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

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. 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).

Выполнение

Вариант 12. БД «Сессия»

Создание хранимых процедур:

1. Для повышения стипендии отличникам на 10%.

```
postgres=# CREATE OR REPLACE PROCEDURE increase_scholarship_for_excellent_students()
postgres-# LANGUAGE plpgsql
postgres-# AS $$
postgres$# BEGIN
postgres$#     UPDATE "Session".scholarship
postgres$#     SET sum_of_scholarship = sum_of_scholarship * 1.1
postgres$#     WHERE scholarship_id IN (
postgres$#         SELECT scholarship_id
postgres$#         FROM "Session".scholarship
postgres$#         JOIN "Session".attestation ON "Session".scholarship.studying_student_id = "Session".attestation.studying_student_id
postgres$#         JOIN "Session"."studying student" ON "Session".scholarship.studying_student_id = "Session"."studying student".studying_student_id
postgres$#         WHERE "Session".attestation.mark = 'A'
postgres$#         AND (
postgres$#             (
postgres$#                 EXTRACT(MONTH FROM CURRENT_DATE) = 12
postgres$#                 AND (
postgres$#                     (EXTRACT(MONTH FROM "Session".attestation.attestation_date) = 1 AND EXTRACT(YEAR FROM "Session".attestation.attestation_date) = EXTRACT(YEAR FROM CURRENT_DATE) + 1)
postgres$#                     OR
postgres$#                     (EXTRACT(MONTH FROM "Session".attestation.attestation_date) = 12 AND EXTRACT(YEAR FROM "Session".attestation.attestation_date) = EXTRACT(YEAR FROM CURRENT_DATE))
postgres$#                 )
postgres$#             )
postgres$#             OR (
postgres$#                 EXTRACT(MONTH FROM CURRENT_DATE) BETWEEN 1 AND 4
postgres$#                 AND (
postgres$#                     (EXTRACT(MONTH FROM "Session".attestation.attestation_date) = 12 AND EXTRACT(YEAR FROM "Session".attestation.attestation_date) = EXTRACT(YEAR FROM CURRENT_DATE) - 1)
postgres$#                     OR
postgres$#                     (EXTRACT(MONTH FROM "Session".attestation.attestation_date) = 1 AND EXTRACT(YEAR FROM "Session".attestation.attestation_date) = EXTRACT(YEAR FROM CURRENT_DATE))
postgres$#                 )
postgres$#             )
postgres$#             OR (
postgres$#                 EXTRACT(MONTH FROM CURRENT_DATE) BETWEEN 5 AND 11
postgres$#                 AND EXTRACT(MONTH FROM "Session".attestation.attestation_date) BETWEEN 5 AND 6
postgres$#                 AND EXTRACT(YEAR FROM "Session".attestation.attestation_date) = EXTRACT(YEAR FROM CURRENT_DATE)
postgres$#             )
postgres$#         )
postgres$#     );
postgres$# END;
postgres$# $$;
CREATE PROCEDURE
```

```
postgres=# SELECT * FROM "Session".scholarship;
 scholarship_id | scholarship_type_id | payment_start | payment_end | studying_student_id | sum_of_scholarship
----------------+---------------------+---------------+-------------+---------------------+--------------------
            141 |                 453 | 2023-01-01    | 2024-06-01  |              648116 |              15000
            142 |                 453 | 2023-01-01    | 2024-06-01  |              977888 |              15000
            179 |                  94 | 2023-01-01    | 2024-06-01  |              254999 |              16500
            180 |                  94 | 2023-01-01    | 2024-06-01  |              825344 |              16500
            337 |                 108 | 2023-01-01    | 2024-06-01  |              460221 |              13500
            338 |                 108 | 2023-01-01    | 2024-06-01  |              887161 |              13500
            413 |                 349 | 2023-01-01    | 2024-06-01  |              582963 |               4000
            414 |                 349 | 2023-01-01    | 2024-06-01  |              916338 |               4000
            439 |                 625 | 2023-01-01    | 2024-06-01  |              749222 |              25000
            440 |                 625 | 2023-01-01    | 2024-06-01  |              977891 |              25000
            482 |                  96 | 2023-01-01    | 2024-06-01  |              256067 |               7500
            483 |                  96 | 2023-01-01    | 2024-06-01  |              828854 |               7500
            587 |                 447 | 2023-01-01    | 2024-06-01  |              633101 |              20000
            588 |                 447 | 2023-01-01    | 2024-06-01  |              931089 |              20000
            621 |                  66 | 2023-01-01    | 2024-06-01  |              106586 |               7000
            622 |                  66 | 2023-01-01    | 2024-06-01  |              820857 |               7000
            678 |                 541 | 2023-01-01    | 2024-06-01  |              674839 |              18000
            679 |                 541 | 2023-01-01    | 2024-06-01  |              977890 |              18000
(18 строк)
```

```
postgres=# CALL increase_scholarship_for_excellent_students();
CALL
postgres=# SELECT * FROM "Session".scholarship;
 scholarship_id | scholarship_type_id | payment_start | payment_end | studying_student_id | sum_of_scholarship
----------------+---------------------+---------------+-------------+---------------------+--------------------
            141 |                 453 | 2023-01-01    | 2024-06-01  |              648116 |              16500
            142 |                 453 | 2023-01-01    | 2024-06-01  |              977888 |              16500
            179 |                  94 | 2023-01-01    | 2024-06-01  |              254999 |              18150
            180 |                  94 | 2023-01-01    | 2024-06-01  |              825344 |              18150
            337 |                 108 | 2023-01-01    | 2024-06-01  |              460221 |              14850
            338 |                 108 | 2023-01-01    | 2024-06-01  |              887161 |              14850
            413 |                 349 | 2023-01-01    | 2024-06-01  |              582963 |               4400
            414 |                 349 | 2023-01-01    | 2024-06-01  |              916338 |               4400
            439 |                 625 | 2023-01-01    | 2024-06-01  |              749222 |              27500
            440 |                 625 | 2023-01-01    | 2024-06-01  |              977891 |              27500
            482 |                  96 | 2023-01-01    | 2024-06-01  |              256067 |               8250
            483 |                  96 | 2023-01-01    | 2024-06-01  |              828854 |               8250
            587 |                 447 | 2023-01-01    | 2024-06-01  |              633101 |              22000
            588 |                 447 | 2023-01-01    | 2024-06-01  |              931089 |              22000
            621 |                  66 | 2023-01-01    | 2024-06-01  |              106586 |               7700
            622 |                  66 | 2023-01-01    | 2024-06-01  |              820857 |               7700
            678 |                 541 | 2023-01-01    | 2024-06-01  |              674839 |              19800
            679 |                 541 | 2023-01-01    | 2024-06-01  |              977890 |              19800
(18 строк)
```

2. Для перевода студентов на следующий курс.

```
postgres=# CREATE OR REPLACE PROCEDURE move_students_to_next_course()
postgres-# AS $$
postgres$# BEGIN
postgres$#     UPDATE "Session"."studying student"
postgres$#     SET status = 'Выпускник'
postgres$#     FROM "Session".student
postgres$#     WHERE "Session".student.grade_book_id = "Session"."studying student".grade_book_id
postgres$#     AND "Session".student.course = 5;
postgres$#
postgres$#     UPDATE "Session".student
postgres$#     SET course = course + 1
postgres$#     WHERE course < 5;
postgres$#
postgres$# END;
postgres$# $$ LANGUAGE plpgsql;
CREATE PROCEDURE
```

```
postgres=# SELECT "Session"."studying student".studying_student_id, "Session".student.course, "Session"."studying student".status
postgres-# FROM "Session".student
postgres-# JOIN "Session"."studying student" ON "Session".student.grade_book_id = "Session"."studying student".grade_book_id;
 studying_student_id | course |  status
---------------------+--------+----------
              867159 |      5 | Обучается
              648114 |      4 | Обучается
              106586 |      5 | Обучается
              825342 |      3 | Обучается
              749222 |      3 | Обучается
              820856 |      5 | Обучается
              633100 |      4 | Обучается
              460218 |      4 | Обучается
              582963 |      3 | Обучается
              563063 |      3 | Обучается
              674836 |      5 | Обучается
              254999 |      3 | Обучается
              977888 |      4 | Обучается
              931087 |      3 | Обучается
              828851 |      3 | Обучается
              887161 |      5 | Обучается
              256066 |      2 | Обучается
              916337 |      2 | Обучается
              389660 |      3 | Обучается
              675687 |      2 | Обучается
              867160 |      5 | Обучается
              648115 |      5 | Обучается
              106587 |      2 | Обучается
              825343 |      2 | Обучается
              749223 |      5 | Обучается
              820857 |      2 | Обучается
              633101 |      3 | Обучается
              460219 |      4 | Обучается
              582964 |      5 | Обучается
              563064 |      3 | Обучается
              674837 |      4 | Обучается
              255000 |      5 | Обучается
              977889 |      3 | Обучается
              931088 |      3 | Обучается
              828852 |      4 | Обучается
              887162 |      5 | Обучается
              256067 |      5 | Обучается
              916338 |      2 | Обучается
              389661 |      3 | Обучается
              675688 |      5 | Обучается
              867161 |      3 | Обучается
              648116 |      4 | Обучается
              106588 |      5 | Обучается
              825344 |      2 | Обучается
              749224 |      5 | Обучается
              820858 |      4 | Обучается
              633102 |      4 | Обучается
              460220 |      4 | Обучается
              582965 |      2 | Обучается
              563065 |      3 | Обучается
              674838 |      4 | Обучается
              255001 |      2 | Обучается
              977890 |      5 | Обучается
              931089 |      5 | Обучается
              828853 |      3 | Обучается
              887163 |      5 | Обучается
              256068 |      2 | Обучается
              916339 |      4 | Обучается
              389662 |      5 | Обучается
              675689 |      2 | Обучается
              867162 |      2 | Обучается
              648117 |      4 | Обучается
              106589 |      5 | Обучается
              825345 |      5 | Обучается
              749225 |      5 | Обучается
              820859 |      5 | Обучается
              633103 |      2 | Обучается
              460221 |      2 | Обучается
              582966 |      5 | Обучается
              563066 |      5 | Обучается
              255002 |      3 | Обучается
              977891 |      3 | Обучается
              931090 |      4 | Обучается
              828854 |      5 | Обучается
              887164 |      4 | Обучается
              256069 |      2 | Обучается
              916340 |      2 | Обучается
              389663 |      5 | Обучается
              674839 |      5 | Обучается
              675686 |      2 | Обучается
(80 строк)
```

```
postgres=# CALL move_students_to_next_course();
CALL
postgres=# SELECT "Session"."studying student".studying_student_id, "Session".student.course, "Session"."studying student".status
postgres-# FROM "Session".student
postgres-# JOIN "Session"."studying student" ON "Session".student.grade_book_id = "Session"."studying student".grade_book_id;
 studying_student_id | course |  status
---------------------+--------+----------
              648114 |      5 | Обучается
              825342 |      4 | Обучается
              749222 |      4 | Обучается
              633100 |      5 | Обучается
              460218 |      5 | Обучается
              582963 |      4 | Обучается
              563063 |      4 | Обучается
              254999 |      4 | Обучается
              977888 |      5 | Обучается
              931087 |      4 | Обучается
              828851 |      4 | Обучается
              256066 |      3 | Обучается
              916337 |      3 | Обучается
              389660 |      4 | Обучается
              675687 |      3 | Обучается
              106587 |      3 | Обучается
              825343 |      3 | Обучается
              820857 |      3 | Обучается
              633101 |      4 | Обучается
              460219 |      5 | Обучается
              563064 |      4 | Обучается
              674837 |      5 | Обучается
              977889 |      4 | Обучается
              931088 |      4 | Обучается
              828852 |      5 | Обучается
              916338 |      3 | Обучается
              389661 |      4 | Обучается
              867161 |      4 | Обучается
              648116 |      5 | Обучается
              825344 |      3 | Обучается
              820858 |      5 | Обучается
              633102 |      5 | Обучается
              460220 |      5 | Обучается
              582965 |      3 | Обучается
              563065 |      4 | Обучается
              674838 |      5 | Обучается
              255001 |      3 | Обучается
              828853 |      4 | Обучается
              256068 |      3 | Обучается
              916339 |      5 | Обучается
              675689 |      3 | Обучается
              867162 |      3 | Обучается
              648117 |      5 | Обучается
              633103 |      3 | Обучается
              460221 |      3 | Обучается
              255002 |      4 | Обучается
              977891 |      4 | Обучается
              931090 |      5 | Обучается
              887164 |      5 | Обучается
              256069 |      3 | Обучается
              916340 |      3 | Обучается
              675686 |      3 | Обучается
              867159 |      5 | Выпускник
              106586 |      5 | Выпускник
              820856 |      5 | Выпускник
              674836 |      5 | Выпускник
              887161 |      5 | Выпускник
              867160 |      5 | Выпускник
              648115 |      5 | Выпускник
              749223 |      5 | Выпускник
              582964 |      5 | Выпускник
              255000 |      5 | Выпускник
              887162 |      5 | Выпускник
              256067 |      5 | Выпускник
              675688 |      5 | Выпускник
              106588 |      5 | Выпускник
              749224 |      5 | Выпускник
              977890 |      5 | Выпускник
              931089 |      5 | Выпускник
              887163 |      5 | Выпускник
              389662 |      5 | Выпускник
              106589 |      5 | Выпускник
              825345 |      5 | Выпускник
              749225 |      5 | Выпускник
              820859 |      5 | Выпускник
              582966 |      5 | Выпускник
              563066 |      5 | Выпускник
              828854 |      5 | Выпускник
              389663 |      5 | Выпускник
              674839 |      5 | Выпускник
(80 строк)
```

3. Для изменения оценки при успешной пересдаче экзамена

```
postgres=# CREATE OR REPLACE PROCEDURE update_exam_mark(
postgres(#     IN attestation_id_update integer,
postgres(#     IN student_id_to_update integer,
postgres(#     IN new_mark character(1),
postgres(#     IN exam_date_update date
postgres(# )
postgres-# LANGUAGE plpgsql
postgres-# AS $$
postgres$# BEGIN
postgres$#     UPDATE "Session".attestation
postgres$#     SET mark = CASE
postgres$#         WHEN "Session".attestation.attestation_id = attestation_id_update
postgres$#             AND "Session".attestation.studying_student_id = student_id_to_update
postgres$#             AND new_mark < "Session".attestation.mark THEN new_mark
postgres$#         ELSE "Session".attestation.mark
postgres$#     END,
postgres$#     attestation_date = CASE
postgres$#         WHEN "Session".attestation.attestation_id = attestation_id_update
postgres$#             AND "Session".attestation.studying_student_id = student_id_to_update THEN exam_date_update
postgres$#         ELSE "Session".attestation.attestation_date
postgres$#     END,
postgres$#     attempt = CASE
postgres$#         WHEN "Session".attestation.attestation_id = attestation_id_update
postgres$#             AND "Session".attestation.studying_student_id = student_id_to_update THEN "Session".attestation.attempt + 1
postgres$#         ELSE "Session".attestation.attempt
postgres$#     END
postgres$#     WHERE
postgres$#         "Session".attestation.attestation_id = attestation_id_update
postgres$#         AND "Session".attestation.studying_student_id = student_id_to_update;
postgres$# END;
postgres$# $$;
CREATE PROCEDURE
```

```
postgres=# SELECT * FROM "Session".attestation WHERE attestation_id = 211;
 attestation_id | discipline_in_syllabus_id | professor_id | studying_student_id | mark | attempt | attestation_date
----------------+---------------------------+--------------+---------------------+------+---------+------------------
            211 |                       826 |          836 |              828852 | C    |       1 | 2024-01-14
(1 строка)


postgres=# CALL update_exam_mark(211, 828852, 'A', '2024-01-15');
CALL
postgres=# SELECT * FROM "Session".attestation WHERE attestation_id = 211;
 attestation_id | discipline_in_syllabus_id | professor_id | studying_student_id | mark | attempt | attestation_date
----------------+---------------------------+--------------+---------------------+------+---------+------------------
            211 |                       826 |          836 |              828852 | A    |       2 | 2024-01-15
(1 строка)
```

Модифицировать триггер

```
postgres=# CREATE OR REPLACE FUNCTION fn_check_time_punch() RETURNS TRIGGER AS $$
postgres$# DECLARE
postgres$#     last_action BOOLEAN;
postgres$# BEGIN
postgres$#     -- Проверка на существование сотрудника
postgres$#     IF NOT EXISTS (
postgres$#         SELECT 1
postgres$#         FROM employee
postgres$#         WHERE id = NEW.employee_id
postgres$#     ) THEN
postgres$#         RETURN NULL;
postgres$#     END IF;
postgres$#
postgres$#     -- Проверка на наличие предыдущего действия для сотрудника
postgres$#     SELECT is_out_punch INTO last_action
postgres$#     FROM time_punch
postgres$#     WHERE employee_id = NEW.employee_id
postgres$#     ORDER BY punch_time DESC, id DESC
postgres$#     LIMIT 1;
postgres$#
postgres$#     -- Проверка на отсутствие предыдущего входа перед выходом
postgres$#     IF last_action IS NULL AND NEW.is_out_punch THEN
postgres$#         RETURN NULL;
postgres$#     END IF;
postgres$#
postgres$#     -- Проверка на повторный выход
postgres$#     IF last_action = NEW.is_out_punch THEN
postgres$#         RETURN NULL;
postgres$#     END IF;
postgres$#
postgres$#     -- Проверка на попытку выхода до времени входа
postgres$#     IF NEW.is_out_punch THEN
postgres$#         IF EXISTS (
postgres$#             SELECT 1
postgres$#             FROM time_punch tps
postgres$#             WHERE tps.employee_id = NEW.employee_id AND tps.is_out_punch = FALSE
postgres$#             ORDER BY tps.punch_time DESC, tps.id DESC
postgres$#             LIMIT 1
postgres$#         ) THEN
postgres$#             IF NEW.punch_time <= (
postgres$#                 SELECT tps.punch_time
postgres$#                 FROM time_punch tps
postgres$#                 WHERE tps.employee_id = NEW.employee_id AND tps.is_out_punch = FALSE
postgres$#                 ORDER BY tps.punch_time DESC, tps.id DESC
postgres$#                 LIMIT 1
postgres$#             ) THEN
postgres$#                 RETURN NULL;
postgres$#             END IF;
postgres$#         END IF;
postgres$#     END IF;
postgres$#
postgres$#     RETURN NEW;
postgres$# END;
postgres$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
postgres=#
postgres=# CREATE TRIGGER check_time_punch BEFORE INSERT ON time_punch
postgres-# FOR EACH ROW EXECUTE FUNCTION fn_check_time_punch();
CREATE TRIGGER
```

```
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
```

```
postgres=# SELECT * FROM time_punch WHERE employee_id = 4;
 id | employee_id | is_out_punch |     punch_time
----+-------------+--------------+---------------------
 19 |           4 | f            | 2023-01-01 12:00:00
 20 |           4 | t            | 2023-01-01 13:00:00
 21 |           4 | f            | 2023-01-01 15:00:00
(3 строки)
```

Вывод

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