Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Российский университет транспорта (МИИТ)» (РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

**О Т Ч Ё Т**

по лабораторной работе №2

по дисциплине:

«Информационное обеспечение систем управления»

на тему:

«Разработка запросов с помощью функций»

Выполнил: ст. гр. ВУЦ-421

Лихачев С. С.

Вариант 5

Проверил: доц., к.т.н.

Васильева М. А.

Москва ­2024

**Цель работы**

Разработать запросы с помощью функций на примере диалекта СУБД Postgres Pro.

**Результаты выполненной работы**

**Задание** **1**

Формулировка задания:

Функция принимает на вход уровень сложности маршрута и возвращает количество маршрутов с этим уровнем.

Текст скрипта:

**CREATE** **OR** **REPLACE** **FUNCTION** count\_trails\_by\_difficulty(difficulty\_name **VARCHAR**)

**RETURNS** **INT** **AS** **$$**

**DECLARE**

trail\_count **INT**;

**BEGIN**

**SELECT** **COUNT**(\*) **INTO** trail\_count

**FROM** trails t

**JOIN** difficulty d **ON** t.difficulty\_id = d.id

**WHERE** d.**name** = difficulty\_name;

**RETURN** trail\_count;

**END**;

**$$** **LANGUAGE** plpgsql;

**SELECT** count\_trails\_by\_difficulty(**'Средний'**);

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).



Рисунок 1 – Результат выполнения первого задания

**Задание** **2**

Формулировка задания:

Функция будет вычислять общую стоимость аренды снаряжения для заданного списка идентификаторов снаряжения. Здесь используется массив параметров, и функция производит арифметические вычисления.

Текст скрипта:

**CREATE** **OR** **REPLACE** **FUNCTION** calculate\_total\_rental\_cost(equipment\_ids **INT**[])

**RETURNS** **NUMERIC** **AS** **$$**

**DECLARE**

total\_cost **NUMERIC** := 0;

**BEGIN**

**SELECT** **SUM**(price) **INTO** total\_cost

**FROM** equipment

**WHERE** id = **ANY**(equipment\_ids);

**RETURN** total\_cost;

**END**;

**$$** **LANGUAGE** plpgsql;

**SELECT** calculate\_total\_rental\_cost(**ARRAY**[1, 2, 3]);

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).



Рисунок 2 – Результат выполнения второго задания

**Задание** **3**

Формулировка задания:

Функция принимает параметры нового маршрута и добавляет его в базу данных, если координаты старта и финиша указаны корректно.

Текст скрипта:

**CREATE** **OR** **REPLACE** **FUNCTION** add\_new\_trail(

trail\_name **VARCHAR**,

trail\_description **VARCHAR**,

trail\_length **DECIMAL**,

trail\_duration **DECIMAL**,

height\_diff **DECIMAL**,

water\_available **BOOLEAN**,

difficulty\_id **INT**,

trail\_type\_id **INT**,

start\_coords GEOGRAPHY,

end\_coords GEOGRAPHY

) **RETURNS** **TEXT** **AS** **$$**

**DECLARE**

new\_id **INT**;

**BEGIN**

-- Получаем следующее значение из последовательности для ID

**SELECT** **nextval**(**'sequence\_generator\_id'**) **INTO** new\_id;

-- Проверка координат

**IF** start\_coords **IS** **NULL** **OR** end\_coords **IS** **NULL** **THEN**

**RETURN** **'Ошибка: Координаты старта и финиша не могут быть пустыми.'**;

**END** **IF**;

-- Вставка нового маршрута

**INSERT** **INTO** trails (id, **name**, description, trail\_lenght, trail\_duration, height\_difference,

water\_avaibability, difficulty\_id, trail\_type\_id, start\_coordinates, end\_coordinates)

**VALUES** (new\_id, trail\_name, trail\_description, trail\_length, trail\_duration, height\_diff,

water\_available, difficulty\_id, trail\_type\_id, start\_coords, end\_coords);

**RETURN** **'Маршрут успешно добавлен. ID нового маршрута: '** || new\_id;

**END**;

**$$** **LANGUAGE** plpgsql;

**SELECT** add\_new\_trail(

**'Новый маршрут'**,

**'Описание маршрута'**,

12.5,

6.5,

500,

**TRUE**,

1,

2,

**'SRID=4326;POINT(55.7558 37.6173)'**,

**'SRID=4326;POINT(55.7558 37.6173)'**

);

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).



Рисунок 3 – Результат выполнения третьего задания

**Задание** **4**

Формулировка задания:

Функция принимает ID страны и возвращает среднюю длительность маршрутов, получая на вход информацию о сложности.

Текст скрипта:

**CREATE** **OR** **REPLACE** **FUNCTION** avg\_trail\_duration\_by\_difficulty(difficulty\_name **VARCHAR**)

**RETURNS** **NUMERIC** **AS** **$$**

**DECLARE**

avg\_duration **NUMERIC**;

**BEGIN**

**SELECT** **AVG**(t.trail\_duration) **INTO** avg\_duration

**FROM** trails t

**JOIN** difficulty d **ON** t.difficulty\_id = d.id

**WHERE** d.**name** = difficulty\_name;

**IF** avg\_duration **IS** **NULL** **THEN**

**RETURN** 0; -- Возвращает 0, если маршрутов с таким уровнем сложности нет.

**END** **IF**;

**RETURN** avg\_duration;

**END**;

**$$** **LANGUAGE** plpgsql;

-- Пример вызова:

**SELECT** avg\_trail\_duration\_by\_difficulty(**'Сложный'**);

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).



Рисунок 3 – Результат выполнения четвертого задания

**Вывод**

В процессе работы были разработаны запросы с помощью функций на примере диалекта СУБД Postgres Pro.