

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

## **Лабораторна робота №10**

з дисципліни

«Організація баз даних та знань»

**Виконав:**

студент групи КН-208

Матвіїв Микола

**Викладач:**

Мельникова Н. І.

Львів – 2019 р.

**Мета роботи:** Навчитися розробляти та виконувати збережені процедури та функції у MySQL.

### Короткі теоретичні відомості.

Більшість СУБД підтримують використання збережених послідовностей команд для виконання часто повторюваних, однотипних дій над даними. Такі збережені процедури дозволяють спростити оброблення даних, а також підвищити безпеку при роботі з базою даних, оскільки в цьому випадку прикладні програми не потребують прямого доступу до таблиць, а отримують потрібну інформацію через процедури.

СУБД MySQL підтримує збережені процедури і збережені функції. Аналогічно до вбудованих функцій (типу COUNT), збережену функцію викликають з деякого виразу і вона повертає цьому виразу обчислене значення. Збережену процедуру викликають за допомогою команди CALL. Процедура повертає значення через вихідні параметри, або генерує набір даних, який передається у прикладну програму.

### Хід роботи

Створив процедуру spGetBook, яка вибирає книжки з каталогу по id

```
CREATE PROCEDURE spGetBook
-- Add the parameters for the stored procedure here
@BookID int
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for procedure here
SELECT * from Book_Catalog where Book_Catalog.book_id = @BookID
END
GO
```

Перевіримо чи працює процедура

```
use Library;

execute spGetBook 1;
```

Вивід:

	book_id	book_id_catalog	place	status
1	1	1	1	0
2	1	2	2	0
3	1	3	3	0
4	1	4	4	1
5	1	5	5	1

Отже процедура працює

Створимо процедуру яка прийматиме два аргумента UserID та BookID

І перевірятиме чи користувач з id = UserID брав книгу з id = BookID

```
create PROCEDURE spUserTookBook
    @UserID int,
    @BookID int
AS
Begin
    SET NOCOUNT ON;

    select * from Book_Borrow inner join [User] on Book_Borrow.book_id = @BookID and [User].id = @UserID
end
go
```

Перевіримо чи користувач з id = 1 брав книгу з id = 1

```
execute spUserGetBook 1,1;
```

	book_id	user_id	borrow_date	return_date	period	id	name	sumame	phone_number	adress
1	1	1	2019-04-25	2019-12-25	244	1	'Nikita'	'Ivanov'	982525255	'Lviv'

## Висновок

на цій лабораторній роботі я навчився розробляти та використовувати збережені процедури і функції у СУБД SSMS.