

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА  
ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота №12**  
**з дисципліни**  
**«Організація баз даних та знань»**

**Виконав:**  
**студент групи КН-209**

**Качмар Олексій**  
**Викладач:**  
**Мельникова Н.І.**

**Львів – 2020 р.**

## Лабораторна робота №12

**Тема:** “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.

### Завдання.

1. Шифрування паролю користувача під час внесення в таблицю.

```
Create trigger customer_password
Before insert on regular_customer
For each row
Set new.password = AES_ENCRYPT(new.password, 'key-key');
```

```
mysql> insert into regular_customer
-> values(6,"Marichka",'2345431',30,5,"Marichka_100500");
Query OK, 1 row affected (0.11 sec)
```

```
mysql> select * from regular_customer;
```

Id	full_name	telephone	sale	restaurant_Id	password
1	Alex	234324	15	1	0x
2	Vova	234324	20	1	0x
3	Maks	234324	25	1	0x
4	Alexandr	234324	20	1	0x
5	Petro	234324	40	1	0x
6	Marichka	2345431	30	5	0x407DAF2F174565C6C20771C4009EDEEA

6 rows in set (0.00 sec)

2. При видаленні ресторану автоматично видаляються зв'язані з ним працівники.

```
mysql> select * from restaurant;
```

Id	name	description	adress	telephone	website	social_media
1	Moms spaghetti	Best spaghetti in the whole world	Stepana Bandery street	0952345234	http://...	vk.com/...
2	MacDac		Doroshenka	09503943	http://localhost	http://fb.com
3	Subway		Rubchaka	023459631	http://fancyhome	http://tw.com
4	KFC	Great Chicken	Doroshenka	930495332	http://	http://viber.
5	Galytska Corona	Luxury restaurant	Doroshenka	930495312	http://	http://
6	Zora		Doroshenka	930495312	http://	http://

```
6 rows in set (0.00 sec)
```

```
mysql> select * from worker;
```

Id	full_name	age	telephone	date_of_employment	restaurant_Id	position_Id
1	Adrian React	2001-10-05	0954343523	2018-01-01	3	2
3	Alex Savitsky	2001-06-29	0954545454	2010-05-03	1	2
4	Max Gioman	2000-04-03	0932348512	2020-01-01	4	1
5	Taras Maksterm	2003-04-04	0954545342	2010-05-02	2	3
6	Olexandr Klib	2000-04-03	0954545342	2010-05-04	3	6
7	Andrew Chak	2001-04-03	0954545342	2010-05-04	2	3

```
6 rows in set (0.00 sec)
```

```
mysql> Create trigger delete_rest before delete on restaurant
-> for each row
-> delete from worker where restaurant_Id = old.Id;
Query OK, 0 rows affected (0.38 sec)
```

```
mysql> delete from restaurant where name = "KFC";
Query OK, 1 row affected (0.19 sec)
```

```
mysql> select * from worker;
```

Id	full_name	age	telephone	date_of_employment	restaurant_Id	position_Id
1	Adrian React	2001-10-05	0954343523	2018-01-01	3	2
3	Alex Savitsky	2001-06-29	0954545454	2010-05-03	1	2
5	Taras Maksterm	2003-04-04	0954545342	2010-05-02	2	3
6	Olexandr Klib	2000-04-03	0954545342	2010-05-04	3	6
7	Andrew Chak	2001-04-03	0954545342	2010-05-04	2	3

```
5 rows in set (0.00 sec)
```

3. При збільшенні кількості грам продукту в страві автоматично збільшується ціна страви відповідно до збільшення.

```
mysql> Create trigger changed_gram_changed_price
-> before update on meal_has_product
-> for each row
-> begin
-> if new.gram <> old.gram then
->   update meal
->   set price = price*(new.gram/old.gram)
->   where meal.Id = new.meal_id;
-> end if;
-> end$$
Query OK, 0 rows affected (0.13 sec)
```

```
mysql> select * from meal_has_product;
Empty set (0.00 sec)

mysql> select * from meal;
+-----+-----+-----+-----+-----+
| Id | dish      | price | gram | restaurant_Id |
+-----+-----+-----+-----+-----+
| 1 | Pasta     | 420   | 400  | 1              |
| 2 | Steak     | 500   | 300  | 1              |
| 3 | Chicken   | 200   | 100  | 1              |
| 4 | BroterBrod | 300   | 150  | 1              |
| 5 | Burger    | 350   | 200  | 1              |
+-----+-----+-----+-----+-----+
5 rows in set (0.04 sec)

mysql> select * from product;
+-----+-----+-----+
| Id | name      | partner_Id |
+-----+-----+-----+
| 1 | onion     | 2          |
| 2 | chees     | 3          |
| 3 | Pasta     | 3          |
| 4 | Meat      | 3          |
| 5 | sauce     | 1          |
| 6 | spices    | 2          |
| 7 | oil       | 3          |
| 8 | rosemary  | 3          |
+-----+-----+-----+
8 rows in set (0.05 sec)
```

```
mysql> insert into meal_has_product
-> Values(1,2,1,20);
Query OK, 1 row affected (0.09 sec)

mysql> select * from meal_has_product;
+-----+-----+-----+-----+
| Id | meal_id | product_id | gram |
+-----+-----+-----+-----+
| 1 | 2       | 1          | 20   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> select * from meal;
+-----+-----+-----+-----+-----+
| Id | dish      | price | gram | restaurant_Id |
+-----+-----+-----+-----+-----+
| 1 | Pasta     | 420   | 400  | 1              |
| 2 | Steak     | 500   | 300  | 1              |
| 3 | Chicken   | 200   | 100  | 1              |
| 4 | BroterBrod | 300   | 150  | 1              |
| 5 | Burger    | 350   | 200  | 1              |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> update meal_has_product
-> set gram = 30
-> where meal_id = 2;
Query OK, 1 row affected (0.16 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from meal;
+-----+-----+-----+-----+-----+
| Id | dish      | price | gram | restaurant_Id |
+-----+-----+-----+-----+-----+
| 1 | Pasta     | 420   | 400  | 1              |
| 2 | Steak     | 750   | 300  | 1              |
| 3 | Chicken   | 200   | 100  | 1              |
| 4 | BroterBrod | 300   | 150  | 1              |
| 5 | Burger    | 350   | 200  | 1              |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

#### 4.Автоматичний показ замовлень клієнта, який щойно зробив нове замовлення.

```
mysql> select * from regular_customer_orders ;
```

IFNULL(meal.dish,'Total')	count	price
BroterBrod	1	300
Burger	1	350
Chicken	1	200
Pasta	1	420
Steak	2	1500
Total	6	2770

```
6 rows in set (0.00 sec)
```

```
mysql> delimiter $$
mysql> Create trigger get_customer_meal after insert
-> on meal_has_regular_customer For each row
-> begin
-> delete from regular_customer_orders;
-> Insert into regular_customer_orders select IFNULL(meal.dish,'Total'),count(meal.Id) as count,Sum(meal.price) AS price
-> from meal_has_regular_customer
-> Inner Join meal
-> Inner Join regular_customer
-> On meal_has_regular_customer.regular_customer_id = regular_customer.Id
-> And meal_has_regular_customer.meal_id = meal.Id
-> Where regular_customer.Id= new.regular_customer_id
-> Group By meal_dish
-> With ROLLUP ;
-> end$$
Query OK, 0 rows affected (0.14 sec)
mysql> delimiter ;
```

```
mysql> delimiter ;
mysql> insert into meal_has_regular_customer
-> values(25,2,4,'2020-02-02');
Query OK, 1 row affected (0.11 sec)

mysql> select * from regular_customer_orders;
```

IFNULL(meal.dish,'Total')	count	price
Burger	1	350
Chicken	4	800
Pasta	2	840
Steak	2	1500
Total	9	3490

```
5 rows in set (0.00 sec)
```

```
mysql> select * from meal_has_regular_customer;
```

meal_has_regular_customer_id	meal_id	regular_customer_id	date_order
1	1	1	2015-05-03
2	2	1	2015-08-04
3	3	1	2015-09-04
4	4	1	2015-09-10
5	5	1	2015-09-15
6	3	2	2012-10-13
7	5	2	2013-10-10
8	1	2	2010-08-08
9	4	3	2017-03-02
10	2	3	2017-05-02
11	4	3	2017-05-03
12	2	1	2017-05-03
13	2	4	2017-05-03
14	1	4	2017-05-03
15	1	4	2017-05-03
19	3	4	2012-10-13
20	3	4	2012-11-13
21	5	4	2013-11-13
22	3	4	2013-01-02
23	3	4	2013-01-02
25	2	4	2020-02-02

```
21 rows in set (0.00 sec)
```

  

```
mysql> select * from meal;
```

Id	dish	price	gram	restaurant_Id
1	Pasta	420	400	1
2	Steak	750	300	1
3	Chicken	200	100	1
4	BroterBrod	300	150	1
5	Burger	350	200	1

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