

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Лабораторна робота №10
з дисципліни
«ОБДЗ»

Виконав:
Студент групи КН – 209
Гречух Тарас

Перевірила:
Мельникова Н.І.

Львів 2020

Тема: Написання збережених процедур на мові SQL

Мета: 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:

1. Функція яка шукає повний заробіток враховуючи середню вартість авто та кількість проданих авто:

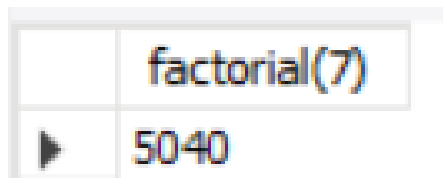
```
delimiter //  
create function profit(avg_salary int, cars_sold int)  
returns int  
deterministic  
begin  
declare profit int default 0;  
set profit = avg_salary * cars_sold;  
return profit;  
end  
//  
  
select profit(5350, 27);
```

	profit(5350, 27)
▶	144450

2. Функція для пошуку факторіалу:

```
delimiter //  
create function factorial(num int)  
returns bigint  
deterministic  
begin  
declare start bigint default 1;  
while num > 0 do  
set start = start * num;  
set num = num - 1;  
end while;  
return start;  
end//
```

```
select factorial(7)
```



3. Процедура яка шукає на яку суму було продано, куплено авто певним користувачем, та його прибуток відносно цих даних:

```
delimiter //  
create procedure salary_count (in my_seller_id int)  
begin  
select  
(select sum(price)  
from carsell.car  
where seller_id = my_seller_id) as profit,  
(select sum(price)  
from carsell.car  
where buyer_id = my_seller_id) as spending,  
(select profit - spending) as clear_profit;  
end//
```

call salary_count(217);

	profit	spending	clear_profit
►	326700	3300	323400

4. Процедура, яка виводить нам всі оголошення які були виставлені в заданий проміжок часу:

```
delimiter //
create procedure ann_count (in date1 date, in date2 date)
begin
declare error char(50);
set error = "Wrong data input";
if (date1 < date2) then
begin
create table if not exists carsell.time_stats (seller int, buyer int, post_date date);
truncate carsell.time_stats;
insert into carsell.time_stats
select seller_id, buyer_id, post_date
from carsell.car
where post_date between date1 and date2
order by post_date;
end;
else select error;
end if;
end//

call ann_count ("2014-07-01", "2018-05-26");
```

	seller	buyer	post_date
►	217	75	2014-07-20
	217	495	2014-07-26
	24	356	2016-11-25
	32	7	2017-02-19
	7	1050	2017-08-01
	217	75	2018-03-17

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