

1)

The screenshot shows a database console interface with a dark theme. On the left, the 'Database Explorer' pane shows a tree structure with 'database' expanded, containing 'airport' and 'postgres'. The 'postgres' database is further expanded to show 'information\_schema'. The main console area has a tab labeled 'console' and contains the following SQL code:

```
1 CREATE VIEW flights_spec AS
2 SELECT f.flight_id, f.flight_no, f.scheduled_departure
3 FROM flights f
4 WHERE DATE(f.scheduled_departure) = '2024-01-22';
```

Below the code, the 'Services' pane shows a log of transactions. The first transaction is a connection at [2025-11-19 13:04:10]. The second transaction is the execution of the SQL code above, starting at [2025-11-19 13:04:10] and completing in 56 ms at [2025-11-19 13:04:10]. The status bar at the bottom indicates '4:49 CRLF UTF-8 4 spaces'.

2)

The screenshot shows the same database console interface. The 'Database Explorer' pane now shows a more detailed tree structure, including 'pg\_catalog' and 'public' schemas. Under 'public', there are 'tables' and 'views'. The 'tables' folder is expanded, showing 'airline', 'airport', 'baggage', 'boarding\_pass', 'booking', 'booking\_flight', 'flights', 'passengers', and 'security\_check'. The 'views' folder is also expanded, showing 'views'. The main console area has a tab labeled 'console' and contains the following SQL code:

```
1 CREATE VIEW next_week_bookings AS
2 SELECT
3     b.booking_id,
4     f.flight_id,
5     f.scheduled_departure,
6     f.departure_airport_id
7 FROM booking b
8 JOIN booking_flight bf ON b.booking_id = bf.booking_id
9 JOIN flights f ON bf.flight_id = f.flight_id
10
11 WHERE f.scheduled_departure BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '7 days';
```

Below the code, the 'Services' pane shows a log of transactions. The first transaction is a connection at [2025-11-19 13:08:46]. The second transaction is the execution of the SQL code above, starting at [2025-11-19 13:08:46] and completing in 30 ms at [2025-11-19 13:08:46]. The status bar at the bottom indicates '10:1 CRLF UTF-8 4 spaces'.

3)

The screenshot shows a database console interface with a 'Database Explorer' on the left and a 'console' tab active. The 'Database Explorer' shows a tree structure with 'database' at the top, followed by 'airport', 'postgres', 'information\_schema', 'pg\_catalog', and 'public'. Under 'public', there are 'tables' and 'views'. The 'tables' list includes 'airline', 'airport', 'baggage', 'baggage\_check', 'boarding\_pass', 'booking', 'booking\_flight', 'flights', 'passengers', 'security\_check', and 'views'. The 'views' list includes 'top\_5\_routes'. The 'console' tab shows a SQL query being executed:

```
1 CREATE VIEW top_5_routes AS
2 SELECT
3     dep.airport_name AS departure_airport,
4     arr.airport_name AS arrival_airport,
5     COUNT(b.booking_id) AS booking_count,
6     a.airline_name
7 FROM booking b
8 JOIN booking_flight bf ON b.booking_id = bf.booking_id
9 JOIN flights f ON bf.flight_id = f.flight_id
10 JOIN airport dep ON f.departure_airport_id = dep.airport_id
11 JOIN airport arr ON f.arrival_airport_id = arr.airport_id
12 JOIN airline a ON f.airline_id = a.airline_id
13 GROUP BY dep.airport_name, arr.airport_name, a.airline_name
14 ORDER BY booking_count DESC
15 LIMIT 5;
```

The 'Services' section at the bottom shows the execution details:

```
JOIN airline a ON f.airline_id = a.airline_id
GROUP BY dep.airport_name, arr.airport_name, a.airline_name
ORDER BY booking_count DESC
LIMIT 5
[2025-11-19 13:10:31] completed in 19 ms
```

The status bar at the bottom indicates 'Database Consoles > database > console', '15:9 CRLF UTF-8 4 spaces', and a PostgreSQL icon.

4)

The screenshot shows a database console interface with a 'Database Explorer' on the left and a 'console' tab active. The 'Database Explorer' shows a tree structure with 'database' at the top, followed by 'airport', 'postgres', 'information\_schema', 'pg\_catalog', and 'public'. Under 'public', there are 'tables' and 'views'. The 'tables' list includes 'airline', 'airport', 'baggage', 'baggage\_check', 'boarding\_pass', 'booking', 'booking\_flight', 'flights', 'passengers', 'security\_check', and 'views'. The 'views' list includes 'airline\_flights'. The 'console' tab shows a SQL query being executed:

```
1 CREATE VIEW airline_flights AS
2 SELECT
3     f.flight_id,
4     f.flight_no,
5     f.scheduled_departure,
6     f.departure_airport_id,
7     dep.airport_name AS departure_airport,
8     arr.airport_name AS arrival_airport,
9     a.airline_name
10 FROM flights f
11 JOIN airline a ON f.airline_id = a.airline_id
12 JOIN airport dep ON f.departure_airport_id = dep.airport_id
13 JOIN airport arr ON f.arrival_airport_id = arr.airport_id
14 WHERE a.airline_name = 'IPC';
```

The 'Services' section at the bottom shows the execution details:

```
JOIN airline a ON f.airline_id = a.airline_id
JOIN airport dep ON f.departure_airport_id = dep.airport_id
JOIN airport arr ON f.arrival_airport_id = arr.airport_id
WHERE a.airline_name = 'IPC'
[2025-11-19 13:12:48] completed in 15 ms
```

The status bar at the bottom indicates 'Database Consoles > database > console', '14:28 CRLF UTF-8 4 spaces', and a PostgreSQL icon.

5)

The screenshot shows a database console interface with a 'Database Explorer' on the left and a 'console' tab active. The 'Database Explorer' shows a tree structure with 'database' > 'public' > 'tables' > 'airline' selected. The console displays the following SQL code:

```
1 CREATE OR REPLACE VIEW airline_flights AS
2 SELECT
3     f.flight_id,
4     f.flight_no,
5     f.scheduled_departure,
6     f.departure_airport_id,
7     dep.airport_name AS departure_airport,
8     arr.airport_name AS arrival_airport,
9     a.airline_name
10 FROM flights f
11 JOIN airline a 1.n<->1 ON f.airline_id = a.airline_id
12 JOIN airport dep 1.n<->1 ON f.departure_airport_id = dep.airport_id
13 JOIN airport arr 1.n<->1 ON f.arrival_airport_id = arr.airport_id
14 WHERE a.airline_name = 'IPC' AND f.scheduled_departure BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '7 days';
```

Below the console, the 'Services' section shows the execution of the query, which completed in 18 ms. The status bar at the bottom indicates 'Database Consoles > database > console' and '14:105 CRLF UTF-8 4 spaces'.

6)

The screenshot shows a database console interface with a 'Database Explorer' on the left and a 'console' tab active. The 'Database Explorer' shows a tree structure with 'database' > 'public' > 'tables' > 'airline' selected. The console displays the following SQL code:

```
1 CREATE VIEW delayed_flights AS
2 SELECT
3     f.flight_id,
4     f.flight_no,
5     (f.actual_departure - f.scheduled_departure) AS delay_duration,
6     a.airline_name,
7     dep.airport_name AS departure_airport,
8     arr.airport_name AS arrival_airport
9 FROM flights f
10 JOIN airline a 1.n<->1 ON f.airline_id = a.airline_id
11 JOIN airport dep 1.n<->1 ON f.departure_airport_id = dep.airport_id
12 JOIN airport arr 1.n<->1 ON f.arrival_airport_id = arr.airport_id
13 WHERE f.actual_departure > f.scheduled_departure + INTERVAL '24 hours'
14 OR (f.actual_departure IS NULL AND CURRENT_TIMESTAMP > f.scheduled_departure + INTERVAL '24 hours');
```

Below the console, the 'Services' section shows the execution of the query, which completed in 15 ms. The status bar at the bottom indicates 'Database Consoles > database > console' and '14:104 CRLF UTF-8 4 spaces'.

7)

The screenshot shows a database console interface with a dark theme. On the left, the 'Database Explorer' pane shows a tree structure of the database schema, including 'database', 'airport', 'postgres', 'information\_schema', 'pg\_catalog', and 'public'. The 'public' schema is expanded, showing tables like 'airline', 'airport', 'baggage', 'boarding\_pass', 'booking', 'booking\_flight', 'flights', 'passengers', 'security\_check', and 'views'. The main console area displays SQL code for creating a view and executing a query. The view 'leffler\_thompson\_passengers' is created with columns 'passenger\_id', 'full\_name', 'country\_of\_residence', and 'booking\_id'. The query selects all rows from this view. The results pane at the bottom shows a single row of data.

```

1 CREATE VIEW leffler_thompson_passengers AS
2 SELECT
3     p.passenger_id,
4     CONCAT(p.first_name, ' ', p.last_name) AS full_name,
5     p.country_of_residence,
6     b.booking_id
7 FROM passengers p
8 JOIN booking b ON p.passenger_id = b.passenger_id
9 WHERE b.booking_platform = 'Leffler-Thompson';
10
11 SELECT * FROM leffler_thompson_passengers;

```

passenger_id	full_name	country_of_residence	booking_id	b
157	Philbert Shambroke	China	15	(15,157,Leffler-Thompson,2023-11-10,2023-04-28,Female,2559.77,)

8)

The screenshot shows the same database console interface. The 'Database Explorer' pane is expanded to show 'views'. The main console area displays SQL code for creating a view and executing a query. The view 'top\_10\_visited\_countries' is created with columns 'visited\_country', 'total\_visits', and 'total\_flights'. The query joins the 'booking', 'booking\_flight', 'flights', and 'airport' tables to calculate the total visits and flights for each country, ordered by total visits in descending order and limited to 10 rows. The results pane at the bottom is empty.

```

1 CREATE VIEW top_10_visited_countries AS
2 SELECT
3     arr.country AS visited_country,
4     COUNT(b.booking_id) AS total_visits,
5     COUNT(DISTINCT f.flight_id) AS total_flights
6 FROM booking b
7 JOIN booking_flight bf ON b.booking_id = bf.booking_id
8 JOIN flights f ON bf.flight_id = f.flight_id
9 JOIN airport arr ON f.arrival_airport_id = arr.airport_id -- страна аэропорта назначения
10 GROUP BY arr.country
11 ORDER BY total_visits DESC
12 LIMIT 10;

```

9)

Database Explorer

- database 2
  - airport 0 of 3
  - postgres 3
    - information\_schema
    - pg\_catalog
    - public
      - tables 10
        - airline
        - airport
        - baggage
        - baggage\_check
        - boarding\_pass
        - booking
        - booking\_flight
        - flights
        - passengers
        - security\_check
        - views 6

console

```
1 CREATE VIEW leffler_thompson_passengers AS
2 SELECT
3     p.passenger_id,
4     CONCAT(p.first_name, ' ', p.last_name) AS full_name,
5     p.country_of_residence,
6     b.booking_id,
7     p.gender,
8     p.passport_number
9 FROM passengers p
10 JOIN booking b ON p.passenger_id = b.passenger_id
11 WHERE b.booking_platform = 'Leffler-Thompson';
12 SELECT * FROM leffler_thompson_passengers;
```

Services

Output

passenger_id	full_name	country_of_residence	booking_id	gender	passport_number
157	Philbert Shambroke	China		15 Male	251255686-7

Database Consoles > database > console

10)

Database Explorer

- database 2
  - airport 0 of 3
  - postgres 3
    - information\_schema
    - pg\_catalog
    - public
      - tables 10
      - views 1

console

```
1 DROP VIEW IF EXISTS flights_on_date;
2 DROP VIEW IF EXISTS next_week_bookings;
3 DROP VIEW IF EXISTS top_5_routes;
4 DROP VIEW IF EXISTS airline_flights;
5 DROP VIEW IF EXISTS delayed_flights;
6 DROP VIEW IF EXISTS leffler_thompson_passengers;
7 DROP VIEW IF EXISTS top_10_visited_countries;
```

Services

Output

schemaname	viewname	viewowner
public	next_week_bookings	postgres
public	top_5_routes	postgres

Database Consoles > database > console