

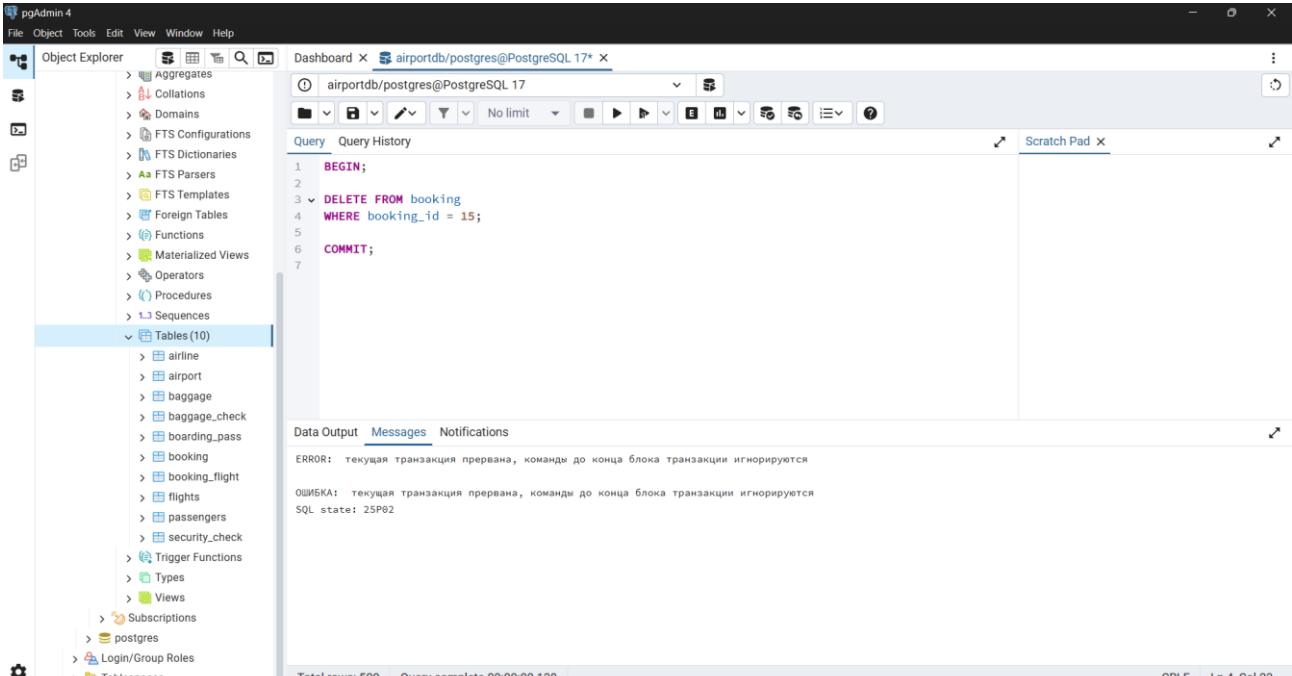
Laboratory work 9

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).



The screenshot shows the pgAdmin 4 interface. In the Object Explorer, the 'Tables' node under 'Tables (10)' is selected. In the main query editor, the following SQL code is present:

```
1 BEGIN;
2
3 v DELETE FROM booking
4 WHERE booking_id = 15;
5
6 COMMIT;
7
```

The 'Data Output' tab shows the following error message:

ERROR: текущая транзакция прервана, команды до конца блока транзакции игнорируются

ОШИБКА: текущая транзакция прервана, команды до конца блока транзакции игнорируются
SQL state: 25P02

Total rows: 500 Query complete 00:00:00.138 CRLF Ln 4, Col 23

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. In the Object Explorer, the 'flights' table is selected, showing 14 columns. In the main pane, a query is run:

```

1 BEGIN;
2
3 UPDATE flights
4 SET scheduled_departure = '2025-12-20 15:30:00'
5 WHERE flight_id = 3;
6
7 COMMIT;
8

```

The Data Output tab shows the message: "Query returned successfully in 167 msec." and "COMMIT". The status bar at the bottom right indicates "CRLF Ln 8, Col 1".

- 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. In the Object Explorer, the 'booking' table is selected, showing 10 columns. In the main pane, a query is run:

```

1 BEGIN;
2
3 UPDATE booking
4 SET price = price - 5000
5 WHERE booking_id = 5;
6
7
8 COMMIT;
9

```

The Data Output tab shows the message: "Query returned successfully in 68 msec." and "COMMIT". The status bar at the bottom right indicates "CRLF Ln 9, Col 1".

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

The screenshot shows the pgAdmin 4 interface with the 'Object Explorer' on the left and a 'Query' window on the right. The 'Query' window contains the following SQL code:

```

1 BEGIN;
2
3 UPDATE passengers
4   SET first_name = 'New Name'
5 WHERE passenger_id = 7;
6
7 COMMIT;
8

```

The 'Data Output' tab shows the result of the query:

```

COMMIT
Query returned successfully in 60 msec.

Total rows: Query complete 00:00:00.060
CRLF Ln 8, Col 1

```

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

The screenshot shows the pgAdmin 4 interface with the 'Object Explorer' on the left and a 'Query' window on the right. The 'Query' window contains the following SQL code:

```

22     price
23 )
24 VALUES (
25     4100,
26     141,
27     'website',
28     'confirmed',
29     45000
30 );
31
32 COMMIT;
33
34
35 SELECT * FROM passengers WHERE passenger_id = 16;
36 SELECT * FROM booking WHERE booking_id = 41;
37

```

The 'Data Output' tab shows the results of the queries:

booking_id	passenger_id	booking_platform	created_at	update_at	status	price	ticket_discount	ticket_dis	ticket_disada
41	133	Brakus LLC	2024-02-05	2023-07-20	Male	7518.15	[null]	[null]	[null]

Total rows: 1 Query complete 00:00:00.086
CRLF Ln 37, Col 1

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

The screenshot shows the pgAdmin 4 interface with the Object Explorer on the left and a query editor on the right. The Object Explorer displays the schema structure with tables like airport, baggage, baggage_check, boarding_pass, and booking. The booking table is selected, showing its 10 columns: booking_id, passenger_id, booking_platform, created_at, update_at, status, price, ticket_discount, ticket_dis, and ticket_disada. The query editor contains the following SQL code:

```

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 tab shows the results of the last query, which returns 10 rows of booking data. The first few rows are:

booking_id	price
1	10462.13
2	7216.60
3	8782.37
4	3102.96
5	4711.04
6	10813.55
7	4346.71
8	8711.67

Total rows: 10 Query complete 00:00:00.170 CRLF Ln 10, Col 1

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 with the Object Explorer on the left and a query editor on the right. The Object Explorer displays the schema structure with tables like airport, baggage, baggage_check, boarding_pass, and booking. The baggage table is selected, showing its 5 columns: baggage_id, passenger_id, booking_platform, created_at, and update_at. The query editor contains the following SQL code:

```

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 tab shows the results of the last query, which returns 1 row of baggage data. The row is:

baggage_id	weight_in_kg	created_date	update_date	booking_id
10	15.00	2023-12-15	2025-11-25	10

Total rows: 1 Query complete 00:00:00.207 CRLF Ln 9, Col 1

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

Object Explorer Dashboard 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 ROLLBACK;
11
12
13 SELECT booking_id, passenger_id, price, ticket_discount FROM booking WHERE booking_id = 30;
14

```

Data Output Messages Notifications

booking_id	passenger_id	price	ticket_discount
30	53	8401.32	[null]

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

File Object Tools Edit View Window Help

Object Explorer Dashboard 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 COMMIT;
11
12
13 SELECT booking_id, passenger_id, status FROM booking LIMIT 10;
14

```

Data Output Messages Notifications

booking_id	passenger_id	status
1	156	rescheduled
2	120	rescheduled
3	14	rescheduled
4	197	rescheduled
5	160	rescheduled
6	92	rescheduled
7	35	rescheduled
8	122	rescheduled

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