

# Laboratory work 6

## JOIN operations tasks

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

The screenshot shows the pgAdmin interface. In the left sidebar, under 'Tables', there is a section for the 'airline' table. In the main area, the SQL tab contains the following query:

```

SELECT f.flight_no,
       f.scheduled_departure,
       f.scheduled_arrival,
       a.airline_name
  FROM flights f
 JOIN airline a ON f.airline_id = a.airline_id
 WHERE a.airline_name = 'IPC';
    
```

The Data Output tab shows the results of the query, listing 12 flights from the 'airline' table. The columns are flight\_no, scheduled\_departure, scheduled\_arrival, and airline\_name. The airline\_name for all flights is 'IPC'.

flight_no	scheduled_departure	scheduled_arrival	airline_name
BR-PE	2024-01-16	2023-06-02	IPC
MZ-G	2023-09-21	2023-11-29	IPC
AU-NT	2023-03-29	2023-05-06	IPC
FR-K	2023-12-26	2023-04-30	IPC
US-VT	2023-08-28	2023-08-02	IPC
PH-BUK	2023-08-20	2023-10-24	IPC
SD-01	2023-12-02	2023-12-17	IPC
PA-8	2023-03-19	2023-08-21	IPC
NA-KU	2023-06-26	2023-12-04	IPC
US-CT	2023-11-07	2023-05-03	IPC
RU-CU	2023-04-16	2023-09-15	IPC
BE-VAN	2023-05-03	2023-06-01	IPC

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

The screenshot shows the pgAdmin interface. In the left sidebar, under 'Tables', there is a section for the 'flights' table. In the main area, the SQL tab contains the following query:

```

SELECT f.flight_no,
       dep.airport_name AS departure_airport,
       f.scheduled_departure
  FROM flights f
 JOIN airport dep ON f.departure_airport_id = dep.airport_id;
    
```

The Data Output tab shows the results of the query, listing 12 flights from the 'flights' table. The columns are flight\_no, departure\_airport, and scheduled\_departure. The departure\_airport for all flights is 'Elorza Airport'.

flight_no	departure_airport	scheduled_departure
US-CT	Elorza Airport	2024-01-22
US-NM	Figari Sud-Corse Airport	2023-07-21
FI-OL	Darchula Airport	2023-03-29
RU-KR	Lime Acres Finsch Mine Airport	2024-01-02
RO-DJ	Hana Airport	2023-07-03
CA-SK	Darchula Airport	2023-07-07
AU-TAS	Ocean Falls Seaplane Base	2023-10-12
US-AZ	Figari Sud-Corse Airport	2023-07-29
IN-OR	Hana Airport	2023-05-18
AU-NT	Longana Airport	2023-11-25
TH-57	Delta County Airport	2023-03-28
CA-NL	Elorza Airport	2023-03-22

At the bottom right, a green message box indicates: 'Successfully run. Total query runtime: 44 msec. 1000 rows affected.'

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

The screenshot shows the pgAdmin interface with a database connection to 'public.flights' on PostgreSQL 17+. The query window contains the following SQL code:

```

1 SELECT a.airline_name
2 FROM airline a
3 LEFT JOIN flights f
4     ON a.airline_id = f.airline_id
5     AND f.scheduled_departure BETWEEN CURRENT_DATE AND (CURRENT_DATE + INTERVAL '1 month')
6 WHERE f.flight_id IS NULL;

```

The results pane displays a table with 12 rows, each containing an airline name:

airline_name
IPC
PDN
KLE
KHS
YLQ
NGL
O
QIG
NQX
SZ
IVA
KOQ

Total rows: 50 Query complete 00:00:00.057

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

The screenshot shows the pgAdmin interface with a database connection to 'public.flights' on PostgreSQL 17+. The query window contains the following SQL code:

```

1 SELECT p.first_name || ' ' || p.last_name AS full_name,
2        f.flight_no
3 FROM passengers p
4 JOIN booking b ON p.passenger_id = b.passenger_id
5 JOIN booking_flight bf ON b.booking_id = bf.booking_id
6 JOIN flights f ON bf.flight_id = f.flight_id
7 WHERE f.flight_no = 'US-AR';

```

The results pane displays a table with 6 rows, each containing a passenger's full name and the flight number:

full_name	flight_no
Tildy Shacklefo...	US-AR
Tom Gidney	US-AR
Humphrey Mc...	US-AR
Franny Orry	US-AR
Margarethe Yule	US-AR
Reinald Pococke	US-AR

Total rows: 6 Query complete 00:00:00.053

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

```

SELECT f.flight_no,
       ROUND(AVG(b.price), 2) AS avg_price,
       SUM(b.price) AS total_price,
       MAX(b.price) AS max_price,
       MIN(b.price) AS min_price
  FROM flights f
 JOIN booking_flight bf ON f.flight_id = bf.flight_id
 JOIN booking b ON bf.booking_id = b.booking_id
 GROUP BY f.flight_no;

```

The screenshot shows the pgAdmin interface with a query editor window. The query above is run against the 'flights' database. The results are displayed in a table with columns: flight\_no, avg\_price, total\_price, max\_price, and min\_price. The table contains 384 rows of flight data from various airlines like FR-O, ZA-NL, AF-FRA, etc., with their respective price statistics.

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_no,
       a.airline_name,
       arr.airport_name AS arrival_airport,
       arr.country AS destination_country
  FROM flights f
 JOIN airline a ON f.airline_id = a.airline_id
 JOIN airport arr ON f.arrival_airport_id = arr.airport_id
 WHERE arr.country = 'Russia';

```

The screenshot shows the pgAdmin interface with a query editor window. The query joins the 'flights', 'airline', and 'airport' tables to filter flights destined for 'Russia'. The results are displayed in a table with columns: flight\_no, airline\_name, arrival\_airport, and destination\_country. The table contains 56 rows of flight data to 'Akunaq Heliport' in 'Russia'.

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

The screenshot shows the pgAdmin interface with a database tree on the left and a query editor on the right. The query editor contains the following SQL code:

```

SELECT p.first_name || ' ' || p.last_name AS full_name,
       arr.country AS arrival_country
  FROM passengers p
  JOIN booking b ON p.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.arrival_airport_id = arr.airport_id
 WHERE EXTRACT(YEAR FROM AGE(CURRENT_DATE, p.date_of_birth)) < 18;

```

The results pane shows a table with columns: full\_name (text) and arrival\_country (character varying(50)).

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 interface with a database tree on the left and a query editor on the right. The query editor contains the following SQL code:

```

SELECT p.first_name || ' ' || p.last_name AS full_name,
       p.passport_number,
       f.actual_arrival
  FROM passengers p
  JOIN booking b ON p.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;

```

The results pane shows a table with columns: full\_name (text), passport\_number (character varying(50)), and actual\_arrival (date). The data includes 1000 rows of passenger information.

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

database\_subj/postgres@PostgreSQL 17\* public.flights/data... public.airport/data...

```
1 SELECT f.flight_no,
2        dep.country AS origin_country,
3        a.airline_country AS airline_country
4 FROM flights f
5 JOIN airline a ON f.airline_id = a.airline_id
6 JOIN airport dep ON f.departure_airport_id = dep.airport_id
7 WHERE a.airline_country = dep.country
8 GROUP BY f.flight_no, dep.country, a.airline_country;
```

Data Output Messages Notifications

	flight_no	origin_country	airline_country
1	AU-NSW	China	China
2	AU-NT	China	China
3	AU-NT	Greece	Greece
4	AU-QLD	Poland	Poland
5	AU-WA	China	China
6	BR-AM	China	China
7	BR-BA	China	China
8	BR-MG	China	China
9	BR-PE	China	China
10	BR-PR	Philippines	Philippines
11	BS-NP	China	China
12	CA-BC	China	China

Total rows: 69 Query complete 00:00:00.049

Successfully run. Total query runtime: 49 msec. 69 rows affected.