

JOIN operations tasks

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

The screenshot shows the pgAdmin 4 interface. The query editor contains the following SQL query:

```
1 SELECT f.flight_id, a.airline_name
2 FROM flights f
3 INNER JOIN airline a
4 ON f.airline_id = a.airline_id;
```

The Data Output pane shows the results of the query, displaying 98 rows. The columns are `flight_id` (integer) and `airline_name` (character varying (50)).

flight_id	airline_name
1	TWITTERNATION
2	PIXOBOO
3	KIMIA
4	ABATA
5	KAYVEO
6	PHOTOLIST
7	LINKTYPE
8	DABLIST
9	CAMIDO
10	MYNTE

Successfully run. Total query runtime: 96 msec. 98 rows affected.

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

The screenshot shows the pgAdmin 4 interface. The query editor contains the following SQL query:

```
1 SELECT f.flight_id, a.airport_name
2 FROM flights f
3 INNER JOIN airport a
4 ON f.departing_airport_id = a.airport_id;
```

The Data Output pane shows the results of the query, displaying 98 rows. The columns are `flight_id` (integer) and `airport_name` (character varying (50)).

flight_id	airport_name
1	JAN Airport
2	BVK Airport
3	QDJ Airport
4	PHO Airport
5	SSH Airport
6	PNI Airport
7	SKW Airport
8	DAV Airport
9	HAY Airport
10	VLS Airport

Successfully run. Total query runtime: 85 msec. 98 rows affected.

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

pgAdmin 4

Welcome airport_db/postgres@PostgreSQL 17*

airport_db/postgres@PostgreSQL 17

Query Query History

```

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

```

Data Output Messages Notifications

Showing rows: 1 to 200 Page No: 1 of 1

airline_name
QUATZ
TOPICLOUNGE
FANODDLE
GEVEE
JANYX
QUAXO
REALFIRE
CHATTERPOINT
RIFFWIRE
ABATA

Total rows: 200 Query complete 00:00:00.227

Successfully run. Total query runtime: 227 msec. 200 rows affected.

2 cm of snow Wednesday

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

pgAdmin 4

Welcome airport_db/postgres@PostgreSQL 17*

airport_db/postgres@PostgreSQL 17

Query Query History

```

1 SELECT
2   p.first_name,
3   p.last_name
4 FROM passengers p
5 JOIN booking b ON p.passenger_id = b.passenger_id
6 WHERE b.flight_id = 100;

```

Data Output Messages Notifications

Showing rows: 1 to 2 Page No: 1 of 1

first_name	last_name
Ebeneser	Klimecki
Aldous	Wild

Total rows: 2 Query complete 00:00:00.117

Successfully run. Total query runtime: 117 msec. 2 rows affected.

5°C Cloudy

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

pgAdmin 4

Welcome airport_db/postgres@PostgreSQL 17*

airport_db/postgres@PostgreSQL 17

Query

```

1 SELECT
2     f.flight_id,
3     AVG(b.ticket_price) AS avg_price,
4     SUM(b.ticket_price) AS total_price,
5     MAX(b.ticket_price) AS max_price,
6     MIN(b.ticket_price) AS min_price
7 FROM flights f
8 JOIN booking b
9     ON f.flight_id = b.flight_id
10 GROUP BY f.flight_id;
11

```

Data Output

Showing rows: 1 to 48 | Page No: 1 of 1

flight_id [PK] integer	avg_price numeric	total_price numeric	max_price numeric	min_price numeric
116	21169.4900000000000000	21169.49	21169.49	21169.49
54	16624.2500000000000000	66497.00	19938.55	9984.88
71	16646.4700000000000000	16646.47	16646.47	16646.47
68	16089.1500000000000000	48267.45	20845.30	12022.32
138	20663.4100000000000000	20663.41	20663.41	20663.41
70	11437.5100000000000000	11437.51	11437.51	11437.51
146	15662.5900000000000000	15662.59	15662.59	15662.59
67	14420.7800000000000000	14420.78	14420.78	14420.78
63	16798.5000000000000000	16798.50	16798.50	16798.50
174	20453.3200000000000000	40906.64	22519.86	18386.78

Total rows: 48 | Query complete 00:00:00.140

Successfully run. Total query runtime: 140 msec. 48 rows affected.

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.

pgAdmin 4

Welcome airport_db/postgres@PostgreSQL 17*

airport_db/postgres@PostgreSQL 17

Query

```

1 SELECT
2     f.flight_id,
3     a.airport_name AS arrival_airport,
4     a.country,
5     ai.airline_name
6 FROM flights f
7 JOIN airport a
8     ON f.arriving_airport_id = a.airport_id
9 JOIN airline ai
10    ON f.airline_id = ai.airline_id
11 WHERE a.country = 'Turkey';
12

```

Data Output

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

Total rows: 0 | Query complete 00:00:00.078

Successfully run. Total query runtime: 78 msec. 0 rows affected.

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

The screenshot shows the pgAdmin 4 interface with a SQL query executed. The query selects the first name, last name, and arrival destination for passengers, joining the passengers, bookings, and flights tables. The results are displayed in the Data Output tab.

```

1 SELECT
2   p.first_name,
3   p.last_name,
4   a.airport_name AS arrival_destination
5 FROM passengers p
6 JOIN bookings b
7   ON p.passenger_id = b.passenger_id
8 JOIN flights f
9   ON b.flight_id = f.flight_id
10 JOIN airport a
11   ON f.arriving_airport_id = a.airport_id
12 WHERE EXTRACT(YEAR FROM AGE(CURRENT_DATE, p.date_of_birth)) < 18;

```

first_name	last_name	arrival_destination
character varying (50)	character varying (50)	character varying (50)

Successfully run. Total query runtime: 74 msec. 0 rows affected.

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 with a SQL query executed. The query selects the full name (concatenated first and last names), passport number, and arrival time for passengers, joining the passengers, bookings, and flights tables. The results are displayed in the Data Output tab.

```

1 SELECT
2   p.first_name || ' ' || p.last_name AS full_name,
3   p.passport_number,
4   f.sch_arrival_time AS arrival_time
5 FROM passengers p
6 JOIN bookings b
7   ON p.passenger_id = b.passenger_id
8 JOIN flights f
9   ON b.flight_id = f.flight_id;

```

full_name	passport_number	arrival_time
text	character varying (20)	timestamp without time zone
1	Buiron Branscombe	P010222
2	Kirbee Elloy	P189647
3	Gino Derkes	P827654
4	Nikoletta Demcak	P767986
5	Tadeas Crichmere	P150074
6	Abramo Pond-Jones	P913700
7	Engracia Fernandez	P056899
8	Aldwin Laviss	P546764
9	Stern Marr	P859822
10	Ebeneser Klimecki	P353234

Successfully run. Total query runtime: 93 msec. 67 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.

pgAdmin 4

File Object Tools Edit View Window Help

Welcome airport_db/postgres@PostgreSQL 17* X

airport_db/postgres@PostgreSQL 17

Query Query History

```
1 SELECT
2     f.flight_id,
3     ai.airline_name,
4     a.country AS airport_country
5 FROM flights f
6 JOIN airline ai
7     ON f.airline_id = ai.airline_id
8 JOIN airport a
9     ON f.departing_airport_id = a.airport_id
10 WHERE ai.airline_country = a.country
11 GROUP BY a.country, f.flight_id, ai.airline_name;
```

Data Output Messages Notifications

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

Total rows: 0 Query complete 00:00:00.080

Successfully run. Total query runtime: 80 msec. 0 rows affected.

5°C Cloudy

Поиск

ENG 23:18 04.11.2025