

Lab-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).

```
269
270 -- 1
271
272 DO $$
273 DECLARE
274     bk_id INT := 1;
275     deleted_t INT;
276 BEGIN
277     PERFORM 1 FROM bookings WHERE booking_id = bk_id;
278     IF NOT FOUND THEN
279         RAISE EXCEPTION 'Booking % not found -> ROLLBACK', bk_id;
280     END IF;
281     DELETE FROM tickets WHERE booking_id = bk_id;
282     GET DIAGNOSTICS deleted_t = ROW_COUNT;
283     RAISE NOTICE '% ticket(s) deleted for booking %', deleted_t, bk_id;
284     DELETE FROM bookings WHERE booking_id = bk_id;
285     RAISE NOTICE 'Booking % deleted (COMMIT)', bk_id;
286 END
287 $$;
288 SELECT * FROM tickets WHERE booking_id = 1;
289 SELECT * FROM bookings WHERE booking_id = 1;
290
291
```

Data Output Messages Notifications

	booking_id [PK] integer	passenger_id integer	flight_id integer	platform character varying (50)	price numeric (10,2)	check_result character varying (50)	created_at date
--	----------------------------	-------------------------	----------------------	------------------------------------	-------------------------	--	--------------------

```
269
270 -- 1
271
272 DO $$
273 DECLARE
274     bk_id INT := 999;
275     deleted_t INT;
276 BEGIN
277     PERFORM 1 FROM bookings WHERE booking_id = bk_id;
278     IF NOT FOUND THEN
279         RAISE EXCEPTION 'Booking % not found -> ROLLBACK', bk_id;
280     END IF;
281     DELETE FROM tickets WHERE booking_id = bk_id;
282     GET DIAGNOSTICS deleted_t = ROW_COUNT;
283     RAISE NOTICE '% ticket(s) deleted for booking %', deleted_t, bk_id;
284     DELETE FROM bookings WHERE booking_id = bk_id;
285     RAISE NOTICE 'Booking % deleted (COMMIT)', bk_id;
286 END
287 $$;
288 SELECT * FROM tickets WHERE booking_id = 1;
289 SELECT * FROM bookings WHERE booking_id = 1;
290
291
```

Data Output Messages Notifications

ERROR: Booking 999 not found -> ROLLBACK
CONTEXT: функция PL/pgSQL inline_code_block, строка 8, оператор RAISE

ОШИБКА: Booking 999 not found -> ROLLBACK
SQL state: P0001

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).

```
290
291 --2
292
293 DO $$
294 DECLARE
295     flight INT := 2;
296     new_time TIMESTAMP := '2024-12-01 09:00:00';
297 BEGIN
298     UPDATE flights
299     SET scheduled_departure = new_time
300     WHERE flight_id = flight;
301 IF NOT FOUND THEN
302     RAISE EXCEPTION 'Flight % not found -> ROLLBACK', flight;
303 END IF;
304 END
305 $$;
306 SELECT flight_id, flight_number, scheduled_departure FROM flights WHERE flight_id = 2;
307
```

Data Output Messages Notifications

	flight_id [PK] integer	flight_number character varying (10)	scheduled_departure timestamp without time zone
1	2	DV3302	2024-12-01 09:00:00

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.

```
308 --3
309
310 DO $$
311 DECLARE
312     flight INT := 3;
313     discount NUMERIC := 20.00;
314 BEGIN
315     PERFORM 1 FROM flights WHERE flight_id = flight;
316 IF NOT FOUND THEN
317     RAISE EXCEPTION 'Flight % not found -> ROLLBACK', flight;
318 END IF;
319
320 UPDATE bookings
321 SET price = price - discount
322 WHERE flight_id = flight;
323
324 UPDATE tickets
325 SET price = price - discount
326 WHERE booking_id IN (SELECT booking_id FROM bookings WHERE flight_id = flight);
327 END
328 $$;
329 SELECT b.booking_id, b.flight_id, b.price FROM bookings b WHERE b.flight_id = 3;
330 SELECT t.ticket_id, t.booking_id, t.price FROM tickets t WHERE t.booking_id IN (SELECT booking_id FROM bookings WHERE flight_id = 3);
331
332
```

Data Output Messages Notifications

	ticket_id [PK] integer	booking_id integer	price numeric (10,2)
1	3	3	200.00

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

```

270  --4
271
272  DO $$
273  DECLARE
274      passenger INT := 1;
275      new_fname TEXT := 'Бауыржан';
276      new_lname TEXT := 'Тлеубеков';
277      new_passport TEXT := 'KZ654321';
278  BEGIN
279      UPDATE passengers
280      SET first_name = new_fname, last_name = new_lname, passport_number = new_passport
281      WHERE passenger_id = passenger;
282  IF NOT FOUND THEN
283      RAISE EXCEPTION 'Passenger % not found -> ROLLBACK', passenger;
284  END IF;
285  END
286  $$;
287  SELECT b.booking_id, b.flight_id, p.first_name, p.last_name, p.passport_number
288  FROM bookings b
289  JOIN passengers p ON b.passenger_id = p.passenger_id
290  WHERE b.passenger_id = 1;
291

```

Data Output Messages Notifications

	booking_id integer	flight_id integer	first_name character varying (50)	last_name character varying (50)	passport_number character varying (20)
1	1	1	Бауыржан	Тлеубеков	KZ654321

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

```

353  --5
354
355
356  DO $$
357  DECLARE
358      new_passenger_id INT;
359      flight INT := 4;
360  BEGIN
361      PERFORM 1 FROM flights WHERE flight_id = flight;
362  IF NOT FOUND THEN
363      RAISE EXCEPTION 'Flight % not found -> ROLLBACK', flight;
364  END IF;
365
366      INSERT INTO passengers(first_name, last_name, gender, birth_date, passport_number, country_of_citizenship)
367      VALUES ('Alina', 'Kuat', 'Female', '2000-05-10', 'KZ999999', 'Kazakhstan')
368      RETURNING passenger_id INTO new_passenger_id;
369
370      INSERT INTO bookings(passenger_id, flight_id, platform, price, check_result, created_at)
371      VALUES (new_passenger_id, flight, 'Web', 300.00, 'Pending', CURRENT_DATE);
372  END
373  $$;
374  SELECT * FROM passengers WHERE passport_number = 'KZ999999';
375  SELECT * FROM bookings WHERE passenger_id = (SELECT passenger_id FROM passengers WHERE passport_number = 'KZ999999');
376

```

Data Output Messages Notifications

	booking_id [PK] integer	passenger_id integer	flight_id integer	platform character varying (50)	price numeric (10,2)	check_result character varying (50)	created_at date
1	21	21	4	Web	300.00	Pending	2025-11-25

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

```
377 --6
378
379 DO $$
380 DECLARE
381     flight INT := 5;
382     increment NUMERIC := 50.00;
383 BEGIN
384     PERFORM 1 FROM bookings WHERE flight_id = flight;
385     IF NOT FOUND THEN
386         RAISE EXCEPTION 'No bookings for flight % -> ROLLBACK', flight;
387     END IF;
388
389     UPDATE bookings
390     SET price = price + increment
391     WHERE flight_id = flight;
392
393     UPDATE tickets
394     SET price = price + increment
395     WHERE booking_id IN (SELECT booking_id FROM bookings WHERE flight_id = flight);
396 END
397 $$;
398 SELECT b.booking_id, b.flight_id, b.price FROM bookings b WHERE b.flight_id = 5;
399 SELECT t.ticket_id, t.booking_id, t.price FROM tickets t WHERE t.booking_id IN (SELECT booking_id FROM bookings WHERE flight_id = 5);
400
401
```

Data Output Messages Notifications

	ticket_id [PK] integer	booking_id integer	price numeric (10,2)
1		5	245.00

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

```
270
271 --7
272
273 DO $$
274 DECLARE
275     b_id INT := 1;
276     new_weight NUMERIC := 25.50;
277 BEGIN
278     UPDATE baggage
279     SET weight = new_weight
280     WHERE baggage_id = b_id;
281     IF NOT FOUND THEN
282         RAISE EXCEPTION 'Baggage % not found -> ROLLBACK', b_id;
283     END IF;
284 END
285 $$;
286 SELECT * FROM baggage WHERE baggage_id = 1;
287
```

Data Output Messages Notifications

	baggage_id [PK] integer	passenger_id integer	weight numeric (6,2)
1	1	1	25.50

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

```
418 --8
419
420 DO $$
421 DECLARE
422     passenger INT := 2;
423     discount NUMERIC := 30.00;
424     updated_rows INT;
425 BEGIN
426     UPDATE bookings
427     SET price = price - discount
428     WHERE passenger_id = passenger;
429     GET DIAGNOSTICS updated_rows = ROW_COUNT;
430     IF updated_rows = 0 THEN
431         RAISE EXCEPTION 'No bookings found for passenger % -> ROLLBACK', passenger;
432     END IF;
433
434     UPDATE tickets
435     SET price = price - discount
436     WHERE booking_id IN (SELECT booking_id FROM bookings WHERE passenger_id = passenger);
437 END
438 $$;
439 SELECT * FROM bookings WHERE passenger_id = 2;
440 SELECT * FROM tickets WHERE booking_id IN (SELECT booking_id FROM bookings WHERE passenger_id = 2);
441
```

Data Output Messages Notifications

	ticket_id [PK] integer	booking_id integer	price numeric (10,2)	created_at date
1		2	150.00	2024-05-25

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

```
442 --9
443
444 DO $$
445 DECLARE
446     old_flight INT := 1;
447     new_flight INT := 6;
448 BEGIN
449     PERFORM 1 FROM flights WHERE flight_id = old_flight;
450     IF NOT FOUND THEN
451         RAISE EXCEPTION 'Old flight % not found -> ROLLBACK', old_flight;
452     END IF;
453
454     PERFORM 1 FROM flights WHERE flight_id = new_flight;
455     IF NOT FOUND THEN
456         RAISE EXCEPTION 'New flight % not found -> ROLLBACK', new_flight;
457     END IF;
458
459     UPDATE bookings
460     SET flight_id = new_flight
461     WHERE flight_id = old_flight;
462 END
463 $$;
464 SELECT * FROM bookings WHERE flight_id IN (1,6) ORDER BY booking_id;
465
```

Data Output Messages Notifications

	booking_id [PK] integer	passenger_id integer	flight_id integer	platform character varying (50)	price numeric (10,2)	check_result character varying (50)	created_at date
1		6	6	Agency	410.00	Checked	2024-06-10