

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

The screenshot shows the DBeaver interface with a PostgreSQL connection. In the SQL Editor, a query is written to select flights from the flights table, joining it with the airline table to filter by airline_id. The results are displayed in a table with columns: flight_id, airline_name, and airline_country. The data shows 190 rows of flight information with their respective airline names and countries.

flight_id	airline_name	airline_country
10	Airline 10	Russia
100	Airline 100	Russia
101	Airline 101	United States
102	Airline 102	Germany
103	Airline 103	France
104	Airline 104	United Kingdom
105	Airline 105	Canada
106	Airline 106	Japan
107	Airline 107	Australia
108	Airline 108	China
109	Airline 109	Brazil
11	Airline 11	United States
110	Airline 110	Russia
111	Airline 111	United States
112	Airline 112	Germany
113	Airline 113	France
114	Airline 114	United Kingdom
115	Airline 115	Canada
...

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

The screenshot shows the DBeaver interface with a PostgreSQL connection. A more complex query is written to select flights, joining them with the airport table to get the departure airport's name, city, country, and name. The results are displayed in a table with columns: flight_id, sch_departure_time, departure_airport, departure_city, departure_country, and airline_name. The data shows 190 rows of flight information with their departure details and the names of the airlines.

flight_id	sch_departure_time	departure_airport	departure_city	departure_country	airline_name
10	2024-01-03 14:00:00.000	Sydney Airport	Sydney	Australia	Airline 10
11	2024-01-03 20:00:00.000	Airport 10	City 10	Russia	Airline 11
12	2024-01-04 02:00:00.000	Airport 11	City 11	United States	Airline 12
13	2024-01-04 08:00:00.000	Airport 12	City 12	Germany	Airline 13
14	2024-01-04 14:00:00.000	Airport 13	City 13	France	Airline 14
15	2024-01-04 20:00:00.000	Airport 14	City 14	United Kingdom	Airline 15
16	2024-01-05 02:00:00.000	Airport 15	City 15	Canada	Airline 16
17	2024-01-05 08:00:00.000	Airport 16	City 16	Japan	Airline 17
18	2024-01-05 14:00:00.000	Airport 17	City 17	Australia	Airline 18
19	2024-01-05 20:00:00.000	Airport 18	City 18	China	Airline 19
20	2024-01-06 02:00:00.000	Airport 19	City 19	Brazil	Airline 20
...	Airline 21

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

```

SELECT
    airline.airline_id,
    airline.airline_code,
    airline.airline_name,
    airline.airline_country,
    flights.flight_id,
    flights.sch_departure_time
FROM
    airline
LEFT JOIN
    flights ON airline.airline_id = flights.airline_id
    AND flights.sch_departure_time >= CURRENT_DATE
    AND flights.sch_departure_time < CURRENT_DATE + INTERVAL '1 month'
WHERE
    flights.airline_id IS NULL
ORDER BY
    airline.airline_name

```

The screenshot shows the Database Navigator interface in DBeaver. The top pane displays the SQL query above. Below it, a results grid shows 190 rows of airline data. The columns are: airline_id, airline_code, airline_name, airline_country, flight_id, and sch_departure_time. The airline names listed are: Airline 10, Airline 100, Airline 101, Airline 102, Airline 103, Airline 104, Airline 105, Airline 106, Airline 107, Airline 108, Airline 109, and Airline 11. The airline codes are mostly A10, with one entry for A11. The airline countries include Russia, United States, Germany, France, United Kingdom, Canada, Japan, Australia, China, Brazil, and United States again. The flight IDs are all null, and the departure times are also null.

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

```

SELECT p.passenger_id, p.first_name, p.last_name, p.date_of_birth, p.gender, p.country_of_citizenship, p.passport_number
FROM passengers p
INNER JOIN booking ON p.passenger_id = booking.passenger_id
INNER JOIN flights f ON booking.flight_id = f.flight_id
LEFT JOIN boarding_pass bp ON booking.booking_id = bp.booking_id
WHERE f.flight_id = 11 AND booking.status != 'CANCELLED'
ORDER BY p.last_name, p.first_name

```

The screenshot shows the Database Navigator interface in DBeaver. The top pane displays the SQL query above. Below it, a results grid shows 1 row of passenger data. The columns are: passenger_id, first_name, last_name, date_of_birth, gender, country_of_citizenship, and passport_number. The passenger listed is Alexander Surname10, born on 1980-01-01, male, from Russia, with passport number RU0000010.

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

```

SELECT f.flight_id,
       f.sch_departure_time,
       arr_airport.airport_name AS departure_airport,
       arr_airport.airport_id AS arrival_airport,
       COUNT(b.booking_id) AS total_bookings,
       ROUND(AVG(b.price), 2) AS average_price,
       SUM(b.price) AS total_revenue,
       MAX(b.price) AS maximum_price,
       MIN(b.price) AS minimum_price
FROM flights f
INNER JOIN booking b ON f.flight_id = b.flight_id
INNER JOIN airport dep_airport ON f.departing_airport_id = dep_airport.airport_id
INNER JOIN airport arr_airport ON f.arriving_airport_id = arr_airport.airport_id
INNER JOIN airline a ON f.airline_id = a.airline_id
GROUP BY
       f.flight_id,
       dep_airport.airport_name,
       arr_airport.airport_name

```

	flight_id	sch_departure_time	departure_airport	arrival_airport	total_bookings	average_price
1	27	2024-01-07 20:00:00.000	Airport 26	Airport 27	1	1,300.34
2	18	2024-01-05 14:00:00.000	Airport 17	Airport 18	1	998.48
3	96	2024-01-25 02:00:00.000	Airport 95	Airport 96	1	1,258.55
4	26	2024-01-07 14:00:00.000	Airport 25	Airport 26	1	747.59
5	16	2024-01-05 02:00:00.000	Airport 15	Airport 16	1	1,036.02
6	93	2024-01-24 08:00:00.000	Airport 92	Airport 93	1	276.84
7	42	2024-01-11 14:00:00.000	Airport 41	Airport 42	1	1,320.23
8	76	2024-01-20 02:00:00.000	Airport 75	Airport 76	1	529.68
9	182	2024-02-15 14:00:00.000	Airport 181	Airport 182	1	712.81
10	143	2024-02-05 20:00:00.000	Airport 142	Airport 143	1	643.74
11	20	2024-01-06 02:00:00.000	Airport 19	Airport 20	1	320.44
12	145	2024-02-06 08:00:00.000	Airport 144	Airport 145	1	211.1

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_id,
       f.sch_departure_time,
       f.sch_arrival_time,
       a.airline_name,
       dep.airport_name AS departure_airport,
       dep.country AS departure_country,
       arr.airport_name AS arrival_airport,
       arr.country AS arrival_country
FROM flights f
INNER JOIN airline a ON f.airline_id = a.airline_id
INNER JOIN airport dep ON f.departing_airport_id = dep.airport_id
INNER JOIN airport arr ON f.arriving_airport_id = arr.airport_id
ORDER BY arr.country

```

	flight_id	sch_departure_time	sch_arrival_time	airline_name	departure_airport	departure_country
1	187	2024-02-16 20:00:00.000	2024-02-17 00:00:00.000	Airline 187	Airport 186	Japan
2	37	2024-01-10 08:00:00.000	2024-01-10 12:00:00.000	Airline 37	Airport 36	Japan
3	137	2024-02-04 08:00:00.000	2024-02-04 12:00:00.000	Airline 137	Airport 136	Japan
4	107	2024-01-27 20:00:00.000	2024-01-28 00:00:00.000	Airline 107	Airport 106	Japan
5	67	2024-01-17 20:00:00.000	2024-01-18 00:00:00.000	Airline 67	Airport 66	Japan
6	167	2024-02-11 20:00:00.000	2024-02-12 00:00:00.000	Airline 167	Airport 166	Japan
7	27	2024-01-07 20:00:00.000	2024-01-08 00:00:00.000	Airline 27	Airport 26	Japan
8	127	2024-02-01 20:00:00.000	2024-02-02 00:00:00.000	Airline 127	Airport 126	Japan
9	47	2024-01-12 20:00:00.000	2024-01-13 00:00:00.000	Airline 47	Airport 46	Japan
10	147	2024-02-06 20:00:00.000	2024-02-07 00:00:00.000	Airline 147	Airport 146	Japan
11	57	2024-01-15 08:00:00.000	2024-01-15 12:00:00.000	Airline 57	Airport 56	Japan
12	157	2024-02-09 08:00:00.000	2024-02-09 12:00:00.000	Airline 157	Airport 156	Japan

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

SQL Query:

```

SELECT
    p.passenger_id,
    p.first_name,
    p.last_name,
    p.date_of_birth,
    DATE_PART('year', AGE(p.date_of_birth)) AS age,
    f.flight_id,
    arr_airport.airport_name AS arrival_airport,
    arr_airport.city AS arrival_city,
    arr_airport.country AS arrival_country,
    f.sch_arrival_time
FROM
    passengers p
INNER JOIN booking b ON p.passenger_id = b.passenger_id
INNER JOIN flights f ON b.flight_id = f.flight_id
INNER JOIN airport arr_airport ON f.arriving_airport_id = arr_airport.airport_id
WHERE
    DATE_PART('year', AGE(p.date_of_birth)) < 18
ORDER BY
    p.date_of_birth DESC

```

Result Grid:

	passenger_id	first_name	last_name	date_of_birth	age	flight_id	arrival_airport	arrival_city	arrival_country
1	50	Natasha	Surname49	2019-01-01	6	50	Airport 50	City	
2	170	Natasha	Surname169	2019-01-01	6	170	Airport 170	City	
3	130	Natasha	Surname129	2019-01-01	6	130	Airport 130	City	
4	90	Natasha	Surname89	2019-01-01	6	90	Airport 90	City	
5	49	James	Surname48	2018-01-01	7	49	Airport 49	City	
6	89	James	Surname88	2018-01-01	7	89	Airport 89	City	
7	129	James	Surname128	2018-01-01	7	129	Airport 129	City	
8	169	James	Surname168	2018-01-01	7	169	Airport 169	City	
9	168	Olga	Surname167	2017-01-01	8	168	Airport 168	City	
10	48	Olga	Surname47	2017-01-01	8	48	Airport 48	City	
11	88	Olga	Surname87	2017-01-01	8	88	Airport 88	City	
12	128	Olga	Surname127	2017-01-01	8	128	Airport 128	City	

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

SQL Query:

```

SELECT
    CONCAT(p.first_name, ' ', p.last_name) AS full_name,
    port_number,
    arr_airport.airport_name AS arrival_airport,
    arr_airport.country AS arrival_country,
    f.act_arrival_time AS actual_arrival_time,
    f.sch_arrival_time AS scheduled_arrival_time,
    CASE
        WHEN f.act_arrival_time > f.sch_arrival_time
        THEN 'DELAYED'
        WHEN f.act_arrival_time < f.sch_arrival_time
        THEN 'EARLY'
        ELSE 'ON TIME'
    END AS arrival_status
FROM
    passengers p
INNER JOIN booking b ON p.passenger_id = b.passenger_id
INNER JOIN flights f ON b.flight_id = f.flight_id
INNER JOIN airport arr_airport ON f.arriving_airport_id = arr_airport.airport_id
ORDER BY
    f.act_arrival_time

```

Result Grid:

	full_name	passport_number	flight_id	arrival_airport	arrival_country	actual_arrival_time
1	Ivan Petrov	RU1234567	1	Heathrow Airport	United Kingdom	2024-01-01 12:10:00.000
2	Maria Ivanova	RU2345678	2	Charles de Gaulle Airport	France	2024-01-01 18:10:00.000
3	John Smith	US3456789	3	Frankfurt Airport	Germany	2024-01-02 00:10:00.000
4	Emma Johnson	UK4567890	4	JFK Airport	United States	2024-01-02 06:10:00.000
5	Hans Mueller	DE5678901	5	LAX Airport	United States	2024-01-02 12:10:00.000
6	Marie Dubois	FR6789012	6	Dubai International Airport	UAE	2024-01-02 18:10:00.000
7	Ahmed Al-Rashid	AE7890123	7	Singapore Changi Airport	Singapore	2024-01-03 00:10:00.000
8	Yuki Tanaka	JP8901234	8	Tokyo Narita Airport	Japan	2024-01-03 06:10:00.000
9	Li Wei	CN9012345	9	Sydney Airport	Australia	2024-01-03 12:10:00.000
10	Sarah Brown	AU0123456	10	Airport 10	Russia	2024-01-03 18:10:00.000
11	Alexander Surname10	RU0000010	11	Airport 11	United States	2024-01-04 00:10:00.000
12	Maria Surname11	RU0000011	12	Airport 12	Germany	2024-01-04 06:10:00.000

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 the DBeaver application interface. On the left, the Database Navigator displays a project named 'Project - General' with various database objects like Bookmarks, Dashboards, Diagrams, and Scripts. In the center, the Connections tab shows a connection to 'DBeaver Sample Database (SQLite)' and 'postgres localhost:5433'. A SQL editor window is open with the following query:

```
select
    f.flight_id,
    f.sch_departure_time,
    f.sch_arrival_time,
    a.airline_name,
    dep.country AS departure_country,
    dep.airport_name AS departure_airport,
    dep.city AS departure_city,
    arr.airport_name AS arrival_airport,
    arr.city AS arrival_city,
    arr.country AS arrival_country
FROM
    flights f
    INNER JOIN airline a ON f.airline_id = a.airline_id
    INNER JOIN airport dep ON f.departing_airport_id = dep.airport_id
    INNER JOIN airport arr ON f.arriving_airport_id = arr.airport_id
WHERE
    a.airline_country = dep.country
ORDER BY
    dep.country
```

Below the query, a results grid is displayed with one row of data:

flight_id	sch_departure_time	sch_arrival_time	airline_name	departure_country	arrival_airport
123	2025-11-05 08:00:00	2025-11-05 10:00:00	Airline A	Country A	Arrival Airport

The bottom status bar indicates 'No data- 0.004s, on 2025-11-05 at 00:27:38'.