

Object Explorer

database_subj/postgres@PostgreSQL 17*

Query History

```
1 BEGIN;
2
3 DELETE FROM booking WHERE booking_id = 123;
4
5 DELETE FROM booking WHERE booking_id = -999;
6
7 ROLLBACK;
```

Data Output

Messages

Notifications

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".

Total rows: Query complete 00:00:00.043

LF Ln 6, Col 1

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer tree, which lists various database objects like state, city, created_at, update_at, and several tables and constraints under the booking and baggage_check schemas. The baggage_check schema is currently selected. In the center is a query editor window with a SQL tab containing a transaction block that attempts to delete two rows from the booking table. The first deletion succeeds, but the second fails due to a foreign key constraint violation. The Data Output tab shows the error message: '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".' The status bar at the bottom indicates the total rows affected and the completion time.

The screenshot shows the pgAdmin 4 interface with a query editor window open. The left sidebar displays a tree view of database objects under the schema 'public.flights'. The main pane contains a SQL script with numbered lines:

```
1 BEGIN;
2
3 UPDATE flights
4 SET scheduled_departure = '2025-12-15 14:30:00',
5      update_at = CURRENT_DATE
6 WHERE flight_id = 101;
7
8 SELECT flight_id, scheduled_departure
9 FROM flights
10 WHERE flight_id = 101;
11
12 UPDATE flights
13 SET scheduled_departure = '2025-12-20 09:00:00'
14 WHERE flight_id = 999999;
15
16 COMMIT;
```

Below the script, the 'Data Output' tab is selected, showing the output of the 'COMMIT' command:

```
Query returned successfully in 40 msec.
```

At the bottom, status information includes 'Total rows: 0' and 'Query complete 00:00:00.040'.

The screenshot shows the pgAdmin 4 interface. On the left is a tree view of the database schema, including 'public.flights' and 'databases_sj'. The main area contains a query editor with the following SQL code:

```
1 BEGIN;
2
3 UPDATE booking b
4   SET price = price * 0.9,
5       update_at = CURRENT_DATE
6  FROM booking_flight bf
7 WHERE b.booking_id = bf.booking_id
8   AND bf.flight_id = 101;
9
10 SELECT b.booking_id, b.price
11    FROM booking b
12   JOIN booking_flight bf ON b.booking_id = bf.booking_id
13 WHERE bf.flight_id = 101;
14
15
16 COMMIT;
```

The 'Data Output' tab shows the results of the last SELECT statement:

Query returned successfully in 51 msec.

Total rows: Query complete 00:00:00.051

A status bar at the bottom right indicates: ✓ Query returned successfully in 51 msec. LF Ln 2, Col 1

The screenshot shows the pgAdmin 4 interface with a database query editor open. The left sidebar displays a tree view of database objects, including FTS Te, Foreign, Functions, Materials, Operators, Procedures, Sequences, Tables, and Views. The Tables section is expanded, showing sub-tables like airflights, airports, bags, bookings, and passengers.

The main window contains a query editor with the following SQL script:

```
1 BEGIN;
2
3 UPDATE passengers
4 SET first_name = 'John',
5     last_name = 'Doe',
6     date_of_birth = '1990-05-10',
7     country_of_residence = 'Kazakhstan',
8     update_at = CURRENT_DATE
9 WHERE passenger_id = 201;
10
11 UPDATE booking b
12 SET update_at = CURRENT_DATE
13 FROM passengers p
14 WHERE b.passenger_id = p.passenger_id
15 AND p.passenger_id = 201;
16
17 SELECT *
18 FROM passengers
19 WHERE passenger_id = 201;
20
21 SELECT *
22 FROM booking
23 WHERE passenger_id = 201;
24
25 COMMIT;
```

The Data Output tab shows the result of the query:

```
Query returned successfully in 44 msec.
```

The status bar at the bottom indicates:

- Total rows: 0
- Query complete 00:00:00.044

A message box in the bottom right corner says:

- ✓ Query returned successfully in 44 msec. X

Other UI elements include a toolbar with various icons (e.g., Open, Save, Run, Stop) and a scratch pad tab.

The screenshot shows the pgAdmin 4 interface with a database query being run against a PostgreSQL 17 database.

Left Panel: Shows the database tree under "Tables". The "book" table is selected, indicated by a blue selection bar. Other tables like "airline", "airports", "bags", "boats", "flight", and "passenger" are also listed.

Top Bar: Shows the connection name "databases_sj/postgres@PostgreSQL 17*" and several tabs open in the background.

Toolbar: Standard pgAdmin toolbar with icons for file operations, search, and database navigation.

Query Editor:

```
1 BEGIN;
2
3 INSERT INTO passengers (
4     passenger_id, first_name, last_name, date_of_birth, gender,
5     country_of_citizenship, country_of_residence, passport_number,
6     created_at, update_at
7 ) VALUES (
8     201, 'Alice', 'Smith', '1995-08-20', 'Female',
9     'Kazakhstan', 'Kazakhstan', 'AB1234567',
10    CURRENT_DATE, CURRENT_DATE
11 );
12
13 INSERT INTO booking (
14     booking_id, passenger_id, booking_platform, created_at, update_at, status, pr
15 ) VALUES (
16     501, 201, 'DropBox', CURRENT_DATE, CURRENT_DATE, 'Confirmed', 1500.00
17 );
18
19 SELECT * FROM passengers WHERE passenger_id = 201;
20 SELECT * FROM booking WHERE passenger_id = 201;
21
22 COMMIT;
```

Data Output:

Commit

Query returned successfully in 53 msec.

Status Bar:

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

The screenshot shows the pgAdmin 4 interface with a database query editor. The left sidebar displays a tree view of tables and schemas. The main area contains a query editor with the following SQL script:

```
1 BEGIN;
2
3 UPDATE booking b
4 SET price = price + 500,
5      update_at = CURRENT_DATE
6 FROM booking_flight bf
7 WHERE b.booking_id = bf.booking_id
8     AND bf.flight_id = 101;
9
10 SELECT b.booking_id, b.price
11 FROM booking b
12 JOIN booking_flight bf ON b.booking_id = bf.booking_id
13 WHERE bf.flight_id = 101;
14
15 COMMIT;
```

The query editor has tabs for "Data Output", "Messages", and "Notifications". The "Messages" tab shows the output:

COMMIT

Query returned successfully in 41 msec.

At the bottom, status bars show "Total rows: 0" and "Query complete 00:00:00.041". A green message bar at the bottom right indicates "✓ Query returned successfully in 41 msec. LF Ln 2, Col 1".

The screenshot shows the pgAdmin 4 interface. On the left is the Object Browser tree, which includes a connection to 'PostgreSQL FastAPI(2)' containing 'Local PostgreSQL' and 'PostgreSQL 17' databases. The 'PostgreSQL 17' database node has expanded to show 'Databases(11)', 'Schemas(1)', and 'Tables'. Under 'Tables', there is a single entry for 'airline'. The main pane contains a query editor window titled 'database_subj/postgres@PostgreSQL 17'. The query itself is:

```
1 BEGIN;
2
3 UPDATE baggage
4 SET weight_in_kg = 23.5,
5     update_date = CURRENT_DATE
6 WHERE baggage_id = 301;
7
8 SELECT baggage_id, weight_in_kg
9 FROM baggage
10 WHERE baggage_id = 301;
11
12
13 COMMIT;
```

Below the query editor, the 'Data Output' tab is selected, showing the following error message:

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).

At the bottom of the pgAdmin interface, the status bar displays 'Total rows: 0' and 'Query complete 00:00:00.045'. The bottom right corner shows 'LF Ln 7, Col 1'.

The screenshot shows the pgAdmin 4 interface with a database query editor. The query window displays the following SQL code:

```
1 BEGIN;
2
3 UPDATE booking
4   SET price = price * 0.9,
5       update_at = CURRENT_DATE
6 WHERE passenger_id = 201;
7
8 SELECT booking_id, passenger_id, price
9 FROM booking
10 WHERE passenger_id = 201;
11
12 COMMIT;
```

The results pane shows the output of the COMMIT command:

Query returned successfully in 53 msec.

Total rows: Query complete 00:00:00.053 ✓ Query returned successfully in 53 msec. LF Ln 2, Col 1

The screenshot shows the pgAdmin 4 interface with a query editor window open. The query window title is "public.passengers/database_subj/postgres@PostgreSQL 17*". The main pane displays the following SQL code:

```
1 BEGIN;
2
3 UPDATE booking_flight
4   SET flight_id = 202,
5       update_at = CURRENT_DATE
6 WHERE flight_id = 101;
7
8 SELECT booking_id, flight_id
9 FROM booking_flight
10 WHERE flight_id = 202;
11
12 COMMIT;
```

Below the code, the "Data Output" tab is selected, showing the result of the query:

```
Commit
```

Query returned successfully in 43 msec.

In the bottom status bar, it says "Total rows: Query complete 00:00:00.043". A green success message in the bottom right corner states "✓ Query returned successfully in 43 msec. LF Ln 11, Col 1".