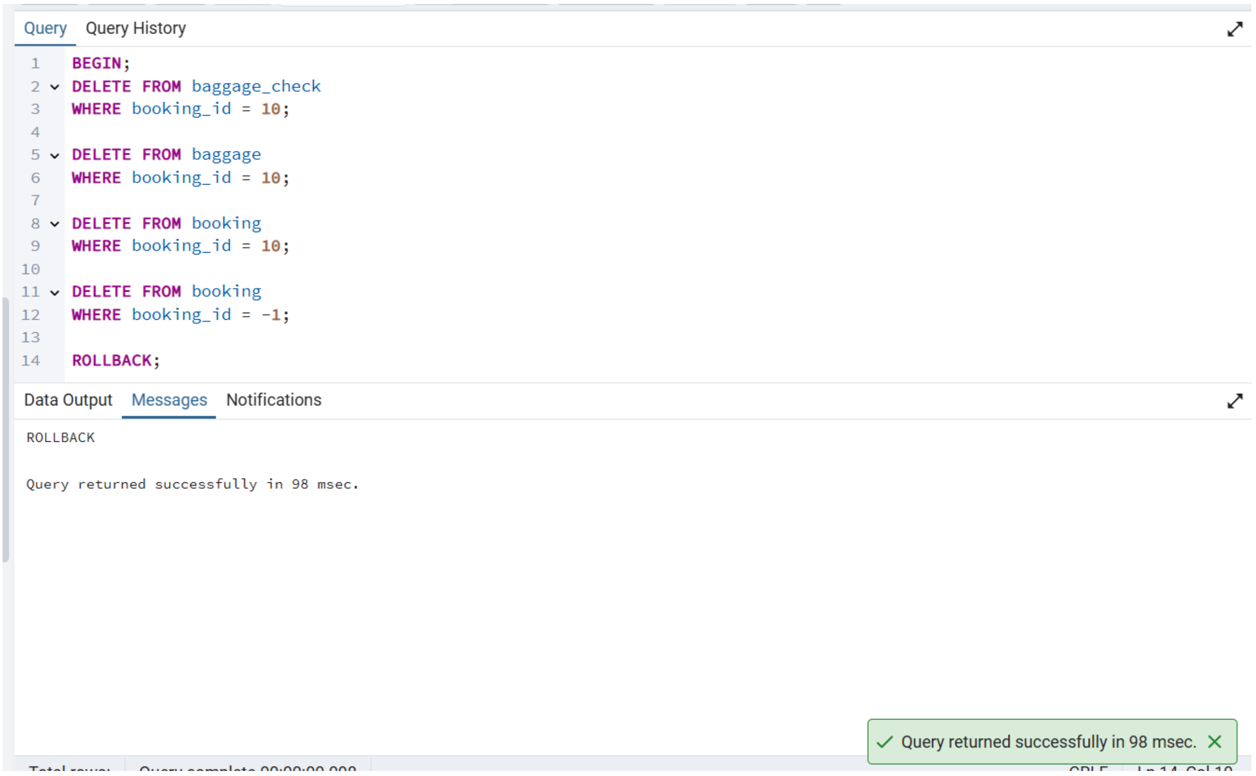
LAB 9. Abdykamat Adilet. TRANSACTION.

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).

```
BEGIN;

DELETE FROM baggage_check

WHERE booking_id = 10;


DELETE FROM baggage

WHERE booking_id = 10;


DELETE FROM booking

WHERE booking_id = 10;


DELETE FROM booking

WHERE booking_id = -1;


ROLLBACK;
```



```
BEGIN;
DELETE FROM baggage_check
WHERE booking_id = 10;

DELETE FROM baggage
WHERE booking_id = 10;

DELETE FROM booking
WHERE booking_id = 10;

DELETE FROM booking
WHERE booking_id = -1;

ROLLBACK;
```

```
ROLLBACK

Query returned successfully in 98 msec.
```

Query returned successfully in 98 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).

BEGIN;

UPDATE flights

SET sch_departure_time = '2025-01-30 14:00:00'

WHERE flight_id = 5;

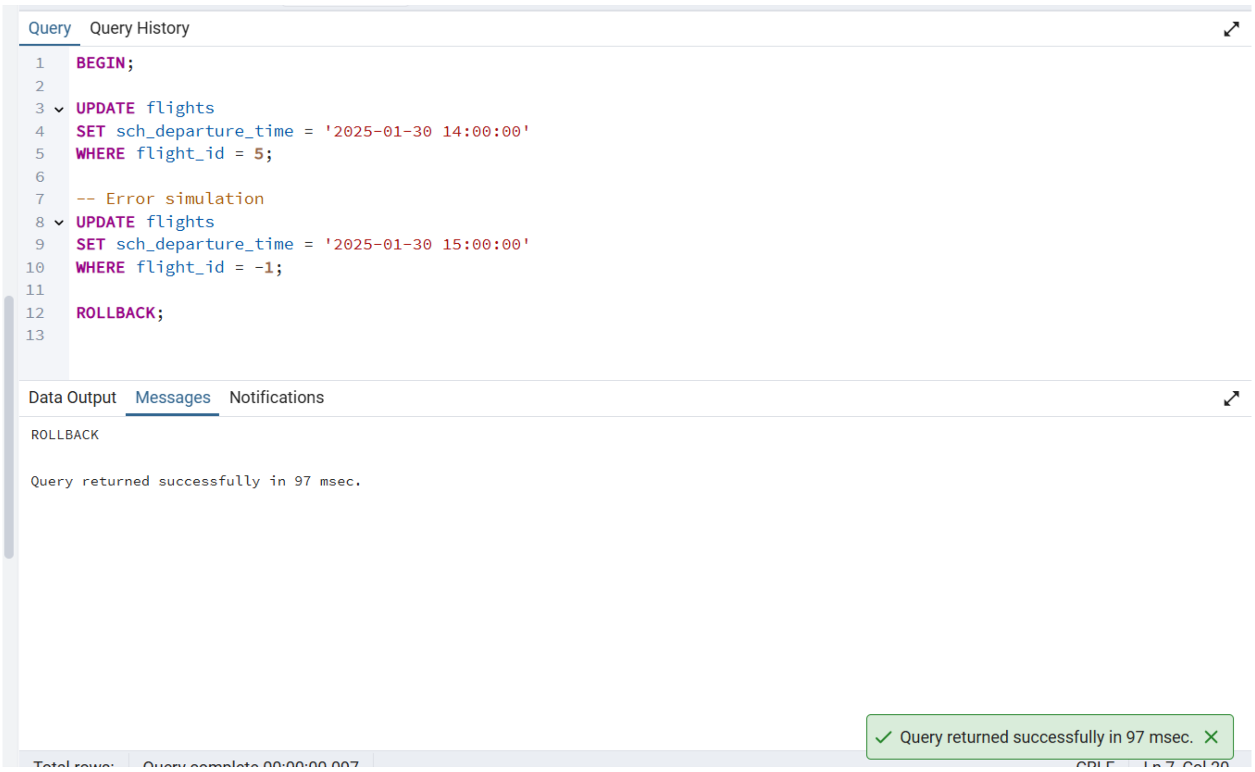
-- Error simulation

UPDATE flights

SET sch_departure_time = '2025-01-30 15:00:00'

WHERE flight_id = -1;

ROLLBACK;



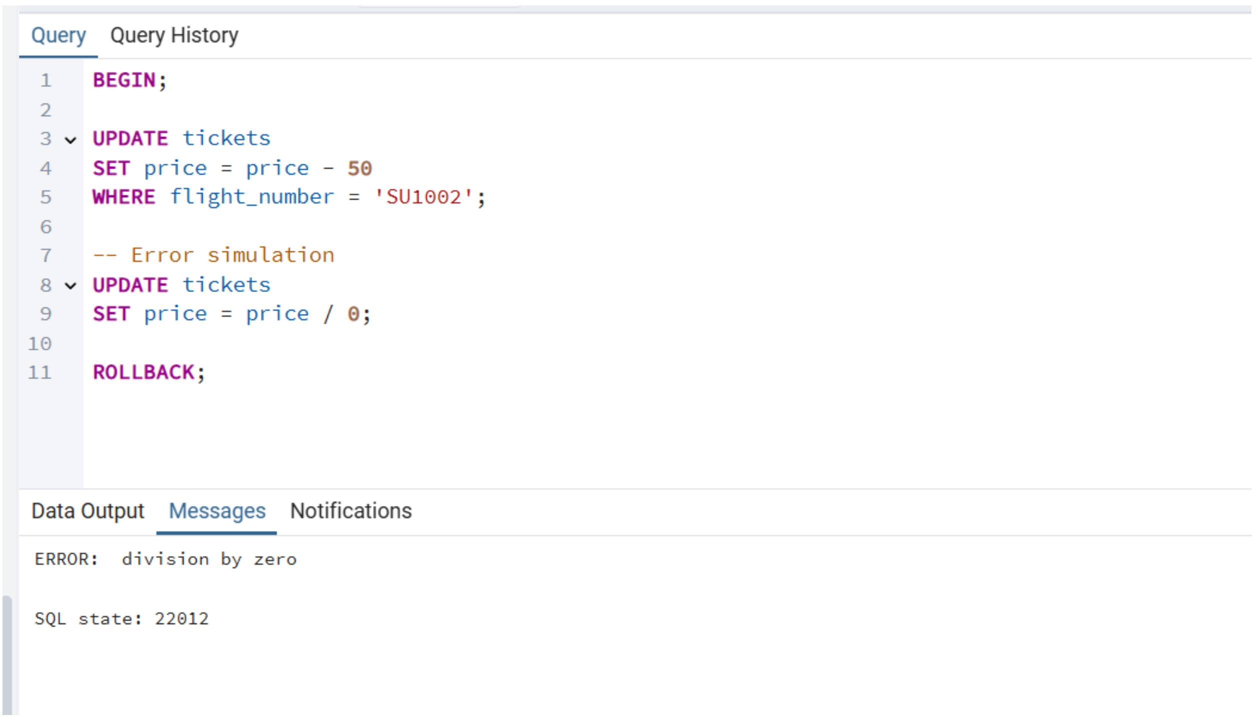
```
BEGIN;

UPDATE flights
SET sch_departure_time = '2025-01-30 14:00:00'
WHERE flight_id = 5;

-- Error simulation
UPDATE flights
SET sch_departure_time = '2025-01-30 15:00:00'
WHERE flight_id = -1;

ROLLBACK;
```

```
ROLLBACK

Query returned successfully in 97 msec.
```

✓ Query returned successfully in 97 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.

```
BEGIN;

UPDATE tickets

SET price = price - 50

WHERE flight_number = 'SU1002';

-- Error simulation

UPDATE tickets

SET price = price / 0;



ROLLBACK;
```



```
BEGIN;

UPDATE tickets
SET price = price - 50
WHERE flight_number = 'SU1002';

-- Error simulation
UPDATE tickets
SET price = price / 0;

ROLLBACK;
```

```
ERROR:  division by zero

SQL state: 22012
```

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

BEGIN;

UPDATE passengers

SET first_name = 'John',

  last_name = 'Williams',

  country_of_citizenship = 'USA'

WHERE passenger_id = 12;

UPDATE booking

SET passenger_id = 12

WHERE passenger_id = 12;

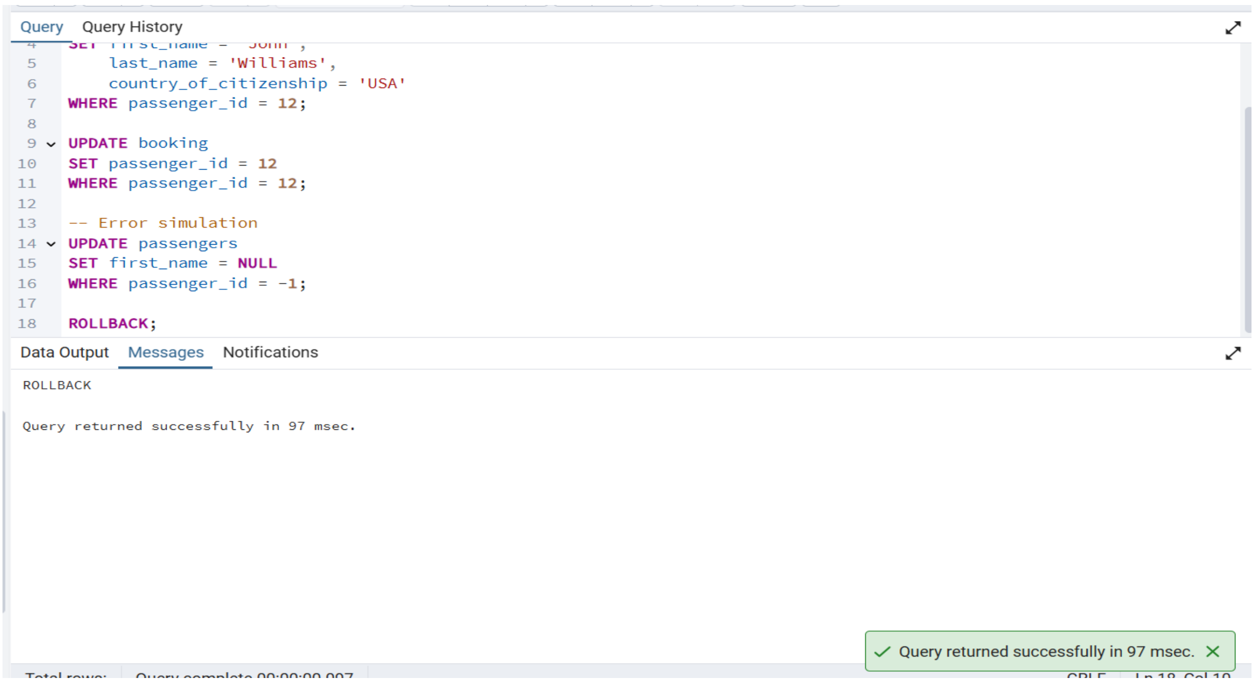
-- Error simulation

UPDATE passengers

SET first_name = NULL

WHERE passenger_id = -1;

ROLLBACK;



```
    last_name = 'Williams',
    country_of_citizenship = 'USA'
WHERE passenger_id = 12;

UPDATE booking
SET passenger_id = 12
WHERE passenger_id = 12;

-- Error simulation
UPDATE passengers
SET first_name = NULL
WHERE passenger_id = -1;

ROLLBACK;
```

```
ROLLBACK

Query returned successfully in 97 msec.
```

Query returned successfully in 97 msec.

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

BEGIN;

INSERT INTO passengers (first_name, last_name, country_of_citizenship, date_of_birth, passport_number,

gender,passenger_id, country_of_residence, created_at, updated_at)

VALUES ('Adam', 'Stone', 'Canada', '1990-01-01', 'AB1234567', 'M', '99', 'India', NOW(), NOW())

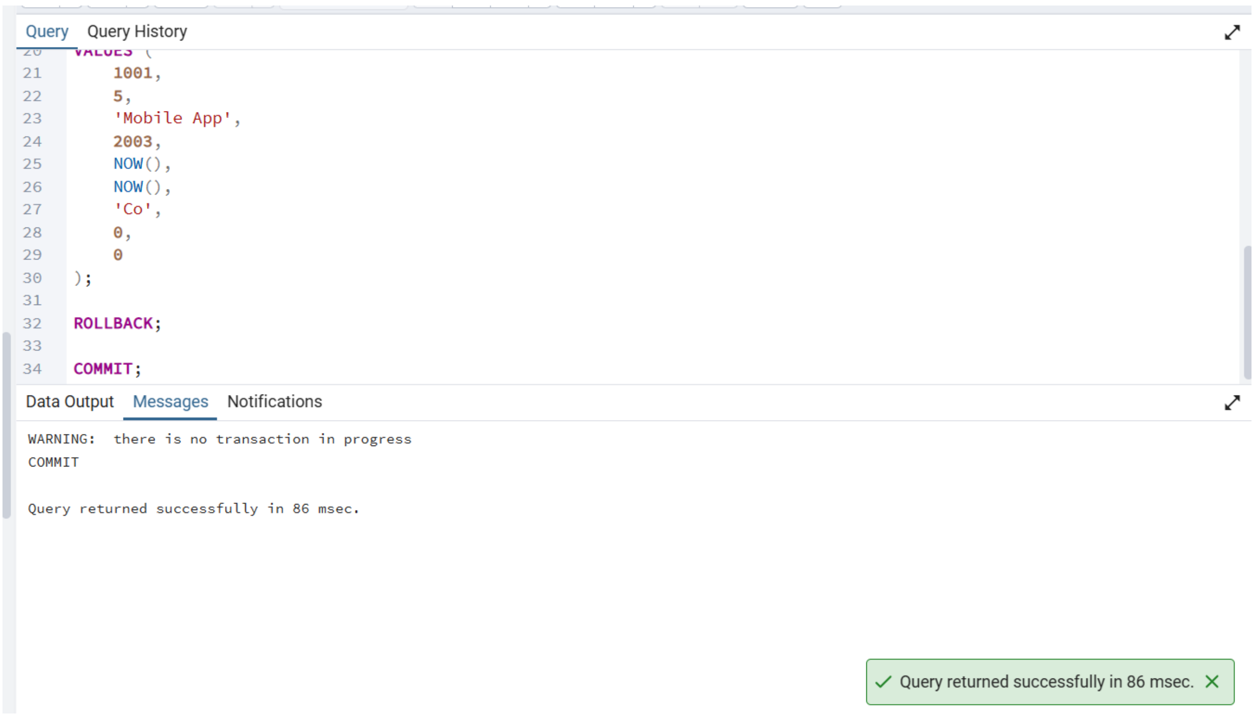
RETURNING passenger_id;

INSERT INTO booking (booking_id, passenger_id, booking_platform, flight_id, created_at, updated_at, status, ticket_price, ticket_discount)

VALUES (1001, 5, 'Mobile App', 2003, NOW(), NOW(), 'Co', 0, 0);

ROLLBACK;

COMMIT;



```
VALUES (
    1001,
    5,
    'Mobile App',
    2003,
    NOW(),
    NOW(),
    'Co',
    0,
    0
);

ROLLBACK;

COMMIT;
```

```
WARNING:  there is no transaction in progress
COMMIT

Query returned successfully in 86 msec.
```

Query returned successfully in 86 msec.

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

BEGIN;

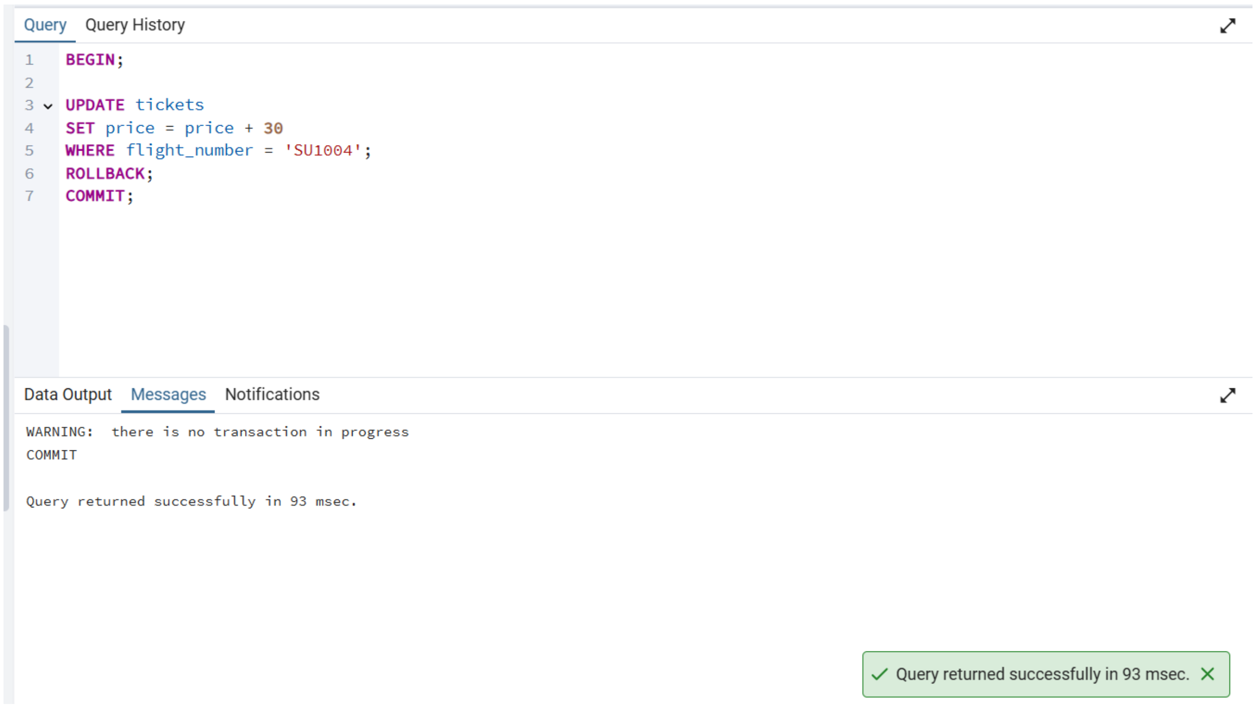
UPDATE tickets

SET price = price + 30

WHERE flight_number = 'SU1004';

ROLLBACK;

COMMIT;



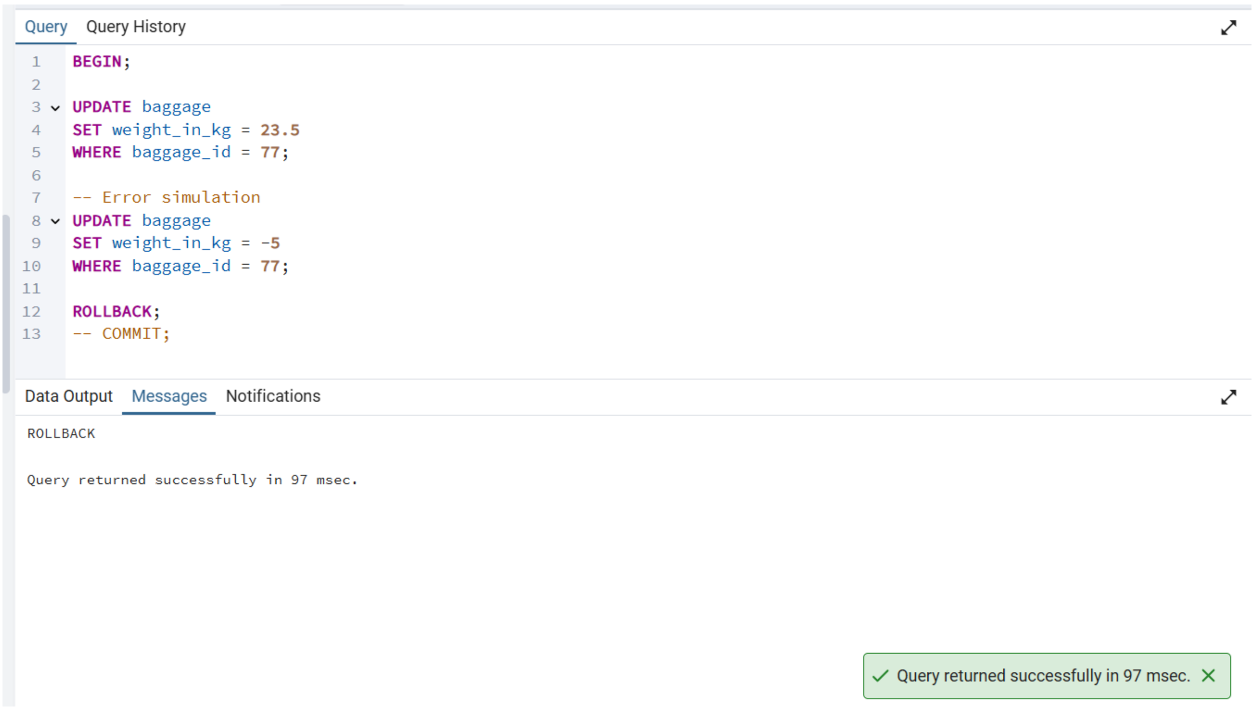
```
BEGIN;

UPDATE tickets
SET price = price + 30
WHERE flight_number = 'SU1004';
ROLLBACK;
COMMIT;
```

```
WARNING:  there is no transaction in progress
COMMIT

Query returned successfully in 93 msec.
```

✓ Query returned successfully in 93 msec.

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

```
BEGIN;


UPDATE baggage

SET weight_in_kg = 23.5

WHERE baggage_id = 77;


-- Error simulation

UPDATE baggage

SET weight_in_kg = -5

WHERE baggage_id = 77;


ROLLBACK;

-- COMMIT;
```



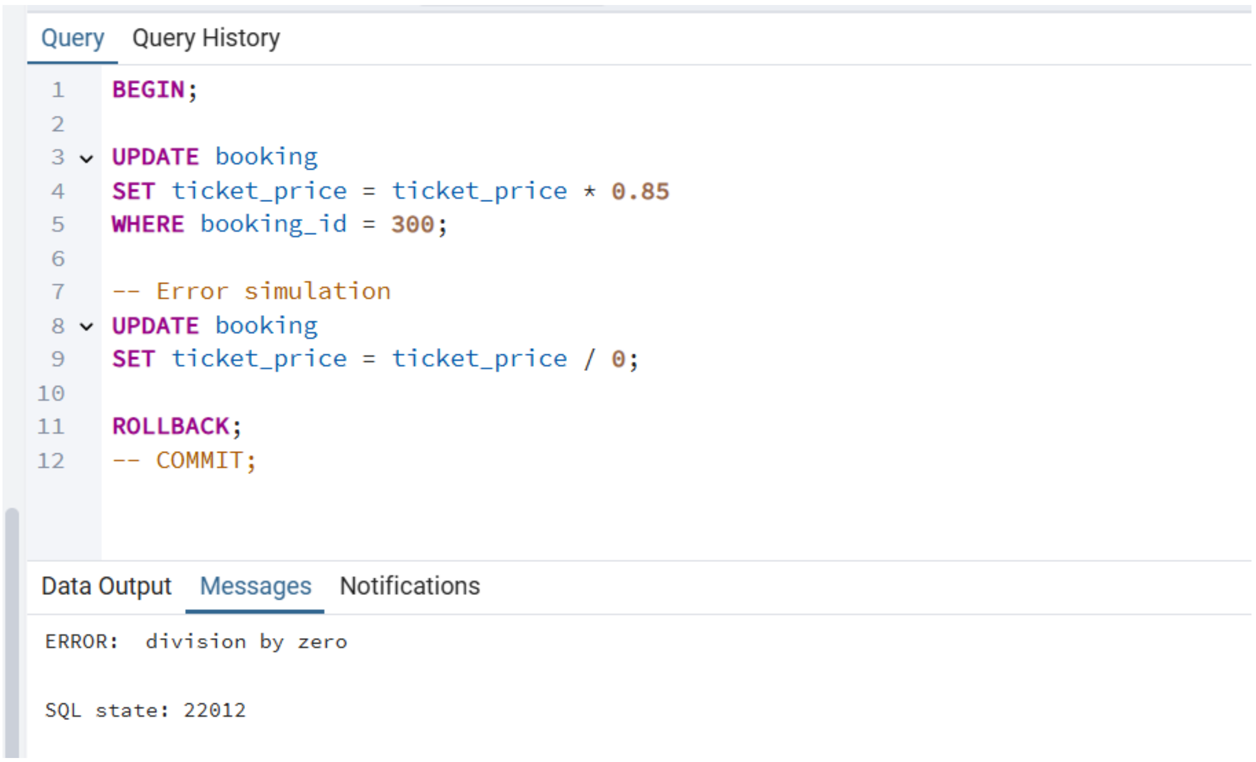
```
BEGIN;

UPDATE baggage
SET weight_in_kg = 23.5
WHERE baggage_id = 77;

-- Error simulation
UPDATE baggage
SET weight_in_kg = -5
WHERE baggage_id = 77;

ROLLBACK;
-- COMMIT;
```

```
ROLLBACK

Query returned successfully in 97 msec.
```

Query returned successfully in 97 msec.

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

```
BEGIN;


UPDATE booking

SET ticket_price = ticket_price * 0.85

WHERE booking_id = 300;


-- Error simulation

UPDATE booking

SET ticket_price = ticket_price / 0;


ROLLBACK;

-- COMMIT;
```

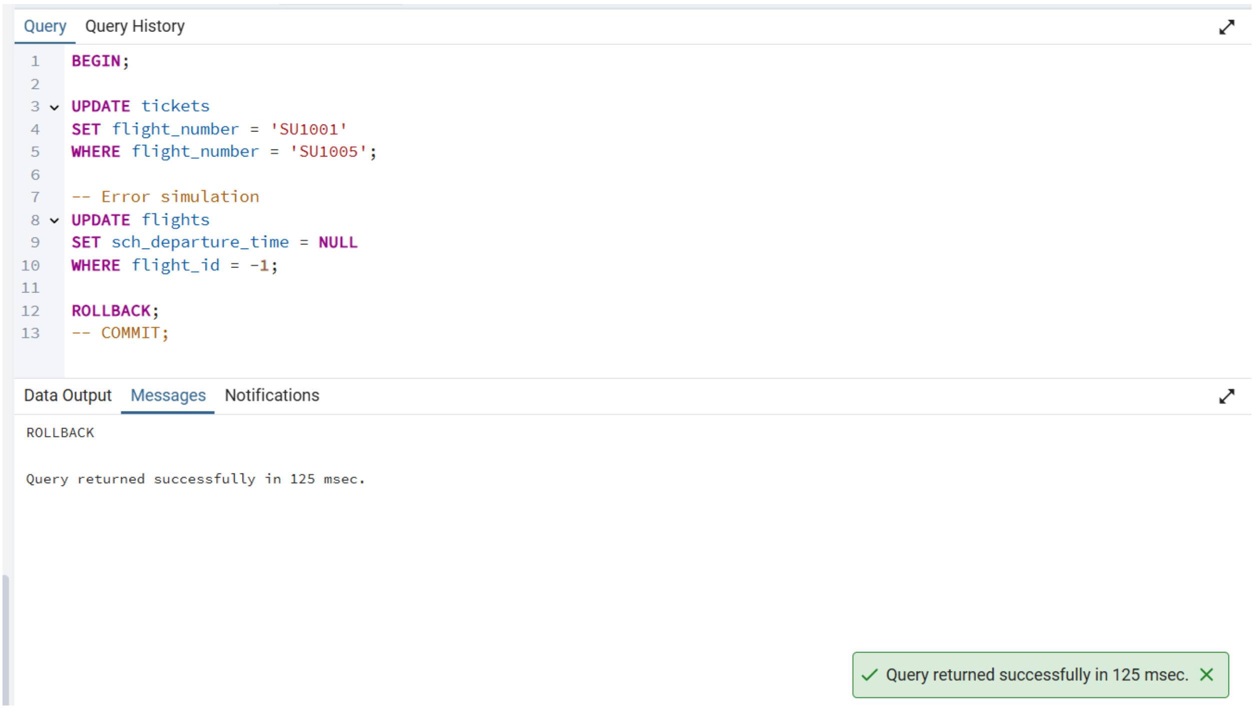


```
BEGIN;

UPDATE booking
SET ticket_price = ticket_price * 0.85
WHERE booking_id = 300;

-- Error simulation
UPDATE booking
SET ticket_price = ticket_price / 0;

ROLLBACK;
-- COMMIT;
```

```
ERROR:  division by zero

SQL state: 22012
```

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

```
BEGIN;


UPDATE tickets

SET flight_number = 'SU1001'

WHERE flight_number = 'SU1005';


-- Error simulation

UPDATE flights

SET sch_departure_time = NULL

WHERE flight_id = -1;


ROLLBACK;

-- COMMIT;
```



```
BEGIN;

UPDATE tickets
SET flight_number = 'SU1001'
WHERE flight_number = 'SU1005';

-- Error simulation
UPDATE flights
SET sch_departure_time = NULL
WHERE flight_id = -1;

ROLLBACK;
-- COMMIT;
```

```
ROLLBACK

Query returned successfully in 125 msec.
```

Query returned successfully in 125 msec.