MySQL

Practical 8 Stored Procedure Function

1 Create procedure called proc1 which declare one integer variable and one varchar variable and display both the variables.

```
delimiter $$
create procedure proc1()
begin
   declare id int;
   declare fname varchar(30);
   set id = 1;
   set fname ='ram';
   select id, fname ;
end $$
delimiter ;

call proc1();
+------+-------+
| id   | fname |
+------+-------+
|    1 | ram   |
+------+-------+
```

2. Create procedure called proc2 in which declare the variable counter = and execute while loop until counter > 0 .

```
delimiter $$
create procedure proc2()
begin
   declare count_1 int;
   set count_1 = 10;

   while count_1 > 0 do
      select count_1;
      set count_1 = count_1- 1;
   end while;
end $$
delimiter ;
```

10

9

8

7

6

5

4

3

2

1

3. create procedure called proc3, which pass the argurment N. and procedure make total of first N number. E.g N =5 then sum = (1+2+3+4+5) = 15 use while loop.

```
delimiter $$
create procedure proc3( IN n int)
begin
   declare i int default 1;
   declare ans int default 0;

   while i<=n do
      set ans = ans + i ;
      set i = i + 1 ;
   end while;
   select ans;
   end $$
  delimiter ;

 call proc3(10);
+------+
| ans  |
+------+
|   55 |
+------+
```

4. create a procedure called proc4 which pass the student id in parameter and find average of marks of given student id from stud_sub table. E.g call proc2(1)

```
CREATE PROCEDURE proc4(IN s_id INT)
BEGIN
 DECLARE t_marks INT;
 DECLARE t_sub INT;
 DECLARE avg_marks DECIMAL(10,2);
 SELECT sum(marks),count(subid)
 INTO t_marks,t_sub FROM stud_sub
 WHERE sid=s_id;
 IF t_sub>0 THEN
 SET avg_marks=t_marks/t_sub;
 SELECT s_id AS StudentId,avg_marks AS MArks;

 ELSE
 SELECT 'No records found for the given student ID' AS result;
 END IF;
END;
$$
Query OK, 0 rows affected (0.01 sec)
mysql> delimiter ;
```

```
mysql> call proc4(3);
+-----------+-------+
| StudentId | MArks |
+-----------+-------+
| 3 | 61.67 |
+-----------+-------+
```

5. Create procedure called proc5 in which pass the number and display whether number is odd or even. [hint if mod(n,2) = 0 then]

```
delimiter $$
create procedure proc5 ( IN n int)
begin
   if mod(n,2) = 0 then
   select concat(n," is even");
   else
   select concat(n," is odd");
   end if;
end $$
delimiter ;
call proc5(10);
+---------------------+
| concat(n," is even") |
+---------------------+
| 10 is even          |
```

6. create procedure called proc6 which pass the orderid as parameter and find the total quantity order form sales_order_detail, total of quantity order should be stored in OUT parameter

```
delimiter $$

create procedure proc6 ( IN o_id varchar(30) , OUT t_qty int)

begin

  select sum(qtyordered) into t_qty from sales_order_details where orderno = o_id;

end $$


delimiter ;


call proc6("O19001", @t_qty);

select @t_qty;

+--------+

| @t_qty |

+--------+

|      8 |

+--------+
```

Exercise for functions

1. Create a function func1 which takes the number as parameter and return the value “odd” or “even” .

```
DELIMITER $$
CREATE FUNCTION func1(n INT)
RETURNS VARCHAR(30)
DETERMINISTIC
BEGIN
   DECLARE val VARCHAR(30);

   IF MOD(n, 2) = 0 THEN
      SET val = 'even';
   ELSE
      SET val = 'odd';
   END IF;
   RETURN val;
END $$
DELIMITER ;

select func1(5);
+----------+
| func1(5) |
+----------+
| odd      |
+----------+
```

2. Create a function func2() which take the age attribute of employee table, if age is

<=25 status will be “young”, if age between 26 to 32 status ”middle” if age > 32

status will be “old”. Function returns the status. Write a select query which display

the name ,age and status of every employee.

```
DELIMITER $$

CREATE FUNCTION func2(age INT)
RETURNS VARCHAR(30)
DETERMINISTIC
BEGIN
   DECLARE st VARCHAR(30);

   IF age <= 25 THEN
      SET st = 'Young';
   ELSEIF age >= 26 AND age <= 32 THEN
      SET st = 'Middle';
   ELSE
      SET st = 'Old';
   END IF;

   RETURN st;
END $$
```

DELIMITER ;

select name, age, func2(age) from emp

```
+--------------+------+------------+
| name         | age  | func2(age) |
+--------------+------+------------+
| John Doe     |   25 | Young      |
| Jane Smith   |   30 | Middle     |
| Bob Johnson  |   22 | Young      |
| Alice Brown  |   28 | Middle     |
| Charlie Davis |   35 | Old       |
+--------------+------+------------+
```

3. Create a function fun3() which takes a orderno as input and returns the

name(description) of the product . hint(use product_master and sales_order_detail)

Use necessary select query to display function output.

```
CREATE FUNCTION fun3(order_no VARCHAR(6)) RETURNS
VARCHAR(255)
 -> BEGIN
 -> DECLARE product_name VARCHAR(255);
 -> SELECT p.description INTO product_name
 -> FROM product_master p
 -> JOIN sales_order_details sod ON p.productno = sod.productno
```

```
 -> WHERE sod.orderno = order_no
 -> LIMIT 1;
 -> RETURN product_name;
 -> END;
 -> $$

  delimiter ;
```

```
mysql> select fun3('O46866');
+----------------+
| fun3('O46866') |
+----------------+
| Denim Shirts |
+----------------+
```

```
select func3('O46866');
```