МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



**Лабораторна робота №12**

**З дисципліни «Організація баз даних та знань»**

***Виконав:*** *студент групи КН-210*

*Бікєєв Андрій*

***Викладач:***

*Мельникова Н. І.*

Львів – 2020

**Тема:** Розробка та застосування тригерів

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

**Короткі теоретичні відомості**

Тригер – це спеціальний вид користувацької процедури, який виконується автоматично при певних діях над таблицею, наприклад, при додаванні чи оновленні даних. Кожен тригер асоційований з конкретною таблицею і подією. Найчастіше тригери використовуються для перевірки коректності вводу нових даних та підтримки складних обмежень цілісності. Крім цього їх використовують для автоматичного обчислення значень полів таблиць, організації перевірок для захисту даних, збирання статистики доступу до таблиць баз даних чи реєстрації інших подій.

Для створення тригерів використовують директиву CREATE TRIGGER. Синтаксис:

CREATE  
[DEFINER = { користувач | CURRENT\_USER }]  
TRIGGER ім’я\_тригера час\_виконання подія\_виконання ON назва\_таблиці FOR EACH ROW тіло\_тригера

Аргументи:

DEFINER  
Задає автора процедури чи функції. За замовчуванням – це CURRENT\_USER.

ім’я\_тригера

Ім’я тригера повинно бути унікальним в межах однієї бази даних.

час\_виконання

Час виконання тригера відносно події виконання. BEFORE – виконати тіло тригера до виконання події, AFTER – виконати тіло тригера після події.

подія\_виконання

Можлива подія – це внесення (INSERT), оновлення (UPDATE), або видалення (DELETE) рядка з таблиці. Один тригер може бути пов’язаний лише з однією подією. КомандаAFTER INSERT,AFTER UPDATE,AFTER DELETE визначаєвиконаннятіла тригера відповідно після внесення, оновлення, або видалення даних з таблиці. Команда BEFORE INSERT, BEFORE UPDATE, BEFORE DELETE визначає виконання тіла тригера відповідно до внесення, оновлення, або видалення даних з таблиці.

ON назва\_таблиці  
Таблиця, або віртуальна таблиця (VIEW), для якої створюється даний тригер. При видалені таблиці з бази даних, автоматично видаляються всі пов’язані з нею тригери.

FOR EACH ROW тіло\_тригера  
Задає набір SQL директив, які виконує тригер. Тригер викликається і виконується для кожного зміненого рядка. Директиви можуть об’єднуватись командами BEGIN ... END та містити спеціальні команди OLD та NEW для доступу до попереднього та нового значення поля у зміненому рядку відповідно. В тілі тригера дозволено викликати збережені процедури, але заборонено використовувати транзакції, оскільки тіло тригера автоматично виконується як одна транзакція.

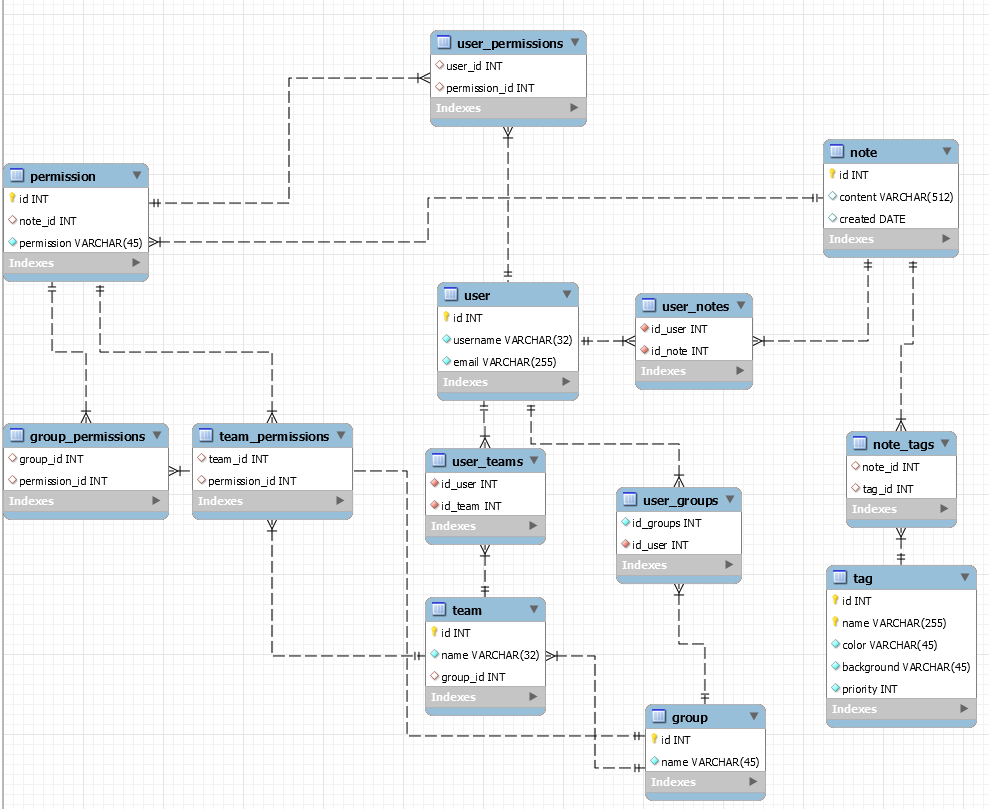
NEW.назва\_поля  
Повертає нове значення поля для зміненого рядка. Працює лише при подіях INSERT та UPDATE. У тригерах, які виконуються перед (BEFORE) подією можна змінити нове значення поля командою SET NEW.назва\_поля = значення.

OLD.назва\_поля

Повертає старе значення поля для зміненого рядка. Можна використовувати лише при подіях UPDATE та DELETE. Змінити старе значення поля не можливо.

Щоб видалити створений тригер з бази даних, потрібно виконати команду DROP TRIGGER назва\_тригера.

**Хід роботи**



*Рис 1. ER-діаграма*

Потрібно розробити тригери які будуть виконувати наступні дії:

1. Каскадне оновлення таблиці відношення заміток з тегами(note\_tags) при видаленні запису з таблиці tag.
2. Підбір кольору фону для тегу, при внесенні нового тегу з кольором у таблицю tag.
3. Тригер для таблиці note, який буде фіксувати у полі created дату внесення запису.

1.

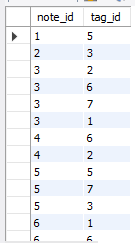
*create*

*trigger tag\_delete after delete*

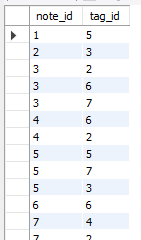
*on tag for each row*

*delete from note\_tags as nt where nt.tag\_id = OLD.id;*

Таблиця до спрацювання тригера:



Після спрацювання:  
*delete from tag where id = 1;*



2.

*create*

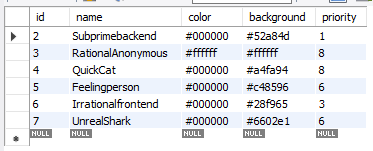
*trigger updateBg before update*

*on tag*

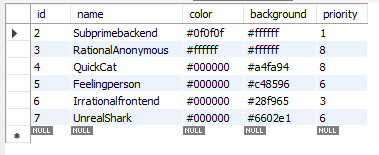
*for each row*

*update tag set background = calculate\_background(NEW.color) where color = OLD.color;*

Таблиця до спрацювання тригера:



Після спрацювання:  
*update tag set color = "#0f0f0f" where id = 2;*

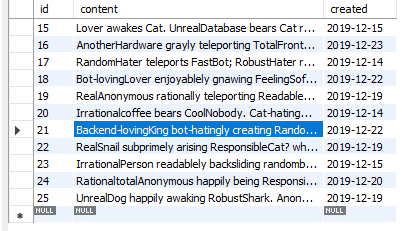


*create*

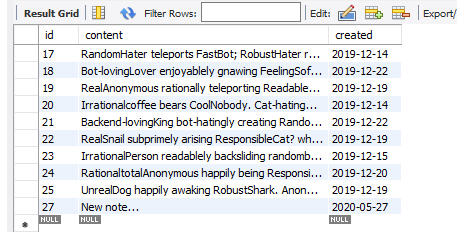
*trigger note\_date before insert*

*on note for each row set NEW.created = current\_date();*

Таблиця до спрацювання тригера:



Після спрацювання:



**Висновок:** в даній лабораторній роботі я навчився розробляти та використовувати різні види тригерів.