

Laboratory work 6

JOIN operations tasks

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

The screenshot shows a PostgreSQL database interface with the following details:

- Object Explorer:** On the left, it lists several tables:
 - airline_info** (Columns: airline_id, airline_code, airline_name, airline_country, created_at, updated_at, info)
 - airline** (Columns: airport_id, airport_name, country, state, city, created_at, updated_at)
 - baggage**
 - baggage_check**
 - boarding_pass**
 - booking** (Columns: booking_id, flight_id, passenger_id, booking_platform)
- SQL Editor:** The main area contains a query window with the following SQL code:

```
1 SELECT f.flight_id, f.sch_departure_time, a.airline_name
2 FROM flights f
3 JOIN airline_info a ON f.airline_id = a.airline_id
4 WHERE airline_name LIKE 'Air%'
```
- Data Output:** Below the SQL editor, a table displays the results of the query:

flight_id	sch_departure_time	airline_name
1	2025-06-20 00:00:00	Air India
2	2024-11-22 00:00:00	Air Canada
3	2024-10-20 00:00:00	Air New Zealand
4	2024-11-24 00:00:00	Air France
- Messages:** A green message bar at the bottom right indicates: "Successfully run. Total query runtime: 66 msec. 4 rows affected." and "LF Ln 4, Col 27".

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

```

SELECT f.flight_id, f.sch_departure_time, a.airport_name
FROM flights f
JOIN airport a ON f.departing_airport_id = a.airport_id;

```

flight_id	sch_departure_time	airport_name
1	2025-07-10 00:00:00	Tumeremo Airport
2	2025-06-20 00:00:00	Yorkton Municipal Airport
3	2025-08-30 00:00:00	Kenosha Regional Airport
4	2025-03-20 00:00:00	Senadora Eunice Micheles Airport
5	2025-03-20 00:00:00	Riverside Airport
6	2024-10-06 00:00:00	Lalmonirhat Airport
7	2025-03-09 00:00:00	Puerto Leda Airport
8	2025-06-04 00:00:00	Bembbridge Airport
9	2025-05-27 00:00:00	Keokuk Municipal Airport
10	2024-12-06 00:00:00	Maxson Airfield
11	2024-10-31 00:00:00	Dongola Airport
12	2025-03-08 00:00:00	Zabreh Ostrava Airport
13	2025-03-28 00:00:00	Cap Manuel Niño International Airport
14	2024-10-30 00:00:00	Asiaat Airport
15	2024-12-27 00:00:00	Myeik Airport
16	2025-09-08 00:00:00	Toccoa Airport - R.G. Letourneau Field

Total rows: 153 Query complete 00:00:00.086

Successfully run. Total query runtime: 86 msec. 153 rows affected.

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

Object Explorer

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents × Processes × db_lab/postgres@... × db_lab/postgres@PostgreSQL 17* ×

Query History

```

1 ✓ SELECT a.airline_name
2   FROM airline_info a
3   LEFT JOIN flights f ON a.airline_id = f.airline_id
4   AND EXTRACT(MONTH FROM f.sch_departure_time) = EXTRACT(MONTH FROM f.sch_departure_time) + 1
5   WHERE f.flight_id IS NULL;

```

Data Output

airline_name
South African Airways
Singapore Airlines
SAS Scandinavian
Air India
Thai Airways
British Airways
Alitalia
Avianca
Qantas
LATAM Airlines
Ethiopian Airlines
All Nippon Airways
Lufthansa
Korean Air
Qatar Airways
Southwest Airlines

Total rows: 31 Query complete 00:00:00.089

Successfully run. Total query runtime: 89 msec. 31 rows affected. LF Ln 3, Col 21

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

Object Explorer

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents × Processes × db_lab/postgres@... × db_lab/postgres@PostgreSQL 17* ×

Query History

```

1 ✓ SELECT f.flight_id, p.first_name, p.last_name
2   FROM passengers p
3   JOIN booking b ON p.passenger_id = b.passenger_id
4   JOIN flights f ON b.flight_id = f.flight_id
5   WHERE f.flight_id BETWEEN 1 AND 200;

```

Data Output

flight_id	first_name	last_name
-----------	------------	-----------

Total rows: 0 Query complete 00:00:00.116

Successfully run. Total query runtime: 116 msec. 0 rows affected. LF Ln 5, Col 37

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

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer displays the database schema with tables like baggage_check, boarding_pass, booking, flights, and passengers. The booking table has 8 columns, and the flights table has 12 columns. The flights table is selected. In the center, the Query History tab contains the following SQL code:

```
1 ✓ SELECT f.flight_id, AVG(b.price), SUM(b.price), MIN(b.price), MAX(b.price)
2   FROM flights f
3   JOIN booking b ON f.flight_id = b.flight_id
4 GROUP BY f.flight_id;
```

The Data Output tab shows a table structure with columns: flight_id [PK] integer, avg numeric, sum numeric, min numeric, max numeric. At the bottom, a message bar indicates: "Successfully run. Total query runtime: 129 msec. 0 rows affected." and "LF Ln 4, Col 22".

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.

Object Explorer

```

    SELECT f.flight_id, a.airline_name, ap.country
    FROM flights f
    JOIN airport ap ON f.arriving_airport_id = ap.airport_id
    JOIN airline_info a ON f.airline_id = a.airline_id
    WHERE ap.country = 'Canada';
  
```

Data Output

flight_id	airline_name	country

Total rows: 0 Query complete 00:00:00.066 ✓ Successfully run. Total query runtime: 66 msec. 0 rows affected. LF Ln 5, Col 29

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

```

    SELECT p.first_name, p.last_name, ap.city
    FROM passengers p
    JOIN booking b ON p.passenger_id = b.passenger_id
    JOIN flights f ON b.flight_id = f.flight_id
    JOIN airport ap ON f.arriving_airport_id = ap.airport_id
    WHERE EXTRACT(YEAR FROM AGE(p.date_of_birth)) < 18;
  
```

Data Output

first_name	last_name	city

Total rows: 0 Query complete 00:00:00.064 ✓ Successfully run. Total query runtime: 64 msec. 0 rows affected. LF Ln 6, Col 52

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

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer displays database schemas and tables. The 'flights' table under the 'bookings' schema has its columns expanded, with 'act_arrival_time' selected. The 'passengers' table also has its columns expanded. In the center, the Query Editor window contains the following SQL query:

```
1 SELECT p.first_name || ' ' || p.last_name AS full_name, p.passport_number, f.act_arrival_time
2 FROM passengers p
3 JOIN bookings b ON p.passenger_id = b.passenger_id
4 JOIN flights f ON b.flight_id = f.flight_id;
```

The Data Output tab shows the results of the query:

full_name	passport_number	act_arrival_time

At the bottom right of the interface, a message box indicates: "Successfully run. Total query runtime: 194 msec. 0 rows affected. LF Ln 4, Col 45".

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

Object Explorer

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents × Processes × db_lab/postgres@... × db_lab/postgres@PostgreSQL 17* ×

Query History

```
1 ✓ SELECT f.flight_id, a.airline_name, ai.country
2 FROM flights f
3 JOIN airline_info a ON f.airline_id = a.airline_id
4 JOIN airports ai ON f.departing_airport_id = ai.airport_id
5 WHERE a.airline_country = ai.country
6 GROUP BY ai.country, f.flight_id, a.airline_name;
7
```

Data Output

flight_id	airline_name	country
integer	character varying (300)	character varying (150)

Total rows: 0 Query complete 00:00:00.104

Successfully run. Total query runtime: 104 msec. 0 rows affected.

LF Ln 6, Col 49

The screenshot shows a PostgreSQL database management interface. On the left, the Object Explorer pane lists several tables: airline_info, flight, airport, baggage, baggage_check, and boarding_pass. The airline_info table is currently selected, showing its columns: airline_id, airline_code, airline_name, airline_country, created_at, updated_at, and info. The flight table is also partially visible. In the center, a SQL query editor window contains a query that joins the flights, airline_info, and airports tables to find flights originating from specific countries. The results are displayed in a table with three columns: flight_id, airline_name, and country. A success message at the bottom indicates the query ran successfully with no rows affected. The status bar at the bottom shows the total runtime and the current line and column of the query.