Programme Name - BTech-ICT

Semester - IV

Course Name - Database Management System

Project Title - Wholesale Database Management System

| | | |
|---|---|---|
| Group Members - | Prima Sanghvi | AU1741045 |
| | Priyanshi Deliwala | AU1741047 |
| | Aayushi Ganatra | AU1741050 |
| | Shashwat Mehta | AU1741100 |

## **DESCRIPTION :**

Wholesale management system is the process of buying the stocks from the buyers and then selling it to the customers. It acts as a middle person between the buyers and the sellers. The manager has to buy the stock from the buyers. Managing the details that is regarding the whole sale is not possible to be kept in track through the pen paper mode. It requires the database system where we can store the information of the people involved in the whole sale system management. It can be stored clearly in the database with more precision and can avoid the redundant data.

The wholesale management database system can contain the details of the stock that is brought from the buyers like id, name, quantity etc. The information of the buyers like buyer id, name, stock, address, contact number etc can be collected and stored in the database. The manager is responsible for collecting all these information regarding the buyers. In case of any complaints about the stock that is purchased from the particular buyer then you will be able to contact him at ease. The details of the customers like name, customer id, address, contact number who buys the stock that is purchased can also be stored. There will be customers who will be having the pending bills that need to be cleared and can be contacted about the pending payment. The profit per month is calculated after deducting all the expenditures spent. So the profit of every month can also be stored. If you want to know the profit gained five years back, you can get it done through this database. Product cannot be sold to the customer if the amount is not present in the stock. The date of delivery can be maintained up to which the stock can be provided.

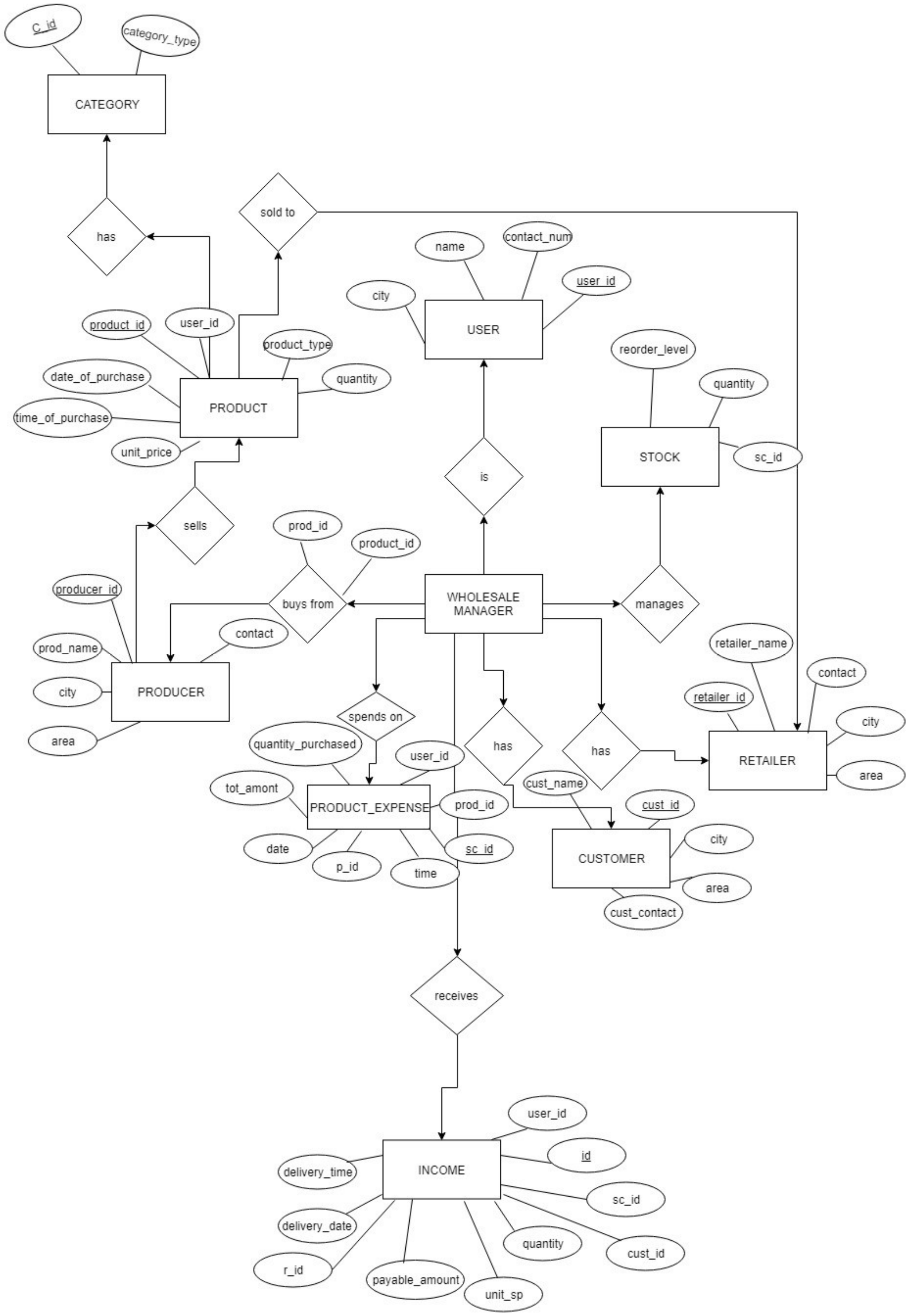
The features that can be included in this application are as follows:

Stocks management system: The details of the stocks that are present or available can be easily maintained through this application.

Manager database management: The details of the manager like name, manager id, address, contact number who buys the stock that is purchased can also be stored.

Sales management: The sales that have taken place on the particular day or month can be calculated through this application.

**Entity-Relationship Diagram :**



## Table Design :

Table Name: User

Description: The table user stores in the values regarding the user that purchases goods

| | Fieldname | Datatype | Size | Constraint | Description |
|---|---|---|---|---|---|
| **user_id** | **user_id** | **varchar (20)** | | **PK** | **Takes a unique id for every user** |
| **name** | **name** | **varchar (50)** | | | **Stores the name of the user** |
| **contact_num** | **contact_num** | **bigInt** | | **Unique** | **Stores the number of the user** |
| **city** | **city** | **varchar (20)** | | **Default ‘Ahmedabad’** | **Stores city for every user** |

Table Name: Producer

Description: The table producer stores in the values regarding the producer that produces all the goods

| | Fieldname | Datatype | Size | Constraint | Description |
|---|---|---|---|---|---|
| **producer_id** | **producer_id** | **varchar (20)** | | **PK** | **Takes a unique id for every producer** |
| **prod_name** | **prod_name** | **varchar (50)** | | | **Stores the name of the producer** |
| **contact** | **contact** | **bigInt** | | **Unique** | **Stores the contact number** |
| **city** | **city** | **varchar (20)** | | **Default ‘Ahmedabad’** | **Stores city for every producer** |
| **area** | **area** | **varchar (20)** | | | **Stores the area for every producer** |

Table Name: Retailer

Description: The table user stores in the values regarding the retailer that purchases goods from the wholesaler.

| | Fieldname | Datatype | Size | Constraint | Description |
|---|---|---|---|---|---|
| **retailer_id** | **retailer_id** | **varchar (20)** | | **PK** | **Takes a unique id for every retailer** |
| **retailer_name** | **retailer_name** | **varchar (50)** | | | **Stores the name of the retailer** |
| **contact** | **contact** | **bigInt** | | **Unique** | **Stores the contact number** |

| | | | | | |
|---|---|---|---|---|---|
| city | city | varchar (20) | | Default 'Ahmedabad' | Stores city for every retailer |
| area | area | varchar (20) | | | Stores the area for every retailer |

Table Name: Customer

Description: The table user stores in the values regarding the customer that purchases goods from the wholesaler.

| | Fieldname | Datatype | Size | Constraint | Description |
|---|---|---|---|---|---|
| cust_id | cust_id | varchar (20) | | PK | Takes a unique id for every customer |
| cust_ name | cust_name | varchar (50) | | | Stores the name of the customer |
| cust_ contact | cust_ contact | bigInt | | Unique | Stores the contact number |
| city | city | varchar (20) | | Default 'Ahmedabad' | Stores city for every customer |
| area | area | varchar (20) | | | Stores the area for every customer |

Table Name: Category

Description: The table user stores in the values regarding the categories of the goods that have been produced by the producer.

| | Fieldname | Datatype | Size | Constraint | Description |
|---|---|---|---|---|---|
| c_id | c_id | varchar (20) | | PK | Takes unique id for each category item |
| category_type | category_ type | varchar (100) | | | Stores the type of item |

Table Name: Sub_category

Description: The table user stores in the values regarding the sub-categories of the goods that have been produced by the producer.

| | Fieldname | Datatype | Size | Constraint | Description |
|---|---|---|---|---|---|
| sc_id | sc_id | varchar (20) | | PK | Takes unique id for each subcategory item |
| product_name | product_ name | varchar (100) | | | Stores the name of the item |

| | | | | | |
|---|---|---|---|---|---|
| **c_id** | **c_id** | **varchar (20)** | | **FK (referencing to category (c_id))** | **Takes unique id for each category item** |

Table Name: Product_expense

Description: The table user stores in the values regarding the expenses that occur according to each categories of the produced products purchased from the producer.

| | **Fieldname** | **Datatype** | **Size** | **Constraint** | **Description** |
|---|---|---|---|---|---|
| **prod_id** | **prod_id** | **varchar (20)** | | **PK** | **Takes unique id for every product** |
| **quantity_purchased** | **quantity_purchased** | **int** | | | **Stores the quantity that is purchased** |
| **date_of_purchase** | **date_of_purchase** | **date** | | **Can be null** | **Stores the date of a purchase** |
| **time_of_purchase** | **time_of_purchase** | **time** | | **Can be null** | **Stores the time of a purchase** |
| **unit_ price** | **unit_ price** | **float** | | | **Stores the price of a single unit** |
| **total_ amount** | **total_ amount** | **bigInt** | | | **Stores the total amount of the purchase** |
| **p_id** | **p_id** | **varchar (20)** | | **FK (referencing to producer (producer_id))** | **Takes id of the producer** |
| **sc_id** | **sc_id** | **varchar (20)** | | **FK (referencing to sub_category(sc_id))** | **Takes id of the subcategory item** |
| **user_id** | **user_id** | **varchar (20)** | | **FK (referencing to user (user_id))** | **Takes id of the user** |

Table Name: Stock_manage

Description: The table user stores in the values regarding the stock management, whether it needs to be reordered or the stock at any moment is sufficient or not.

| | Fieldname | Datatype | Size | Constraint | Description |
|---|---|---|---|---|---|
| quantity_on_hand | quantity_on_hand | int | | FK | Stores the quantity on hand, at a particular moment |
| reorder_level | reorder_level | int | | | Stores the cut-off level for goods, that need to be reordered |
| sc_id | sc_id | varchar (20) | | FK (referencing to sub_category (sc_id)) | Takes unique id for each subcategory item |

Table Name: Income

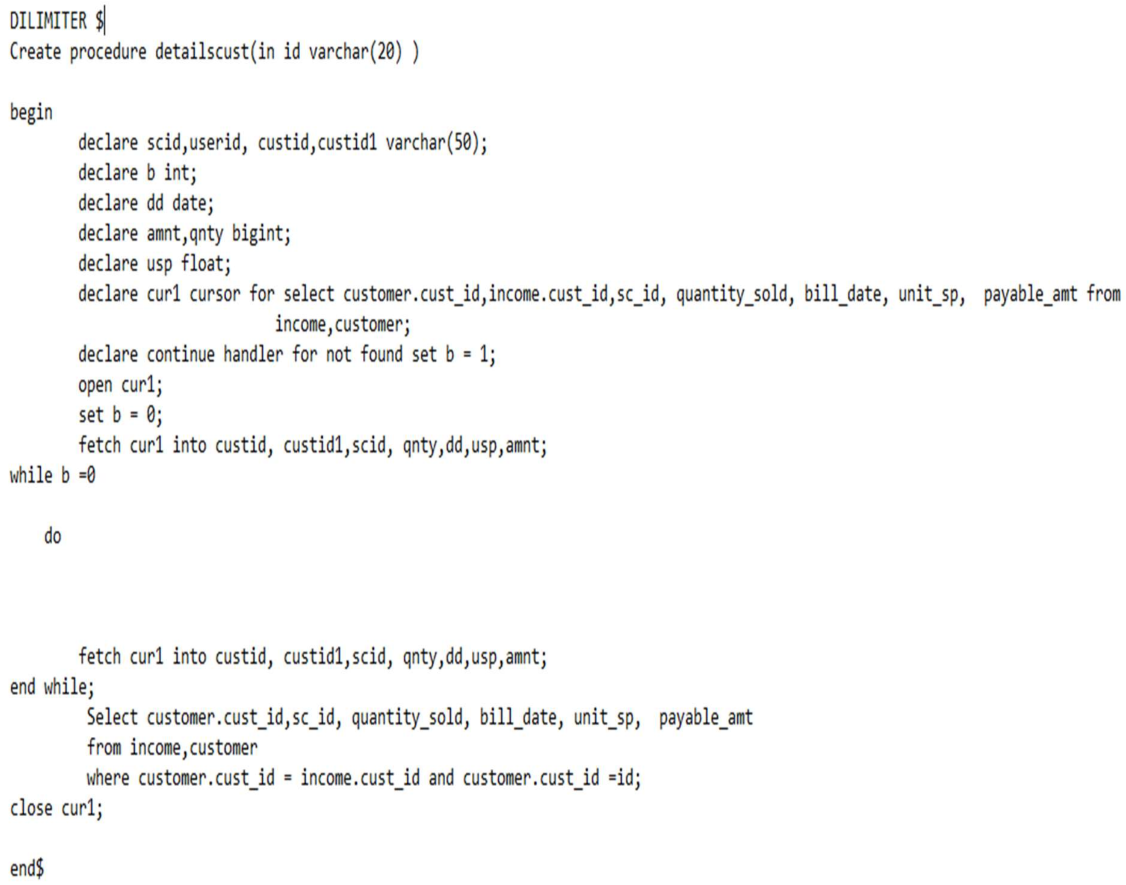
Description: The table user stores in the values regarding the income that comes in, with every product sold to either customer or the retailer.

| | Fieldname | Datatype | Size | Constraint | Description |
|---|---|---|---|---|---|
| id | id | int | | PK | Takes unique id for every product |
| quantiy_sold | quantity_sold | int | | | Stores the quantity that is sold |
| bill_date | bill_date | date | | Can be null | Stores the date of the bill |
| bill_time | bill_time | time | | Can be null | Stores the time of the bill |
| unit_sp | unit_sp | float | | | Stores the price of a single unit |
| payable_amount | payable_amount | bigInt | | | Stores the final payable amount |

| | | | | | |
|---|---|---|---|---|---|
| **r_id** | **r_id** | **varchar (20)** | | **Can be null**<br>**FK (referencing to producer (producer _id))** | **Takes id of the producer** |
| **sc_id** | **sc_id** | **varchar (20)** | | **FK (referencing to sub_category(sc_id))** | **Takes id of the subcategory item** |
| **cust_id** | **cust_id** | **varchar (20)** | | **Can be null**<br>**FK (referencing to customer (cust_id))** | |
| **user_id** | **user_id** | **varchar (20)** | | **FK (referencing to user (user_id))** | **Takes id of the user** |

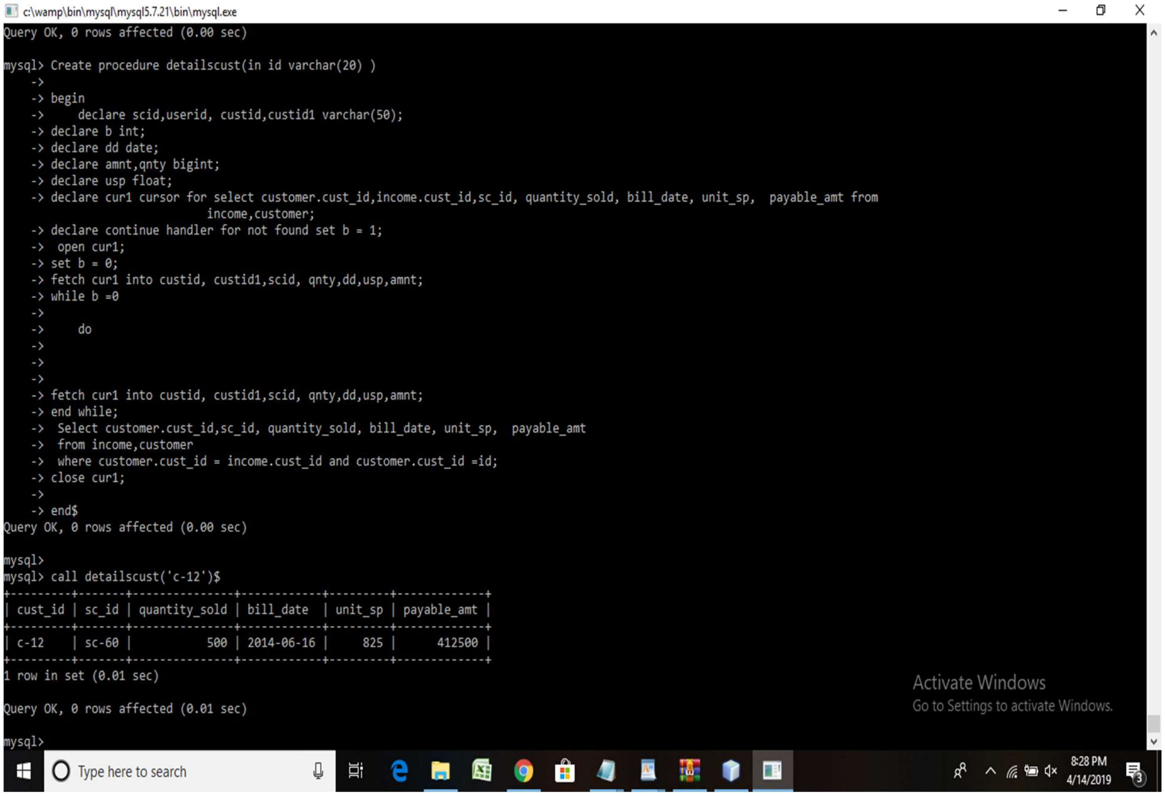
## Stored Procedure, Functions and Triggers :

Procedure 1 : Bill Summary of Customer



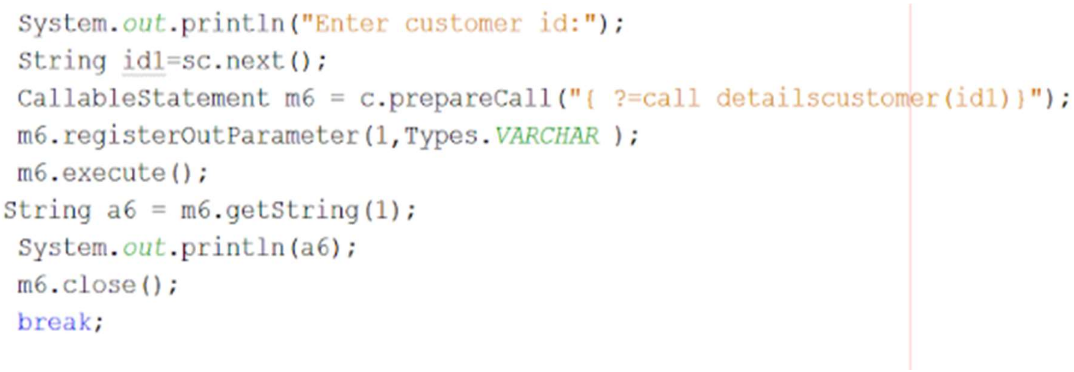
```
DILIMITER $
Create procedure detailscust(in id varchar(20) )

begin
        declare scid,userid, custid,custid1 varchar(50);
        declare b int;
        declare dd date;
        declare amnt,qnty bigint;
        declare usp float;
        declare cur1 cursor for select customer.cust_id,income.cust_id,sc_id, quantity_sold, bill_date, unit_sp,  payable_amt from
                              income,customer;
        declare continue handler for not found set b = 1;
        open cur1;
        set b = 0;
        fetch cur1 into custid, custid1,scid, qnty,dd,usp,amnt;
while b =0

    do



        fetch cur1 into custid, custid1,scid, qnty,dd,usp,amnt;
end while;
         Select customer.cust_id,sc_id, quantity_sold, bill_date, unit_sp,  payable_amt
         from income,customer
         where customer.cust_id = income.cust_id and customer.cust_id =id;
close cur1;

end$
```

Procedure 1 : OUTPUT


c:\wamp\bin\mysql\mysql5.7.21\bin\mysql.exe
Activate Windows
Go to Settings to activate Windows.
Type here to search
8:28 PM
4/14/2019

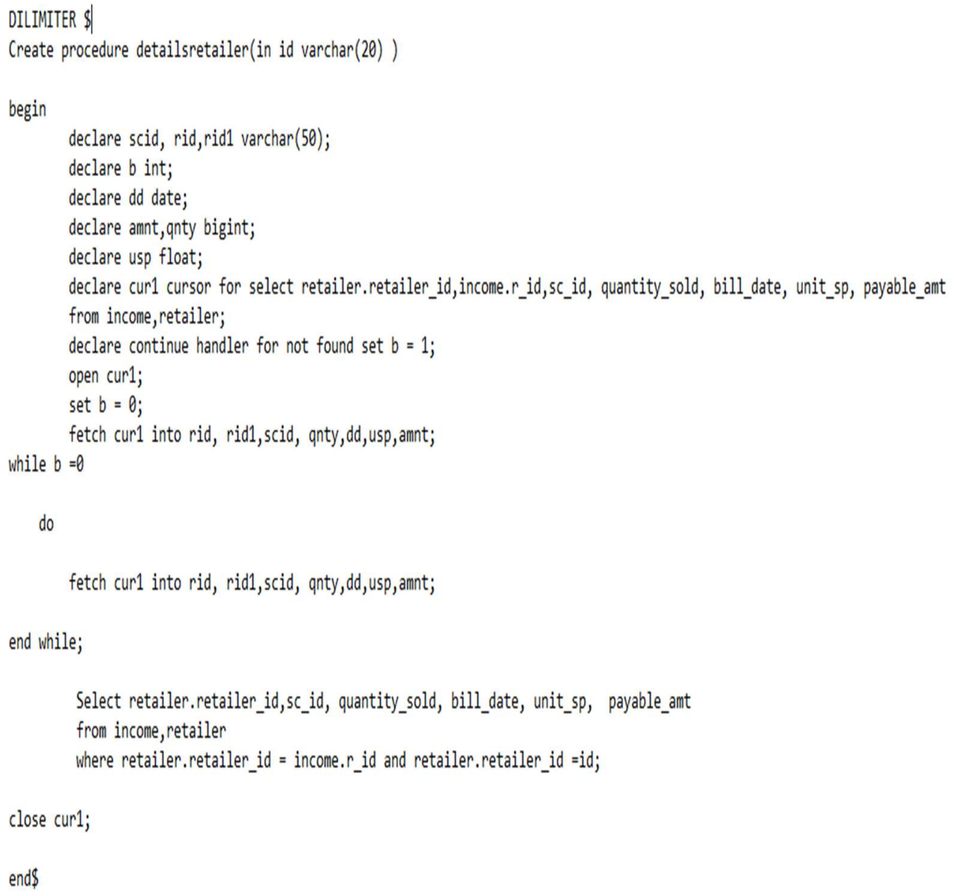
```
Query OK, 0 rows affected (0.00 sec)

mysql> Create procedure detailscust(in id varchar(20) )
    ->
    -> begin
    ->     declare scid,userid, custid,custid1 varchar(50);
    -> declare b int;
    -> declare dd date;
    -> declare amnt,qnty bigint;
    -> declare usp float;
    -> declare cur1 cursor for select customer.cust_id,income.cust_id,sc_id, quantity_sold, bill_date, unit_sp,  payable_amt from
                           income,customer;
    -> declare continue handler for not found set b = 1;
    ->  open cur1;
    -> set b = 0;
    -> fetch cur1 into custid, custid1,scid, qnty,dd,usp,amnt;
    -> while b =0
    ->
    ->     do
    ->
    ->
    ->
    -> fetch cur1 into custid, custid1,scid, qnty,dd,usp,amnt;
    -> end while;
    ->  Select customer.cust_id,sc_id, quantity_sold, bill_date, unit_sp,  payable_amt
    ->  from income,customer
    ->  where customer.cust_id = income.cust_id and customer.cust_id =id;
    -> close cur1;
    ->
    -> end$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> call detailscust('c-12')$
+---------+-------+---------------+------------+---------+-------------+
| cust_id | sc_id | quantity_sold | bill_date  | unit_sp | payable_amt |
+---------+-------+---------------+------------+---------+-------------+
| c-12    | sc-60 |           500 | 2014-06-16 |     825 |      412500 |
+---------+-------+---------------+------------+---------+-------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>
```

CUI :



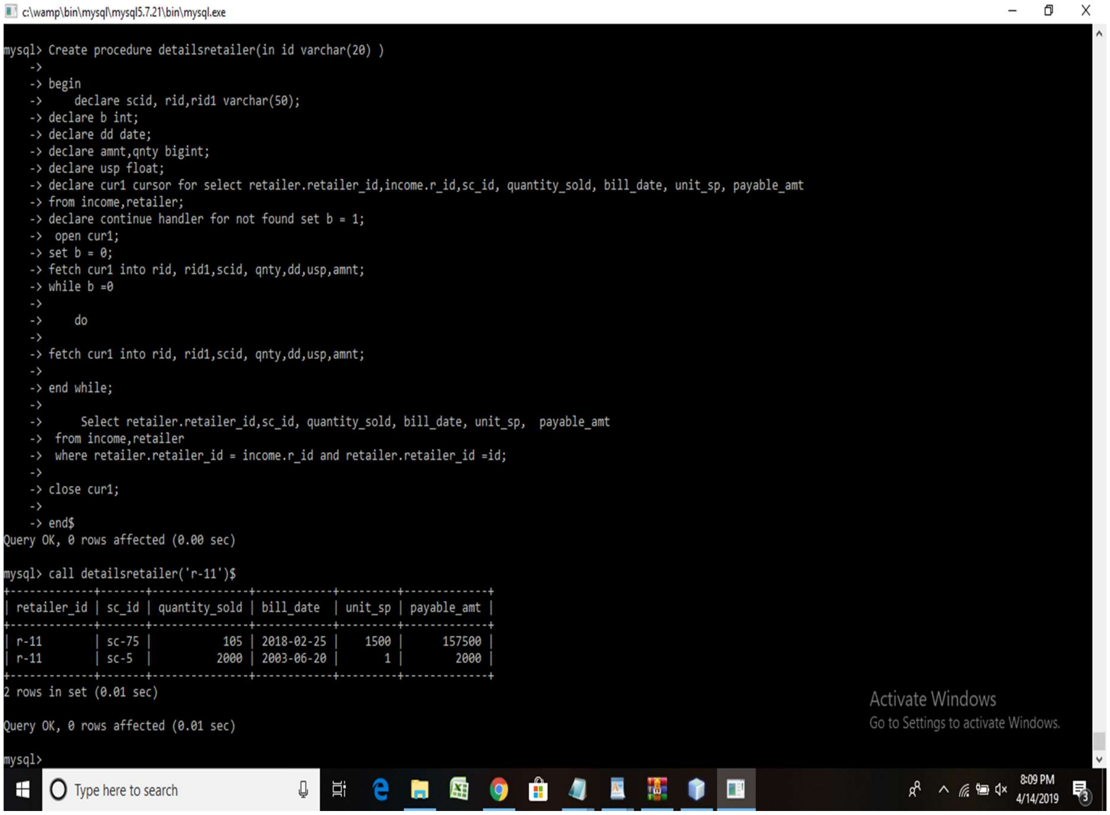
```
 System.out.println("Enter customer id:");
 String id1=sc.next();
 CallableStatement m6 = c.prepareCall("{ ?=call detailscustomer(id1)}");
 m6.registerOutParameter(1,Types.VARCHAR );
 m6.execute();
String a6 = m6.getString(1);
 System.out.println(a6);
 m6.close();
 break;
```

Procedure 2 : Bill Summary of Retailer



```
DILIMITER $
Create procedure detailsretailer(in id varchar(20) )

begin
        declare scid, rid,rid1 varchar(50);
        declare b int;
        declare dd date;
        declare amnt,qnty bigint;
        declare usp float;
        declare cur1 cursor for select retailer.retailer_id,income.r_id,sc_id, quantity_sold, bill_date, unit_sp, payable_amt
        from income,retailer;
        declare continue handler for not found set b = 1;
        open cur1;
        set b = 0;
        fetch cur1 into rid, rid1,scid, qnty,dd,usp,amnt;
while b =0

    do

        fetch cur1 into rid, rid1,scid, qnty,dd,usp,amnt;

end while;

         Select retailer.retailer_id,sc_id, quantity_sold, bill_date, unit_sp,  payable_amt
         from income,retailer
         where retailer.retailer_id = income.r_id and retailer.retailer_id =id;

close cur1;

end$
```

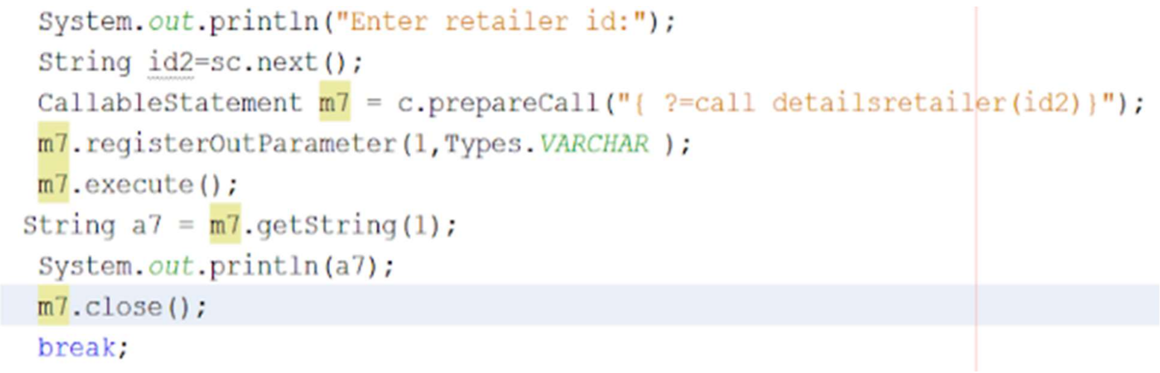
Procedure 2 : OUTPUT



c:\wamp\bin\mysql\mysql5.7.21\bin\mysql.exe
Activate Windows
Go to Settings to activate Windows.
Type here to search
8:09 PM
4/14/2019

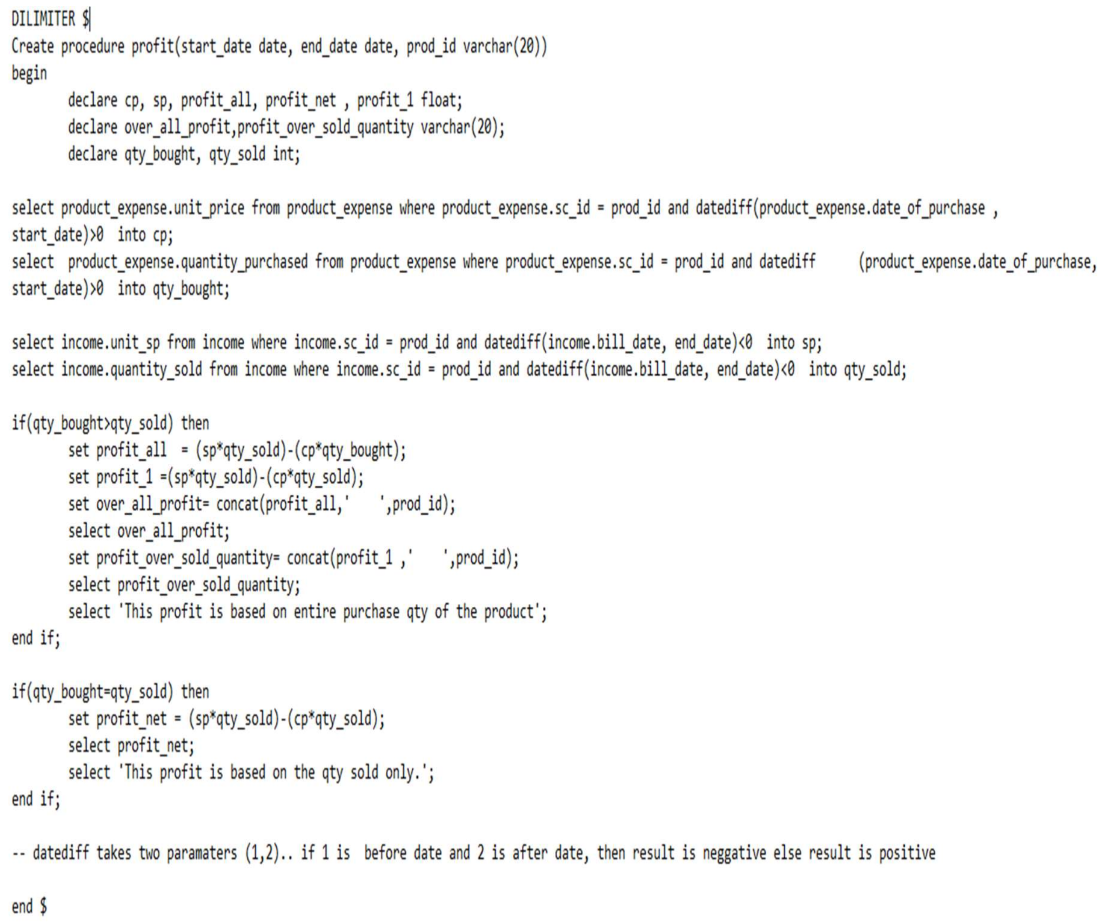
```
mysql> Create procedure detailsretailer(in id varchar(20) )
    ->
    -> begin
    ->     declare scid, rid,rid1 varchar(50);
    -> declare b int;
    -> declare dd date;
    -> declare amnt,qnty bigint;
    -> declare usp float;
    -> declare cur1 cursor for select retailer.retailer_id,income.r_id,sc_id, quantity_sold, bill_date, unit_sp, payable_amt
    -> from income,retailer;
    -> declare continue handler for not found set b = 1;
    ->  open cur1;
    -> set b = 0;
    -> fetch cur1 into rid, rid1,scid, qnty,dd,usp,amnt;
    -> while b =0
    ->
    ->     do
    ->
    -> fetch cur1 into rid, rid1,scid, qnty,dd,usp,amnt;
    ->
    -> end while;
    ->
    ->      Select retailer.retailer_id,sc_id, quantity_sold, bill_date, unit_sp,  payable_amt
    ->  from income,retailer
    ->  where retailer.retailer_id = income.r_id and retailer.retailer_id =id;
    ->
    -> close cur1;
    ->
    -> end$
Query OK, 0 rows affected (0.00 sec)

mysql> call detailsretailer('r-11')$
+-------------+-------+---------------+------------+---------+-------------+
| retailer_id | sc_id | quantity_sold | bill_date  | unit_sp | payable_amt |
+-------------+-------+---------------+------------+---------+-------------+
| r-11        | sc-75 |           105 | 2018-02-25 |    1500 |      157500 |
| r-11        | sc-5  |          2000 | 2003-06-20 |       1 |        2000 |
+-------------+-------+---------------+------------+---------+-------------+
2 rows in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>
```

CUI :



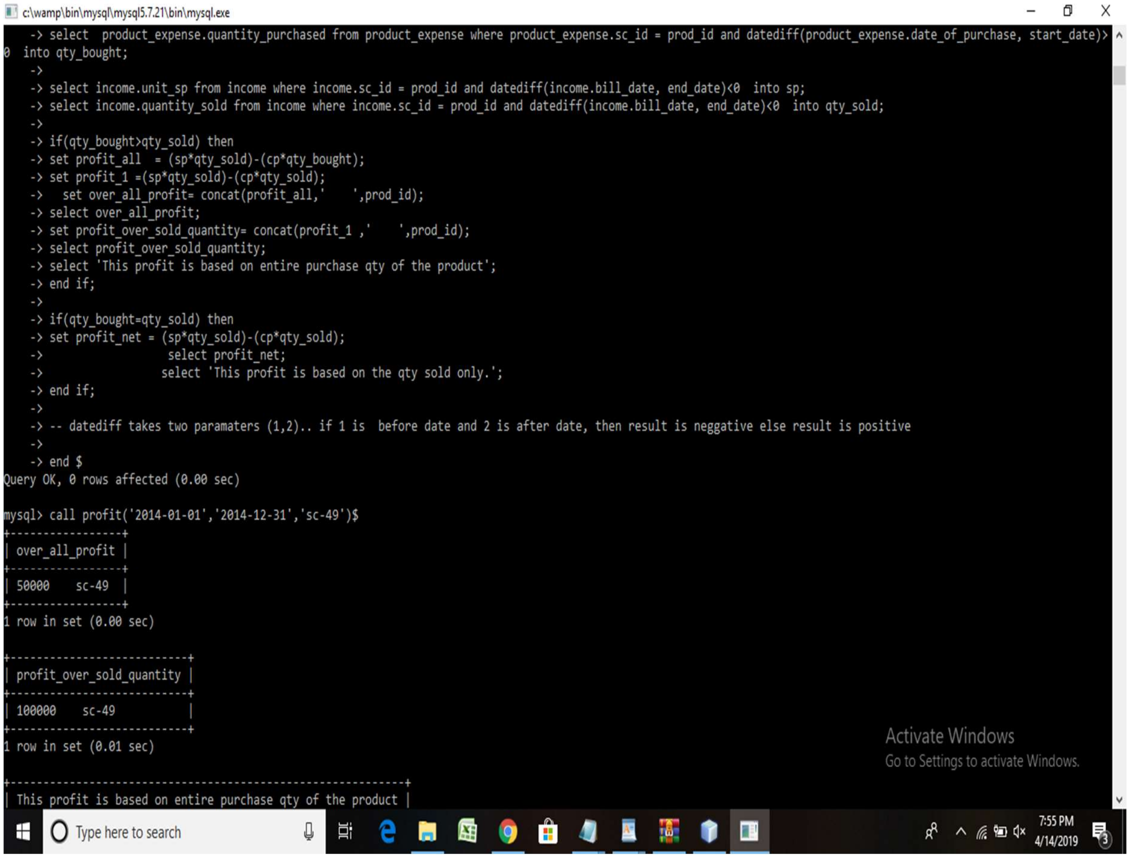
```
  System.out.println("Enter retailer id:");
  String id2=sc.next();
  CallableStatement m7 = c.prepareCall("{ ?=call detailsretailer(id2)}");
  m7.registerOutParameter(1,Types.VARCHAR );
  m7.execute();
 String a7 = m7.getString(1);
  System.out.println(a7);
  m7.close();
  break;
```

Procedure 3 : To Calculate profit over specific sale



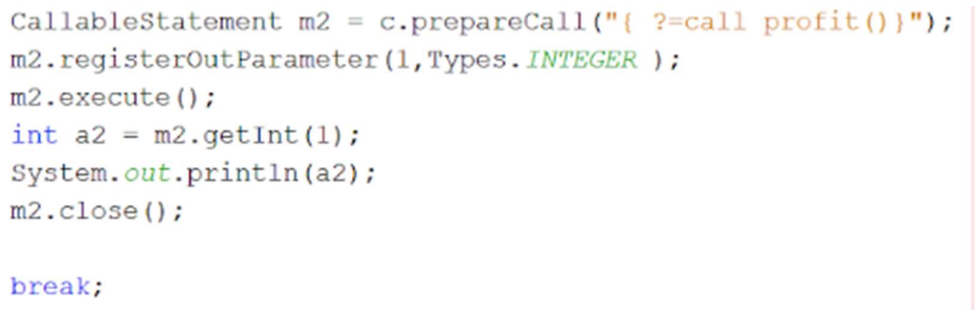
```
DILIMITER $
Create procedure profit(start_date date, end_date date, prod_id varchar(20))
begin
        declare cp, sp, profit_all, profit_net , profit_1 float;
        declare over_all_profit,profit_over_sold_quantity varchar(20);
        declare qty_bought, qty_sold int;

select product_expense.unit_price from product_expense where product_expense.sc_id = prod_id and datediff(product_expense.date_of_purchase ,
start_date)>0  into cp;
select  product_expense.quantity_purchased from product_expense where product_expense.sc_id = prod_id and datediff      (product_expense.date_of_purchase,
start_date)>0  into qty_bought;

select income.unit_sp from income where income.sc_id = prod_id and datediff(income.bill_date, end_date)<0  into sp;
select income.quantity_sold from income where income.sc_id = prod_id and datediff(income.bill_date, end_date)<0  into qty_sold;

if(qty_bought>qty_sold) then
        set profit_all  = (sp*qty_sold)-(cp*qty_bought);
        set profit_1 =(sp*qty_sold)-(cp*qty_sold);
        set over_all_profit= concat(profit_all,'    ',prod_id);
        select over_all_profit;
        set profit_over_sold_quantity= concat(profit_1 ,'    ',prod_id);
        select profit_over_sold_quantity;
        select 'This profit is based on entire purchase qty of the product';
end if;

if(qty_bought=qty_sold) then
        set profit_net = (sp*qty_sold)-(cp*qty_sold);
        select profit_net;
        select 'This profit is based on the qty sold only.';
end if;

-- datediff takes two paramaters (1,2).. if 1 is  before date and 2 is after date, then result is neggative else result is positive

end $
```

## Procedure 3 : OUTPUT



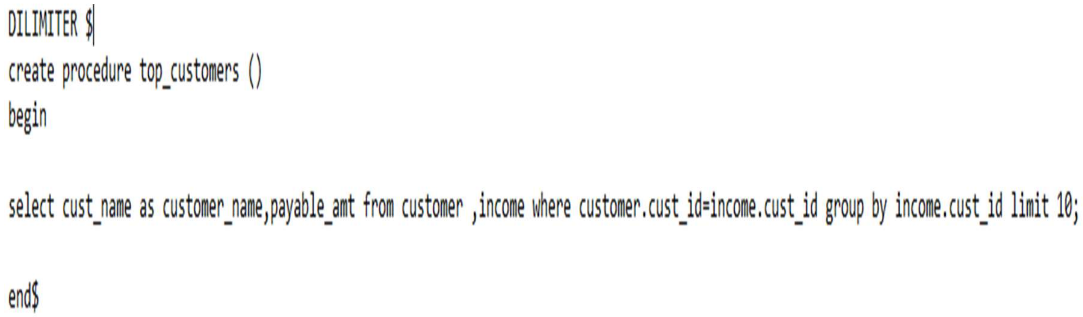
```
    -> select  product_expense.quantity_purchased from product_expense where product_expense.sc_id = prod_id and datediff(product_expense.date_of_purchase, start_date)>
0  into qty_bought;
    ->
    -> select income.unit_sp from income where income.sc_id = prod_id and datediff(income.bill_date, end_date)<0  into sp;
    -> select income.quantity_sold from income where income.sc_id = prod_id and datediff(income.bill_date, end_date)<0  into qty_sold;
    ->
    -> if(qty_bought>qty_sold) then
    -> set profit_all  = (sp*qty_sold)-(cp*qty_bought);
    -> set profit_1 =(sp*qty_sold)-(cp*qty_sold);
    ->   set over_all_profit= concat(profit_all,'    ',prod_id);
    -> select over_all_profit;
    -> set profit_over_sold_quantity= concat(profit_1 ,'    ',prod_id);
    -> select profit_over_sold_quantity;
    -> select 'This profit is based on entire purchase qty of the product';
    -> end if;
    ->
    -> if(qty_bought=qty_sold) then
    -> set profit_net = (sp*qty_sold)-(cp*qty_sold);
    ->                  select profit_net;
    ->                 select 'This profit is based on the qty sold only.';
    -> end if;
    ->
    -> -- datediff takes two paramaters (1,2).. if 1 is  before date and 2 is after date, then result is neggative else result is positive
    ->
    -> end $
Query OK, 0 rows affected (0.00 sec)

mysql> call profit('2014-01-01','2014-12-31','sc-49')$
+-----------------+
| over_all_profit |
+-----------------+
| 50000    sc-49  |
+-----------------+
1 row in set (0.00 sec)

+---------------------------+
| profit_over_sold_quantity |
+---------------------------+
| 100000    sc-49           |
+---------------------------+
1 row in set (0.01 sec)

+------------------------------------------------------------+
| This profit is based on entire purchase qty of the product |
```

CUI :



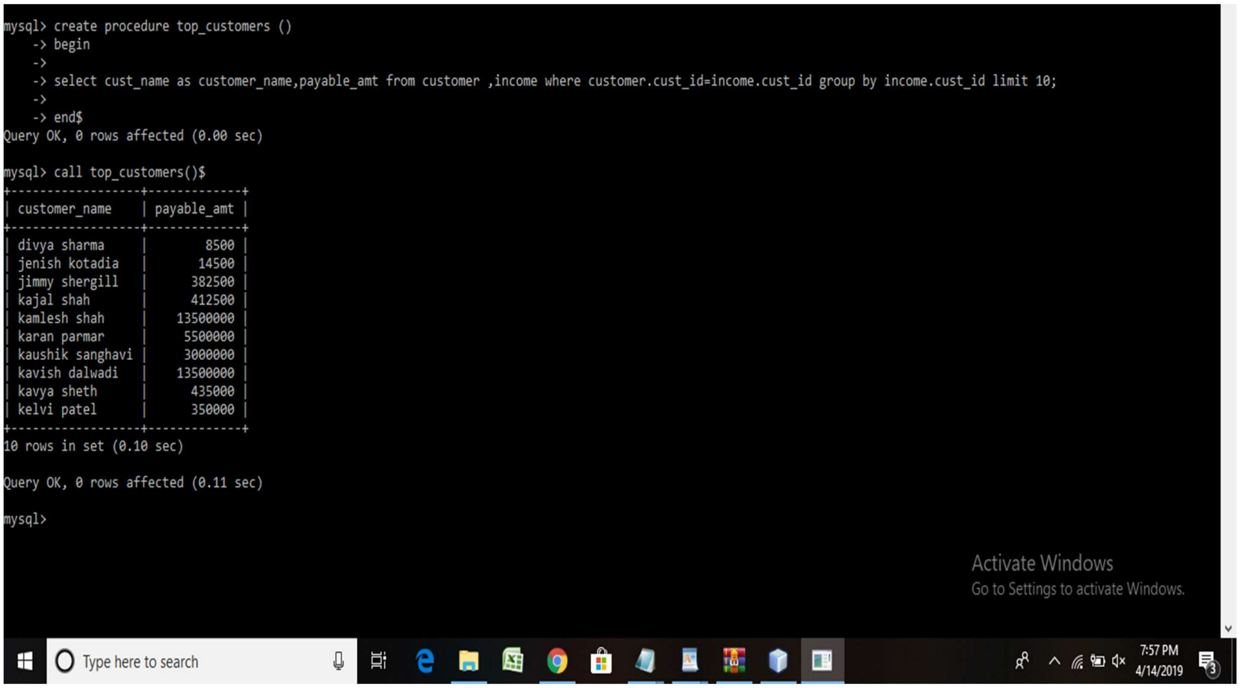
```
CallableStatement m2 = c.prepareCall("{ ?=call profit()}");
m2.registerOutParameter(1,Types.INTEGER );
m2.execute();
int a2 = m2.getInt(1);
System.out.println(a2);
m2.close();

break;
```

Procedure 4 : To display top 10 Customers



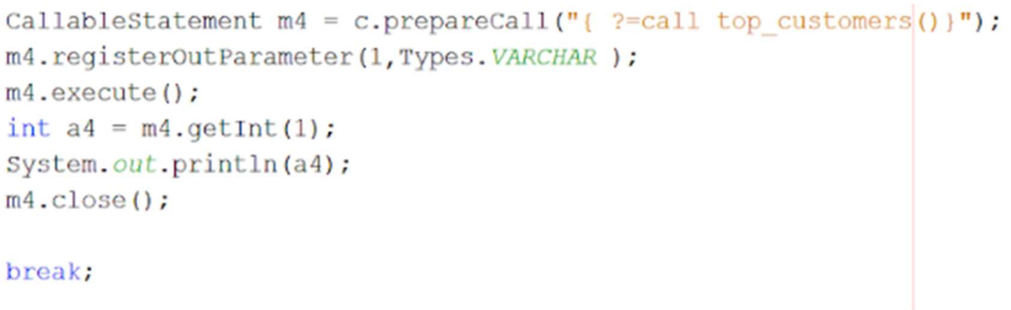
```
DILIMITER $
create procedure top_customers ()
begin

select cust_name as customer_name,payable_amt from customer ,income where customer.cust_id=income.cust_id group by income.cust_id limit 10;

end$
```

Procedure 4 : OUTPUT



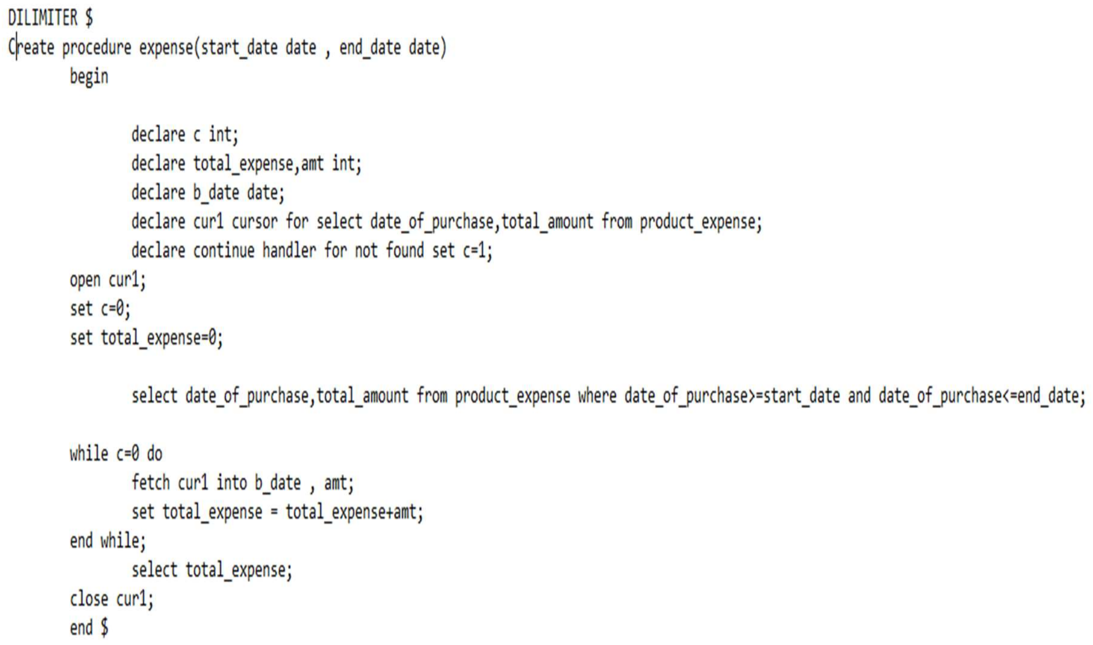
```
mysql> create procedure top_customers ()
    -> begin
    ->
    -> select cust_name as customer_name,payable_amt from customer ,income where customer.cust_id=income.cust_id group by income.cust_id limit 10;
    ->
    -> end$
Query OK, 0 rows affected (0.00 sec)

mysql> call top_customers()$
+------------------+-------------+
| customer_name    | payable_amt |
+------------------+-------------+
| divya sharma     |        8500 |
| jenish kotadia   |       14500 |
| jimmy shergill   |      382500 |
| kajal shah       |      412500 |
| kamlesh shah     |    13500000 |
| karan parmar     |     5500000 |
| kaushik sanghavi |     3000000 |
| kavish dalwadi   |    13500000 |
| kavya sheth      |      435000 |
| kelvi patel      |      350000 |
+------------------+-------------+
10 rows in set (0.10 sec)

Query OK, 0 rows affected (0.11 sec)

mysql>
```

CUI :



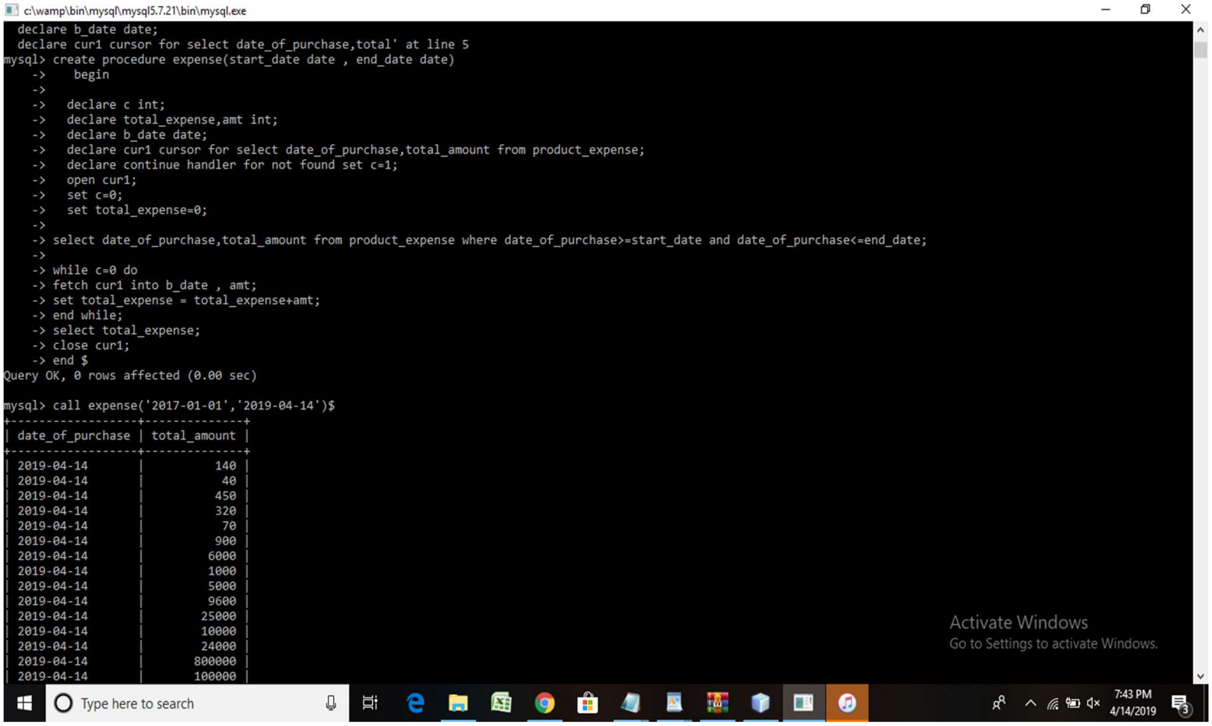
```
CallableStatement m4 = c.prepareCall("{ ?=call top_customers()}");
m4.registerOutParameter(1,Types.VARCHAR );
m4.execute();
int a4 = m4.getInt(1);
System.out.println(a4);
m4.close();

break;
```

Procedure 5 : To display Total expense

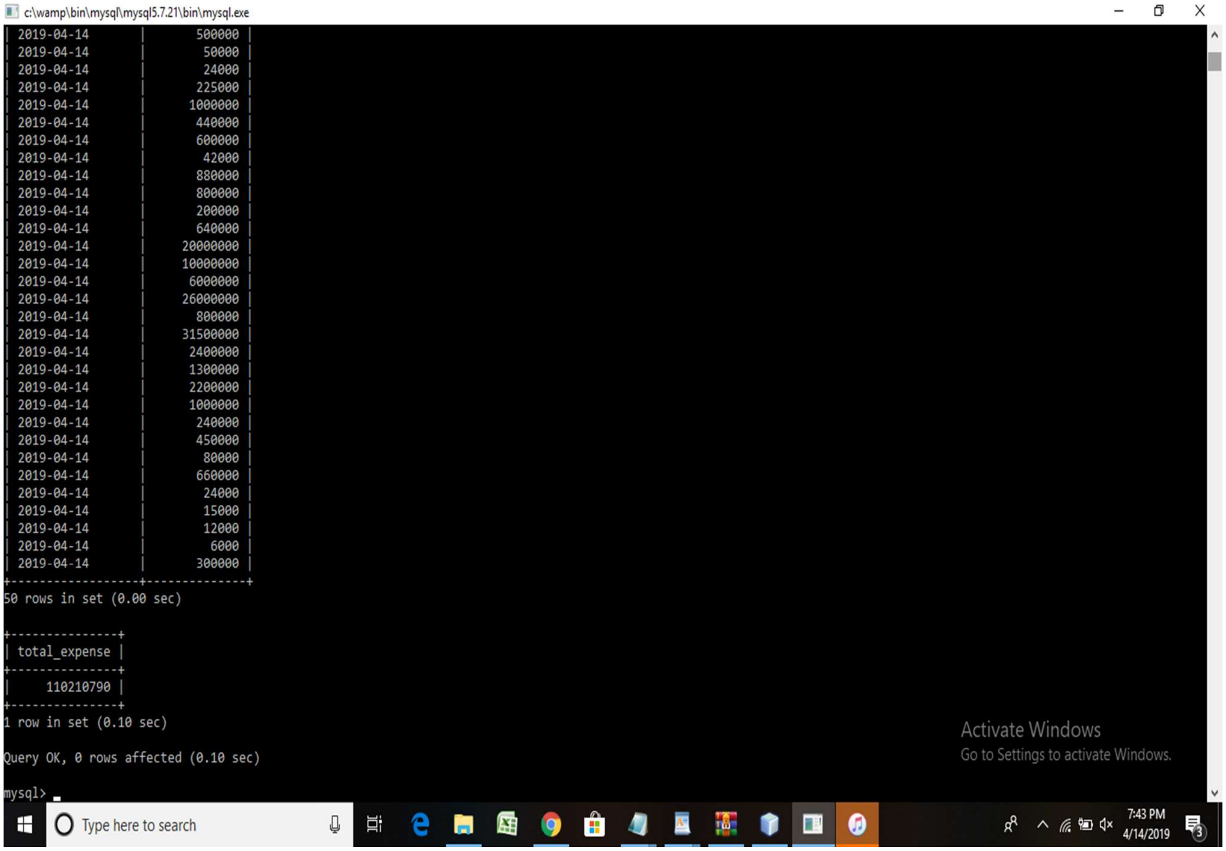


```
DILIMITER $
Create procedure expense(start_date date , end_date date)
        begin

                declare c int;
                declare total_expense,amt int;
                declare b_date date;
                declare cur1 cursor for select date_of_purchase,total_amount from product_expense;
                declare continue handler for not found set c=1;
        open cur1;
        set c=0;
        set total_expense=0;

                select date_of_purchase,total_amount from product_expense where date_of_purchase>=start_date and date_of_purchase<=end_date;

        while c=0 do
                fetch cur1 into b_date , amt;
                set total_expense = total_expense+amt;
        end while;
                select total_expense;
        close cur1;
        end $
```

Procedure 5 : OUTPUT

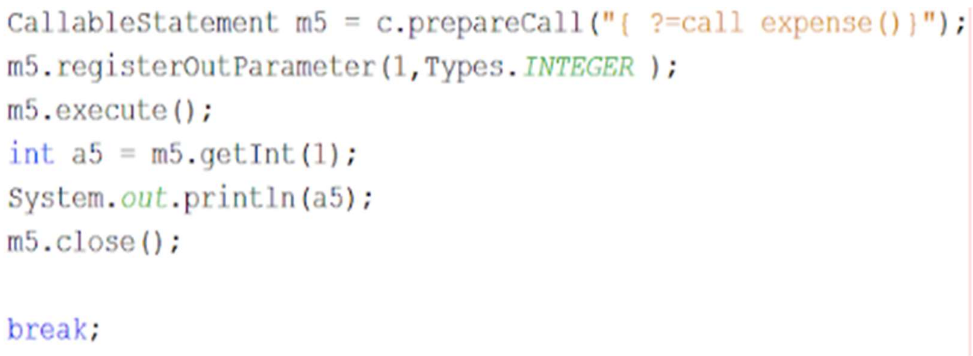


```
  declare b_date date;
  declare cur1 cursor for select date_of_purchase,total' at line 5
mysql> create procedure expense(start_date date , end_date date)
    ->    begin
    ->
    ->   declare c int;
    ->   declare total_expense,amt int;
    ->   declare b_date date;
    ->   declare cur1 cursor for select date_of_purchase,total_amount from product_expense;
    ->   declare continue handler for not found set c=1;
    ->   open cur1;
    ->   set c=0;
    ->   set total_expense=0;
    ->
    -> select date_of_purchase,total_amount from product_expense where date_of_purchase>=start_date and date_of_purchase<=end_date;
    ->
    -> while c=0 do
    -> fetch cur1 into b_date , amt;
    -> set total_expense = total_expense+amt;
    -> end while;
    -> select total_expense;
    -> close cur1;
    -> end $
Query OK, 0 rows affected (0.00 sec)

mysql> call expense('2017-01-01','2019-04-14')$
+------------------+--------------+
| date_of_purchase | total_amount |
+------------------+--------------+
| 2019-04-14       |          140 |
| 2019-04-14       |           40 |
| 2019-04-14       |          450 |
| 2019-04-14       |          320 |
| 2019-04-14       |           70 |
| 2019-04-14       |          900 |
| 2019-04-14       |         6000 |
| 2019-04-14       |         1000 |
| 2019-04-14       |         5000 |
| 2019-04-14       |         9600 |
| 2019-04-14       |        25000 |
| 2019-04-14       |        10000 |
| 2019-04-14       |        24000 |
| 2019-04-14       |       800000 |
| 2019-04-14       |       100000 |
```


c:\wamp\bin\mysql\mysql5.7.21\bin\mysql.exe
Activate Windows
Go to Settings to activate Windows.
Type here to search
7:43 PM
4/14/2019

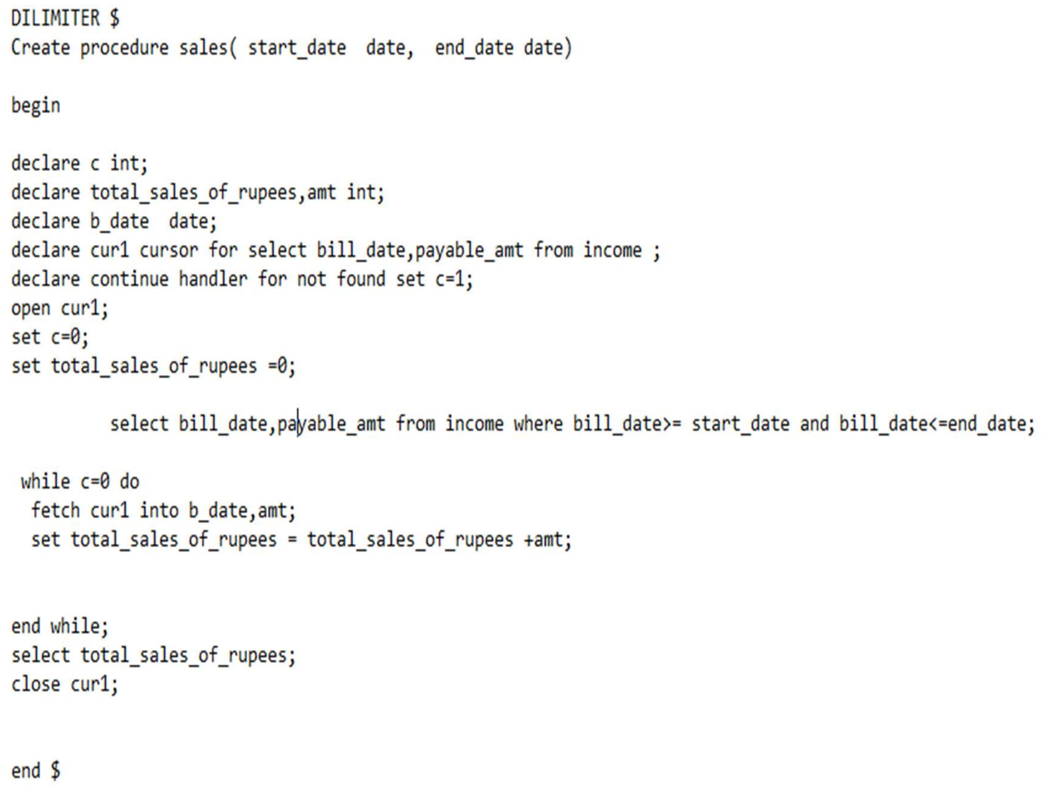
```
| 2019-04-14       |       500000 |
| 2019-04-14       |        50000 |
| 2019-04-14       |        24000 |
| 2019-04-14       |       225000 |
| 2019-04-14       |      1000000 |
| 2019-04-14       |       440000 |
| 2019-04-14       |       600000 |
| 2019-04-14       |        42000 |
| 2019-04-14       |       880000 |
| 2019-04-14       |       800000 |
| 2019-04-14       |       200000 |
| 2019-04-14       |       640000 |
| 2019-04-14       |     20000000 |
| 2019-04-14       |     10000000 |
| 2019-04-14       |      6000000 |
| 2019-04-14       |     26000000 |
| 2019-04-14       |       800000 |
| 2019-04-14       |     31500000 |
| 2019-04-14       |      2400000 |
| 2019-04-14       |      1300000 |
| 2019-04-14       |      2200000 |
| 2019-04-14       |      1000000 |
| 2019-04-14       |       240000 |
| 2019-04-14       |       450000 |
| 2019-04-14       |        80000 |
| 2019-04-14       |       660000 |
| 2019-04-14       |        24000 |
| 2019-04-14       |        15000 |
| 2019-04-14       |        12000 |
| 2019-04-14       |         6000 |
| 2019-04-14       |       300000 |
+------------------+--------------+
50 rows in set (0.00 sec)

+---------------+
| total_expense |
+---------------+
|     110210790 |
+---------------+
1 row in set (0.10 sec)

Query OK, 0 rows affected (0.10 sec)

mysql> _
```

CUI :



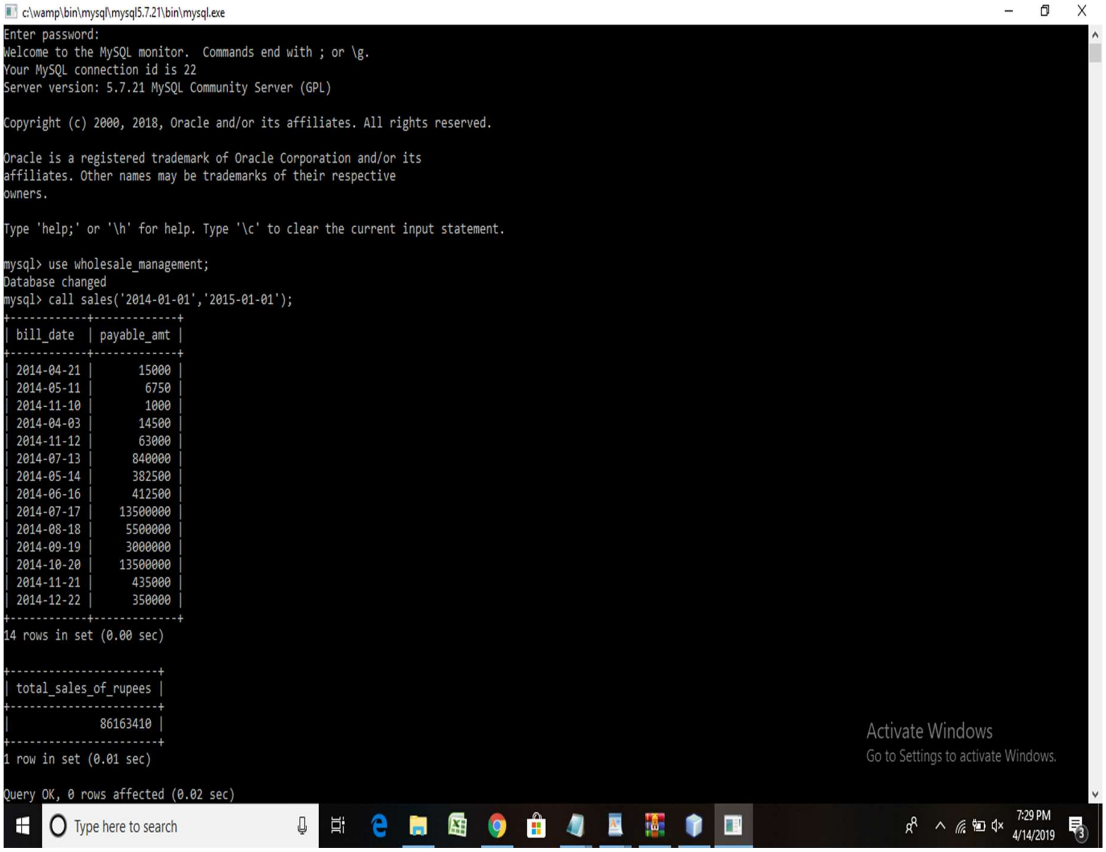
```
CallableStatement m5 = c.prepareCall("{ ?=call expense()}");
m5.registerOutParameter(1,Types.INTEGER );
m5.execute();
int a5 = m5.getInt(1);
System.out.println(a5);
m5.close();

break;
```

Procedure 6 : To display total sales



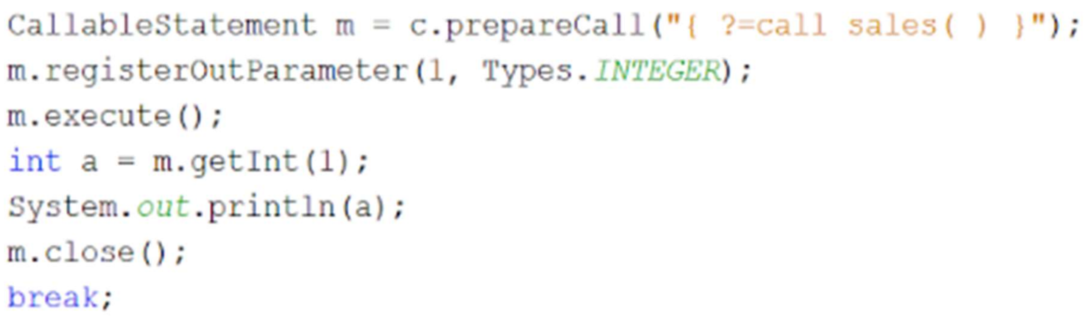
```
DILIMITER $
Create procedure sales( start_date  date,  end_date date)

begin

declare c int;
declare total_sales_of_rupees,amt int;
declare b_date  date;
declare cur1 cursor for select bill_date,payable_amt from income ;
declare continue handler for not found set c=1;
open cur1;
set c=0;
set total_sales_of_rupees =0;

          select bill_date,payable_amt from income where bill_date>= start_date and bill_date<=end_date;

 while c=0 do
  fetch cur1 into b_date,amt;
  set total_sales_of_rupees = total_sales_of_rupees +amt;


end while;
select total_sales_of_rupees;
close cur1;


end $
```

Procedure 6 : OUTPUT


c:\wamp\bin\mysql\mysql5.7.21\bin\mysql.exe
Activate Windows
Go to Settings to activate Windows.
Type here to search
7:29 PM
4/14/2019

```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 22
Server version: 5.7.21 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use wholesale_management;
Database changed
mysql> call sales('2014-01-01','2015-01-01');
+------------+-------------+
| bill_date  | payable_amt |
+------------+-------------+
| 2014-04-21 |       15000 |
| 2014-05-11 |        6750 |
| 2014-11-10 |        1000 |
| 2014-04-03 |       14500 |
| 2014-11-12 |       63000 |
| 2014-07-13 |      840000 |
| 2014-05-14 |      382500 |
| 2014-06-16 |      412500 |
| 2014-07-17 |    13500000 |
| 2014-08-18 |     5500000 |
| 2014-09-19 |     3000000 |
| 2014-10-20 |    13500000 |
| 2014-11-21 |      435000 |
| 2014-12-22 |      350000 |
+------------+-------------+
14 rows in set (0.00 sec)

+----------------------+
| total_sales_of_rupees |
+----------------------+
|             86163410 |
+----------------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.02 sec)
```

CUI :

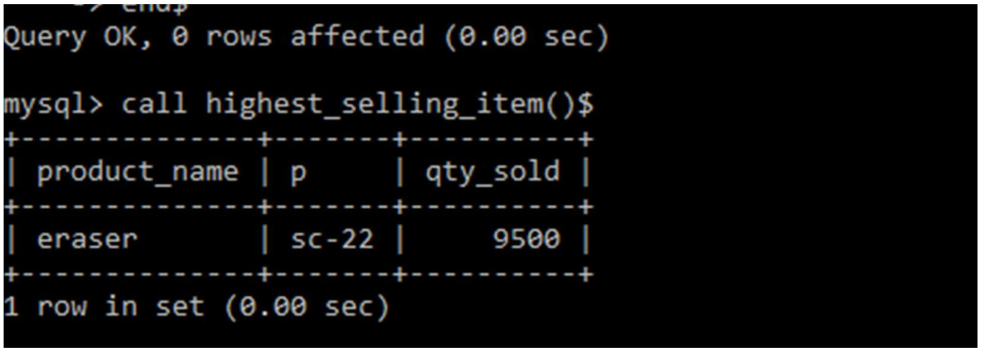


```
CallableStatement m = c.prepareCall("{ ?=call sales( ) }");
m.registerOutParameter(1, Types.INTEGER);
m.execute();
int a = m.getInt(1);
System.out.println(a);
m.close();
break;
```

Procedure 7 : To find the maximum product sold

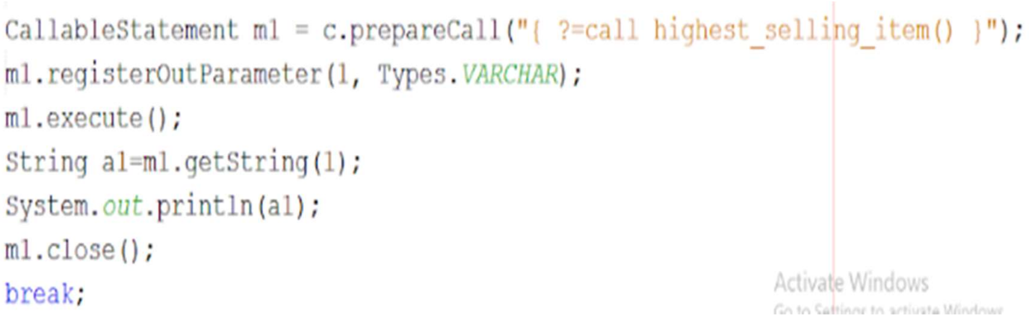
```
DILIMITER $
Create procedure highest_selling_item()

begin

select sub_category.product_name,income.sc_id as p, sum(quantity_sold) as qty_sold
from income,sub_category
where sub_category.sc_id = income.sc_id
group by income.sc_id order by qty_sold desc limit 1;

end$
```

Procedure 7 : OUTPUT



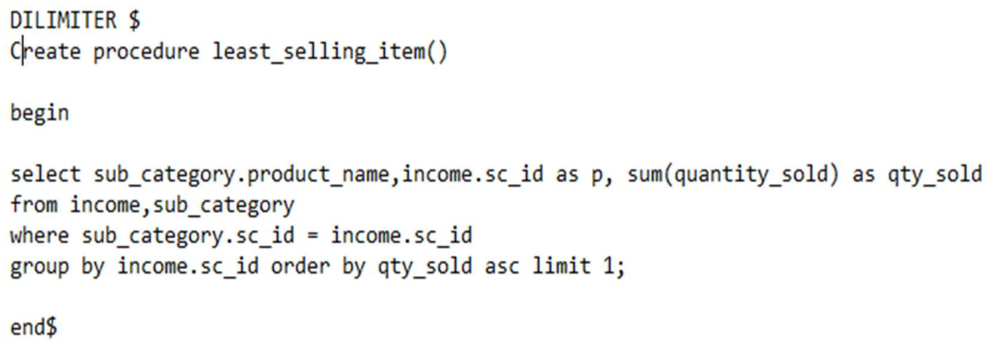
```
    -> end$
Query OK, 0 rows affected (0.00 sec)

mysql> call highest_selling_item()$
+--------------+-------+----------+
| product_name | p     | qty_sold |
+--------------+-------+----------+
| eraser       | sc-22 |     9500 |
+--------------+-------+----------+
1 row in set (0.00 sec)
```

CUI :



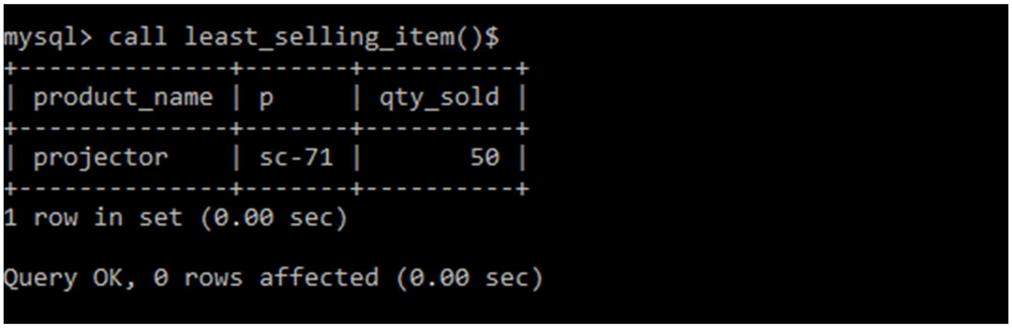
```
CallableStatement m1 = c.prepareCall("{ ?=call highest_selling_item() }");
m1.registerOutParameter(1, Types.VARCHAR);
m1.execute();
String a1=m1.getString(1);
System.out.println(a1);
m1.close();
break;
```

Procedure 8 : To find the least sold product



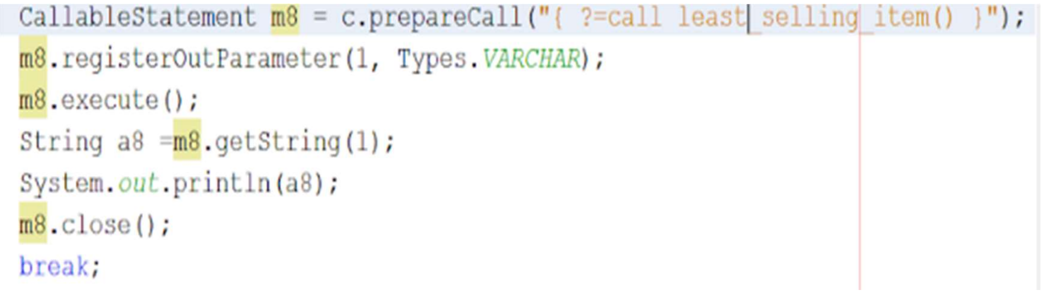
```
DILIMITER $
Create procedure least_selling_item()

begin

select sub_category.product_name,income.sc_id as p, sum(quantity_sold) as qty_sold
from income,sub_category
where sub_category.sc_id = income.sc_id
group by income.sc_id order by qty_sold asc limit 1;

end$
```

Procedure 8 : OUTPUT



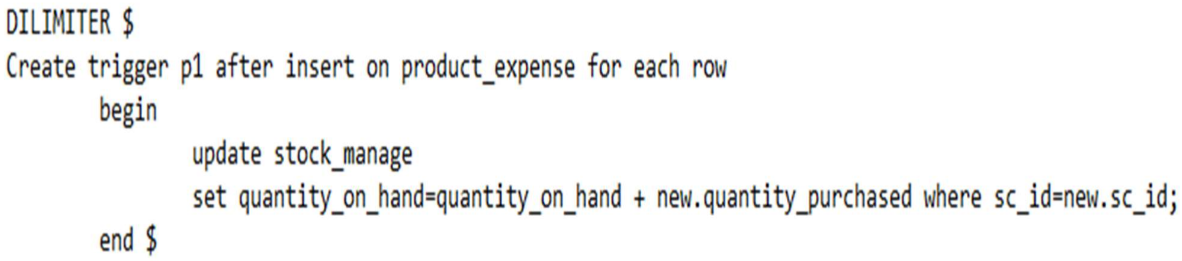
```
mysql> call least_selling_item()$
+--------------+-------+----------+
| product_name | p     | qty_sold |
+--------------+-------+----------+
| projector    | sc-71 |       50 |
+--------------+-------+----------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

CUI:



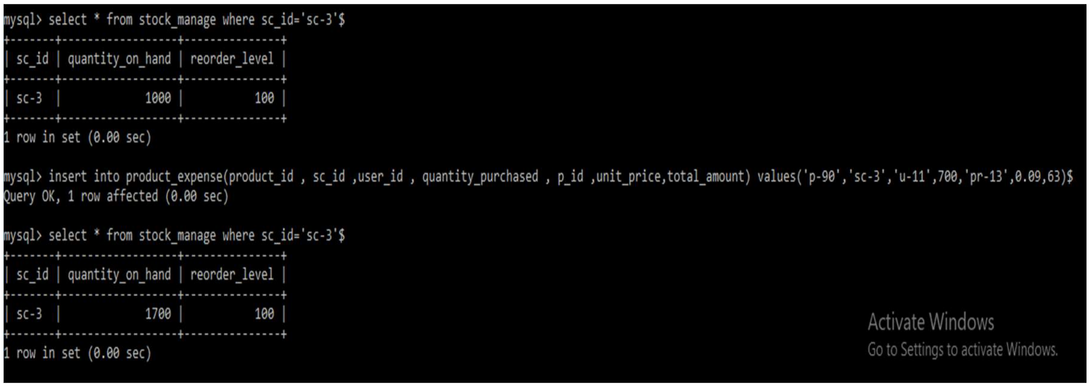
```
CallableStatement m8 = c.prepareCall("{ ?=call least_selling_item() }");
m8.registerOutParameter(1, Types.VARCHAR);
m8.execute();
String a8 =m8.getString(1);
System.out.println(a8);
m8.close();
break;
```

Trigger 1 : Update Stock Manage on Product_Expense



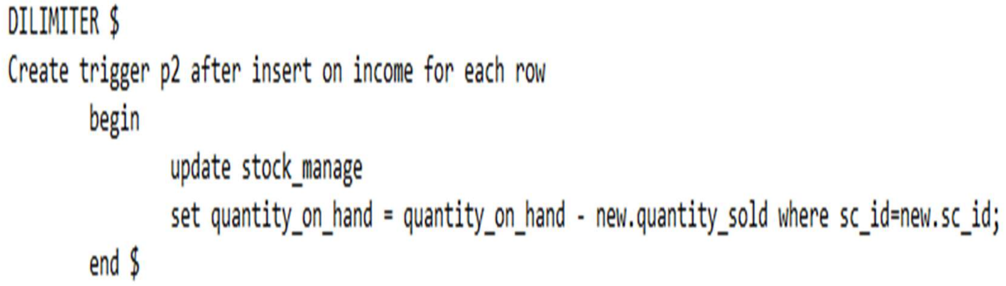
```
DILIMITER $
Create trigger p1 after insert on product_expense for each row
        begin
                update stock_manage
                set quantity_on_hand=quantity_on_hand + new.quantity_purchased where sc_id=new.sc_id;
        end $
```

Trigger 1 Output : Quantity on Hand is Updated Automatically on Product Expense



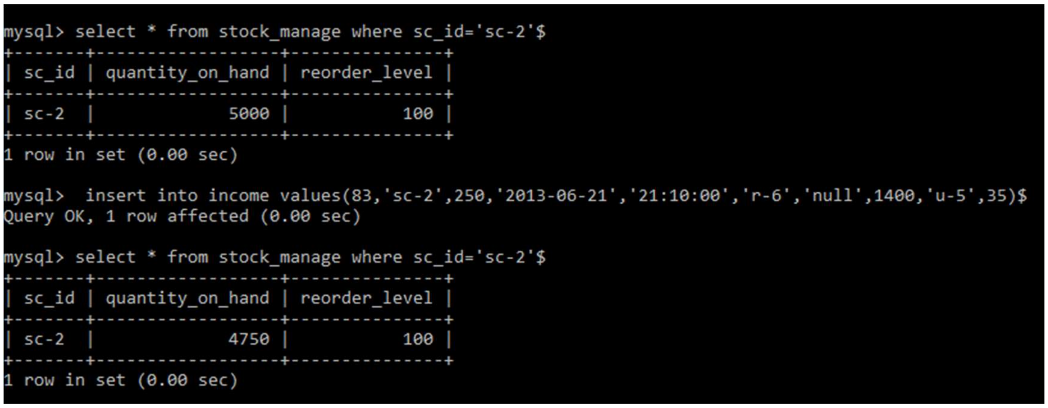
```
mysql> select * from stock_manage where sc_id='sc-3'$
+-------+------------------+---------------+
| sc_id | quantity_on_hand | reorder_level |
+-------+------------------+---------------+
| sc-3  |             1000 |           100 |
+-------+------------------+---------------+
1 row in set (0.00 sec)

mysql> insert into product_expense(product_id , sc_id ,user_id , quantity_purchased , p_id ,unit_price,total_amount) values('p-90','sc-3','u-11',700,'pr-13',0.09,63)$
Query OK, 1 row affected (0.00 sec)

mysql> select * from stock_manage where sc_id='sc-3'$
+-------+------------------+---------------+
| sc_id | quantity_on_hand | reorder_level |
+-------+------------------+---------------+
| sc-3  |             1700 |           100 |
+-------+------------------+---------------+
1 row in set (0.00 sec)
```

Trigger 2 : Update Stock Manage on Income



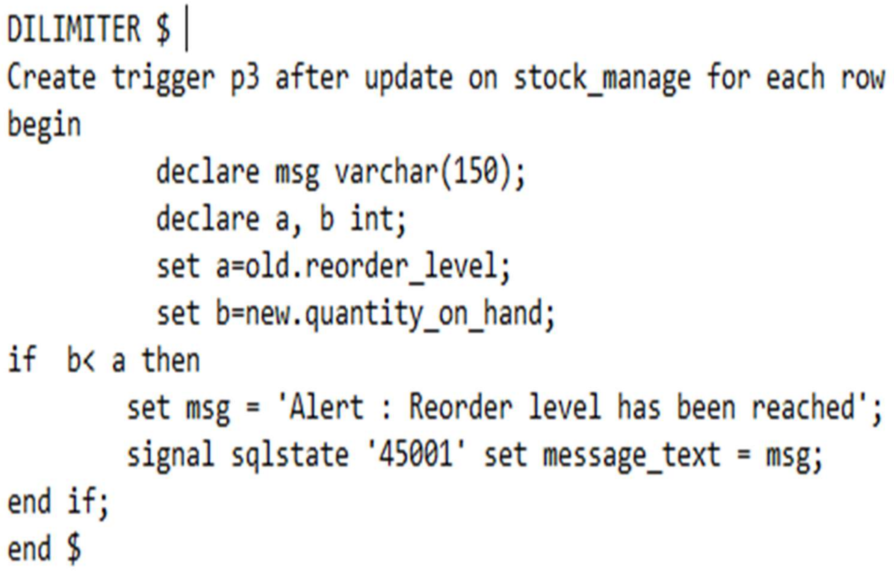
```
DILIMITER $
Create trigger p2 after insert on income for each row
        begin
                update stock_manage
                set quantity_on_hand = quantity_on_hand - new.quantity_sold where sc_id=new.sc_id;
        end $
```

Trigger 2 Output : Quantity on Hand is Updated Automatically on Income



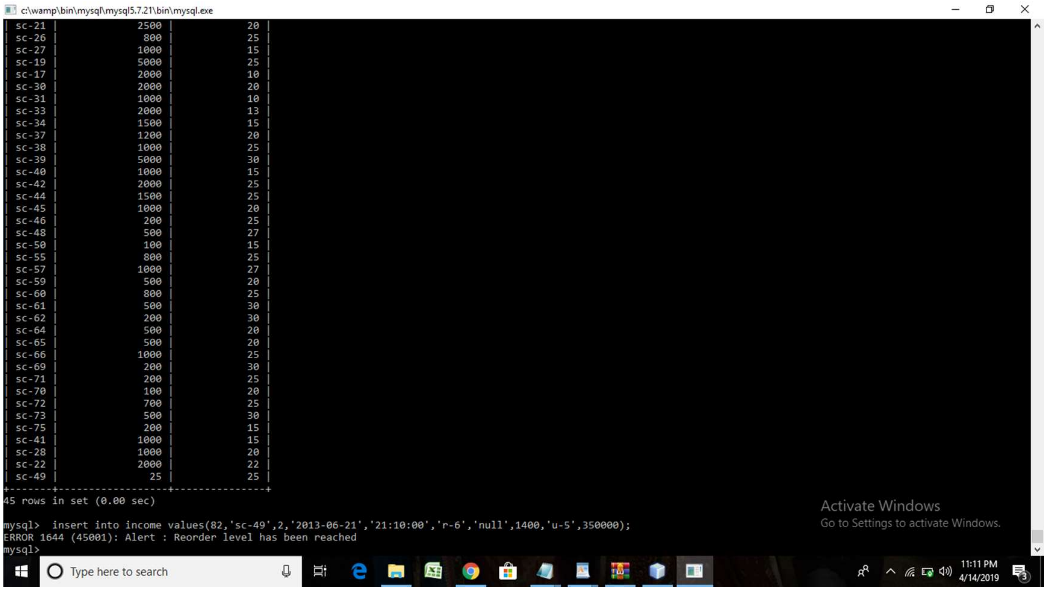
```
mysql> select * from stock_manage where sc_id='sc-2'$
+-------+------------------+---------------+
| sc_id | quantity_on_hand | reorder_level |
+-------+------------------+---------------+
| sc-2  |             5000 |           100 |
+-------+------------------+---------------+
1 row in set (0.00 sec)

mysql>  insert into income values(83,'sc-2',250,'2013-06-21','21:10:00','r-6','null',1400,'u-5',35)$
Query OK, 1 row affected (0.00 sec)

mysql> select * from stock_manage where sc_id='sc-2'$
+-------+------------------+---------------+
| sc_id | quantity_on_hand | reorder_level |
+-------+------------------+---------------+
| sc-2  |             4750 |           100 |
+-------+------------------+---------------+
1 row in set (0.00 sec)
```

Trigger 3 : After update on Stock Manage



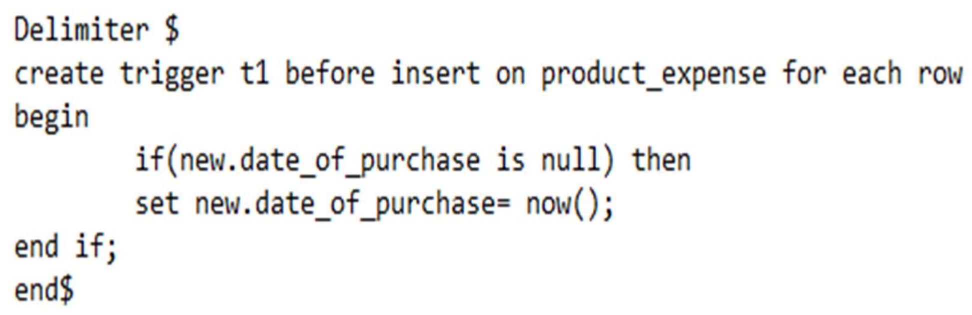
```
DILIMITER $
Create trigger p3 after update on stock_manage for each row
begin
          declare msg varchar(150);
          declare a, b int;
          set a=old.reorder_level;
          set b=new.quantity_on_hand;
if  b< a then
        set msg = 'Alert : Reorder level has been reached';
        signal sqlstate '45001' set message_text = msg;
end if;
end $
```

Trigger 3 Error Screenshot :


c:\wamp\bin\mysql\mysql5.7.21\bin\mysql.exe
Activate Windows
Go to Settings to activate Windows.
Type here to search
11:11 PM
4/14/2019

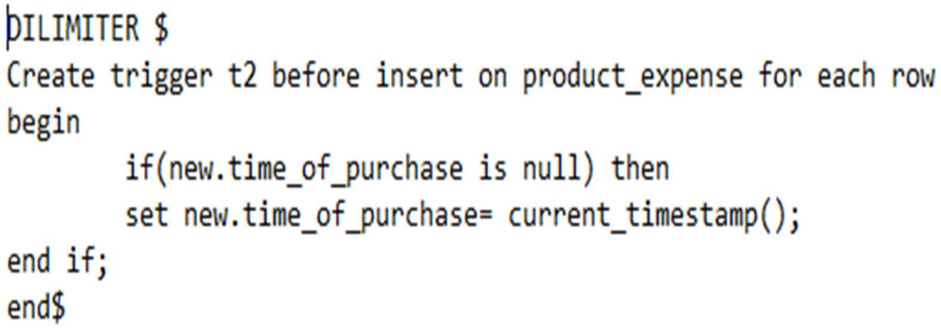
```
| sc-21 |            2500 |           20 |
| sc-26 |             800 |           25 |
| sc-27 |            1000 |           15 |
| sc-19 |            5000 |           25 |
| sc-17 |            2000 |           10 |
| sc-30 |            2000 |           20 |
| sc-31 |            1000 |           10 |
| sc-33 |            2000 |           13 |
| sc-34 |            1500 |           15 |
| sc-37 |            1200 |           20 |
| sc-38 |            1000 |           25 |
| sc-39 |            5000 |           30 |
| sc-40 |            1000 |           15 |
| sc-42 |            2000 |           25 |
| sc-44 |            1500 |           25 |
| sc-45 |            1000 |           20 |
| sc-46 |             200 |           25 |
| sc-48 |             500 |           27 |
| sc-50 |             100 |           15 |
| sc-55 |             800 |           25 |
| sc-57 |            1000 |           27 |
| sc-59 |             500 |           20 |
| sc-60 |             800 |           25 |
| sc-61 |             500 |           30 |
| sc-62 |             200 |           30 |
| sc-64 |             500 |           20 |
| sc-65 |             500 |           20 |
| sc-66 |            1000 |           25 |
| sc-69 |             200 |           30 |
| sc-71 |             200 |           25 |
| sc-70 |             100 |           20 |
| sc-72 |             700 |           25 |
| sc-73 |             500 |           30 |
| sc-75 |             200 |           15 |
| sc-41 |            1000 |           15 |
| sc-28 |            1000 |           20 |
| sc-22 |            2000 |           22 |
| sc-49 |              25 |           25 |
+-------+-----------------+--------------+
45 rows in set (0.00 sec)

mysql>  insert into income values(82,'sc-49',2,'2013-06-21','21:10:00','r-6','null',1400,'u-5',350000);
ERROR 1644 (45001): Alert : Reorder level has been reached
mysql>
```

Trigger 4 :    To add current date on Product Expense



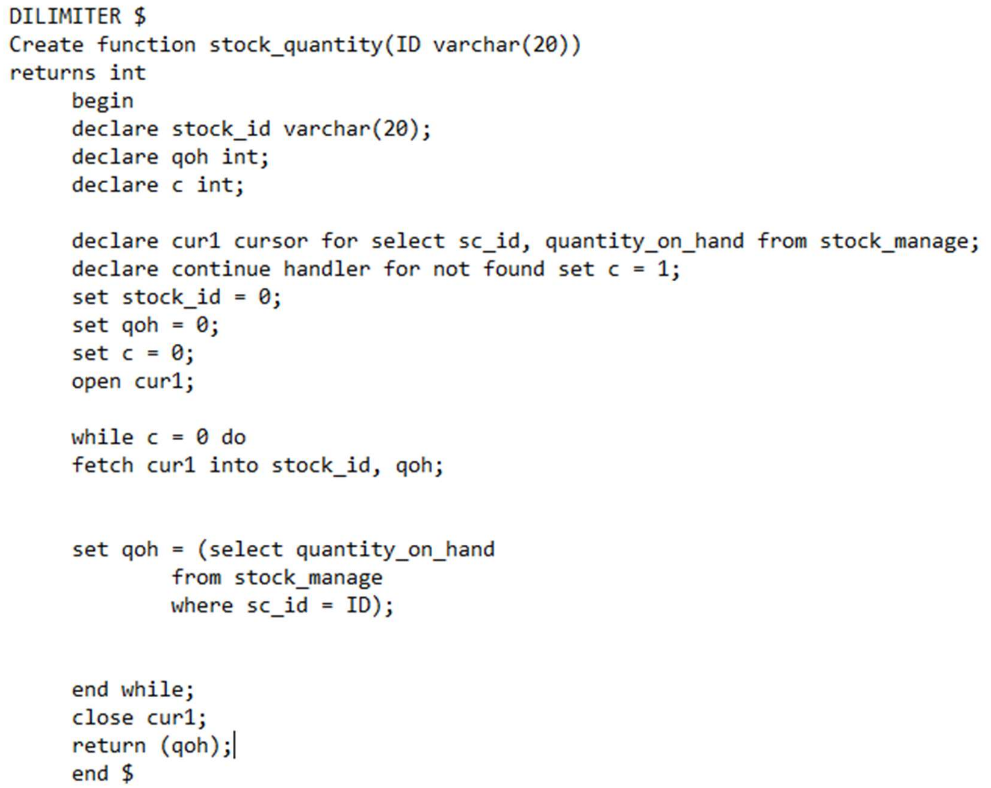
```
Delimiter $
create trigger t1 before insert on product_expense for each row
begin
        if(new.date_of_purchase is null) then
        set new.date_of_purchase= now();
end if;
end$
```

Trigger 5 : To add current time on Product Expense



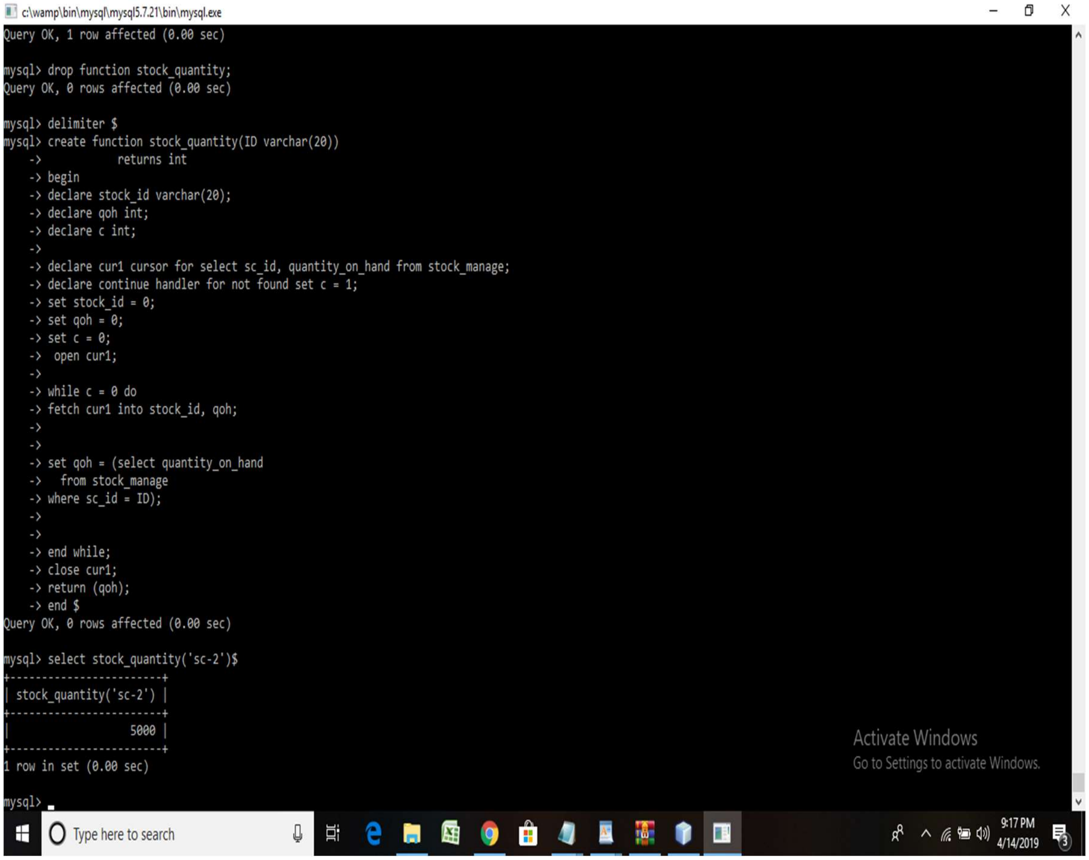
```
DILIMITER $
Create trigger t2 before insert on product_expense for each row
begin
        if(new.time_of_purchase is null) then
        set new.time_of_purchase= current_timestamp();
end if;
end$
```

Function 1 : Function to display the quantity available



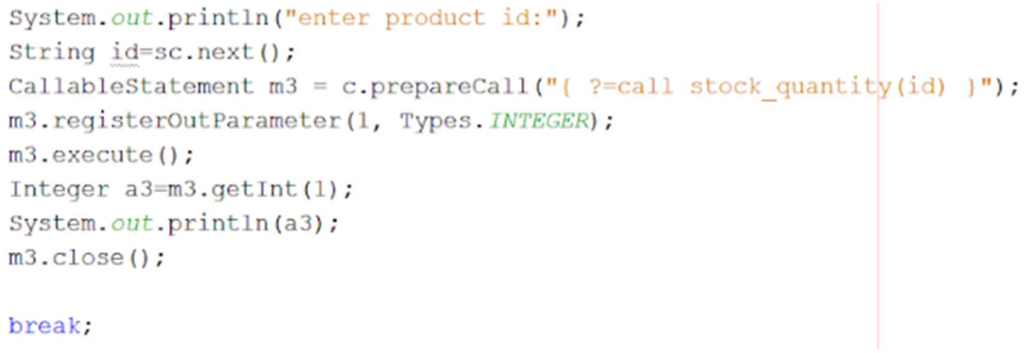
```
DILIMITER $
Create function stock_quantity(ID varchar(20))
returns int
     begin
     declare stock_id varchar(20);
     declare qoh int;
     declare c int;

     declare cur1 cursor for select sc_id, quantity_on_hand from stock_manage;
     declare continue handler for not found set c = 1;
     set stock_id = 0;
     set qoh = 0;
     set c = 0;
     open cur1;

     while c = 0 do
     fetch cur1 into stock_id, qoh;


     set qoh = (select quantity_on_hand
             from stock_manage
             where sc_id = ID);


     end while;
     close cur1;
     return (qoh);
     end $
```

Function 1 : OUTPUT


c:\wamp\bin\mysql\mysql5.7.21\bin\mysql.exe
Activate Windows
Go to Settings to activate Windows.
Type here to search
9:17 PM
4/14/2019

```
Query OK, 1 row affected (0.00 sec)

mysql> drop function stock_quantity;
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter $
mysql> create function stock_quantity(ID varchar(20))
    ->             returns int
    -> begin
    -> declare stock_id varchar(20);
    -> declare qoh int;
    -> declare c int;
    ->
    -> declare cur1 cursor for select sc_id, quantity_on_hand from stock_manage;
    -> declare continue handler for not found set c = 1;
    -> set stock_id = 0;
    -> set qoh = 0;
    -> set c = 0;
    ->  open cur1;
    ->
    -> while c = 0 do
    -> fetch cur1 into stock_id, qoh;
    ->
    ->
    -> set qoh = (select quantity_on_hand
    ->   from stock_manage
    -> where sc_id = ID);
    ->
    ->
    -> end while;
    -> close cur1;
    -> return (qoh);
    -> end $
Query OK, 0 rows affected (0.00 sec)

mysql> select stock_quantity('sc-2')$
+------------------------+
| stock_quantity('sc-2') |
+------------------------+
|                   5000 |
+------------------------+
1 row in set (0.00 sec)

mysql> _
```

CUI :



```
System.out.println("enter product id:");
String id=sc.next();
CallableStatement m3 = c.prepareCall("{ ?=call stock_quantity(id) }");
m3.registerOutParameter(1, Types.INTEGER);
m3.execute();
Integer a3=m3.getInt(1);
System.out.println(a3);
m3.close();

break;
```