

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**



**Звіт до лабораторної роботи №12**  
**З ПРЕДМЕТУ “ОРГАНІЗАЦІЯ БАЗ ДАНИХ ТА ЗНАНЬ”**

**Виконав:**

ст. гр. КН-211

Шебеко Андрій

**Викладач:**

Якимишин Х.М.

Львів – 2020

Лабораторна робота №12  
з курсу "ОБДЗ"  
на тему:  
**"Розробка та застосування тригерів"**

**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 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. 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.

## Хід роботи

1. Створимо тригер, який буде міняти значення 'машина' яка належить водію на 'null' при її видаленні.

```
create
trigger car_delete before delete
on shop.car for each row
update shop.driver set car_id = null where car_id = old.car_id;
```

Початковий стан таблиць:

driver_id	car_id	name	age	phone_number	experience
1	1	Andrii	14	21341413	4
2	2	Lor	15	32414234	5
3	3	Van	12	54153453	1
NULL	NULL	NULL	NULL	NULL	NULL

car_id	capacity
1	123
2	200
3	400
4	140
NULL	NULL

```
delete from shop.car where car_id = 1;
```

driver_id	car_id	name	age	phone_number	experience
1	NULL	Andrii	14	21341413	4
2	2	Lor	15	32414234	5
3	3	Van	12	54153453	1
NULL	NULL	NULL	NULL	NULL	NULL

car_id	capacity
2	200
3	400
4	140
NULL	NULL

2. Створимо тригер, який буде фіксувати дату, коли користувач здійснив останній свій коментар.

```
create trigger user_order after
insert on shop.order for each row
update shop.client set client.lastseen = date(curdate())
where client.client_id = NEW.client_id;
```

Початковий стан таблиць:

client_id	first name	second name	age	email	phone_number	register_date	password	pass	lastseen
1	Andrii	Sheba	18	andy@net	123131	2020-03-01 00:00:00	andy123	NULL	2020-05-15
2	Vitya	Surname	22	vik@ru	13425	2020-03-01 00:00:00	eaz	NULL	2020-05-13
3	Lera	Vitya	13	maol@mail	157415	2020-04-01 00:00:00	157415	NULL	2020-05-01

order_id	client_id	car_id	way_id	total_price	status	additional_inf
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
INSERT INTO `shop`.`order` (`order_id`, `client_id`, `car_id`, `way_id`, `total_price`, `status`, `additional_inf`) VALUES
('1', '1', '1', '1', '23414', 'Active', 'none'),
('2', '2', '2', '2', '1432', 'Active', 'Be carefull!'),
('3', '3', '3', '3', '33', 'Active', 'MMM');
```

client_id	first name	second name	age	email	phone_number	register_date	password	pass	lastseen
1	Andrii	Sheba	18	andy@net	123131	2020-03-01 00:00:00	andy123	NULL	2020-05-20
2	Vitya	Surname	22	vik@ru	13425	2020-03-01 00:00:00	eaz	NULL	2020-05-20
3	Lera	Vitya	13	maol@mail	157415	2020-04-01 00:00:00	157415	NULL	2020-05-20

order_id	client_id	car_id	way_id	total_price	status	additional_inf
1	1	1	1	23414	Active	none
2	2	2	2	1432	Active	Be carefull!
3	3	3	3	33	Active	MMM

3. Створимо тригер, який віднімає товари зі складу, відповідно до кількості, які є куплені покупцем.

```
create trigger user_order after
insert on shop.basket for each row
update shop.product set product.amount = product.amount - NEW.amount
where NEW.product_id = product.product_id;
```

Початковий стан таблиць:

product_id	product_name	description	weight	colour	amount	price
1	Table	Big round-shape table	40	green	690	499
2	Table	Big square-shape table	40	green	690	499
3	Chair	small light chair	11	light-green	2090	199
4	Umbrella	Big square `1715` umbrella	24	black with grttn lines	690	359
5	Table	Small round-shape table	20	green	890	279
NULL	NULL	NULL	NULL	NULL	NULL	NULL

	id	client_id	product_id	amount
►*	NULL	NULL	NULL	NULL

```
insert into basket values
```

```
(1, 1, 1, 4),
```

```
(3, 1, 1, 16);
```

product_id	product_name	description	weight	colour	amount	price
1	Table	Big round-shape table	40	green	686	499
2	Table	Big square-shape table	40	green	674	499
3	Chair	small light chair	11	light-green	2090	199
4	Umbrella	Big square `1715` umbrella	24	black with grttn lines	690	359
5	Table	Small round-shape table	20	green	890	279
NULL	NULL	NULL	NULL	NULL	NULL	NULL

id	client_id	product_id	amount
1	1	1	4
3	1	1	16
NULL	NULL	NULL	NULL

Висновок: я навчився розробляти та застосовувати тригери.