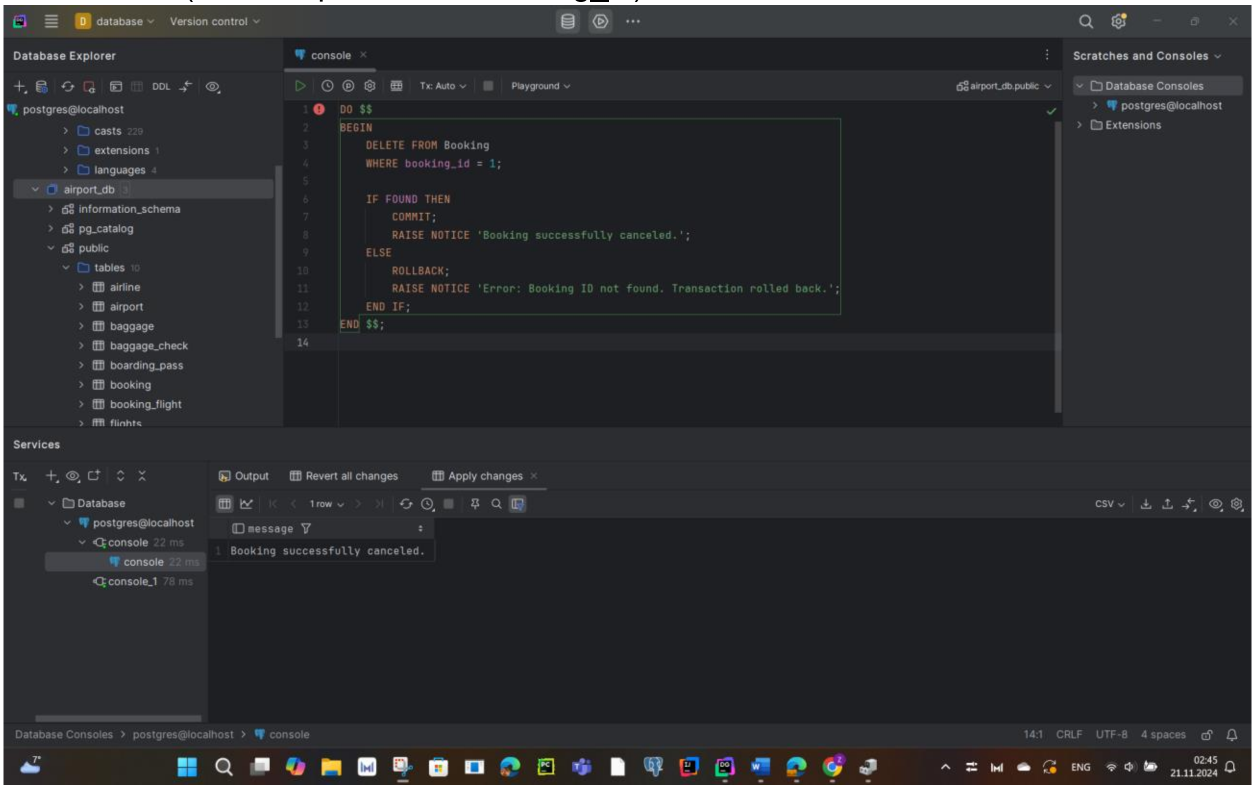
Laboratory work 9

TRANSACTION.

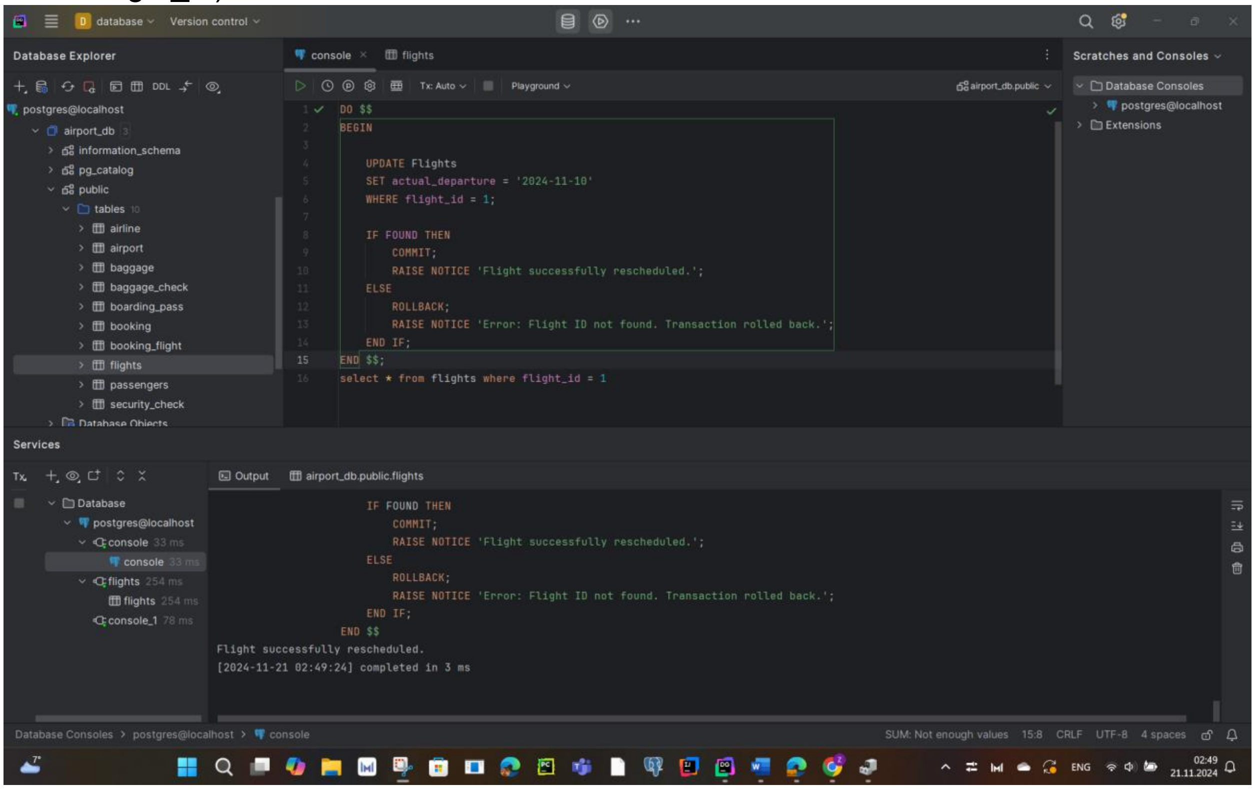
1. A passenger cancels their booking. You need to remove the booking for the flight. Ensure the 'booking' table no longer contains the booking. Simulate an error to test rollback (for example, invalid booking_id).



```sql
DO $$
BEGIN
    DELETE FROM Booking
    WHERE booking_id = 1;

    IF FOUND THEN
        COMMIT;
        RAISE NOTICE 'Booking successfully canceled.';
    ELSE
        ROLLBACK;
        RAISE NOTICE 'Error: Booking ID not found. Transaction rolled back.';
    END IF;
END $$;
```

```
message
Booking successfully canceled.
```

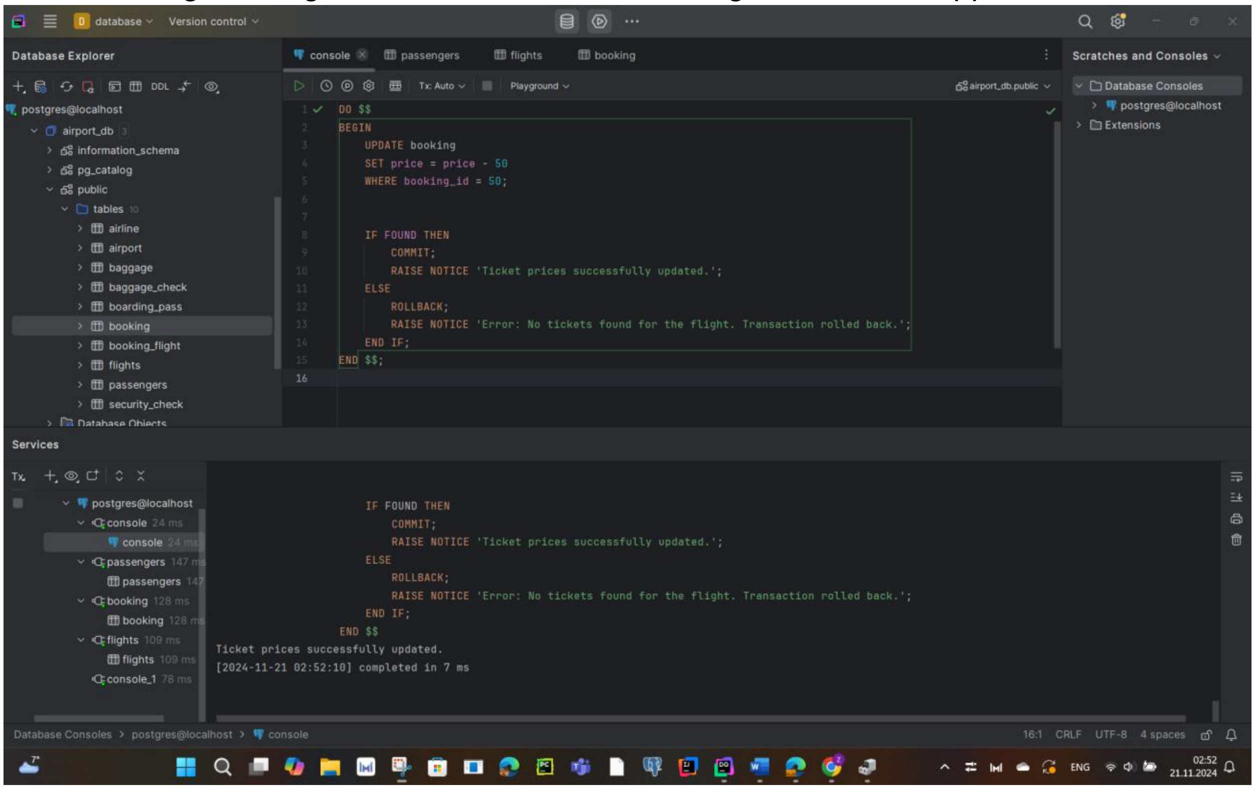
2. Rescheduling a flight. You need to reschedule a flight. Verify the 'flights' table reflects the new departure time. Simulate an error to test rollback (for example, invalid flight_id).



```sql
DO $$
BEGIN

    UPDATE Flights
    SET actual_departure = '2024-11-10'
    WHERE flight_id = 1;

    IF FOUND THEN
        COMMIT;
        RAISE NOTICE 'Flight successfully rescheduled.';
    ELSE
        ROLLBACK;
        RAISE NOTICE 'Error: Flight ID not found. Transaction rolled back.';
    END IF;
END $$;
select * from flights where flight_id = 1
```

```
    IF FOUND THEN
        COMMIT;
        RAISE NOTICE 'Flight successfully rescheduled.';
    ELSE
        ROLLBACK;
        RAISE NOTICE 'Error: Flight ID not found. Transaction rolled back.';
    END IF;
END $$
Flight successfully rescheduled.
[2024-11-21 02:49:24] completed in 3 ms
```

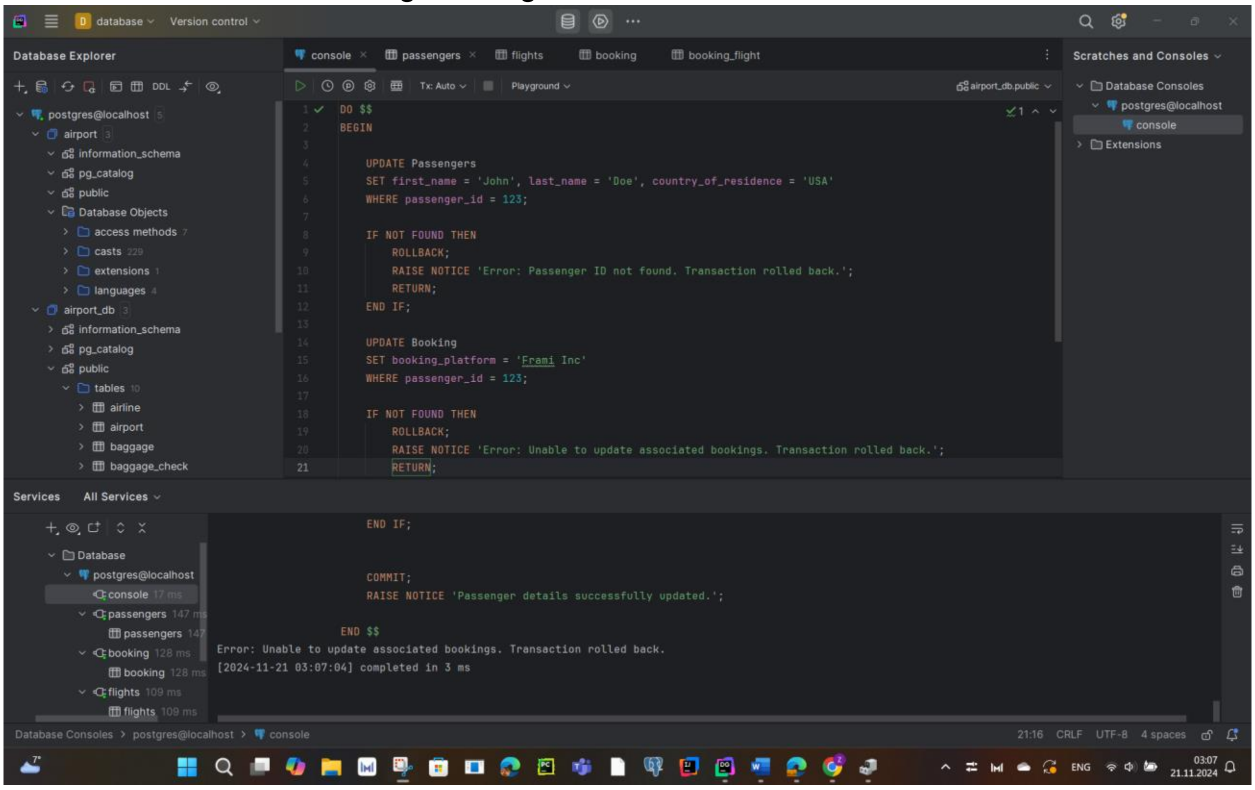
3. Updating ticket prices. You need to decrease the ticket price for a specific flight for all existing bookings. If an error occurs, no changes should be applied.



```
DO $$
BEGIN
    UPDATE booking
    SET price = price - 50
    WHERE booking_id = 50;


    IF FOUND THEN
        COMMIT;
        RAISE NOTICE 'Ticket prices successfully updated.';
    ELSE
        ROLLBACK;
        RAISE NOTICE 'Error: No tickets found for the flight. Transaction rolled back.';
    END IF;
END $$;
```

```
        IF FOUND THEN
            COMMIT;
            RAISE NOTICE 'Ticket prices successfully updated.';
        ELSE
            ROLLBACK;
            RAISE NOTICE 'Error: No tickets found for the flight. Transaction rolled back.';
        END IF;
    END $$
Ticket prices successfully updated.
[2024-11-21 02:52:10] completed in 7 ms
```

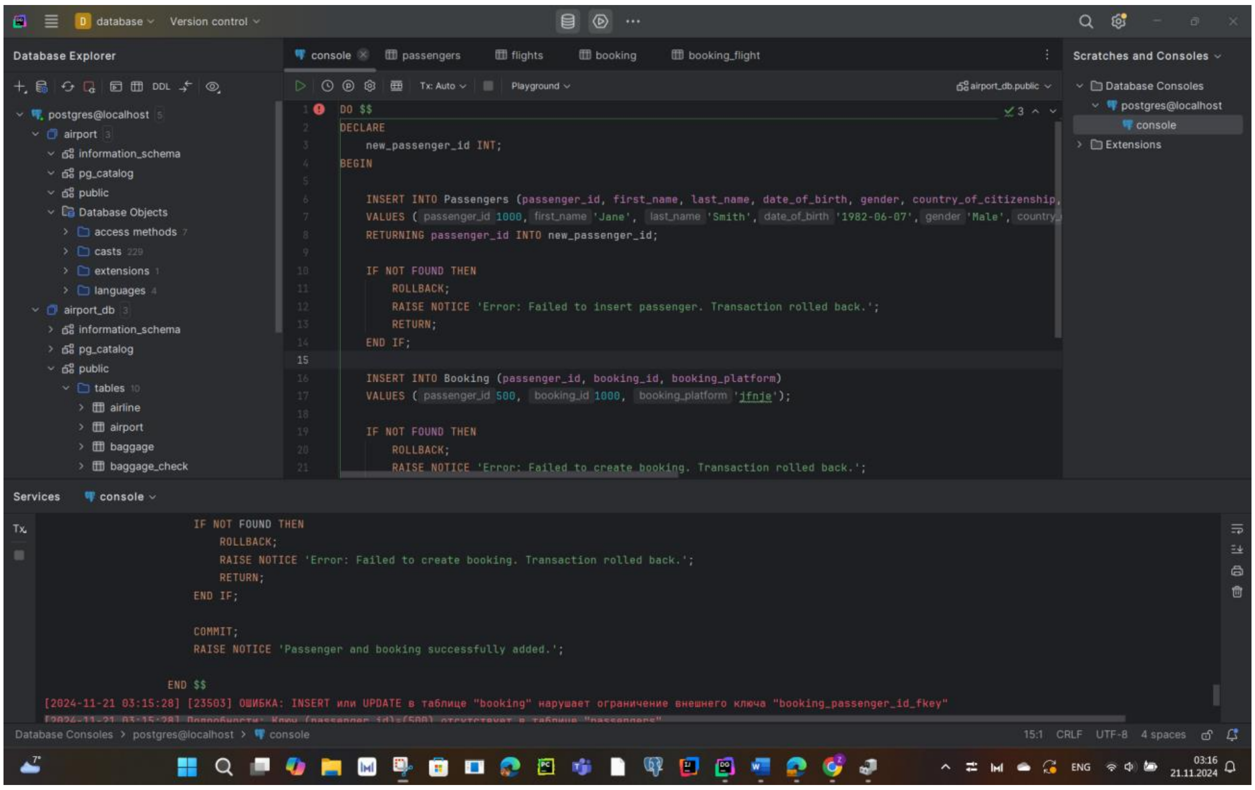
4. A passenger updates their details. Ensure the update is reflected across all associated records, including bookings.



```
DO $$
BEGIN

    UPDATE Passengers
    SET first_name = 'John', last_name = 'Doe', country_of_residence = 'USA'
    WHERE passenger_id = 123;

    IF NOT FOUND THEN
        ROLLBACK;
        RAISE NOTICE 'Error: Passenger ID not found. Transaction rolled back.';
        RETURN;
    END IF;

    UPDATE Booking
    SET booking_platform = 'Frami Inc'
    WHERE passenger_id = 123;

    IF NOT FOUND THEN
        ROLLBACK;
        RAISE NOTICE 'Error: Unable to update associated bookings. Transaction rolled back.';
        RETURN;
```

```
        END IF;


        COMMIT;
        RAISE NOTICE 'Passenger details successfully updated.';

    END $$
Error: Unable to update associated bookings. Transaction rolled back.
[2024-11-21 03:07:04] completed in 3 ms
```

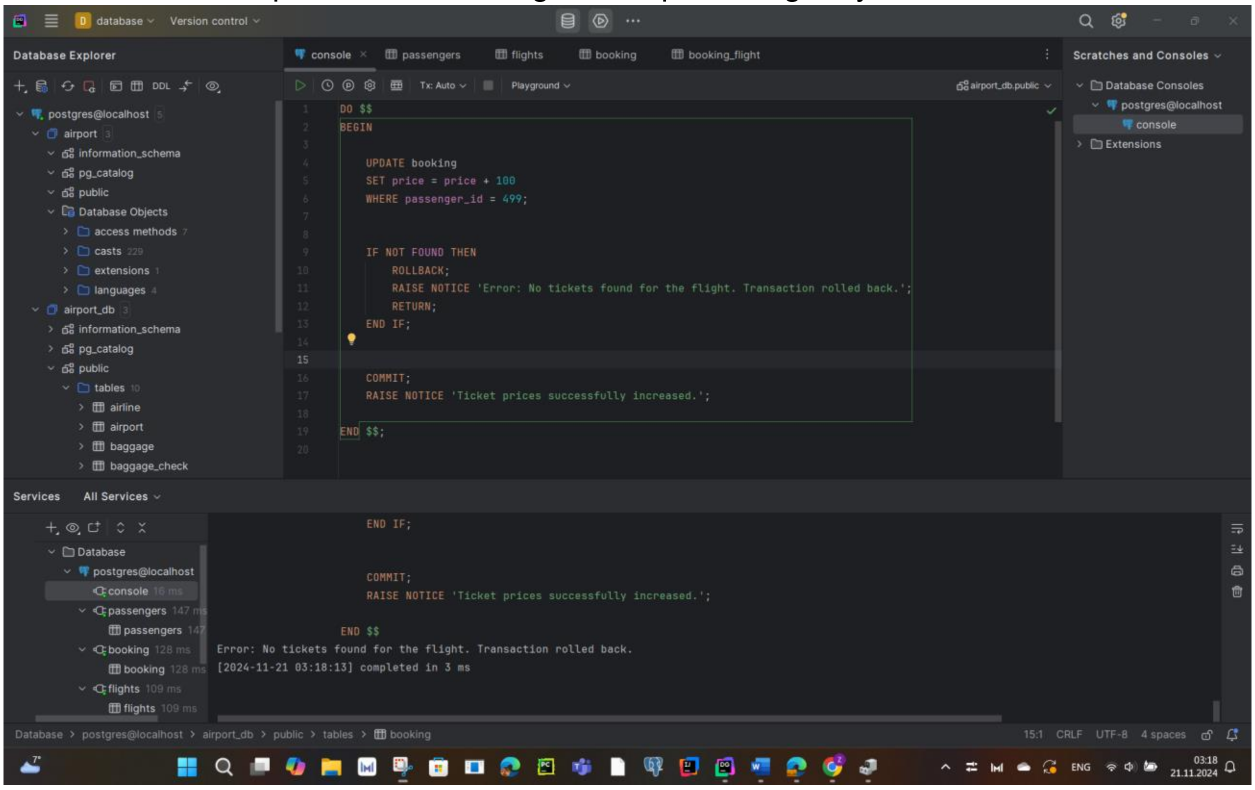
5. A new passenger is registered, and a booking is created. Ensure the new passenger is added and the booking succeeds.



```
DO $$
DECLARE
    new_passenger_id INT;
BEGIN

    INSERT INTO Passengers (passenger_id, first_name, last_name, date_of_birth, gender, country_of_citizenship,
    VALUES ( 1000, 'Jane', 'Smith', '1982-06-07', 'Male', 
    RETURNING passenger_id INTO new_passenger_id;

    IF NOT FOUND THEN
        ROLLBACK;
        RAISE NOTICE 'Error: Failed to insert passenger. Transaction rolled back.';
        RETURN;
    END IF;

    INSERT INTO Booking (passenger_id, booking_id, booking_platform)
    VALUES ( 500, 1000, 'jfnje');

    IF NOT FOUND THEN
        ROLLBACK;
        RAISE NOTICE 'Error: Failed to create booking. Transaction rolled back.';
```

```
                    IF NOT FOUND THEN
                        ROLLBACK;
                        RAISE NOTICE 'Error: Failed to create booking. Transaction rolled back.';
                        RETURN;
                    END IF;

                    COMMIT;
                    RAISE NOTICE 'Passenger and booking successfully added.';

                END $$
[2024-11-21 03:15:28] [23503] ОШИБКА: INSERT или UPDATE в таблице "booking" нарушает ограничение внешнего ключа "booking_passenger_id_fkey"
```

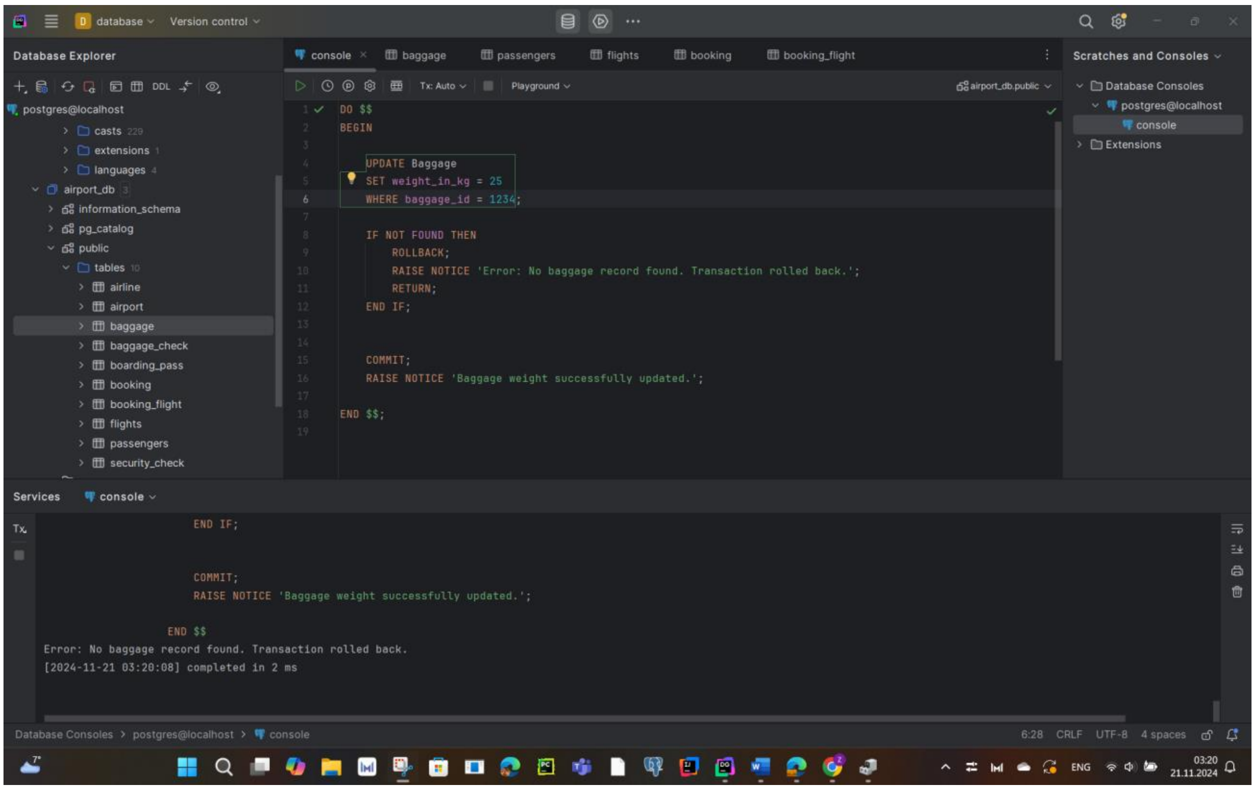
6. Increase the ticket price for all bookings on a specific flight by a fixed amount.



```
DO $$
BEGIN

    UPDATE booking
    SET price = price + 100
    WHERE passenger_id = 499;


    IF NOT FOUND THEN
        ROLLBACK;
        RAISE NOTICE 'Error: No tickets found for the flight. Transaction rolled back.';
        RETURN;
    END IF;


    COMMIT;
    RAISE NOTICE 'Ticket prices successfully increased.';

END $$;
```

```
                    END IF;


                    COMMIT;
                    RAISE NOTICE 'Ticket prices successfully increased.';

                END $$
Error: No tickets found for the flight. Transaction rolled back.
[2024-11-21 03:18:13] completed in 3 ms
```

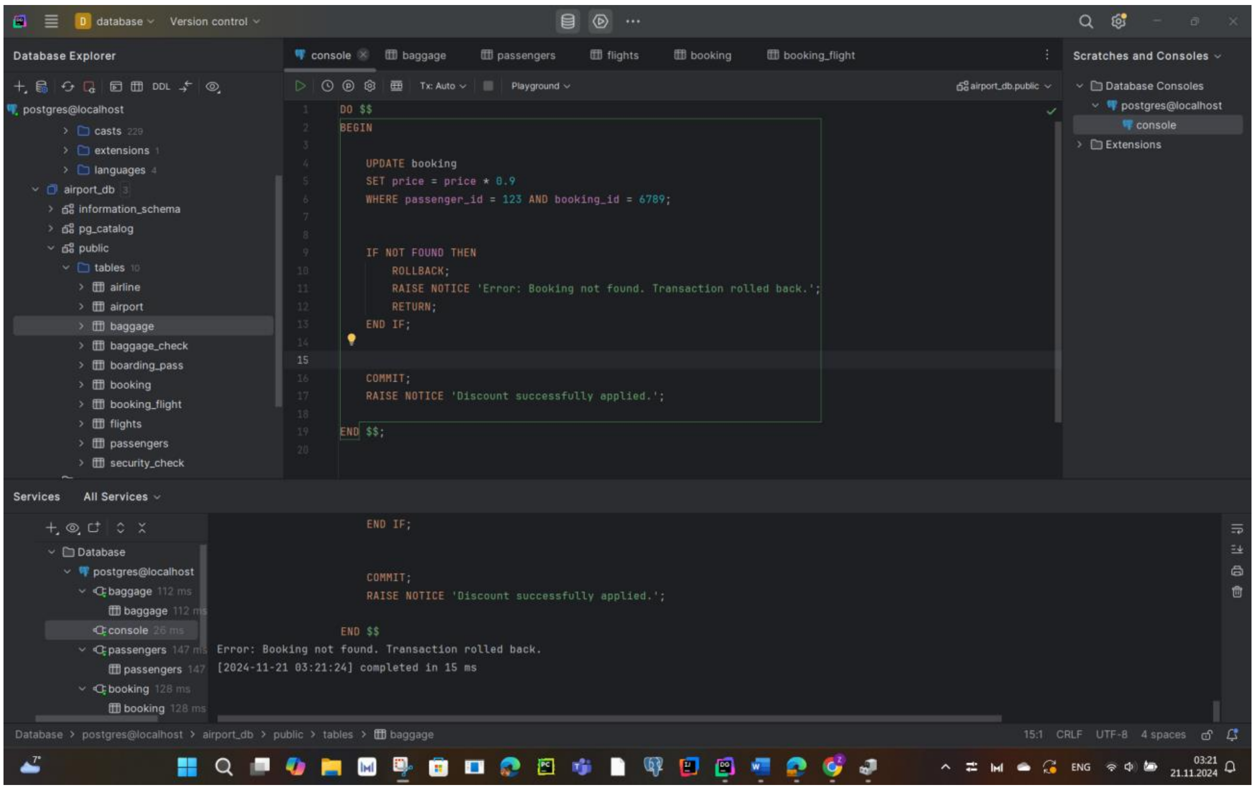
7. Update a baggage weight. A passenger updates the declared weight of their baggage. Ensure that the change is correctly reflected in the database.



```
DO $$
BEGIN

    UPDATE Baggage
    SET weight_in_kg = 25
    WHERE baggage_id = 1234;

    IF NOT FOUND THEN
        ROLLBACK;
        RAISE NOTICE 'Error: No baggage record found. Transaction rolled back.';
        RETURN;
    END IF;


    COMMIT;
    RAISE NOTICE 'Baggage weight successfully updated.';

END $$;
```

```
                    END IF;


                    COMMIT;
                    RAISE NOTICE 'Baggage weight successfully updated.';

                END $$
Error: No baggage record found. Transaction rolled back.
[2024-11-21 03:20:08] completed in 2 ms
```

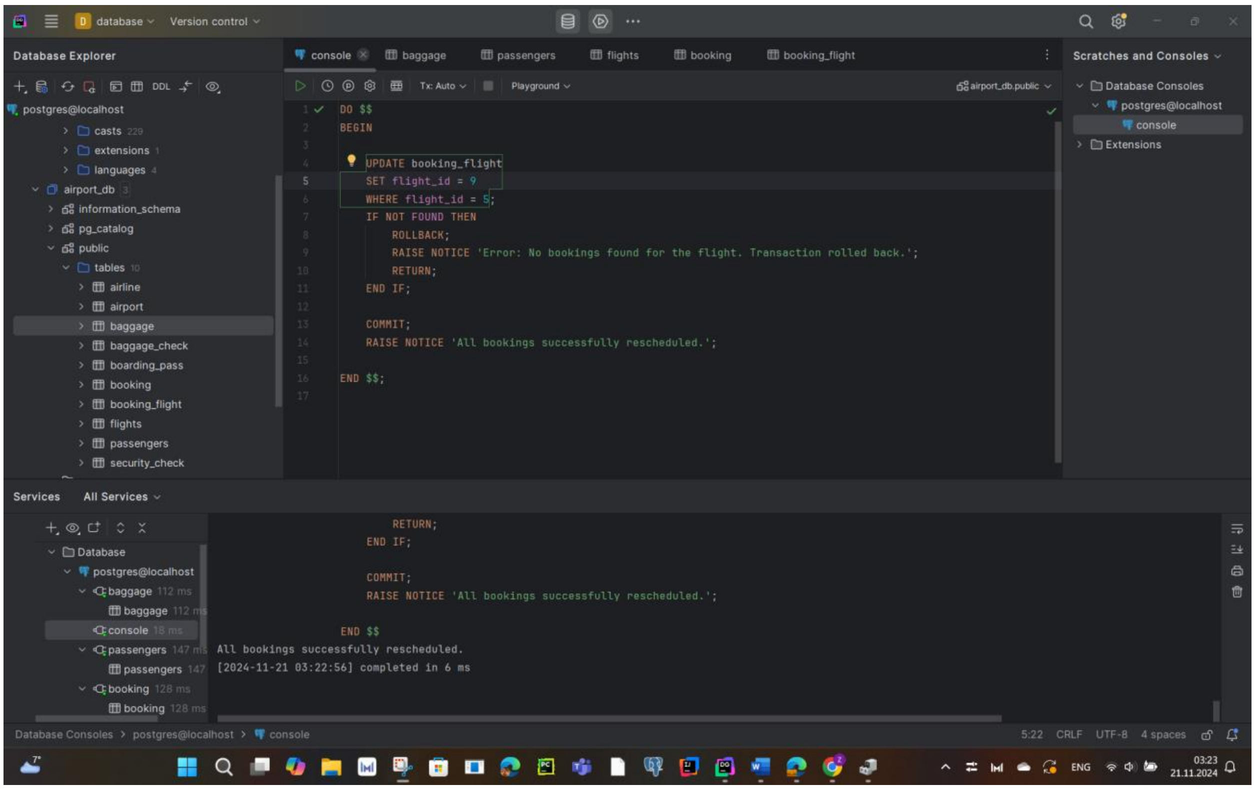
8. Apply a discount to a booking for a specific passenger. If any error occurs, roll back.



```
DO $$
BEGIN

    UPDATE booking
    SET price = price * 0.9
    WHERE passenger_id = 123 AND booking_id = 6789;


    IF NOT FOUND THEN
        ROLLBACK;
        RAISE NOTICE 'Error: Booking not found. Transaction rolled back.';
        RETURN;
    END IF;


    COMMIT;
    RAISE NOTICE 'Discount successfully applied.';

END $$;
```

```
                    END IF;


                    COMMIT;
                    RAISE NOTICE 'Discount successfully applied.';

                END $$
Error: Booking not found. Transaction rolled back.
[2024-11-21 03:21:24] completed in 15 ms
```

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



```sql
DO $$
BEGIN

    UPDATE booking_flight
    SET flight_id = 9
    WHERE flight_id = 5;
    IF NOT FOUND THEN
        ROLLBACK;
        RAISE NOTICE 'Error: No bookings found for the flight. Transaction rolled back.';
        RETURN;
    END IF;

    COMMIT;
    RAISE NOTICE 'All bookings successfully rescheduled.';

END $$;
```

```
                RETURN;
            END IF;

            COMMIT;
            RAISE NOTICE 'All bookings successfully rescheduled.';

        END $$
All bookings successfully rescheduled.
[2024-11-21 03:22:56] completed in 6 ms
```