

1)

Database Explorer

```

1 -- 1. A passenger cancels their booking.
2 -- You need to remove the booking for the flight.
3 -- Ensure the 'booking' table no longer contains the booking.
4 -- Simulate an error to test rollback (for example, invalid booking_id).
5
6 CREATE OR REPLACE PROCEDURE cancel_booking(pbooking_id INT)
7   LANGUAGE plpgsql
8 AS
9 $$
10 BEGIN
11   PERFORM 1 FROM booking WHERE booking_id = pbooking_id;
12
13   DELETE FROM booking_flight WHERE booking_flight.booking_id = pbooking_id;
14   DELETE FROM baggage WHERE booking_id = pbooking_id;
15   DELETE FROM security_check WHERE booking_id = pbooking_id;
16   DELETE FROM boarding_pass WHERE booking_id = pbooking_id;
17   DELETE FROM baggage_check WHERE booking_id = pbooking_id;
18   DELETE FROM booking WHERE booking_id = pbooking_id;
19
20 EXCEPTION
21   WHEN OTHERS THEN
22     RAISE NOTICE 'Transaction failed.';
23     RAISE;
24
25 $$/;
26
27 SELECT * FROM booking;
28 CALL cancel_booking(booking_id 123);
29

```

Services

Flight ID	Passenger ID	Booking Platform	Created At	Updated At	Status	Price
122	122	168 Swaniawski Group	2023-03-26	2023-08-24	Female	3247.88
123	124	100 Wintzler, Beier and Ferry	2023-08-13	2023-05-03	Female	9882.32
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	3686.39
126	127	195 Koltsos-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 Kraz, Daniel and Schmelz	2022-06-08	2023-04-15	Male	2077.55
129	130	64 Heaney, Corwin and Hintz	2099 rows	2023-03-31	Male	2087.55

2)

Database Explorer

```

1 -- 2. Rescheduling a flight.
2 -- You need to reschedule a flight.
3 -- Verify the 'flights' table reflects the new departure time.
4 -- Simulate an error to test rollback (for example, invalid flight_id).
5
6 CREATE OR REPLACE PROCEDURE reschedule_flight(pflight_id INT, pnew_departure DATE, pnew_arrival DATE)
7   LANGUAGE plpgsql
8 AS
9 $$
10 BEGIN
11   PERFORM 1 FROM flights WHERE flight_id = pflight_id;
12
13   IF NOT FOUND THEN
14     RAISE NOTICE 'Flight id % not found', pflight_id;
15     RETURN;
16   END IF;
17
18   UPDATE flights
19   SET scheduled_departure = pnew_departure,
20       scheduled_arrival = pnew_arrival
21   WHERE flight_id = pflight_id;
22
23   RAISE EXCEPTION 'force rollback';
24
25 $$/;
26
27 SELECT * FROM flights;
28 CALL reschedule_flight(flight_id 900, pnew_departure DATE '2023-10-05', pnew_arrival DATE '2023-04-02');
29

```

Services

Flight ID	Flight No	Scheduled Departure	Scheduled Arrival	Departure Airport ID	Arrival Airport ID
898	US-NY	2023-05-01	2023-11-21	15	6
899	DZ-01	2023-08-28	2023-07-26	5	3
900	PF-U-A	2023-10-02	2023-04-01	15	11

Database Consoles > postgres@localhost > console

Services

Flight ID	Flight No	Scheduled Departure	Scheduled Arrival	Departure Airport ID	Arrival Airport ID
990	NL-DR	2023-05-24	2023-12-01	15	15
999	US-IL	2023-04-12	2024-03-07	9	4
1000	PF-U-A	2023-10-03	2023-04-02	15	11

3)

Database Explorer

```

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.

```

Services

Database Consoles > postgres@localhost > console

Output dbwork.public.flights

```

[2025-11-26 04:24:48] completed in 3 ms

```

82:4 CRLF UTF-8 4 spaces

4)

Database Explorer

```

CREATE OR REPLACE PROCEDURE update_passenger_details(
    passenger_id INT,
    pfir_name VARCHAR,
    plast_name VARCHAR,
    gender VARCHAR,
    pcitizen VARCHAR,
    pcountry_citizenship VARCHAR,
    pcountry_residence VARCHAR,
    passport_number VARCHAR
)
LANGUAGE plpgsql
AS $$
BEGIN
    PERFORM 1 FROM passengers WHERE passenger_id = passenger_id;
    IF NOT FOUND THEN
        RAISE NOTICE 'Passenger id % not found', passenger_id;
        RETURN;
    END IF;

    UPDATE passengers
    SET first_name = pfir_name,
        last_name = plast_name,
        sex = gender,
        country_of_citizenship = pcitizen,
        country_of_residence = pcountry_residence,
        passport_number = passport_number,
        update_at = CURRENT_DATE
    WHERE passenger_id = passenger_id;

    UPDATE security_check
    SET update_at = CURRENT_DATE
    WHERE passenger_id = passenger_id;

    UPDATE baggage_check
    SET update_at = CURRENT_DATE
    WHERE passenger_id = passenger_id;

    UPDATE booking
    SET update_at = CURRENT_DATE,
        status = gender
    WHERE passenger_id = passenger_id;
EXCEPTION
    WHEN OTHERS THEN
        RAISE NOTICE 'Force rollback';
END;
$$;

```

Services

Database Consoles > postgres@localhost > console

Output dbwork.public.flights

```

[2025-11-26 04:54:01] completed in 2 ms

```

135:4 CRLF UTF-8 4 spaces

5)

Database Explorer

```

CREATE OR REPLACE PROCEDURE register_passenger_and_booking(
    pfirstname 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, new_passenger_id, pfirstname, pdob, pgender, pcountry_citizenship, pcountry_residence, ppassport_number, now(), now());
    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, pbooking_platform, now(), now(), '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', '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;

```

Services

Database

	booking_created_at	booking_updated_at	status	price	passenger.passenger_id	first_name	last_name	date_of_birth	gender	country_of_citizenship
1	2025-11-26	2025-11-26	confirmed	200.00	201	Vasya	Ivanov	1990-01-01	Male	Russia

Database Consoles > postgres@localhost > console

6)

Database Explorer

```

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);

```

Services

Database

	booking_id	flight_id	price	status	created_at	updated_at
1	534	101	210.00	confirmed	2025-11-26 05:16:19	2025-11-26 05:16:19
2	530	101	210.00	confirmed	2025-11-26 05:17:01	2025-11-26 05:17:01
3	531	101	210.00	confirmed	2025-11-26 05:17:01	2025-11-26 05:17:01
4	532	101	210.00	confirmed	2025-11-26 05:17:01	2025-11-26 05:17:01

Database Consoles > postgres@localhost > console

7)

Database Explorer

```

204 -- Ensure that the change is correctly reflected in the database.
205
206 CREATE PROCEDURE update_baggage_weight(pbaggage_id INT, pnew_weight NUMERIC)
207 LANGUAGE plpgsql
208 AS $$
209 BEGIN
210     PERFORM 1 FROM baggage WHERE baggage_id = pbaggage_id;
211
212     IF NOT FOUND THEN
213         RAISE NOTICE 'Baggage id % not found', pbaggage_id;
214         RETURN;
215     END IF;
216
217     UPDATE baggage
218     SET weight_in_kg = pnew_weight,
219         update_date = CURRENT_DATE
220     WHERE baggage_id = pbaggage_id;
221
222     EXCEPTION
223     WHEN OTHERS THEN
224         RAISE NOTICE 'force rollback';
225     END;
226     $$;
227     CALL update_baggage_weight( pbaggage_id 47, pnew_weight 8.42);
228     SELECT * FROM baggage WHERE baggage_id = 47;
229

```

Services

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

Database Consoles > postgres@localhost > console

```

227 ✓ CALL update_baggage_weight( pbaggage_id 47, pnew_weight 8.42);
228 ✓ SELECT * FROM baggage WHERE baggage_id = 47;
229

```

Services

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

Database Consoles > postgres@localhost > console

8)

Database Explorer

```

233 CREATE OR REPLACE PROCEDURE apply_discount(ppassenger_id INT, pdiscount NUMERIC)
234 LANGUAGE plpgsql
235 AS $$
236 BEGIN
237     PERFORM 1 FROM booking WHERE booking_id = ppassenger_id;
238
239     IF NOT FOUND THEN
240         RAISE NOTICE 'Booking id % not found', ppassenger_id;
241         RETURN;
242     END IF;
243
244     UPDATE booking
245     SET price = price - CASE
246         WHEN price < pdiscount THEN price
247             ELSE pdiscount END,
248         update_at = CURRENT_DATE
249     WHERE passenger_id = ppassenger_id;
250
251     EXCEPTION
252     WHEN OTHERS THEN
253         RAISE NOTICE 'force rollback';
254     END;
255     $$;
256     CALL apply_discount( ppassenger_id 201, pdiscount 100);
257

```

Services

	passenger_id	price	update_at
1	201	100	2025-11-26 06:07:46

Database Consoles > postgres@localhost > console

```

[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)

The screenshot shows a PostgreSQL database interface with the following details:

- Database Explorer:** Shows the schema structure of the database, including tables like `booking`, `flights`, and `passenger`.
- console tab:** Contains the code for a stored procedure named `reschedule_flight`. The code is a CREATE OR REPLACE PROCEDURE statement with various parameters and logic for updating flight and booking records.
- Services:** Shows a transaction (Tx) and several database connections (postgreSQL@localhost, dbwork.public, passengers).
- Output tab:** Displays the command issued and its execution time: `[2025-11-26 06:42:45] dbwork.public> CALL reschedule_flight(101, '2025-12-01', '2025-12-01', 1, 2, '42', '3288', NULL, NULL)` completed in 7 ms.

```

CREATE OR REPLACE PROCEDURE reschedule_flight(
    from_flight_id INT,
    pscheduled_departure DATE,
    pscheduled_arrival DATE,
    departing_airport_id INT,
    arriving_airport_id INT,
    departing_gate VARCHAR,
    arriving_gate VARCHAR,
    actual_departure DATE,
    actual_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, scheduled_departure, pscheduled_arrival,
            scheduled_arrival_pscheduled_arrival, departure_airport_id, departing_airport_id, arrival_airport_id, arriving_airport_id,
            departing_gate, departing_gate, arriving_gate, arriving_gate, airline_id, pairline_id, status 'Delayed',
            actual_departure, actual_arrival, created_at CURRENT_DATE, updated_at CURRENT_DATE);
    UPDATE booking 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(pflight_id 101, pnew_departure '2025-12-01', pnew_arrival '2025-12-01', 1, 2, '42', '3288', NULL, NULL);

```