МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Звіт

про виконання лабораторної роботи №12

«Розробка та застосування тригерів»

з дисципліни «ОБДЗ»

Виконав:

студент групи КН-209

Ханас Михайло-Юрій

Викладач:

Мельникова Н.І.

Львів – 2020 р.

**Мета роботи:** Розробити SQL запити, які моделюють роботу тригерів: каскадне знищення, зміна та доповнення записів у зв’язаних таблицях.

**Хід роботи**

1. Створимо 2 таблиці(performance, auditorium) та заповнимо їх.

CREATE DATABASE theatr;

USE theatr;

CREATE TABLE auditorium

(

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(32) NOT NULL,

seats\_number INT NOT NULL,

count\_use INT NOT NULL

);

CREATE TABLE performance

(

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(32) NOT NULL,

price VARCHAR(32) NOT NULL,

duration INT NOT NULL ,

id\_auditorium INT NOT NULL ,

data\_performance TIMESTAMP NOT NULL,

CONSTRAINT fk\_performance\_auditorium

FOREIGN KEY (id\_auditorium)

REFERENCES auditorium (id)

ON DELETE CASCADE

);

INSERT INTO auditorium VALUES

(1,"Big auditorium", 240,1),

(2,"Small auditorium", 110,2),

(3,"Middle auditorium", 170,1);

INSERT INTO performance VALUES

(1, "Muna Mazailo", 200, 2, 1, "2020-03-04 10:10:00"),

(2, "Stolen happiness", 400, 3, 2, "2020-03-03 10:50:00"),

(3, "Coriolanus", 600, 3, 2, "2020-04-03 15:00:00"),

(4, "Fireplace master", 450, 2, 3, "2020-04-04 16:30:00");

1. Каскадне оновлення таблиці PERFORMANCE при видаленні AUDITORIUM з бази даних:

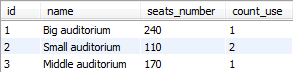
CREATE

TRIGGER auditorium\_delete BEFORE DELETE

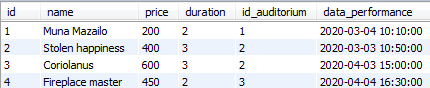
ON theatr.auditorium FOR EACH ROW

DELETE FROM performance WHERE id\_auditorium=OLD.id;

**До видалення аудиторії:**



Auditorium



Performance

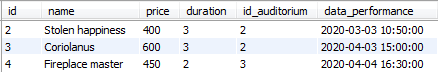
**Видаляємо аудиторію під id = 1:**

DELETE FROM auditorium WHERE auditorium.id = 1;

**Результат після видалення аудиторії:**



Auditorium



Performance

Отже, можна побачити, шо після видалення аудиторії під id=1, також видалилася і вистава яка належала цій аудиторії.

1. Створимо тригер, який буде рахувати кількість використань аудиторії.

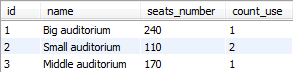
CREATE TRIGGER price\_level AFTER

INSERT ON theatr.performance FOR EACH ROW

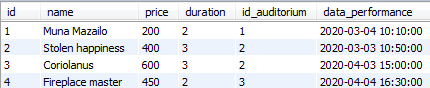
UPDATE theatr.auditorium SET count\_use = (SELECT count\_use FROM auditorium WHERE id = new.id\_auditorium)+1

WHERE auditorium.id = new.id\_auditorium;

**Перед вставленням вистави:**



Auditorium



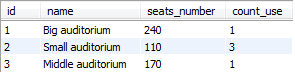
Performance

**Вставляємо виставу:**

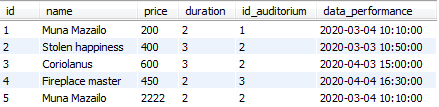
INSERT INTO performance VALUES

(5, "Muna Mazailo", 2222, 2, 2, "2020-05-04 10:10:00");

**Після вставлення вистави:**



Auditorium



Performance

Отже, можна побачити, що коли ми вставили нову виставу, яка буде використовувати аудиторію під id=2, кількість використання аудиторії збільшилося на 1.

**Висновок:** на лабораторній роботі я розробив SQL запити, які моделюють роботу тригерів: каскадне знищення, зміна та доповнення записів у зв’язаних таблицях.