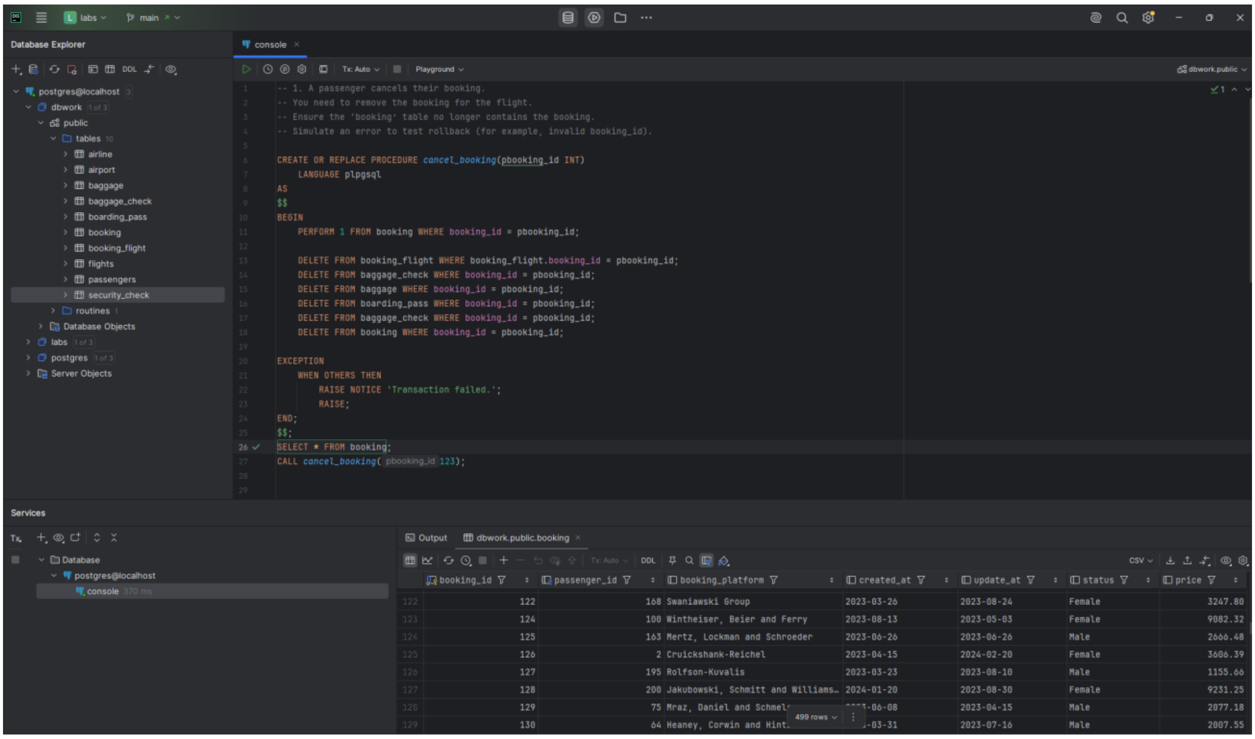
1)



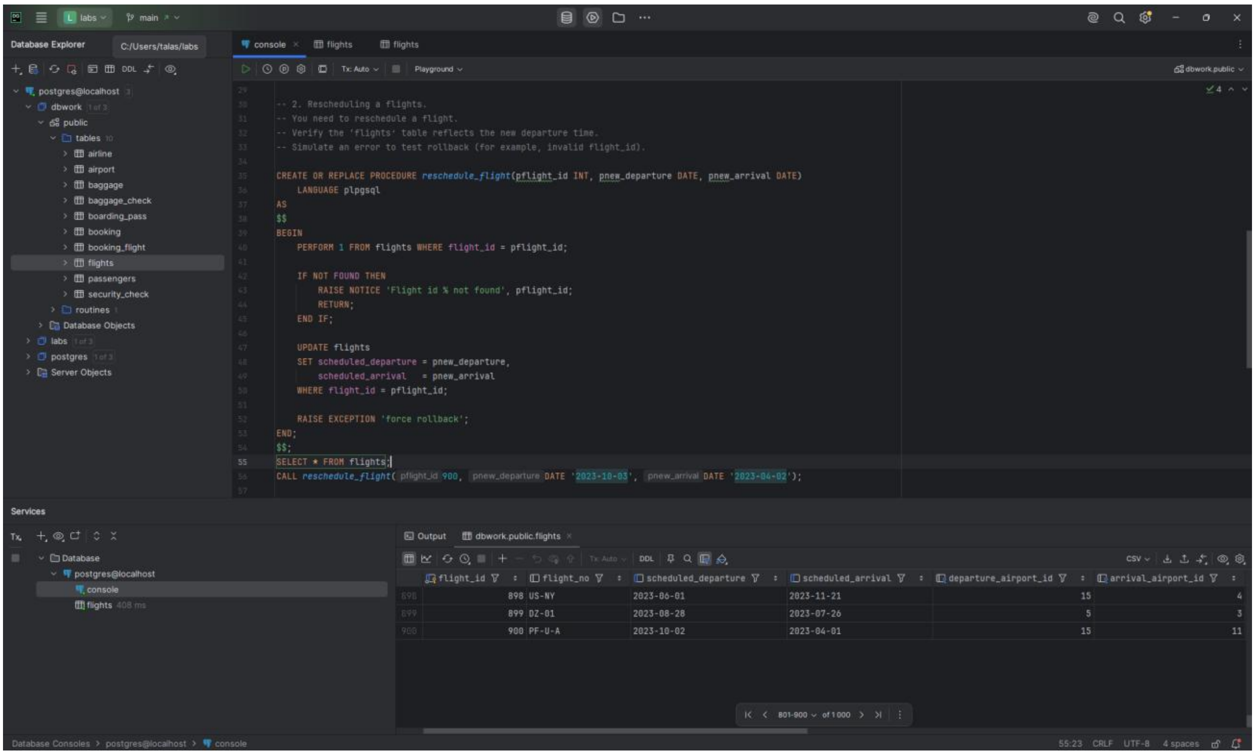
```
-- 1. A passenger cancels their booking.
-- You need to remove the booking for the flight.
-- Ensure the 'booking' table no longer contains the booking.
-- Simulate an error to test rollback (for example, invalid booking_id).

CREATE OR REPLACE PROCEDURE cancel_booking(pbooking_id INT)
    LANGUAGE plpgsql
AS
$$
BEGIN
    PERFORM 1 FROM booking WHERE booking_id = pbooking_id;

    DELETE FROM booking_flight WHERE booking_flight.booking_id = pbooking_id;
    DELETE FROM baggage_check WHERE booking_id = pbooking_id;
    DELETE FROM baggage WHERE booking_id = pbooking_id;
    DELETE FROM boarding_pass WHERE booking_id = pbooking_id;
    DELETE FROM baggage_check WHERE booking_id = pbooking_id;
    DELETE FROM booking WHERE booking_id = pbooking_id;

EXCEPTION
    WHEN OTHERS THEN
        RAISE NOTICE 'Transaction failed.';
        RAISE;
END;
$$;
SELECT * FROM booking;
CALL cancel_booking( pbooking_id 123);
```

dbwork.public.booking

	booking_id	passenger_id	booking_platform	created_at	update_at	status	price
122	122	168	Swaniawski Group	2023-03-26	2023-08-24	Female	3247.80
123	124	100	Wintheiser, Beier and Ferry	2023-08-13	2023-05-03	Female	9082.32
124	125	163	Mertz, Lockman and Schroeder	2023-06-26	2023-06-26	Male	2666.48
125	126	2	Cruickshank-Reichel	2023-04-15	2024-02-20	Female	3606.39
126	127	195	Rolfson-Kuvalis	2023-03-23	2023-08-10	Male	1155.66
127	128	200	Jakubowski, Schmitt and Williams…	2024-01-20	2023-08-30	Female	9231.25
128	129	75	Mraz, Daniel and Schmel	-06-08	2023-04-15	Male	2077.18
129	130	64	Heaney, Corwin and Hint.	-03-31	2023-07-16	Male	2007.55

499 rows

2)

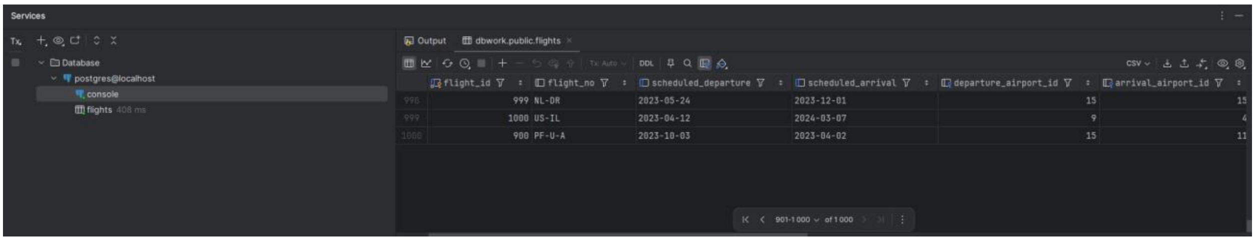


```
-- 2. Rescheduling a flights.
-- You need to reschedule a flight.
-- Verify the 'flights' table reflects the new departure time.
-- Simulate an error to test rollback (for example, invalid flight_id).

CREATE OR REPLACE PROCEDURE reschedule_flight(pflight_id INT, pnew_departure DATE, pnew_arrival DATE)
    LANGUAGE plpgsql
AS
$$
BEGIN
    PERFORM 1 FROM flights WHERE flight_id = pflight_id;

    IF NOT FOUND THEN
        RAISE NOTICE 'Flight id % not found', pflight_id;
        RETURN;
    END IF;

    UPDATE flights
    SET scheduled_departure = pnew_departure,
        scheduled_arrival   = pnew_arrival
    WHERE flight_id = pflight_id;

    RAISE EXCEPTION 'force rollback';
END;
$$;
SELECT * FROM flights;
CALL reschedule_flight( pflight_id 900, pnew_departure DATE '2023-10-03', pnew_arrival DATE '2023-04-02');
```

dbwork.public.flights

	flight_id	flight_no	scheduled_departure	scheduled_arrival	departure_airport_id	arrival_airport_id
898	898	US-NY	2023-06-01	2023-11-21	15	4
899	899	DZ-01	2023-08-28	2023-07-26	5	3
900	900	PF-U-A	2023-10-02	2023-04-01	15	11

801-900 of 1 000

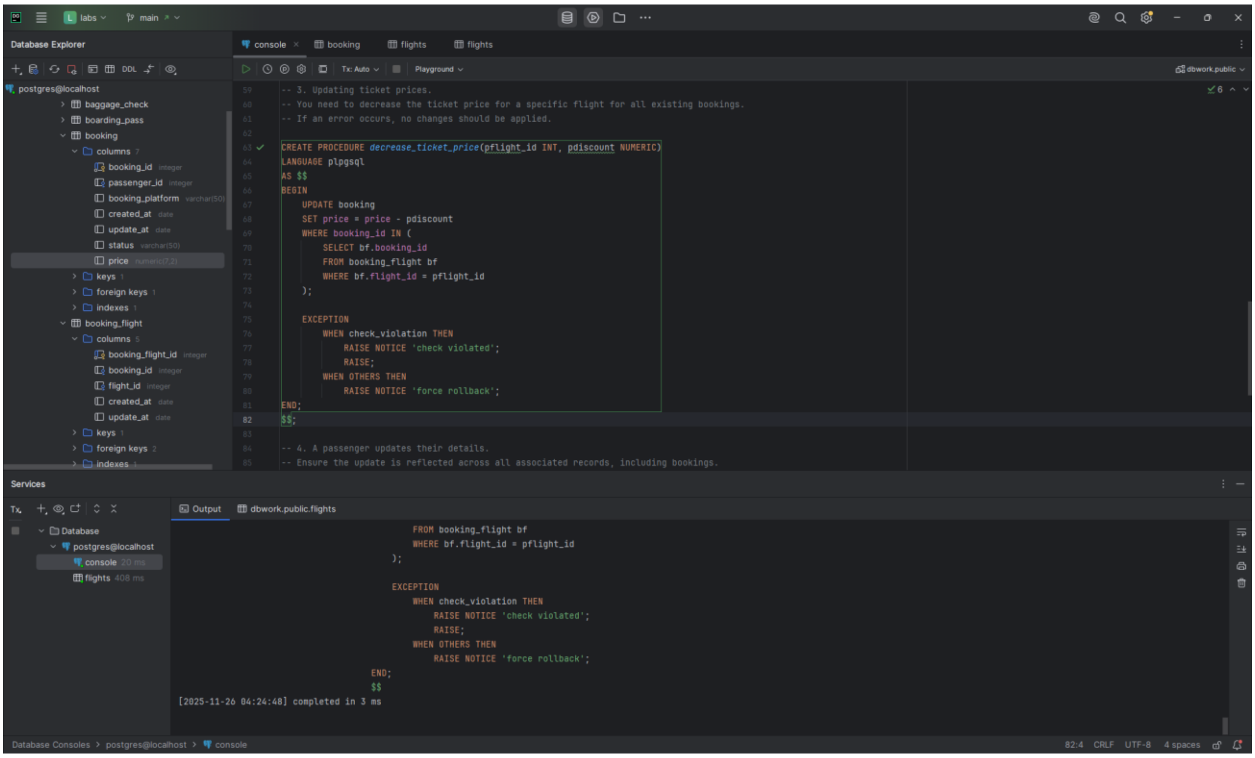


dbwork.public.flights

	flight_id	flight_no	scheduled_departure	scheduled_arrival	departure_airport_id	arrival_airport_id
998	999	NL-DR	2023-05-24	2023-12-01	15	15
999	1000	US-IL	2023-04-12	2024-03-07	9	4
1000	900	PF-U-A	2023-10-03	2023-04-02	15	11

901-1 000 of 1 000

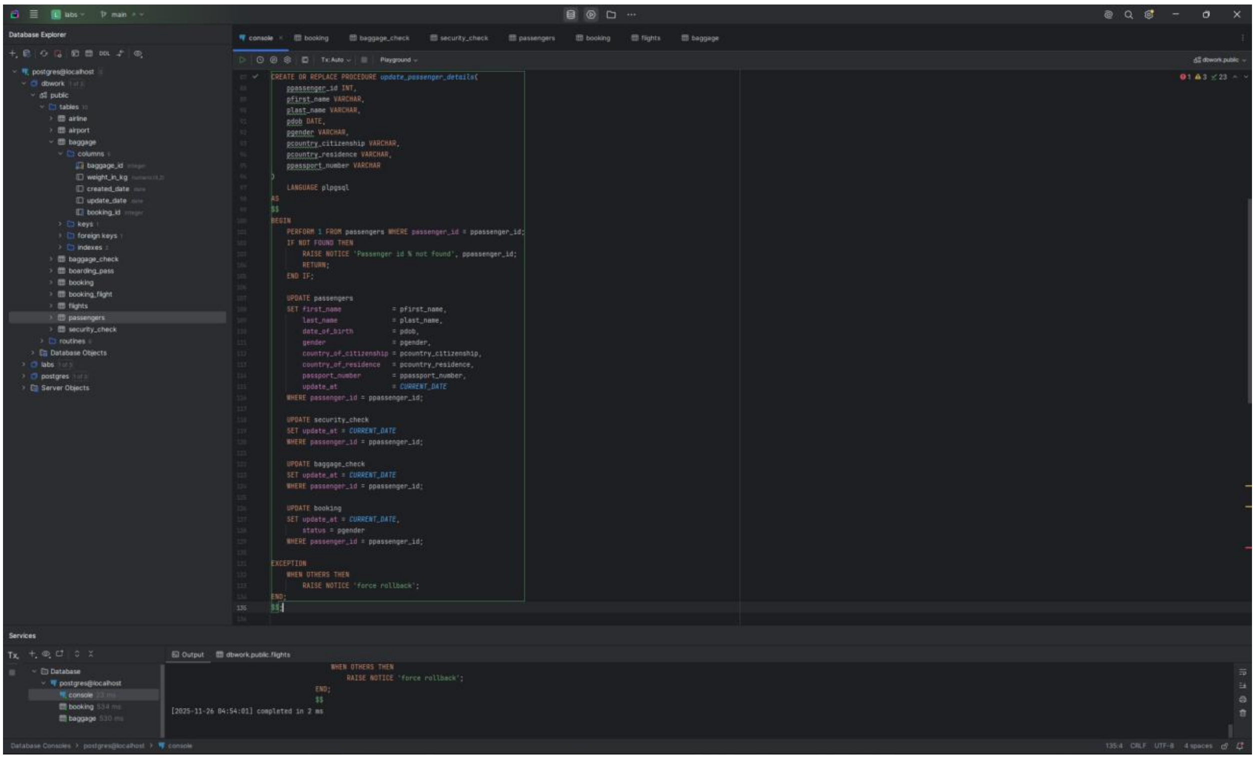
3)



```
-- 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.

CREATE PROCEDURE decrease_ticket_price(pflight_id INT, pdiscount NUMERIC)
LANGUAGE plpgsql
AS $$
BEGIN
    UPDATE booking
    SET price = price - pdiscount
    WHERE booking_id IN (
        SELECT bf.booking_id
        FROM booking_flight bf
        WHERE bf.flight_id = pflight_id
    );

    EXCEPTION
        WHEN check_violation THEN
            RAISE NOTICE 'check violated';
            RAISE;
        WHEN OTHERS THEN
            RAISE NOTICE 'force rollback';
END;
$$;

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

```
                        FROM booking_flight bf
                        WHERE bf.flight_id = pflight_id
                    );

                    EXCEPTION
                        WHEN check_violation THEN
                            RAISE NOTICE 'check violated';
                            RAISE;
                        WHEN OTHERS THEN
                            RAISE NOTICE 'force rollback';
                END;
                $$
[2025-11-26 04:24:48] completed in 3 ms
```

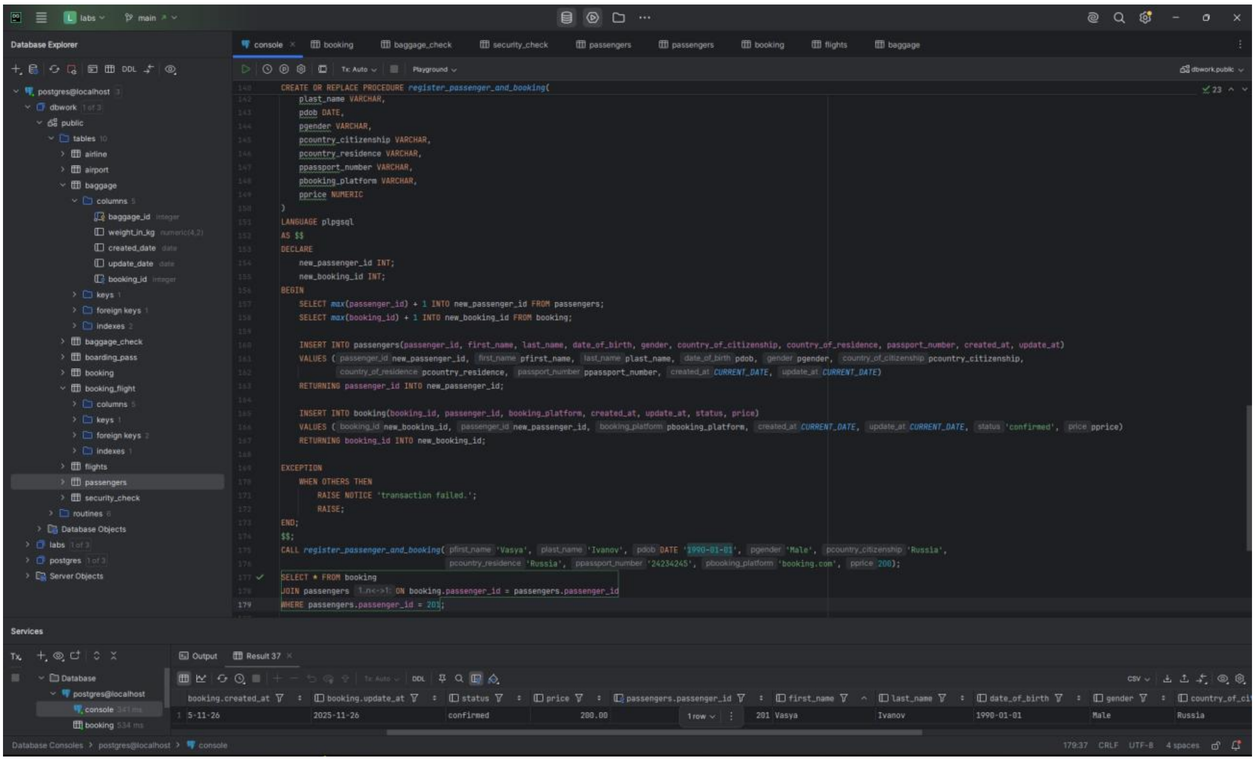
4)



```
CREATE OR REPLACE PROCEDURE update_passenger_details(
    ppassenger_id INT,
    pfirst_name VARCHAR,
    plast_name VARCHAR,
    pdob DATE,
    pgender VARCHAR,
    pcountry_citizenship VARCHAR,
    pcountry_residence VARCHAR,
    ppassport_number VARCHAR
)
    LANGUAGE plpgsql
AS
$$
BEGIN
    PERFORM 1 FROM passengers WHERE passenger_id = ppassenger_id;
    IF NOT FOUND THEN
        RAISE NOTICE 'Passenger id % not found', ppassenger_id;
        RETURN;
    END IF;

    UPDATE passengers
    SET first_name             = pfirst_name,
        last_name              = plast_name,
        date_of_birth          = pdob,
        gender                 = pgender,
        country_of_citizenship = pcountry_citizenship,
        country_of_residence   = pcountry_residence,
        passport_number        = ppassport_number,
        update_at              = CURRENT_DATE
    WHERE passenger_id = ppassenger_id;

    UPDATE security_check
    SET update_at = CURRENT_DATE
    WHERE passenger_id = ppassenger_id;

    UPDATE baggage_check
    SET update_at = CURRENT_DATE
    WHERE passenger_id = ppassenger_id;

    UPDATE booking
    SET update_at = CURRENT_DATE,
        status = pgender
    WHERE passenger_id = ppassenger_id;

EXCEPTION
    WHEN OTHERS THEN
        RAISE NOTICE 'force rollback';
END;
$$;
```

```
                    WHEN OTHERS THEN
                        RAISE NOTICE 'force rollback';
                END;
                $$
[2025-11-26 04:54:01] completed in 2 ms
```

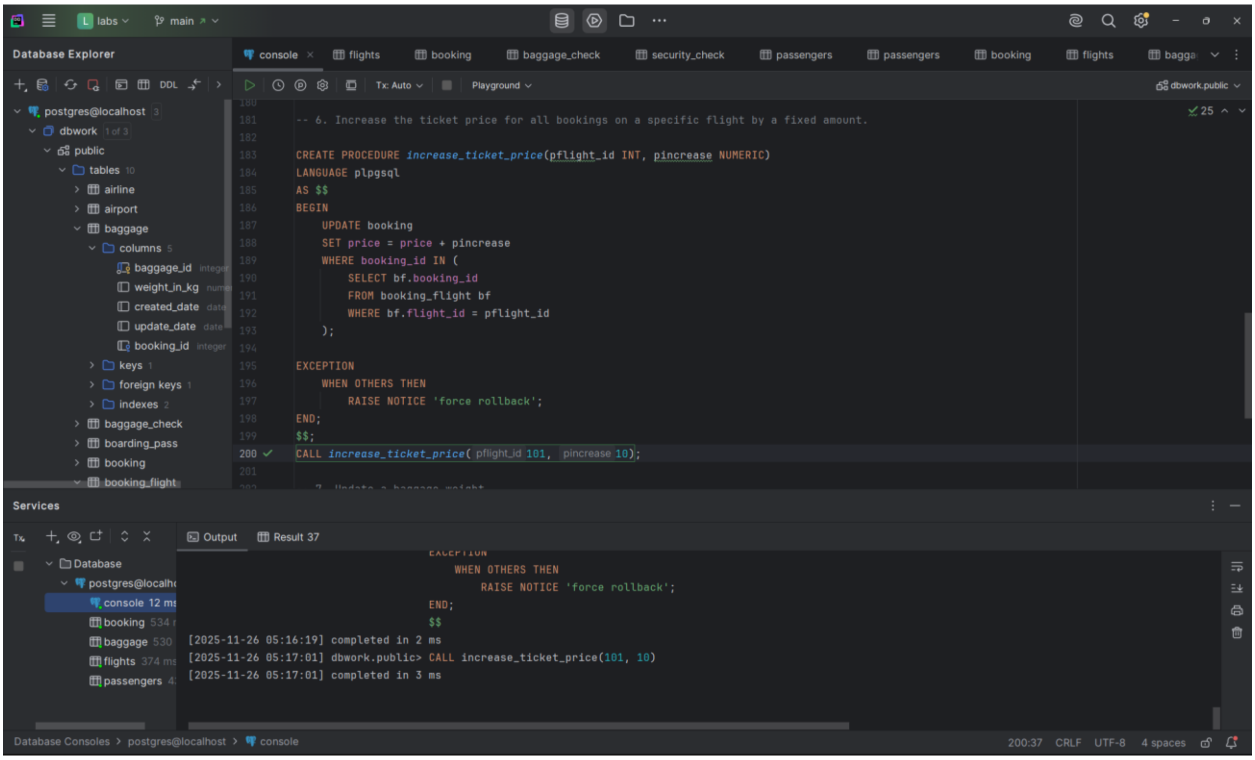
5)



```sql
CREATE OR REPLACE PROCEDURE register_passenger_and_booking(
    plast_name VARCHAR,
    pdob DATE,
    pgender VARCHAR,
    pcountry_citizenship VARCHAR,
    pcountry_residence VARCHAR,
    ppassport_number VARCHAR,
    pbooking_platform VARCHAR,
    pprice NUMERIC
)
LANGUAGE plpgsql
AS $$
DECLARE
    new_passenger_id INT;
    new_booking_id INT;
BEGIN
    SELECT max(passenger_id) + 1 INTO new_passenger_id FROM passengers;
    SELECT max(booking_id) + 1 INTO new_booking_id FROM booking;

    INSERT INTO passengers(passenger_id, first_name, last_name, date_of_birth, gender, country_of_citizenship, country_of_residence, passport_number, created_at, update_at)
    VALUES ( new_passenger_id, pfirst_name, plast_name, pdob, pgender, pcountry_citizenship,
             pcountry_residence, ppassport_number, CURRENT_DATE, CURRENT_DATE)
    RETURNING passenger_id INTO new_passenger_id;

    INSERT INTO booking(booking_id, passenger_id, booking_platform, created_at, update_at, status, price)
    VALUES ( new_booking_id, new_passenger_id, pbooking_platform, CURRENT_DATE, CURRENT_DATE, 'confirmed', pprice)
    RETURNING booking_id INTO new_booking_id;

EXCEPTION
    WHEN OTHERS THEN
        RAISE NOTICE 'transaction failed.';
        RAISE;
END;
$$;
CALL register_passenger_and_booking( 'Vasya', 'Ivanov', DATE '1990-01-01', 'Male', 'Russia',
                                     'Russia', '24234245', 'booking.com', 200);
SELECT * FROM booking
JOIN passengers ON booking.passenger_id = passengers.passenger_id
WHERE passengers.passenger_id = 201;
```

	booking.created_at	booking.update_at	status	price	passengers.passenger_id	first_name	last_name	date_of_birth	gender	country_of_ci
1	5-11-26	2025-11-26	confirmed	200.00	201	Vasya	Ivanov	1990-01-01	Male	Russia

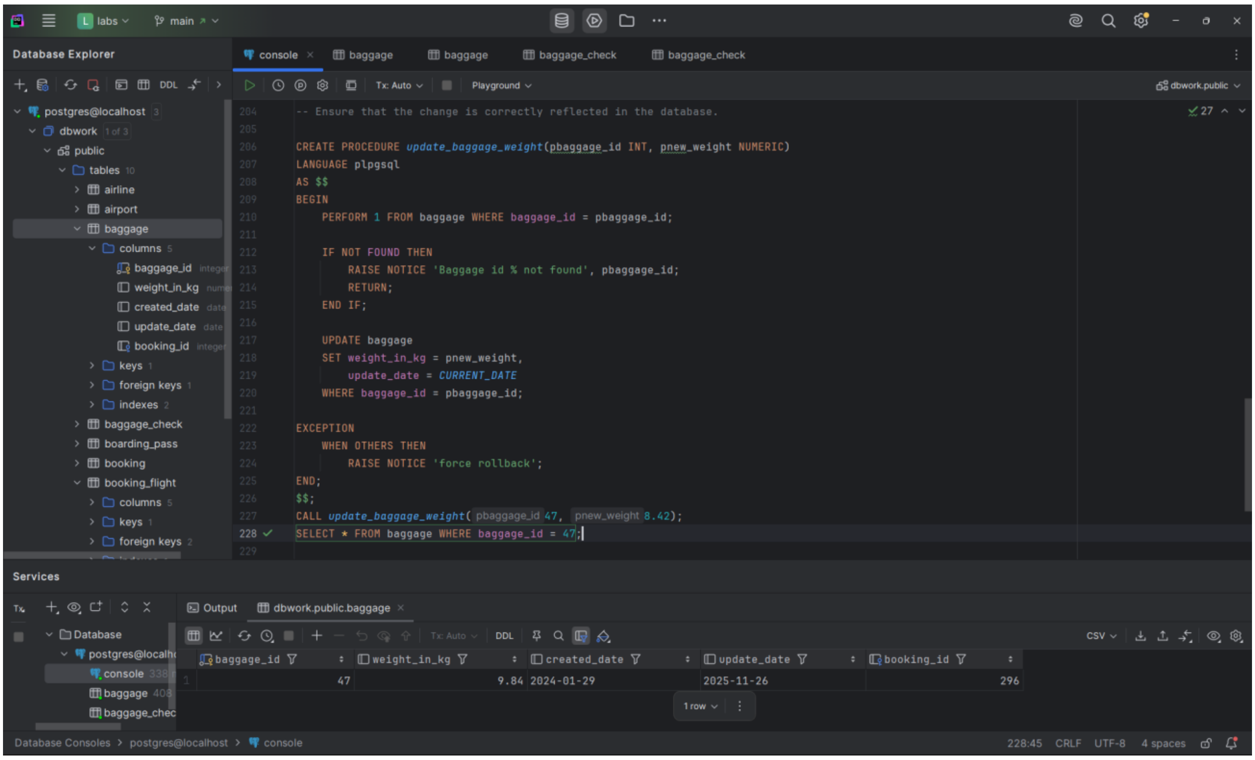
6)



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

CREATE PROCEDURE increase_ticket_price(pflight_id INT, pincrease NUMERIC)
LANGUAGE plpgsql
AS $$
BEGIN
    UPDATE booking
    SET price = price + pincrease
    WHERE booking_id IN (
        SELECT bf.booking_id
        FROM booking_flight bf
        WHERE bf.flight_id = pflight_id
    );

EXCEPTION
    WHEN OTHERS THEN
        RAISE NOTICE 'force rollback';
END;
$$;
CALL increase_ticket_price( 101, 10);
```

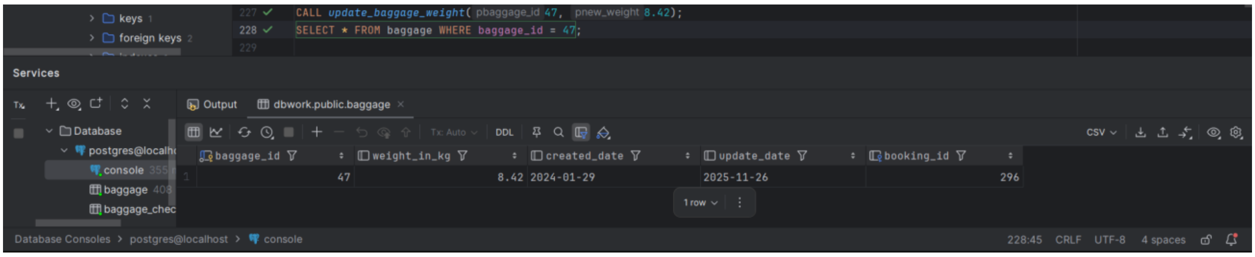
```
            EXCEPTION
                WHEN OTHERS THEN
                    RAISE NOTICE 'force rollback';
            END;
            $$
[2025-11-26 05:16:19] completed in 2 ms
[2025-11-26 05:17:01] dbwork.public> CALL increase_ticket_price(101, 10)
[2025-11-26 05:17:01] completed in 3 ms
```

7)



```sql
-- Ensure that the change is correctly reflected in the database.

CREATE PROCEDURE update_baggage_weight(pbaggage_id INT, pnew_weight NUMERIC)
LANGUAGE plpgsql
AS $$
BEGIN
    PERFORM 1 FROM baggage WHERE baggage_id = pbaggage_id;

    IF NOT FOUND THEN
        RAISE NOTICE 'Baggage id % not found', pbaggage_id;
        RETURN;
    END IF;

    UPDATE baggage
    SET weight_in_kg = pnew_weight,
        update_date = CURRENT_DATE
    WHERE baggage_id = pbaggage_id;

EXCEPTION
    WHEN OTHERS THEN
        RAISE NOTICE 'force rollback';
END;
$$;
CALL update_baggage_weight(47, 8.42);
SELECT * FROM baggage WHERE baggage_id = 47;
```

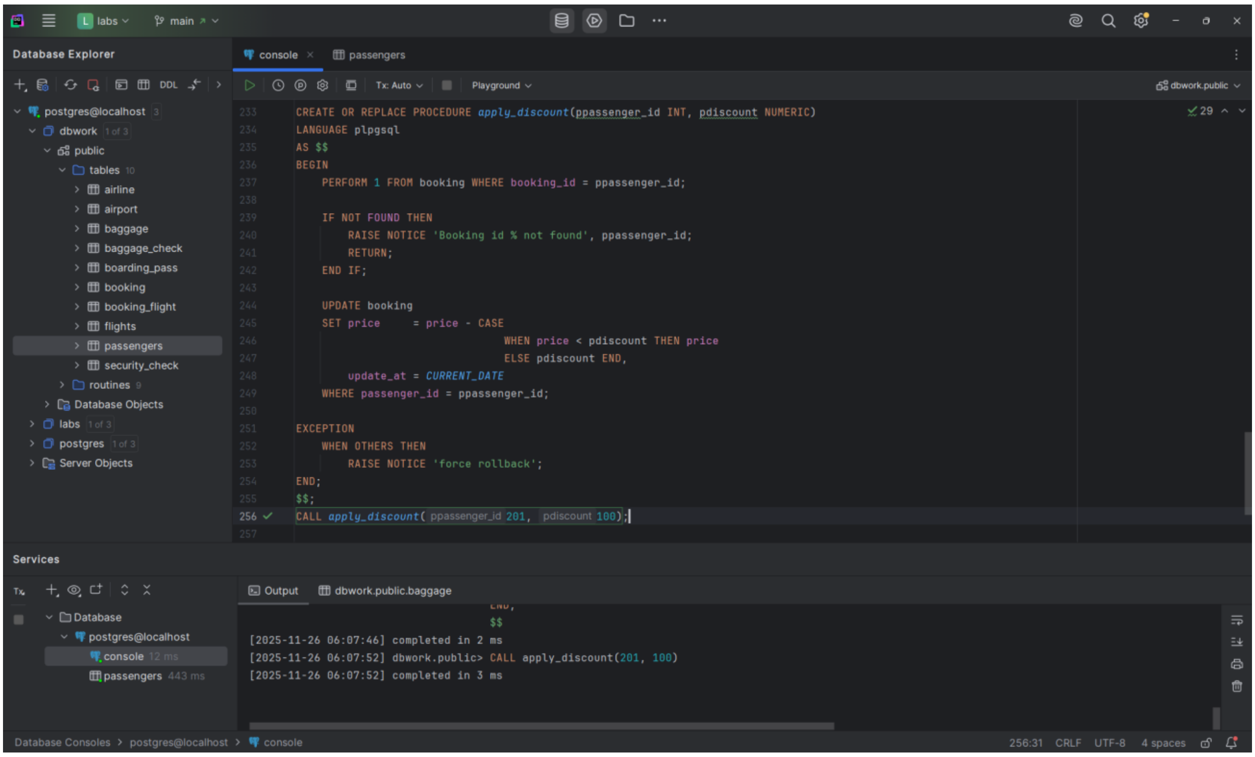
	baggage_id	weight_in_kg	created_date	update_date	booking_id
1	47	9.84	2024-01-29	2025-11-26	296



```sql
CALL update_baggage_weight(47, 8.42);
SELECT * FROM baggage WHERE baggage_id = 47;
```

	baggage_id	weight_in_kg	created_date	update_date	booking_id
1	47	8.42	2024-01-29	2025-11-26	296

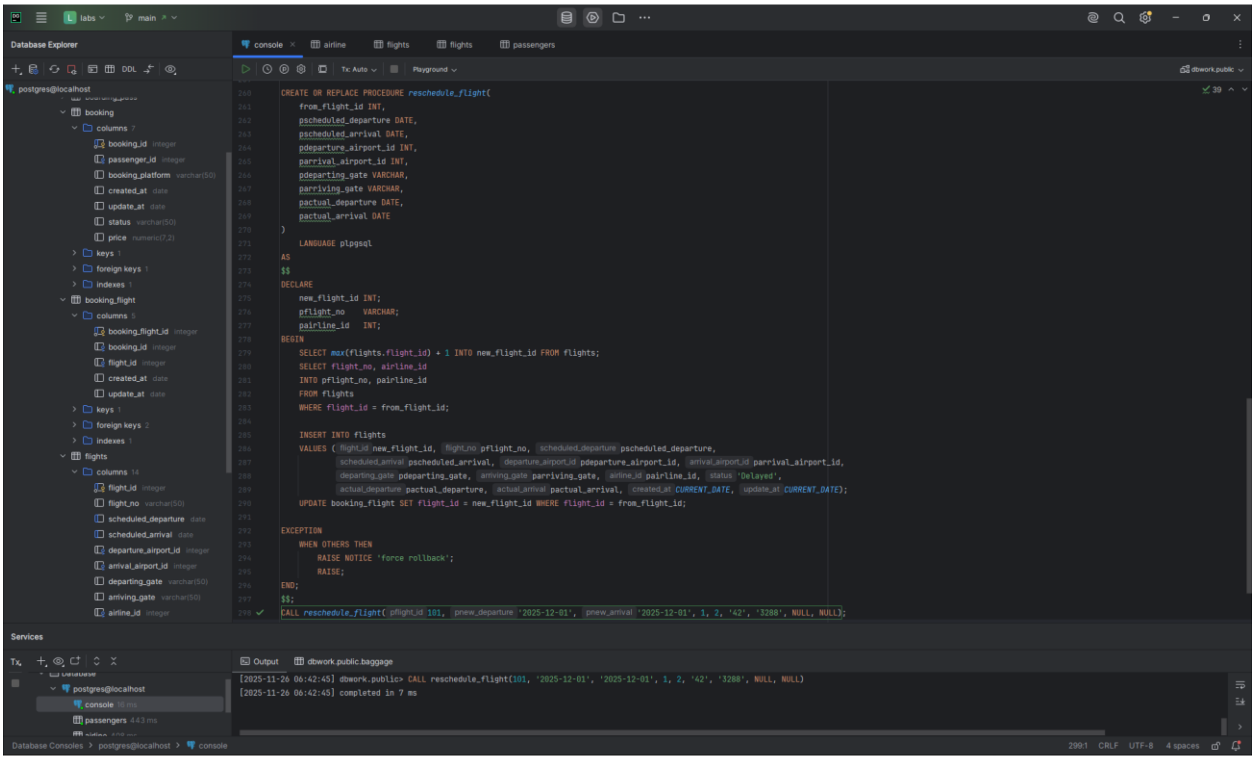
8)



```sql
CREATE OR REPLACE PROCEDURE apply_discount(ppassenger_id INT, pdiscount NUMERIC)
LANGUAGE plpgsql
AS $$
BEGIN
    PERFORM 1 FROM booking WHERE booking_id = ppassenger_id;

    IF NOT FOUND THEN
        RAISE NOTICE 'Booking id % not found', ppassenger_id;
        RETURN;
    END IF;

    UPDATE booking
    SET price     = price - CASE
                               WHEN price < pdiscount THEN price
                               ELSE pdiscount END,
        update_at = CURRENT_DATE
    WHERE passenger_id = ppassenger_id;

EXCEPTION
    WHEN OTHERS THEN
        RAISE NOTICE 'force rollback';
END;
$$;
CALL apply_discount(201, 100);
```

```
                              END,
                              $$
[2025-11-26 06:07:46] completed in 2 ms
[2025-11-26 06:07:52] dbwork.public> CALL apply_discount(201, 100)
[2025-11-26 06:07:52] completed in 3 ms
```

9)



```sql
CREATE OR REPLACE PROCEDURE reschedule_flight(
    from_flight_id INT,
    pscheduled_departure DATE,
    pscheduled_arrival DATE,
    pdeparture_airport_id INT,
    parrival_airport_id INT,
    pdeparting_gate VARCHAR,
    parriving_gate VARCHAR,
    pactual_departure DATE,
    pactual_arrival DATE
)
    LANGUAGE plpgsql
AS
$$
DECLARE
    new_flight_id INT;
    pflight_no    VARCHAR;
    pairline_id   INT;
BEGIN
    SELECT max(flights.flight_id) + 1 INTO new_flight_id FROM flights;
    SELECT flight_no, airline_id
    INTO pflight_no, pairline_id
    FROM flights
    WHERE flight_id = from_flight_id;

    INSERT INTO flights
    VALUES ( new_flight_id, pflight_no, pscheduled_departure,
            pscheduled_arrival, pdeparture_airport_id, parrival_airport_id,
            pdeparting_gate, parriving_gate, pairline_id, 'Delayed',
            pactual_departure, pactual_arrival, CURRENT_DATE, CURRENT_DATE);
    UPDATE booking_flight SET flight_id = new_flight_id WHERE flight_id = from_flight_id;

EXCEPTION
    WHEN OTHERS THEN
        RAISE NOTICE 'force rollback';
        RAISE;
END;
$$;
CALL reschedule_flight( 101, '2025-12-01', '2025-12-01', 1, 2, '42', '3288', NULL, NULL);
```

```
[2025-11-26 06:42:45] dbwork.public> CALL reschedule_flight(101, '2025-12-01', '2025-12-01', 1, 2, '42', '3288', NULL, NULL)
[2025-11-26 06:42:45] completed in 7 ms
```