



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ «ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Факультет автоматизации и информатики
Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №6

по дисциплине

“Базы данных”

Студент АС-21-1

(подпись, дата)

Станиславчук С. М.

Руководитель

Доцент

(подпись, дата)

Алексеев В. А.

Липецк 2023

Цель работы

Изучить возможности программирования серверной логики базы данных с использованием триггеров и хранимых процедур. Получить практические навыки программирования триггеров и хранимых процедур для выбранной СУБД.

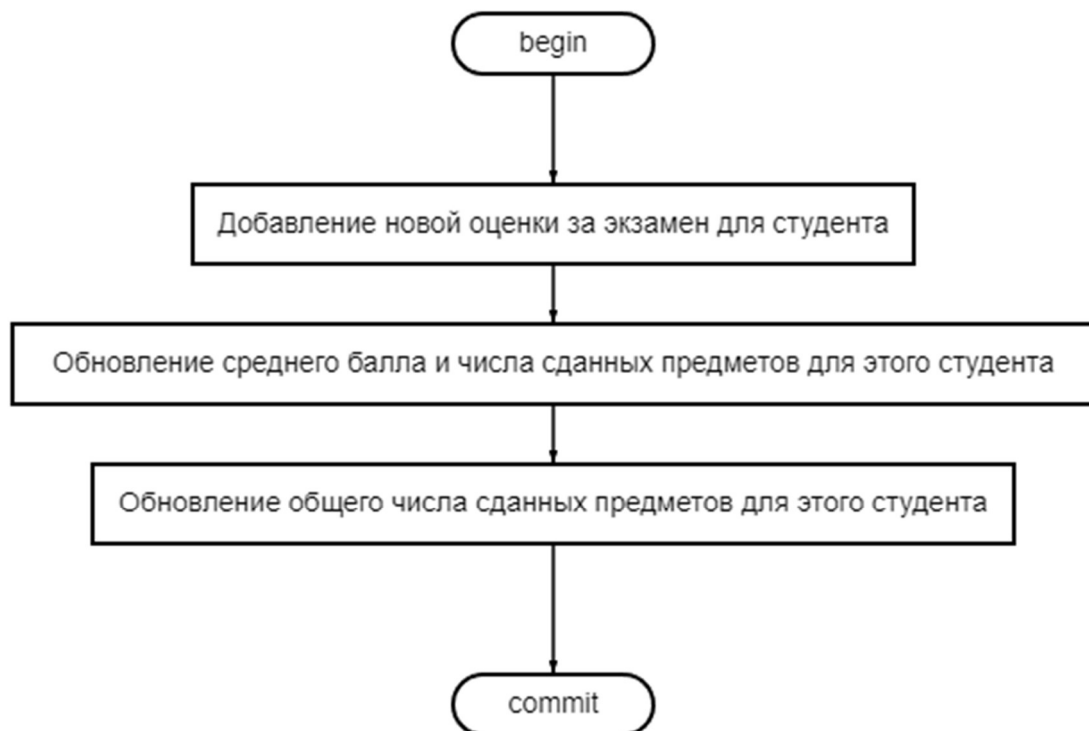
Задание

Реализовать в БД триггеры для поддержки бизнес-логики информационной системы (не менее 2-х). Реализовать хранимую процедуру для транзакции, разработанной в лабораторной работе #5. Проверить правильность работы созданных объектов БД.

По желанию могут быть реализованы также пользовательские (хранимые) функции БД

Ход работы

Блок-схема транзакции для лр#5:



3.1 Занятость преподавателя

В одно и то же преподаватель не может вести более одной пары.

Триггер: при вставке записи в таблицу расписания (schedule), проверять, не накладываются ли новые данные на уже существующие по времени занятия для преподавателя.

Расписание
<u>Id</u>
Аудитория
Учебный_год
Семестр
День_недели
Время_начала
Название_предмета
Id_учителя

Триггер:

```
CREATE OR REPLACE FUNCTION check_schedule_conflict()
RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (
        SELECT 1
        FROM schedule s
        WHERE s.id_teacher = NEW.id_teacher
              AND s.weekday_schedule = NEW.weekday_schedule
              AND s.start_time_schedule = NEW.start_time_schedule
    )
    THEN
        RAISE EXCEPTION 'Этот преподаватель занят в это время!';
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE TRIGGER schedule_conflict_trigger
BEFORE INSERT OR UPDATE ON schedule
FOR EACH ROW EXECUTE FUNCTION check_schedule_conflict();
```

Теперь при попытке добавить в расписание запись о преподавателе, который уже ведет занятия в это время, вызовется эксепшен:

```
1 INSERT INTO schedule (class_schedule, academic_year_schedule, semester_schedule,
2                       weekday_schedule, start_time_schedule, name_subject, id_teacher)
3 VALUES
4   ('101', 2023, 1, 'Среда', '09:00:00', 'Математическое программирование', 1),
5   ('102', 2023, 1, 'Среда', '09:00:00', 'Программирование', 1);
6
```

Data Output Messages Notifications

```
ERROR: Этот преподаватель занят в это время!
CONTEXT: PL/pgSQL function check_schedule_conflict() line 11 at RAISE
SQL state: P0001
```

3.2 Средний балл студента

При каждой оценке, добавляемой в таблицу успеваемости (perfomance), автоматически пересчитывать средний балл студента.

Успеваемость
<u>Id</u> Балл Дата Семестр Название_предмета Номер_студенческого_билета_студента Id_преподавателя

До SQL-запроса (обратите внимание на первую запись):

	id_performance [PK] integer	mark_performance numeric (3)	date_performance date	semester_performance character varying (6)	name_subject character varying (256)	id_number_student numeric	id_teacher integer	student_average_mark numeric	student_passed_subjects integer
1	69	55	2023-12-30	1	Операционные системы	5750846942	1	55.0000000000000000	1
2	41	46	2023-10-01	1	Математическое программирование	2429998970	1	46.0000000000000000	0
3	44	60	2023-09-04	1	Основы теории управления	6335220554	4	86.6666666666666667	3
4	45	100	2023-09-05	1	Математический анализ	6335220554	5	86.6666666666666667	3

Триггер:

```
CREATE OR REPLACE FUNCTION update_student_average_mark() RETURNS TRIGGER AS $$
```

```
DECLARE
```

```
    is_recursive_update BOOLEAN;
```

```
BEGIN
```

```
    -- Проверяем, был ли триггер вызван ранее в рамках текущей транзакции
```

```
    SELECT (pg_trigger_depth() > 1) INTO is_recursive_update;
```

```
    -- Если это не рекурсивный вызов триггера, выполняем обновление
```

```
    IF NOT is_recursive_update THEN
```

```
        UPDATE performance
```

```
        SET student_average_mark = (
```

```
            SELECT AVG(mark_performance)
```

```
            FROM performance p
```

```
            WHERE p.id_number_student = NEW.id_number_student
```

```
        )
```

```
        WHERE performance.id_number_student = NEW.id_number_student;
```

```
    END IF;
```

```
    RETURN NEW;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE TRIGGER update_average_mark_trigger
```

```
AFTER INSERT OR UPDATE ON performance
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION update_student_average_mark();
```

После SQL-запроса (на обновление оценки с 55 до 85 баллов):

	id_performance [PK] integer	mark_performance numeric (3)	date_performance date	semester_performance character varying (6)	name_subject character varying (256)	id_number_student numeric	id_teacher integer	student_average_mark numeric	student_passed_subjects integer
1	69	85	2023-12-30	1	Операционные системы	5750846942	1	85.0000000000000000	1
2	41	46	2023-10-01	1	Математическое программирование	2429998970	1	46.0000000000000000	0
3	44	60	2023-09-04	1	Основы теории управления	6335220554	4	86.6666666666666667	3
4	45	100	2023-09-05	1	Математический анализ	6335220554	5	86.6666666666666667	3

4. Хранимая процедура

Данная процедура используется для того, чтобы занести в БД запись о новой оценке студента. Благодаря чему, помимо добавления самой оценки, автоматически обновится еще и средний балл студента по этому предмету, а также общее число сданных предметов.

Успеваемость	
<u>Id</u>	
Балл	
Дата	
Семестр	
Название_предмета	
Номер_студенческого_билета_студента	
Id_преподавателя	

До:

	id_performance [PK] integer	mark_performance numeric (3)	date_performance date	semester_performance character varying (6)	name_subject character varying (256)	id_number_student numeric	id_teacher integer	student_average_mark numeric	student_passed_subjects integer
1	48	60	2023-11-08	1	Математический анализ	6906157794	2	60.0000000000000000	1

```
CREATE OR REPLACE PROCEDURE update_student_performance(  
    IN p_id_number_student NUMERIC(10),  
    IN p_name_subject VARCHAR(256),  
    IN p_mark_performace NUMERIC,  
    IN p_semester_performance VARCHAR(6),  
    IN p_id_teacher INT  
)  
AS $$  
BEGIN  
    -- 1. Добавление новой оценки за экзамен для студента  
    INSERT INTO performace (id_number_student, name_subject, mark_performace,  
        date_performance, semester_performance, id_teacher)  
    VALUES (p_id_number_student, p_name_subject, p_mark_performace,  
        CURRENT_DATE, p_semester_performance, p_id_teacher);  
  
    -- 2. Обновление среднего балла для студента  
    UPDATE performace  
    SET  
        student_average_mark = (  
            SELECT AVG(mark_performace)  
            FROM performace p  
            WHERE p.id_number_student = p_id_number_student  
                AND p.name_subject = p_name_subject  
        ),  
        student_passed_subjects = (  
            SELECT COUNT(DISTINCT name_subject)  
            FROM performace p
```

```

        WHERE p.id_number_student = p_id_number_student
        AND mark_performance >= 53
    )
WHERE EXISTS (
    SELECT 1
    FROM performance p
    WHERE p.id_number_student = p_id_number_student
        AND p.name_subject = p_name_subject
        AND p.mark_performance >= 53
);
END;
$$ LANGUAGE plpgsql;

```

```
== CALL update_student_performance(6906157794, 'Базы данных', 80, '1', 1); ==
```

После:

	id_performance [FK] integer	mark_performance numeric (3)	date_performance date	semester_performance character varying (6)	name_subject character varying (256)	id_number_student numeric	id_teacher integer	student_average_mark numeric	student_passed_subjects integer
1	123	80	2023-12-17	1	Базы данных	6906157794	1	80.000000000000000000	2
2	48	60	2023-11-08	1	Математический анализ	6906157794	2	60.000000000000000000	2

И еще одна оценка:

```
== CALL update_student_performance(6906157794, 'Базы данных', 90, '1', 2); ==
```

	id_performance [FK] integer	mark_performance numeric (3)	date_performance date	semester_performance character varying (6)	name_subject character varying (256)	id_number_student numeric	id_teacher integer	student_average_mark numeric	student_passed_subjects integer
1	125	90	2023-12-17	1	Базы данных	6906157794	2	85.000000000000000000	2
2	123	80	2023-12-17	1	Базы данных	6906157794	1	85.000000000000000000	2
3	48	60	2023-11-08	1	Математический анализ	6906157794	2	60.000000000000000000	2