

JOIN operations tasks

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

The screenshot shows a PostgreSQL database connection named 'lab1-db' in DBeaver. In the SQL editor, a query is written to select flights from the 'flights' table, join it with the 'airline' table on the 'airline_id' column, and filter the results where the airline name is 'Emirates'. The results table shows 15 rows of flight information, including departure and arrival times, airport IDs, and airline names. The airline names in the results table are all 'Emirates'.

```
304 ✓ select
305   f.flight_id,
306   f.sch_departure_time,
307   f.sch_arrival_time,
308   f.departing_airport_id,
309   f.arriving_airport_id,
310   a.airline_name,
311   a.airline_country
312 from flights f
313 join airline a
314   1..n->1: on f.airline_id = a.airline_id
315 where a.airline_name = 'Emirates';
316
```

flight_id	sch_departure_time	sch_arrival_time	departing_airport_id	arriving_airport_id	airline_name	airline_country
1	2014-12-29 06:06:41.000000	2005-11-26 17:49:53.000000	26	165	Emirates	Ukraine
2	49 2003-10-11 15:16:38.000000	2016-06-22 13:42:17.000000	124	84	Emirates	Bulgaria
3	83 2012-05-24 04:54:26.000000	2015-07-10 19:07:23.000000	151	92	Emirates	South Africa
4	85 2022-07-07 23:33:38.000000	2005-09-01 19:02:17.000000	17	39	Emirates	China
5	97 2010-09-19 16:25:24.000000	2024-07-15 22:58:58.000000	65	4	Emirates	Venezuela
6	106 2009-06-02 00:26:58.000000	2013-10-05 00:43:46.000000	93	89	Emirates	Nicaragua
7	107 2024-02-11 01:19:59.000000	2025-02-01 19:23:44.000000	171	81	Emirates	South Africa
8	149 2017-04-01 21:07:51.000000	2020-11-10 23:00:11.000000	21	6	Emirates	China
9	157 2018-08-21 17:05:33.000000	2028-02-01 07:18:09.000000	141	94	Emirates	Bulgaria
10	237 2024-10-03 03:49:58.000000	2025-06-06 04:32:45.000000	341	1	Emirates	China
11	262 2025-09-13 17:03:38.000000	2025-01-07 07:08:28.000000	22	60	Emirates	Bulgaria
12	301 2024-10-19 08:39:31.000000	2024-10-21 11:02:16.000000	334	228	Emirates	China
13	311 2025-04-07 00:55:43.000000	2025-04-16 10:48:23.000000	69	44	Emirates	China

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

The screenshot shows a PostgreSQL database connection named 'lab1-db' in DBeaver. In the SQL editor, a query is written to select flight details and join them with the 'airport' table on the 'departing_airport_id' column. The results table shows 15 rows of flight information, including departure and arrival times, and the names of the departure airports. The departure airports listed include Blue Canyon Nyack Airport, Viru Viru International Airport, Lauro Kurtz Airport, Solano Airport, Douglas Lake Airport, Wantoat Airport, Semonkong Airport, Manners Creek Airport, Sisimiut Airport, Azara, Lepaterique, El Guamo, and others.

```
210 ✓
211 select
212   f.flight_id,
213   a.airport_name as departure_airport,
214   a.city as departure_city,
215   f.sch_departure_time,
216   f.sch_arrival_time
217 from flights f
218 join airport a
219   1..n->1: on f.departing_airport_id = a.airport_id;
```

flight_id	departure_airport	departure_city	sch_departure_time	sch_arrival_time
1	3 Blue Canyon Nyack Airport	Huolu	2007-05-07 19:10:38.000000	2025-05-21 13:22:47.000000
2	4 Viru Viru International Airport	Longos	2016-07-26 03:27:54.000000	2002-03-30 22:46:52.000000
3	5 Lauro Kurtz Airport	Vänersborg	2011-12-15 18:50:20.000000	2019-09-19 20:20:51.000000
4	6 Solano Airport	Umeå	2006-08-01 00:29:29.000000	2024-03-28 09:04:26.000000
5	7 Douglas Lake Airport	Riebini	2014-04-12 05:27:46.000000	2003-04-12 18:50:18.000000
6	8 Wantoat Airport	Yanhe	2012-01-25 03:52:31.000000	2014-11-02 01:10:35.000000
7	9 Semonkong Airport	Rokyancy	2002-12-23 15:19:38.000000	2012-03-23 17:59:28.000000
8	10 Manners Creek Airport	Azara	2014-03-31 08:30:51.000000	2014-05-06 21:05:45.000000
9	11 Sisimiut Airport	Lepaterique	2009-09-03 13:50:00.000000	2012-06-08 04:32:02.000000
10	12 Azaza Airport	El Guamo	2014-03-25 05:09:23.000000	2019-05-07 15:39:15.000000
11	13 Lefkoniko Airport	Uberaba	2012-06-18 00:43:59.000000	2005-12-21 12:07:13.000000
12	14 LTS Pulau Redang Airport	Yanwu	2020-06-07 05:02:50.000000	2015-12-05 02:42:04.000000
13	15 Mahendruwaa Airport	Ansan	2001-04-24 00:42:41.000000	2005-06-24 00:20:01.000000

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

```

330
331 select a.airline_id, a.airline_name
332 from airline a
333 where a.airline_id not in (
334     select f.airline_id
335         from flights f
336         where f.sch_departure_time >= date_trunc('month', current_date) + interval '1 month'
337         and f.sch_departure_time < date_trunc('month', current_date) + interval '2 month'
338 );
339
340
  
```

Services

Output postgres.public.airline

airline_id	airline_name
1	Alitalia
2	Aero New Zealand
3	Qantas
4	Southwest Aerolines
5	Aero India
6	All Nippon Aeroways
7	Emirates
8	LATAM Aerolines
9	Thai Aereways
10	Aero India
11	Southwest Aerolines
12	Southwest Aerolines
13	Iberia
14	Aero France
15	American Aerolines

404 rows

Database Consoles > db_international_airports > console [db_international_airports]

338.7 LF UTF-8 4 spaces ⌂ ⌂

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

```

341
342 select
343     p.passenger_id,
344     p.first_name,
345     p.last_name,
346     p.passport_number,
347     f.flight_no,
348     b.flight_id
349     from passengers p
350     join booking b
351     1<-1:n: on p.passenger_id = b.passenger_id
352     join flights f
353     l:n->1: on f.flight_id = b.flight_id
354     where f.flight_no = 'AA9129';
355
  
```

Services

Output Result 8

passenger_id	first_name	last_name	passport_number	flight_no	flight_id
50	Sam	Morten	872-605-22-31	AA9129	2
110	Korella	Safont	337-864-29-38	AA9129	2
174	Windham	Rehn	651-634-33-73	AA9129	2
34	Lurleen	Arzu	214-137-98-26	AA9129	1
45	Court	Portman	893-940-15-82	AA9129	2
141	Maighdiln	Abbatini	822-753-31-01	AA9129	2
128	Rickey	Calderon	478-396-96-34	AA9129	1
123	Emmalee	Carling	186-267-54-38	AA9129	1
319	Beatrix	Rudram	22-526-5588	AA9129	2
12	Denni	Liston	273-947-95-15	AA9129	1

10 rows

Database Consoles > db_international_airports > console [db_international_airports]

354.30 LF UTF-8 4 spaces ⌂ ⌂

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

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

- Connections:** lab1-db, main
- Current Schema:** db_international_airports
- Query Editor:** Tx: Auto, Playground
- Query:**

```
357 ✓ select f.flight_id,
358     f.flight_no,
359     max(b.ticket_price) as max_price,
360     min(b.ticket_price) as min_price,
361     avg(b.ticket_price) as average_price,
362     sum(ticket_price) as total_price
363
364     from flights f
365     join booking b 1<->1..n: on f.flight_id = b.flight_id
366     group by f.flight_id, f.flight_no;
```

- Services:** Services
- Output:** Result 9
- Table Data:** The results of the query are displayed in a table format.

	flight_id	flight_no	max_price	min_price	average_price	total_price
1	1	AA9129	49.99	3.49	27.865	111.46
2	2	AA9129	89.99	4.29	25.2086666666666667	151.24
3	3	<null>	24.99	3.69	14.34	28.68
4	4	<null>	34.99	2.99	22.6566666666666667	67.97
5	7	<null>	99.99	12.99	56.49	112.98
6	9	<null>	4.99	4.99	4.99	4.99
7	11	<null>	39.99	39.99	39.99	39.99
8	12	<null>	2.99	0.75	1.87	3.74
9	15	<null>	5.99	5.99	5.99	5.99
10	16	<null>	79.99	1.49	34.8233333333333333	104.47
11	17	<null>	34.99	4.99	16.99	50.97
12	18	<null>	8.49	8.49	8.49	8.49
13	19	<null>	49.99	49.99	49.99	49.99
14	20	<null>	29.99	228 rows	29.99	29.99
15	21	<null>	8.99	8.99	8.99	8.99

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.

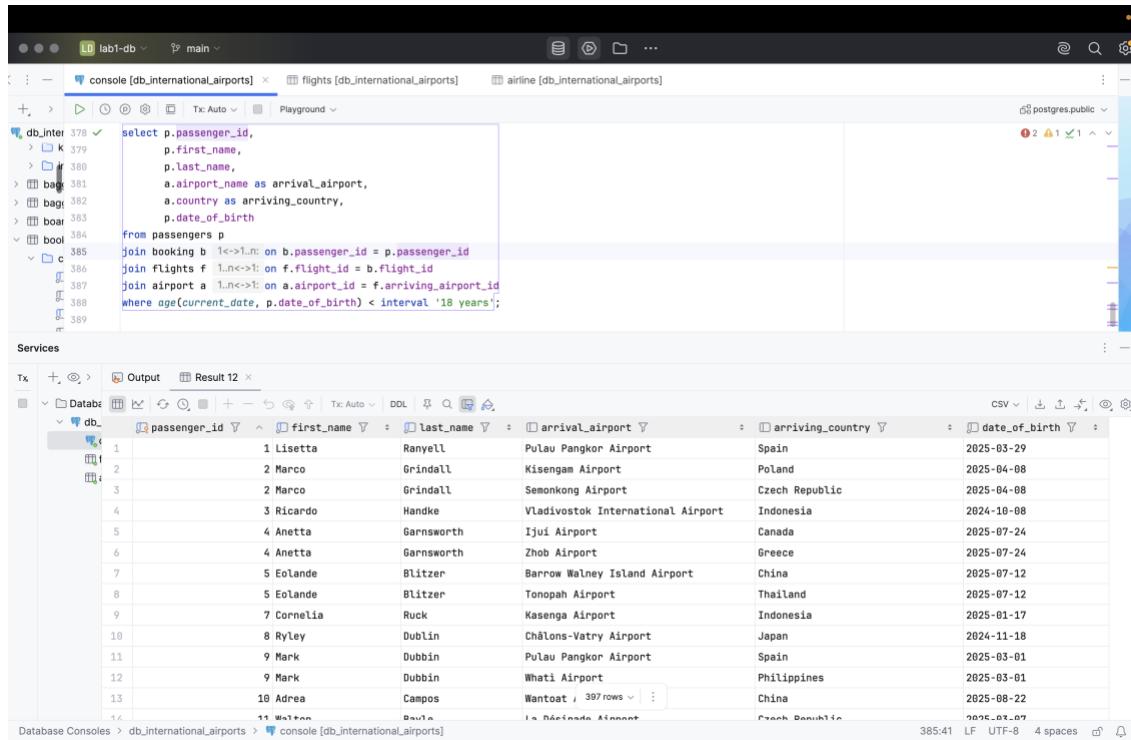
The screenshot shows a PostgreSQL database interface with two tabs: 'SQL' and 'Results'. The SQL tab contains the following query:

```
select f.flight_id,
       a.airline_name,
       dep.airport_name as departure_airport,
       arr.airport_name as arrival_airport,
       arr.country as destination_country
  from flights f
 join airline a 1..n<->1: on f.airline_id = a.airline_id
 join airport dep 1..n<->1: on f.departing_airport_id = dep.airport_id
 join airport arr 1..n<->1: on f.arriving_airport_id = arr.airport_id
 where arr.country = 'Japan';
```

The Results tab displays the output of the query, showing 8 rows of flight information:

Flight_id	airline_name	departure_airport	arrival_airport	destination_country
1	8 Southwest Airlines	Wantat Airport	Julius Nyerere International Airport	Japan
2	24 United Aerolines	Osmannabad Airport	Châlons-Vatry Airport	Japan
3	52 Iberia	Carutapera Airport	Châlons-Vatry Airport	Japan
4	68 South African Aeroways	Zhoushuizi Airport	Ambatolhy Airport	Japan
5	143 SAS Scandinavian	Valdosta Regional Airport	Châlons-Vatry Airport	Japan
6	239 Ethiopian Aerolines	Mbarara Airport	Mushaf Air Base	Japan
7	299 Korean Aero	Seward Airport	Sikasso Airport	Japan
8	316 Alitalia	Neyveli Airport	Gasuke Airport	Japan

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



The screenshot shows the DBeaver interface with a PostgreSQL connection named 'lab1-db'. In the top-left pane, there is a tree view of database objects under 'db_international_airports'. The main query editor window contains the following SQL code:

```

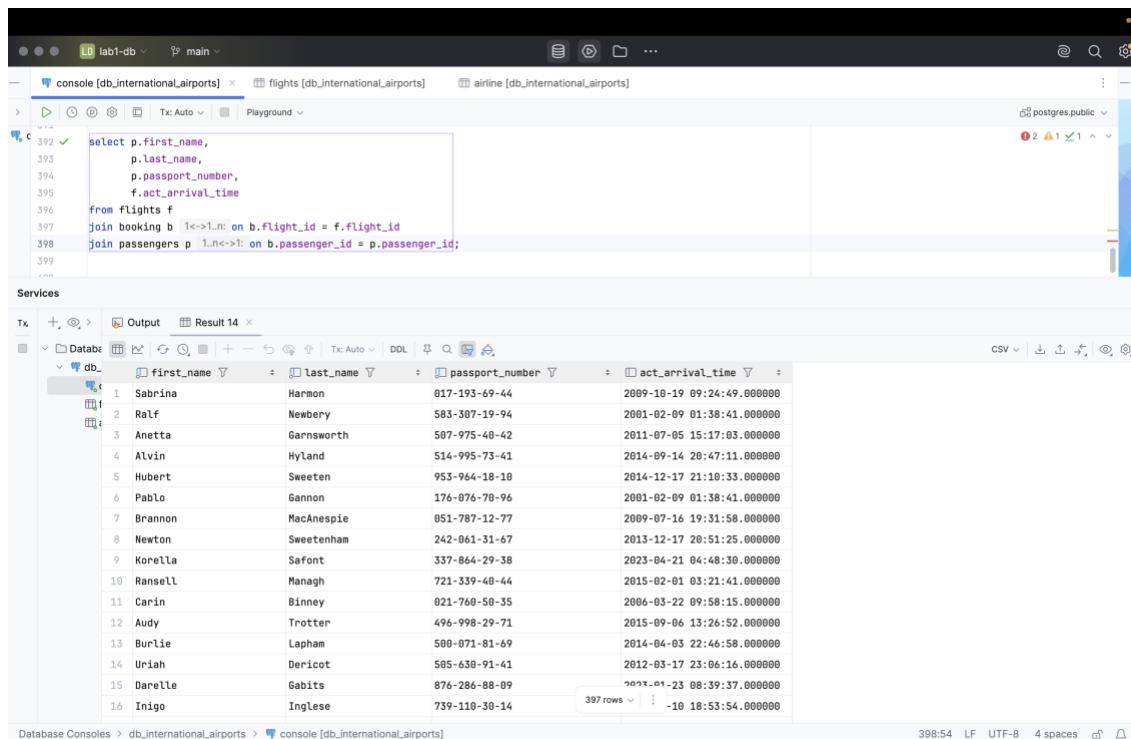
select p.passenger_id,
       p.first_name,
       p.last_name,
       a.airport_name as arrival_airport,
       a.country as arriving_country,
       p.date_of_birth
  from passengers p
 join booking b 1<->1..n on b.passenger_id = p.passenger_id
 join flights f 1..n->1..1 on f.flight_id = b.flight_id
 join airport a 1..n->1..1 on a.airport_id = f.arriving_airport_id
 where age(current_date, p.date_of_birth) < interval '18 years';

```

The results pane below displays a table with 14 rows of passenger information, including their first name, last name, arrival airport, arriving country, and date of birth. The table has columns: passenger_id, first_name, last_name, arrival_airport, arriving_country, and date_of_birth.

	first_name	last_name	arrival_airport	arriving_country	date_of_birth
1	Lisetta	Ranyell	Pulau Pangkor Airport	Spain	2025-03-29
2	Marco	Grindall	Kisengam Airport	Poland	2025-04-08
3	Marco	Grindall	Semongkong Airport	Czech Republic	2025-04-08
4	Ricardo	Handke	Vladivostok International Airport	Indonesia	2024-10-08
5	Anetta	Garnsworth	Ijui Airport	Canada	2025-07-24
6	Anetta	Garnsworth	Zhob Airport	Greece	2025-07-24
7	Eolande	Blitzer	Barrow Walney Island Airport	China	2025-07-12
8	Eolande	Blitzer	Tonopah Airport	Thailand	2025-07-12
9	Cornelia	Ruck	Kasenga Airport	Indonesia	2025-01-17
10	Ryley	Dublin	Châlons-Vatry Airport	Japan	2024-11-18
11	Mark	Dubbin	Pulau Pangkor Airport	Spain	2025-03-01
12	Mark	Dubbin	Whati Airport	Philippines	2025-03-01
13	Adrea	Campos	Wantoat	China	2025-08-22
14	Walter	Paula	La Réunion Airport	French Guiana	2025-03-07

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



The screenshot shows the DBeaver interface with a PostgreSQL connection named 'lab1-db'. In the top-left pane, there is a tree view of database objects under 'db_international_airports'. The main query editor window contains the following SQL code:

```

select p.first_name,
       p.last_name,
       p.passport_number,
       f.act_arrival_time
  from flights f
 join booking b 1<->1..n on b.flight_id = f.flight_id
 join passengers p 1..n->1..1 on b.passenger_id = p.passenger_id;

```

The results pane below displays a table with 16 rows of passenger information, including their first name, last name, passport number, and actual arrival time. The table has columns: first_name, last_name, passport_number, and act_arrival_time.

	first_name	last_name	passport_number	act_arrival_time
1	Sabrina	Harmon	017-193-69-44	2009-10-19 09:24:49.000000
2	Ralf	Newberry	583-307-19-94	2001-02-09 01:38:41.000000
3	Anetta	Garnsworth	507-975-48-42	2011-07-05 15:17:03.000000
4	Alvin	Hyland	514-995-73-41	2014-09-14 20:47:11.000000
5	Hubert	Sweeten	953-964-18-18	2014-12-17 21:10:33.000000
6	Pablo	Gannon	176-076-79-96	2001-02-09 01:38:41.000000
7	Brannon	MacAnespie	051-787-12-77	2009-07-16 19:31:58.000000
8	Newton	Sweetenham	242-061-31-67	2013-12-17 20:51:25.000000
9	Korella	Safont	337-864-29-38	2023-04-21 04:48:30.000000
10	Ransell	Managh	721-339-48-44	2015-02-01 03:21:41.000000
11	Carin	Binney	021-760-58-35	2006-03-22 09:58:15.000000
12	Audy	Trotter	496-998-29-71	2015-09-06 13:26:52.000000
13	Burlie	Lapham	500-071-81-69	2014-04-03 22:46:58.000000
14	Uriah	Dericot	505-638-91-41	2012-03-17 23:06:16.000000
15	Darelle	Gabits	876-286-88-09	2022-01-23 08:39:37.000000
16	Inigo	Ingles	739-110-30-14	397 rows

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 PostgreSQL database interface with the following details:

Query:

```
dt: 400 ✓
  select f.flight_id,
         a.airline_name,
         a.airline_country,
         ar.country as airport_country
    from flights f
   join airline a 1..n<->1 on f.airline_id = a.airline_id
   join airport ar 1..n<->1 on f.departing_airport_id = ar.airport_id
  where a.airline_country = ar.country
 group by f.flight_id, a.airline_name, a.airline_country, ar.country;
```

Result:

flight_id	airline_name	airline_country	airport_country
1	Lufthansa	China	China
2	South African Aeroways	China	China
3	South African Aeroways	China	China
4	Emirates	China	China
5	Qantas	Brazil	Brazil
6	Cathay Pacific	China	China
7	Lufthansa	Indonesia	Indonesia
8	Cathay Pacific	China	China
9	Singapore Airlines	Japan	Japan
10	Southwest Airlines	China	China
11	Lufthansa	Indonesia	Indonesia
12	KLM Royal Dutch Aeroline	China	China
13	Avianca	Russia	Russia
14	South African Aeroways	China	China
15	Avianca	Russia	Russia
16	Ethiopian Airlines	China	China