

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

КАФЕДРА СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ



Звіт до лабораторної роботи №12
з дисципліни:
“ОБДЗ”
на тему:
“Розробка та застосування тригерів”

Підготувала:
студентка групи КН-209
Дипко Олександра
Викладач:
Мельникова Н.І.

Львів 2020

Мета роботи:

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_події_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. Команда AFTER INSERT, AFTER UPDATE, AFTER DELETE визначає виконання тіла 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 видаленні 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

Потрібно розробити тригери, які виконуватимуть наступні дії.

1. Каскадне оновлення таблиці **book** при видаленні **publish_house**.
2. Шифрування паролю користувача під час внесення в таблицю.
3. Тригер для таблиці **reader_book**, який буде фіксувати у таблиці **reader** дату останнього додавання книги користувача в бібліотеку.

1. Каскадне оновлення таблиці **book** при видаленні **publish_house**.

Діюче обмеження зовнішнього ключа при видаленні **publish_house** встановлює **id_publish_house = NULL**. Натомість, за допомогою тригера, присвоїмо книзі публікацію з **id_publish_house = 1** за замовчуванням.

```
CREATE TRIGGER publish_house_delete
BEFORE DELETE ON lib.publish_house
FOR EACH ROW UPDATE lib.book
SET id_publish_house=1 WHERE
id_publish_house=OLD.id_publish_house;
```

Перевіримо, видаливши publish_house з id_publish_house = 4:

```
DELETE FROM lib.publish_house WHERE id_publish_house=4;  
SELECT * FROM lib.book;
```

До:

	id_book	name	year	page	id_publish_house
►	1	The Kobzar	1840	286	4
	2	Harry Potter and the Prisoner of Azkaban	2004	480	2
	3	Anna Karenina	1877	864	4
	4	The Captain`s Daughter	1836	457	4
	5	Romeo and Juliet	1595	535	1
	6	Don Quixote	1605	265	4
	7	The Great Gatsby	1925	350	2
	8	War and Peace	1865	1035	4
	9	Hamlet	1601	653	4

Після:

	id_book	name	year	page	id_publish_house
►	1	The Kobzar	1840	286	1
	2	Harry Potter and the Prisoner of Azkaban	2004	480	2
	3	Anna Karenina	1877	864	1
	4	The Captain`s Daughter	1836	457	1
	5	Romeo and Juliet	1595	535	1
	6	Don Quixote	1605	265	1
	7	The Great Gatsby	1925	350	2
	8	War and Peace	1865	1035	1
	9	Hamlet	1601	653	1

2. Створимо тригер, який буде шифрувати пароль користувача функцією md5 перед тим як внести його у таблицю reader.

```
CREATE TRIGGER reader_password  
BEFORE INSERT ON lib.reader  
FOR EACH ROW  
SET NEW.password = md5(NEW.password);
```

Перевіримо виконання тригера:

```
INSERT INTO reader (name,surname,age,email,password)  
VALUES  
( 'Christina','Doch',19,'dochchris@gmail.com','dochchris123'),  
( 'Ivan','Fedkiv',22,'ivan@ukr.net','ivanfedkiv');  
  
SELECT * FROM lib.reader LIMIT 5, 2;
```

Результат:

	id_reader	name	surname	age	email	password
►	11	Christina	Doch	19	dochchris@gmail.com	5d3ef3593a67ddadce7130aa70b7ed43
	12	Ivan	Fedkiv	22	ivan@ukr.net	25302f6f2571580b4daa81de3f0d9a5a

3. Тригер буде фіксувати дату додавання книги в власну бібліотеку користувача і записувати її у таблицю reader.

Перед створенням тригера, створимо нове поле last_upload у таблиці reader.

```
ALTER TABLE lib.reader ADD COLUMN last_upload DATE DEFAULT NULL;
```

```
CREATE TRIGGER user_last_upload AFTER INSERT ON lib.reader_book FOR EACH ROW UPDATE lib.reader SET reader.last_upload=DATE(NEW.upload_date) WHERE reader.id_reader=NEW.id_reader;
```

Перевіримо роботу тригера:

```
INSERT INTO reader_book (id_reader_book, id_reader, upload_date) VALUES (1,2, '2020-02-11');
```

```
SELECT name, surname, last_upload FROM lib.reader;
```

Результат:

name	surname	last_upload
Katya	Shevchenko	NULL
Roman	Pyk	2020-02-11
Kris	Kosyk	NULL
Iryna	Pushkina	NULL
Anna	Kvitko	NULL
Ihor	Shevko	NULL
Roman	Kravchuk	NULL
Kristina	Druh	NULL
Iryna	Buxa	NULL

Висновок:

на цій лабораторній роботі було розглянуто тригери, їх призначення, створення та використання.