

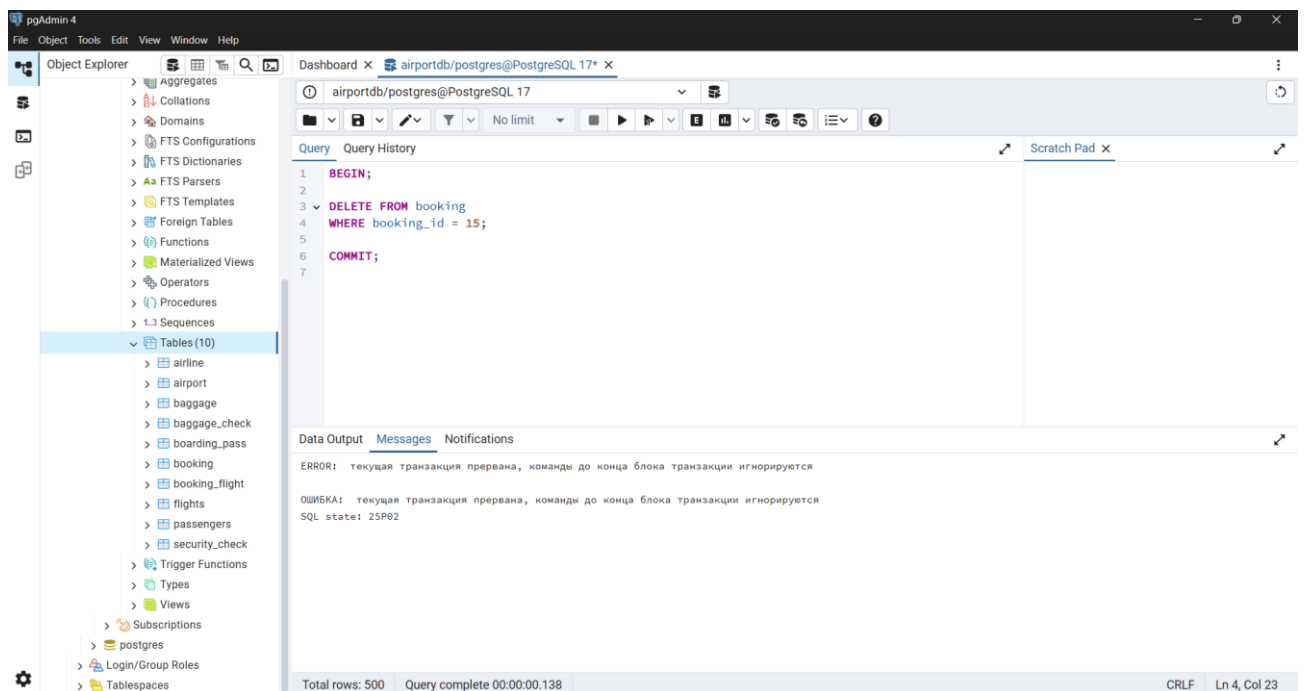
Laboratory work 9

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

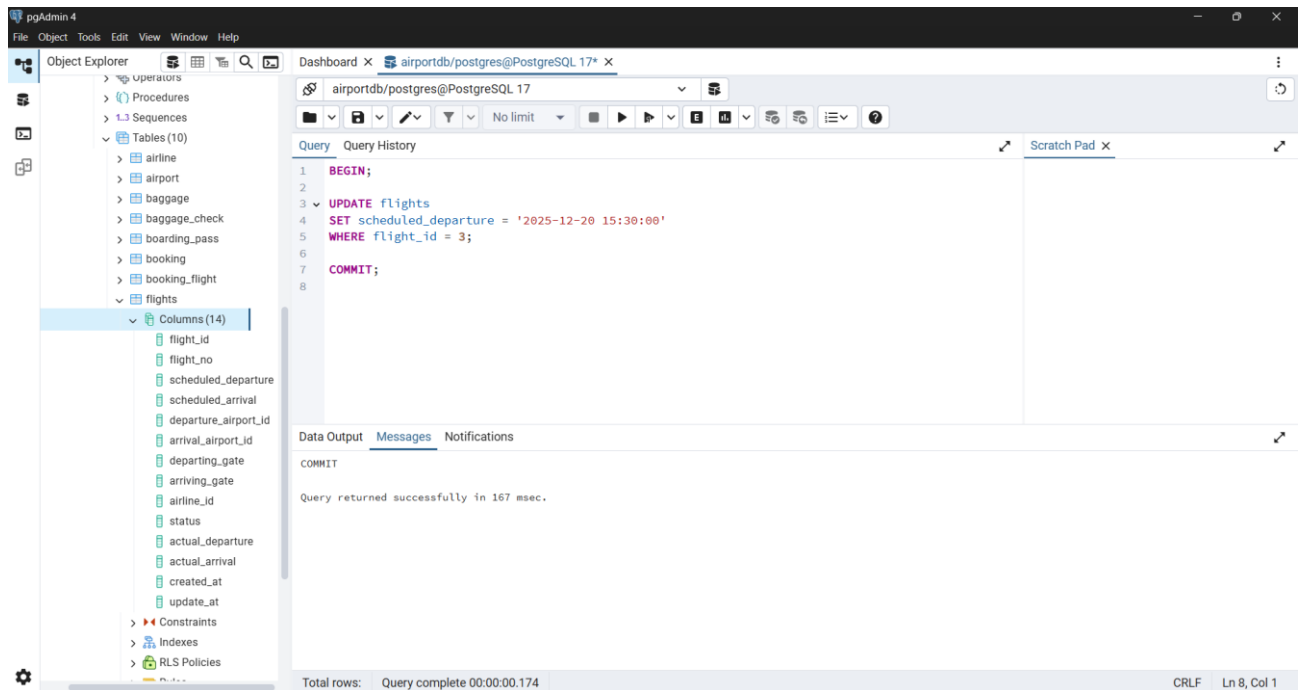
Take a full-page screenshot that covers the code and results of each task.

TRANSACTION.

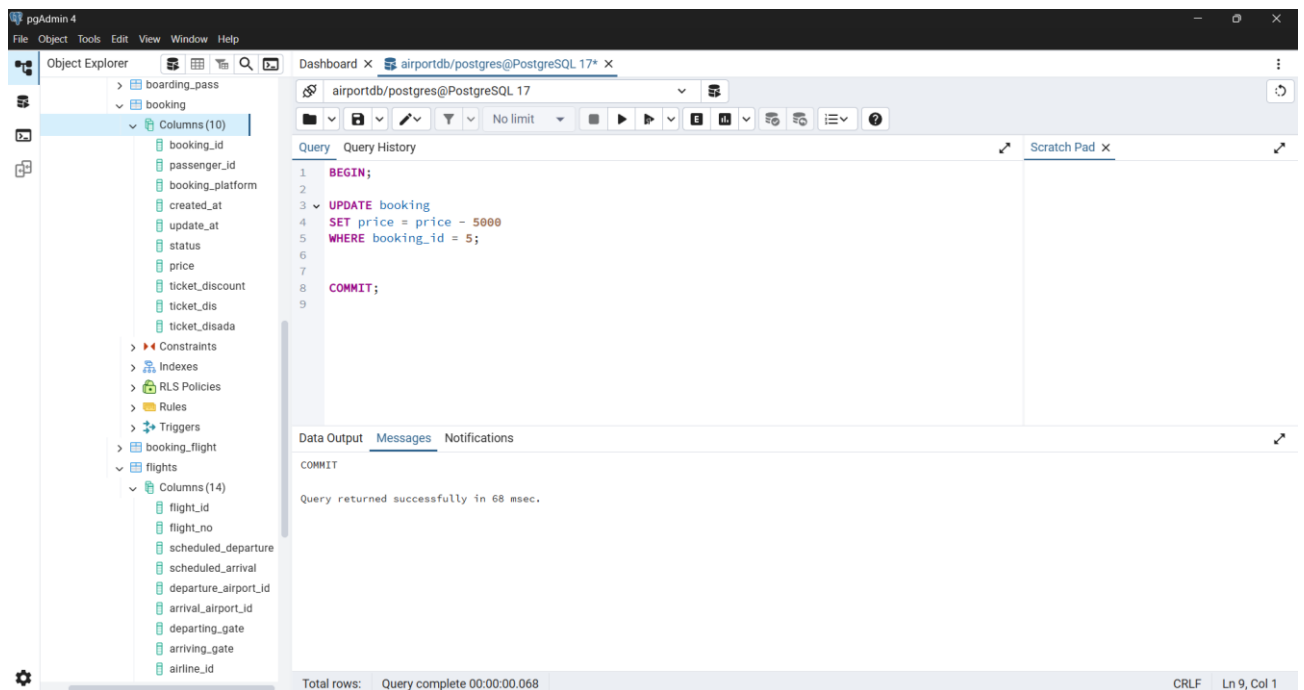
1. A passenger cancels their booking. You need to remove the booking for the flight. Ensure the 'booking' table no longer contains the booking. Simulate an error to test rollback (for example, invalid booking_id).



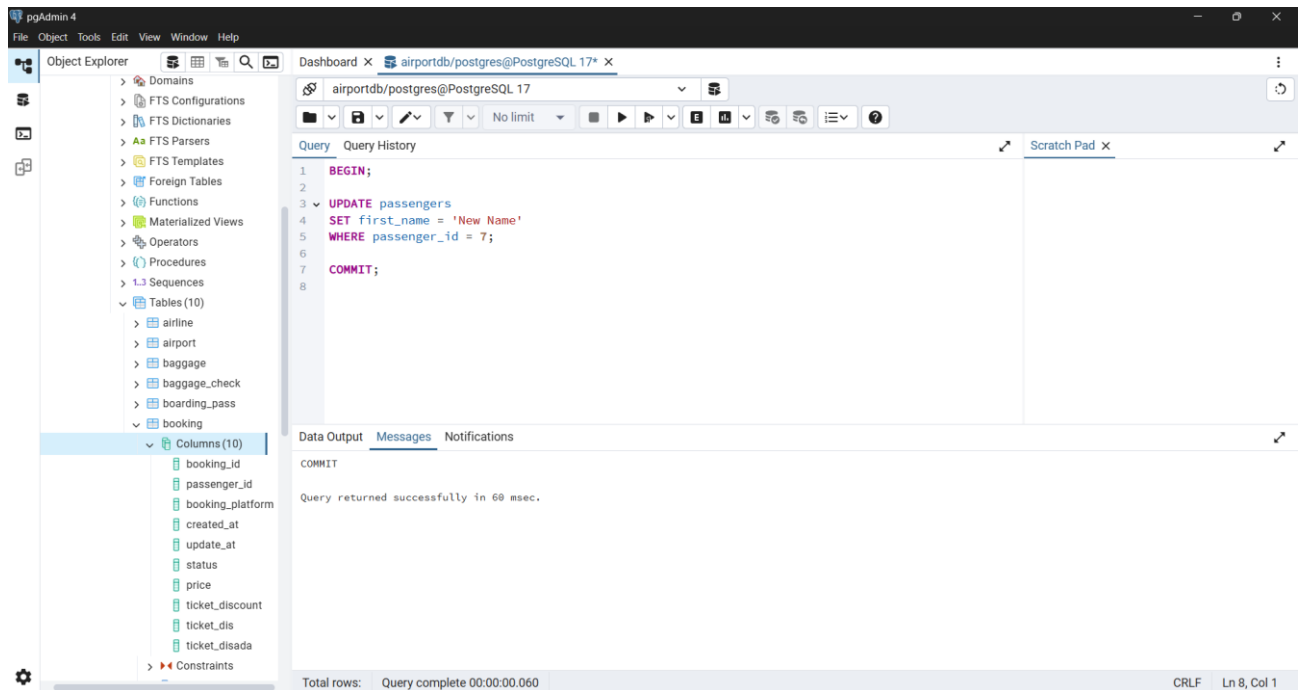
2. Rescheduling a flight. You need to reschedule a flight. Verify the 'flights' table reflects the new departure time. Simulate an error to test rollback (for example, invalid flight_id).



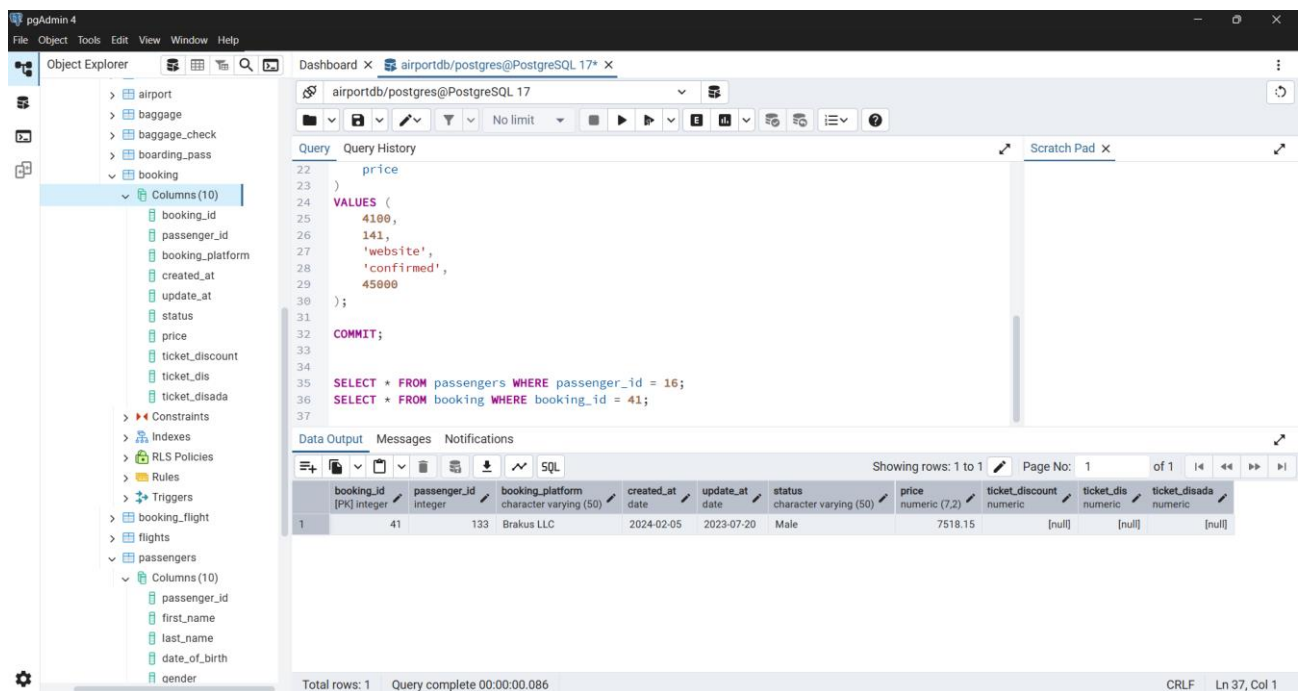
- Updating ticket prices. You need to decrease the ticket price for a specific flight for all existing bookings. If an error occurs, no changes should be applied.



- 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.



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

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer displays the database schema for 'airportdb', including tables like 'airport', 'baggage', 'boarding_pass', 'booking', 'booking_flight', 'flights', and 'passengers'. The 'booking' table is selected, showing its columns: booking_id, passenger_id, booking_platform, created_at, update_at, status, price, ticket_discount, ticket_dis, and ticket_disada.

The central Query Editor shows the following SQL script:

```

1 BEGIN;
2
3 UPDATE booking
4 SET price = price + 3000
5 WHERE price IS NOT NULL;
6
7 COMMIT;
8
9 SELECT booking_id, price FROM booking ORDER BY booking_id LIMIT 10;
10

```

The Data Output pane at the bottom displays the results of the query as a table with 10 rows:

booking_id [PK] integer	price numeric (7,2)
1	10462.13
2	7216.60
3	8782.37
4	3102.96
5	4711.04
6	10813.55
7	4346.71
8	8711.67
9	
10	

Total rows: 10. Query complete 00:00:00.170.

- 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. The Object Explorer on the left shows the 'baggage' table selected, with columns: baggage_id, passenger_id, booking_platform, created_at, update_at, status, price, ticket_discount, ticket_dis, and ticket_disada.

The central Query Editor shows the following SQL script:

```

1 BEGIN;
2
3 UPDATE baggage
4 SET weight_in_kg = 15,
5     update_date = now()
6 WHERE baggage_id = 10;
7
8 COMMIT;
9
10 SELECT * FROM baggage WHERE baggage_id = 10;
11

```

The Data Output pane at the bottom displays the results of the query as a table with 1 row:

baggage_id [PK] integer	weight_in_kg numeric (4,2)	created_date date	update_date date	booking_id integer
10	15.00	2023-12-15	2025-11-25	10

Total rows: 1. Query complete 00:00:00.207.

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

pgAdmin 4

Object Explorer

- airportdb
- baggage
- baggage_check
- boarding_pass
- booking
 - Columns(10)
 - booking_id
 - passenger_id
 - booking_platform
 - created_at
 - update_at
 - status
 - price
 - ticket_discount
 - ticket_dis
 - ticket_disada
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - booking_flight
 - flights
 - passengers
 - Columns(10)
 - passenger_id
 - first_name
 - last_name
 - date_of_birth
 - gender

Dashboard X airportdb/postgres@PostgreSQL 17 X

airportdb/postgres@PostgreSQL 17

Query Query History

```
1 BEGIN;  
2  
3 UPDATE booking  
4 SET ticket_discount = COALESCE(ticket_discount, 0) + 20,  
5 price = price * 0.8  
6 WHERE booking_id = 30  
7 AND passenger_id = 12;  
8  
9  
10  
11 ROLLBACK;  
12  
13 SELECT booking_id, passenger_id, price, ticket_discount FROM booking WHERE booking_id = 30;  
14
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

booking_id [PK] integer	passenger_id integer	price numeric (7,2)	ticket_discount numeric
1	30	53	8401.32

Total rows: 1 Query complete 00:00:00.158 CRLF Ln 14, Col 1

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

pgAdmin 4

Object Explorer

- airportdb
- baggage
- baggage_check
- boarding_pass
- booking
 - Columns(10)
 - booking_id
 - passenger_id
 - booking_platform
 - created_at
 - update_at
 - status
 - price
 - ticket_discount
 - ticket_dis
 - ticket_disada
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - booking_flight
 - flights
 - passengers
 - Columns(10)
 - passenger_id
 - first_name
 - last_name
 - date_of_birth
 - gender

Dashboard X airportdb/postgres@PostgreSQL 17 X

airportdb/postgres@PostgreSQL 17

Query Query History

```
1 BEGIN;  
2  
3 UPDATE booking  
4 SET status = 'rescheduled'  
5 WHERE passenger_id IN (  
6 SELECT passenger_id  
7 FROM booking  
8 );  
9  
10  
11 COMMIT;  
12  
13 SELECT booking_id, passenger_id, status FROM booking LIMIT 10;  
14
```

Data Output Messages Notifications

Showing rows: 1 to 10 Page No: 1 of 1

booking_id [PK] integer	passenger_id integer	status character varying (50)
1	1	rescheduled
2	6	rescheduled
3	7	rescheduled
4	8	rescheduled
5	9	rescheduled
6	10	rescheduled
7	11	rescheduled
8	12	rescheduled
9	13	rescheduled
10	14	rescheduled

Total rows: 10 Query complete 00:00:00.169 CRLF Ln 14, Col 1