МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Лабораторна робота №12

з курсу

«ОБДЗ»

на тему:

«Написання збережених процедур на мові SQL»

Виконав: Студент групи КН-209 Гладун Ярослав

Лабораторна робота №12

Тема: Розробка та застосування тригерів.

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

Хід роботи:

1. Я вирішив написати тригер, який буде кодувати пароль користувача.

```
DROP TRIGGER encode_password;

CREATE

TRIGGER encode_password BEFORE

INSERT ON AiGram.person FOR

EACH ROW

SET NEW.password = AES_ENCRYPT(NEW.password, 'pass');
```

Також варто зазначити, що тип поля password - VARBINARY(45)

2. Щоб перевірити роботу тригера зробемо вставку у таблицю з паролем helloworld

```
INSERT INTO person (name, phone_number, email, password, registration_date, avatar_path, about)
VALUES ('User', '292929', 'aaa@gmail.com', 'helloworld','2020-01-01', 'http/1', 'hello1');
```

3. Далі перевіремо роботу виконавши наступну комаду щоб переконатися у правельності роботи тригера:

```
SELECT CAST(AES_DECRYPT(password, 'pass') AS CHAR(45)) FROM person;

Table CAST(AES_DECRYPT(password, 'pass') AS CHAR(45)) 

1 helloworld
```

Як бачимо, результат вірний

4. Далі я вирішив створити тришер, який робить відповідність емейлу, номеру телефону та фотографії користувача, коли його видаляють із системи.

```
DROP TRIGGER remove_user;

CREATE

TRIGGER remove_user BEFORE

DELETE ON AiGram.person FOR

EACH ROW

BEGIN

INSERT INTO photo_deleted_users (phone_number, email, photo_id)

SELECT person.phone_number, person.email, photo.id FROM photo

RIGHT JOIN person on person.id = photo.persons_id

WHERE persons_id=OLD.id;

UPDATE photo SET photo.persons_id=NULL WHERE photo.persons_id=OLD.id;

END $$
```

5. Зробимо вставку юзерів у таблицю та спробує та спробуємо видалити юзера за id 7

```
INSERT INTO person (name, phone_number, email, password, registration_date, avatar_path, about) VALUES
('1', '1', '1', '2', '2020-01-01', '1', '1'),
('2', '2', '2', '2', '2020-01-01', '2', '2'),
('3', '3', '3', '3', '2020-01-01', '3', '3');
INSERT INTO photo (image_path, title, tag, persons_id) VALUES
('a', 'a', 'forest', 7),
('b', 'b', 'forest', 7),
('c', 'c', 'forest', 9);
```

DELETE FROM person WHERE id=7;

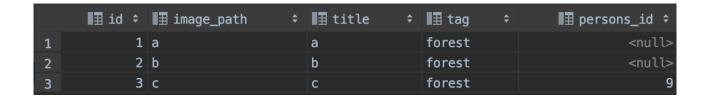
6. Вигляд таблиць до

| | I ∄ id ÷ I ∄ name | ‡ ■ phone_number | ≎ 🔢 email | ≎ III password | | | ‡ I II about ‡ |
|---|---------------------------------|-------------------------|-----------|----------------|---------------------|---|----------------------------|
| 1 | 7 1 | 1 | 1 | 1 | 2020-01-01 00:00:00 | 1 | 1 |
| 2 | 8 2 | 2 | 2 | 2 | 2020-01-01 00:00:00 | 2 | 2 |
| 3 | 9 3 | | 3 | 3 | 2020-01-01 00:00:00 | | 3 |

| | ∎ id ≑ | I image_path | Ⅲ title | I tag | ■ persons_id ÷ |
|---|--------|---------------------|----------------|--------------|-----------------------|
| 1 | 1 | a | а | forest | 7 |
| 2 | 2 | b | b | forest | 7 |
| 3 | 3 | С | С | forest | 9 |

Вигляд таблиць після

| | I ∄ id ≎ I ∄ name | ‡ II phone_number | ‡ II email | ‡ II password | | | ‡ III about ‡ |
|---|---------------------------------|--------------------------|--------------------|---------------|---------------------|---|---------------|
| 1 | 8 2 | 2 | 2 | 2 | 2020-01-01 00:00:00 | 2 | 2 |
| 2 | 9 3 | | 3 | | 2020-01-01 00:00:00 | | 3 |



Таблиця з відповідністю



Висновок: на цій лабораторній роботі я ознайомився із механізмом тригерів у СУБД MySQL.