

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 III курсу

Групи КА-96

Вітковський Д.О.

Перевірила:

Афанасьєва І.В.

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.

2. 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.