

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 the pgAdmin 4 interface. At the top, there's a connection bar with 'lab2/postgres@MyServer'. Below it is a toolbar with various icons. The main area has tabs for 'Query' (which is selected), 'Query History', and 'Notifications'. The 'Query' tab contains the following SQL code:

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

Below the code is a 'Data Output' tab which is currently inactive. At the bottom of the interface, there's a status bar with 'Total rows: 0' and 'Query complete 00:00:00.123'. To the right of the status bar, a green message box says 'Successfully run. Total query runtime: 123 msec. 0 rows affected.' with a close button.

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 History

```

1 v 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

	flight_id	airline_name	dep_airport
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	Kyzylorda Regional Air Station

Showing rows: 1 to 20 | Page No: 1 of 1 | Back | Next | SQL

✓ 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')
```

```
        );
```

Query History

```

1 v 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

	airline_name
1	KazAir
2	AirEasy
3	FlyHigh
4	FlyFly
5	uibv
6	Paris Wings
7	Lyon Air
8	Lisboa Sky
9	Porto Airlines

Showing rows: 1 to 24 | Page No: 1 of 1 | Back | Next | SQL

✓ 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 interface with a query editor and a results grid. The query editor at the top contains the following SQL code:

```
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';
```

The results grid below displays the following data:

	first_name	last_name	passenger_id
1	Иван	Иван	1
2	Елена	Козлова	4
3	Михаил	Попов	9
4	Михаил	Попов	9
5	Юлия	Юлия	16

A message bar at the bottom right 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 History

```

1 v 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

	flight_number character varying (10)	average_price numeric	maximum_price numeric	minimum_price numeric	total_price numeric
1	SU1005	20125.00000000000000	28000.00	14500.00	80500.00
2	SU1008	15500.00000000000000	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	24600.00000000000000	31000.00	12000.00	123000.00
6	SU1001	19300.00000000000000	27500.00	13000.00	96500.00
7	SU1004	19750.00000000000000	30000.00	14000.00	79000.00
8	SU1003	20200.00000000000000	35000.00	12500.00	101000.00

Showing rows: 1 to 8 | Page No: 1 of 1 | Back | Forward | SQL

✓ Successfully run. Total query runtime: 172 msec. 8 rows affected. ✘

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 database interface with a query editor and a results viewer.

Query History:

```

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';

```

Data Output:

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

Messages:

✓ Successfully run. Total query runtime: 154 msec. 0 rows affected.

Total rows: 0 | Query complete 00:00:00.154 | CPU: 1m6.00s Col: 24

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 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

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

Total rows: 0 Query complete 00:00:00.070 CTEs: 1 In 11 Col 50

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

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 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

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

Total rows: 0 Query complete 00:00:00.066 CTEs: 1 In 2 Col 50

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

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 database interface with two main sections: 'Query' and 'Data Output'.

Query: Contains the SQL code from the previous block. The code is highlighted in blue and pink, indicating syntax and identifiers.

```
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;
```

Data Output: Shows the schema of the resulting table with three columns: flight_id, airline_name, and country. The airline_name column is currently selected.

flight_id	airline_name	country
integer	character varying (50)	character varying (50)

A green success message at the bottom right indicates: "Successfully run. Total query runtime: 68 msec. 0 rows affected.".