

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**»

Автор: Кузнецов А. Г.

Факультет: ИКТ

Группа: К3240

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

Оглавление

Цель работы	3
Практическое задание	3
Выполнение	4
Создание хранимых процедур:	4
Модифицировать триггер	8
Вывод	10

Цель работы

Овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

Практическое задание

1. Создать процедуры/функции согласно индивидуальному заданию (часть 4).
- 2.1 Модифицировать триггер (триггерную функцию) на проверку корректности входа и выхода сотрудника (см. Практическое задание 1 Лабораторного практикума (Приложение)) с максимальным учетом «узких» мест некорректных данных по входу и выходу).
- 2.2 Создать авторский триггер по варианту индивидуального задания.





Выполнение

Вариант 12. БД «Сессия»

Создание хранимых процедур:

- Для повышения стипендии отличникам на 10%.

```
CREATE OR REPLACE PROCEDURE increase_scholarship_for_excellent_students()  
  
LANGUAGE plpgsql  
  
AS $$  
  
BEGIN  
  
    UPDATE "Session"."scholarship type" AS st  
  
    SET sum_of_scholarship = st.sum_of_scholarship * 1.1  
  
    WHERE st.scholarship_type_id IN (  
  
        SELECT s.scholarship_type_id  
  
        FROM "Session".scholarship AS s  
  
        JOIN "Session".attestation AS a ON s.studying_student_id =  
a.studying_student_id  
  
        WHERE a.mark = 'A'  
  
        GROUP BY s.scholarship_type_id  
  
        HAVING COUNT(DISTINCT a.attempt) = 1  
  
    );  
  
END;  
  
$$;
```

 sum_of_scholarship integer 	 sum_of_scholarship integer 
7700	8470
16500	18150
8250	9075
13500	14850
4400	4840
22000	24200
16500	18150
19800	21780
27500	30250

- Для перевода студентов на следующий курс.

```
CREATE OR REPLACE PROCEDURE move_students_to_next_course()  
  
LANGUAGE plpgsql  
  
AS $$  
  
UPDATE "Session".student  
  
SET course = CASE  
  
    WHEN course < 6 THEN course + 1  
  
    ELSE course  
  
END;  
  
$$;
```

	grade_book_id [PK] integer	course integer	student_surname character varying (20)	student_name character varying (20)	student_patronymic character varying (20)
1	164932	1	Зимин	Юстин	Фролович
2	164933	1	Самойлов	Флор	Лукьевич
3	164934	5	Рожкова	Мэри	Рубеновна
4	164935	1	Пестов	Осип	Петрович
5	182991	5	Королёв	Давид	Максович
6	182992	5	Фролов	Михаил	Феликсович
7	182993	2	Гаврилова	Глория	Оскарвна
8	182994	1	Щербаков	Владислав	Тарасович
9	264239	3	Морозов	Игорь	Онисимович
10	264240	4	Рогов	Мечеслав	Данилович
11	264241	3	Кулагина	Иоланта	Федосеевна
12	264242	3	Фокин	Юлий	Антонинович
13	440851	4	Суханов	Рубен	Альбертович
14	440852	1	Лукин	Леонард	Пётрович
15	440853	4	Гурьева	Камилла	Егоровна
16	440854	5	Воробьёв	Орест	Станиславович
17	491382	2	Молчанов	Май	Игоревич
18	491383	1	Моисеева	Евдокия	Пантелеймоновна
19	491384	1	Владимирова	Илона	Васильевна
20	491385	5	Куликов	Ипполит	Георгьевич
21	494414	2	Назаров	Владлен	Данилович
22	494415	5	Байрамов	Сардор	Байрамович

	grade_book_id [PK] integer	course integer	student_surname character varying (20)	student_name character varying (20)	student_patronymic character varying (20)
1	164932	2	Зимин	Юстин	Фролович
2	164933	2	Самойлов	Флор	Лукьевич
3	164934	5	Рожкова	Мэри	Рубеновна
4	164935	2	Пестов	Осип	Петрович
5	182991	5	Королёв	Давид	Максович
6	182992	5	Фролов	Михаил	Феликсович
7	182993	3	Гаврилова	Глория	Оскарвна
8	182994	2	Щербаков	Владислав	Тарасович
9	264239	4	Морозов	Игорь	Онисимович
10	264240	5	Рогов	Мечеслав	Данилович
11	264241	4	Кулагина	Иоланта	Федосеевна
12	264242	4	Фокин	Юлий	Антонинович
13	440851	5	Суханов	Рубен	Альбертович
14	440852	2	Лукин	Леонард	Пётрович
15	440853	5	Гурьева	Камилла	Егоровна
16	440854	5	Воробьёв	Орест	Станиславович
17	491382	3	Молчанов	Май	Игоревич
18	491383	2	Моисеева	Евдокия	Пантелеймоновна
19	491384	2	Владимирова	Илона	Васильевна
20	491385	5	Куликов	Ипполит	Георгьевич
21	494414	2	Назаров	Владлен	Данилович

- Для изменения оценки при успешной пересдаче экзамена

```
CREATE OR REPLACE PROCEDURE update_exam_mark(
    IN student_id_to_update integer,
    IN discipline_syllabus_id integer,
    IN new_mark character(1)
)
LANGUAGE plpgsql
AS $$
UPDATE "Session".attestation AS a
SET mark = CASE
    WHEN a.studying_student_id = student_id_to_update
        AND a.discipline_in_syllabus_id = discipline_syllabus_id
        AND new_mark > a.mark THEN new_mark
    ELSE a.mark
END
WHERE a.studying_student_id = student_id_to_update
    AND a.discipline_in_syllabus_id = discipline_syllabus_id;
$$;
```

	attestation_id [PK] integer	discipline_in_syllabus_id integer	professor_id integer	studying_student_id integer	mark character	attempt integer	attestation_date date
1	20	228	477	389660	B	3	2023-11-16

	attestation_id [PK] integer	discipline_in_syllabus_id integer	professor_id integer	studying_student_id integer	mark character	attempt integer	attestation_date date
1	20	228	477	389660	A	3	2023-11-16

Модифицировать триггер

```
CREATE OR REPLACE FUNCTION fn_check_time_punch() RETURNS TRIGGER AS $$
DECLARE
    last_action BOOLEAN;
BEGIN
    -- Проверка на существование сотрудника
    IF NOT EXISTS (
        SELECT 1
        FROM employee
        WHERE id = NEW.employee_id
    ) THEN
        RETURN NULL;
    END IF;

    -- Проверка на наличие предыдущего действия для сотрудника
    SELECT is_out_punch INTO last_action
    FROM time_punch
    WHERE employee_id = NEW.employee_id
    ORDER BY punch_time DESC, id DESC
    LIMIT 1;

    -- Проверка на отсутствие предыдущего входа перед выходом
    IF last_action IS NULL AND NEW.is_out_punch THEN
        RETURN NULL;
    END IF;

    -- Проверка на повторный выход
    IF last_action = NEW.is_out_punch THEN
        RETURN NULL;
    END IF;

    -- Проверка на попытку выхода до времени входа
    IF NEW.is_out_punch THEN
        IF EXISTS (
            SELECT 1
            FROM time_punch tps
            WHERE tps.employee_id = NEW.employee_id AND tps.is_out_punch = FALSE
            ORDER BY tps.punch_time DESC, tps.id DESC
            LIMIT 1
        ) THEN
            IF NEW.punch_time <= (
                SELECT tps.punch_time
                FROM time_punch tps
                WHERE tps.employee_id = NEW.employee_id AND tps.is_out_punch = FALSE
                ORDER BY tps.punch_time DESC, tps.id DESC
                LIMIT 1
            ) THEN
                RETURN NULL;
            END IF;
        END IF;
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER check_time_punch BEFORE INSERT ON time_punch
FOR EACH ROW EXECUTE FUNCTION fn_check_time_punch();
```



```
postgres=# INSERT INTO public.employee(id, username) VALUES (4, 'Николай');
INSERT 0 1
postgres=# INSERT INTO time_punch (employee_id, is_out_punch, punch_time) VALUES (4, TRUE, NOW());
INSERT 0 0
postgres=# INSERT INTO time_punch (employee_id, is_out_punch, punch_time) VALUES (4, FALSE, '2023-01-01 12:00:00');
INSERT 0 1
postgres=# INSERT INTO time_punch (employee_id, is_out_punch, punch_time) VALUES (4, FALSE, '2023-01-01 13:00:00');
INSERT 0 0
postgres=# INSERT INTO time_punch (employee_id, is_out_punch, punch_time) VALUES (4, TRUE, '2023-01-01 13:00:00');
INSERT 0 1
postgres=# INSERT INTO time_punch (employee_id, is_out_punch, punch_time) VALUES (4, TRUE, '2023-01-01 14:00:00');
INSERT 0 0
postgres=# INSERT INTO time_punch (employee_id, is_out_punch, punch_time) VALUES (4, FALSE, '2023-01-01 15:00:00');
INSERT 0 1
postgres=# INSERT INTO time_punch (employee_id, is_out_punch, punch_time) VALUES (4, TRUE, '2023-01-01 10:00:00');
INSERT 0 0
```

Вывод

В ходе лабораторной работы удалось овладеть навыками создания и использования процедур, функций и триггеров в базе данных PostgreSQL.