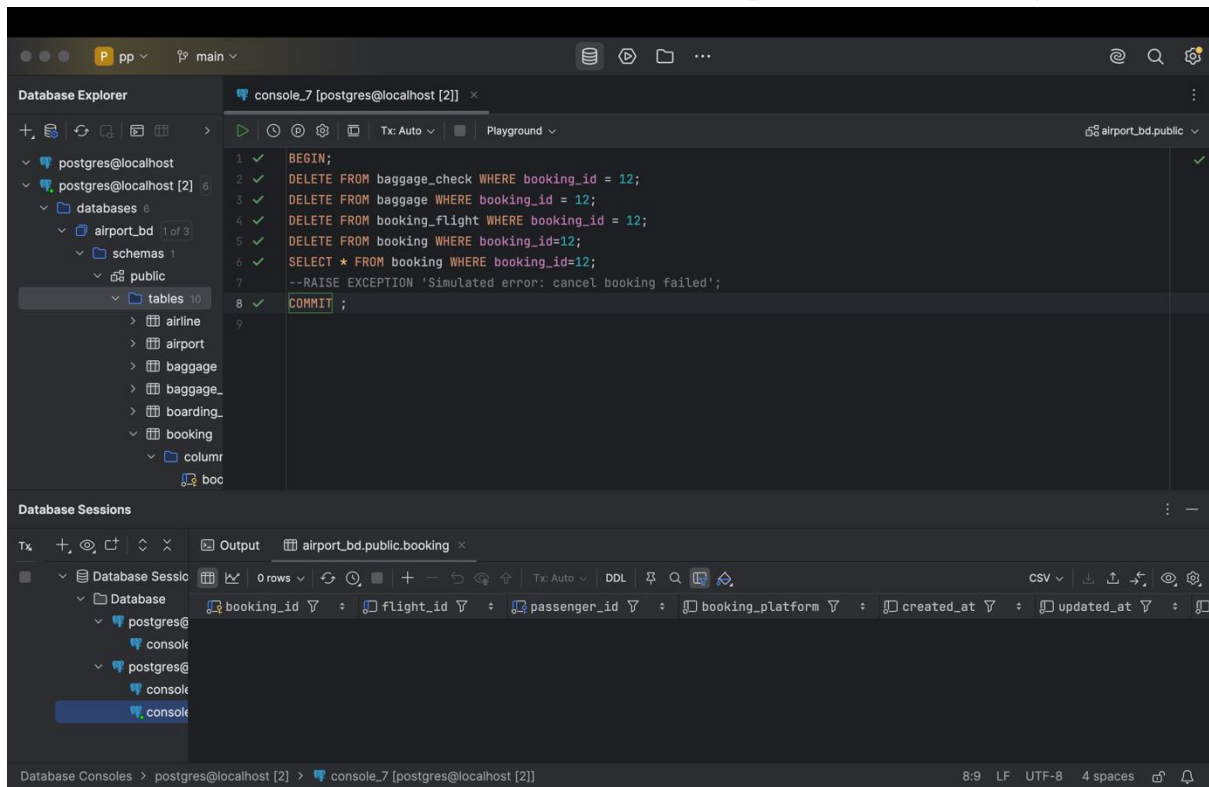


## Laboratory work 9

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)

The screenshot shows a database console with the following SQL code:

```

1 ROLLBACK ;
2
3 BEGIN;
4 UPDATE flights
5 SET sch_departure_time = '2025-12-12 10:00:00',
6     sch_arrival_time = '2025-12-12 13:00:00',
7     updated_at=now()
8 WHERE flight_id = 6;
9
10 SELECT flight_id,sch_departure_time, sch_arrival_time FROM flights
11 WHERE flight_id=6;
12
13 -- RAISE EXCEPTION 'Simulated error: reschedule flight';
14 COMMIT ;

```

The output shows the result of the SELECT query:

flight_id	sch_departure_time	sch_arrival_time
6	2025-12-12 10:00:00.000000	2025-12-12 13:00:00.000000

- 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 a database console with the following SQL code:

```

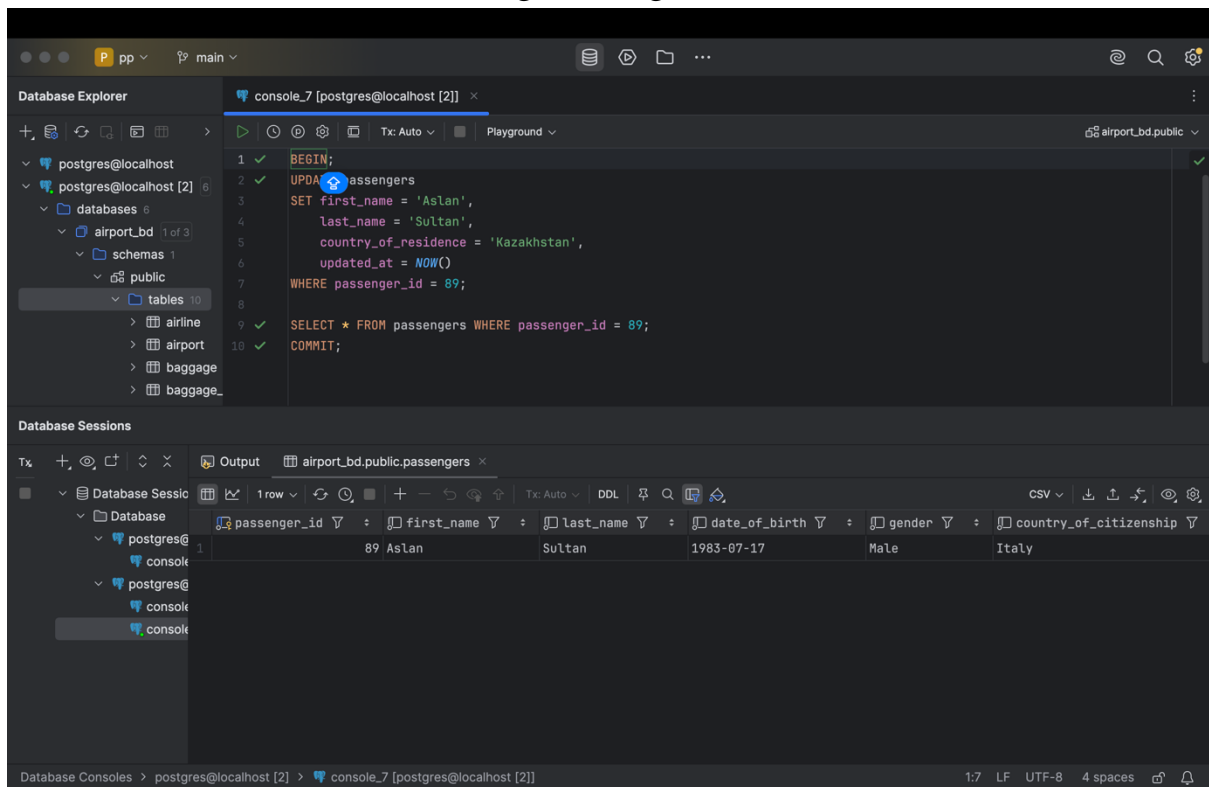
1 BEGIN;
2 UPDATE booking
3 SET ticket_price = ticket_price - 10,
4     updated_at = NOW()
5 WHERE flight_id = 5;
6 SELECT booking_id, ticket_price
7 FROM booking
8 WHERE flight_id = 5;
9 -- RAISE EXCEPTION 'Simulated error reducing price';
10 COMMIT;

```

The output shows the result of the SELECT query:

booking_id	ticket_price
1	340.50
2	540.25
3	300.00
4	540.25
5	310.00
6	545.00
7	555.00
8	315.00
9	560.00
10	325.00
11	565.00

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



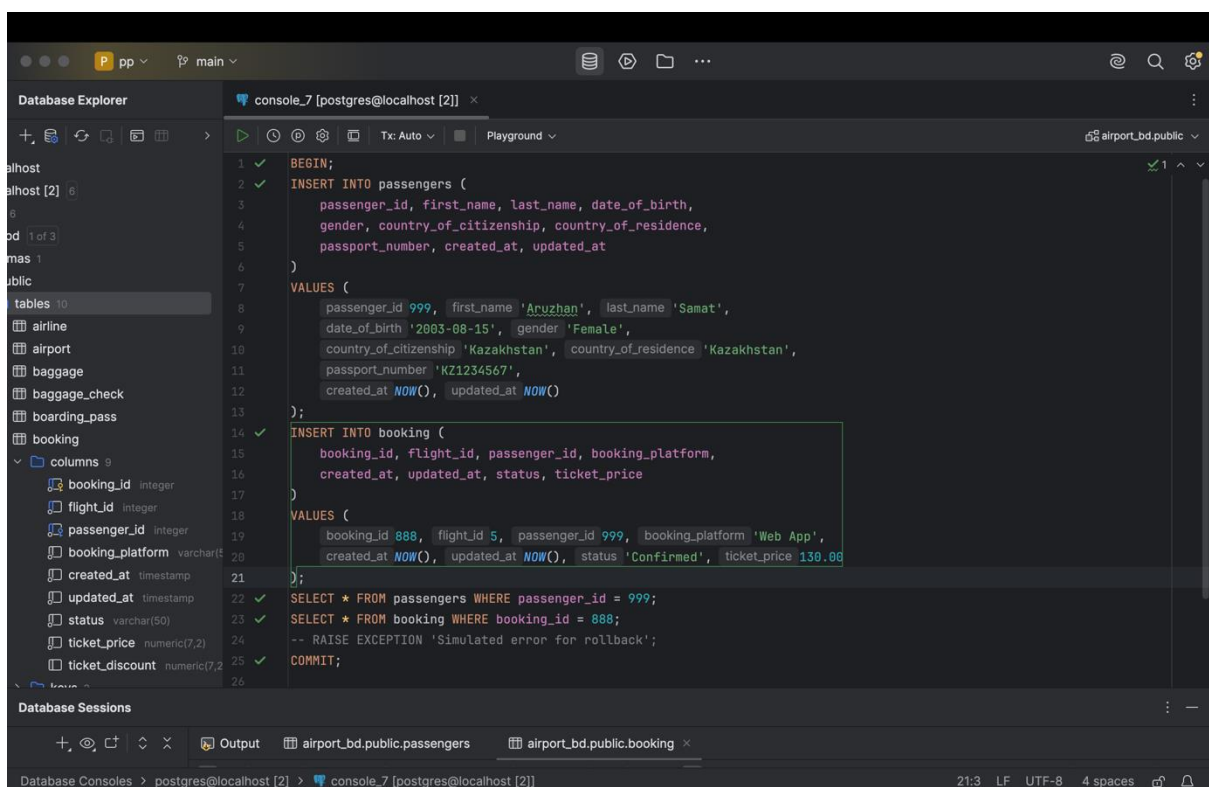
The screenshot shows a PostgreSQL console window with the following SQL query executed:

```
1 BEGIN;
2 UPDATE passengers
3 SET first_name = 'Aslan',
4     last_name = 'Sultan',
5     country_of_residence = 'Kazakhstan',
6     updated_at = NOW()
7 WHERE passenger_id = 89;
8
9 SELECT * FROM passengers WHERE passenger_id = 89;
10 COMMIT;
```

The result of the SELECT query is displayed in the Output pane:

passenger_id	first_name	last_name	date_of_birth	gender	country_of_citizenship
89	Aslan	Sultan	1983-07-17	Male	Italy

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



The screenshot shows a PostgreSQL console window with the following SQL queries executed:

```
1 BEGIN;
2 INSERT INTO passengers (
3     passenger_id, first_name, last_name, date_of_birth,
4     gender, country_of_citizenship, country_of_residence,
5     passport_number, created_at, updated_at
6 )
7 VALUES (
8     passenger_id 999, first_name 'Aruzhan', last_name 'Samat',
9     date_of_birth '2003-08-15', gender 'Female',
10    country_of_citizenship 'Kazakhstan', country_of_residence 'Kazakhstan',
11    passport_number 'KZ1234567',
12    created_at NOW(), updated_at NOW()
13 );
14 INSERT INTO booking (
15     booking_id, flight_id, passenger_id, booking_platform,
16     created_at, updated_at, status, ticket_price
17 )
18 VALUES (
19     booking_id 888, flight_id 5, passenger_id 999, booking_platform 'Web App',
20     created_at NOW(), updated_at NOW(), status 'Confirmed', ticket_price 130.00
21 );
22 SELECT * FROM passengers WHERE passenger_id = 999;
23 SELECT * FROM booking WHERE booking_id = 888;
24 -- RAISE EXCEPTION 'Simulated error for rollback';
25 COMMIT;
```

The image displays two screenshots of a PostgreSQL IDE interface, likely DBeaver, showing SQL queries and their results.

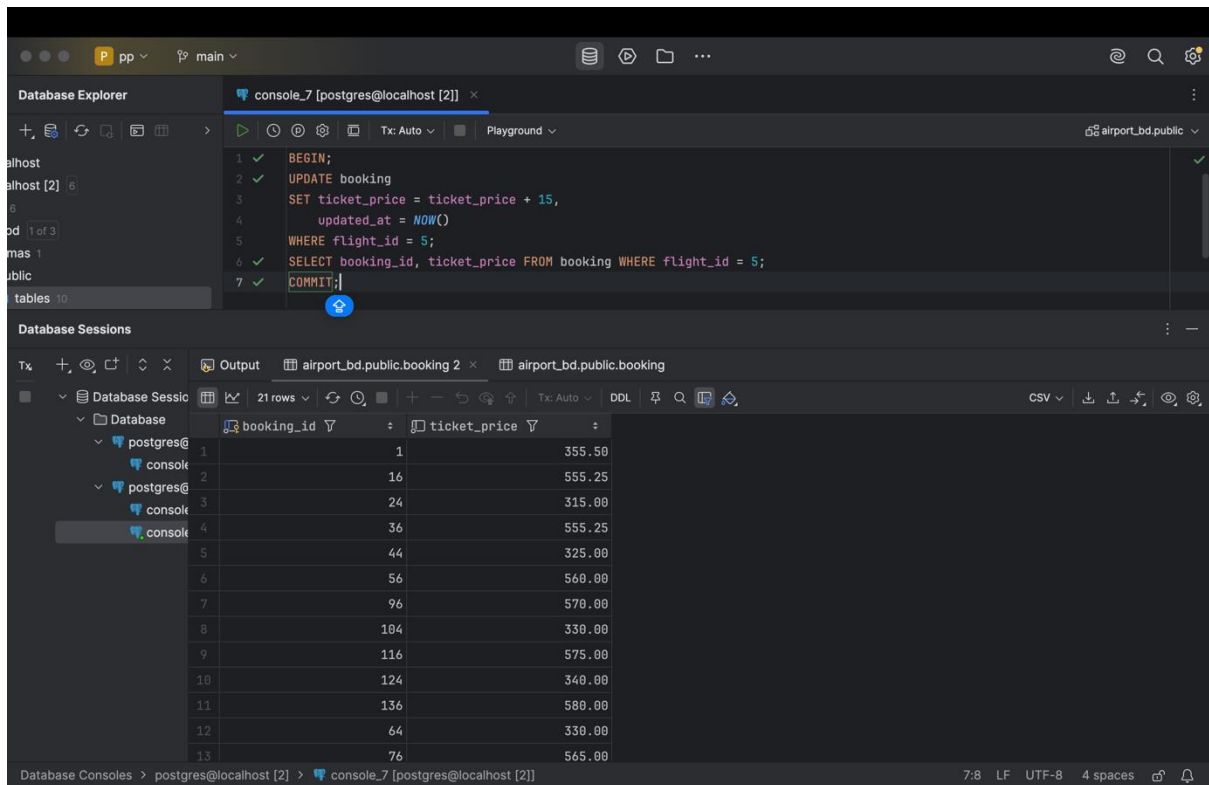
**Top Screenshot:**

- Database Explorer:** Shows a tree view with 'airline' and 'public' schemas.
- SQL Editor:** Contains an `INSERT INTO booking` statement. The `VALUES` clause includes `booking_id 888`, `flight_id 5`, `passenger_id 999`, `booking_platform 'Web App'`, `created_at NOW()`, `updated_at NOW()`, `status 'Confirmed'`, and `ticket_price 130.00`.
- Database Sessions:** Shows a table view for `airport_bd.public.booking` with columns: `booking_id`, `flight_id`, `passenger_id`, `booking_platform`, `created_at`, and `updated_at`. The first row shows values: 888, 5, 999, Web App, 2025-11-25 21:36:58.350713, 2025-11-25 21:36:58.350713.

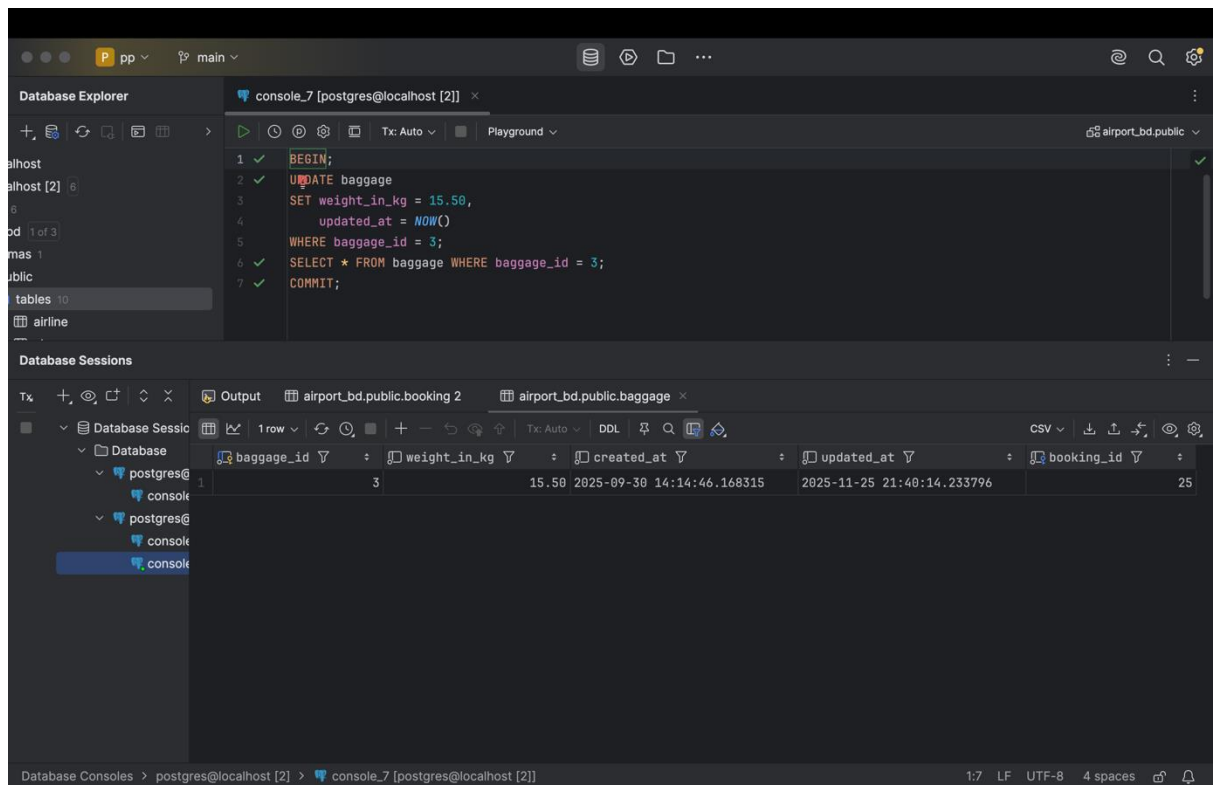
**Bottom Screenshot:**

- Database Explorer:** Similar to the top screenshot.
- SQL Editor:** Contains the same `INSERT INTO booking` statement as the top screenshot.
- Database Sessions:** Shows a table view for `airport_bd.public.passengers` with columns: `passenger_id`, `first_name`, `last_name`, `date_of_birth`, `gender`, and `country_of_citizenship`. The first row shows values: 999, Aruzhan, Samat, 2003-08-15, Female, Kazakhstan.

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



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



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

roll back.

The screenshot shows a database console interface with a SQL script being executed. The script performs an update on the 'booking' table and includes a rollback clause. The output window shows the state of the 'booking' table after the update.

```
1 BEGIN;
2 UPDATE booking
3 SET ticket_price = ticket_price * 0.85,
4     updated_at = NOW()
5 WHERE booking_id = 8;
6
7 SELECT booking_id, ticket_price FROM booking
8 WHERE booking_id = 8;
9 -- RAISE EXCEPTION 'Discount error: rollback required';
10 COMMIT;
```

booking_id	ticket_price
1	8
2	246.50

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

The screenshot shows a database console interface with a SQL script being executed. The script updates the 'flight\_id' for all bookings in the 'booking' table. The output window shows the state of the 'booking' table after the update.

```
1 BEGIN;
2 UPDATE booking
3 SET flight_id = 6,
4     updated_at = NOW()
5 WHERE flight_id = 5;
6
7 SELECT booking_id, flight_id FROM booking
8 WHERE flight_id IN (5,6);
9 -- RAISE EXCEPTION 'Reschedule failure: rollback';
10 COMMIT;
```

booking_id	flight_id
1	9
2	29
3	49
4	69
5	89
6	1
7	16
8	24
9	36