

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

КАФЕДРА СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ



Звіт до лабораторної роботи №10
з дисципліни:
“ОБДЗ”
на тему:
«Збережені процедури та функції у MySQL»

Виконала:
студентка групи КН-211
Досяк Ірина
Викладач:
Якимишин Х.Я.

Лабораторна робота №10

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.

Синтаксис команд для створення збережених процедур описано нижче.

CREATE

[DEFINER = { користувач | CURRENT_USER }]

FUNCTION назва_функції ([параметри_функції ...])

RETURNS тип

[характеристика ...] тіло_функції

CREATE

[DEFINER = { користувач | CURRENT_USER }]

PROCEDURE назва_процедури ([параметри_процедури ...]) [характеристика ...] тіло_процедури

Завдання:

1. Напишемо функцію, яка буде обгорткою стандартної функції шифрування.
2. Напишемо функцію, яка буде обгорткою стандартної функції дешифрування.
3. Напишемо процедуру, яка буде обчислювати кількість користувачів, які прочитали книгу за вказаний проміжок часу.

Виконання:

1. Функція шифрування.

```
CREATE FUNCTION encode (password TINYBLOB)
RETURNS CHAR(48) RETURN md5(password);
```

2. Функція дешифрування.

```
CREATE FUNCTION decode(password CHAR(45))
RETURNS BLOB
DETERMINISTIC
RETURN CAST(AES_DECRYPT(AES_ENCRYPT(password,'ABC'),
'ABC') AS CHAR(45));
```

Використаємо функції для кодування та декодування паролів з reader:

	password	encode	decode
►	katya1234	93d69187b82cb7e0b62878d4e7209c14	katya1234
	romaroma12	3450ca9e7cfc00db633b386b1c925441	romaroma12
	kriskosykkris	e2d7b52ace8a85a9a3ec4877c5aa5fe8	kriskosykkris
	irynapushkina283	0081b43570e83482c862d49991982369	irynapushkina283
	annaanna34	1bdd7156b55871c4d126b6151efc23bf	annaanna34

3. Процедура повинна рахувати кількість користувачів, які прочитали книгу за певний проміжок часу. У процедуру потрібно передати першу і другу дату.

Перед основними директивами додамо перевірку коректності задання початкової і кінцевої дати (IF date1<=date2 THEN...). Результати обчислень будуть записуватись у таблицю number, яку процедура завжди очищує (командою TRUNCATE library.number) і заповнює з нуля.

```
DELIMITER //
CREATE PROCEDURE number (IN date1 DATE, IN date2 DATE)
BEGIN
DECLARE error CHAR(30);
```

```

SET error = 'Invalid date entered';
IF (date1 <= date2) THEN BEGIN
    CREATE TABLE IF NOT EXISTS library.number (date1 DATE, date2 DATE,
number int);
    TRUNCATE library.number;
    INSERT INTO library.number SELECT date1, date2,
COUNT(reader_book.id_reader) AS number
FROM reader_book
WHERE reader_book.upload_date BETWEEN date1 AND date2 AND
reader_book.status = 'read';
END;
ELSE SELECT error; END IF;
END// DELIMITER
Викликаємо процедуру:
CALL number('2020-01-01', '2020-05-05');
SELECT * FROM number;

```

	date1	date2	number
►	2020-01-01	2020-05-05	4

```
CALL number('2010-01-01', '2009-05-05');
```

	error
►	Invalid date entered

Висновок: на цій лабораторній роботі я навчився розробляти та використовувати збережені процедури і функції у MySQL.

