

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

КАФЕДРА СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ



Звіт до лабораторної роботи №10
з дисципліни:
“ОБДЗ”
на тему:
“Написання збережених процедур на мові SQL ”

Підготувала:
студентка групи КН-209
Дипко Олександра

Викладач:
Мельникова Н.І.

Львів 2020

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
```

параметри_процедури:

[IN | OUT | INOUT] ім'я_параметру тип

Параметр, позначений як IN, передає значення у процедуру. OUT-параметр передає значення у точку виклику процедури. Параметр, позначений як INOUT, задається при виклику, може бути змінений всередині процедури і зчитаний після її завершення. Типом параметру може бути будь-який із типів даних, що підтримується MySQL.

параметри_функції:

ім'я_параметру тип

У випадку функцій параметри використовують лише для передачі значень у функцію. При створенні процедур і функцій можна вказувати їхні додаткові характеристики.

характеристика:

CONTAINS SQL | NO SQL

Вказує на те, що процедура містить (за замовчуванням), або не містить директиви SQL.

READS SQL DATA

Вказує на те, що процедура містить директиви, які тільки зчитують дані з таблиць.

MODIFIES SQL DATA

Вказує на те, що процедура містить директиви, які можуть змінювати дані в таблицях.

Хід роботи

Напишемо функції, які будуть обгортками стандартних функцій шифрування, та процедуру, яка буде обчислювати кількість користувачів, які завантажили книгу за вказаний проміжок часу.

1. Функція шифрування.

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

Використаємо функцію для кодування паролів з reader:

```
SELECT password, decode(password) FROM reader;
```

password	decode(password)
katya1234	93d69187b82cb7e0b62878d4e7209c14
romaroma12	3450ca9e7cfc00db633b386b1c925441
kriskosykkris	e2d7b52ace8a85a9a3ec4877c5aa5fe8
irynapushkina283	0081b43570e83482c862d49991982369
igor1234	01b01ae5838923468e1e48cec74bb3bf
kravroma12	ffec01730562218bf63cf9011f77b6db
kriskris	1ed3dc03b33526dcb1c3dacb73fa6ed8
irynabuxa283	a3175118ddf8cd7e705198ad3e7c00c0
annakvit23	26f41e2f4036760d4a384ccc39a91d48

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

Усі дати завантаження книг користувачами:

	id_reader	upload_date
►	2	2020-02-11
	1	2020-03-05
	4	2020-02-28
	3	2020-02-25
	2	2020-03-23
	9	2020-04-01
	7	2019-12-11
	9	2020-04-01
	6	2020-01-15
	6	2020-04-01
	6	2020-04-05

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 mycms.stats) 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 lib.number (date1 DATE, date2 DATE,
        number int) ;
```

```

TRUNCATE lib.number;
INSERT INTO lib.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', '2020-04-01');
SELECT * FROM number;

```

	date1	date2	number
►	2020-01-01	2020-04-01	5

```

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

```

	error
►	Invalid date entered

3. Процедура повинна рахувати кількість книг, які були видані певним видавництвом.
У процедуру потрібно передати назву видавництва.

Усі книги:

id_book	name	year	page	id_publish_house
1	The Kobzar	1840	286	4
2	Harry Potter and the Prisoner of Azkaban	2004	480	2
3	Anna Karenina	1877	864	4
4	The Captain`s Daughter	1836	457	4
5	Romeo and Juliet	1595	535	1
6	Don Quixote	1605	265	4
7	The Great Gatsby	1925	350	2
8	War and Peace	1865	1035	4
9	Hamlet	1601	653	4
10	The Divine Comedy	1472	535	1
11	Gullivers Travels	1726	543	2
12	Alices Adventures in Wonderland	1862	425	3

Результати обчислень будуть записуватись у таблицю lib.pub, яку процедура завжди очищує (командою TRUNCATE) і заповнює з нуля.

```

DELIMITER //
CREATE PROCEDURE publish (IN pub_house VARCHAR(30))
BEGIN
    CREATE TABLE IF NOT EXISTS lib.pub (pub_house VARCHAR(30), number
    int);
    TRUNCATE lib.pub;
    INSERT INTO lib.pub SELECT pub_house, COUNT(book.id_publish_house)
    AS number
    FROM book, publish_house WHERE publish_house.name = pub_house
    AND book.id_publish_house = publish_house.id_publish_house;
END //
DELIMITER ;

```

Викликаємо процедуру:

```
CALL publish('Nash Format');  
SELECT * FROM pub;
```

pub_house	number
Nash Format	3

Висновок:

у цій лабораторній роботі навчилася розробляти та виконувати збережені процедури та функції у MySQL.