

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5
«Процедуры, функции, триггеры в PostgreSQL»
по дисциплине «Проектирование и реализация баз данных»**

**Обучающиеся Меньшинин Евгений
Факультет прикладной информатики
Группа К3241
Направление подготовки 09.03.03 Прикладная информатика
Образовательная программа Мобильные и сетевые технологии 2023
Преподаватель Говорова Марина Михайловна**

**Санкт-Петербург
2025**

Цель работы: овладеть практическими создания и использования процедур, функций и триггеров в базе данных PostgreSQL

Практическое задание.

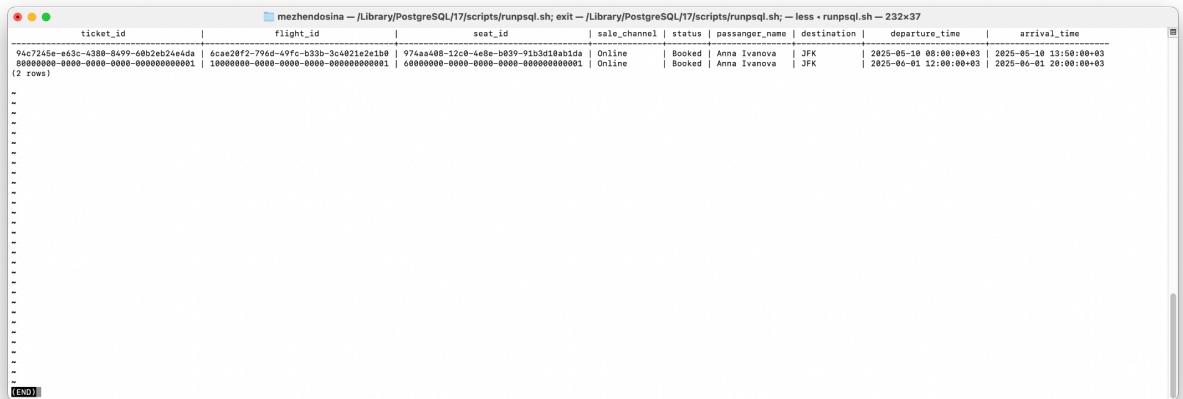
1. Создать 3 процедуры для индивидуальной БД согласно варианту (часть 4 ЛР 2). Допустимо использование IN/OUT параметров. Допустимо создать авторские процедуры. (3 балла)
2. Создать триггеры для индивидуальной БД согласно варианту: 7 оригинальных триггеров - 7 баллов (max).

Ход работы

```
CREATE          OR          REPLACE        FUNCTION
search_tickets_by_destination(dest_code text)
RETURNS TABLE (
    ticket_id uuid,
    flight_id uuid,
    seat_id uuid,
    sale_channel text,
    status text,
    passanger_name text,
    destination text,
    departure_time timestamptz,
    arrival_time timestamptz
) AS $$

BEGIN
RETURN QUERY
SELECT
    t.ticket_id,
    t.flight_id,
    t.seat_id,
    t.sale_channel,
    t.status,
    p.full_name,
    f.destination_airport,
    f.departure_time,
    f.arrival_time
FROM public."TICKETS" t
JOIN public."PASSENGERS" p ON t.passanger_id = p.passenger_id
JOIN public."FLIGHT" f ON t.flight_id = f.flight_id
WHERE f.destination_airport = dest_code;
END;
```

```
$$ LANGUAGE plpgsql;
```



A screenshot of a terminal window titled "mezhendosina". The command entered was "/Library/PostgreSQL/17/scripts/runsql.sh; exit - /Library/PostgreSQL/17/scripts/runpsql.sh; less - runpsql.sh". The output shows a table with two rows of data. The columns are: ticket_id, flight_id, seat_id, sale_channel, status, passenger_name, destination, departure_time, and arrival_time. The data is as follows:

ticket_id	flight_id	seat_id	sale_channel	status	passenger_name	destination	departure_time	arrival_time
94c7245e-e63c-4380-8499-69b2eb24e4da	6cae20f2-794d-49fc-b33b-3c4a21e2e1b8	974aa408-12c8-4e8e-b039-91b3d10ab1da	Online	Booked	Anna Ivanova	JFK	2025-05-10 08:00:00+03	2025-05-10 13:50:00+03
80000000-0000-0000-0000-000000000001	10000000-0000-0000-0000-000000000001	60000000-0000-0000-0000-000000000001	Online	Booked	Anna Ivanova	JFK	2025-06-01 12:00:00+03	2025-06-01 20:00:00+03

The terminal window also shows numerous vertical ellipsis characters (dots) indicating a large amount of data has been omitted.

```

CREATE OR REPLACE PROCEDURE create_cash_register(new_address
text, new_status text)
LANGUAGE plpgsql
AS $$

DECLARE
    new_id uuid;

BEGIN
    new_id := uuid_generate_v4();
    INSERT INTO public."CASH_REGISTERS" (cash_register_id, address,
status)
        VALUES (new_id, new_address, new_status);
    RAISE NOTICE 'Created cash register: %', new_id;
END;
$$;

```

```

;
;

[COMMIT;
CALL create_cash_register("Moscow SVO terminal", "Opened");
aeroport# clear
[aeroport# CALL create_cash_register("Moscow SVO terminal", "Opened");
[aeroport# SELECT * FROM public."CASH_REGISTERS";
[aeroport# "
[aeroport# ]
aeroport# ;
aeroport# ;
ERROR: syntax error at or near "Opened"
LINE 1: CALL create_cash_register("Moscow SVO terminal, "Opened")
[
aeroport=# CALL create_cash_register('MSVO terminal', 'Open');
NOTICE: Created cash register: 42c90417-0b85-4dc4-a283-bb529b449ea4
[CALL
aeroport# SELECT * FROM public."CASH_REGISTERS";
cash_register_id | address | status
-----+-----+-----+
163304b5-3f0b-48a1-9534-a7cc61e944ab | New York Terminal 4 | InUse
fa1d8dba-60fa-4f7d-bc3a-8817f09d2791 | London Heathrow Terminal 3 | Open
2e5d7bee-43bb-450d-88f5-18e2c563c5b8 | Paris CDG Terminal 2E | Closed
a430db59-ca6e-4eae-acf3-7a77733cfe09 | Moscow Airport Terminal B | UnderMaintenance
42c90417-0b85-4dc4-a283-bb529b449ea4 | MSVO terminal | Open
(5 rows)

aeroport# ]

```

Триггер функции

1. Логирование изменений в билетах

```
CREATE OR REPLACE FUNCTION log_ticket_status_change()
RETURNS trigger AS $$

BEGIN
    IF TG_OP = 'UPDATE' AND NEW.status <> OLD.status THEN
        INSERT INTO ticket_changes_log(ticket_id, old_status, new_status, user_name, operation)
        VALUES (OLD.ticket_id, OLD.status, NEW.status, current_user, 'update');
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trg_log_ticket_status_change
AFTER UPDATE ON public."TICKETS"
FOR EACH ROW EXECUTE PROCEDURE log_ticket_status_change();
```

```

mezhendosina — /Library/PostgreSQL/17/scripts/runpsql.sh; exit — /Library/PostgreSQL/17/scripts/runpsql.sh; — psq...
42c90417-0b85-4dc4-a283-bb529b449ea4 | MSVO terminal | Open
(5 rows)

aeroport=# CREATE TABLE ticket_changes_log (
aeroport#   log_id serial PRIMARY KEY,
aeroport#   ticket_id uuid,
aeroport#   old_status text,
aeroport#   new_status text,
aeroport#   change_time timestamptz DEFAULT now(),
aeroport#   user_name text,
aeroport#   operation text -- update/delete
aeroport# );
CREATE TABLE
[aeroport=#
aeroport=# CREATE OR REPLACE FUNCTION log_ticket_status_change()
aeroport=# RETURNS trigger AS $$
aeroport$# BEGIN
aeroport$#   IF TG_OP = 'UPDATE' AND NEW.status <> OLD.status THEN
aeroport$#     INSERT INTO ticket_changes_log(ticket_id, old_status, new_status, user_name, operation)
aeroport$#       VALUES (OLD.ticket_id, OLD.status, NEW.status, current_user, 'update');
aeroport$#   END IF;
aeroport$#   RETURN NEW;
aeroport$# END;
aeroport$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
aeroport=# CREATE TRIGGER trg_log_ticket_status_change
aeroport=# AFTER UPDATE ON public."TICKETS"
aeroport=# FOR EACH ROW EXECUTE PROCEDURE log_ticket_status_change();
CREATE TRIGGER
aeroport=# INSERT INTO public."TICKETS" (
aeroport#   ticket_id, flight_id, seat_id, sale_channel, cash_register_id, status, additional_fee, passanger_id
aeroport# ) VALUES (
aeroport#   uuid_generate_v4(), -- подойдет любой uuid
aeroport#   uuid_generate_v4(), -- можно временно подставить несуществующие (если FK отключить NOT VALID)
aeroport#   uuid_generate_v4(),
'aOnline',
aeroport#   'a430db59-ca6e-4eae-acf3-7a77733cfe09',
'aBooked',
aeroport#   0,
aeroport#   '858d93c1-5f17-40d8-bacd-aabd23d8f1b5'
aeroport# );
ERROR: insert or update on table "TICKETS" violates foreign key constraint "TICKETS_flight_id_fkey"
DETAIL: Key (flight_id)=(597f4683-f18f-4c99-9e49-fc74232b5276) is not present in table "FLIGHT".
aeroport=#
aeroport=# UPDATE public."TICKETS"
aeroport# SET status = 'Cancelled'
aeroport# WHERE status = 'Booked'
aeroport# RETURNING ticket_id;
      ticket_id
-----
94c7245e-e63c-4380-8499-60b2eb24e4da
80000000-0000-0000-000000000001
80000000-0000-0000-000000000002
80000000-0000-0000-000000000004
80000000-0000-0000-000000000005
80000000-0000-0000-000000000007
80000000-0000-0000-000000000008
80000000-0000-0000-000000000009
80000000-0000-0000-000000000010
(9 rows)

UPDATE 9
aeroport=#
aeroport=# SELECT * FROM ticket_changes_log ORDER BY log_id DESC LIMIT 5;
aeroport=#

```

```
mezhendosina — /Library/PostgreSQL/17/scripts/runpsql.sh; exit — /Library/PostgreSQL/17/scripts/...
log_id | ticket_id | old_status | new_status | change_time
| user_name | operation
-----+-----+-----+-----+-----+
      9 | 80000000-0000-0000-0000-000000000010 | Booked   | Cancelled | 2025-05-04 14:16:28.91264+
03 | postgres | update
      8 | 80000000-0000-0000-0000-000000000009 | Booked   | Cancelled | 2025-05-04 14:16:28.91264+
03 | postgres | update
      7 | 80000000-0000-0000-0000-000000000008 | Booked   | Cancelled | 2025-05-04 14:16:28.91264+
03 | postgres | update
      6 | 80000000-0000-0000-0000-000000000007 | Booked   | Cancelled | 2025-05-04 14:16:28.91264+
03 | postgres | update
      5 | 80000000-0000-0000-0000-000000000005 | Booked   | Cancelled | 2025-05-04 14:16:28.91264+
03 | postgres | update
(5 rows)

(END)
```

2. Автоматическая постановка MaintenanceRequired

```
mezhendosina — /Library/PostgreSQL/17/scripts/runpsql.sh; exit — /Library/PostgreSQL/17/scripts/runpsql.sh;...
aeroport(#     uuid_generate_v4(),
aeroport(#     'Online',
aeroport(#     'a430db59-ca6e-4eae-acf3-7a77733cfe09',
aeroport(#     'Booked',
aeroport(#     0,
aeroport(#     '858d93c1-5f17-40d8-bacd-aabd23d8f1b5'
aeroport(# );
ERROR: insert or update on table "TICKETS" violates foreign key constraint "TICKETS_flight_id_fkey"
DETAIL: Key (flight_id)=(597f4683-f18f-4c99-9e49-fc74232b5276) is not present in table "FLIGHT".
aeroport=#
aeroport=# UPDATE public."TICKETS"
aeroport-# SET status = 'Cancelled'
aeroport-# WHERE status = 'Booked'
aeroport-# RETURNING ticket_id;
          ticket_id
-----
94c7245e-e63c-4380-8499-60b2eb24e4da
80000000-0000-0000-000000000001
80000000-0000-0000-000000000002
80000000-0000-0000-000000000004
80000000-0000-0000-000000000005
80000000-0000-0000-000000000007
80000000-0000-0000-000000000008
80000000-0000-0000-000000000009
80000000-0000-0000-000000000010
(9 rows)

UPDATE 9
aeroport=#
aeroport=# SELECT * FROM ticket_changes_log ORDER BY log_id DESC LIMIT 5;
aeroport=# CREATE OR REPLACE FUNCTION auto_maintenance_required()
aeroport-# RETURNS trigger AS $$
aeroport$# BEGIN
aeroport$#     IF NEW.last_technical_service < now() - INTERVAL '180 days' THEN
aeroport$#         NEW.status := 'MaintenanceRequired';
aeroport$#     END IF;
aeroport$#     RETURN NEW;
aeroport$# END;
aeroport$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
aeroport=# CREATE TRIGGER trg_plane_maintenance
aeroport-# BEFORE INSERT OR UPDATE ON public."PLANES"
aeroport-# FOR EACH ROW EXECUTE PROCEDURE auto_maintenance_required();
CREATE TRIGGER
aeroport=#
aeroport=# INSERT INTO public."PLANES" (
aeroport(#     plane_id, model_id, company_id, status, flight_duration, last_technical_service
aeroport(# ) VALUES (
aeroport(#     uuid_generate_v4(),
aeroport(#     'e1468bc8-ca46-4b03-9e18-2f07dfe6999f',
aeroport(#     'e1169c5a-068a-4ffa-802f-bf6896c3c47c',
aeroport(#     'InService',
aeroport(#     '05:00:00',
aeroport(#     now() - INTERVAL '200 days'
aeroport(# );
INSERT 0 1
aeroport=#
aeroport=# SELECT * FROM public."PLANES" ORDER BY last_technical_service ASC LIMIT 1;
[aeroport=# SELECT * FROM public."PLANES" ORDER BY last_technical_service;
aeroport=# SELECT * FROM public."PLANES" ORDER BY last_technical_service WHERE flight_duration = '05:00:00';
ERROR: syntax error at or near "WHERE"
LINE 1: ...M public."PLANES" ORDER BY last_technical_service WHERE flig...
[
aeroport=# SELECT * FROM public."PLANES" WHERE flight_duration = '05:00:00';
aeroport=#
[
```

```
mezhendosina — /Library/PostgreSQL/17/scripts/runpsql.sh; exit — /Library/PostgreSQL/17/scripts/runpsql.sh;...
          plane_id      |           model_id      |           company_id
          |   status       |   flight_duration |   last_technical_service |   year_built
-----+-----+-----+-----+-----+
23461c1d-fb24-4d33-81e2-c4faa1fa0fa0 | e1468bc8-ca46-4b03-9e18-2f07dfe6999f | e1169c5a-068a-4ffa-802f-bf6896c3
c47c | MaintenanceRequired | 05:00:00      | 2024-10-16 14:18:44.767901+03 |
(1 row)

(END)
```

3. Запрещать удаление занятых кресел

```
mezhendosina — /Library/PostgreSQL/17/scripts/runpsql.sh; exit — /Library/PostgreSQL/17/scripts/runpsql.sh;...  
[aeroport=#  
[aeroport=#  
[aeroport=#  
[aeroport=#  
[aeroport=#  
[aeroport=# CREATE OR REPLACE FUNCTION autofill_medical_reason()  
[aeroport-# RETURNS trigger AS $$  
[aeroport$# BEGIN  
[aeroport$#     IF NEW.medical_status <> 'FitforDuty' AND (NEW.medical_reason IS NULL OR NEW.medical_reason = '')  
[aeroport$# THEN  
[aeroport$#         NEW.medical_reason := 'Не допущен по медоснованиям';  
[aeroport$#     END IF;  
[aeroport$#     RETURN NEW;  
[aeroport$# END;  
[aeroport$# $$ LANGUAGE plpgsql;  
[aeroport=CREATE FUNCTION  
[aeroport=#  
[aeroport=# CREATE TRIGGER trg_autofill_medical_reason  
[aeroport-# BEFORE INSERT OR UPDATE ON public."CREW"  
[aeroport-# FOR EACH ROW EXECUTE PROCEDURE autofill_medical_reason();  
[aeroport=CREATE TRIGGER  
[aeroport=# INSERT INTO public."CREW"(crew_id, role, medical_check_date, medical_status, medical_reason)  
[aeroport-# VALUES (uuid_generate_v4(), 'Pilot', CURRENT_DATE, 'UnfitforDuty', NULL);  
[aeroport=INSERT 0 1  
[aeroport=# SELECT medical_reason FROM public."CREW" WHERE medical_status='UnfitforDuty';  
[aeroport=medical_reason  
[aeroport=-----  
[aeroport=| Не допущен по медоснованиям  
[aeroport=(1 row)  
[aeroport=
```

4. Автозаполнение medical_reason при не допуске Crew

```
mezhendosina — /Library/PostgreSQL/17/scripts/runpsql.sh; exit — /Library/PostgreSQL/17/scripts/runpsql.sh;...  
[aeroport=#  
[aeroport=# CREATE OR REPLACE FUNCTION prevent_delete_booked_seat()  
[aeroport-# RETURNS trigger AS $$  
[aeroport$# BEGIN  
[aeroport$#     IF OLD.is_booked THEN  
[aeroport$#         RAISE EXCEPTION 'Cannot delete booked seat!';  
[aeroport$#     END IF;  
[aeroport$#     RETURN OLD;  
[aeroport$# END;  
[aeroport$# $$ LANGUAGE plpgsql;  
[aeroport=CREATE FUNCTION  
[aeroport=# CREATE TRIGGER trg_prevent_delete_booked_seat  
[aeroport-# BEFORE DELETE ON public."SEATS"  
[aeroport-# FOR EACH ROW EXECUTE PROCEDURE prevent_delete_booked_seat();  
[aeroport=CREATE TRIGGER  
[aeroport=# INSERT INTO public."SEATS"(seat_id, flight_id, seat_number, seat_type, base_price, is_booked)  
[aeroport-# VALUES (uuid_generate_v4(), uuid_generate_v4(), '1A', 'Economy', 120, TRUE);  
[aeroport=ERROR: insert or update on table "SEATS" violates foreign key constraint "SEATS_flight_id_fkey"  
[aeroport=DETAIL: Key (flight_id)=(cd95e7e6-a5c5-42cf-bb74-2733c2755b65) is not present in table "FLIGHT".  
[aeroport=# INSERT INTO public."SEATS"(seat_id, flight_id, seat_number, seat_type, base_price, is_booked)  
[aeroport-# VALUES (uuid_generate_v4(), '', '1A', 'Economy', 120, TRUE);  
[aeroport=ERROR: invalid input syntax for type uuid: ""  
[aeroport=LINE 2: VALUES (uid_generate_v4(), '', '1A', 'Economy', 120, TRUE);  
[aeroport=^  
[aeroport=# INSERT INTO public."SEATS"(seat_id, flight_id, seat_number, seat_type, base_price, is_booked)  
[aeroport-# VALUES (uid_generate_v4(), '6cae20f2-796d-49fc-b33b-3c4021e2e1b0', '1A', 'Economy', 120, TRUE);  
[aeroport=INSERT 0 1  
[aeroport=# DELETE FROM public."SEATS" WHERE seat_number='1A';  
[aeroport=ERROR: Cannot delete booked seat!  
[aeroport=CONTEXT: PL/pgSQL function prevent_delete_booked_seat() line 4 at RAISE  
[aeroport=
```

5. Автоматическая блокировка покупки билета при закрытой кассе

```
CREATE OR REPLACE FUNCTION prevent_ticket_if_cash_register_closed()
```

```
RETURNS trigger AS $$
```

```
DECLARE
```

```
    reg_status text;
```

```
BEGIN
```

```
    SELECT status INTO reg_status FROM public."CASH_REGISTERS" WHERE  
cash_register_id = NEW.cash_register_id;
```

```
    IF reg_status = 'Closed' THEN
```

```
        RAISE EXCEPTION 'Ticket sales are closed at this cash register!';
```

```
    END IF;
```

```
    RETURN NEW;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trg_prevent_ticket_if_cash_register_closed
```

```
BEFORE INSERT ON public."TICKETS"
```

```
FOR EACH ROW EXECUTE PROCEDURE prevent_ticket_if_cash_register_closed();
```

The screenshot shows a PostgreSQL terminal window with the following session:

```
mezhendosina - /Library/PostgreSQL/17/scripts/runpsql.sh; exit - /Library/PostgreSQL/17/scripts/runpsql.sh; - psql - runpsql.sh - 229x17
LINE 2: VALUES ('44444444-4444-4444-4444-444444444444', '0cae20f2-79...
aeroport# aeroport# INSERT INTO public."TICKETS" (ticket_id, flight_id, seat_id, sale_channel, cash_register_id, status, additional_fee, passenger_id)
aeroport# VALUES
aeroport#   ('0cae20f2-796d-49f-833b-3c4021e2e1b0',
aeroport#   '0cae20f2-796d-49f-833b-3c4021e2e1b0',
aeroport#   '44444444-4444-4444-4444-444444444444',
aeroport#   'Online',
aeroport#   '2e293be-43bb-45bd-48f5-18e2c563c5b8',
aeroport#   'Booked'
aeroport#   '0',
aeroport#   '858d93c1-5f17-40d8-bacd-aabd23d8f1b5'
aeroport# );
ERROR:  Ticket sales are closed at this cash register!
CONTEXT:  PL/pgSQL function prevent_ticket_if_cash_register_closed() line 7 at RAISE
aeroport#
```

6. Автоматическое освобождение места при отмене билета

The screenshot shows a PostgreSQL terminal window with the following session:

```
mezhendosina - /Library/PostgreSQL/17/scripts/runpsql.sh; exit - /Library/PostgreSQL/17/scripts/runpsql.sh; - psql - runpsql.sh - 229x42
aeroport# aeroport# CREATE OR REPLACE FUNCTION free_seat_on_cancel()
aeroport# RETURNS trigger AS $$
aeroport# BEGIN
aeroport#     IF NEW.status IN ('Cancelled', 'Refunded')
aeroport#         AND OLD.status NOT IN ('Cancelled', 'Refunded') THEN
aeroport#         UPDATE public."SEATS"
aeroport#             SET is_booked = FALSE
aeroport#             WHERE seat_id = OLD.seat_id;
aeroport#     END IF;
aeroport#     RETURN NEW;
aeroport# END;
aeroport# $$ LANGUAGE plpgsql;
CREATE FUNCTION
aeroport# aeroport# CREATE TRIGGER trg_free_seat_on_cancel
aeroport# AFTER UPDATE ON public."TICKETS"
aeroport# FOR EACH ROW EXECUTE PROCEDURE free_seat_on_cancel();
CREATE TRIGGER
aeroport# SELECT * FROM public."TICKETS"
aeroport# ;
aeroport# UPDATE public."TICKETS" SET status = 'Booked';
UPDATE 9
aeroport# SELECT * FROM public."TICKETS"
;
aeroport# UPDATE public."TICKETS" SET status = 'Cancelled' WHERE ticket_id = '94c7245e-e63c-4380-8499-60b2eb24e4da';
UPDATE 1
aeroport# SELECT * FROM public."TICKETS"
;
aeroport# SELECT * FROM public."SEATS" WHERE seat_id = '974aa408-12c0-4e8e-b039-91b3d10ab1da';
-----+-----+-----+-----+-----+
seat_id          | flight_id       | seat_number | seat_type | base_price | is_booked
-----+-----+-----+-----+-----+
974aa408-12c0-4e8e-b039-91b3d10ab1da | 0cae20f2-796d-49f-833b-3c4021e2e1b0 | 12A          | Economy    |      15880  | f
(1 row)
aeroport#
```

7. Уведомлять о рейсах без назначенного самолета

```
aeroport=# 
aeroport=# CREATE OR REPLACE FUNCTION notify_missing_plane()
aeroport=# RETURNS trigger AS $$
aeroport=$ BEGIN
aeroport=$$ IF NEW.plane_id IS NULL THEN
aeroport=$$$ RAISE NOTICE 'Flight % registered without a plane!', NEW.flight_id;
aeroport=$$$ END IF;
aeroport=$$$ RETURN NEW;
aeroport=$$$ END;
aeroport=$ LANGUAGE plpgsql;
CREATE FUNCTION
aeroport=$ 
aeroport=$# CREATE TRIGGER trg_missing_plane
aeroport=$#   INSTEAD OF INSERT ON public."FLIGHT"
aeroport=$#   FOR EACH ROW EXECUTE PROCEDURE notify_missing_plane();
CREATE TRIGGER
aeroport=$# TRIGFT INTO public."FLIGHT"(status, flight_id, departure_airport, destination_airport, arrival_time, departure_time)
aeroport=$# VALUES ('Scheduled', uuid_generate_v4(), 'SVO', 'JFK', now() + interval '5 hours', now());
NOTICE: Flight 86d0daa7-9f4e-4ebb-aabb-38d69e86a4aa registered without a plane!
INSERT 0 1
aeroport= #
```

Дополнительное задание

```
create or replace function fn_check_time_punch() returns trigger as $psql$ 
begin
  if (
    select COUNT(tps.punch_time)
    from time_punch tps
    where tps.punch_time = new.punch_time) > 0 OR (
    select COUNT(tps.employee_id)
    from time_punch tsp
    WHERE tps.employee_id = new.employee_id) = 0 AND new.is_out_punch =
true
  ) OR new.is_out_punch = (
    select tps.is_out_punch
    from time_punch tps
    where tps.employee_id = new.employee_id
    order by tps.id desc limit 1
  )
then
  return null;
end if;
return new;
end;
$psql$ language plpgsql;
```

Эта функция проверяет одновременный вход, чтобы первым был только вход, чтобы последнее значение входа было не равно добавленному. Это покрывает следующие новые случаи:

1. Два id сотрудника заходят одновременно
2. Зашел вышел одновременно
3. Впервые вышел, но не зашел
4. Два id сотрудника вышли одновременно

Выводы

Научился создавать/вызывать функции и процедуры, создал несколько триггеров и протестировал их. Изменил триггер функцию из лабораторного практикума, чтобы ее не так просто было сломать.