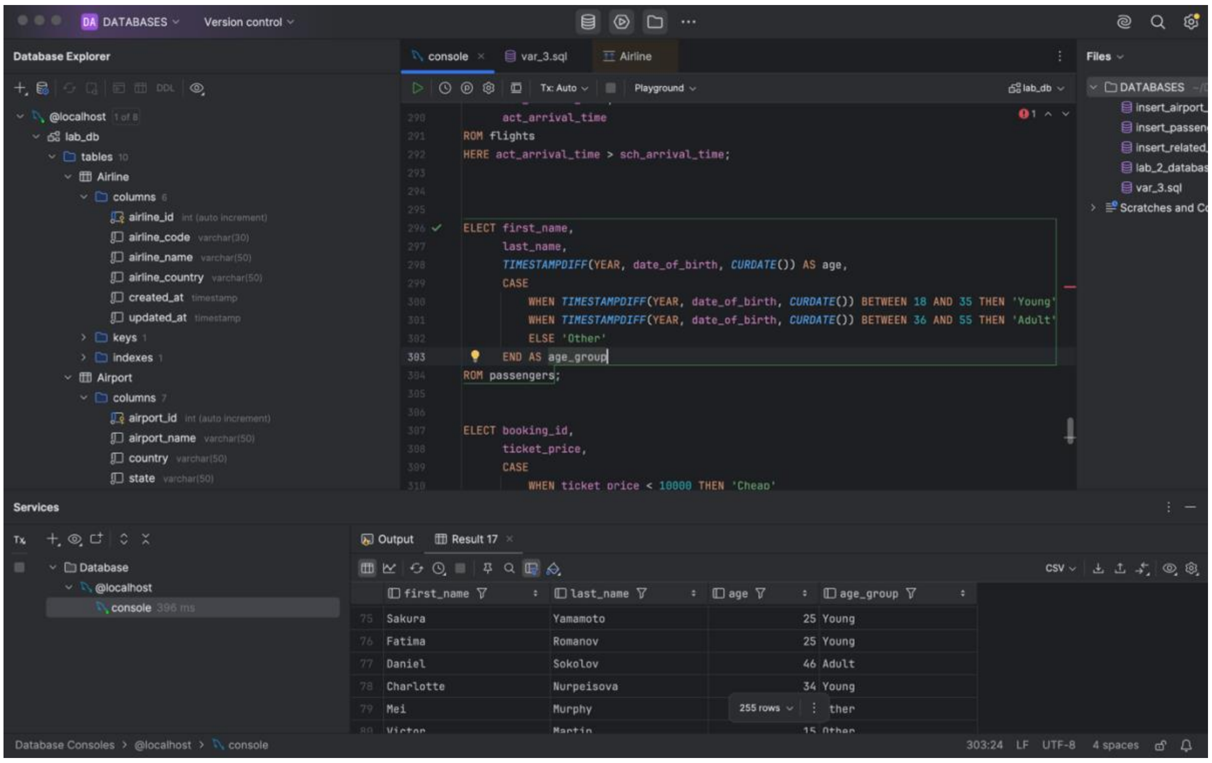
Task 6. Find flight numbers that have been delayed



```
FROM airport
WHERE airport_name LIKE '%Regional%'
  AND airport_name LIKE '%Air%';



SELECT CONCAT(first_name, ' ', last_name) AS full_name,
       DATE_FORMAT(date_of_birth, '%M %d, %Y') AS formatted_birthdate
FROM passengers;


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



SELECT first_name,
       last_name,
```

```
flight_id | sch_arrival_time | act_arrival_time

0 rows
```

Task 7. Divide passengers into age groups (Young / Adult)



```
@localhost 1 of 8
  lab_db
    tables 10
      Airline
        columns 6
          airline_id int (auto increment)
          airline_code varchar(30)
          airline_name varchar(50)
          airline_country varchar(50)
          created_at timestamp
          updated_at timestamp
        keys 1
        indexes 1
      Airport
        columns 7
          airport_id int (auto increment)
          airport_name varchar(50)
          country varchar(50)
          state varchar(50)
```

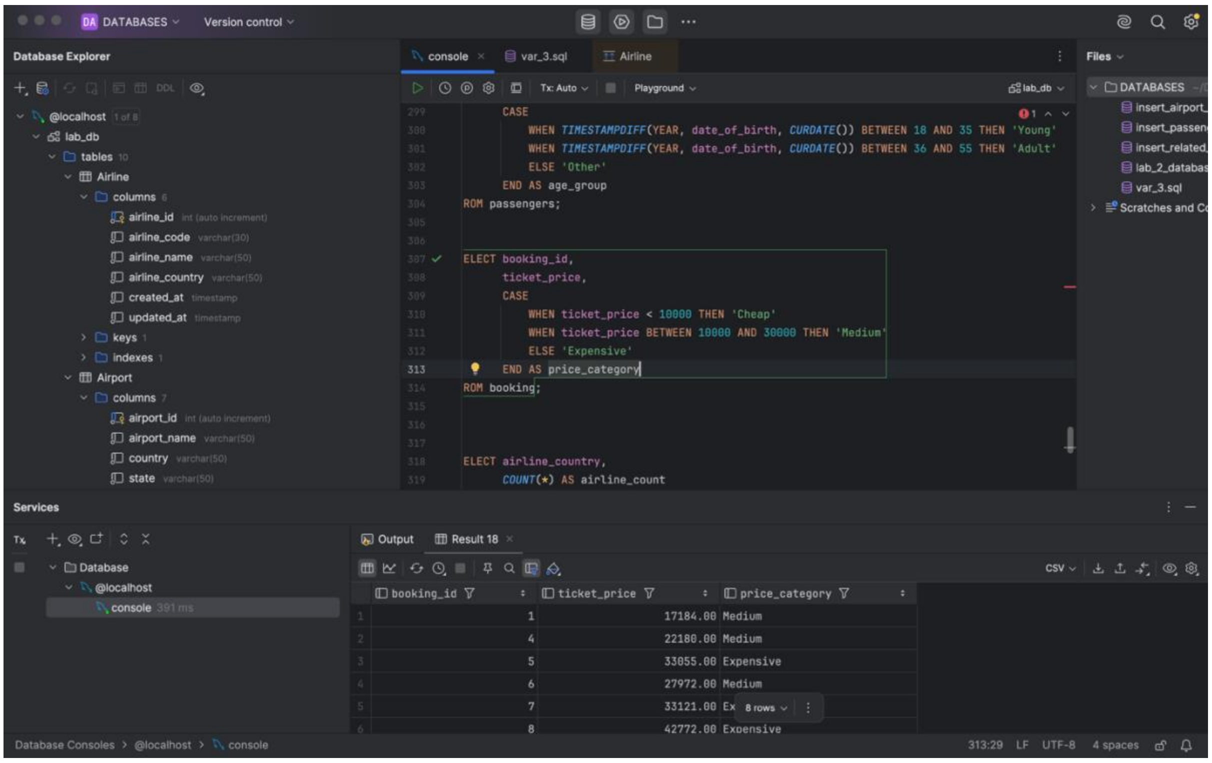
```sql
       act_arrival_time
ROM flights
HERE act_arrival_time > sch_arrival_time;


ELECT first_name,
       last_name,
       TIMESTAMPDIFF(YEAR, date_of_birth, CURDATE()) AS age,
       CASE
           WHEN TIMESTAMPDIFF(YEAR, date_of_birth, CURDATE()) BETWEEN 18 AND 35 THEN 'Young'
           WHEN TIMESTAMPDIFF(YEAR, date_of_birth, CURDATE()) BETWEEN 36 AND 55 THEN 'Adult'
           ELSE 'Other'
       END AS age_group
ROM passengers;


ELECT booking_id,
       ticket_price,
       CASE
           WHEN ticket_price < 10000 THEN 'Cheap'
```

Result 17

	first_name	last_name	age	age_group
75	Sakura	Yamamoto	25	Young
76	Fatima	Romanov	25	Young
77	Daniel	Sokolov	46	Adult
78	Charlotte	Nurpeisova	34	Young
79	Mei	Murphy		ther
80	Victor	Martin	15	Other

255 rows

Task 8. Categorize ticket prices as Cheap, Medium or Expensive

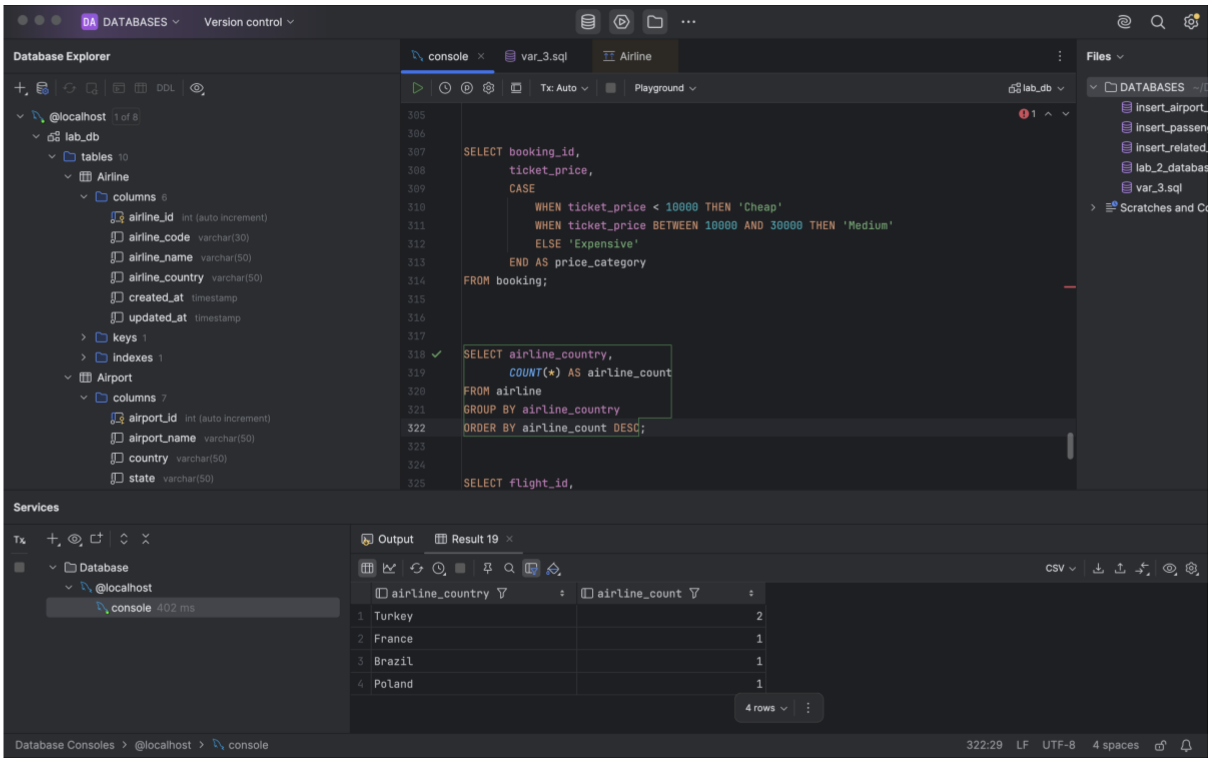


```sql
    CASE
        WHEN TIMESTAMPDIFF(YEAR, date_of_birth, CURDATE()) BETWEEN 18 AND 35 THEN 'Young'
        WHEN TIMESTAMPDIFF(YEAR, date_of_birth, CURDATE()) BETWEEN 36 AND 55 THEN 'Adult'
        ELSE 'Other'
    END AS age_group
ROM passengers;


ELECT booking_id,
    ticket_price,
    CASE
        WHEN ticket_price < 10000 THEN 'Cheap'
        WHEN ticket_price BETWEEN 10000 AND 30000 THEN 'Medium'
        ELSE 'Expensive'
    END AS price_category
ROM booking;



ELECT airline_country,
    COUNT(*) AS airline_count
```

	booking_id	ticket_price	price_category
1	1	17184.00	Medium
2	4	22180.00	Medium
3	5	33055.00	Expensive
4	6	27972.00	Medium
5	7	33121.00	Ex
6	8	42772.00	Expensive

8 rows

Task 9. Find number of airline names in each country

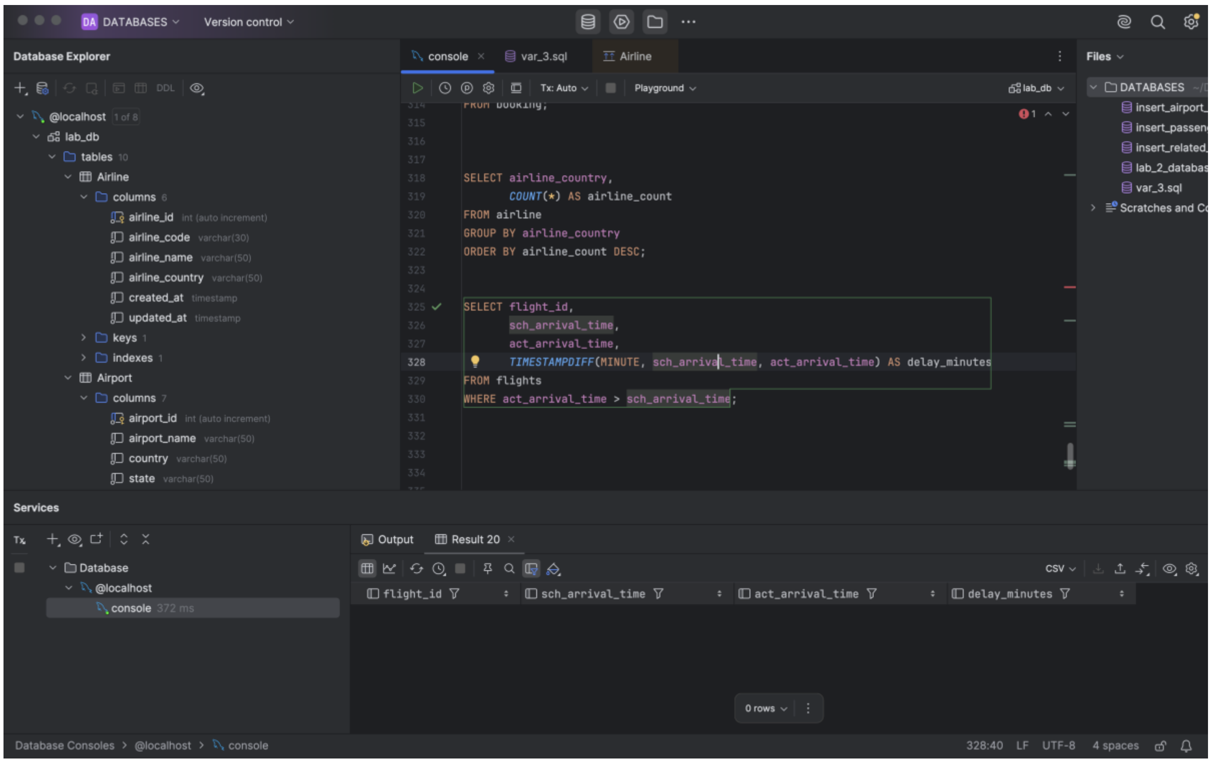


```
SELECT booking_id,
       ticket_price,
       CASE
           WHEN ticket_price < 10000 THEN 'Cheap'
           WHEN ticket_price BETWEEN 10000 AND 30000 THEN 'Medium'
           ELSE 'Expensive'
       END AS price_category
FROM booking;



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


SELECT flight_id,
```

	airline_country	airline_count
1	Turkey	2
2	France	1
3	Brazil	1
4	Poland	1

4 rows

Task 10. Find flights that arrived late



```
FROM booking;



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


SELECT flight_id,
       sch_arrival_time,
       act_arrival_time,
       TIMESTAMPDIFF(MINUTE, sch_arrival_time, act_arrival_time) AS delay_minutes
FROM flights
WHERE act_arrival_time > sch_arrival_time;
```

```
flight_id | sch_arrival_time | act_arrival_time | delay_minutes
```

0 rows

Conclusion

During the laboratory work, queries were performed using the string, date, and condition functions.

The skills of formatting texts and dates, using CASE, GROUP BY, LIKE, UPPER, REPLACE, and DATE_FORMAT are fixed.

The result was the ability to create more flexible and informative SQL queries for airport data analysis.