Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра «Системи штучного інтелекту»



**Лабораторна Робота №12**

**З предмету: «Організація баз даних та знань»**

Виконав

студент групи КН-209

Дрібнюк Ярослав

Прийняла :

Мельникова Н. І.

**Львів-2020**

Тема: “Розробка та застосування тригерів”

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

Тригери бувають 6 видів: для 3 типів операцій (видалення, оновлення та вставлення даних), та для 2 станів (перед і після дією). У своїй роботі я використав 3 типи тригерів по одному на кожну операцію.

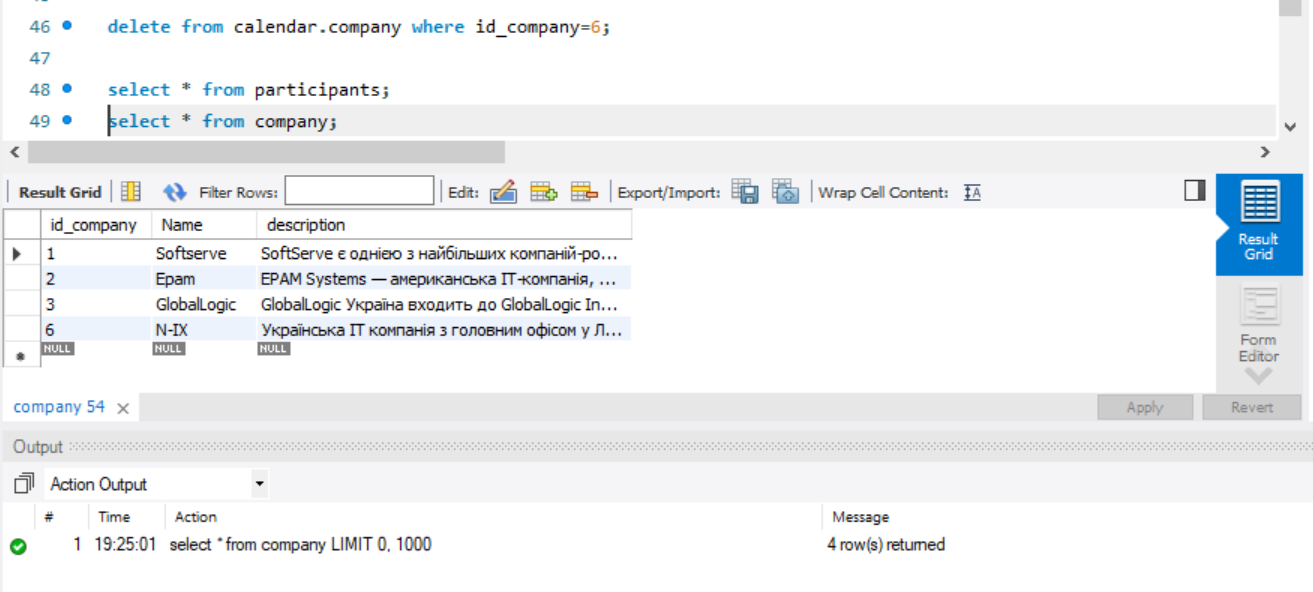
***create*** ***trigger*** delete\_company ***after***

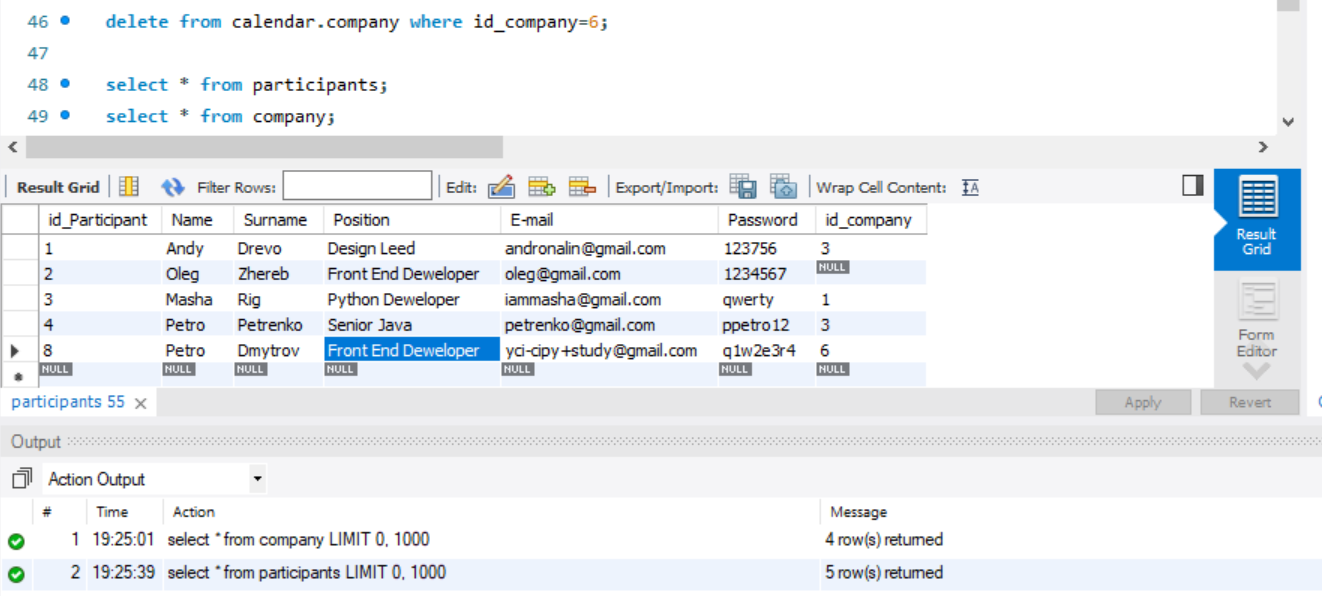
***delete on*** calendar.company ***for each row***

***update*** calendar.participants ***set*** Position='Безробітний' ***where*** id\_company=OLD.id\_company;

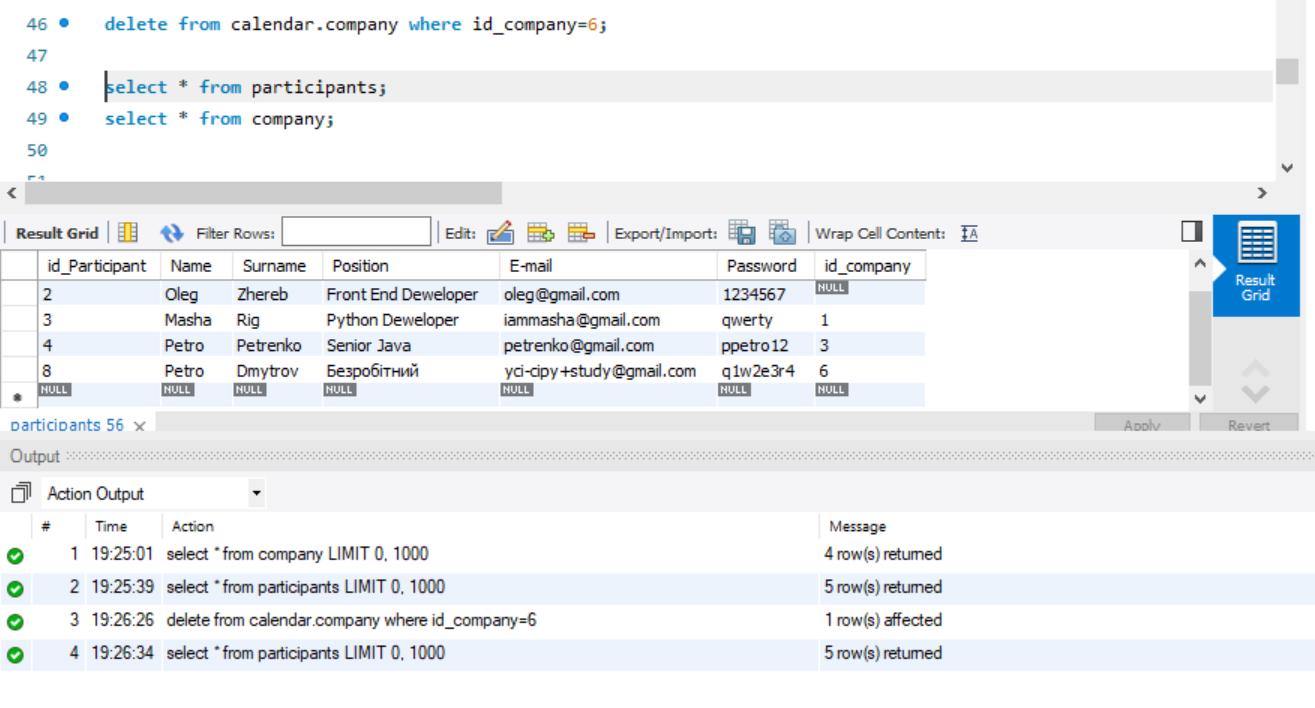
Напишу запит для видалення компанії:

***delete from*** calendar.company ***where*** id\_company=6;





Видалимо компанію і переглянемо нову інформацію про учасника:



Як бачимо, змінилася інформація про учасника у полі Position встановилося значення «Безробітний».

Наступний тригер я використав при додаванні нового користувача виконується шифрування пароль.

***create trigger*** hide\_password before insert

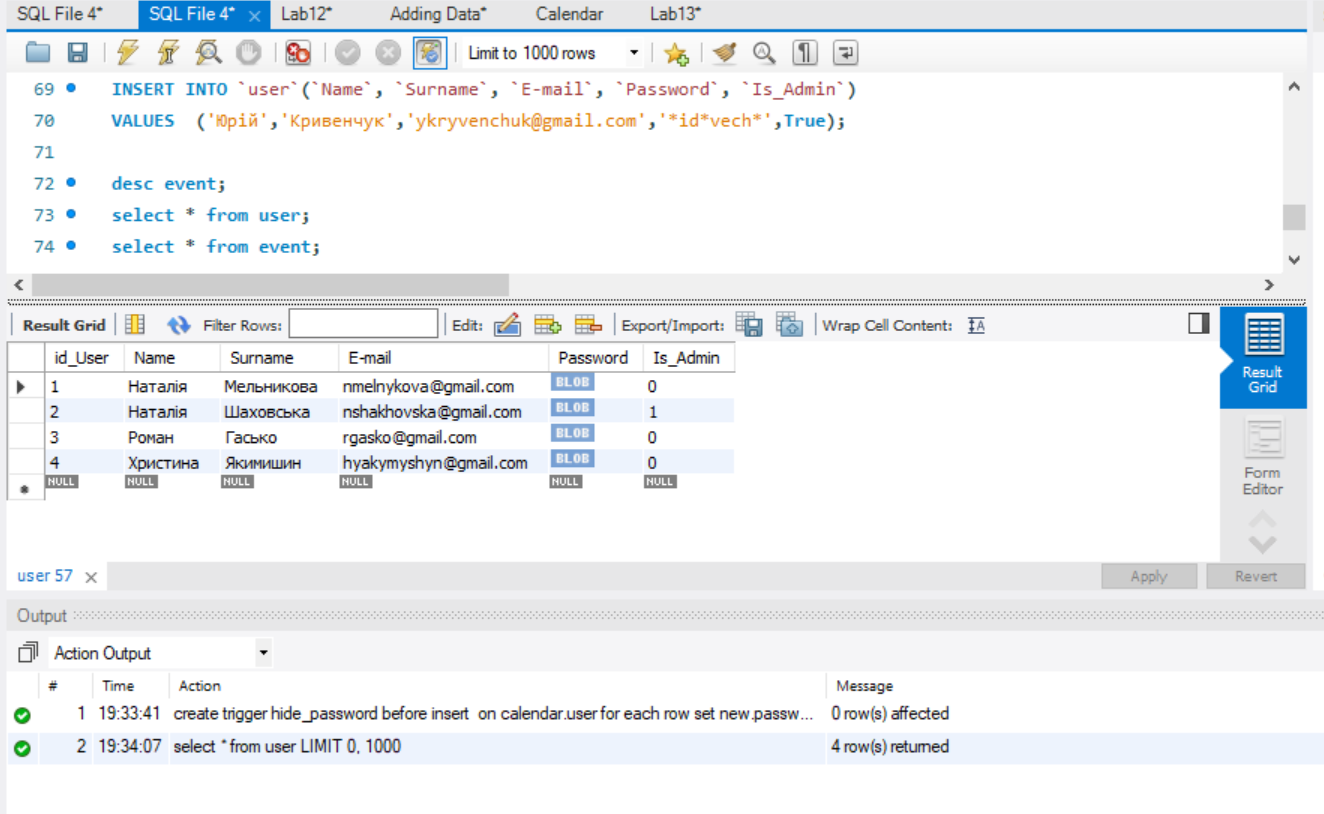
***on*** calendar.user ***for each row***

***set*** new.password = *aes\_encrypt*(new.password,'key-key');

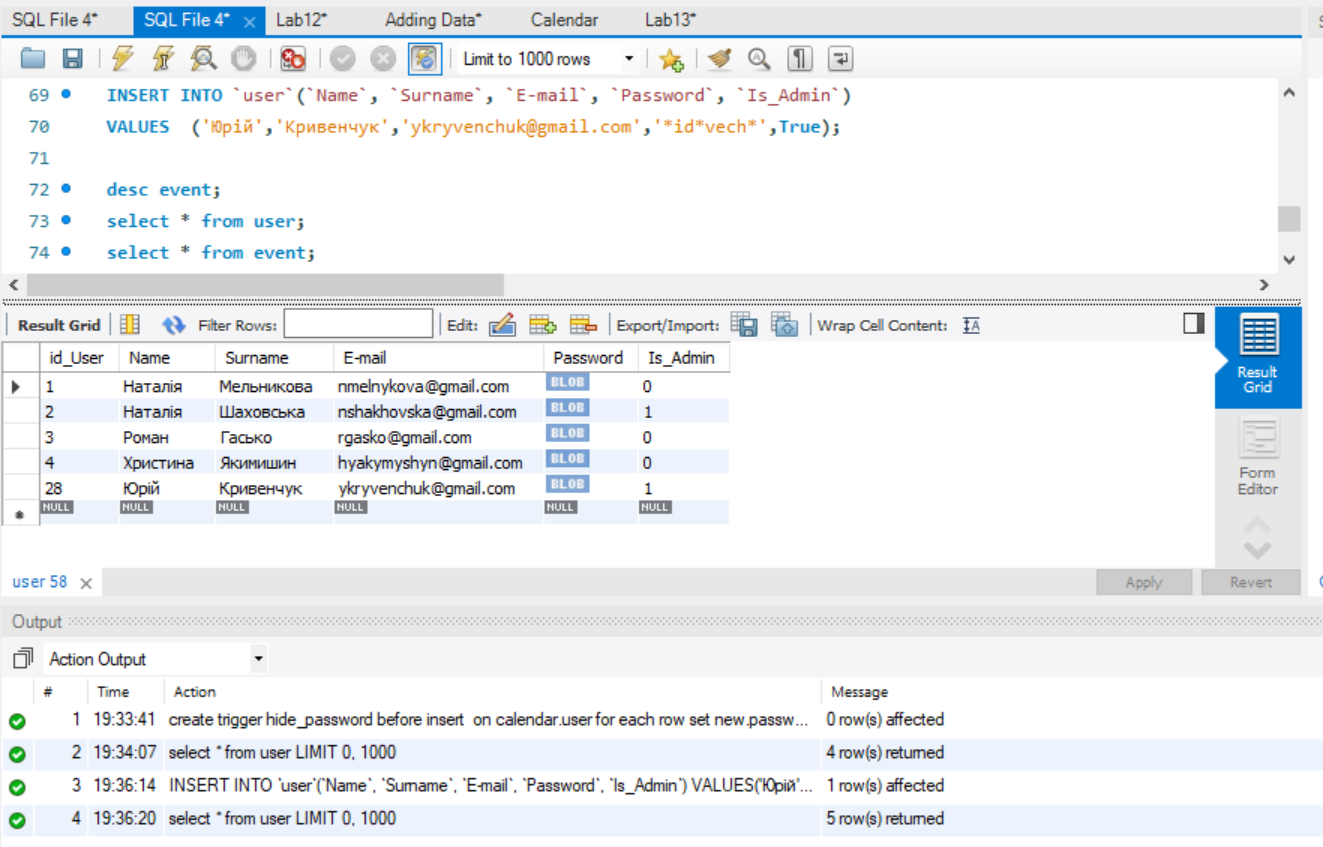
Додам запит додавання нового користувача у таблицю user:

***INSERT INTO*** `user`(`Name`, `Surname`, `E-mail`, `Password`, `Is\_Admin`)

***VALUES*** ('Юрій','Кривенчук','ykryvenchuk@gmail.com','\*id\*vech\*',True);



Створю нового користувача і перегляну дані про нього:



Тригер щодо оновлення я використав для поля Is\_Admin у таблиці користувача: якщо змінити значення цього поля на 0 то всі події, створені користувачем стають локальними.

***delimiter*** &&

***create trigger*** update\_position

***after update on*** user ***for each row***

***begin***

***if*** new.is\_Admin=0

***then update*** (user ***inner join*** event) ***SET*** event.Status="Локальна"

***where*** NEW.id\_User=event.id\_user;

***end if;***

***end;***&&

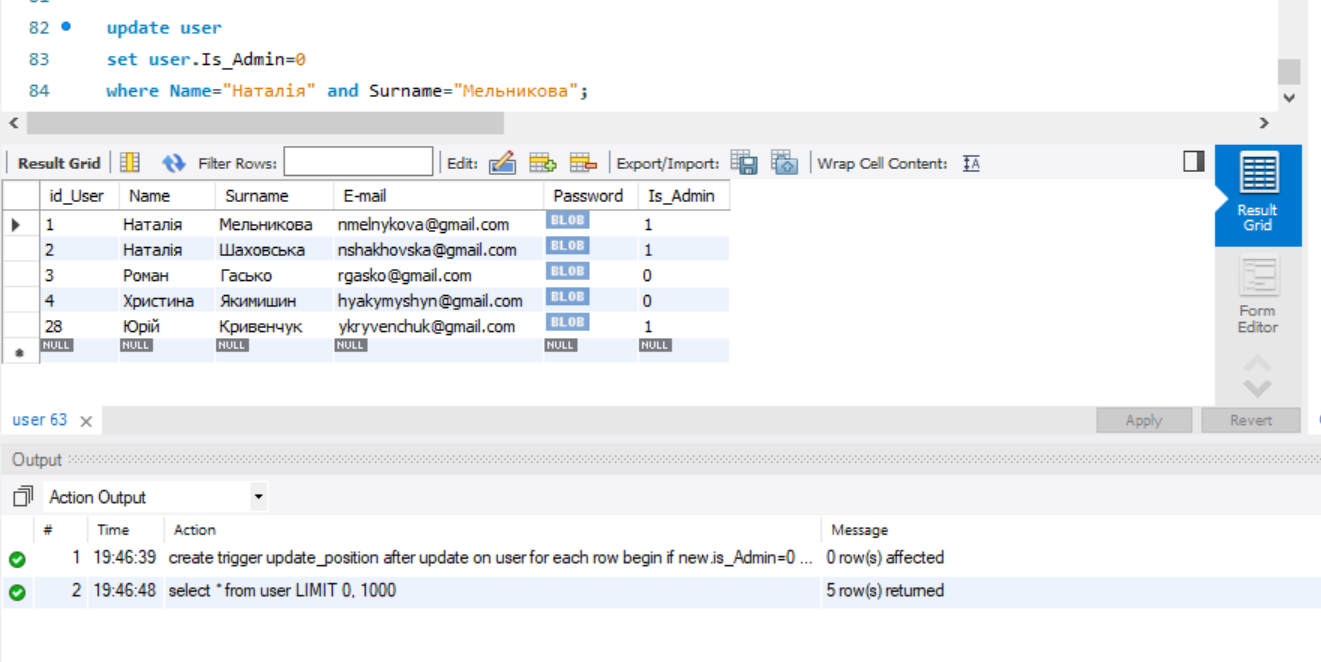
***delimiter*** ;

Виконаю update таким запитом:

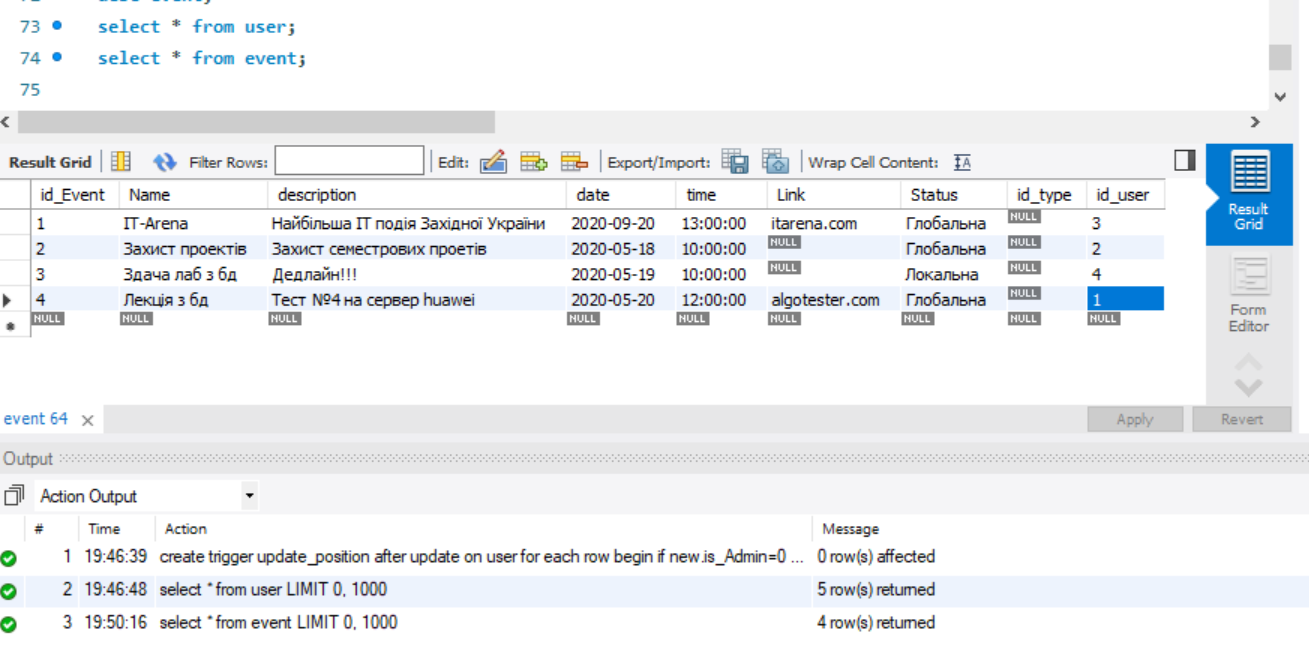
update user

set user.Is\_Admin=0

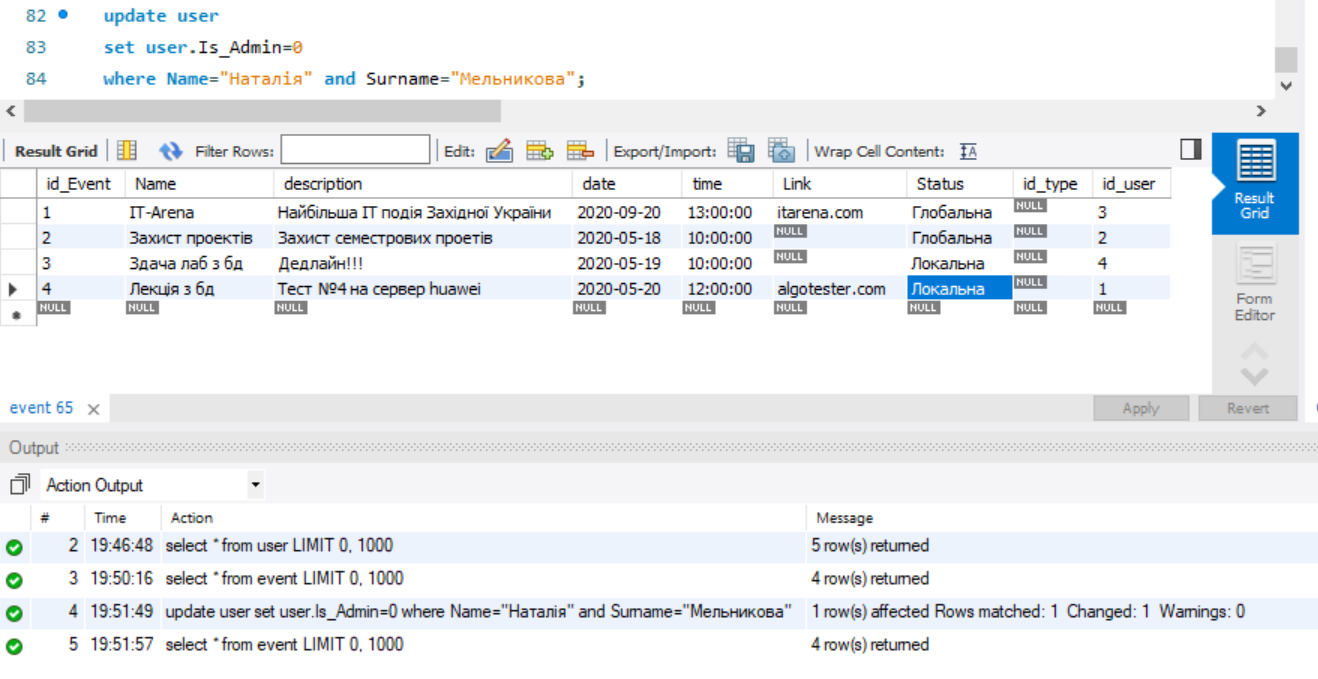
where Name="Наталія" and Surname="Мельникова";



Спочатку is\_admin=1 і статус події яка посилається на id 1 коритсувача= “Глобальна”



Запущу оновлення даних і перегляну події. Як можна побачити, зразу після видалення статус події змінився на “Локальна”



Висновок: на даній лабораторній роботі я розробив SQL запити, які моделюють роботу тригерів: каскадне знищення, зміна та доповнення записів у зв’язаних таблицях.