

**Kazakh-British Technical University**

**DATABASES**

**Lab 6**

**Alexandr Kislistsin**

**23B031861**

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

The screenshot shows the DBeaver interface with the 'lab\_3 - console' tab selected. In the 'Database Explorer' panel, the 'Flights' table is selected. The 'console' tab contains the following SQL query:

```
398 ADD CONSTRAINT chk_ticket_discount
400 );
401
402 ✓ SELECT f.flight_id,
403     f.sch_departure_time,
404     f.sch_arrival_time,
405     a.airline_name
406     FROM flights f
407     JOIN airline a ON f.airline_id = a.airline_id
408 WHERE a.airline_name = 'KazAir';
```

The 'Output' tab shows the results of the query:

flight_id	sch_departure_time	sch_arrival_time	airline_name
1	2025-11-16 10:33:57	2025-11-16 10:33:57	KazAir

Services panel shows a transaction named 'Flights' with a duration of 410 ms.

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

The screenshot shows the DBeaver interface with the 'lab\_3 - console' tab selected. In the 'Database Explorer' panel, the 'Flights' table is selected. The 'console' tab contains the following SQL query:

```
404 );
405
406 ✓ SELECT f.flight_id,
407     f.sch_departure_time,
408     f.sch_arrival_time,
409     a.airline_name
410     FROM flights f
411     JOIN airline a ON f.airline_id = a.airline_id
412 WHERE a.airline_name = 'KazAir';
413
414 ✓ SELECT f.flight_id,
415     f.sch_departure_time,
416     a.airport_name AS departure_airport
417     FROM flights f
418     JOIN airport a ON f.departing_airport_id = a.airport_id;
```

The 'Output' tab shows the results of the query:

flight_id	sch_departure_time	departure_airport
1	2025-07-20 10:27:20	East Patricia Intl Airport
2	2025-06-19 22:56:44	Garciaville International Airport
3	2024-11-23 14:49:10	Montoyachester Intl Airport
4	2025-05-29 12:25:08	Port Beverly Intl Airport
5	2025-09-05 11:24:17	West Carolinestad International Airport
6	2025-09-24 01:45:31	New Christopherside Intl Airport
7	2025-06-17 14:56:28	view Intl Airport
8	2025-01-17 17:24:40	San Intl Airport

Services panel shows a transaction named 'Flights' with a duration of 410 ms.

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

The screenshot shows the DataGrip IDE interface. In the Database Explorer, under the 'localhost' database, the 'Flights' table is selected. In the top-right console window, a query is being typed:

```
415     f.sch_departure_time,
416     a.airport_name AS departure_airport
417 FROM flights f
418 JOIN airport a ON f.departing_airport_id = a.airport_id;
419
420
421
422 SELECT a.airline_name
423 FROM airline a
424 LEFT JOIN flights f ON a.airline_id = f.airline_id
425     AND MONTH(f.sch_departure_time) = MONTH(CURDATE()) + 1
426 WHERE f.flight_id IS NULL;
```

Below the console, the Services tab shows the 'Output' section for the 'airline' table, displaying a list of airline names:

airline_name
FlyFly
Nelson Ltd
Ramirez, Green and Taylor
Parker Ltd
Bryan-Dixon
Anderson-Green
Ballard-Hamilton
Sloan-Welch

The status bar at the bottom indicates the query took 426 ms to execute.

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

The screenshot shows the DataGrip IDE interface. In the Database Explorer, under the 'localhost' database, the 'Flights' table is selected. In the top-right console window, a query is being typed:

```
421
422     SELECT a.airline_name
423     FROM airline a
424     LEFT JOIN flights f ON a.airline_id = f.airline_id
425         AND MONTH(f.sch_departure_time) = MONTH(CURDATE()) + 1
426
427
428
429
430     SELECT p.first_name, p.last_name, f.flight_id
431     FROM passengers p
432     JOIN booking b ON p.passenger_id = b.passenger_id
433     JOIN flights f ON b.flight_id = f.flight_id
434 WHERE f.flight_id = ?;
```

Below the console, the Services tab shows the 'Output' section for the 'Result 22' table, displaying two rows of passenger information:

first_name	last_name	flight_id
Carlos	Allen	7
Erik	Ferrell	7

The status bar at the bottom indicates the query took 434 ms to execute.

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

The screenshot shows the DBeaver interface with the 'console' tab selected. In the Database Explorer, the 'Flights' table is highlighted. The SQL query entered in the console is:

```

SELECT p.first_name, p.last_name, f.flight_id
FROM passengers p
JOIN booking b ON p.passenger_id = b.passenger_id
JOIN flights f ON b.flight_id = f.flight_id
WHERE f.flight_id = 7;

SELECT flight_id,
       AVG(ticket_price) AS avg_price,
       SUM(ticket_price) AS total_price,
       MAX(ticket_price) AS max_price,
       MIN(ticket_price) AS min_price
FROM booking
GROUP BY flight_id;

```

The results are displayed in a table titled 'Result 24' with columns: flight\_id, avg\_price, total\_price, max\_price, and min\_price. The data is as follows:

flight_id	avg_price	total_price	max_price	min_price
5	1152.990000	1152.99	1152.99	1152.99
6	1830.066667	5490.20	2309.21	1544.11
7	1568.410000	3136.82	1984.49	1152.33
8	1740.360000	1740.36	1740.36	1740.36
9	1089.160000	3027.48	1612.05	602.60
12	1090.755000	2181.51	1095.99	1085.52
15	1188.603333	7565.81	1985.05	495.93
17	1048.460000	145.38	1505.02	346.02

## 6. Create a query that shows all flights flying to a specific country by combining flights, airports and airline.

The screenshot shows the DBeaver interface with the 'console' tab selected. In the Database Explorer, the 'Flights' table is highlighted. The SQL query entered in the console is:

```

SELECT f.flight_id,
       a.airline_name,
       ap.airport_name AS destination_airport,
       ap.country AS destination_country
FROM flights f
JOIN airline a ON f.airline_id = a.airline_id
JOIN airport ap ON f.arriving_airport_id = ap.airport_id
WHERE ap.country = 'China';

```

The results are displayed in a table titled 'Result 25' with columns: flight\_id, airline\_name, destination\_airport, and destination\_country. The data is as follows:

flight_id	airline_name	destination_airport	destination_country
7	Roberts-Delgado	New Patricktown Intl Airport	China
86	Pham-Mann	New Patricktown Intl Airport	China
21	Lewis, Gonzalez and Combs	Lake Gregoryburgh Intl Airport	China
35	West-Hawkins	Shawnport Intl Airport	China
84	West-Hawkins	Shawnport Intl Airport	China
87	AirEasy	Shawnport Intl Airport	China

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

The screenshot shows a database management interface with a central console window. The query being run is:

```
451 FROM flights f
452 JOIN airline a ON f.airline_id = a.airline_id
453 JOIN airport ap ON f.arriving_airport_id = ap.airport_id
454 WHERE ap.country = 'China';
455
456
457 SELECT p.first_name, p.last_name, ap.airport_name AS arrival_airport
458 FROM passengers p
459 JOIN booking b ON p.passenger_id = b.passenger_id
460 JOIN flights f ON b.flight_id = f.flight_id
461 JOIN airport ap ON f.arriving_airport_id = ap.airport_id
462 WHERE TIMESTAMPOFF(YEAR, p.date_of_birth, CURDATE()) < 18;
```

The results of the query are displayed in the "Result 26" tab, showing 0 rows.

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

The screenshot shows a database management interface with a central console window. The query being run is:

```
455
456
457 SELECT p.first_name, p.last_name, ap.airport_name AS arrival_airport
458 FROM passengers p
459 JOIN booking b ON p.passenger_id = b.passenger_id
460 JOIN flights f ON b.flight_id = f.flight_id
461 JOIN airport ap ON f.arriving_airport_id = ap.airport_id
462 WHERE TIMESTAMPOFF(YEAR, p.date_of_birth, CURDATE()) < 18;
463
464
465 SELECT p.first_name, p.last_name,
466     p.passport_number,
467     f.act_arrival_time AS arrival_time
468 FROM passengers p
469 JOIN booking b ON p.passenger_id = b.passenger_id
470 JOIN flights f ON b.flight_id = f.flight_id;
```

The results of the query are displayed in the "Result 27" tab, showing 8 rows of passenger information with their arrival times.

first_name	last_name	passport_number	arrival_time
Anthony	Allison	P000057	2025-02-03 15:24:31
Mary	Porter	P000038	2025-10-05 00:53:39
Robert	Andrews	P000070	2025-06-02 14:27:32
Cory	Espinosa	P000033	2025-03-01 12:52:06
Devin	Richardson	P000060	2025-05-11 21:37:21
Marcia	Hanson	P000093	2025-10-30 06:23:56
Samantha	Kennedy	P000075	2025-06-20 09:15:44
Samantha	Kennedy	P000080	2025-10-30 06:23:56

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

The screenshot shows a database management interface with a sidebar for 'Database Explorer' and a main area for 'console'. The 'Database Explorer' sidebar lists databases, tables, and objects. The 'console' area contains two SQL queries. The first query selects passenger details and flight arrival times. The second query, highlighted with a green border, joins flights with airline and airport tables to find flights where the airline's country of origin matches the airport's country. The results show 0 rows.

```
465 SELECT p.first_name, p.last_name,
466     p.passport_number,
467     f.act_arrival_time AS arrival_time
468 FROM passengers p
469 JOIN booking b ON p.passenger_id = b.passenger_id
470 JOIN flights f ON b.flight_id = f.flight_id;
471
472
473 SELECT f.flight_id,
474     al.airline_name,
475     ap.country AS airport_country
476 FROM flights f
477 JOIN airline al ON f.airline_id = al.airline_id
478 JOIN airport ap ON f.departing_airport_id = ap.airport_id
479 WHERE al.airline_country = ap.country
480 GROUP BY ap.country, f.flight_id, al.airline_name;
```

Output Result 28

flight_id	airline_name	airport_country

0 rows

## Conclusion

In this lab work, I learned how to use different types of SQL JOINs to combine data from multiple related tables.

I practiced using **INNER JOIN**, **LEFT JOIN**, **RIGHT JOIN**, and **FULL JOIN** to extract connected information such as flight details, passengers, and airlines.

This helped me understand how relationships between tables work and how to write more complex queries to get meaningful results from a relational database.