

DBeaver 25.2.5 - <postgres> Script

```

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 $$ 
begin
    insert into flights (flight_number, departure_airport, arrival_airport, departure_time, arrival_time, status, airline)
    values (flight_num, dep_airport, arr_airport, dep_time, arr_time, status, airline);
end;
$$;

```

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), airline VARCHAR(50)) language plpgsql as \$\$ begin insert into flights (flight_number, departure_airport, arrival_airport, departure_time, arrival_time, status, airline) values (flight_num, dep_airport, arr_airport, dep_time, arr_time, status, airline); end; \$\$;

DBeaver 25.2.5 - <postgres> Script

```

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 update flights set status = p_new_status where flight_number = p_flight_num; end; \$\$;

DBVisual 2020 - PostgreSQL Script

Auto postgres public@postgres boarding_pass <none> dz-4.sql country_of_c... ↻

```

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] :

DBVisual 2020 - PostgreSQL Script

Auto postgres public@postgres boarding_pass <none> dz-4.sql country_of_c... ↻

```

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... ??

```

se (SQLite)
3

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	<pre> 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; \$\$;</pre>

SQL Commit Rollback T Auto postgres public@postgres D B S ?

Navi... X Projects D

Sample Database (SQLite)
localhost:5433

```

public * <postgres>... X <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	<pre> 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; \$\$;</pre>

public

```

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

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;

Auto

postgres

public

```

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

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

The screenshot shows the pgAdmin interface with a database connection to a PostgreSQL database named 'dz-4'. The current schema is 'public'. A new procedure named 'highestticketpriceflight' is being created. The code is as follows:

```
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 code, there is a 'Statistics' tab showing the following results:

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 group by f.flight_number, f.departure_airport, f.arrival_airport order by max(t.price) desc limit 1; end; \$\$;