Санкт-Петербургский национальный исследовательский университет ИТМО

Факультет инфокоммуникационных технологий

Лабораторная работа №3

«Процедуры, функции, триггеры в PostgreSQL»

по дисциплине «Проектирование и реализация баз данных»

Автор:

Ивенкова Е.Д.

группа K32422

Преподаватель:

Говорова М.М.

Санкт-Петербург

2023

Цель работы: овладеть практическими создания и использования процедур, функций и триггеров в базе данных PostgreSQL.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL, SQL Shell (psql).

Практическое задание:

Вариант 1

1. Создать процедуры/функции согласно индивидуальному заданию и (согласно индивидуальному заданию, часть 4).

2. Создать триггер для логирования событий вставки, удаления, редактирования данных в базе данных PostgreSQL (согласно индивидуальному заданию, часть 5). Допустимо создать универсальный триггер или отдельные триггеры на логирование действий.

Вариант 2

1. Создать процедуры/функции согласно индивидуальному заданию и (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.

Указание. Работа выполняется в консоли SQL Shell (psql).

Индивидуальное задание:

Вариант 2. БД «Сессия»

Задание 4. Создайте хранимые процедуры:

1. Для повышения стипендии отличникам на 10%.

2. Для перевода студентов на следующий курс.

3. Для изменения оценки при успешной пересдаче экзамена.

Задание 5. Создайте необходимые триггеры.

Работа выполнена **в варианте 1.**

Выполнение:

Процедуры и функции

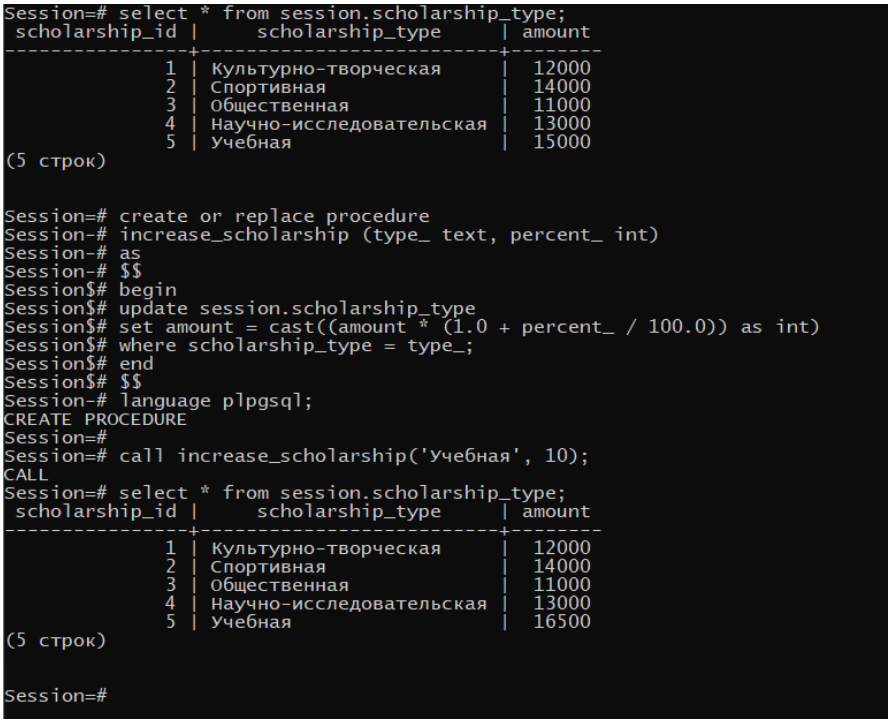
1. Для повышения стипендии отличникам на 10%.

Увеличить стипендию на 10% отличникам – значит увеличить на 10% стипендию типа «Учебная» из таблицы scholarship_type, которая выдается по результатам сессии.

Процедура:

```
create or replace procedure
increase_scholarship (type_ text, percent_ int)
as
$$
begin
	update session.scholarship_type
	set amount = cast((amount * (1.0 + percent_ / 100.0)) as int)
	where scholarship_type = type_;
end
$$
language plpgsql;
```

Результат:



2. Для перевода студентов на следующий курс.

Перевести студента на следующий курс значит перевести его из весеннего семестра в осенний семестр того же года с измененным названием группы, соответствующим новому курсу (+1). Не рассматриваются крайние случаи вроде «перевод из группы в группу», «перевод с потерей курса» и т.п**.**

Для перевода нужно добавить новую запись в таблицу student_in_group, кроме того, если новая группа, в которую нужно перевести студента, еще не существует, нужно в таблицу group добавить эту самую группу. Для вызова процедуры перевода необходимо в аргументе процедуры передать id студента, учащегося в определенной группе на данный момент (student_in_group_id).

Процедура:

```
create or replace procedure
next_course(st_in_group_id int)
as
$$
declare
sem_number int;
year_ int;
old_group_id int;
new_group_id int;
name_ character varying;
new_name_ character varying;
st_id int;
begin
select semester, study_year,
      group_name
into sem_number, year_,
      name_
from session.student_in_group
inner join session.group on
      session.student_in_group.group_id = session.group.group_id
where student_in_group_id = st_in_group_id;

-- проверка на семестр
if sem_number % 2 = 1
then
      raise exception
      'Transfer to a new course only available in spring semester';
end if;

--формирование нового названия группы
new_name_ = concat(
      substring(name_, 1, 2),
      cast((cast(substring(name_, 3, 1) as int) + 1) as text),
      substring(name_, 4, 2)
);

--существует ли уже эта группа?
```

```
if exists (select group_id
               from session.group
               where study_year = year_
               and group_name = new_name_
               and semester = sem_number + 1
               limit 1)
then
      select group_id into old_group_id
               from session.group
               where study_year = year_
               and group_name = new_name_
               and semester = sem_number + 1
               limit 1;
      new_group_id = old_group_id;
else
      -- создать группу


      insert into session.group
      (study_year, group_name, semester, edu_plan_id)
      select
            study_year, new_name_, sem_number + 1, edu_plan_id
      from session.group
      where group_id = (select group_id
                              from session.student_in_group
                              where student_in_group_id =
                                   st_in_group_id
                              limit 1);

      select group_id into new_group_id
      from session.group
      where
            semester = sem_number + 1
            and study_year = year_
            and group_name = new_name_
      limit 1;

end if;

-- запомним айдишник
select student_id into st_id from
session.student_in_group
where student_in_group_id =
      st_in_group_id;

--если студента еще не перевели на след курс:
if not exists (select student_id from
                     session.student_in_group
                     where group_id = new_group_id
                     and student_id = st_id
                     )
then
      insert into session.student_in_group
      (start_date, end_date, student_in_group_id,
       student_id, status, group_id)

      select
```

```
            cast (concat((cast(year_ as text)), '-09-01') as date),
            cast (concat((cast(year_ as text)), '-12-31') as date),
            max(student_in_group_id) + 1,
            st_id,
            'Обучается',
            new_group_id
      from session.student_in_group;

end if;

end;
$$ language plpgsql;
```

Реализация:

Создание процедуры

```
Session$# --если студента еще не перевели на след курс:
Session$# if not exists (select student_id from
Session$#   session.student_in_group
Session$#   where group_id = new_group_id
Session$#   and student_id = st_id
Session$#   )
Session$# then
Session$# insert into session.student_in_group
Session$# (start_date, end_date, student_in_group_id,
Session$#  student_id, status, group_id)
Session$#
Session$# select
Session$# cast (concat((cast(year_ as text)), '-09-01') as date),
Session$# cast (concat((cast(year_ as text)), '-12-31') as date),
Session$# max(student_in_group_id) + 1,
Session$# st_id,
Session$# 'Обучается',
Session$# new_group_id
Session$# from session.student_in_group;
Session$#
Session$# end if;
Session$#
Session$# end;
Session$# $$ language plpgsql;
CREATE PROCEDURE
```

Вызов процедуры для студента, о котором заведомо известно, что он обучается в 1 семестре:

```
Session=# call next_course(2);
ОШИБКА:  Transfer to a new course only available in spring semester
КОНТЕКСТ:  функция PL/pgSQL next_course(integer), строка 23, оператор RAISE
Session=#
```

Переведем на следующий курс (5) учащегося с student_in_group_id = 540:

```
Session=# select * from session.student_in_group where group_id = 24;
 start_date |  end_date  | student_id | student_in_group_id |  status   | group_id
------------+------------+------------+---------------------+-----------+----------
 2017-01-01 | 2017-08-31 |          2 |                 517 | Обучается |       24
 2017-01-01 | 2017-08-31 |         11 |                 518 | Академ    |       24
 2017-01-01 | 2017-08-31 |         12 |                 519 | Академ    |       24
 2017-01-01 | 2017-08-31 |         13 |                 520 | Академ    |       24
 2017-01-01 | 2017-08-31 |         17 |                 521 | Обучается |       24
 2017-01-01 | 2017-08-31 |         21 |                 522 | Обучается |       24
 2017-01-01 | 2017-08-31 |         30 |                 523 | Обучается |       24
 2017-01-01 | 2017-08-31 |         33 |                 524 | Академ    |       24
 2017-01-01 | 2017-08-31 |         35 |                 525 | Обучается |       24
 2017-01-01 | 2017-08-31 |         43 |                 526 | Обучается |       24
 2017-01-01 | 2017-08-31 |         47 |                 527 | Обучается |       24
 2017-01-01 | 2017-08-31 |         54 |                 528 | Академ    |       24
 2017-01-01 | 2017-08-31 |         60 |                 529 | Академ    |       24
 2017-01-01 | 2017-08-31 |         61 |                 530 | Академ    |       24
 2017-01-01 | 2017-08-31 |         63 |                 531 | Обучается |       24
 2017-01-01 | 2017-08-31 |         65 |                 532 | Обучается |       24
 2017-01-01 | 2017-08-31 |         73 |                 533 | Обучается |       24
 2017-01-01 | 2017-08-31 |         76 |                 534 | Обучается |       24
 2017-01-01 | 2017-08-31 |         80 |                 535 | Обучается |       24
 2017-01-01 | 2017-08-31 |         81 |                 536 | Академ    |       24
 2017-01-01 | 2017-08-31 |         83 |                 537 | Обучается |       24
 2017-01-01 | 2017-08-31 |         88 |                 538 | Обучается |       24
 2017-01-01 | 2017-08-31 |         93 |                 539 | Академ    |       24
 2017-01-01 | 2017-08-31 |         96 |                 540 | Обучается |       24
(24 строки)
```

Появилась соответствующая запись как в исходной таблице, так и в таблице с группами:

```
Session=# call next_course(540);
CALL
Session=# select * from session.group offset 20;
 group_id | study_year | group_name | semester | edu_plan_id
----------+------------+------------+----------+-------------
       21 |       2019 | К3318      |        5 |           9
       22 |       2020 | К3318      |        6 |           9
       23 |       2020 | К3418      |        7 |           9
       24 |       2021 | К3418      |        8 |           9
       25 |       2017 | V3120      |        1 |           5
       26 |       2018 | V3120      |        2 |           5
       27 |       2018 | V3220      |        3 |           5
       28 |       2019 | v3220      |        4 |           5
       29 |       2019 | V3320      |        5 |           5
       30 |       2020 | V3320      |        6 |           5
       31 |       2020 | V3420      |        7 |           5
       32 |       2021 | V3420      |        8 |           5
       38 |       2021 | V3520      |        9 |           5
       39 |       2021 | К3518      |        9 |           9
(14 строк)
```

```
Session=# select * from session.student_in_group offset 700;
 start_date |  end_date  | student_id | student_in_group_id |  status   | group_id
------------+------------+------------+---------------------+-----------+----------
 2017-01-01 | 2017-08-31 |         65 |                 701 | Обучается |       32
 2017-01-01 | 2017-08-31 |         73 |                 702 | Академ    |       32
 2017-01-01 | 2017-08-31 |         74 |                 703 | Обучается |       32
 2017-01-01 | 2017-08-31 |         78 |                 704 | Академ    |       32
 2017-01-01 | 2017-08-31 |         80 |                 705 | Академ    |       32
 2017-01-01 | 2017-08-31 |         82 |                 706 | Обучается |       32
 2017-01-01 | 2017-08-31 |         84 |                 707 | Академ    |       32
 2017-01-01 | 2017-08-31 |         87 |                 708 | Академ    |       32
 2017-01-01 | 2017-08-31 |         90 |                 709 | Обучается |       32
 2017-01-01 | 2017-08-31 |         91 |                 710 | Академ    |       32
 2017-01-01 | 2017-08-31 |         95 |                 711 | Обучается |       32
 2017-01-01 | 2017-08-31 |         98 |                 712 | Академ    |       32
 2021-09-01 | 2021-12-31 |         98 |                 713 | Обучается |       38
 2021-09-01 | 2021-12-31 |         61 |                 714 | Обучается |       38
 2021-09-01 | 2021-12-31 |         96 |                 715 | Обучается |       39
(15 строк)
```

Попробуем перевести с 2 на 3 курс студента, у которого уже есть запись об обучении на 3 курсе: как видно, общее количество записей в таблицах не изменилось.

```
Session=# select * from session.student_in_group where group_id = 20;
 start_date |  end_date  | student_id | student_in_group_id |  status   | group_id
------------+------------+------------+---------------------+-----------+----------
 2019-01-01 | 2019-08-31 |          2 |                 429 | Обучается |       20
 2019-01-01 | 2019-08-31 |          9 |                 430 | Обучается |       20
 2019-01-01 | 2019-08-31 |         15 |                 431 | Академ    |       20
 2019-01-01 | 2019-08-31 |         19 |                 432 | Обучается |       20
 2019-01-01 | 2019-08-31 |         25 |                 433 | Академ    |       20
 2019-01-01 | 2019-08-31 |         29 |                 434 | Обучается |       20
 2019-01-01 | 2019-08-31 |         32 |                 435 | Обучается |       20
 2019-01-01 | 2019-08-31 |         34 |                 436 | Обучается |       20
 2019-01-01 | 2019-08-31 |         36 |                 437 | Академ    |       20
 2019-01-01 | 2019-08-31 |         45 |                 438 | Обучается |       20
 2019-01-01 | 2019-08-31 |         47 |                 439 | Обучается |       20
 2019-01-01 | 2019-08-31 |         54 |                 440 | Академ    |       20
 2019-01-01 | 2019-08-31 |         57 |                 441 | Академ    |       20
 2019-01-01 | 2019-08-31 |         60 |                 442 | Академ    |       20
 2019-01-01 | 2019-08-31 |         66 |                 443 | Академ    |       20
 2019-01-01 | 2019-08-31 |         71 |                 444 | Обучается |       20
 2019-01-01 | 2019-08-31 |         72 |                 445 | Академ    |       20
 2019-01-01 | 2019-08-31 |         74 |                 446 | Академ    |       20
 2019-01-01 | 2019-08-31 |         76 |                 447 | Академ    |       20
 2019-01-01 | 2019-08-31 |         82 |                 448 | Обучается |       20
 2019-01-01 | 2019-08-31 |         83 |                 449 | Академ    |       20
 2019-01-01 | 2019-08-31 |         89 |                 450 | Обучается |       20
 2019-01-01 | 2019-08-31 |         96 |                 451 | Академ    |       20
(23 строки)


Session=# select count(*) from session.group;
 count
-------
    34
(1 строка)


Session=# select count(*) from session.student_in_group;
 count
-------
   715
(1 строка)


Session=# call next_course(444);
CALL
Session=# select count(*) from session.student_in_group;
 count
-------
   715
(1 строка)


Session=# select count(*) from session.group;
 count
-------
    34
(1 строка)
```

3. Для изменения оценки при успешной пересдаче экзамена.

Пересдача экзамена – это добавление в таблицу attestation записи о сдаче экзамена с указанием номера попытки на 1 больше предыдущей попытки. Не имеет значения, на какой балл был сдан экзамен (т.е. смог ли студент перейти порог в 60 баллов, необходимые для зачета), единственное условие – оценка за новую попытку должна быть больше или равна предыдущей. Также важно, чтобы соответствующей записи о новой попытке сдачи еще не существовало в таблице. *Под условие поставленной задачи не попадает ситуация «изменить значение оценки, выставленное за попытку ошибочно».*

Процедура:

```
create or replace procedure add_new_grade(st_in_group_id int, new_grade
int, subject int)
as
$$
declare
current_grade int;
new_attempt_count int;
begin

select max(attempt_count) + 1 into new_attempt_count
from session.attestation
where student_in_group_id = st_in_group_id
and edu_plan_subject_id = subject;

-- проверка на органичение: максимум 3 попытки
if new_attempt_count > 3
then
      raise exception
            'Student can''t get more than 3 attempts to get a grade.';
end if;

-- проверка: если запись о перепрохождении уже существует
-- согласно архитектуре процедуры, этот экспешен никогда не появится..
if exists (select attestation_id
              from session.attestation
              where student_in_group_id = st_in_group_id
              and attempt_count = new_attempt_count
              and edu_plan_subject_id = subject)
then
      raise exception
            'Replacing grade for a completed exam is for admin only.';
end if;

select grade into current_grade
from session.attestation
where student_in_group_id = st_in_group_id
and attempt_count = new_attempt_count - 1
and edu_plan_subject_id = subject;
```

```
	-- нельзя новой аттестацией занизить уже имеющуюся оценку
	if current_grade > new_grade
	then
		raise exception
			'New attempt is not to decrease student''s total points.';
	end if;

	insert into session.attestation
	(grade, attestation_date, teacher_id, edu_plan_subject_id,
student_in_group_id, attempt_count)

	select
	new_grade,
	attestation_date + 1,
	teacher_id,
	edu_plan_subject_id,
	student_in_group_id,
	new_attempt_count
	from session.attestation
	where student_in_group_id = st_in_group_id
	and attempt_count = new_attempt_count - 1
	and edu_plan_subject_id = subject;
	end;
	$$ language plpgsql;
```

```
Session$# end if;
Session$#
Session$# insert into session.attestation
Session$# (grade, attestation_date, teacher_id, edu_plan_subject_id, student_in_group_id, attempt_count)
Session$#
Session$# select
Session$# new_grade,
Session$# attestation_date + 1,
Session$# teacher_id,
Session$# edu_plan_subject_id,
Session$# student_in_group_id,
Session$# new_attempt_count
Session$# from session.attestation
Session$# where student_in_group_id = st_in_group_id
Session$# and attempt_count = new_attempt_count - 1
Session$# and edu_plan_subject_id = subject;
Session$# end;
Session$# $$ language plpgsql;
CREATE PROCEDURE
```

Реализация:

```
Session=# select * from session.attestation limit 10;
 grade | attestation_date | teacher_id | edu_plan_subject_id | student_in_group_id | attempt_count | attestation_id
-------+------------------+------------+---------------------+---------------------+---------------+----------------
    91 | 2017-09-26       |          1 |                  41 |                   1 |             1 |              1
    93 | 2017-09-17       |         20 |                  18 |                   1 |             1 |              2
    70 | 2017-09-23       |         22 |                  15 |                   1 |             1 |              3
    59 | 2017-09-26       |          1 |                  41 |                   2 |             1 |              4
    76 | 2017-09-26       |          1 |                  41 |                   2 |             2 |              5
    73 | 2017-09-17       |         20 |                  18 |                   2 |             1 |              6
    48 | 2017-09-23       |         22 |                  15 |                   2 |             1 |              7
    64 | 2017-09-23       |         22 |                  15 |                   2 |             2 |              8
    79 | 2017-09-26       |          1 |                  41 |                   3 |             1 |              9
    81 | 2017-09-17       |         20 |                  18 |                   3 |             1 |             10
(10 строк)
```

Попробуем студенту с student_in_group_id = 1 поставить оценку 90 за экзамен по предмету 41:

```
Session=# call add_new_grade(1, 90, 41);
ОШИБКА:  New attempt is not to decrease student's total points.
КОНТЕКСТ:  функция PL/pgSQL add_new_grade(integer,integer,integer), строка 41, оператор RAISE
Session=#
```

Попробуем увеличить его оценку за экзамен (это было успешно):

```
Session=#
Session=#  call add_new_grade(1, 97, 41);
CALL
Session=#  select * from session.attestation where student_in_group_id = 1 and edu_plan_subject_id = 41;
 grade | attestation_date | teacher_id | edu_plan_subject_id | student_in_group_id | attempt_count | attestation_id
-------+------------------+------------+---------------------+---------------------+---------------+----------------
    91 | 2017-09-26       |          1 |                  41 |                   1 |             1 |              1
    97 | 2017-09-27       |          1 |                  41 |                   1 |             2 |           2448
(2 строки)
```

Найдем студента (11-ый) с 3 попытками сдать один предмет (15-ый) и попробуем дать ему новую оценку:

```
Session=# select * from session.attestation where attempt_count = 3 limit 10;
 grade | attestation_date | teacher_id | edu_plan_subject_id | student_in_group_id | attempt_count | attestation_id
-------+------------------+------------+---------------------+---------------------+---------------+----------------
    66 | 2017-09-26       |          1 |                  41 |                  10 |             3 |             42
    67 | 2017-09-23       |         22 |                  15 |                  11 |             3 |             50
    72 | 2017-09-23       |         22 |                  15 |                  15 |             3 |             66
    72 | 2017-09-23       |         22 |                  15 |                  22 |             3 |             94
    71 | 2018-01-22       |          9 |                   1 |                  26 |             3 |            114
    76 | 2018-01-20       |         24 |                  21 |                  26 |             3 |            118
    66 | 2018-01-22       |          8 |                  50 |                  38 |             3 |            173
    73 | 2018-09-18       |         20 |                  20 |                  55 |             3 |            256
    73 | 2018-09-17       |          2 |                  29 |                  55 |             3 |            259
    69 | 2019-01-23       |         11 |                  23 |                  71 |             3 |            328
(10 строк)


Session=# call add_new_grade(11, 69, 15);
ОШИБКА:  Student can't get more than 3 attempts to get a grade.
КОНТЕКСТ:  функция PL/pgSQL add_new_grade(integer,integer,integer), строка 15, оператор RAISE
```

Удалим оценку какого-нибудь студента, который сдал экзамен на 3 попытке меньше, чем на 60 баллов, и попробуем добавить ему другую запись о данной попытке:

```
Session=# select * from session.attestation where attempt_count = 3 and grade < 60 limit 10;
 grade | attestation_date | teacher_id | edu_plan_subject_id | student_in_group_id | attempt_count | attestation_id
-------+------------------+------------+---------------------+---------------------+---------------+----------------
    57 | 2019-09-20       |          4 |                   4 |                 100 |             3 |            451
    58 | 2018-09-18       |         12 |                  31 |                 408 |             3 |           1300
    56 | 2019-09-20       |         10 |                  23 |                 464 |             3 |           1500
    56 | 2017-09-30       |          3 |                  10 |                 546 |             3 |           1801
    55 | 2020-09-15       |         22 |                  41 |                 681 |             3 |           2345
(5 строк)


Session=# select * from session.attestation where student_in_group_id = 100 and edu_plan_subject_id = 4;
 grade | attestation_date | teacher_id | edu_plan_subject_id | student_in_group_id | attempt_count | attestation_id
-------+------------------+------------+---------------------+---------------------+---------------+----------------
    47 | 2019-09-20       |          4 |                   4 |                 100 |             1 |            449
    52 | 2019-09-20       |          4 |                   4 |                 100 |             2 |            450
    57 | 2019-09-20       |          4 |                   4 |                 100 |             3 |            451
(3 строки)


Session=# delete from session.attestation where student_in_group_id = 100 and edu_plan_subject_id = 4 and attempt_count
= 3;
DELETE 1
Session=#
Session=# select * from session.attestation where student_in_group_id = 100 and edu_plan_subject_id = 4;
 grade | attestation_date | teacher_id | edu_plan_subject_id | student_in_group_id | attempt_count | attestation_id
-------+------------------+------------+---------------------+---------------------+---------------+----------------
    47 | 2019-09-20       |          4 |                   4 |                 100 |             1 |            449
    52 | 2019-09-20       |          4 |                   4 |                 100 |             2 |            450
(2 строки)


Session=# call add_new_grade(100, 74, 4);
CALL
Session=# select * from session.attestation where student_in_group_id = 100 and edu_plan_subject_id = 4;
 grade | attestation_date | teacher_id | edu_plan_subject_id | student_in_group_id | attempt_count | attestation_id
-------+------------------+------------+---------------------+---------------------+---------------+----------------
    47 | 2019-09-20       |          4 |                   4 |                 100 |             1 |            449
    52 | 2019-09-20       |          4 |                   4 |                 100 |             2 |            450
    74 | 2019-09-21       |          4 |                   4 |                 100 |             3 |           2449
(3 строки)
```

Триггеры

Триггер для логирования событий (вставка, модификация, удаление) в таблице building (инфрмация о корпусах университета)

```
create type cud_operation as enum('update', 'insert', 'delete');

-- Создание таблицы для логов
create table session.building_logs(
    id_log serial primary key,
    operation_type cud_operation not null,
    operation_time timestamp without time zone,
    affected_building_id int null,
    affected_building_name text null
);

-- Создание метода для триггера
create or replace function building_add_to_log() returns trigger as $$
declare
    var_operation_type cud_operation;
begin
    if  tg_op = 'insert' then
        var_operation_type := 'insert';
        insert into
            session.building_logs(operation_type, operation_time,
affected_building_id, affected_building_name)
        values
            (var_operation_type, now(), new.building_id, new.building_name);
        return new;
    elsif tg_op = 'update' then
        var_operation_type := 'update';
        insert into
            session.building_logs(operation_type, operation_time,
affected_building_id, affected_building_name)
        values
            (var_operation_type, now(), old.building_id, old.building_name);
        return new;
    elsif tg_op = 'delete' then
        var_operation_type := 'delete';
        insert into
            session.building_logs(operation_type, operation_time,
affected_building_id, affected_building_name)
        values
            (var_operation_type, now(), old.building_id, old.building_name);
        return old;
    end if;
end
$$ language plpgsql;
```

```
create trigger log_trigger
    after insert or update or delete on session.building
    for each row execute procedure building_add_to_log();
```

```
Session=# create type cud_operation as enum('update', 'insert', 'delete');
CREATE TYPE
Session=#
Session=# -- Создание таблицы для логов
Session=# create table session.building_logs(
Session(#     id_log serial primary key,
Session(#     operation_type cud_operation not null,
Session(#     operation_time timestamp without time zone,
Session(#     affected_building_id int null,
Session(#     affected_building_name text null
Session(# );
CREATE TABLE
```

```
Session=# -- Создание метода для триггера
Session=# CREATE OR REPLACE FUNCTION building_add_to_log() RETURNS TRIGGER AS $$
Session$# DECLARE
Session$#     var_operation_type cud_operation;
Session$# BEGIN
Session$#     IF  TG_OP = 'INSERT' THEN
Session$#         var_operation_type := 'insert';
Session$#         insert into
Session$#             session.building_logs(operation_type, operation_time, affected_building_
id, affected_building_name)
Session$#         VALUES
Session$#             (var_operation_type, now(), new.building_id, new.building_name);
Session$#         RETURN NEW;
Session$#     ELSIF TG_OP = 'UPDATE' THEN
Session$#         var_operation_type := 'update';
Session$#         insert into
Session$#             session.building_logs(operation_type, operation_time, affected_building_
id, affected_building_name)
Session$#         VALUES
Session$#             (var_operation_type, now(), old.building_id, old.building_name);
Session$#         RETURN NEW;
Session$#     ELSIF TG_OP = 'DELETE' THEN
Session$#         var_operation_type := 'delete';
Session$#         insert into
Session$#             session.building_logs(operation_type, operation_time, affected_building_
id, affected_building_name)
Session$#         VALUES
Session$#             (var_operation_type, now(), old.building_id, old.building_name);
Session$#         RETURN OLD;
Session$#     END IF;
Session$# END
Session$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
Session=#
Session=#
Session=#
Session=# create trigger log_trigger
Session-#     after insert or update or delete on session.building
Session-#     for each row execute procedure building_add_to_log();
CREATE TRIGGER
```

Проверка:

```
Session=# select * from session.building;
 building_id |                address                 | building_name
-------------+----------------------------------------+---------------
           1 | Гороховая улица, 35-37                 | Корпус 1
           2 | Литейный проспект, 57                  | Корпус 2
           3 | Лесной проспект, 20к5                  | Корпус 3
           4 | 11-я линия Васильевского острова, 38   | Корпус 4
           5 | Гатчинская улица, 8                    | Корпус 5
           6 | улица Оптиков, 35к2                    | Корпус 6
(6 строк)
```

```
Session=# update session.building
Session-# set
Session-#     building_name = 'корпус228'
Session-# where building_id=5;
UPDATE 1
Session=# select * from session.building;
 building_id |               address                | building_name
-------------+--------------------------------------+---------------
           1 | Гороховая улица, 35-37               | Корпус 1
           2 | Литейный проспект, 57                | Корпус 2
           3 | Лесной проспект, 20к5                | Корпус 3
           4 | 11-я линия Васильевского острова, 38 | Корпус 4
           6 | улица Оптиков, 35к2                  | Корпус 6
           5 | Гатчинская улица, 8                  | корпус228
(6 строк)


Session=# insert into session.building(building_id, address, building_name) values
Session-#     (7, 'sample address', 'корпус30');
INSERT 0 1
Session=# select * from session.building;
 building_id |               address                | building_name
-------------+--------------------------------------+---------------
           1 | Гороховая улица, 35-37               | Корпус 1
           2 | Литейный проспект, 57                | Корпус 2
           3 | Лесной проспект, 20к5                | Корпус 3
           4 | 11-я линия Васильевского острова, 38 | Корпус 4
           6 | улица Оптиков, 35к2                  | Корпус 6
           5 | Гатчинская улица, 8                  | корпус228
           7 | sample address                       | корпус30
(7 строк)


Session=# delete from session.building where building_id=7;
DELETE 1
Session=# select * from session.building;
 building_id |               address                | building_name
-------------+--------------------------------------+---------------
           1 | Гороховая улица, 35-37               | Корпус 1
           2 | Литейный проспект, 57                | Корпус 2
           3 | Лесной проспект, 20к5                | Корпус 3
           4 | 11-я линия Васильевского острова, 38 | Корпус 4
           6 | улица Оптиков, 35к2                  | Корпус 6
           5 | Гатчинская улица, 8                  | корпус228
(6 строк)


Session=# select * from session.building_logs;
 id_log | operation_type |       operation_time       | affected_building_id | affected_buildi
ng_name
--------+----------------+----------------------------+----------------------+----------------
--------
      1 | update         | 2023-10-03 17:05:43.073776 |                    5 | корпус228
      2 | insert         | 2023-10-03 17:06:32.140539 |                    7 | корпус30
      3 | delete         | 2023-10-03 17:06:47.323604 |                    7 | корпус30
(3 строки)
```

Выводы:

В данной лабораторной работе (вариант 1) были созданы процедуры и функции согласно индивидуальному заданию (для повышения стипендии отличникам на 10%, для перевода студентов на следующий курс, для изменения оценки при успешной пересдаче экзамена), а также созданы триггеры на логирование таких действий в отношении базы данных, как вставка, удаление, редактирование данных.