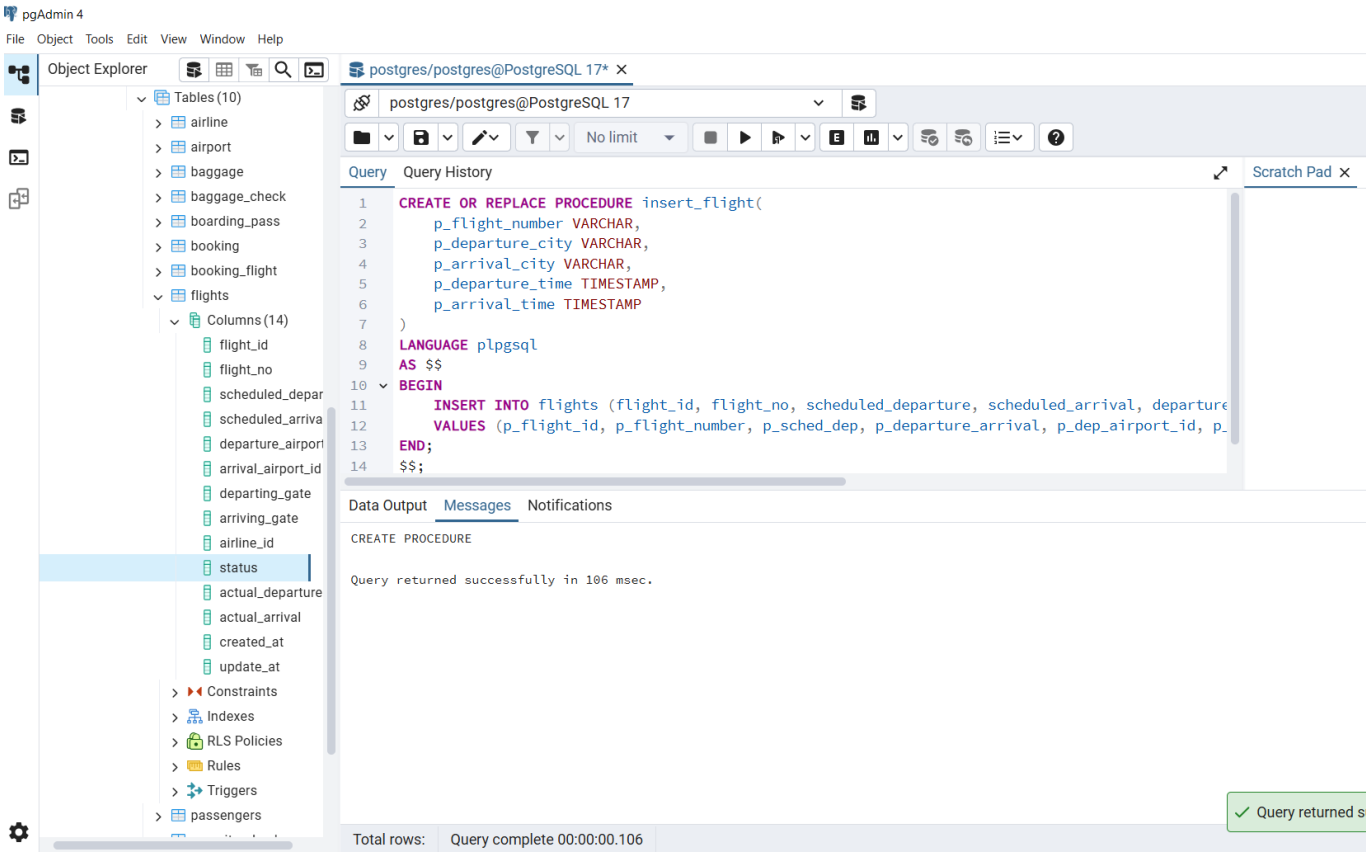
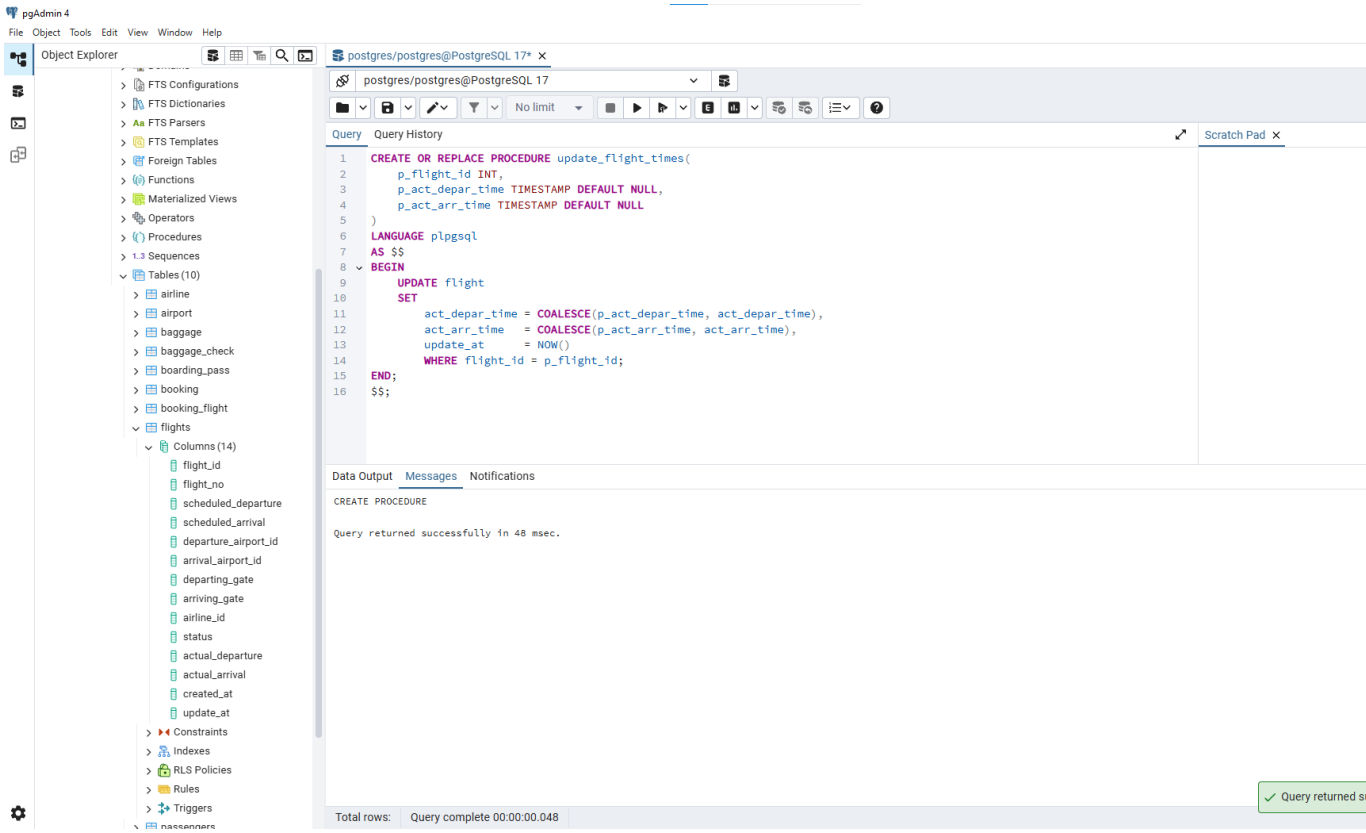
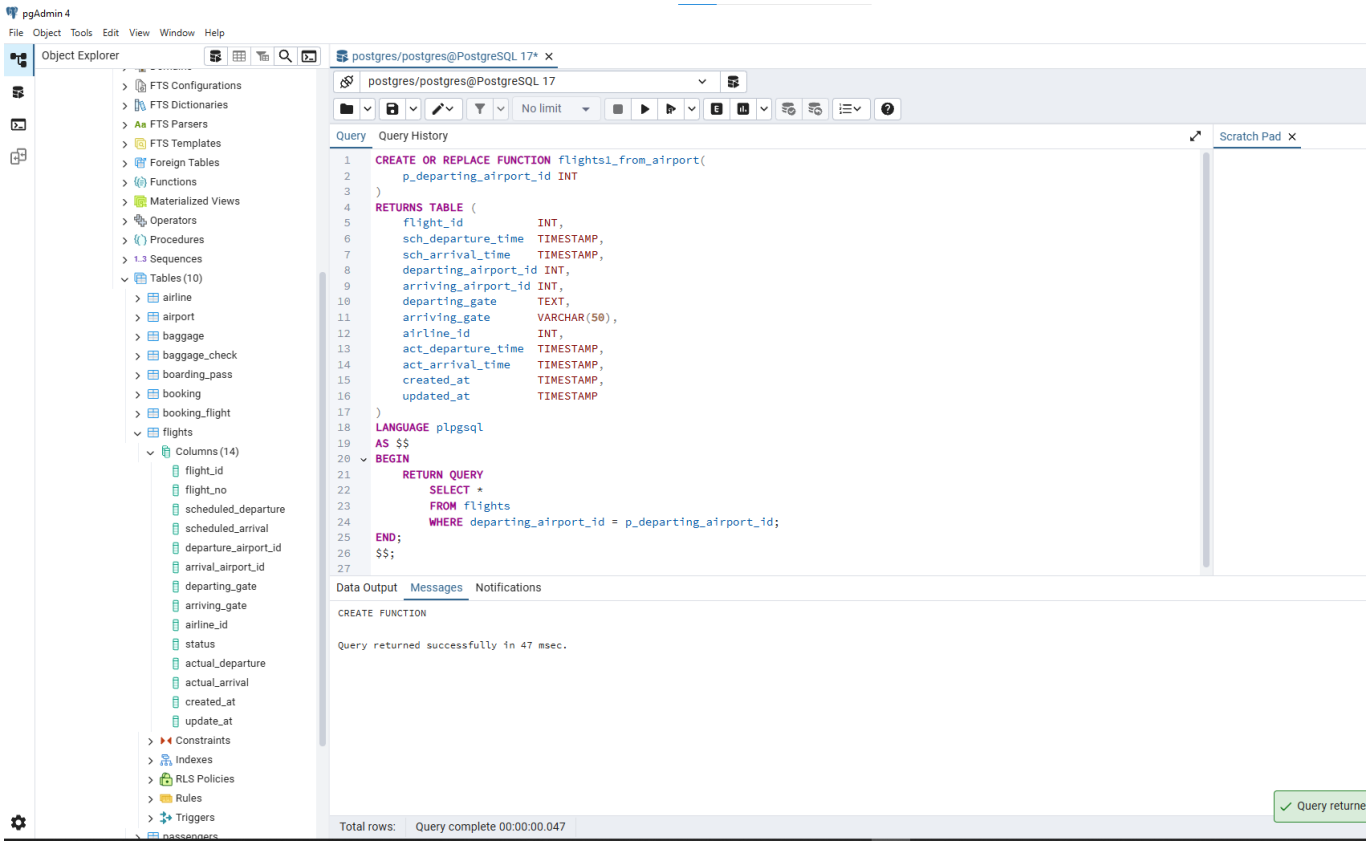
1)


pgAdmin 4
File Object Tools Edit View Window Help
Object Explorer
postgres/postgres@PostgreSQL 17*
postgres/postgres@PostgreSQL 17
No limit
Query Query History
Scratch Pad
CREATE OR REPLACE PROCEDURE insert_flight(
p_flight_number VARCHAR,
p_departure_city VARCHAR,
p_arrival_city VARCHAR,
p_departure_time TIMESTAMP,
p_arrival_time TIMESTAMP
)
LANGUAGE plpgsql
AS $$
BEGIN
INSERT INTO flights (flight_id, flight_no, scheduled_departure, scheduled_arrival, departure
VALUES (p_flight_id, p_flight_number, p_sched_dep, p_departure_arrival, p_dep_airport_id, p_
END;
$$;
Data Output Messages Notifications
CREATE PROCEDURE
Query returned successfully in 106 msec.
Total rows:
Query complete 00:00:00.106
Tables (10)
airline
airport
baggage
baggage_check
boarding_pass
booking
booking_flight
flights
Columns (14)
flight_id
flight_no
scheduled_depar
scheduled_arriva
departure_airport
arrival_airport_id
departing_gate
arriving_gate
airline_id
status
actual_departure
actual_arrival
created_at
update_at
Constraints
Indexes
RLS Policies
Rules
Triggers
passengers

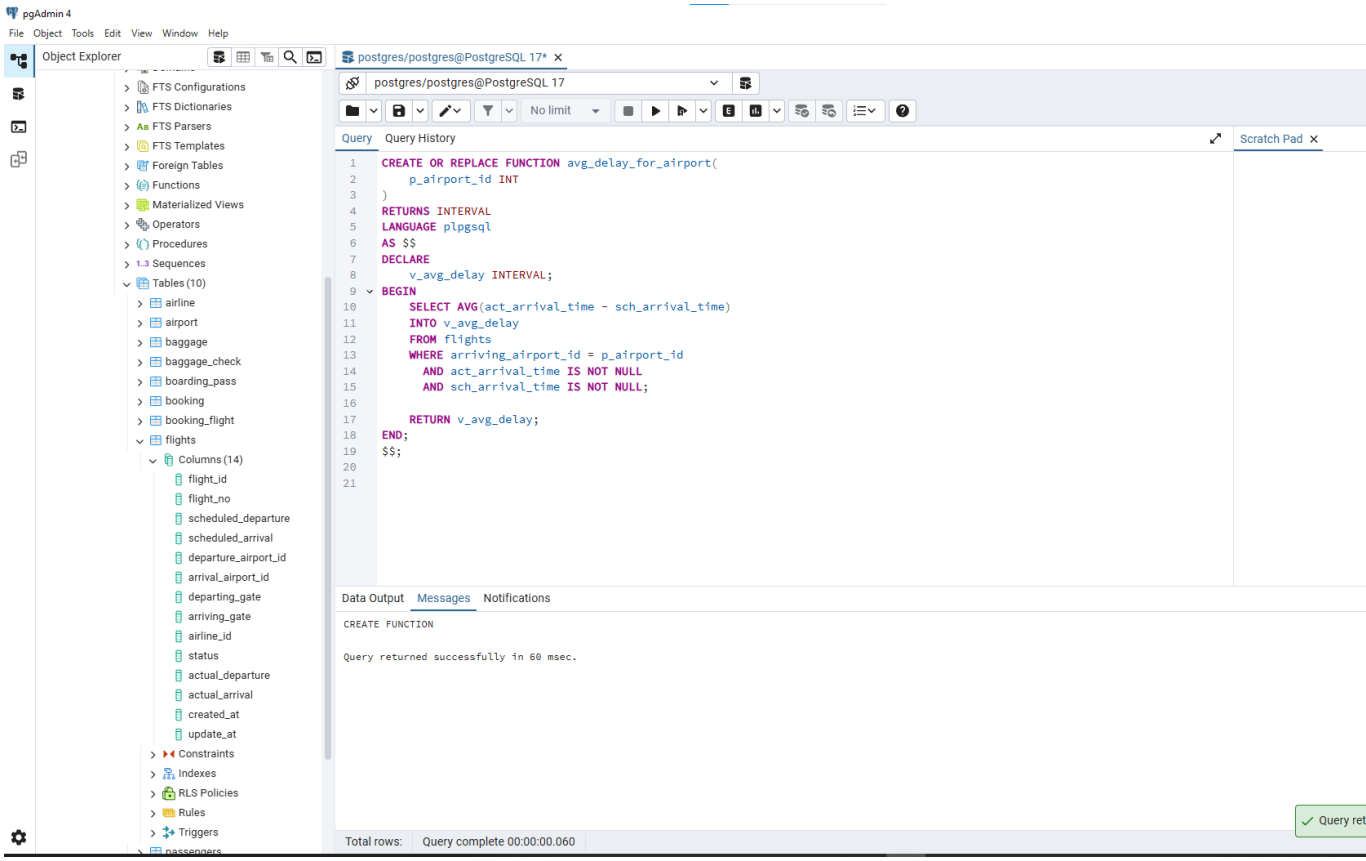
2)


pgAdmin 4
File Object Tools Edit View Window Help
Object Explorer
postgres/postgres@PostgreSQL 17*
postgres/postgres@PostgreSQL 17
No limit
Query Query History
Scratch Pad
CREATE OR REPLACE PROCEDURE update_flight_times(
p_flight_id INT,
p_act_depar_time TIMESTAMP DEFAULT NULL,
p_act_arr_time TIMESTAMP DEFAULT NULL
)
LANGUAGE plpgsql
AS $$
BEGIN
UPDATE flight
SET
act_depar_time = COALESCE(p_act_depar_time, act_depar_time),
act_arr_time = COALESCE(p_act_arr_time, act_arr_time),
update_at = NOW()
WHERE flight_id = p_flight_id;
END;
$$;
Data Output Messages Notifications
CREATE PROCEDURE
Query returned successfully in 48 msec.
Total rows:
Query complete 00:00:00.048
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
Columns (14)
flight_id
flight_no
scheduled_departure
scheduled_arrival
departure_airport_id
arrival_airport_id
departing_gate
arriving_gate
airline_id
status
actual_departure
actual_arrival
created_at
update_at
Constraints
Indexes
RLS Policies
Rules
Triggers

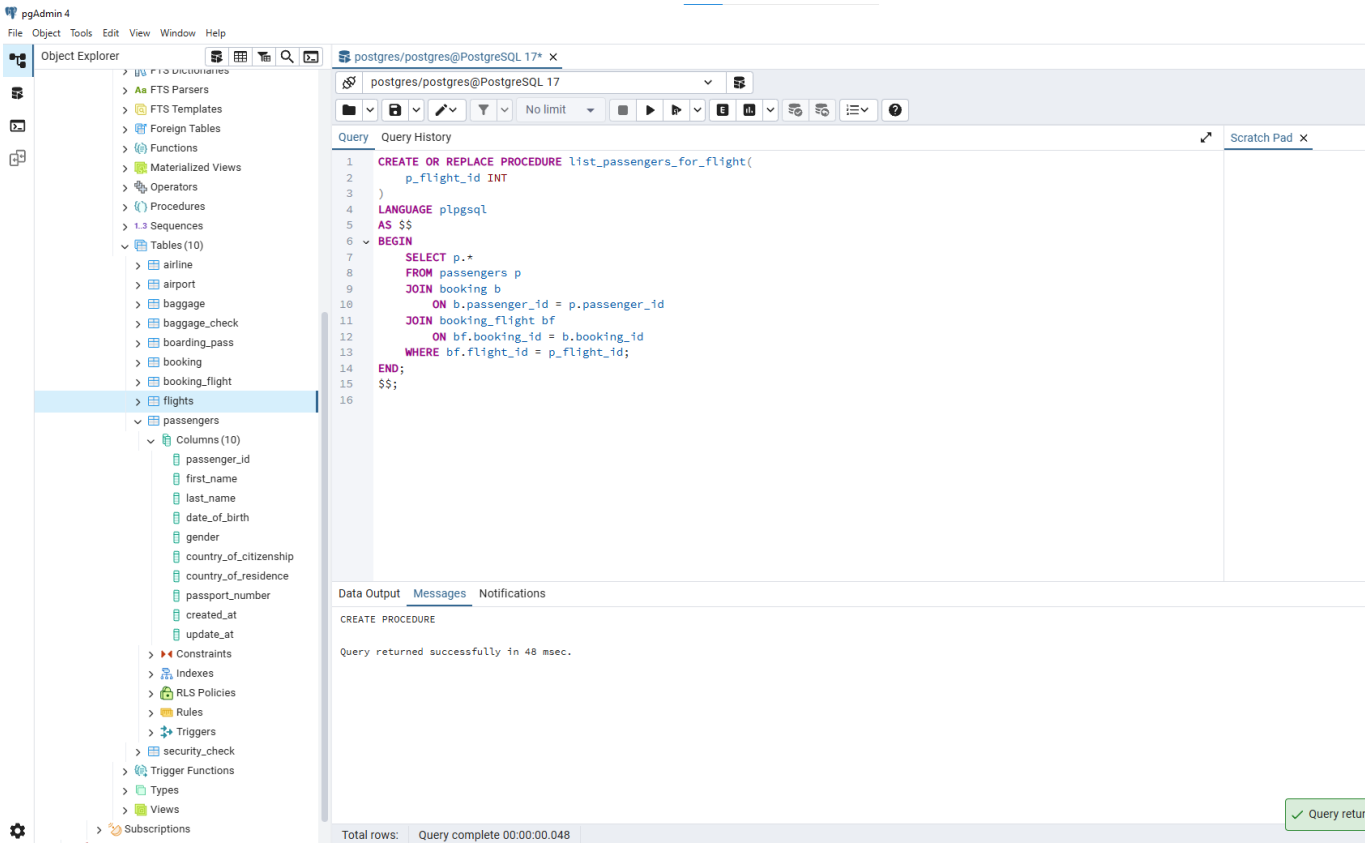
3)


pgAdmin 4
File Object Tools Edit View Window Help
Object Explorer
postgres/postgres@PostgreSQL 17*
postgres/postgres@PostgreSQL 17
No limit
Query Query History
Scratch Pad
CREATE OR REPLACE FUNCTION flights1_from_airport(
p_departing_airport_id INT
)
RETURNS TABLE (
flight_id INT,
sch_departure_time TIMESTAMP,
sch_arrival_time TIMESTAMP,
departing_airport_id INT,
arriving_airport_id INT,
departing_gate TEXT,
arriving_gate VARCHAR(50),
airline_id INT,
act_departure_time TIMESTAMP,
act_arrival_time TIMESTAMP,
created_at TIMESTAMP,
updated_at TIMESTAMP
)
LANGUAGE plpgsql
AS $$
BEGIN
RETURN QUERY
SELECT *
FROM flights
WHERE departing_airport_id = p_departing_airport_id;
END;
$$;
Data Output Messages Notifications
CREATE FUNCTION
Query returned successfully in 47 msec.
Total rows:
Query complete 00:00:00.047
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
Columns (14)
flight_id
flight_no
scheduled_departure
scheduled_arrival
departure_airport_id
arrival_airport_id
departing_gate
arriving_gate
airline_id
status
actual_departure
actual_arrival
created_at
update_at
Constraints
Indexes
RLS Policies
Rules
Triggers

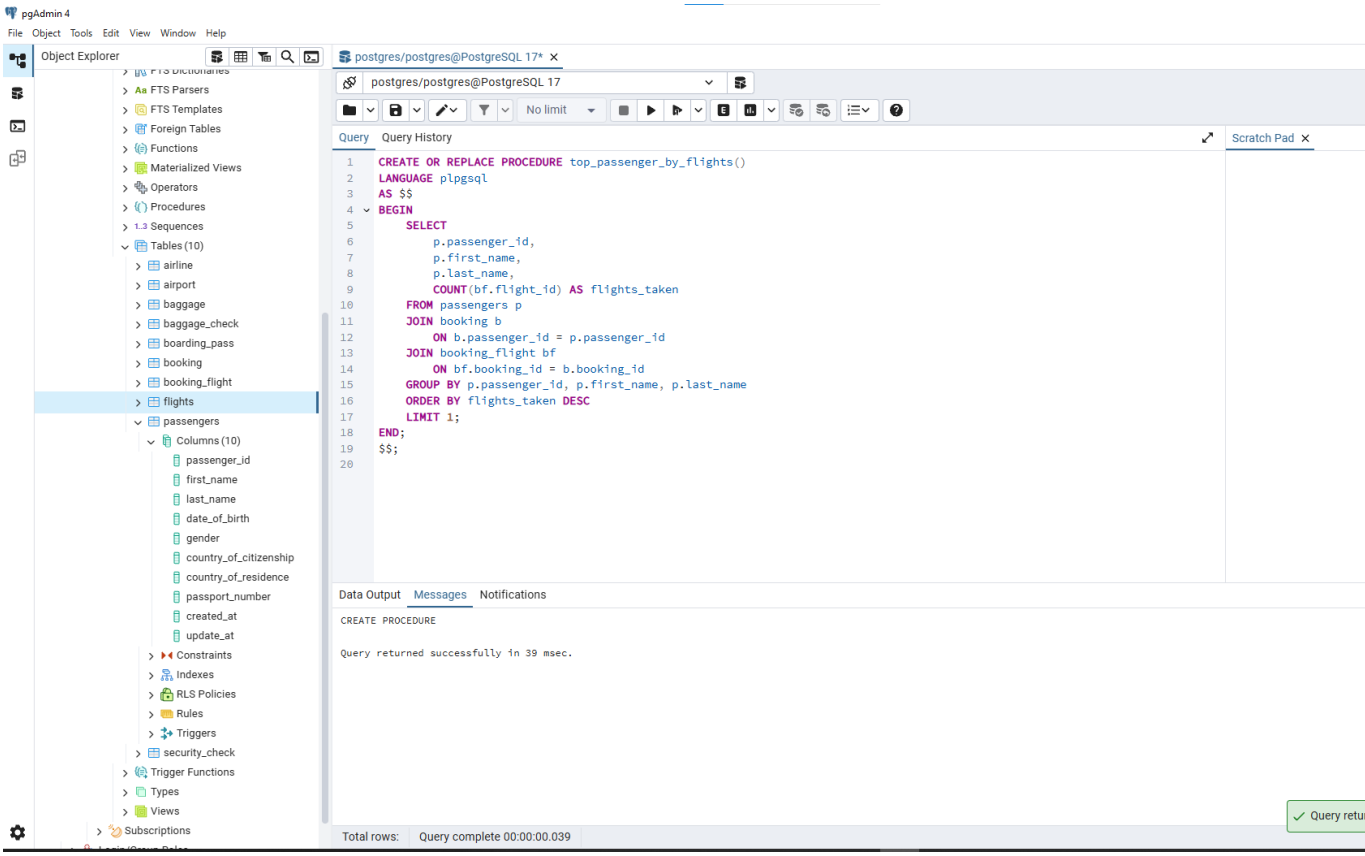
4)


pgAdmin 4
File Object Tools Edit View Window Help
Object Explorer
postgres/postgres@PostgreSQL 17*
postgres/postgres@PostgreSQL 17
No limit
Query Query History
Scratch Pad
CREATE OR REPLACE FUNCTION avg_delay_for_airport(
p_airport_id INT
)
RETURNS INTERVAL
LANGUAGE plpgsql
AS $$
DECLARE
v_avg_delay INTERVAL;
BEGIN
SELECT AVG(act_arrival_time - sch_arrival_time)
INTO v_avg_delay
FROM flights
WHERE arriving_airport_id = p_airport_id
AND act_arrival_time IS NOT NULL
AND sch_arrival_time IS NOT NULL;
RETURN v_avg_delay;
END;
$$;
Data Output Messages Notifications
CREATE FUNCTION
Query returned successfully in 60 msec.
Total rows:
Query complete 00:00:00.060
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
Columns (14)
flight_id
flight_no
scheduled_departure
scheduled_arrival
departure_airport_id
arrival_airport_id
departing_gate
arriving_gate
airline_id
status
actual_departure
actual_arrival
created_at
update_at
Constraints
Indexes
RLS Policies
Rules
Triggers

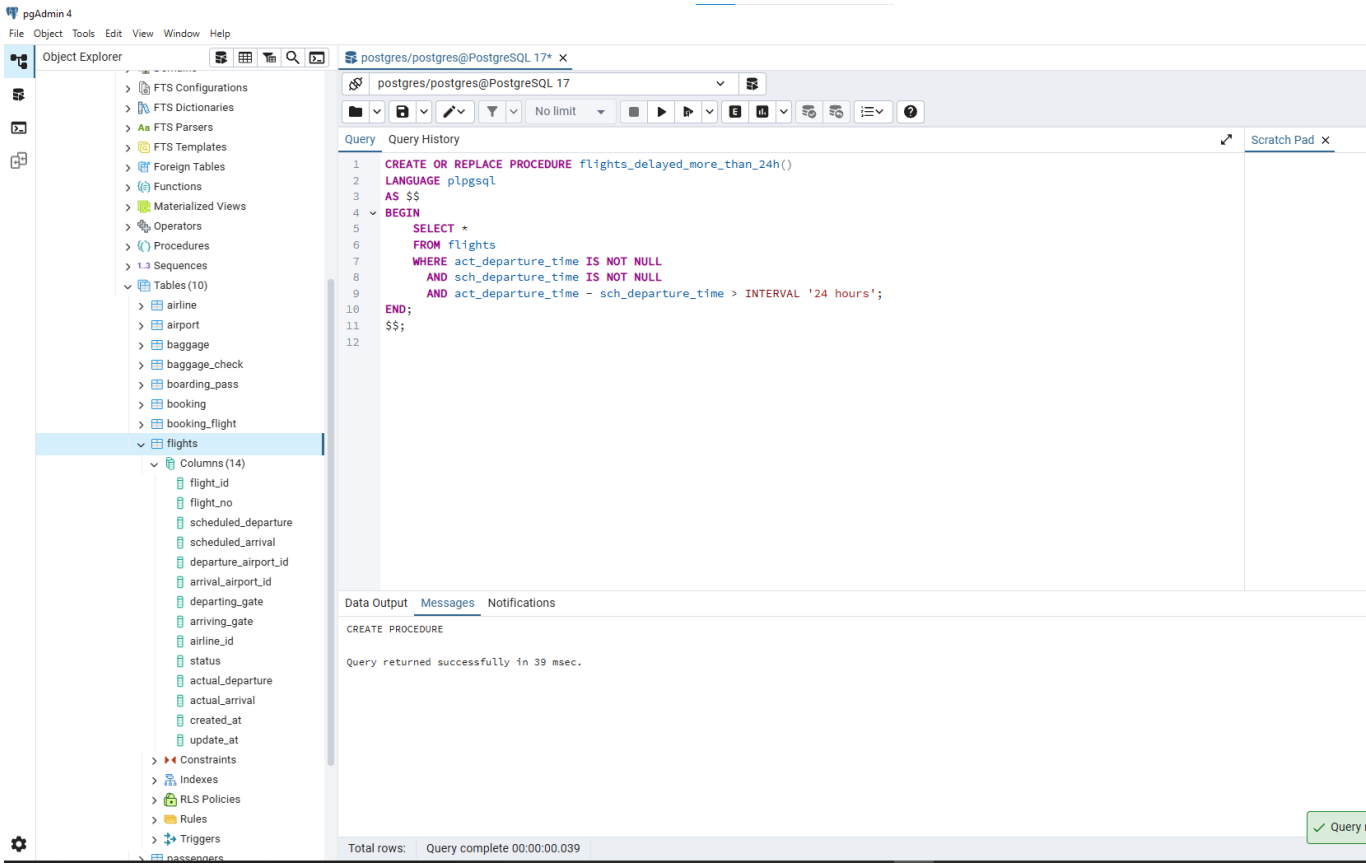
5)


pgAdmin 4
File Object Tools Edit View Window Help
Object Explorer
postgres/postgres@PostgreSQL 17*
postgres/postgres@PostgreSQL 17
No limit
Query Query History
Scratch Pad
CREATE OR REPLACE PROCEDURE list_passengers_for_flight(
p_flight_id INT
)
LANGUAGE plpgsql
AS $$
BEGIN
SELECT p.*
FROM passengers p
JOIN booking b
ON b.passenger_id = p.passenger_id
JOIN booking_flight bf
ON bf.booking_id = b.booking_id
WHERE bf.flight_id = p_flight_id;
END;
$$;
Data Output Messages Notifications
CREATE PROCEDURE
Query returned successfully in 48 msec.
Total rows:
Query complete 00:00:00.048

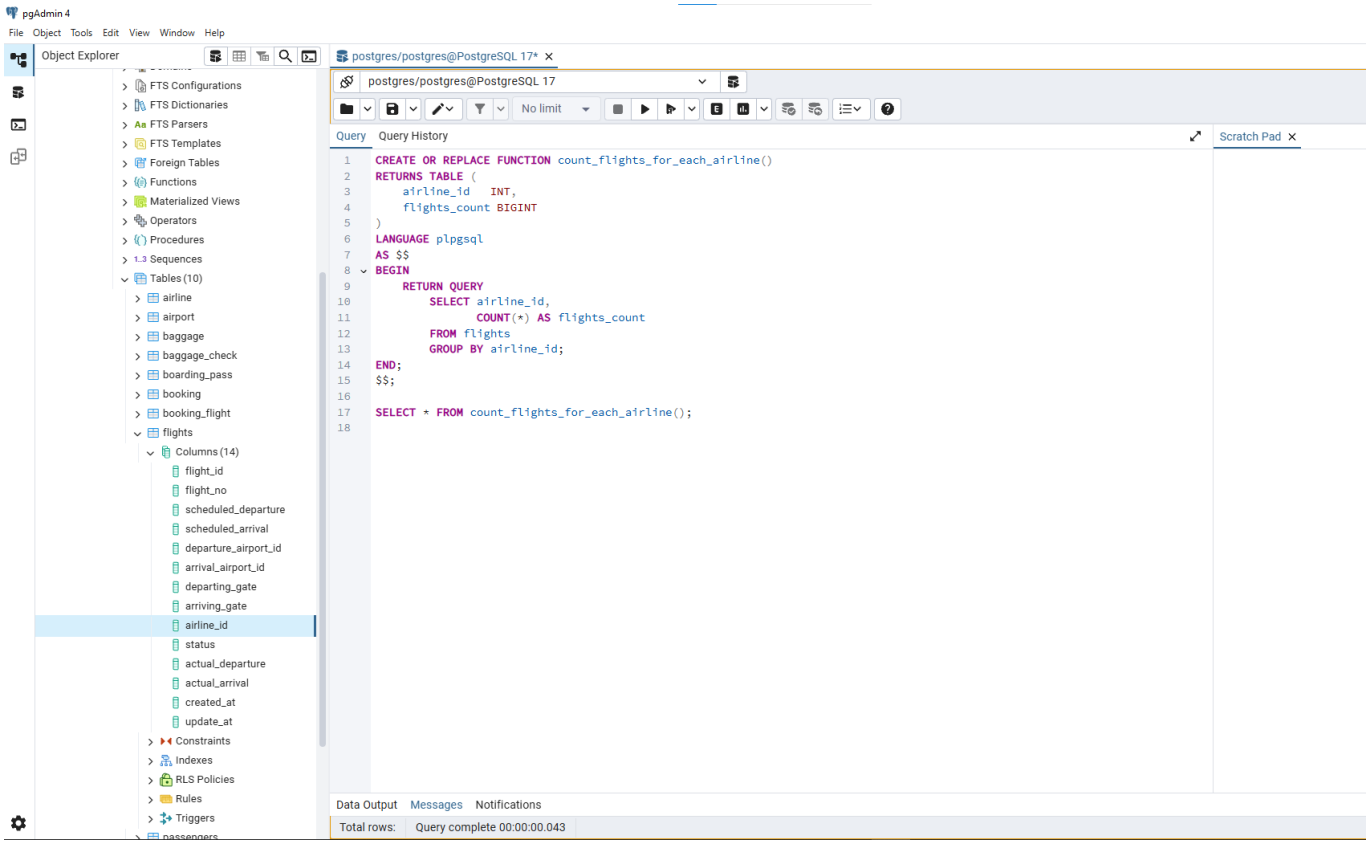
6)


pgAdmin 4
File Object Tools Edit View Window Help
Object Explorer
postgres/postgres@PostgreSQL 17*
postgres/postgres@PostgreSQL 17
No limit
Query Query History
Scratch Pad
CREATE OR REPLACE PROCEDURE top_passenger_by_flights()
LANGUAGE plpgsql
AS $$
BEGIN
SELECT
p.passenger_id,
p.first_name,
p.last_name,
COUNT(bf.flight_id) AS flights_taken
FROM passengers p
JOIN booking b
ON b.passenger_id = p.passenger_id
JOIN booking_flight bf
ON bf.booking_id = b.booking_id
GROUP BY p.passenger_id, p.first_name, p.last_name
ORDER BY flights_taken DESC
LIMIT 1;
END;
$$;
Data Output Messages Notifications
CREATE PROCEDURE
Query returned successfully in 39 msec.
Total rows:
Query complete 00:00:00.039

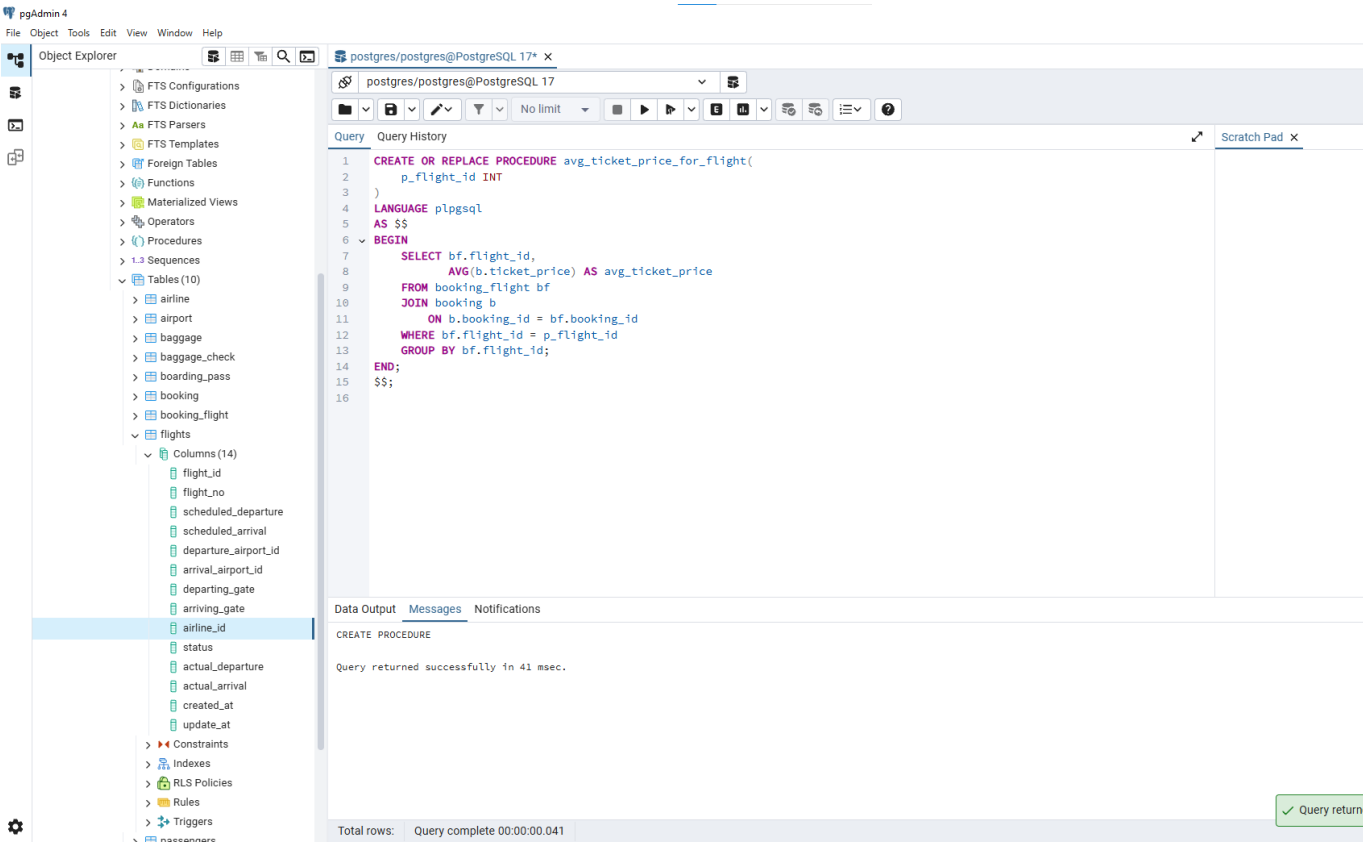
7)


pgAdmin 4
File Object Tools Edit View Window Help
Object Explorer
postgres/postgres@PostgreSQL 17*
postgres/postgres@PostgreSQL 17
No limit
Query Query History
Scratch Pad
CREATE OR REPLACE PROCEDURE flights_delayed_more_than_24h()
LANGUAGE plpgsql
AS $$
BEGIN
SELECT *
FROM flights
WHERE act_departure_time IS NOT NULL
AND sch_departure_time IS NOT NULL
AND act_departure_time - sch_departure_time > INTERVAL '24 hours';
END;
$$;
Data Output Messages Notifications
CREATE PROCEDURE
Query returned successfully in 39 msec.
Total rows:
Query complete 00:00:00.039

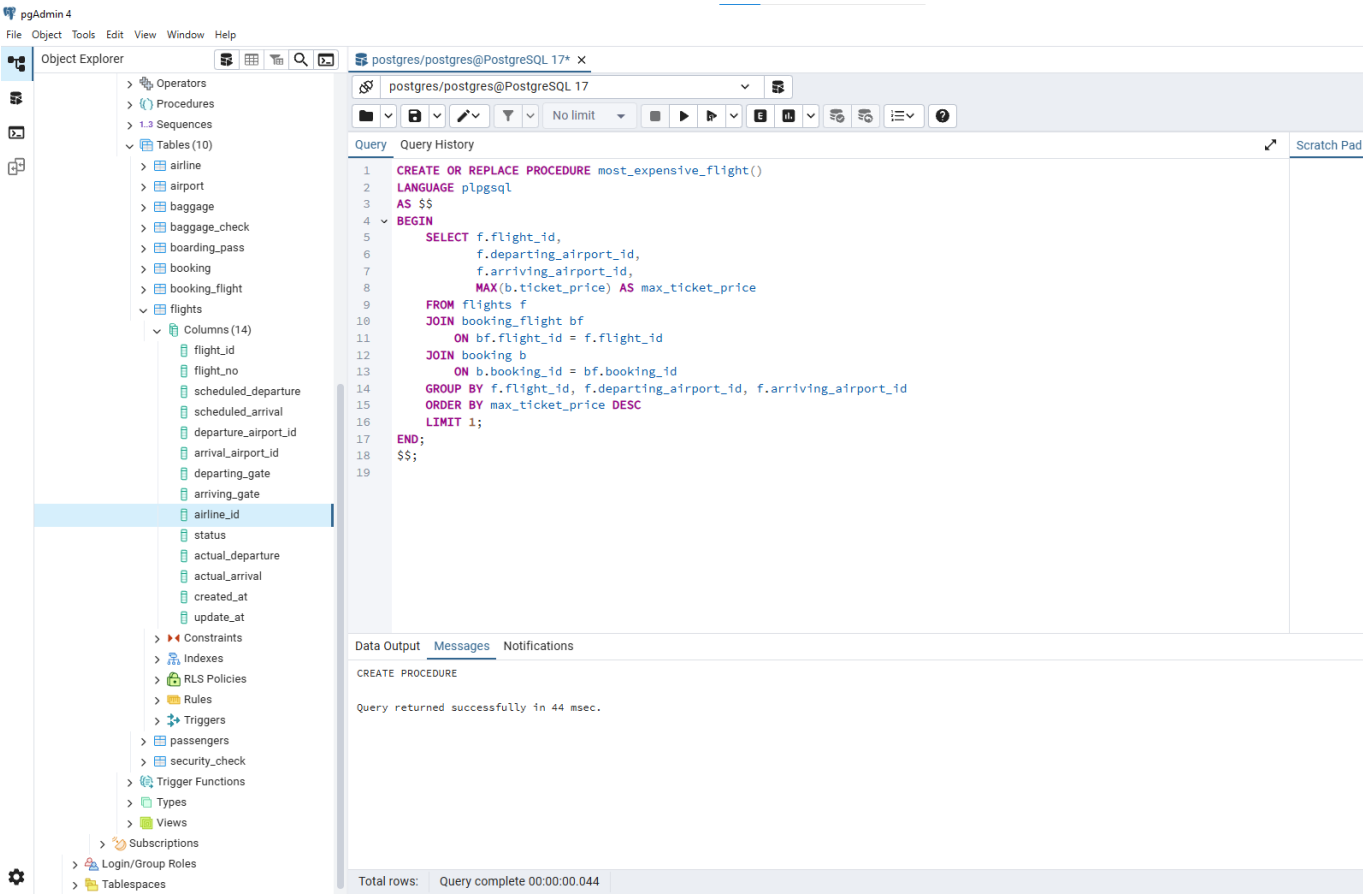
8)


pgAdmin 4
File Object Tools Edit View Window Help
Object Explorer
postgres/postgres@PostgreSQL 17*
postgres/postgres@PostgreSQL 17
No limit
Query Query History
Scratch Pad
CREATE OR REPLACE FUNCTION count_flights_for_each_airline()
RETURNS TABLE (
airline_id INT,
flights_count BIGINT
)
LANGUAGE plpgsql
AS $$
BEGIN
RETURN QUERY
SELECT airline_id,
COUNT(*) AS flights_count
FROM flights
GROUP BY airline_id;
END;
$$;
SELECT * FROM count_flights_for_each_airline();
Data Output Messages Notifications
Total rows:
Query complete 00:00:00.043

9)


pgAdmin 4
File Object Tools Edit View Window Help
Object Explorer
postgres/postgres@PostgreSQL 17*
postgres/postgres@PostgreSQL 17
No limit
Query Query History
Scratch Pad
CREATE OR REPLACE PROCEDURE avg_ticket_price_for_flight(
p_flight_id INT
)
LANGUAGE plpgsql
AS $$
BEGIN
SELECT bf.flight_id,
AVG(b.ticket_price) AS avg_ticket_price
FROM booking_flight bf
JOIN booking b
ON b.booking_id = bf.booking_id
WHERE bf.flight_id = p_flight_id
GROUP BY bf.flight_id;
END;
$$;
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
Columns (14)
flight_id
flight_no
scheduled_departure
scheduled_arrival
departure_airport_id
arrival_airport_id
departing_gate
arriving_gate
airline_id
status
actual_departure
actual_arrival
created_at
update_at
Constraints
Indexes
RLS Policies
Rules
Triggers
Data Output Messages Notifications
CREATE PROCEDURE
Query returned successfully in 41 msec.
Total rows:
Query complete 00:00:00.041

10)


pgAdmin 4
File Object Tools Edit View Window Help
Object Explorer
postgres/postgres@PostgreSQL 17*
postgres/postgres@PostgreSQL 17
No limit
Query Query History
Scratch Pad
CREATE OR REPLACE PROCEDURE most_expensive_flight()
LANGUAGE plpgsql
AS $$
BEGIN
SELECT f.flight_id,
f.departing_airport_id,
f.arriving_airport_id,
MAX(b.ticket_price) AS max_ticket_price
FROM flights f
JOIN booking_flight bf
ON bf.flight_id = f.flight_id
JOIN booking b
ON b.booking_id = bf.booking_id
GROUP BY f.flight_id, f.departing_airport_id, f.arriving_airport_id
ORDER BY max_ticket_price DESC
LIMIT 1;
END;
$$;
Operators
Procedures
Sequences
Tables (10)
airline
airport
baggage
baggage_check
boarding_pass
booking
booking_flight
flights
Columns (14)
flight_id
flight_no
scheduled_departure
scheduled_arrival
departure_airport_id
arrival_airport_id
departing_gate
arriving_gate
airline_id
status
actual_departure
actual_arrival
created_at
update_at
Constraints
Indexes
RLS Policies
Rules
Triggers
passengers
security_check
Trigger Functions
Types
Views
Subscriptions
Login/Group Roles
Tablespaces
Data Output Messages Notifications
CREATE PROCEDURE
Query returned successfully in 44 msec.
Total rows:
Query complete 00:00:00.044