

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”  
ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ

Кафедра систем штучного інтелекту

Лабораторна робота №12  
із дисципліни  
Бази даних

Виконав:  
Ст. групи КН-207  
Гуменчук Б.Р.  
Прийняв:  
Мельникова Н.І.

Львів – 2018 р.

**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
```

### Хід роботи.

## 1. Каскадне оновлення таблиці користувачів при видаленні посади з Appointment.

До видалення:

```
172 • insert into performer values (5, "Volodymir", "2000-10-20", "volodyamort@gmail.com");
173 • insert into book values (11, "test_book_lab_12", 1, 1, 13, "Volodymir", 5);
174 • select * from performer;
175 • select * from book;
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

idperformer	name	birth_date	email				
1	Vasiliy	2014-12-10	vasya_homilka@gmail.com				
3	Kolya	1991-03-30	Kolyan@gmail.com				
4	Petro	2001-10-12	petya_geniy@gmail.com				
5	Volodymir	2000-10-20	volodyamort@gmail.com				
6	Sviatoslav	1991-04-04	sviatik_bos@gmail.com				

  

idbooks	description	customer_idcustomer	printer_idprinter	amount	performer	performer_idperformer
4	book_1	1	1	10	world_man1	1
5	book_2	1	1	10	world_man2	1
6	book_3	1	1	10	world_man3	1
9	just new book	2	1	10	Kolya	3
10	just new book	2	1	10	Kolya	3
11	test_book_lab_12	1	1	13	Volodymir	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Після видалення:

```
177 • drop trigger performer_delete;
178 • create trigger performer_delete before delete
179 on performer for each row
180 update book set performer_idperformer = OLD.idperformer - 1,
181 performer = (select name from performer where idperformer = OLD.idperformer - 1)
182 where performer_idperformer = OLD.idperformer;
183
184 • delete from performer where idperformer = 5;
185 • delete from book where idbooks = 11;
186 • select * from performer;
187 • select * from book;
```

< **Result Grid** | Filter Rows:  | Edit: | Export/Import: | Wrap Cell Content:

	idperformer	name	birth_date	email
▶	1	Vasiliy	2014-12-10	vasya_homilka@gmail.com
	3	Kolya	1991-03-30	Kolyan@gmail.com
	4	Petro	2001-10-12	petya_geniy@gmail.com
	6	Sviatoslav	1991-04-04	sviatik_bos@gmail.com
*	NULL	NULL	NULL	NULL

	idbooks	description	customer_idcustomer	printer_idprinter	amount	performer	performer_idperformer
▶	4	book_1	1	1	10	world_man1	1
	5	book_2	1	1	10	world_man2	1
	6	book_3	1	1	10	world_man3	1
	9	just new book	2	1	10	Kolya	3
	10	just new book	2	1	10	Kolya	3
	11	test_book_lab_12	1	1	13	Petro	4
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**Висновок:** на цій лабораторній роботі було розглянуто тригери в MySQL.