

Lab 6 database. Abdykamat Adilet

1. Write a query that displays all flights of a specific airline.

```
SELECT * FROM flights
```

```
WHERE airline_id = (SELECT airline_id FROM airline WHERE airline_name = 'Air Astana' )
```

The screenshot shows a PostgreSQL query editor interface. The top bar indicates the connection is to 'lab2/postgres@MyServer'. Below the toolbar, the 'Query' tab is active, showing the following SQL query:

```
1 SELECT * FROM flights
2 WHERE airline_id = (SELECT airline_id FROM airline WHERE airline_name = 'Air Astana' )
```

The 'Data Output' tab is also visible, showing a table with the following columns and data types:

| flight_id | sch_departure_time | sch_arrival_time | departing_airport_id | arriving_airport_id | departing_gate | arriving_gate |
|--------------|-----------------------------|-----------------------------|----------------------|---------------------|----------------|------------------------|
| [PK] integer | timestamp without time zone | timestamp without time zone | integer | integer | text | character varying (50) |

At the bottom, a status bar indicates 'Total rows: 0' and 'Query complete 00:00:00.123'. A green message box states: 'Successfully run. Total query runtime: 123 msec. 0 rows affected.'

2. Compose a query to obtain a list of all flights with the names of departure airports.

```
SELECT f.flight_id, a.airline_name, dep.airport_name AS dep_airport
FROM flights f
JOIN airline a
    ON f.airline_id = a.airline_id
JOIN airport dep
    ON f.departing_airport_id = dep.airport_id;
```

Query Query History

```

1 SELECT f.flight_id, a.airline_name, dep.airport_name AS dep_airport
2 FROM flights f
3 JOIN airline a
4     ON f.airline_id = a.airline_id
5 JOIN airport dep
6     ON f.departing_airport_id = dep.airport_id;

```

Data Output Messages Notifications

Showing rows: 1 to 20 Page No: 1 of 1

| | flight_id integer | airline_name character varying (50) | dep_airport character varying (50) |
|---|----------------------|--|---------------------------------------|
| 1 | 1 | KazAir | Astana Regional Air Center |
| 2 | 2 | AirEasy | Airport_2 Airport |
| 3 | 3 | FlyHigh | Shymkent Regional Airport |
| 4 | 4 | KazAir | Airport_4 Airport |
| 5 | 5 | FlyFly | Airport_5 Airport |
| 6 | 6 | uibv | Airport_6 Airport |
| 7 | 7 | FlyHigh | Taraz Regional Air Base |
| 8 | 8 | AirEasy | Pavlodar Regional Air Port |
| 9 | 9 | KazAir | Kvzvlorda Regional Air Station |

Successfully run. Total query runtime: 133 msec. 20 rows affected.

3. Create a query that finds all airlines that have no flights scheduled for the next month.

SELECT a.airline_name FROM airline a

WHERE airline_id NOT IN(

SELECT DISTINCT f.airline_id FROM flights f

WHERE f.sch_departure_time BETWEEN DATE_TRUNC('month',CURRENT_DATE + INTERVAL '1 month')

AND DATE_TRUNC('month',CURRENT_DATE + INTERVAL '2 month')

);

lab2/postgres@MyServer

Query Query History

```

1 SELECT a.airline_name FROM airline a
2 WHERE airline_id NOT IN(
3     SELECT DISTINCT f.airline_id FROM flights f
4     WHERE f.sch_departure_time BETWEEN DATE_TRUNC('month',CURRENT_DATE + INTERVAL '1 month')
5     AND DATE_TRUNC('month',CURRENT_DATE + INTERVAL '2 month')
6 );

```

Data Output Messages Notifications

Showing rows: 1 to 24 Page No: 1 of 1

| | airline_name character varying (50) |
|---|--|
| 1 | KazAir |
| 2 | AirEasy |
| 3 | FlyHigh |
| 4 | FlyFly |
| 5 | uibv |
| 6 | Paris Wings |
| 7 | Lyon Air |
| 8 | Lisboa Sky |
| 9 | Porto Airlines |

Successfully run. Total query runtime: 161 msec. 24 rows affected.

4. Create a query to display a list of passengers on a specific flight.

```
SELECT p.first_name, p.last_name, p.passenger_id FROM passengers p  
JOIN tickets t ON p.passenger_id = t.passenger_id  
WHERE t.flight_number = 'SU1001';
```

The screenshot shows a database query editor with a 'Query' tab. The query text is as follows:

```
1 SELECT p.first_name, p.last_name, p.passenger_id FROM passengers p  
2 JOIN tickets t ON p.passenger_id = t.passenger_id  
3 WHERE t.flight_number = 'SU1001';
```

Below the query editor, the 'Data Output' tab is active, displaying the results of the query in a table. The table has three columns: first_name, last_name, and passenger_id. The results are as follows:

| | first_name character varying (50) | last_name character varying (50) | passenger_id [PK] integer |
|---|--------------------------------------|-------------------------------------|------------------------------|
| 1 | Иван | Иван | 1 |
| 2 | Елена | Козлова | 4 |
| 3 | Михаил | Попов | 9 |
| 4 | Михаил | Попов | 9 |
| 5 | Юлия | Юлия | 16 |

At the bottom right of the interface, a green status bar indicates: 'Successfully run. Total query runtime: 180 msec. 5 rows affected.'

5. Write a query that calculates the average, total, maximum and minimum price of tickets for each flight.

```
SELECT flight_number,  
       AVG(price) as Average_price,  
       MAX(price) as Maximum_price,  
       MIN(price) as Minimum_price,  
       SUM(price) as Total_price  
FROM tickets  
GROUP BY flight_number;
```

| Query | | Query History | | |
|-------|------------|-------------------|--|--|
| 1 | SELECT | flight_number, | | |
| 2 | AVG(price) | as Average_price, | | |
| 3 | MAX(price) | as Maximum_price, | | |
| 4 | MIN(price) | as Minimum_price, | | |
| 5 | SUM(price) | as Total_price | | |
| 6 | FROM | tickets | | |
| 7 | GROUP BY | flight_number; | | |

| Data Output | | Messages | Notifications | | |
|------------------------|---------------|--------------------|---------------|-------------|-----------|
| Showing rows: 1 to 8 | | Page No: 1 | of 1 | | |
| flight_number | average_price | maximum_price | minimum_price | total_price | |
| character varying (10) | numeric | numeric | numeric | numeric | |
| 1 | SU1005 | 20125.000000000000 | 28000.00 | 14500.00 | 80500.00 |
| 2 | SU1008 | 15500.000000000000 | 15500.00 | 15500.00 | 15500.00 |
| 3 | SU1006 | 29666.666666666667 | 33000.00 | 24000.00 | 89000.00 |
| 4 | SU1007 | 24166.666666666667 | 27000.00 | 22500.00 | 72500.00 |
| 5 | SU1002 | 24600.000000000000 | 31000.00 | 12000.00 | 123000.00 |
| 6 | SU1001 | 19300.000000000000 | 27500.00 | 13000.00 | 96500.00 |
| 7 | SU1004 | 19750.000000000000 | 30000.00 | 14000.00 | 79000.00 |
| 8 | SU1003 | 20200.000000000000 | 35000.00 | 12500.00 | 87700.00 |

| | | | | | |
|---------------|-----------------------------|-------------------|--|--|--|
| Total rows: 8 | Query complete 00:00:00.172 | CRLF Ln 7, Col 24 | | | |
|---------------|-----------------------------|-------------------|--|--|--|

6. Create a query that shows all flights flying to a specific country by combining flights, airports and airline, and using the condition on the country name.

```
SELECT f.flight_id , a.airline_name , dep.airport_name , arr.airport_name, arr.country
FROM flights f
JOIN airline a ON f.airline_id = a.airline_id
JOIN airport dep ON f.departing_airport_id = dep.airport_id
JOIN airport arr ON f.arriving_airport_id = arr.airport_id
WHERE arr.country = 'Kazakhstan';
```

The screenshot shows a SQL query editor with a toolbar at the top containing icons for file operations, query execution, and settings. The query text is as follows:

```
1 SELECT f.flight_id , a.airline_name , dep.airport_name , arr.airport_name, arr.country
2 FROM flights f
3 JOIN airline a ON f.airline_id = a.airline_id
4 JOIN airport dep ON f.departing_airport_id = dep.airport_id
5 JOIN airport arr ON f.arriving_airport_id = arr.airport_id
6 WHERE arr.country = 'Kazakhstan';
```

Below the query editor is a 'Data Output' tab showing the schema of the result set:

| flight_id | airline_name | airport_name | airport_name | country |
|-----------|------------------------|------------------------|------------------------|------------------------|
| integer | character varying (50) | character varying (50) | character varying (50) | character varying (50) |

A status message at the bottom right indicates: 'Successfully run. Total query runtime: 154 msec. 0 rows affected.'

7. Display a list of minor passengers and their arrival destination.

SELECT

p.first_name,

p.last_name,

arr.city AS arrival_city

FROM passengers p

JOIN tickets t ON p.passenger_id = t.passenger_id

JOIN booking b ON t.passenger_id = b.passenger_id

JOIN booking_flight bf ON b.booking_id = bf.booking_id

JOIN flights f ON bf.flight_id = f.flight_id

JOIN airport arr ON f.arriving_airport_id = arr.airport_id

WHERE AGE(p.date_of_birth) < INTERVAL '18 years';

Query Query History

```

1 SELECT
2     p.first_name,
3     p.last_name,
4     arr.city AS arrival_city
5 FROM passengers p
6 JOIN tickets t ON p.passenger_id = t.passenger_id
7 JOIN booking b ON t.passenger_id = b.passenger_id
8 JOIN booking_flight bf ON b.booking_id = bf.booking_id
9 JOIN flights f ON bf.flight_id = f.flight_id
10 JOIN airport arr ON f.arriving_airport_id = arr.airport_id
11 WHERE AGE(p.date_of_birth) < INTERVAL '18 years';

```

Data Output Messages Notifications

| first_name | last_name | arrival_city |
|------------------------|------------------------|------------------------|
| character varying (50) | character varying (50) | character varying (50) |

✓ Successfully run. Total query runtime: 70 msec. 0 rows affected. ✕

Total rows: 0 Query complete 00:00:00.070

8. Display the passenger's full name, passport number, and the passenger's current time of arrival at the destination.

```

SELECT p.first_name ,p.last_name, p.passport_number, f.sch_arrival_time FROM passengers p
JOIN tickets t ON p.passenger_id = t.passenger_id
JOIN booking b ON t.passenger_id = b.passenger_id
JOIN booking_flight bf ON b.booking_id = bf.booking_id
JOIN flights f ON bf.flight_id = f.flight_id;

```

Query Query History

```

1 SELECT p.first_name ,p.last_name, p.passport_number, f.sch_arrival_time FROM passengers p
2 JOIN tickets t ON p.passenger_id = t.passenger_id
3 JOIN booking b ON t.passenger_id = b.passenger_id
4 JOIN booking_flight bf ON b.booking_id = bf.booking_id
5 JOIN flights f ON bf.flight_id = f.flight_id;

```

Data Output Messages Notifications

| first_name | last_name | passport_number | sch_arrival_time |
|------------------------|------------------------|------------------------|-----------------------------|
| character varying (50) | character varying (50) | character varying (20) | timestamp without time zone |

✓ Successfully run. Total query runtime: 66 msec. 0 rows affected. ✕

Total rows: 0 Query complete 00:00:00.066

9. Print a list of flights where the airline's home country and origin country are the same. Group them by the airport country.

```
SELECT f.flight_id, a.airline_name, dep.country FROM flights f
JOIN airline a ON f.airline_id = a.airline_id
JOIN airport dep ON f.departing_airport_id = dep.airport_id
WHERE a.airline_country = dep.country
GROUP BY f.flight_id, a.airline_name, dep.country;
```

The screenshot shows a SQL query editor with a query history tab. The query is as follows:

```
1 SELECT f.flight_id, a.airline_name, dep.country FROM flights f
2 JOIN airline a ON f.airline_id = a.airline_id
3 JOIN airport dep ON f.departing_airport_id = dep.airport_id
4 WHERE a.airline_country = dep.country
5 GROUP BY f.flight_id, a.airline_name, dep.country;
```

Below the query editor is a 'Data Output' tab. It shows a table with three columns: flight_id (integer), airline_name (character varying (50)), and country (character varying (50)). The table is currently empty.

At the bottom right, a green status bar indicates: 'Successfully run. Total query runtime: 68 msec. 0 rows affected.'