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 №3

по дисциплине:

«Информационное обеспечение систем управления»

на тему:

«Разработка запросов с помощью триггерных 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 триггерных функций на примере диалекта СУБД Postgres Pro.

**Результаты выполненной работы**

**Задание** **1**

Формулировка задания:

Триггер будет вызывать исключение, если удаляется маршрут, который используется в активных походах (статус "В процессе").

Текст скрипта:

**CREATE** **OR** **REPLACE** **FUNCTION** prevent\_trail\_deletion()

**RETURNS** **TRIGGER** **AS** **$$**

**BEGIN**

-- Проверяем, есть ли активные походы, связанные с маршрутом, который удаляется

**IF** **EXISTS** (

**SELECT** 1

**FROM** treks

**WHERE** trail\_id = **OLD**.id

**AND** trek\_status\_id = (**SELECT** id **FROM** trek\_status **WHERE** **name** = **'В процессе'**)

) **THEN**

**RAISE** **EXCEPTION** **'Невозможно удалить маршрут, связанный с активными походами.'**;

**END** **IF**;

**RETURN** **OLD**; -- Обязательное возвращение удаляемой строки

**END**;

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

**CREATE** **TRIGGER** before\_trail\_delete

**BEFORE** **DELETE** **ON** trails

**FOR** **EACH** **ROW**

**EXECUTE** **FUNCTION** prevent\_trail\_deletion();

-- Создаем маршрут

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

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

**VALUES** (999, **'Тестовый маршрут'**, **'Для проверки триггера'**, 10.0, 2.0, 100.0, **TRUE**, 1, 1,

**'POINT(55.7558 37.6173)'**, **'POINT(55.7558 37.6173)'**);

-- Создаем активный поход, связанный с этим маршрутом

**INSERT** **INTO** treks (id, start\_date, end\_date, distance, spend\_time, trek\_status\_id, trail\_id)

**VALUES** (999, current\_timestamp, **NULL**, 10.0, 3600, (**SELECT** id **FROM** trek\_status **WHERE** **name** = **'В процессе'**), 999);

-- Пытаемся удалить маршрут

**DELETE** **FROM** trails **WHERE** id = 999;

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

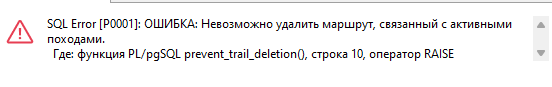


Рисунок 1 – Результат выполнения первого задания

**Задание** **2**

Формулировка задания:

Триггер автоматически проверяет, что координаты начала и конца маршрута указаны, иначе выбрасывает исключение.

Текст скрипта:

**CREATE** **OR** **REPLACE** **FUNCTION** validate\_trail\_coordinates()

**RETURNS** **TRIGGER** **AS** **$$**

**BEGIN**

**IF** **NEW**.start\_coordinates **IS** **NULL** **THEN**

**RAISE** **EXCEPTION** **'Координаты старта маршрута не могут быть пустыми.'**;

**END** **IF**;

**IF** **NEW**.end\_coordinates **IS** **NULL** **THEN**

**RAISE** **EXCEPTION** **'Координаты конца маршрута не могут быть пустыми.'**;

**END** **IF**;

**RETURN** **NEW**;

**END**;

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

**CREATE** **TRIGGER** before\_trail\_insert

**BEFORE** **INSERT** **ON** trails

**FOR** **EACH** **ROW**

**EXECUTE** **FUNCTION** validate\_trail\_coordinates();

-- Попытка добавить маршрут без координат

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

water\_avaibability, difficulty\_id, trail\_type\_id)

**VALUES** (1000, **'Маршрут без координат'**, **'Проверка триггера'**, 5.0, 1.5, 50.0, **FALSE**, 2, 2);

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).



Рисунок 2 – Результат выполнения второго задания

**Задание** **3**

Формулировка задания:

Триггер автоматически устанавливает текущую дату как время старта (start\_date) и добавляет 1 день для времени завершения (end\_date), если они не указаны.

Текст скрипта:

**CREATE** **OR** **REPLACE** **FUNCTION** set\_trek\_dates()

**RETURNS** **TRIGGER** **AS** **$$**

**BEGIN**

**IF** **NEW**.start\_date **IS** **NULL** **THEN**

**NEW**.start\_date = **current\_timestamp**;

**END** **IF**;

**IF** **NEW**.end\_date **IS** **NULL** **THEN**

**NEW**.end\_date = **NEW**.start\_date + **INTERVAL** **'1 day'**;

**END** **IF**;

**RETURN** **NEW**;

**END**;

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

**CREATE** **TRIGGER** before\_trek\_insert

**BEFORE** **INSERT** **ON** treks

**FOR** **EACH** **ROW**

**EXECUTE** **FUNCTION** set\_trek\_dates();

-- Добавление похода без указания времени

**INSERT** **INTO** treks (id, distance, spend\_time, trek\_status\_id, trail\_id)

**VALUES** (1001, 15.0, 7200, (**SELECT** id **FROM** trek\_status **WHERE** **name** = **'Планируется'**), 1);

-- Проверка данных в таблице treks

**SELECT** \* **FROM** treks **WHERE** id = 1001;

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).



Рисунок 3 – Результат выполнения третьего задания

Вывод: были разработаны запросы с помощью триггерных функций на примере диалекта СУБД Postgres Pro.