Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут прикладного системного аналізу

Кафедра математичних методів системного аналізу

**ЗВІТ**

про виконання комп’ютерного практикуму №2

з дисципліни «Системи баз даних»

Виконав:

Студент ІІІ курсу

Групи КА-96

Вітковський Д.О.

Перевірила:

Афанасьєва І.В.

Київ – 2021

**Завдання:** Відповідно до моєї теми курсового проекту, буду працювати із таблицею “Plugins”.

1. **Словесний опис:**

Завдання 1. Створити таблицю “Plugins”

Таблиця “Plugins” має містити такі поля:

* Поле з унікальним ключем-ідентифікатором
* Строкове поле (VARCHAR) з назвою плагіну
* Строкове поле (TEXT) з описом плагіну
* Поле із зовнішнім ключем на ідентифікатор категорії плагіну

Завдання 2. Створити тригер для таблиці “Plugins”, який буде спрацьовувати після додавання запису у таблицю та збільшувати лічильник загальної кількості зареєстрованих плагінів.

Завдання 3. Створити процедуру додавання плагіну до таблиці.

Процедура має містити такі поля:

* Строкове поле (VARCHAR) з назвою плагіну (in)
* Строкове поле (TEXT) з описом плагіну (in)
* Поле з ідентифікатором категорії плагіну (in)
* Поле з JSON списком ідентифікаторів авторів плагіну (in)

Завдання 4. Створити процедуру зміни плагіну у таблиці.

Процедура має містити такі поля:

* Поле з ідентифікатором плагіну (in)
* Строкове поле (VARCHAR) з новою назвою плагіну (in)
* Строкове поле (TEXT) з новим описом плагіну (in)
* Поле з новим ідентифікатором категорії плагіну (in)
* Поле з новим JSON списком авторів плагіну (in)

Завдання 5. Створити процедуру видалення плагіну з таблиці.

Процедура має містити поле, що ідентифікує плагін – ідентифікатор або назву плагіну.

Завдання 6. Створити процедуру, що повертає агреговане поле – кількість записів у таблиці плагінів.

Процедура полів не містить.

1. **Лістинг скрипту**

create table if not exists `course\_project`.`plugins`(

id bigint not null auto\_increment,

title varchar(200) not null unique,

plugin\_description text,

category\_id int null,

constraint plugin\_pk primary key (id),

constraint plugins\_categories\_fk foreign key (category\_id) references `course\_project`.`categories` (id) on delete set null on update cascade

);

set @total\_plugins = 0;

create trigger total\_plugins\_increment after insert on `course\_project`.`plugins`

for each row set @total\_plugins = @total\_plugins + 1;

create trigger total\_plugins\_decrement after insert on `course\_project`.`plugins`

for each row set @total\_plugins = @total\_plugins - 1;

delimiter //

create procedure publish\_plugin (in pl\_title varchar(200), in pl\_description text, in pl\_category int, in pl\_authors json)

begin

insert into `course\_project`.`plugins` (title, plugin\_description, category\_id) values(pl\_title, pl\_description, pl\_category);

set @current\_pl\_id = -1;

select id from `course\_project`.`plugins` where title = pl\_title into @current\_pl\_id;

set @indx = 0;

repeat

set @author\_id\_ = json\_extract(pl\_authors, concat('$[', @indx, ']'));

insert into `course\_project`.`plugins\_authors` values(@author\_id\_, @current\_pl\_id);

set @indx = @indx + 1;

until @indx = json\_length(pl\_authors)

end repeat;

end//

create procedure delete\_plugin (in pl\_title varchar(200))

begin

delete from `course\_project`.`plugins` where title = pl\_title;

end//

create procedure delete\_plugin\_by\_id (in pl\_id bigint)

begin

delete from `course\_project`.`plugins` where id = pl\_id;

end//

create procedure update\_plugin (in pl\_id bigint, in pl\_title varchar(200), in pl\_description text, in pl\_category int, in pl\_authors json)

begin

update `course\_project`.`plugins`

set title = pl\_title,

plugin\_description = pl\_description,

category\_id = pl\_category,

plugin\_description = pl\_description,

category\_id = pl\_category

where id = pl\_id;

delete from `course\_project`.`plugins\_authors` where plugin\_id = pl\_id;

set @indx = 0;

repeat

set @author\_id\_ = json\_extract(pl\_authors, concat('$[', @indx, ']'));

insert into `course\_project`.`plugins\_authors` values(@author\_id\_, pl\_id);

set @indx = @indx + 1;

until @indx = json\_length(pl\_authors)

end repeat;

end//

create function count\_plugins ()

returns bigint

begin

set @amount = 0;

select count(1) from `course\_project`.`plugins` into @amount;

return @amount;

end//

delimiter ;

**Висновки**

При виконанні комп’ютерного практикуму було вивчено, як створювати тригери, процедури та функції, що у подальшому допоможуть при виконанні курсового проекту.