

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 image displays two screenshots of the pgAdmin 4 interface, showing the execution of SQL queries in the 'airport_db/postgres@PostgreSQL 17' database.

Top Screenshot: The 'Query' tab shows a script to delete a booking with ID 100. The 'Messages' tab shows a successful 'COMMIT' message.

```
1 begin ;
2
3 delete from booking
4 where booking_id = 100 ;
5
6 commit ;
```

COMMIT
Query returned successfully in 100 msec.

Total rows: Query complete 00:00:00.100 CRLF Ln 6, Col 9

Bottom Screenshot: The 'Query' tab shows a script to delete a booking with ID 121212, followed by a rollback. The 'Messages' tab shows a successful 'ROLLBACK' message.

```
1 begin ;
2
3 delete from booking
4 where booking_id = 121212 ;
5
6 rollback ;
```

ROLLBACK
Query returned successfully in 113 msec.

Total rows: Query complete 00:00:00.113 CRLF Ln 6, Col 11

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. On the left, the 'airport_db' database is selected, showing its schema structure. The main pane displays a SQL query in the 'Query' tab:

```
1 begin ;
2
3 update flights
4 set scheduled_departure = '2024-01-01'
5 where flight_id = 5 ;
6
7 commit ;
```

The 'Execute script' button is highlighted. Below the query editor, the 'Messages' tab shows the output: 'COMMIT' and 'Query returned successfully in 104 msec.' A green status bar at the bottom right confirms: 'Query returned successfully in 104 msec.'

The screenshot shows the pgAdmin 4 interface with a different SQL query in the 'Query' tab:

```
1 begin;
2
3 update flights
4 set scheduled_departure = '2023-12-04'
5 where flight_id = 4565 ;
6
7 rollback ;
```

The 'Explain Settings' button is highlighted. The 'Messages' tab shows the output: 'ROLLBACK' and 'Query returned successfully in 106 msec.' A green status bar at the bottom right confirms: 'Query returned successfully in 106 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.

The screenshot shows the pgAdmin 4 interface with the 'airport_db/postgres@PostgreSQL 17*' connection selected. The left sidebar displays the database structure, including 'airport_db' and its components like 'public' schema, 'Aggregates', 'Collations', 'Domains', 'FTS Configurations', 'FTS Dictionaries', 'FTS Parsers', 'FTS Templates', 'Foreign Tables', 'Functions', 'Materialized Views', 'Operators', 'Procedures', 'Sequences', 'Tables', 'Trigger Functions', and 'Types'. The main query editor contains the following SQL code:

```
1 begin ;
2
3 update booking b
4 set price = price - 5000
5 from booking_flight bf
6 where b.booking_id = bf.booking_id
7 and bf.flight_id = 10 ;
8
9 commit ;
```

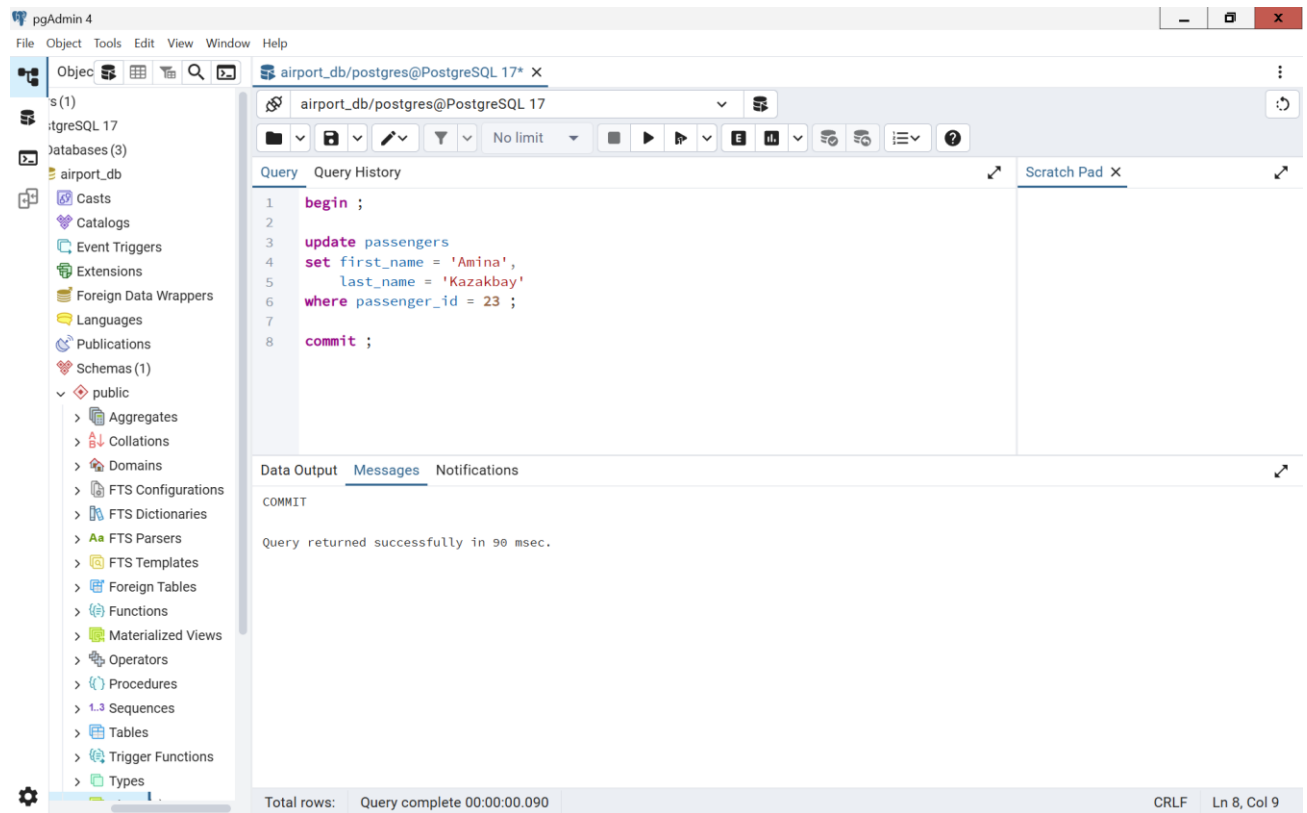
The 'Data Output' tab shows the result: 'COMMIT' and 'Query returned successfully in 147 msec.' A green notification box at the bottom right confirms: 'Query returned successfully in 147 msec.' The status bar at the bottom indicates 'Total rows: Query complete 00:00:00.147' and 'CRLF Ln 9, Col 9'.

The screenshot shows the pgAdmin 4 interface with the same connection. The main query editor contains the following SQL code:

```
1 begin ;
2
3 update booking b
4 set price = price - 5000
5 from booking_flight bf
6 where b.booking_id = bf.booking_id
7 and bf.flight_id = 34572 ;
8
9 rollback ;
```

The 'Data Output' tab shows the result: 'ROLLBACK' and 'Query returned successfully in 104 msec.' A green notification box at the bottom right confirms: 'Query returned successfully in 104 msec.' The status bar at the bottom indicates 'Total rows: Query complete 00:00:00.104' and 'CRLF Ln 9, Col 11'.

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



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*

airport_db/postgres@PostgreSQL 17

Query Query History

```

1 begin;
2
3 insert into passengers(passenger_id, first_name, last_name, date_of_birth, gender
4                          country_of_citizenship, country_of_residence,
5                          passport_number, created_at, update_at)
6 values
7 (205, 'Alma', 'Kaisar', '2003-11-10', 'female',
8  'Kazakhstan', 'Kazakhstan', '987654321', '2024-01-01', '2025-01-01');
9
10 insert into booking(booking_id, passenger_id, booking_platform, price, status)
11 values
12 (501,
13  (select passenger_id from passengers where passport_number = '987654321'),
14  'web',
15  35000,
16  'confirmed');
17

```

Data Output Messages Notifications

COMMIT

Query returned successfully in 157 msec.

✓ 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*

airport_db/postgres@PostgreSQL 17

Query Query History

```

1 begin ;
2
3 update booking b
4 set price = price + 5000
5 from booking_flight bf
6 where b.booking_id = bf.booking_id
7 and bf.flight_id = 15 ;
8
9 commit ;

```

Data Output Messages Notifications

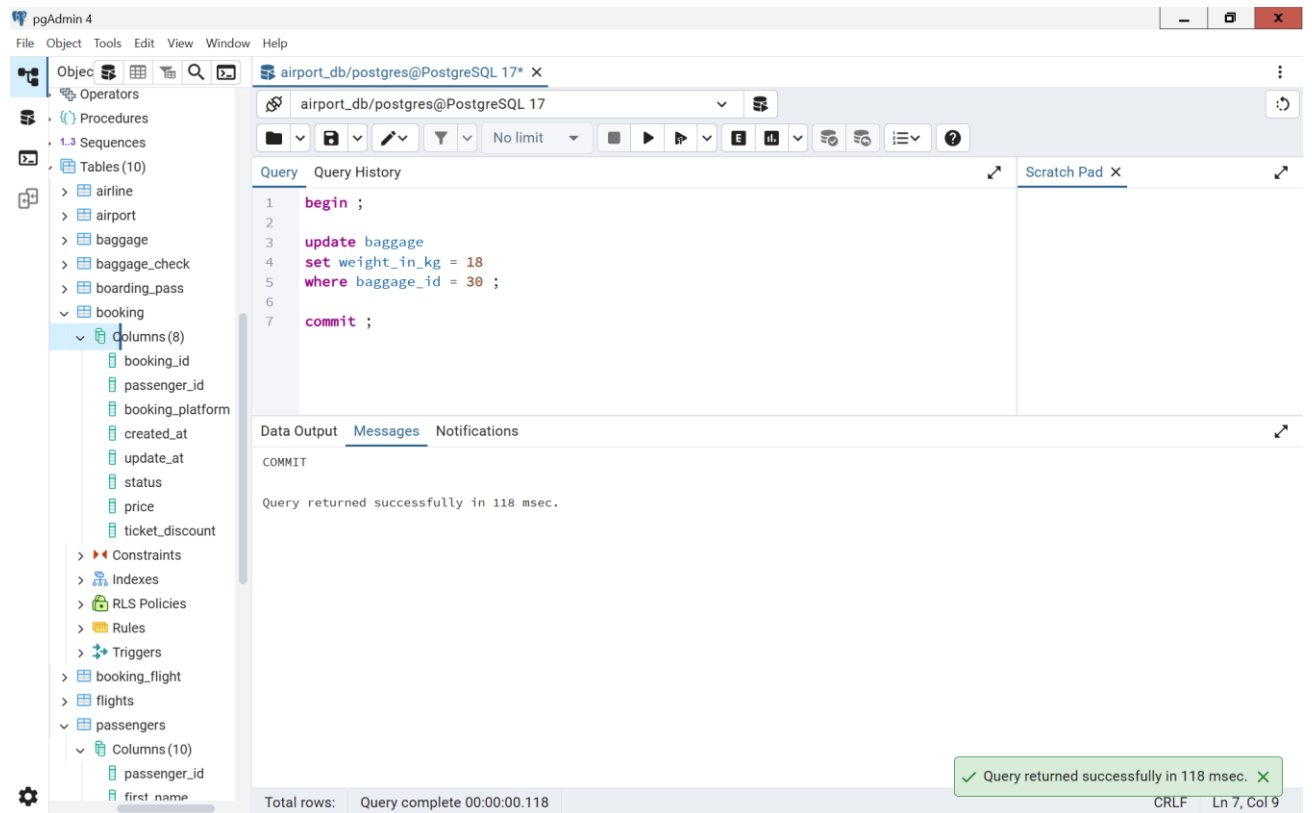
COMMIT

Query returned successfully in 114 msec.

✓ 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.



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* X

airport_db/postgres@PostgreSQL 17

Query Query History

```
1 begin ;
2
3 update booking
4 set price = price * 0.90
5 where booking_id = 110
6 and passenger_id = 44;
7
8 commit ;
```

Data Output Messages Notifications

COMMIT

Query returned successfully in 116 msec.

✓ Query returned successfully in 116 msec. ✕

Total rows: Query complete 00:00:00.116 CRLF Ln 8, Col 9

booking

- booking_id
- passenger_id
- booking_platform
- created_at
- update_at
- status
- price
- ticket_discount

Constraints

Indexes

RLS Policies

Rules

Triggers

booking_flight

flights

passengers

- passenger_id
- first name

pgAdmin 4

File Object Tools Edit View Window Help

airport_db/postgres@PostgreSQL 17* X

airport_db/postgres@PostgreSQL 17

Query Query History

```
1 begin ;
2
3 update booking
4 set price = price * 0.90
5 where booking_id = 10000 ;
6
7 rollback ;
```

Data Output Messages Notifications

ROLLBACK

Query returned successfully in 104 msec.

✓ Query returned successfully in 104 msec. ✕

Total rows: Query complete 00:00:00.104 CRLF Ln 7, Col 11

booking

- booking_id
- passenger_id
- booking_platform
- created_at
- update_at
- status
- price
- ticket_discount

Constraints

Indexes

RLS Policies

Rules

Triggers

booking_flight

flights

passengers

- passenger_id
- first name

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* X

airport_db/postgres@PostgreSQL 17

Query Query History

```
1 begin ;
2
3 update booking_flight
4 set flight_id = 22
5 where flight_id = 9;
6
7 commit ;
```

Scratch Pad X

Data Output Messages Notifications

COMMIT

Query returned successfully in 100 msec.

✓ Query returned successfully in 100 msec. ✕

Total rows: Query complete 00:00:00.100 CRLF Ln 7, Col 9

Object Explorer

- Operators
- Procedures
- Sequences
- 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