



















pgAdmin 4 Object Tools Edit View Window Help

pgAdmin 4

Object Explorer

- Languages
- Publications
- Schemas (1)
  - public
    - Aggregates
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Operators
    - Procedures
    - Sequences
    - Tables (10)
      - airline
      - airport
      - baggage
      - baggage\_check
      - boarding\_pass
      - booking
      - booking\_flight
      - flights
      - passengers
      - security\_check
    - Trigger Functions
    - Types
    - Views
  - Subscriptions
  - Login/Group Roles
  - Tablespaces

techdat/postgres@PostgreSQL 17\*

Query Query History

```
1  ---9
2  create or replace procedure avg_tick(p_flight_no varchar)
3  language plpgsql
4  as $$
5  declare
6      r numeric;
7  begin
8      select avg(b.price) into r
9      from booking b join booking_flight bf on b.booking_id=bf.booking_id
10     join flights f on bf.flight_id=f.flight_id where f.flight_no=p_flight_no;
11     raise notice 'Average price for flight %:%',p_flight_no,r;
12 end;
13 $$
14
15 call avg_tick('RU-KR');
```

Data Output Messages Notifications

NOTICE: Average price for flight RU-KR:4094.13000000000000000000  
CALL

Query returned successfully in 48 msec.

Total rows: Query complete 00:00:00.048

✓ Query returned successfully in 48 msec.  
LF Ln 15, Col 24

Mon 1 Dec 17:07:03

pgAdmin 4 Object Tools Edit View Window Help

techdat/postgres@PostgreSQL 17\* x public.booking/tec... x public.booking\_flg... x public.flights/tec...

Object Explorer

- Languages
- Publications
- Schemas (1)
  - public
    - Aggregates
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Operators
    - Procedures
    - Sequences
    - Tables (10)
      - airline
      - airport
      - baggage
      - baggage\_check
      - boarding\_pass
      - booking
      - booking\_flight
      - flights
      - passengers
      - security\_check
    - Trigger Functions
    - Types
    - Views
- Subscriptions
- Login/Group Roles
- Tablespaces

Query Query History

```
1  ---10
2  CREATE OR REPLACE PROCEDURE most_expensive_flight_simple()
3  LANGUAGE plpgsql
4  AS $$
5  declare
6      r varchar;
7  BEGIN
8
9      select f.flight_no into r
10     FROM flights f
11    JOIN booking_flight bf ON f.flight_id = bf.flight_id
12    JOIN booking b ON bf.booking_id = b.booking_id
13   ORDER BY b.price DESC
14   LIMIT 1;
15   RAISE NOTICE 'Most expensive flight:%',r;
16 END;
17 $$;
18
19 CALL most_expensive_flight_simple();
20
```

Data Output Messages Notifications

NOTICE: Most expensive flight:NP-SA  
CALL

Query returned successfully in 49 msec.

Total rows: Query complete 00:00:00.049 LF Ln 20, Col 1

Mon 1 Dec 17:10:13