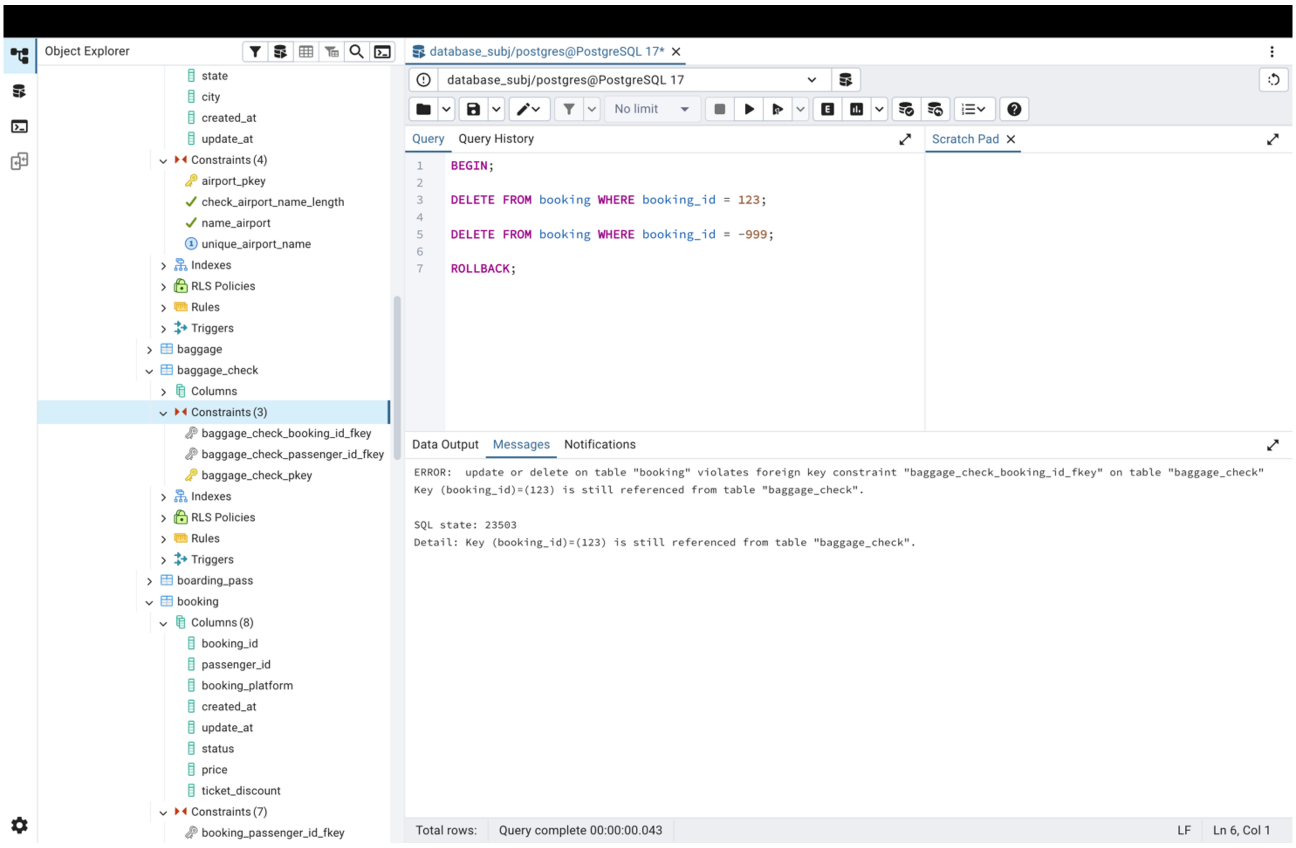
Laboratory work 9

We continue to work with the database from the previous laboratory works.

Take a full-page screenshot that covers the code and results of each task.

TRANSACTION.

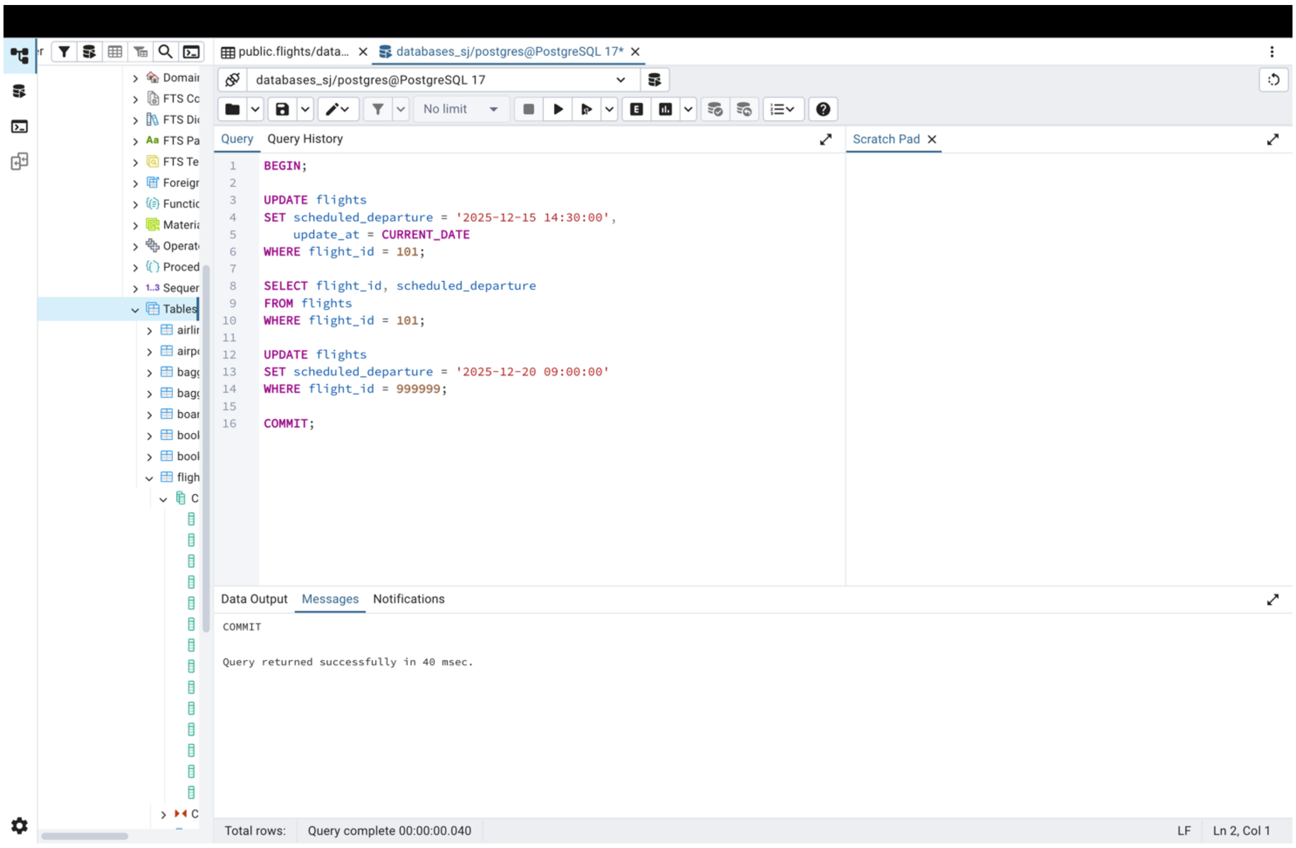
1. A passenger cancels their booking. You need to remove the booking for the flight. Ensure the ‘booking’ table no longer contains the booking. Simulate an error to test rollback (for example, invalid booking_id).



```
BEGIN;

DELETE FROM booking WHERE booking_id = 123;

DELETE FROM booking WHERE booking_id = -999;

ROLLBACK;
```

```
ERROR:  update or delete on table "booking" violates foreign key constraint "baggage_check_booking_id_fkey" on table "baggage_check"
Key (booking_id)=(123) is still referenced from table "baggage_check".

SQL state: 23503
Detail: Key (booking_id)=(123) is still referenced from table "baggage_check".
```

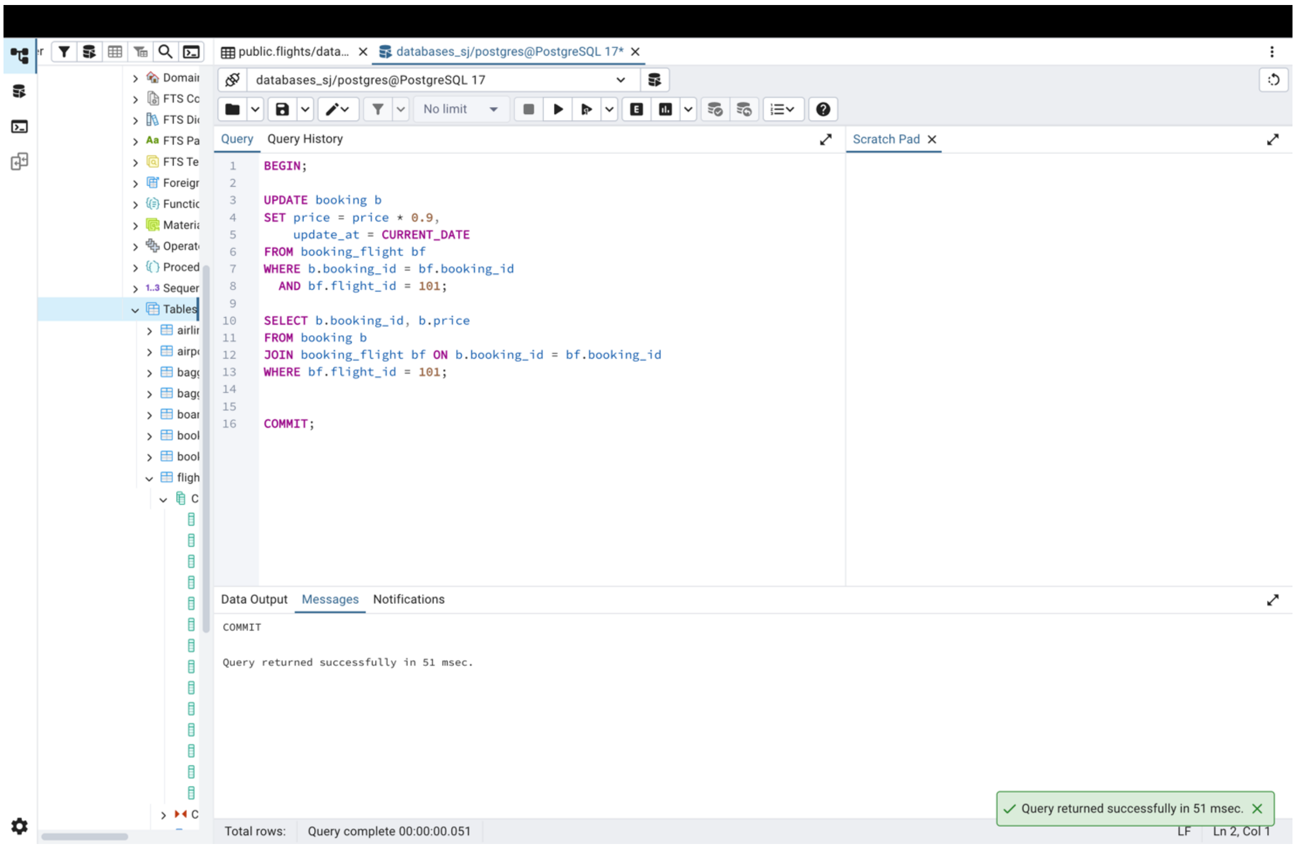
2. Rescheduling a flight. You need to reschedule a flight. Verify the ‘flights’ table reflects the new departure time. Simulate an error to test rollback (for example, invalid flight_id).



```sql
BEGIN;

UPDATE flights
SET scheduled_departure = '2025-12-15 14:30:00',
    update_at = CURRENT_DATE
WHERE flight_id = 101;

SELECT flight_id, scheduled_departure
FROM flights
WHERE flight_id = 101;

UPDATE flights
SET scheduled_departure = '2025-12-20 09:00:00'
WHERE flight_id = 999999;

COMMIT;
```

```
COMMIT

Query returned successfully in 40 msec.
```

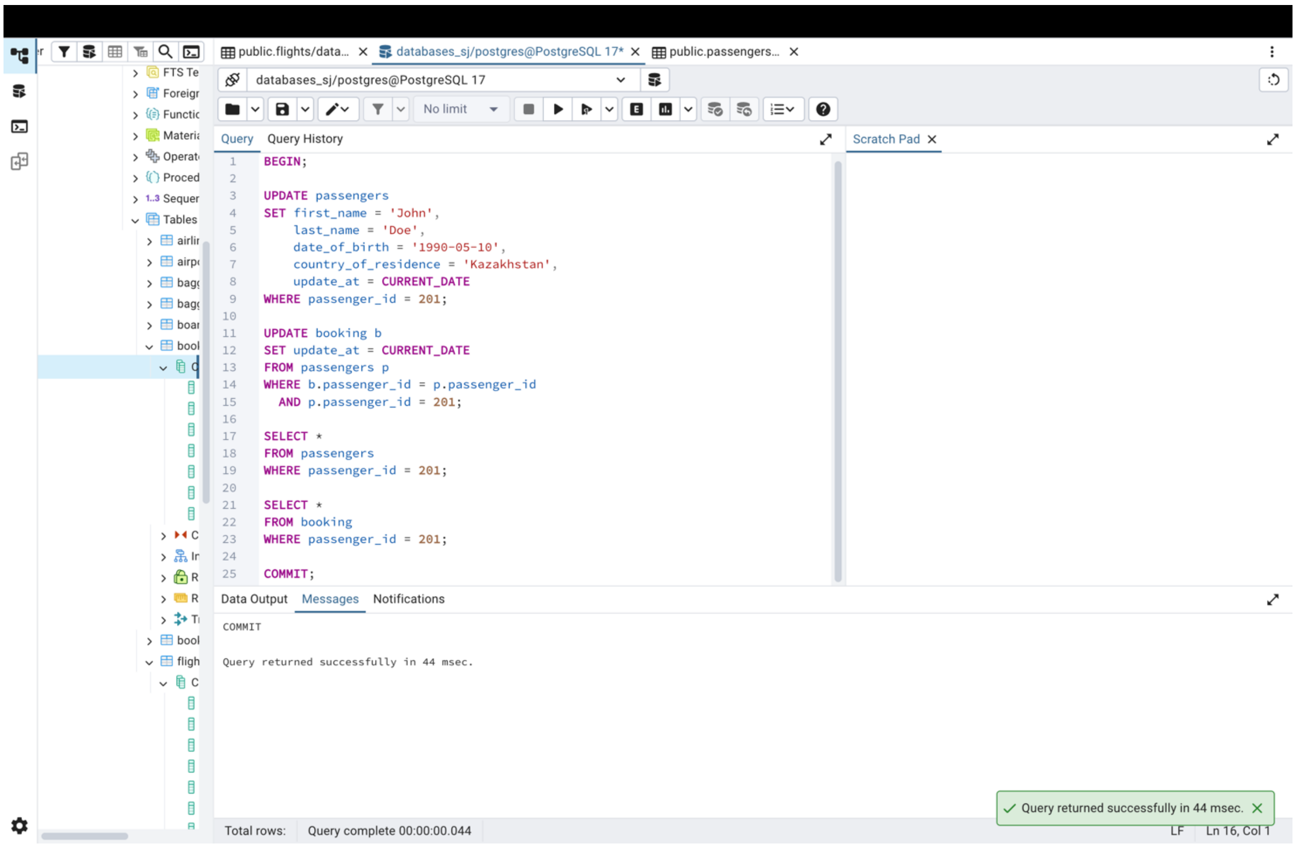
3. Updating ticket prices. You need to decrease the ticket price for a specific flight for all existing bookings. If an error occurs, no changes should be applied.


public.flights/data...
databases_sj/postgres@PostgreSQL 17*
databases_sj/postgres@PostgreSQL 17
No limit
Query
Query History
Scratch Pad
Data Output
Messages
Notifications
Query returned successfully in 51 msec.
Total rows:
Query complete 00:00:00.051
LF
Ln 2, Col 1

```sql
BEGIN;

UPDATE booking b
SET price = price * 0.9,
    update_at = CURRENT_DATE
FROM booking_flight bf
WHERE b.booking_id = bf.booking_id
  AND bf.flight_id = 101;

SELECT b.booking_id, b.price
FROM booking b
JOIN booking_flight bf ON b.booking_id = bf.booking_id
WHERE bf.flight_id = 101;


COMMIT;
```

```
COMMIT

Query returned successfully in 51 msec.
```

4. A passenger updates their details. Ensure the update is reflected across all associated records, including bookings.

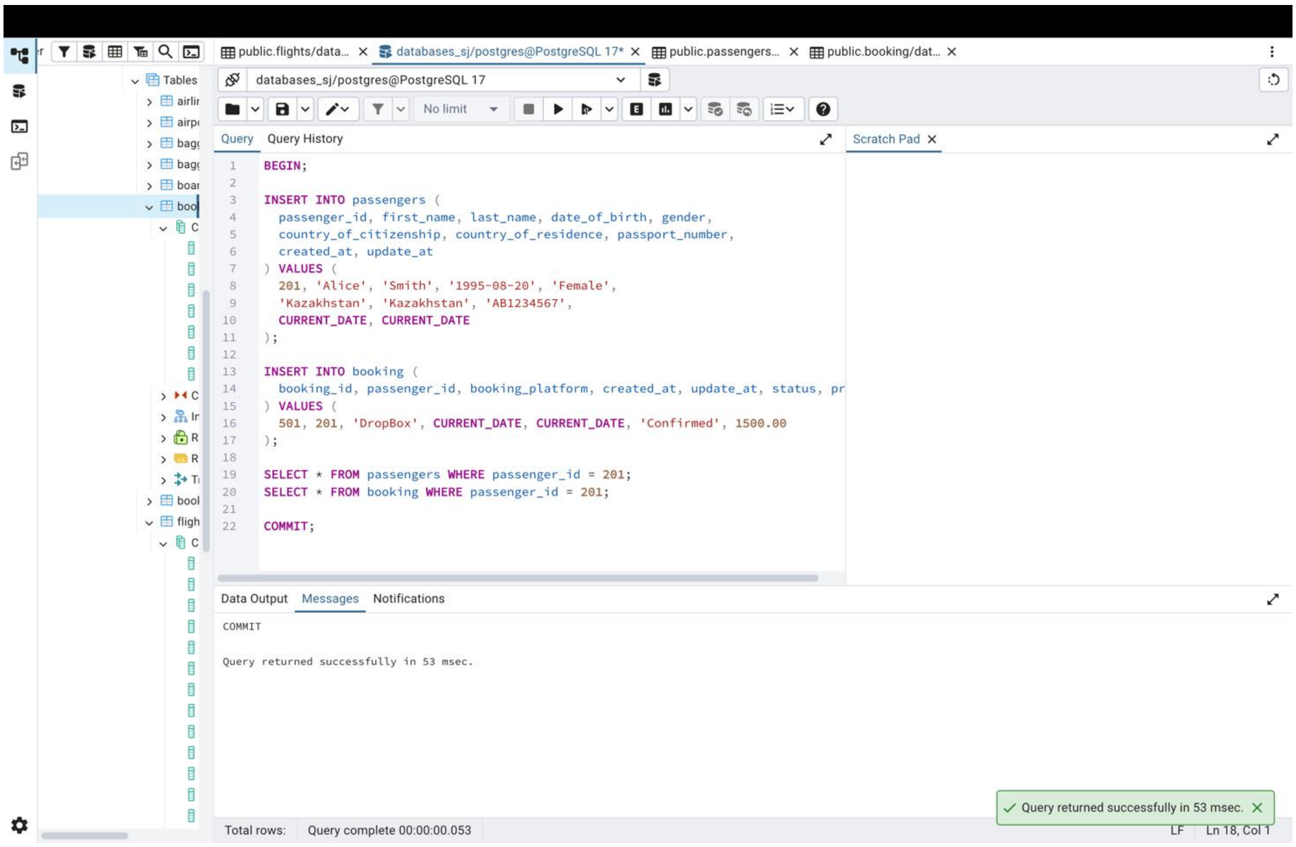


```sql
BEGIN;

UPDATE passengers
SET first_name = 'John',
    last_name = 'Doe',
    date_of_birth = '1990-05-10',
    country_of_residence = 'Kazakhstan',
    update_at = CURRENT_DATE
WHERE passenger_id = 201;

UPDATE booking b
SET update_at = CURRENT_DATE
FROM passengers p
WHERE b.passenger_id = p.passenger_id
  AND p.passenger_id = 201;

SELECT *
FROM passengers
WHERE passenger_id = 201;

SELECT *
FROM booking
WHERE passenger_id = 201;

COMMIT;
```

```
COMMIT

Query returned successfully in 44 msec.
```

```
Query returned successfully in 44 msec.
Total rows:    Query complete 00:00:00.044    LF    Ln 16, Col 1
```

5. A new passenger is registered, and a booking is created. Ensure the new passenger is added and the booking succeeds.

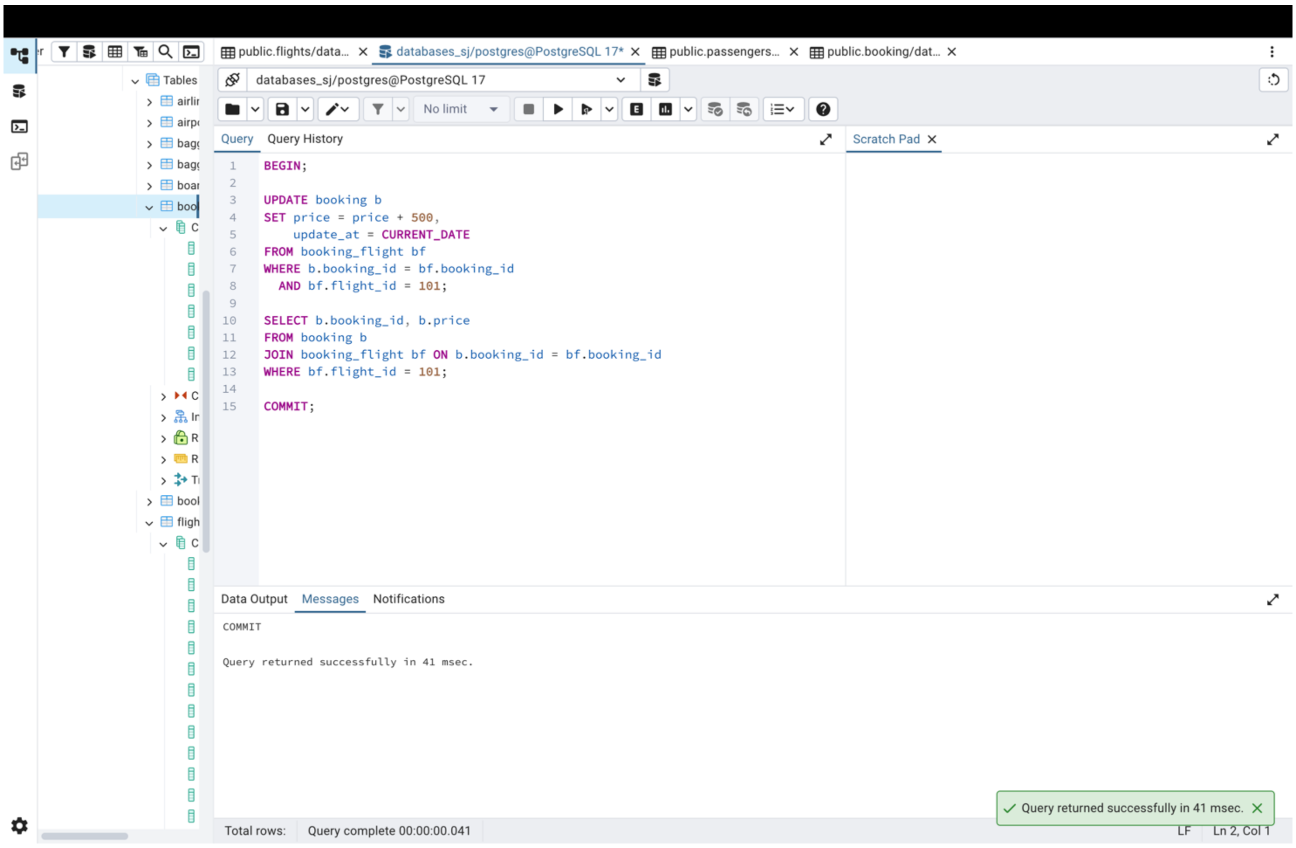


```sql
BEGIN;

INSERT INTO passengers (
  passenger_id, first_name, last_name, date_of_birth, gender,
  country_of_citizenship, country_of_residence, passport_number,
  created_at, update_at
) VALUES (
  201, 'Alice', 'Smith', '1995-08-20', 'Female',
  'Kazakhstan', 'Kazakhstan', 'AB1234567',
  CURRENT_DATE, CURRENT_DATE
);

INSERT INTO booking (
  booking_id, passenger_id, booking_platform, created_at, update_at, status, pr
) VALUES (
  501, 201, 'DropBox', CURRENT_DATE, CURRENT_DATE, 'Confirmed', 1500.00
);

SELECT * FROM passengers WHERE passenger_id = 201;
SELECT * FROM booking WHERE passenger_id = 201;

COMMIT;
```

```
COMMIT

Query returned successfully in 53 msec.
```

```
Query returned successfully in 53 msec.
Total rows:    Query complete 00:00:00.053    LF    Ln 18, Col 1
```

6. Increase the ticket price for all bookings on a specific flight by a fixed amount.



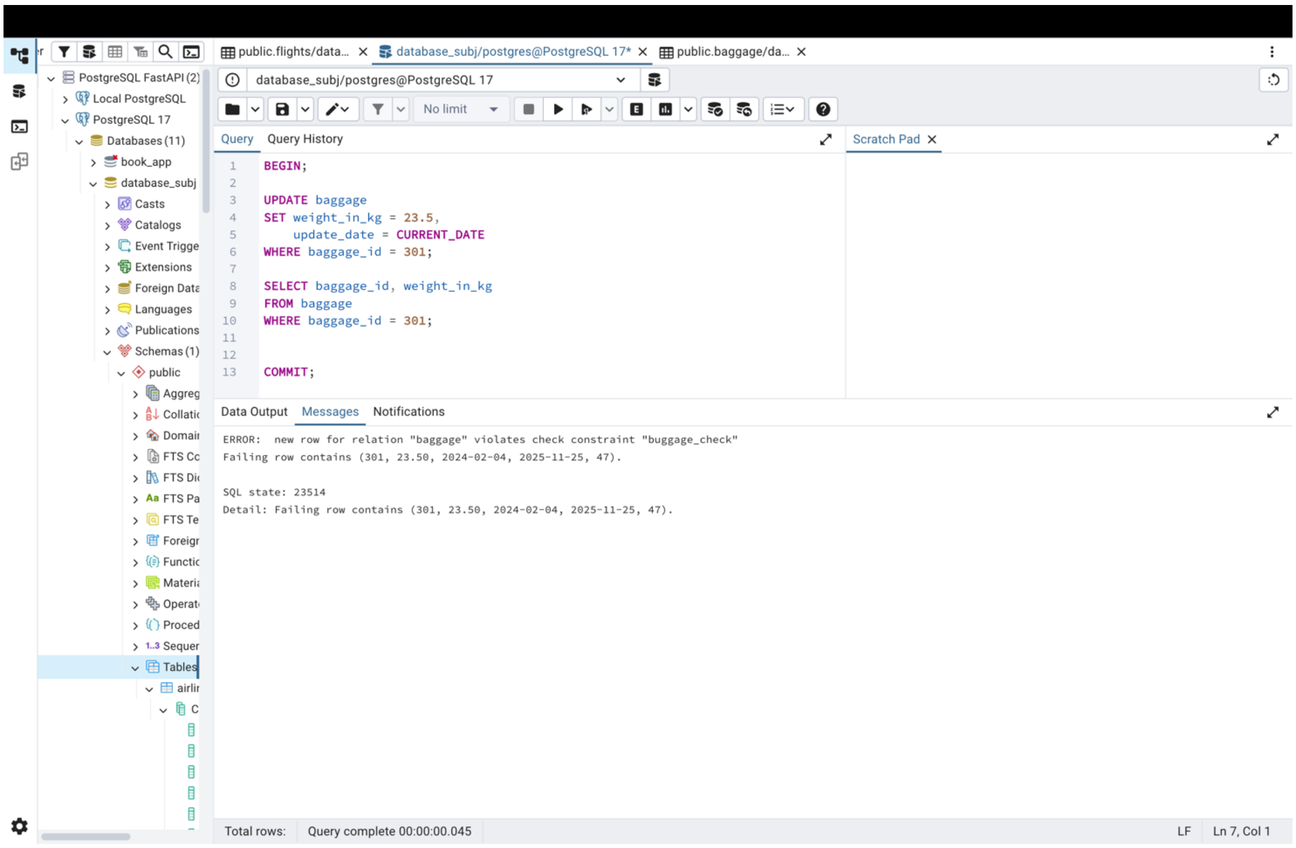
```
BEGIN;

UPDATE booking b
SET price = price + 500,
    update_at = CURRENT_DATE
FROM booking_flight bf
WHERE b.booking_id = bf.booking_id
  AND bf.flight_id = 101;

SELECT b.booking_id, b.price
FROM booking b
JOIN booking_flight bf ON b.booking_id = bf.booking_id
WHERE bf.flight_id = 101;

COMMIT;
```

```
COMMIT

Query returned successfully in 41 msec.
```

Total rows: Query complete 00:00:00.041

Query returned successfully in 41 msec.

7. Update a baggage weight. A passenger updates the declared weight of their baggage. Ensure that the change is correctly reflected in the database.

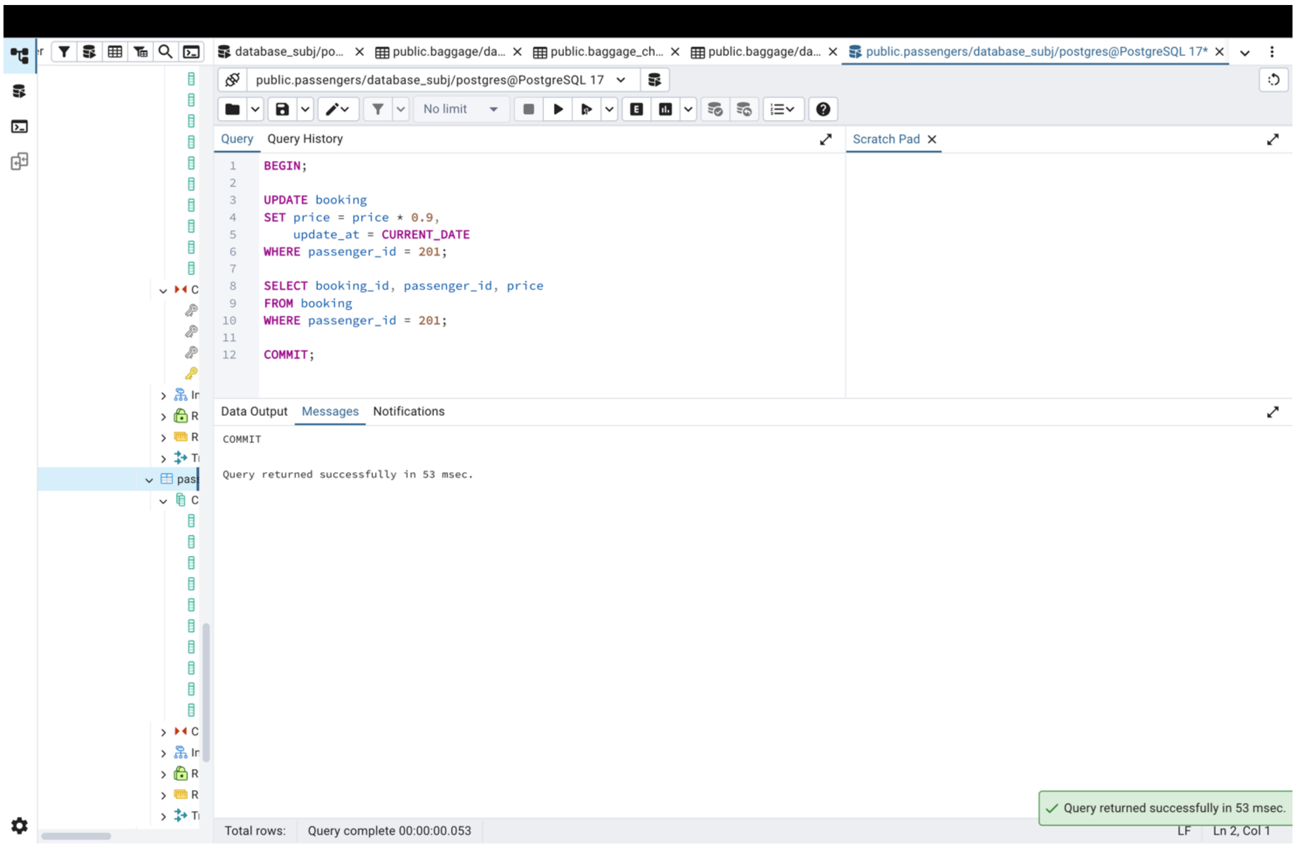


```
BEGIN;

UPDATE baggage
SET weight_in_kg = 23.5,
    update_date = CURRENT_DATE
WHERE baggage_id = 301;

SELECT baggage_id, weight_in_kg
FROM baggage
WHERE baggage_id = 301;


COMMIT;
```

```
ERROR:  new row for relation "baggage" violates check constraint "baggage_check"
Failing row contains (301, 23.50, 2024-02-04, 2025-11-25, 47).

SQL state: 23514
Detail: Failing row contains (301, 23.50, 2024-02-04, 2025-11-25, 47).
```

Total rows: Query complete 00:00:00.045

8. Apply a discount to a booking for a specific passenger. If any error occurs, roll back.

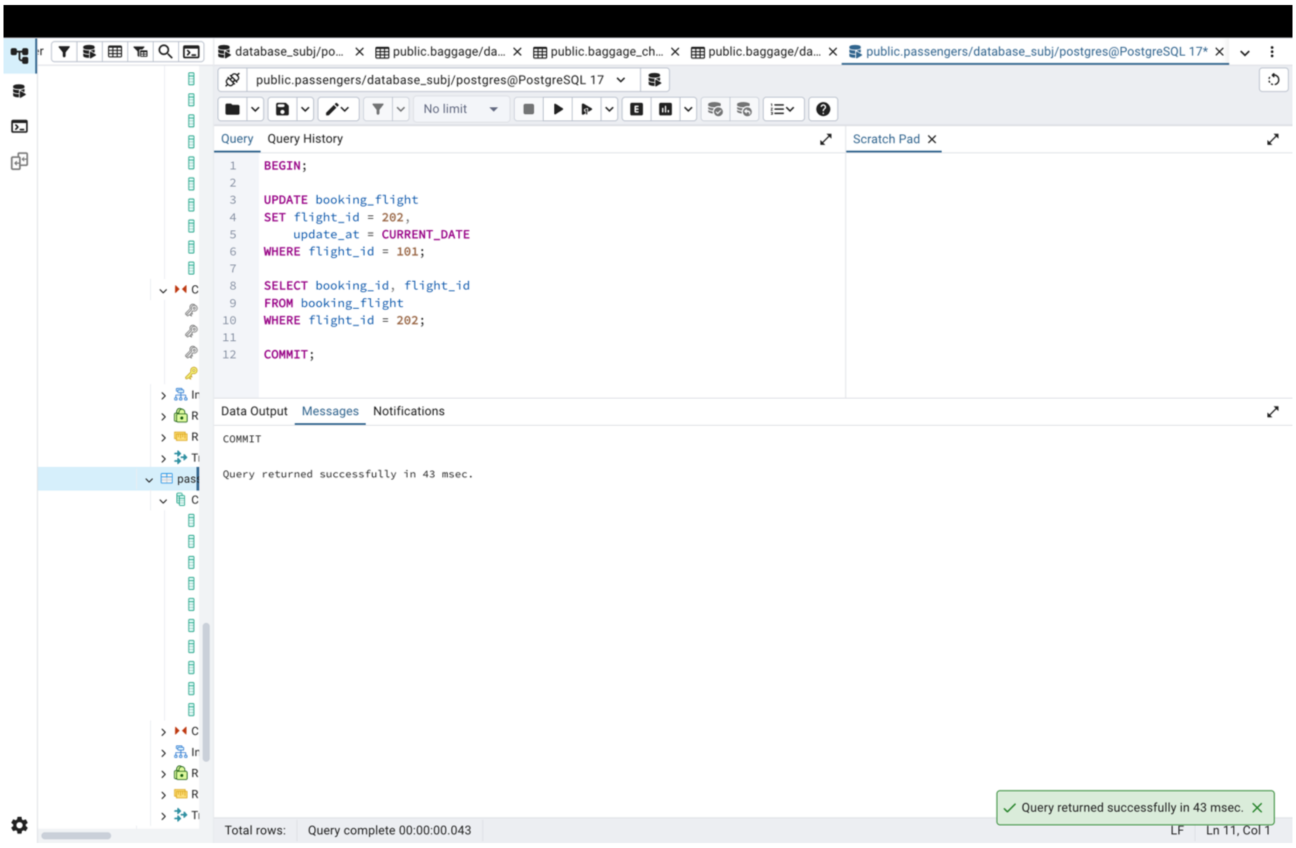


```
BEGIN;

UPDATE booking
SET price = price * 0.9,
    update_at = CURRENT_DATE
WHERE passenger_id = 201;

SELECT booking_id, passenger_id, price
FROM booking
WHERE passenger_id = 201;

COMMIT;
```

```
COMMIT

Query returned successfully in 53 msec.
```

Total rows: Query complete 00:00:00.053

9. Reschedule all bookings for a flight to a new flight.



```
BEGIN;

UPDATE booking_flight
SET flight_id = 202,
    update_at = CURRENT_DATE
WHERE flight_id = 101;

SELECT booking_id, flight_id
FROM booking_flight
WHERE flight_id = 202;

COMMIT;
```

```
COMMIT

Query returned successfully in 43 msec.
```

Total rows: Query complete 00:00:00.043