



Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного бюджетного  
образовательного учреждения высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)

**ФАКУЛЬТЕТ** **ИУК «Информатика и управление»**

**КАФЕДРА** **ИУК2 «Информационные системы и сети»**

## **Лабораторная работа №3**

### **Триггеры**

**ДИСЦИПЛИНА:** «Базы данных»

Выполнил: студент гр. ИУК5-32Б

\_\_\_\_\_  
(Подпись)

Ли Р. В.

(Ф.И.О.)

Проверил:

\_\_\_\_\_  
(Подпись)

Крысин И. А.

(Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Цель: полученные практических навыков разработки и использования триггеров.

Задачи: разработать триггеры для контроля целостности данных предметной области.

### Листинг

```
CREATE OR REPLACE FUNCTION get_timestamp()
RETURNS TRIGGER AS $$
BEGIN
    RAISE NOTICE 'Surname: %, Name: %, Group: %, Year: %', new.student_surname,
new.student_name, new.student_group, new student_year;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER update_trigger
AFTER UPDATE on students
for each row
    execute FUNCTION get_timestamp();

CREATE TRIGGER delete_trigger
BEFORE DELETE on students
for each row
    execute FUNCTION get_timestamp();

CREATE OR REPLACE FUNCTION instead_of_insert_trigger_function()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO students (student_surname, student_name, student_group,
student_year, student_age)
    VALUES (new.student_surname, new.student_name, new.student_group,
new.student_year, new.student_age);
    raise notice 'INSERTED NEW STUDENT % %', new.student_surname, new.student_name;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE VIEW new_students as
    SELECT student_surname, student_name, student_group FROM students;

create trigger insert_trigger
instead of update on new_students
for each row
    execute function instead_of_insert_trigger_function();
```

Рис 1. Создание триггеров.

```
students_db=# insert into new_students(student_surname, student_name, student_group)
values('Jack', 'Smith', 'D');
INSERT 0 1
```

Рис 2. Срабатывание триггера INSERT

```
INSERT 0 1
students_db=# update students set student_group = 'A' where student_group = 'B';
NOTICE: Surname: Brown, Name: David, Group: A, Year: (5,Brown,David,23,A,4)
NOTICE: Surname: Brown, Name: David, Group: A, Year: (10,Brown,David,23,A,4)
UPDATE 2
```

Рис 3. Срабатывание триггера UPDATE

```
students_db=# delete from students where student_name = 'Jack';
NOTICE: Surname: <NULL>, Name: <NULL>, Group: <NULL>, Year: <NULL>
DELETE 0
```

Рис 4. Срабатывание триггера DELETE

Вывод: были получены практические навыки разработки и использования триггеров.