

public.passengers... x database_subj/po... x database_subj/po... x public.booking/dat... x database_subj/postgres@PostgreSQL 17* x

PostgreSQL FastAPI (2)
Local PostgreSQL
Databases (11)
book_app
database_subj
databaseprac1
databases_sj
mydb
postgres
practise_sql
share_recipe
taskmanager_mol
test_booking_db
todo_db
Login/Group Roles
Tablespaces
PostgreSQL 17

database_subj/postgres@PostgreSQL 17

Query Query History Scratch Pad

```
1 CREATE OR REPLACE PROCEDURE insert_new_flight(  
2     p_flight_id INT,  
3     p_flight_no VARCHAR,  
4     p_scheduled_departure DATE,  
5     p_scheduled_arrival DATE,  
6     p_departure_airport_id INT,  
7     p_arrival_airport_id INT,  
8     p_departing_gate VARCHAR,  
9     p_arriving_gate VARCHAR,  
10    p_airline_id INT,  
11    p_status VARCHAR,  
12    p_actual_departure DATE,  
13    p_actual_arrival DATE  
14 )  
15 LANGUAGE plpgsql  
16 AS $$  
17 BEGIN  
18     INSERT INTO flights(  
19         flight_id,  
20         flight_no,  
21         scheduled_departure,  
22         scheduled_arrival,
```

Data Output Messages Notifications

CREATE PROCEDURE

Query returned successfully in 78 msec.

✓ Query returned successfully in 78 msec. ✕

Total rows: Query complete 00:00:00.078 LF Ln 51, Col 4

PostgreSQL FastAPI (2)

Local PostgreSQL

Databases (11)

book_app

database_subj

databaseprac1

databases_sj

mydb

postgres

practise_sql

share_recipe

taskmanager_mol

test_booking_db

todo_db

Login/Group Roles

Tablespaces

PostgreSQL 17

public.passengers...

database_subj/po...

database_subj/po...

public.booking/dat...

database_subj/postgres@PostgreSQL 17*

database_subj/postgres@PostgreSQL 17

Query

Query History

Scratch Pad

```
1 CREATE OR REPLACE PROCEDURE update_flight_status(  
2   p_flight_id INT,  
3   p_new_status VARCHAR  
4 )  
5 LANGUAGE plpgsql  
6 AS $$  
7 BEGIN  
8   IF NOT EXISTS (SELECT 1 FROM flights WHERE flight_id = p_flight_id) THEN  
9     RAISE EXCEPTION 'Flight with id % does not exist', p_flight_id;  
10  END IF;  
11  
12  UPDATE flights  
13  SET  
14    status = p_new_status,  
15    update_at = CURRENT_DATE  
16  WHERE flight_id = p_flight_id;  
17  
18 END;  
19 $$;
```

Data Output

Messages

Notifications

CREATE PROCEDURE

Query returned successfully in 78 msec.

Total rows:

Query complete 00:00:00.078

LF Ln 5, Col 17

✓ Query returned successfully in 78 msec. ✕

PostgreSQL FastAPI (2)

Local PostgreSQL

Databases (11)

book_app

database_subj

databaseprac1

databases_sj

mydb

postgres

practise_sql

share_recipe

taskmanager_mol

test_booking_db

todo_db

Login/Group Roles

Tablespaces

PostgreSQL 17

database_subj/postgres@PostgreSQL 17

Query

Query History

Scratch Pad

```
1 CREATE OR REPLACE PROCEDURE list_flights_from_airport(p_airport_id INT)
2 LANGUAGE plpgsql
3 AS $$
4 BEGIN
5     SELECT *
6     FROM flights
7     WHERE departure_airport_id = p_airport_id;
8 END;
9 $$;
```

Data Output

Messages

Notifications

CREATE PROCEDURE

Query returned successfully in 42 msec.

Total rows: Query complete 00:00:00.042

LF Ln 9, Col 4

✓ Query returned successfully in 42 msec. ✕

PostgreSQL FastAPI (2)
Local PostgreSQL
Databases (11)
book_app
database_subj
databaseprac1
databases_sj
mydb
postgres
practise_sql
share_recipe
taskmanager_mol
test_booking_db
todo_db
Login/Group Roles
Tablespaces
PostgreSQL 17

database_subj/postgres@PostgreSQL 17

Query

Query History

Scratch Pad

```
1 CREATE OR REPLACE FUNCTION average_delay_at_airport(p_airport_id INT)
2 RETURNS NUMERIC
3 LANGUAGE plpgsql
4 AS $$
5 DECLARE
6     avg_delay NUMERIC;
7 BEGIN
8     SELECT AVG(actual_arrival - scheduled_arrival)
9     INTO avg_delay
10    FROM flights
11   WHERE arrival_airport_id = p_airport_id
12         AND actual_arrival IS NOT NULL;
13
14     RETURN avg_delay;
15 END;
16 $$;
```

Data Output

Messages

Notifications

CREATE FUNCTION

Query returned successfully in 41 msec.

Total rows:

Query complete 00:00:00.041

LF Ln 8, Col 53

✓ Query returned successfully in 41 msec. ✕

Schemas (1)
public
Aggreg
Collatic
Domain
FTS Co
FTS Dic
FTS Pa
FTS Te
Foreign
Funcit
Materi
Operat
Proced
Sequer
Tables
airlin
airpo
bagg
bagg
boar
bool
bool
flight
C
C
In
R
R
Ti
pass
secl
Trigger
Types
Views
Subscription

database_subj/postgres@PostgreSQL 17
Query
Query History
Scratch Pad
1 CREATE OR REPLACE PROCEDURE list_passengers_by_flight(
2 p_flight_no VARCHAR)
3 LANGUAGE plpgsql
4 AS \$\$
5 BEGIN
6 SELECT p.passenger_id,
7 p.first_name,
8 p.last_name,
9 p.date_of_birth,
10 p.gender,
11 p.passport_number
12 FROM passengers p
13 JOIN booking b ON p.passenger_id = b.passenger_id
14 JOIN booking_flight bf ON b.booking_id = bf.booking_id
15 JOIN flights f ON bf.flight_id = f.flight_id
16 WHERE f.flight_no = p_flight_no
17 ORDER BY p.last_name, p.first_name;
18 END;
19 \$\$;

Data Output Messages Notifications
CREATE PROCEDURE
Query returned successfully in 52 msec.

Total rows: Query complete 00:00:00.052
LF Ln 18, Col 5

database_subj/postgres@PostgreSQL 17
No limit
Query
Query History
Scratch Pad
1 CREATE OR REPLACE PROCEDURE list_passengers_by_flight(
2 p_flight_no VARCHAR)
3 LANGUAGE plpgsql
4 AS \$\$
5 BEGIN
6 SELECT p.passenger_id,
7 p.first_name,
8 p.last_name,
9 p.date_of_birth,
10 p.gender,
11 p.passport_number
12 FROM passengers p
13 JOIN booking b ON p.passenger_id = b.passenger_id
14 JOIN booking_flight bf ON b.booking_id = bf.booking_id
15 JOIN flights f ON bf.flight_id = f.flight_id
16 WHERE f.flight_no = p_flight_no
17 ORDER BY p.last_name, p.first_name;
18 END;
19 \$\$;

Data Output Messages Notifications
CREATE PROCEDURE
Query returned successfully in 52 msec.

Total rows: Query complete 00:00:00.052
LF Ln 18, Col 5

Schemas (1)

- public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parser
 - FTS Templates
 - Foreign Data Wrappers
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables
 - airline
 - airport
 - baggage
 - baggage
 - board
 - boolean
 - boolean
 - flight
 - Views

database_subj/postgres@PostgreSQL 17

Query

Query History

Execute script

FS

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

CREATE OR REPLACE PROCEDURE passeng

LANGUAGE plpgsql

AS \$\$

BEGIN

SELECT p.passenger_id,

p.first_name,

p.last_name,

COUNT(bf.flight_id) AS flights_taken

FROM passengers p

JOIN booking b ON p.passenger_id = b.passenger_id

JOIN booking_flight bf ON b.booking_id = bf.booking_id

GROUP BY p.passenger_id, p.first_name, p.last_name

ORDER BY flights_taken DESC

LIMIT 1;

END;

\$\$;

Data Output

Messages

Notifications

CREATE PROCEDURE

Query returned successfully in 44 msec.

Total rows:

Query complete 00:00:00.044

✓ Query returned successfully in 44 msec. ✕

LF Ln 16, Col 4

Schemas (1)
public
Aggreg
Collatic
Domain
FTS Co
FTS Dic
FTS Pa
FTS Te
Foreign
Function
Material
Operat
Proced
Sequer
Tables
airlin
airpo
bagg
bagg
boar
bool
bool
flight
C
C
In
R
R
Ti
pass
sect
Trigger
Types
Views
Subscription

database_subj/postgres@PostgreSQL 17
Query
Query History
Scratch Pad
1 CREATE OR REPLACE PROCEDURE flights_delayed_over_24h()
2 LANGUAGE plpgsql
3 AS \$\$
4 BEGIN
5 SELECT flight_id,
6 flight_no,
7 scheduled_departure,
8 actual_departure,
9 scheduled_arrival,
10 actual_arrival,
11 (actual_arrival - scheduled_arrival) AS delay_days
12 FROM flights
13 WHERE actual_arrival IS NOT NULL
14 AND (actual_arrival - scheduled_arrival) > 1 -- 24 hours
15 ORDER BY delay_days DESC;
16 END;
17 \$\$;

Data Output
Messages
Notifications
CREATE PROCEDURE
Query returned successfully in 43 msec.

Total rows: Query complete 00:00:00.043

database_subj/postgres@PostgreSQL 17
database_subj/postgres@PostgreSQL 17
public.booking/dat...
public.flights/data...
LF Ln 17, Col 4

✓ Query returned successfully in 43 msec. ✕

Schemas (1)

- public
 - Aggreg
 - Collatic
 - Domain
 - FTS Co
 - FTS Dic
 - FTS Pa
 - FTS Te
 - Foreign
 - Funcio
 - Material
 - Operat
 - Proced
 - Sequer
 - Tables
 - airli
 - airp
 - bag
 - bag
 - boar
 - bool
 - bool
 - flight

database_subj/postgres@PostgreSQL 17

Query

Query History

Scratch Pad

```
1 CREATE OR REPLACE FUNCTION count_flights_per_airline()
2 RETURNS TABLE (
3     airline_id INT,
4     airline_name VARCHAR,
5     flights_count INT
6 )
7 LANGUAGE plpgsql
8 AS $$
9 BEGIN
10     RETURN QUERY
11     SELECT a.airline_id,
12            a.airline_name,
13            COUNT(f.flight_id) AS flights_count
14     FROM airline a
15     LEFT JOIN flights f ON a.airline_id = f.airline_id
16     GROUP BY a.airline_id, a.airline_name
17     ORDER BY flights_count DESC;
18 END;
19 $$;
```

Data Output

Messages

Notifications

CREATE FUNCTION

Query returned successfully in 51 msec.

Total rows:

Query complete 00:00:00.051

✓ Query returned successfully in 51 msec. ✕

LF Ln 19, Col 4

Schemas (1)

- public
 - Aggreg
 - Collatic
 - Domain
 - FTS Co
 - FTS Dic
 - FTS Pa
 - FTS Te
 - Foreign
 - Funcit
 - Materia
 - Operat
 - Proced
 - Sequer
 - Tables
 - airlin
 - airpo
 - bagg
 - bagg
 - boar
 - bool
 - bool
 - flight
 - C
 - C
 - In
 - R
 - R
 - Ti
 - pass
 - secl
 - Trigger
 - Types
 - Views
 - Subscription

database_subj/postgres@PostgreSQL 17

Query

Query History

Scratch Pad

```
1 CREATE OR REPLACE PROCEDURE average_ticket_price(p_flight_no VARCHAR)
2 LANGUAGE plpgsql
3 AS $$
4 BEGIN
5     SELECT f.flight_no,
6           AVG(b.price) AS avg_ticket_price
7     FROM flights f
8    JOIN booking_flight bf ON f.flight_id = bf.flight_id
9    JOIN booking b ON bf.booking_id = b.booking_id
10   WHERE f.flight_no = p_flight_no
11   GROUP BY f.flight_no;
12 END;
13 $$;
```

Data Output

Messages

Notifications

```
CREATE PROCEDURE
Query returned successfully in 39 msec.
```

Total rows: 1

Query complete 00:00:00.039

LF Ln 13, Col 4

✓ Query returned successfully in 39 msec. ✕

Schemas (1)

- public
 - Aggreg
 - Collatic
 - Domain
 - FTS Co
 - FTS Dic
 - FTS Pa
 - FTS Te
 - Foreign
 - Funcit
 - Materia
 - Operat
 - Proced
 - Sequer
 - Tables
 - airlin
 - airpo
 - bagg
 - bagg
 - boar
 - bool
 - bool
 - flight

database_subj/postgres@PostgreSQL 17

Query

Query History

Scratch Pad

```
1 CREATE OR REPLACE PROCEDURE most_expensive_flight()
2 LANGUAGE plpgsql
3 AS $$
4 BEGIN
5     SELECT f.flight_no,
6           dep.airport_name AS departure_airport,
7           arr.airport_name AS arrival_airport,
8           MAX(b.price) AS max_ticket_price
9     FROM flights f
10    JOIN booking_flight bf ON f.flight_id = bf.flight_id
11    JOIN booking b ON bf.booking_id = b.booking_id
12    JOIN airport dep ON f.departure_airport_id = dep.airport_id
13    JOIN airport arr ON f.arrival_airport_id = arr.airport_id
14   GROUP BY f.flight_no, dep.airport_name, arr.airport_name
15  ORDER BY max_ticket_price DESC
16  LIMIT 1;
17 END;
18 $$;
```

Data Output

Messages

Notifications

CREATE PROCEDURE

Query returned successfully in 42 msec.

Total rows:

Query complete 00:00:00.042

LF Ln 18, Col 4

✓ Query returned successfully in 42 msec. ✕