

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).

The screenshot displays two pgAdmin 4 sessions. Both sessions are connected to the 'airport\_db' database.

**Session 1 (Top):**

- Query:

```
begin ;
delete from booking
where booking_id = 100 ;
commit ;
```

- Messages tab output:

```
COMMIT
Query returned successfully in 100 msec.
```

**Session 2 (Bottom):**

- Query:

```
begin ;
delete from booking
where booking_id = 121212 ;
rollback ;
```

- Messages tab output:

```
ROLLBACK
Query returned successfully in 113 msec.
```

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).

The screenshot shows the pgAdmin 4 interface with the 'airport\_db' database selected. In the center pane, a query window displays the following SQL script:

```
begin;
update flights
set scheduled_departure = '2024-01-01'
where flight_id = 5 ;
commit ;
```

The 'Data Output' tab shows the results of the query: 'Query returned successfully in 104 msec.' and 'Total rows: Query complete 00:00:00.104'. A green status bar at the bottom right indicates 'Query returned successfully in 104 msec.'

The screenshot shows the pgAdmin 4 interface with the 'airport\_db' database selected. In the center pane, a query window displays the following SQL script:

```
begin;
update flights
set scheduled_departure = '2023-12-04'
where flight_id = 4565 ;
rollback ;
```

The 'Data Output' tab shows the results of the query: 'ROLLBACK' and 'Query returned successfully in 106 msec.'. A green status bar at the bottom right indicates 'Query returned success'.

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.

The screenshot shows the pgAdmin 4 interface with the 'airport\_db' database selected. In the center query editor, the following SQL code is run:

```
begin ;
update booking b
set price = price - 5000
from booking_flight bf
where b.booking_id = bf.booking_id
and bf.flight_id = 10 ;
commit ;
```

The 'Data Output' tab shows the results of the query, indicating a successful COMMIT operation:

Query returned successfully in 147 msec.

Total rows: Query complete 00:00:00.147

A green status bar at the bottom right indicates: ✓ Query returned successfully in 147 msec. CRLF Ln 9, Col 9.

The screenshot shows the pgAdmin 4 interface with the 'airport\_db' database selected. In the center query editor, the same SQL code is run:

```
begin ;
update booking b
set price = price - 5000
from booking_flight bf
where b.booking_id = bf.booking_id
and bf.flight_id = 34572 ;
rollback ;
```

The 'Data Output' tab shows the results of the query, indicating a ROLLBACK operation:

ROLLBACK

Query returned successfully in 104 msec.

Total rows: Query complete 00:00:00.104

A green status bar at the bottom right indicates: ✓ Query returned successfully in 104 msec. CRLF Ln 9, Col 11.

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

The screenshot shows the pgAdmin 4 interface. On the left is a tree view of the database schema, including objects like Schemas, Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, and Tables. The main area contains a query editor window titled 'Query' with the following SQL code:

```
begin ;
update passengers
set first_name = 'Amina',
    last_name = 'Kazakbay'
where passenger_id = 23 ;
commit ;
```

Below the query editor is a 'Data Output' tab showing the results of the query:

Commit  
Query returned successfully in 90 msec.

At the bottom of the pgAdmin window, status information includes 'Total rows: 1 Query complete 00:00:00.090' and 'CRLF Ln 8, Col 9'.

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

pgAdmin 4

File Object Tools Edit View Window Help

airport\_db/postgres@PostgreSQL 17\*

Query History

```

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
(205, 'Alma', 'Kaisar', '2003-11-10', 'female',
 'Kazakhstan', 'Kazakhstan', '987654321', '2024-01-01', '2025-01-01');

insert into booking(booking_id, passenger_id, booking_platform, price, status)
values
(501,
 (select passenger_id from passengers where passport_number = '987654321'),
 'web',
 35000,
 'confirmed');

```

Data Output Messages Notifications

Query returned successfully in 157 msec.

Total rows: Query complete 00:00:00.157 CRLF Ln 12, Col 5

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

pgAdmin 4

File Object Tools Edit View Window Help

airport\_db/postgres@PostgreSQL 17\*

Query History

```

begin;

update booking b
set price = price + 5000
from booking_flight bf
where b.booking_id = bf.booking_id
and bf.flight_id = 15 ;

commit ;

```

Data Output Messages Notifications

Query returned successfully in 114 msec.

Total rows: Query complete 00:00:00.114 CRLF Ln 9, Col 9

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

The screenshot shows the pgAdmin 4 interface. On the left is the object browser tree, which includes nodes for Operators, Procedures, Sequences, Tables (10), and Columns (8). The 'Tables' node is expanded, showing 'airline', 'airport', 'baggage', 'baggage\_check', 'boarding\_pass', 'booking', and 'columns(8)'. The 'columns(8)' node is also expanded, listing 'booking\_id', 'passenger\_id', 'booking\_platform', 'created\_at', 'update\_at', 'status', 'price', and 'ticket\_discount'. The main window contains a query editor with the following SQL code:

```
begin ;
update baggage
set weight_in_kg = 18
where baggage_id = 30 ;
commit ;
```

Below the query editor is a 'Data Output' tab showing the results of the query. It displays a single row with the value 'COMMIT'. Below that, it says 'Query returned successfully in 118 msec.' At the bottom of the pgAdmin window, there is a status bar with the message 'Query returned successfully in 118 msec.' and 'CRLF Ln 7, Col 9'.

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

pgAdmin 4

File Object Tools Edit View Window Help

airport\_db/postgres@PostgreSQL 17\*

Query History

```
begin ;
update booking
set price = price * 0.90
where booking_id = 110
and passenger_id = 44;

commit ;
```

Data Output Messages Notifications

Query returned successfully in 116 msec.

Total rows: Query complete 00:00:00.116

✓ Query returned successfully in 116 msec. ✘ CRLF Ln 8, Col 9

This screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with tables like airline, airport, baggage, baggage\_check, boarding\_pass, booking, and passengers. The 'booking' table is currently selected, showing its columns: booking\_id, passenger\_id, booking\_platform, created\_at, update\_at, status, price, and ticket\_discount. The main query editor contains a PostgreSQL script that begins with a BEGIN block, followed by an UPDATE statement targeting the 'booking' table where booking\_id is 110 and passenger\_id is 44. The update sets the price to 90% of its original value. A COMMIT block concludes the transaction. The results pane shows a single row affected, with a message indicating success: 'Query returned successfully in 116 msec.' A status bar at the bottom provides additional details: 'Total rows: Query complete 00:00:00.116' and a note about carriage returns ('CRLF') and the current line ('Ln 8, Col 9').

pgAdmin 4

File Object Tools Edit View Window Help

airport\_db/postgres@PostgreSQL 17\*

Query History

```
begin ;
update booking
set price = price * 0.90
where booking_id = 10000 ;

rollback ;
```

Data Output Messages Notifications

ROLLBACK

Query returned successfully in 104 msec.

Total rows: Query complete 00:00:00.104

✓ Query returned successfully in 104 msec. ✘ CRLF Ln 7, Col 11

This screenshot shows the pgAdmin 4 interface, similar to the one above but with a different query. The schema is identical. The 'booking' table is selected, showing its columns. The query editor contains a BEGIN block, an UPDATE statement that multiplies the price by 0.90 for a booking with ID 10000, and a ROLLBACK block. The results pane shows a message indicating a rollback: 'ROLLBACK'. A status bar at the bottom provides details: 'Query returned successfully in 104 msec.' and a note about carriage returns ('CRLF') and the current line ('Ln 7, Col 11').

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

pgAdmin 4

File Object Tools Edit View Window Help

airport\_db/postgres@PostgreSQL 17\*

airport\_db/postgres@PostgreSQL 17

Tables (10)

airline

airport

baggage

baggage\_check

boarding\_pass

booking

columns (8)

booking\_id

passenger\_id

booking\_platform

created\_at

update\_at

status

price

ticket\_discount

Constraints

Indexes

RLS Policies

Rules

Triggers

booking\_flight

flights

passengers

Columns (10)

passenger\_id

first\_name

Object

Query History

```
begin ;
update booking_flight
set flight_id = 22
where flight_id = 9;
commit ;
```

Data Output Messages Notifications

Query returned successfully in 100 msec.

Commit

Total rows: Query complete 00:00:00.100

✓ Query returned successfully in 100 msec. X

CRLF Ln 7, Col 9

The screenshot shows the pgAdmin 4 interface with a database connection to 'airport\_db/postgres@PostgreSQL 17'. In the left sidebar, the 'Tables' section is expanded, showing ten tables: airline, airport, baggage, baggage\_check, boarding\_pass, booking, columns (8), passengers, and flights. The 'columns (8)' table is currently selected. The main pane displays a SQL query window with the following code:

```
begin ;
update booking_flight
set flight_id = 22
where flight_id = 9;
commit ;
```

Below the query window, the 'Data Output' tab is active, showing the message "Query returned successfully in 100 msec." and a "Commit" message. A green status bar at the bottom right indicates "✓ Query returned successfully in 100 msec. X". The status bar also shows "CRLF" and "Ln 7, Col 9".