

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Лабораторна робота №10

з дисципліни

«Організація баз даних та знань»

Виконав:
студент групи КН-209

Качмар Олексій

Викладач:
Мельникова Н.І.

Львів – 2020 р.

Лабораторна робота №10

Тема: “Написання збережених процедур на мові SQL”

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.

1. Запишемо функцію і викличемо функцію, яка повертатиме нам факторіал числа. Позначимо цю функцію Deterministic, адже при однакових вхідних даних вона повертає однаковий результат. Використаємо директиву while do.

```
mysql> Create Function Factorial(number INT)
-> Returns BIGINT
-> Deterministic
-> Begin
-> Declare start BIGINT DEFAULT 1;
-> while number > 0 Do
-> SET start = start * number;
-> SET number = number - 1;
-> end while;
-> Return start;
-> End$$
```

Query OK, 0 rows affected (0.15 sec)

```
mysql> DELIMITER ;
mysql> select Factorial(5);
```

```
+-----+
| Factorial(5) |
+-----+
|          120 |
+-----+
```

1 row in set (0.05 sec)

2. Запишемо функцію FizzBuzz, за мотивами гри: маємо конкатенацію стрічок, при числі, яке ділиться на 3 без остачі нам видає fizz при числі, що ділиться на 5 buzz, при числі, що ділиться і на 3 і на 5 нам видає fizzbuzz. Використали case і while до директиви.

```
mysql> Create Function FizzBuzz(leftEdge Int, rightEdge Int)
  -> Returns Varchar(1000)
  -> Deterministic
  -> Begin
  -> Set @str = '';
  -> While leftEdge <= rightEdge Do
  -> Case
  -> When leftEdge % 3 = 0 And leftEdge % 5 = 0 Then SET @str = CONCAT(@str, 'fizzbuzz');
  -> When leftEdge % 3 = 0 Then SET @str = Concat(@str, 'fizz');
  -> When leftEdge % 5 = 0 Then SET @str = Concat(@str, 'buzz');
  -> Else Set @str = Concat(@str, Cast(leftEdge AS Char));
  -> End Case;
  -> SET leftEdge = leftEdge + 1;
  -> End While;
  -> Return @str;
  -> End$$
Query OK, 0 rows affected (1.43 sec)

mysql> Delimiter ;
mysql> select FizzBuzz(1,15);
+-----+
| FizzBuzz(1,15) |
+-----+
| 12fizz4buzzfizz78fizzbuzz11fizz1314fizzbuzz |
+-----+
1 row in set (0.04 sec)
```

3. Створимо процедуру, що за назвою вулиці видаватиме нам кількість ресторанів, що знаходяться на ній. Використали параметр out.

```
mysql> delimiter $$
mysql> Create procedure count_of_restaurants_on_street(IN name_of_Street varchar(15), OUT number_of_rests Smallint)
  -> Begin
  -> select count(*) Into number_of_rests
  -> from restaurant
  -> where adress = name_of_Street;
  -> End$$
Query OK, 0 rows affected (0.10 sec)
```

```
mysql> delimiter ;
```

```
mysql> call count_of_restaurants_on_street("Doroshenka", @count);
Query OK, 1 row affected (0.00 sec)

mysql> select @count;
+-----+
| @count |
+-----+
|      4 |
+-----+
1 row in set (0.00 sec)
```

4. Процедура, що визначає за назвою страви, в яких ресторанах її готують і з яких продуктів

```
mysql> delimiter $$
mysql> Create procedure get_restaurant_meal_products(IN name_of_meal varchar(45))
  -> Begin
  -> select restaurant.name as restName,meal.dish as Dish,Group_Concat(product.name) as pName
  -> from meal Inner join restaurant
  -> Inner join meal_has_product
  -> Inner join product
  -> On meal.restaurant_Id = restaurant.Id
  -> And meal.Id = meal_has_product.meal_Id
  -> And meal_has_product.product_id=product.Id
  -> where meal.Dish = name_of_meal;
  -> End$$
Query OK, 0 rows affected (0.09 sec)

mysql> delimiter ;
mysql> call get_restaurant_meal_products("Steak");
+-----+-----+-----+
| restName      | Dish  | pName                                     |
+-----+-----+-----+
| Moms spaghetti | Steak | Meat,spices,oil,rosemary |
+-----+-----+-----+
1 row in set (0.00 sec)
```

5.Процедура з бд world.Вибираємо 2 таблиці :countrylanguage та country.

Робимо вибірку країни і офіційні мови на її території, крім тих, де континент – Антарктика, адже на цьому континенті не розташовано жодної країни.

```
mysql> delimiter $$
mysql> Create procedure check_official_country_languages(IN name_of_country Varchar(50))
  -> Begin
  -> Declare error Varchar(100);
  -> Set error = 'На цьому континенті не розташовано жодної країни';
  -> select Continent into @Cont from country where Name = name_of_country;
  -> if @Cont <> "Antarctica" Then
  -> select country.Name,Group_Concat(countrylanguage.Language),countrylanguage.IsOfficial from country
  -> Inner join countrylanguage
  -> On country.Name = name_of_country
  -> And country.Code=countrylanguage.CountryCode
  -> where countrylanguage.IsOfficial = 'T'
  -> group by country.Name;
  -> Else select error;
  -> End if;
  -> End$$
Query OK, 0 rows affected (0.18 sec)

mysql> delimiter ;
mysql> call check_official_country_languages("French Southern territories");
+-----+-----+
| error                                     |
+-----+-----+
| На цьому континенті не розташовано жодної країни |
+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> call check_official_country_languages("Antarctica");
+-----+-----+
| error                                     |
+-----+-----+
| На цьому континенті не розташовано жодної країни |
+-----+-----+
```

```
mysql> call check_official_country_languages("Belgium");
+-----+-----+-----+
| Name   | Group_Concat(countrylanguage.Language) | IsOfficial |
+-----+-----+-----+
| Belgium | Dutch,French,German                    | T          |
+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

Висновок: я навчився розробляти та виконувати збережені процедури та функції у MySQL.