14_Nemkov

Task 1

Сделать так, чтобы при добавлении нового сотрудника или обновлении зарплаты сотрудника или удалении сотрудника, автоматически изменялась сумма зарплат всех сотрудников отдела.

```
-- INSERT
CREATE OR REPLACE FUNCTION insert_trigger()
RETURNS TRIGGER AS $$
BEGIN
  UPDATE departments
  SET total_salary = total_salary + NEW.salary
  WHERE id = NEW.department_id;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

-- UPDATE
CREATE OR REPLACE FUNCTION update_trigger()
RETURNS TRIGGER AS $$
BEGIN
  IF NEW.department_id = OLD.department_id THEN
    UPDATE departments
    SET total_salary = total_salary + NEW.salary - OLD.salary
    WHERE id = NEW.department_id;
  ELSE
    UPDATE departments
    SET total_salary = total_salary - OLD.salary
    WHERE id = OLD.department_id;

    UPDATE departments
    SET total_salary = total_salary + NEW.salary
    WHERE id = NEW.department_id;
  END IF;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

-- DELETE
CREATE OR REPLACE FUNCTION delete_trigger()
RETURNS TRIGGER AS $$
```

```
BEGIN
  UPDATE departments
  SET total_salary = total_salary - OLD.salary
  WHERE id = OLD.department_id;
  RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER trg_insert_salary
AFTER INSERT ON employees
FOR EACH ROW
EXECUTE FUNCTION insert_trigger();

CREATE OR REPLACE TRIGGER trg_update_salary
AFTER UPDATE ON employees
FOR EACH ROW
EXECUTE FUNCTION update_trigger();

CREATE OR REPLACE TRIGGER trg_delete_salary
AFTER DELETE ON employees
FOR EACH ROW
EXECUTE FUNCTION delete_trigger();
```

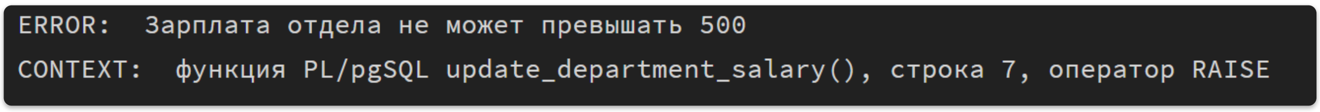
Task 2

Запретить добавление нового сотрудника в отдел, если общая сумма зарплат всех сотрудников отдела превышает 500.
Подсказка: нужно использовать исключение.

```
CREATE OR REPLACE FUNCTION check_department_salary_limit()
RETURNS TRIGGER AS $$
DECLARE
 current_total INTEGER;
 salary_limit INTEGER := 500;
BEGIN
 SELECT total_salary INTO current_total
 FROM departments
 WHERE id = NEW.department_id;

 IF (current_total + NEW.salary > salary_limit) THEN
 RAISE EXCEPTION 'Зарплата отдела не может превышать 500',
 current_total, NEW.salary, salary_limit;
 END IF;
 RETURN NEW;
END;
$$ LANGUAGE plpgsql;
CREATE TRIGGER check_salary_limit_before_insert
BEFORE INSERT ON employees
FOR EACH ROW EXECUTE FUNCTION check_department_salary_limit();
```

Проверка

```
INSERT INTO employees (name, department_id, grade, salary, email, boss_id)
VALUES ('Проверка', 2, 'junior', 460, 'test@mirea.ru', 7);
```



```
ERROR:  Зарплата отдела не может превышать 500
CONTEXT:  функция PL/pgSQL update_department_salary(), строка 7, оператор RAISE
```

image-21.png

Task 3

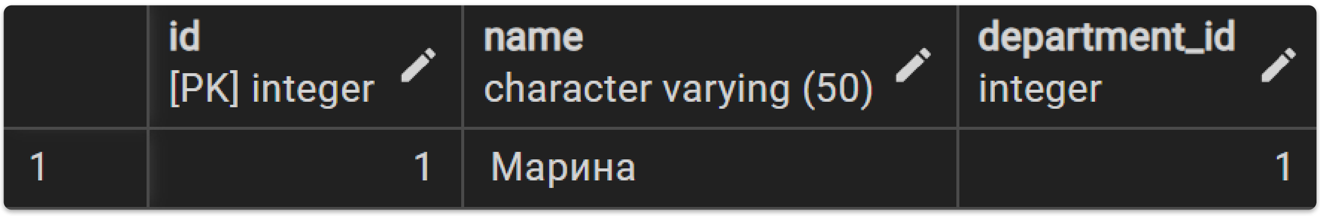
Написать функцию для перевода сотрудника из одного отдела в другой и залогировать это действие.
Сигнатура функции: transfer_employee(employee_id INT, new_department_id INT)
Таблица для логов:

```
CREATE TABLE department_transfer_log (
id SERIAL PRIMARY KEY,
employee_id INT,
old_department_id INT,
new_department_id INT,
transfer_date TIMESTAMP DEFAULT NOW() );
```

```
CREATE OR REPLACE FUNCTION transfer_employee(employee_id INT,
new_department_id INT)
RETURNS VOID AS $$
DECLARE
 old_dept_id INT;
BEGIN
 SELECT department_id INTO old_dept_id
 FROM employees
 WHERE id = employee_id;
 UPDATE employees
 SET department_id = new_department_id
 WHERE id = employee_id;

 INSERT INTO department_transfer_log (employee_id,
old_department_id, new_department_id)
 VALUES (employee_id, old_dept_id, new_department_id);

 RAISE NOTICE 'Работник % перевелся из отдела % в %',
 employee_id, old_dept_id, new_department_id;
END;
$$ LANGUAGE plpgsql;
```

Проверка

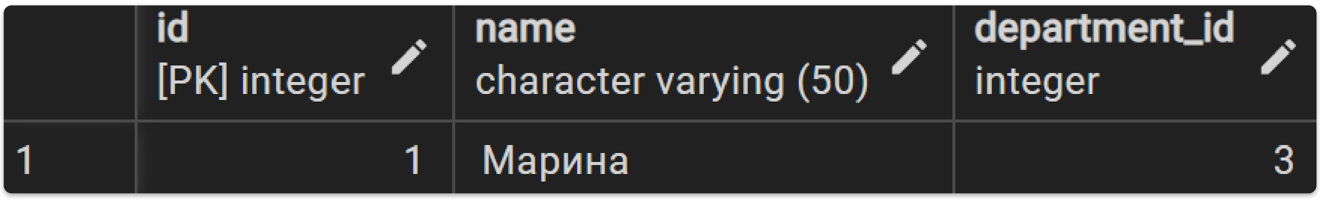
```
SELECT id, name, department_id FROM employees WHERE name = 'Марина';
```



	id [PK] integer	name character varying (50)	department_id integer
1	1	Марина	1

image-22.png

```
SELECT transfer_employee(1, 3);
SELECT id, name, department_id FROM employees WHERE id = 1;
```



	id [PK] integer	name character varying (50)	department_id integer
1	1	Марина	3

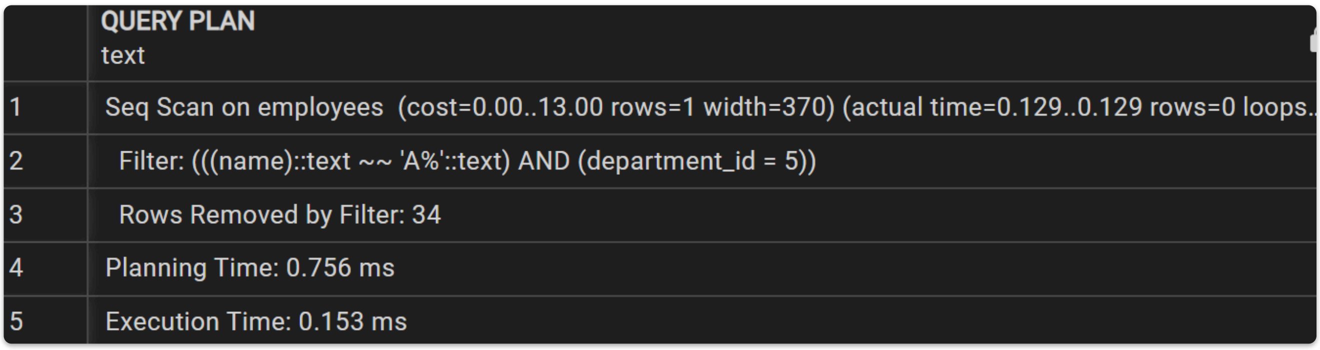
image-23.png

Task 4

1. Оптимизировать запрос для поиска сотрудника по имени и отделу.
   Пример запроса:

```
SELECT * FROM employees
WHERE name LIKE 'A%' AND department_id = 5;
```

```
EXPLAIN ANALYSE SELECT * FROM employees
WHERE name LIKE 'A%' AND department_id = 5;
```

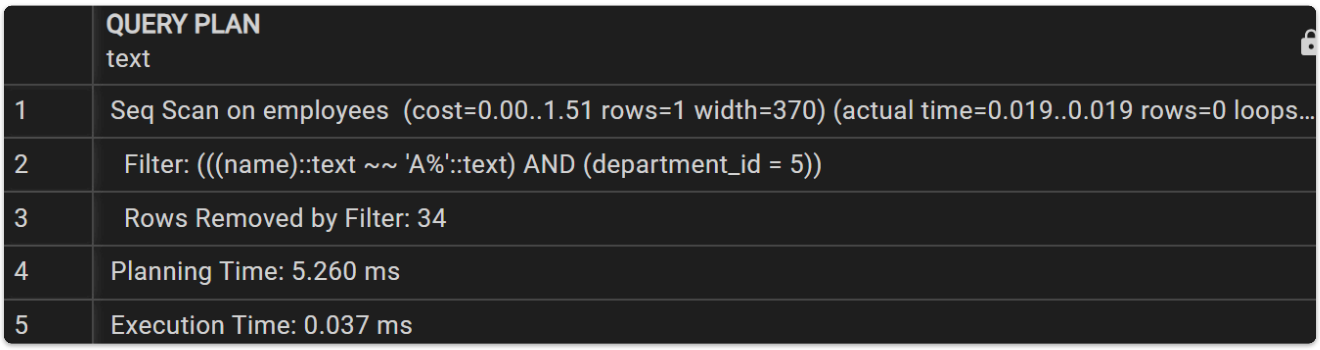


```
QUERY PLAN
text
1  Seq Scan on employees  (cost=0.00..13.00 rows=1 width=370) (actual time=0.129..0.129 rows=0 loops..
2    Filter: (((name)::text ~~ 'A%'::text) AND (department_id = 5))
3    Rows Removed by Filter: 34
4  Planning Time: 0.756 ms
5  Execution Time: 0.153 ms
```

image-24.png

```
CREATE INDEX idx_employee_name_dept ON employees(name, department_id);
```

```
EXPLAIN ANALYSE SELECT * FROM employees
WHERE name LIKE 'A%' AND department_id = 5;
```



```
QUERY PLAN
text
1  Seq Scan on employees  (cost=0.00..1.51 rows=1 width=370) (actual time=0.019..0.019 rows=0 loops...
2    Filter: (((name)::text ~~ 'A%'::text) AND (department_id = 5))
3    Rows Removed by Filter: 34
4  Planning Time: 5.260 ms
5  Execution Time: 0.037 ms
```

image-25.png

Task 5

Написать функцию, которая меняет домен для почты сотрудника.

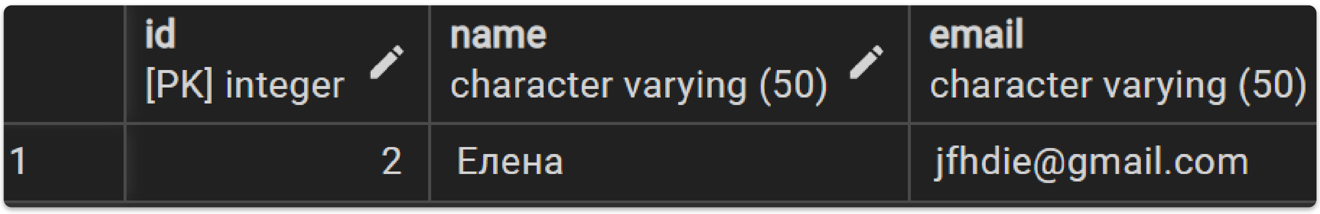
Сигнатура функции: change_domain(employee_id INT, new_domain varchar(50))

Пример вызова функции: change_domain(2, `@mirea.ru`)

```
CREATE OR REPLACE FUNCTION change_domain(employee_id INT,
new_domain VARCHAR(50))
RETURNS VOID AS $$
DECLARE
 old_email VARCHAR(50);
 new_email VARCHAR(50);
```

```
  username VARCHAR(50);
BEGIN
  SELECT email INTO old_email
  FROM employees
  WHERE id = employee_id;

  username := split_part(old_email, '@', 1);

  new_email := username || new_domain;

  UPDATE employees
  SET email = new_email
  WHERE id = employee_id;

  RAISE NOTICE 'Изменен email сотрудника % был: % стал: %',
  employee_id, old_email, new_email;
END;
$$ LANGUAGE plpgsql;
```

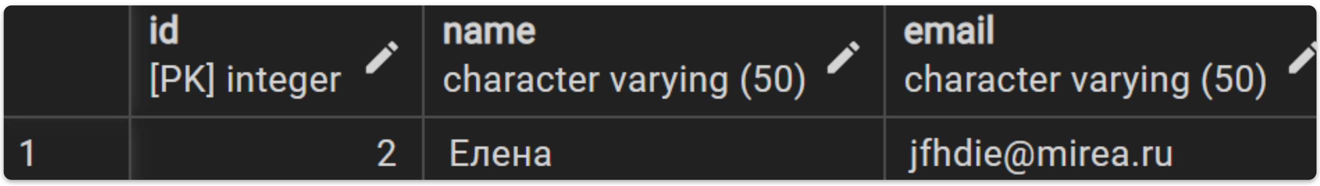
```
SELECT id, name, email FROM employees WHERE id = 2;
```



	id [PK] integer	name character varying (50)	email character varying (50)
1	2	Елена	jfhdie@gmail.com

image-26.png

```
SELECT change_domain(2, '@mirea.ru');
SELECT id, name, email FROM employees WHERE id = 2;
```



	id [PK] integer	name character varying (50)	email character varying (50)
1	2	Елена	jfhdie@mirea.ru

image-27.png