Міністерство освіти і науки України
Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

Лабораторна робота №10
на тему:
«Написання збережених процедур на мові SQL»
з курсу:
«ОБДЗ»

Виконала:
ст. гр. КН-210
Ямнюк Аліна
Прийняла:
Мельникова Н.І.

Львів – 2020 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.

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

Аргументи:

DEFINER
Задає автора процедури чи функції. За замовчуванням – це *CURRENT_USER.*
RETURNS
Вказує тип значення, яке повертає функція.
тіло_функції, тіло_процедури
Послідовність директив SQL. В тілі процедур і функцій можна оголошувати локальні змінні, використовувати директиви *BEGIN ... END, CASE*, цикли тощо. В тілі процедур також можна виконувати транзакції. Тіло функції обов'язково повинно містити команду *RETURN* і повертати значення.

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.

Хід роботи:

1) Напишемо функцію, яка виводитиме ціну процедури зі заданого проміжку, у нашому випадку 200-450. Якщо ціна процедури < 200 або >450, то RETURN(0)

```
USE beauty;
CREATE FUNCTION check_price(price INT, lower_bound INT, upper_bound INT)
    RETURNS INT
    DETERMINISTIC
    BEGIN
        IF (price BETWEEN lower_bound AND upper_bound) THEN
            RETURN (price);
        END IF;
        RETURN (0);
    END;

SELECT name_service, price FROM beauty.service WHERE price = check_price( price: price,  lower_bound: 200,  upper_bound: 450)
```

Результат функції:

	name_service	price
1	haircut	400
2	manicure	250
3	pedicure	350

2) Написати процедуру, яка за іменем і прізвищем клієнта виводитиме телефон, дату, час, послугу, ціну послуги і майстра, який виконує роботу.

Додамо перевірку коректності ім'я та прізвища користувача за допомогою умовного оператора if. Об'єднаємо таблиці, з яких потрібно отримати інформацію про клієнта

```
DELIMITER //
USE beauty;
CREATE PROCEDURE client_info (IN f_name VARCHAR(30), IN l_name VARCHAR(30))
BEGIN
DECLARE error VARCHAR(15);
SET error = 'Wrong input';
IF (f_name = (SELECT first_name FROM beauty.isClient WHERE first_name = f_name) AND
    l_name =  (SELECT last_name FROM beauty.isClient WHERE last_name = l_name)) THEN
    BEGIN
        SELECT concat_ws(' ', c.first_name, c.last_name) AS Client, c.telephone,
        oc.dateOfTheOrder AS date , oc.timeOfTheOrder AS time, sr.name_service AS service,
        oc.priceOfTheOrder AS price, concat_ws(' ', s.first_name, s.last_name) AS master
        FROM beauty.order AS o
        INNER JOIN beauty.orderoftheclient AS oc ON  o.id_order = oc.id_order
        INNER JOIN beauty.isClient AS c ON oc.id_client = c.id_client
        INNER JOIN beauty.staff AS s ON o.id_staff = s.id_staff
        INNER JOIN beauty.service AS sr ON o.id_service = sr.id_service
        WHERE c.first_name = f_name AND c.last_name = l_name;
    END;
    ELSE SELECT error;
END IF;
END;

DELIMITER ;
```

Викликаємо процедуру з вірними даними:

```
CALL client_info( f_name: 'Solomia',  l_name: 'Leshchak');
```

	Client	telephone	date	time	service	price	master
1	Solomia Leshchak	0937641190	2020-06-05	14:00:00	manicure	250	Katia Okal

Викликаємо процедуру з невірними даними:

```
CALL client_info( f_name: 'Maria',  l_name: 'Ivaniv');
```

	error
1	Wrong input

Висновок: на цій лабораторній роботі я навчилась розробляти та виконувати збережені процедури та функції у MySQL.