

Database Navigator

- Filter connections by name
- DBeaver Sample Database (SQLite)
- postgres localhost:5433
 - Databases
 - postgres
 - Schemas
 - public
 - Tables
 - Foreign Tables
 - Views
 - Materialized Views
 - Indexes
 - Functions
 - Sequences
 - Data types
 - Aggregate functions
 - Event Triggers
 - Extensions
 - Storage
 - System Info
 - Roles
 - Administer
 - System Info

SQL Editor

```
create or replace procedure InsertFlight(  
    flight_num VARCHAR(10),  
    dep_airport VARCHAR(50),  
    arr_airport VARCHAR(50),  
    dep_time TIMESTAMP,  
    arr_time TIMESTAMP,  
    status VARCHAR(20),  
    airline VARCHAR(50))  
language plpgsql  
as $$  
begin  
    insert into flights (flight_number, departure_airport, arrival_airport, departure_time, arrival_time,  
        values (flight_num, dep_airport, arr_airport, dep_time, arr_time, status, airline);  
end;  
$$;
```

Statistics 1

Name	Value
Updated Rows	0
Execute time	0.0s
Start time	Wed Dec 03 13:12:35 ALMT 2025
Finish time	Wed Dec 03 13:12:35 ALMT 2025
Query	create or replace procedure InsertFlight(flight_num VARCHAR(10), dep_airport VARCHAR(50), arr_airport VARCHAR(50), dep_time TIMESTAMP, arr_time TIMESTAMP, status VARCHAR(20),

ALMT en Writable Smart Insert 1:1 [476]

gate Search SQL Editor Database Window Help

DBeaver 25.2.5 - <postgres> Script

Commit Rollback Auto

Projects public *<postgres>... <postgres>... boarding_pass <none> dz-4.sql country_of_c...

base (SQLite) 433

bles

ed Views

functions

```
create or replace procedure updateFlightStatus(  
    p_flight_num VARCHAR(10),  
    p_new_status VARCHAR(20)  
)  
language plpgsql  
as $$  
begin  
    update flights set status = p_new_status where flight_number = p_flight_num;  
end;  
$$;
```

Statistics 1

Name	Value
Updated Rows	0
Execute time	0.0s
Start time	Wed Dec 03 13:15:45 ALMT 2025
Finish time	Wed Dec 03 13:15:45 ALMT 2025
Query	create or replace procedure updateFlightStatus(p_flight_num VARCHAR(10), p_new_status VARCHAR(20)) language plpgsql as \$\$ begin

Commit Rollback Auto postgres public@postgres

Projects public *postgres>... <postgres>... boarding_pass <none> dz-4.sql country_of_c...

Database (SQLite) host:5433

```

create or replace procedure listflightsdeparting(
    p_dep_airport varchar(50)
)
language plpgsql
as $$
begin
    select * from flights where departure_airport = p_dep_airport;
end;
$$

```

Statistics 1

Name	Value
Updated Rows	0
Execute time	0.0s
Start time	Wed Dec 03 13:20:51 ALMT 2025
Finish time	Wed Dec 03 13:20:51 ALMT 2025
Query	create or replace procedure listflightsdeparting(p_dep_airport varchar(50)) language plpgsql as \$\$ begin select * from flights where departure_airport = p_dep_airport; end; \$\$

ALMT en Writable Smart Insert 1:1 [186]

Commit Rollback Auto postgres public@postgres

Projects public *postgres>... <postgres>... boarding_pass <none> dz-4.sql country_of_c...

Database (SQLite) host:5433

```

create or replace function averagedelay(p_arr_airport varchar(50))
returns integer
language plpgsql
as $$
declare
    avg_delay integer;
begin
    select avg(extract(epoch from (actual_arrival - scheduled_arrival))/60)
    into avg_delay
    from flights
    where arrival_airport = p_arr_airport and actual_arrival is not null;
    return coalesce(avg_delay, 0);
end;
$$

```

Statistics 1

Name	Value
Updated Rows	0
Execute time	0.0s
Start time	Wed Dec 03 13:24:15 ALMT 2025
Finish time	Wed Dec 03 13:24:15 ALMT 2025
Query	create or replace function averagedelay(p_arr_airport varchar(50)) returns integer language plpgsql as \$\$ declare avg_delay integer; begin select avg(extract(epoch from (actual_arrival - scheduled_arrival))/60) into avg_delay from flights where arrival_airport = p_arr_airport and actual_arrival is not null; return coalesce(avg_delay, 0); end; \$\$

Commit (^4) public *<postgres>... X <postgres> ... boarding_pass <none> dz-4.sql country_of_c...

```

create or replace procedure listpassengers(
    p_flight_num varchar(10)
)
language plpgsql
as $$
begin
    select p.* from passengers p
    join bookings b on p.passenger_id = b.passenger_id
    where b.flight_number = p_flight_num;
end;
$$;

```

Statistics 1 X

Name	Value
Updated Rows	0
Execute time	0.0s
Start time	Wed Dec 03 13:26:52 ALMT 2025
Finish time	Wed Dec 03 13:26:52 ALMT 2025
Query	create or replace procedure listpassengers(p_flight_num varchar(10)) language plpgsql as \$\$ begin select p.* from passengers p

SQL Commit Rollback Auto postgres public@postgres <postgres> ... boarding_pass <none> dz-4.sql country_of_c...

```

create or replace procedure frpassenger()
language plpgsql
as $$
begin
    select p.*, count(b.flight_number) as flight_count
    from passengers p
    join bookings b on p.passenger_id = b.passenger_id
    group by p.passenger_id
    order by count(b.flight_number) desc
    limit 1;
end;
$$;

```

Statistics 1 X

Name	Value
Updated Rows	0
Execute time	0.0s
Start time	Wed Dec 03 13:27:20 ALMT 2025
Finish time	Wed Dec 03 13:27:20 ALMT 2025
Query	create or replace procedure frpassenger() language plpgsql as \$\$ begin select p.*, count(b.flight_number) as flight_count from passengers p join bookings b on p.passenger_id = b.passenger_id

jects public *<postgres>... X <postgres> ... boarding_pass <none> dz-4.sql country_of_c...

```
create or replace procedure flightsdelayedover24()
language plpgsql
as $$
begin
select * from flights
where extract(epoch from (actual_departure - scheduled_departure))/3600 > 24;
end;
$$;
```

Statistics 1 X

Name	Value
Updated Rows	0
Execute time	0.0s
Start time	Wed Dec 03 13:27:43 ALMT 2025
Finish time	Wed Dec 03 13:27:43 ALMT 2025
Query	create or replace procedure flightsdelayedover24() language plpgsql as \$\$ begin select * from flights where extract(epoch from (actual_departure - scheduled_departure))/3600 > 24; end;

ALMT en Writable Smart Insert 9 : 1 : 197

ommit Rollback Auto postgres public@postgres dz-4.sql country_of_c...

```
create or replace procedure averageticketprice(
p_flight_num varchar(10)
)
language plpgsql
as $$
begin
select avg(price) as avg_price
from tickets
where flight_number = p_flight_num;
end;
$$;
```

Statistics 1 X

Name	Value
Updated Rows	0
Execute time	0.0s
Start time	Wed Dec 03 13:28:55 ALMT 2025
Finish time	Wed Dec 03 13:28:55 ALMT 2025
Query	create or replace procedure averageticketprice(p_flight_num varchar(10)) language plpgsql as \$\$ begin select avg(price) as avg_price

Connections - General connections ALMT en

The screenshot shows the pgAdmin interface. On the left is the Object Explorer with a tree view of the database structure. The main pane displays a SQL query in the Query Tool:

```

create or replace procedure highestticketpriceflight()
language plpgsql
as $$
begin
select f.flight_number, f.departure_airport, f.arrival_airport, max(t.price) as max_price
from flights f
join tickets t on f.flight_number = t.flight_number
group by f.flight_number, f.departure_airport, f.arrival_airport
order by max(t.price) desc
limit 1;
end;
$$;

```

Below the query editor, the Statistics window is open, showing the following execution details:

Name	Value
Updated Rows	0
Execute time	0.0s
Start time	Wed Dec 03 13:29:07 ALMT 2025
Finish time	Wed Dec 03 13:29:07 ALMT 2025
Query	create or replace procedure highestticketpriceflight() language plpgsql as \$\$ begin select f.flight_number, f.departure_airport, f.arrival_airport, max(t.price) as max_price from flights f join tickets t on f.flight_number = t.flight_number

At the bottom of the interface, there is a status bar showing "ALMT en Writable", "Smart Insert", and a progress indicator "13 : 1 : 375".