

DB DB_lab2 Version control

Database Explorer

postgres@localhost

- Information_schema
- pg_catalog
- public
 - tables 13
 - airline
 - airport
 - airports
 - baggage
 - baggage_ch
 - boarding_pass
 - booking
 - booking_flight
 - flight
 - flights
 - passengers
 - security_check
 - ticket
 - views 8
 - Database Objects
 - Server Objects

Services

Output zhanel.public.flights zhanel.public.view_lt_passengers zhanel.public.view_flights_by_airline

	flight_id	flight_no	scheduled_departure	scheduled_arrival	airline_name	departure_airport
1	13	BR-PE	2024-01-16	2023-06-02	IPC	Figari Sud-Corse A
2	33	MZ-G	2023-09-21	2023-11-29	IPC	Garbaharey Airport
3	36	AU-NT	2023-03-29	2023-05-06	IPC	Elorza Airport
4	73	FR-K	2023-12-26	2023-04-30	IPC	Darchula Airport
5	78	US-VT	2023-08-28	2023-08-02	IPC	Melilla Airport

64 rows

Database Consoles > postgres@localhost > console

9:26 LF UTF-8 4 spaces

The screenshot shows a PostgreSQL database management interface with the following components:

- Database Explorer** (left sidebar): Shows connections to `postgres@localhost` (2), `postgres` (3), `zhanel` (3), and `Server Objects`.
- console** (top center): A code editor window titled "console" showing the creation of a view named `flights_by_date`. The query is:

```
--1
CREATE VIEW flights_by_date AS
SELECT *
FROM flights
WHERE actual_departure = CURRENT_DATE;
```
- Services** (bottom center): A terminal-like window showing the execution of the command. The output is:

```
[2025-11-19 09:40:49] Connected to zhanel
[2025-11-19 09:40:49] zhanel> --1
CREATE VIEW flights_by_date AS
SELECT *
FROM flights
WHERE actual_departure = CURRENT_DATE
[2025-11-19 09:40:49] completed in 22 ms
```

The Services panel displays the following log entries:

```
[2025-11-19 09:40:49] Connected to zhanel
[2025-11-19 09:40:49] zhanel> --1
CREATE VIEW flights_by_date AS
SELECT *
FROM flights
WHERE actual_departure = CURRENT_DATE
[2025-11-19 09:40:49] completed in 22 ms
```

The screenshot shows a PostgreSQL database management interface with the following components:

- Top Bar:** Includes tabs for "DB DB_lab2" and "Version control", and icons for database management.
- Database Explorer:** Shows connections to "postgres@localhost" (2), "postgres" (3), "zhanel" (3), and "Server Objects".
- Console:** Displays a query to create a view named "flights_next_week".

```
--2
CREATE VIEW flights_next_week AS
SELECT *
FROM flights
WHERE actual_departure BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '7 days';
```
- Services:** Shows the execution details of the query.
 - TX: Database
 - postgres@localhost
 - console 28 ms
 - [2025-11-19 09:46:21] zhanel.public> --2
CREATE VIEW flights_next_week AS
SELECT *
FROM flights
WHERE actual_departure BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '7 days'
 - [2025-11-19 09:46:21] completed in 8 ms
- Bottom Status Bar:** Shows "Database Consoles > postgres@localhost > console", and system information like "5:82 LF UTF-8 4 spaces".

Services

A screenshot of the pgAdmin 4 interface. At the top, there's a toolbar with icons for Tx, a plus sign, an eye, a refresh, and arrows. Below the toolbar, a sidebar shows a tree structure. The root node is 'Database', which has a single child node 'postgres@localhost'. This node is expanded, revealing a sub-node 'console' with a timestamp of '26 ms'. A green circular status indicator with a white dot is positioned next to the 'console' node. The main pane below the sidebar is currently empty.

```
[2025-11-19 10:03:22] zhanel.public> --3
CREATE VIEW top5_routes AS
SELECT f.departure_airport_id,
       f.arrival_airport_id,
       COUNT(b.booking_id) AS total_bookings
FROM booking_flight b
JOIN flights f ON b.flight_id = f.flight_id
GROUP BY f.departure_airport_id, f.arrival_airport_id
ORDER BY total_bookings DESC
LIMIT 5

[2025-11-19 10:03:22] completed in 8 ms
```

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

Database Explorer (Left Panel):

- Connected to **postgres@localhost** (2 connections)
- Contains databases: **postgres** (3 objects), **zhanel** (3 objects), and **Server Objects**.

console (Main Panel):

- Toolbar icons: Run, Stop, Refresh, DDL, and a magnifying glass.
- TX: Auto dropdown.
- Playground dropdown.
- Current connection: **zhanel.public**.
- SQL Editor content:

```
--4
CREATE VIEW flights_by_airline AS
SELECT *
FROM flights
WHERE airline_id = 1;
```

Services (Bottom Panel):

- Tx toolbar: New, Stop, Refresh, Close.
- Database list:
 - Database node (selected)
 - PostgreSQL node (selected)
 - Console node (highlighted in blue, 24 ms)
- Output tab: **zhanel.public.flights**
 - Log entry: [2025-11-19 10:07:24] zhanel.public> --4
 - SQL command: CREATE VIEW flights_by_airline AS
 - SQL command: SELECT *
 - SQL command: FROM flights
 - SQL command: WHERE airline_id = 1
 - Log entry: [2025-11-19 10:07:24] completed in 9 ms

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

- Top Bar:** Shows tabs for "DB DB_lab2" and "Version control".
- Database Explorer:** Shows connections to "postgres@localhost" (2 databases: "postgres" and "zhanel") and "Server Objects".
- Console:** Displays the command "DROP VIEW flights_by_airline;" with a green checkmark indicating success.
- Services:** Shows a transaction bar with "Database" selected, and a log output for "zhanel.public.flights" showing the execution of the command and its completion.

```
--5
DROP VIEW flights_by_airline;
```

```
[2025-11-19 10:09:32] zhanel.public> DROP VIEW flights_by_airline
[2025-11-19 10:09:32] completed in 6 ms
```

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

- Top Bar:** Shows tabs for "DB DB_lab2" and "Version control".
- Database Explorer:** Shows connections to "postgres@localhost" (2 databases: "postgres" and "zhanel") and "Server Objects".
- Console:** A code editor window titled "console" containing the following SQL code:

```
--5
CREATE VIEW flights_by_airline AS
SELECT *
FROM flights
WHERE airline_id = 1
AND actual_departure BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '7 days';
```
- Services:** A panel showing the execution of the query in the "Output" tab under "zhanel.public.flights".

```
[2025-11-19 10:17:18] zhanel.public> CREATE VIEW flights_by_airline AS
          SELECT *
          FROM flights
         WHERE airline_id = 1
               AND actual_departure BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '7 days'
[2025-11-19 10:17:18] completed in 4 ms
```

The Services panel shows the execution details of the query:

```
[2025-11-19 10:17:18] zhanel.public> CREATE VIEW flights_by_airline AS
          SELECT *
          FROM flights
         WHERE airline_id = 1
               AND actual_departure BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '7 days'
[2025-11-19 10:17:18] completed in 4 ms
```

At the bottom, the navigation bar indicates the path: "Database Consoles > postgres@localhost > console".

The screenshot shows a PostgreSQL database management interface with the following components:

- Database Explorer** (left sidebar): Shows the database structure:
 - DB_lab2 (selected)
 - Version control
 - Tables: booking, booking_flight, flight, flights (selected)
 - Columns: 14 (under flights)
 - Indexes: 3
 - Passengers
- Services** (bottom left): Shows the transaction state (Tx) and a list of databases:
 - Database (selected)
 - postgres@localhost
- console** (top center): A code editor window with the following SQL code:

```
--6
CREATE VIEW view_delayed_24h AS
SELECT *
FROM flights
WHERE actual_departure IS NOT NULL
AND (actual_departure - scheduled_departure) > 1440;
```
- Output** (bottom right): The execution log:

```
[2025-11-19 10:30:33] zhang.public> --6
CREATE VIEW view_delayed_24h AS
SELECT *
FROM flights
WHERE actual_departure IS NOT NULL
AND (actual_departure - scheduled_departure) > 1440
[2025-11-19 10:30:33] completed in 4 ms
```

DB DB_lab2 Version control

Database Explorer

postgres@localhost

- information_schema
- pg_catalog
- public
 - tables 13
 - airline
 - airport
 - airports
 - baggage
 - baggage_ch
 - boarding_pass
 - booking
 - booking_flight
 - flight
 - flights
 - passengers
 - security_check
 - ticket
 - views 6
- Database Objects
- Server Objects

console

1 --7
2 CREATE VIEW view_lt_passengers AS
3 SELECT
4 p.full_name,
5 p.country_of_citizenship,
6 p.country_of_residence
7 FROM passengers p
8 JOIN booking b ON p.passenger_id = b.passenger_id
9 WHERE b.booking_platform = 'Leffler-Thompson';
11 ✓ SELECT * FROM view_lt_passengers;

Services

Tx + ⌂ ⌂ ×

Database

postgres@localhost

console 398 ms

Output zhanel.public.flights zhanel.public.view_lt_passengers

full_name country_of_citizenship country_of_residence

1 <null> Colombia China

1 row ▾

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

Database Explorer (Left Panel):

- Connected to **DB_lab2**.
- Current schema: **public**.
- Tables listed under **public**: airline, airport, airports, baggage, baggage_ch, boarding_pass, booking, booking_flight, flight, flights, passengers, security_check, ticket.
- Views listed under **public**: view_top10_countries.
- Database Objects and Server Objects are also listed.

console (Top Right Panel):

```
1 ✓ --8
2 CREATE VIEW view_top10_countries AS
3     SELECT
4         a.country,
5             COUNT(f.flight_id) AS total_visits
6     FROM flights f
7     JOIN airport a ON f.arrival_airport_id = a.airport_id
8     GROUP BY a.country
9     ORDER BY total_visits DESC
10    LIMIT 10;
```

Playground (Bottom Right Panel):

full_name	country_of_citizenship	country_of_residence
<null>	Colombia	China

Services (Bottom Left Panel):

- Transactions (Tx).
- Database: **postgres@localhost**.
- Console: **console** (22 ms).

Bottom Status Bar:

Database Consoles > postgres@localhost > **console** 10:10 LF UTF-8 4 spaces

