CS F212

Database Systems - Assignment

Semester II 2017-18

BITS Pilani
Dubai Campus

Topic : Restaurant Management System

Group members :

Parth Poply	**2016A7PS0249U**
Sujith Sizon	**2016A7PS0277U**
Somil Mathur	**2016A7PS0271U**
Jaskanwar Singh	**2016A7PS0075U**
Tushar Agarwal	**2016A7PS0080U**

Objective of the project :

The main objective of the project is to develop a database on a restaurant management system. By using this project, students get a clear idea of how the database works and how they can use it. Also by working on this project students are able to understand the use and retrieval of databases in general.

Purpose/Relevance of the project :

A Database Management System is a system software for creating and managing databases. The DBMS provides the users and programmers with a systematic way to create, retrieve update and manage data.
The project on Restaurant Management system helps to understand how databases work in restaurants. Every restaurant has many entities and each entity has many attributes. By identifying the entities, attributes, and relations in the restaurant management system it would ease the task of working on the database.

Need for Restaurant Management System ?

People are different as are restaurants. By producing meal experiences with unique characteristics, restaurants cater for the needs of specific customer categories.
A look at the restaurant market of a city gives

a fascinating illustration of the culture and diversity of that city.
From an economic point of view, this diverse supply of meal experiences is perfectly rational when the demand is equally diverse and customers are prepared to pay a premium price for the extra; be it an extra service experience, an extra culinary experience or an extra aesthetic experience of an elegant dining room.
A problem in an economic context, as often in economics, is information. Restaurant managers must not only know the preferences and the needs of their customers, but also allocate resources in a way that makes it possible to produce the particular meal experiences that their customers expect.
Accounting systems provide an instrument for budgeting and for an efficient allocation of resources. A restaurant manager must be aware of the fact that customers normally do not come only to eat, but come for a multidimensional experience and restaurant managers should, therefore, plan and budget multidimensionally. Accounting systems provide information about goal achievement and signals for management control. A multidimensional budget will force managers to follow up activities in detail and assess whether resources efficiently perform or support the specific activities they are allocated to. Traditional management control is concerned with cost of resources, e.g. food cost and cost of labour, whether management control related to customer experiences should be concerned both with the cost of resources and the value of the meal experience created for the customer.
Accounting systems also provide necessary information for cost-based pricing. The price charges the customer for the cost of producing a meal experience. Traditionally, restaurant managers determine a cost-based mark-up price by adding 150–300 per cent overhead charges on the food cost. Apart from the fact that this practice ignores the value of the customer experience, it can, from a theoretical point of view, be criticised for a number of reasons.

Identification of Entities :

1. Restaurant
2. Cashier
3. Bill
4. Manager
5. Customer
6. Chef
7. Waiter
8. Item
9. Order

Identification of Attributes :

1. ***Restaurant*** : Name, Ph. No. , Address
2. ***Cashier*** : Bill No. , Name , Txn ID
3. ***Bill*** : Order Detail, Bill No. , Amount , Customer ID
4. ***Manager*** : Manager ID , Name , Ph. No.
5. ***Customer*** : Name , Ph. No. , Address , Customer ID
6. ***Chef*** : Name , Chef ID
7. ***Waiter*** : Name , ID
8. ***Item*** : Item No. , Amount , Quantity , Description , Order No.
9. ***Order*** : No. Of Items , Order No.

Identification of Relations :

1. Restaurant HAS A Manager
2. Restaurant HAS A Cashier
3. Bill PAID TO Cashier
4. Manager TELLS Chef
5. Manager TAKES Order
6. Customer PLACES Order
7. Customer PAYS Bill
8. Waiter BRINGS Order
9. Waiter DELIVERS TO Customer
10. Chef PREPARES Order
11. Order CONTAINS Item

ER diagram :

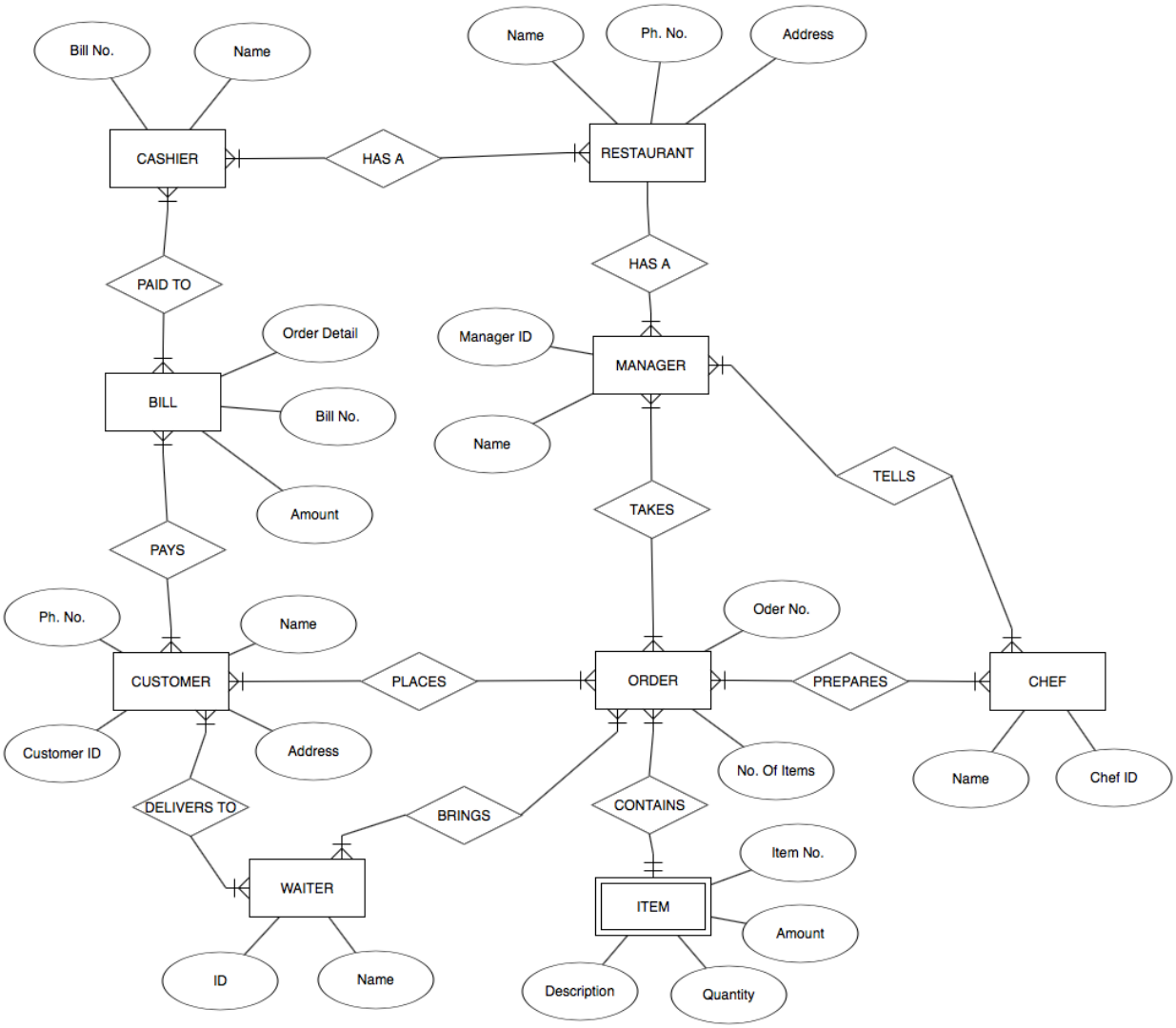


Fig : ER diagram for Restaurant Management System (made using ERDPlus.com)

Relational model :

(Pre - Normalisation)

Tables :

1. Restaurant (Name, Ph. No. , Address)
2. Cashier (Bill No. , Name , Txn ID)
3. Bill (Order Detail, Bill No. , Amount , Customer ID)
4. Manager (Manager ID , Name , Ph. No.)
5. Customer (Name , Ph. No. , Address , Customer ID)
6. Chef (Name , Chef ID)
7. Waiter (Name , ID)
8. Order (Item No. , Amount , Quantity , Description , No. Of Items , Order No. , Customer ID)

Normalisation :

1)

Table Restaurant (Name, Ph. No. , Address)

Functional dependencies :

Ph. No. —> Name , Address

Normal form : Already in 3NF (no need for decomposition)

2)

Table Cashier (Bill No. , Name , Txn ID)

Functional dependencies :

Txn ID —> Name , Bill No.

Normal form : Already in 3NF (no need for decomposition)

3)

Table Bill (Order Detail, Bill No. , Amount , Customer ID)

Functional dependencies :

Bill No. —> Order detail , Amount , Customer ID

Normal form : Already in 3NF (no need for decomposition)

4)

Table Manager (Manager ID , Name , Ph. No.)

Functional dependencies :

Manager ID —> Name, Ph. No.

Normal form : Already in 3NF (no need for decomposition)

5)

Table Customer (Name , Ph. No. , Address , Customer ID)

Functional dependencies :

Customer ID —> Name , Address
Ph. No. —> Name , Address

Normal form : Already in 3NF (no need for decomposition)

6)

Table Chef (Name , Chef ID)

Functional dependencies :

Chef ID —> Name

Normal form : Already in 3NF (no need for decomposition)

7)

Table Waiter (Name , ID)

Functional dependencies :

ID —> Name

Normal form : Already in 3NF (no need for decomposition)

8)

Table Order (Item No. , Amount , Quantity , Description , No. Of Items , Order No. , Customer ID)

Functional dependencies :

Order No. —> No. Of Items , Description , **Item No**. , Customer ID

Item No. —> Amount , Quantity , Description

There exists **transitive dependency** (through Item No.)

So decomposition is required .

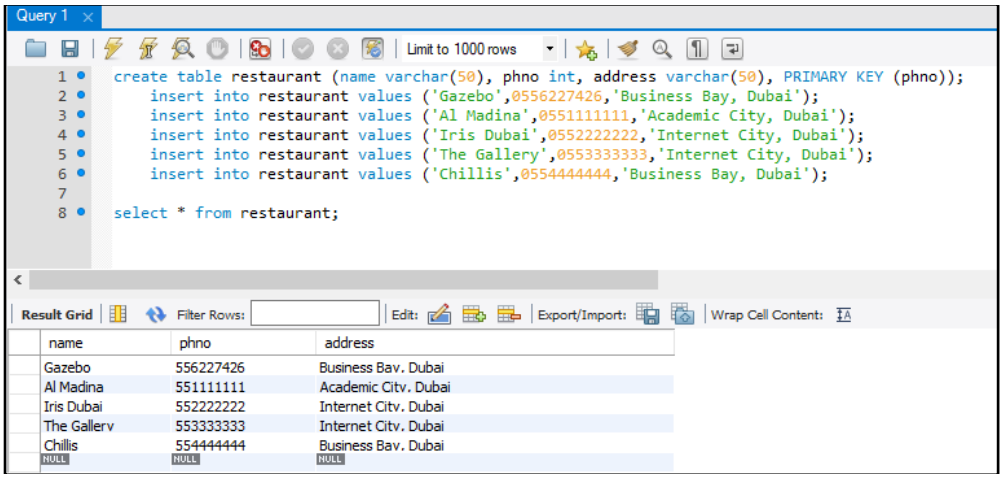
Table Order can be Decomposed to :

Table Item (Item No. , Amount , Quantity , Description)
And
Table Order(Order No. , Item No. , No. Of Items , Customer ID)

New Relational Schema (after Normalisation and with primary keys) :

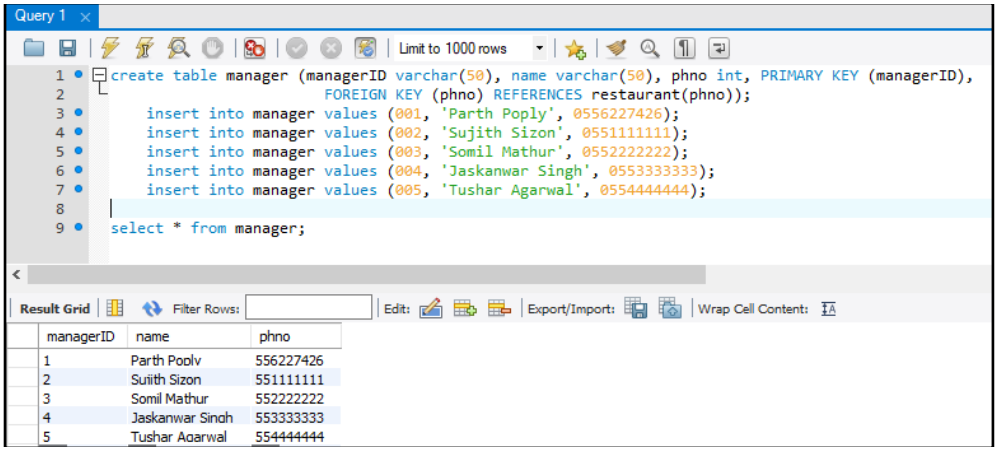
1. *Restaurant* (Name, Ph. No. , Address)
2. *Cashier* (Bill No. , Name , Txn ID)
3. *Bill* (Order Detail, Bill No. , Amount , Customer ID)
4. *Manager* (Manager ID , Name , Ph. No.)
5. *Customer* (Name , Ph. No. , Address , Customer ID)
6. *Chef* (Name , Chef ID)
7. *Waiter* (Name , ID)
8. *Order* (No. Of Items , Order No. , Item No. , Customer ID)
9. *Item* (Item No. , Order No. , Amount , Quantity , Description)

Table Creation/Insertions :



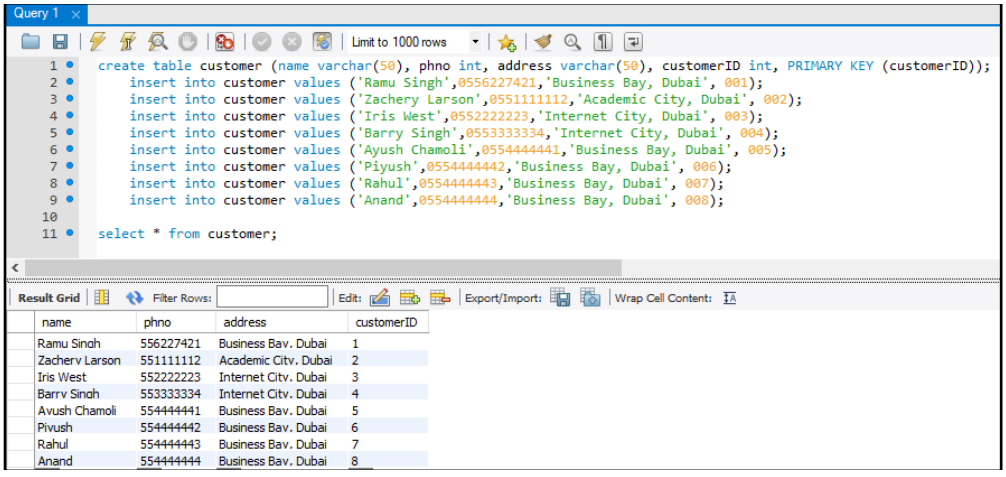
```
create table restaurant (name varchar(50), phno int, address varchar(50), PRIMARY KEY (phno));
    insert into restaurant values ('Gazebo',0556227426,'Business Bay, Dubai');
    insert into restaurant values ('Al Madina',0551111111,'Academic City, Dubai');
    insert into restaurant values ('Iris Dubai',0552222222,'Internet City, Dubai');
    insert into restaurant values ('The Gallery',0553333333,'Internet City, Dubai');
    insert into restaurant values ('Chillis',0554444444,'Business Bay, Dubai');

select * from restaurant;
```

name	phno	address
Gazebo	556227426	Business Bay, Dubai
Al Madina	551111111	Academic City, Dubai
Iris Dubai	552222222	Internet City, Dubai
The Gallery	553333333	Internet City, Dubai
Chillis	554444444	Business Bay, Dubai
NULL	NULL	NULL



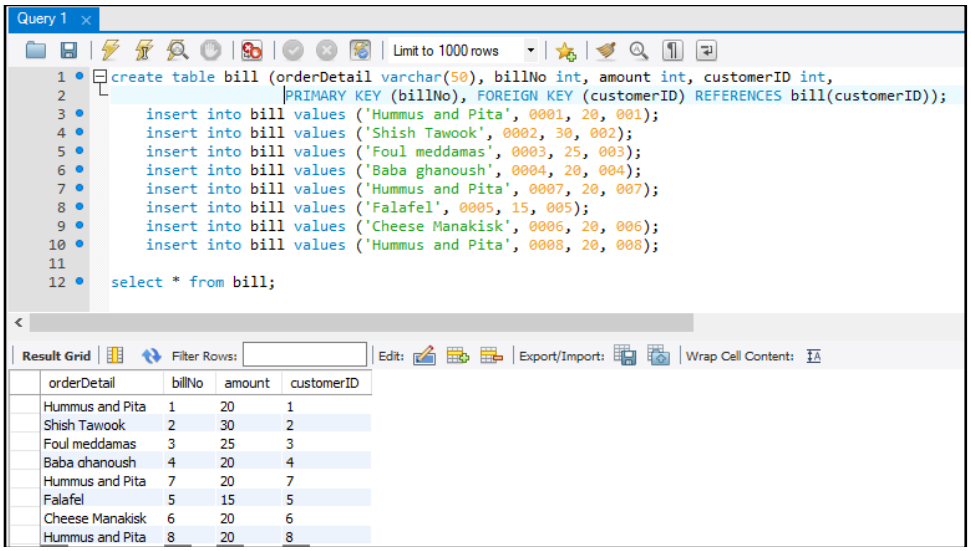
```
create table manager (managerID varchar(50), name varchar(50), phno int, PRIMARY KEY (managerID),
                      FOREIGN KEY (phno) REFERENCES restaurant(phno));
    insert into manager values (001, 'Parth Poply', 0556227426);
    insert into manager values (002, 'Sujith Sizon', 0551111111);
    insert into manager values (003, 'Somil Mathur', 0552222222);
    insert into manager values (004, 'Jaskanwar Singh', 0553333333);
    insert into manager values (005, 'Tushar Agarwal', 0554444444);

select * from manager;
```

managerID	name	phno
1	Parth Poply	556227426
2	Sujith Sizon	551111111
3	Somil Mathur	552222222
4	Jaskanwar Singh	553333333
5	Tushar Agarwal	554444444



```
create table customer (name varchar(50), phno int, address varchar(50), customerID int, PRIMARY KEY (customerID));
    insert into customer values ('Ramu Singh',0556227421,'Business Bay, Dubai', 001);
    insert into customer values ('Zachery Larson',0551111112,'Academic City, Dubai', 002);
    insert into customer values ('Iris West',0552222223,'Internet City, Dubai', 003);
    insert into customer values ('Barry Singh',0553333334,'Internet City, Dubai', 004);
    insert into customer values ('Ayush Chamoli',0554444441,'Business Bay, Dubai', 005);
    insert into customer values ('Piyush',0554444442,'Business Bay, Dubai', 006);
    insert into customer values ('Rahul',0554444443,'Business Bay, Dubai', 007);
    insert into customer values ('Anand',0554444444,'Business Bay, Dubai', 008);

select * from customer;
```

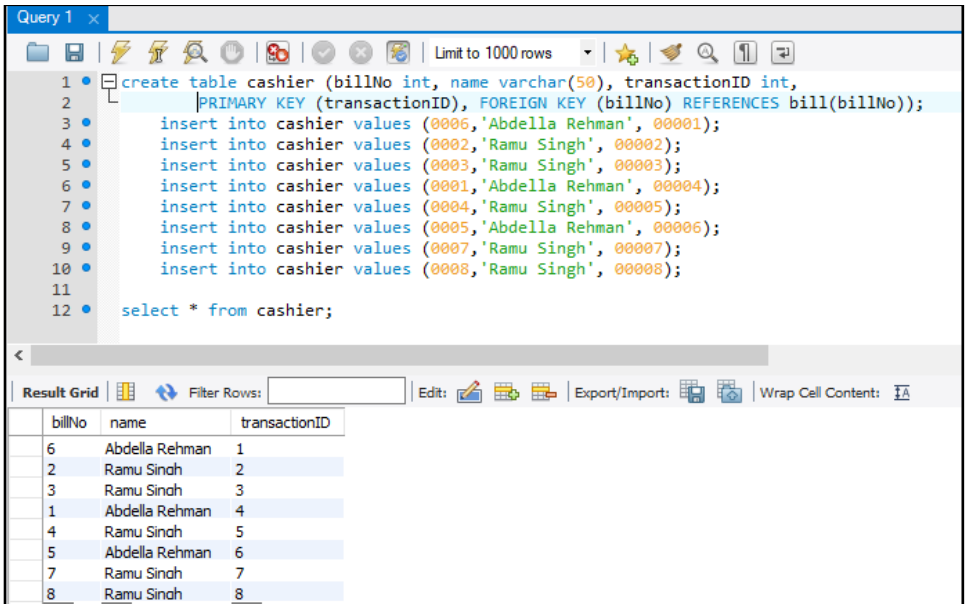
name	phno	address	customerID
Ramu Singh	556227421	Business Bay, Dubai	1
Zachery Larson	551111112	Academic City, Dubai	2
Iris West	552222223	Internet City, Dubai	3
Barry Singh	553333334	Internet City, Dubai	4
Ayush Chamoli	554444441	Business Bay, Dubai	5
Piyush	554444442	Business Bay, Dubai	6
Rahul	554444443	Business Bay, Dubai	7
Anand	554444444	Business Bay, Dubai	8



```
create table bill (orderDetail varchar(50), billNo int, amount int, customerID int,
                   PRIMARY KEY (billNo), FOREIGN KEY (customerID) REFERENCES bill(customerID));
    insert into bill values ('Hummus and Pita', 0001, 20, 001);
    insert into bill values ('Shish Tawook', 0002, 30, 002);
    insert into bill values ('Foul meddamas', 0003, 25, 003);
    insert into bill values ('Baba ghanoush', 0004, 20, 004);
    insert into bill values ('Hummus and Pita', 0007, 20, 007);
    insert into bill values ('Falafel', 0005, 15, 005);
    insert into bill values ('Cheese Manakisk', 0006, 20, 006);
    insert into bill values ('Hummus and Pita', 0008, 20, 008);

select * from bill;
```

Result Grid

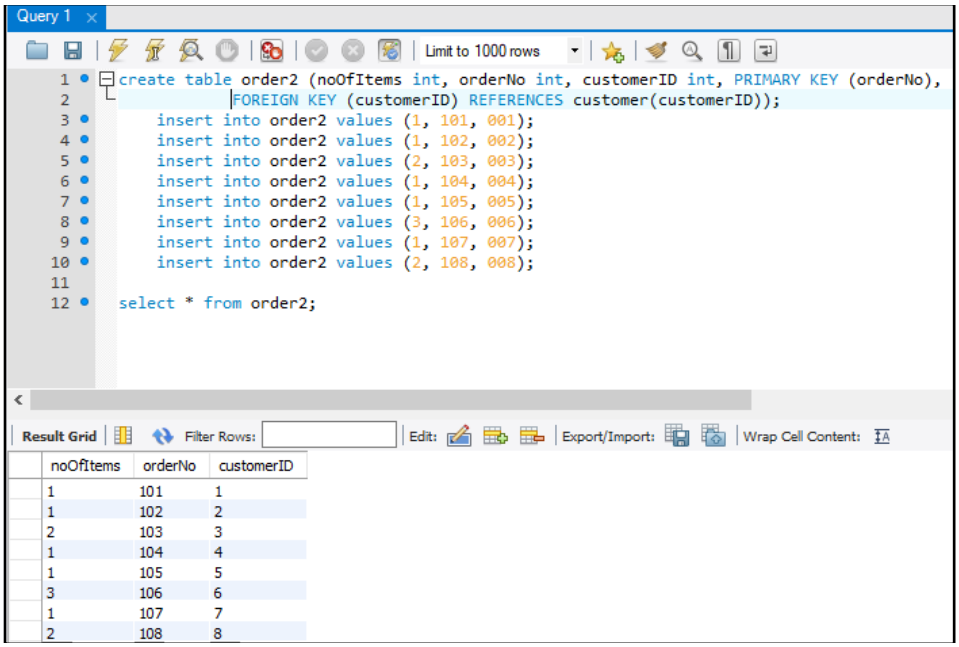
orderDetail	billNo	amount	customerID
Hummus and Pita	1	20	1
Shish Tawook	2	30	2
Foul meddamas	3	25	3
Baba ghanoush	4	20	4
Hummus and Pita	7	20	7
Falafel	5	15	5
Cheese Manakisk	6	20	6
Hummus and Pita	8	20	8



```
create table cashier (billNo int, name varchar(50), transactionID int,
        PRIMARY KEY (transactionID), FOREIGN KEY (billNo) REFERENCES bill(billNo));
    insert into cashier values (0006,'Abdella Rehman', 00001);
    insert into cashier values (0002,'Ramu Singh', 00002);
    insert into cashier values (0003,'Ramu Singh', 00003);
    insert into cashier values (0001,'Abdella Rehman', 00004);
    insert into cashier values (0004,'Ramu Singh', 00005);
    insert into cashier values (0005,'Abdella Rehman', 00006);
    insert into cashier values (0007,'Ramu Singh', 00007);
    insert into cashier values (0008,'Ramu Singh', 00008);

select * from cashier;
```

Result Grid

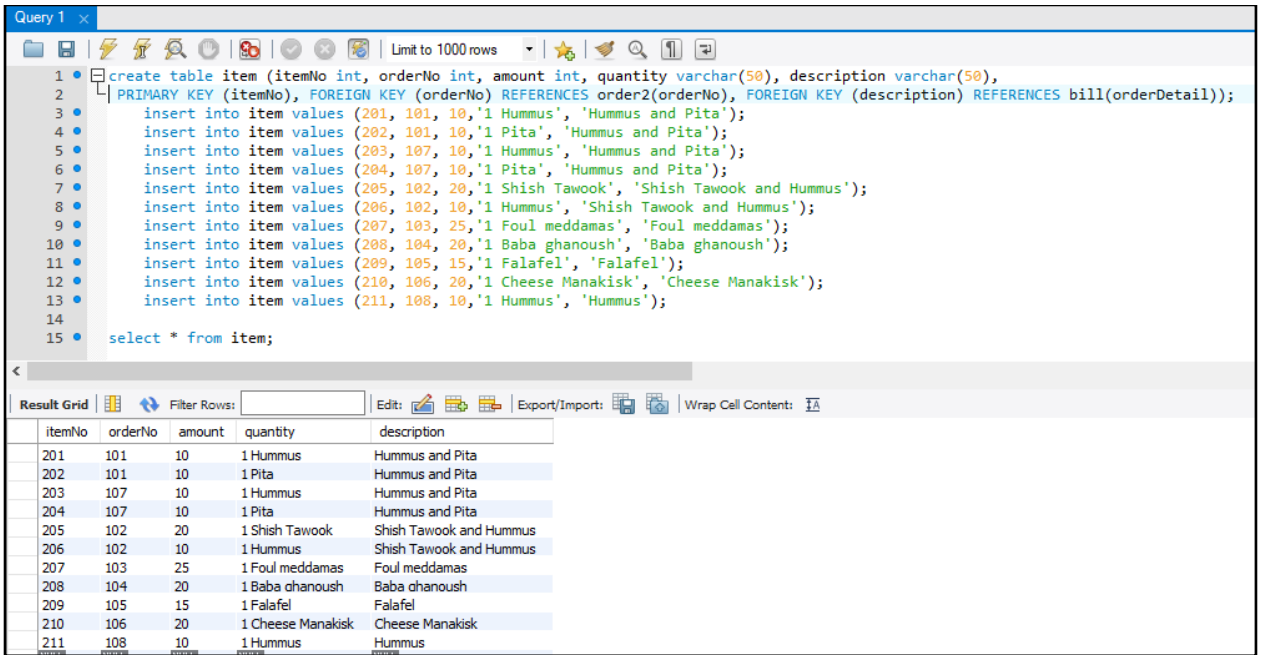
billNo	name	transactionID
6	Abdella Rehman	1
2	Ramu Singh	2
3	Ramu Singh	3
1	Abdella Rehman	4
4	Ramu Singh	5
5	Abdella Rehman	6
7	Ramu Singh	7
8	Ramu Singh	8



```
create table order2 (noOfItems int, orderNo int, customerID int, PRIMARY KEY (orderNo),
            FOREIGN KEY (customerID) REFERENCES customer(customerID));
    insert into order2 values (1, 101, 001);
    insert into order2 values (1, 102, 002);
    insert into order2 values (2, 103, 003);
    insert into order2 values (1, 104, 004);
    insert into order2 values (1, 105, 005);
    insert into order2 values (3, 106, 006);
    insert into order2 values (1, 107, 007);
    insert into order2 values (2, 108, 008);

select * from order2;
```

Result Grid

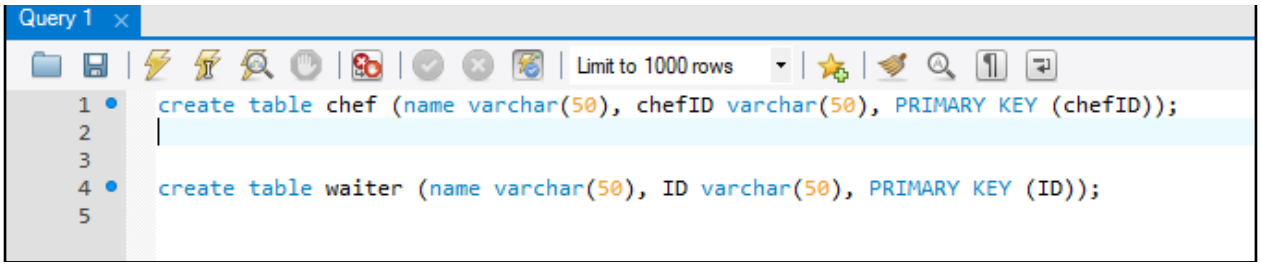
noOfItems	orderNo	customerID
1	101	1
1	102	2
2	103	3
1	104	4
1	105	5
3	106	6
1	107	7
2	108	8



```
create table item (itemNo int, orderNo int, amount int, quantity varchar(50), description varchar(50),
PRIMARY KEY (itemNo), FOREIGN KEY (orderNo) REFERENCES order2(orderNo), FOREIGN KEY (description) REFERENCES bill(orderDetail));
    insert into item values (201, 101, 10,'1 Hummus', 'Hummus and Pita');
    insert into item values (202, 101, 10,'1 Pita', 'Hummus and Pita');
    insert into item values (203, 107, 10,'1 Hummus', 'Hummus and Pita');
    insert into item values (204, 107, 10,'1 Pita', 'Hummus and Pita');
    insert into item values (205, 102, 20,'1 Shish Tawook', 'Shish Tawook and Hummus');
    insert into item values (206, 102, 10,'1 Hummus', 'Shish Tawook and Hummus');
    insert into item values (207, 103, 25,'1 Foul meddamas', 'Foul meddamas');
    insert into item values (208, 104, 20,'1 Baba ghanoush', 'Baba ghanoush');
    insert into item values (209, 105, 15,'1 Falafel', 'Falafel');
    insert into item values (210, 106, 20,'1 Cheese Manakisk', 'Cheese Manakisk');
    insert into item values (211, 108, 10,'1 Hummus', 'Hummus');

select * from item;
```

Result Grid

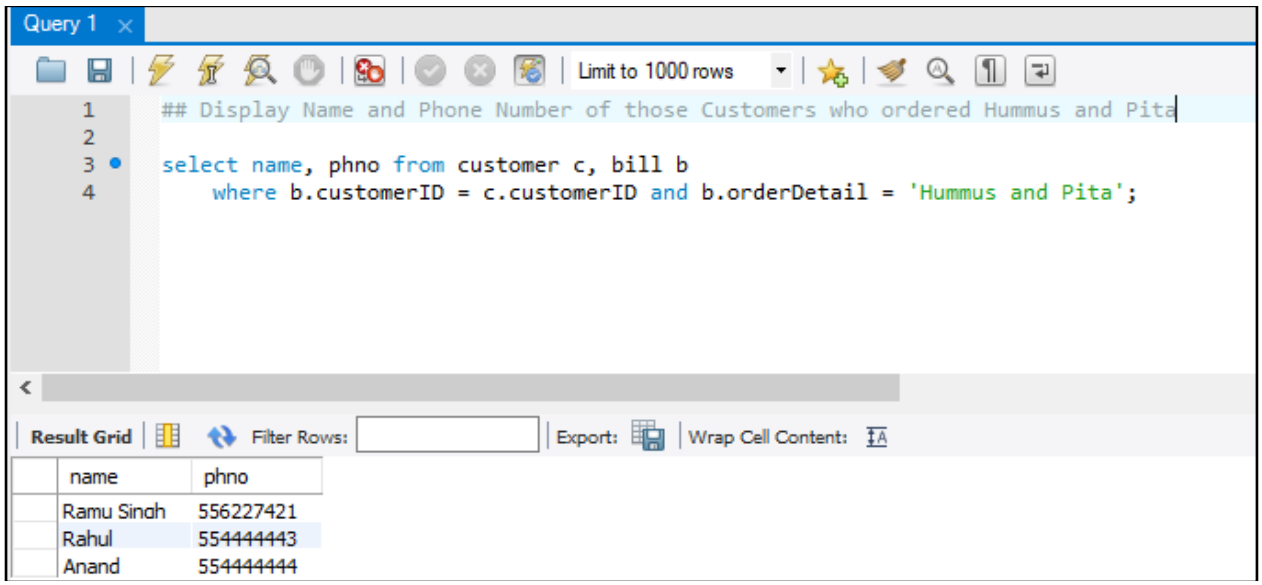
itemNo	orderNo	amount	quantity	description
201	101	10	1 Hummus	Hummus and Pita
202	101	10	1 Pita	Hummus and Pita
203	107	10	1 Hummus	Hummus and Pita
204	107	10	1 Pita	Hummus and Pita
205	102	20	1 Shish Tawook	Shish Tawook and Hummus
206	102	10	1 Hummus	Shish Tawook and Hummus
207	103	25	1 Foul meddamas	Foul meddamas
208	104	20	1 Baba ghanoush	Baba ghanoush
209	105	15	1 Falafel	Falafel
210	106	20	1 Cheese Manakisk	Cheese Manakisk
211	108	10	1 Hummus	Hummus



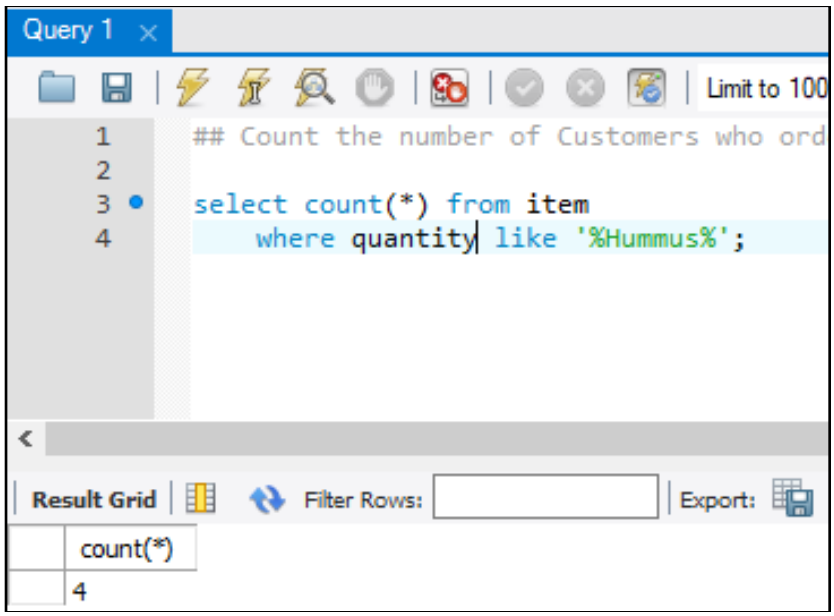
```
create table chef (name varchar(50), chefID varchar(50), PRIMARY KEY (chefID));


create table waiter (name varchar(50), ID varchar(50), PRIMARY KEY (ID));
```

Queries :

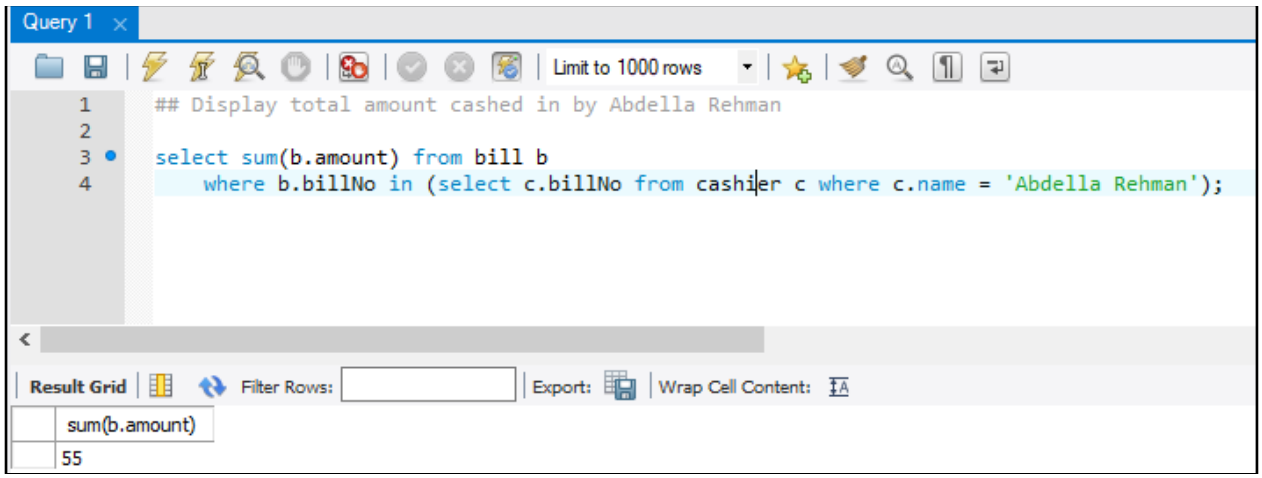
Display the Name and Phone Number of those Customers who ordered Hummus and Pita :



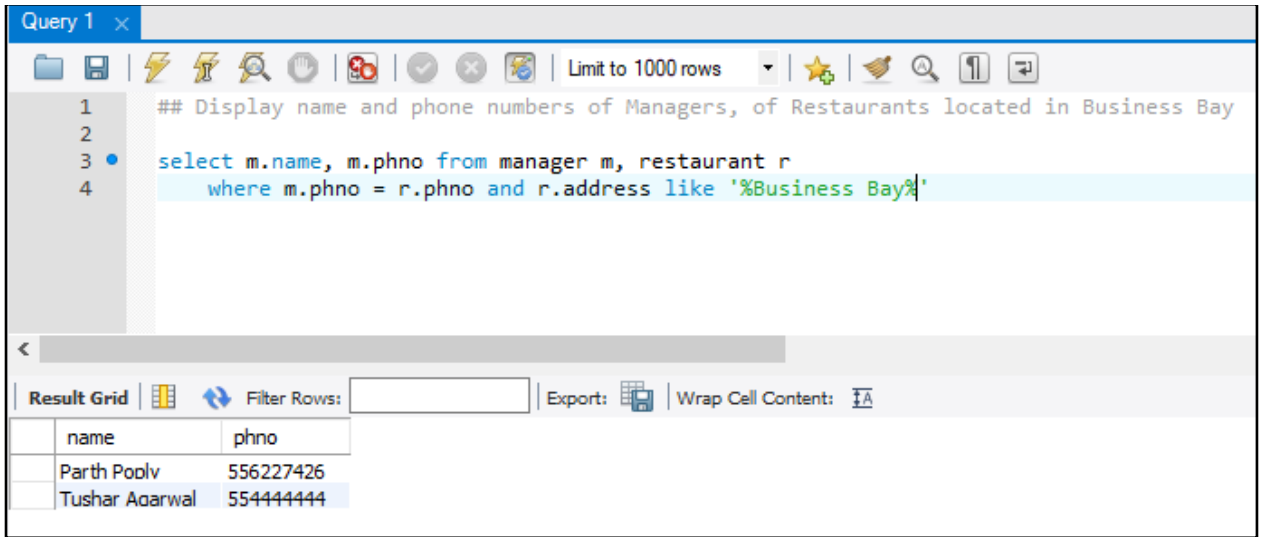
Count the number of Customers whose order(s) included Hummus :



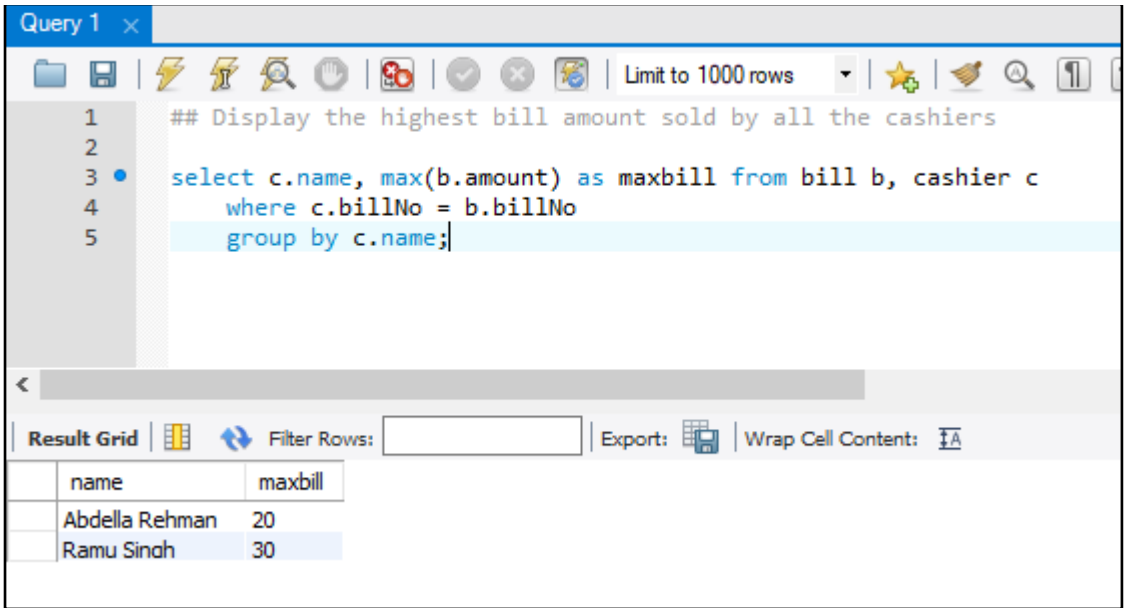
Display total amount cashed in by Cashier Abdella Rehman :



Display name and phone numbers of Managers of Restaurants located in Business Bay :

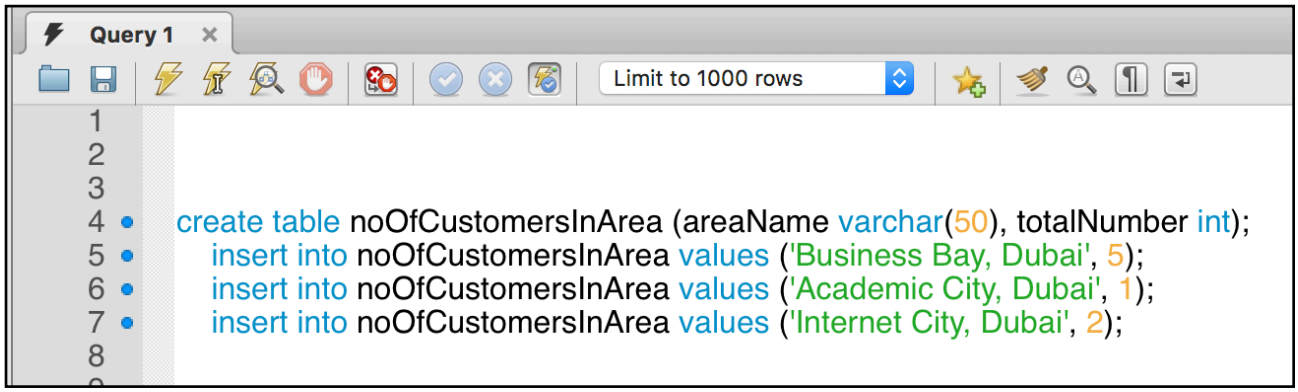


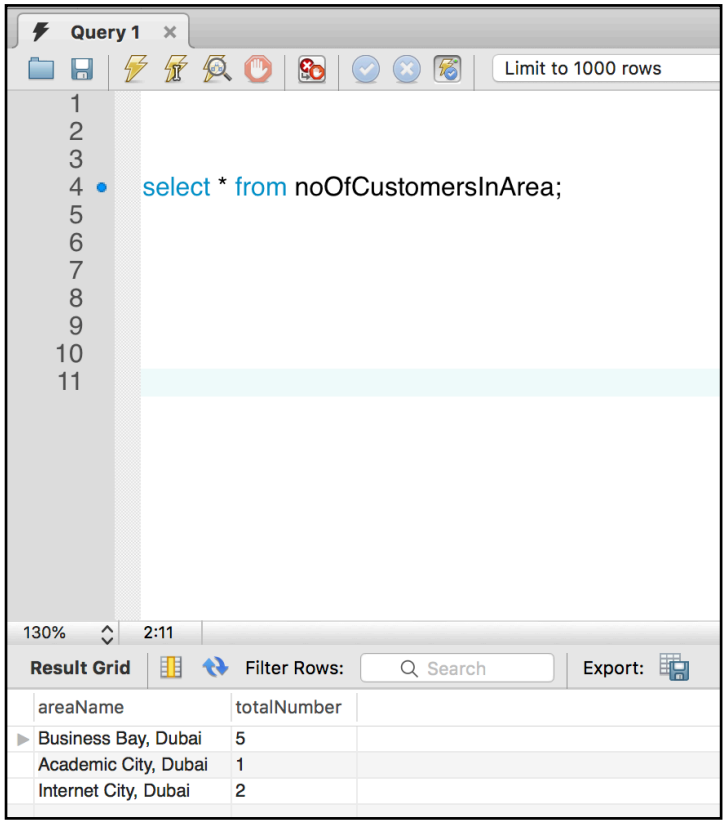
Display the max/highest bill amount sold by all the cashiers :



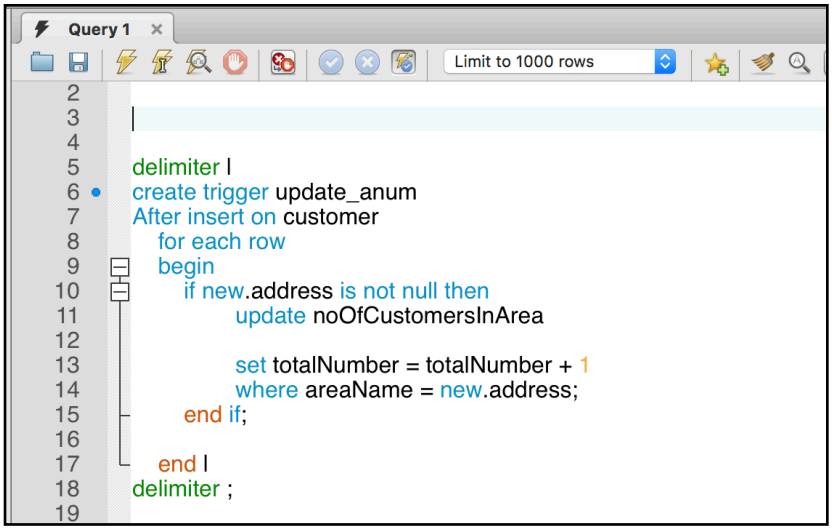
Triggers and Procedures :

Creation of table for keeping record of customers from specific areas :

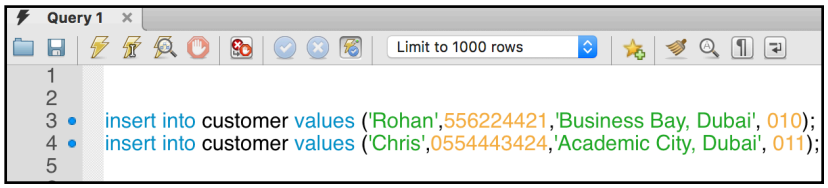


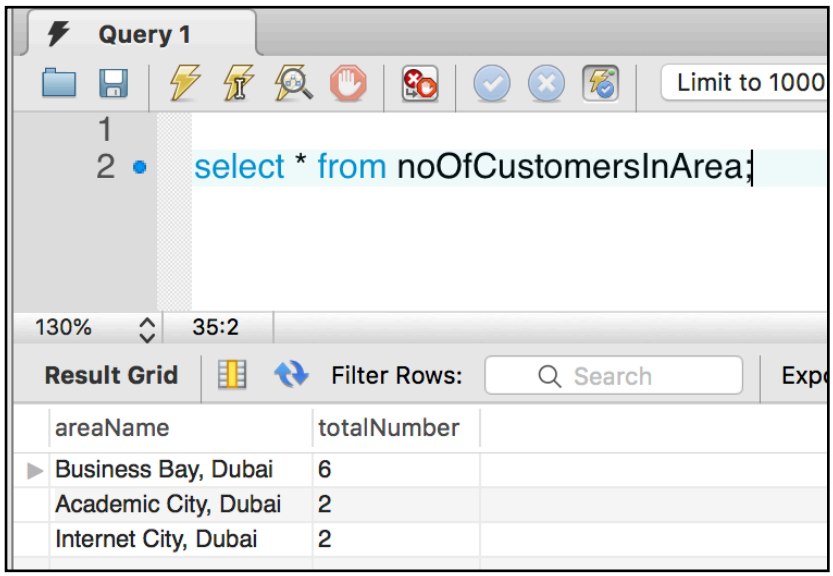


Trigger to update table noOfCustomerinArea after every insert of customer :

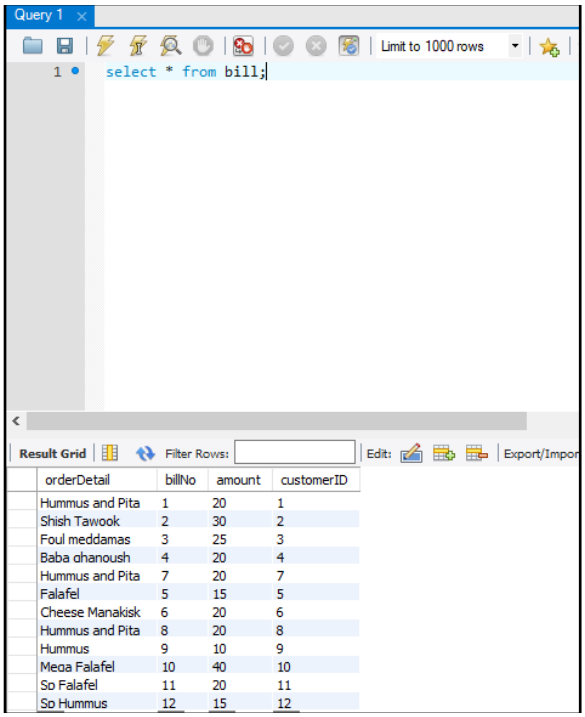


After Inserting new Customers :



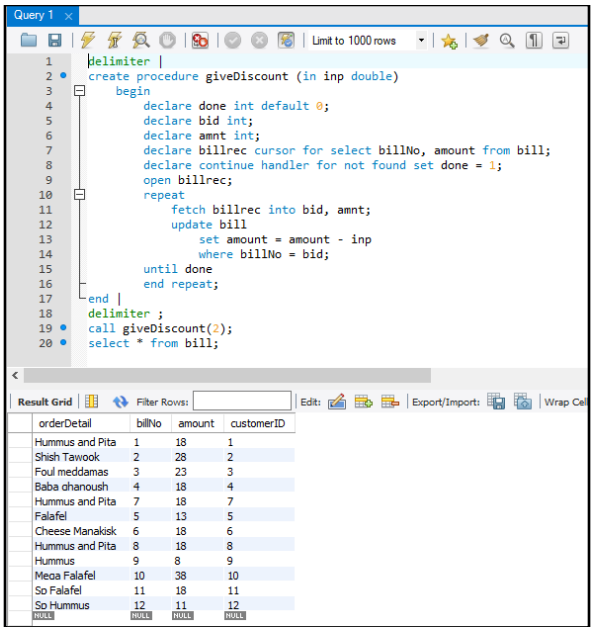


Procedure to give Discount on all Customer bills eg: 2dhs discount on all bills :



orderDetail	billNo	amount	customerID
Hummus and Pita	1	20	1
Shish Tawook	2	30	2
Foul meddamas	3	25	3
Baba ghanoush	4	20	4
Hummus and Pita	7	20	7
Falafel	5	15	5
Cheese Manakisk	6	20	6
Hummus and Pita	8	20	8
Hummus	9	10	9
Mega Falafel	10	40	10
So Falafel	11	20	11
So Hummus	12	15	12

After creation and calling Procedure :

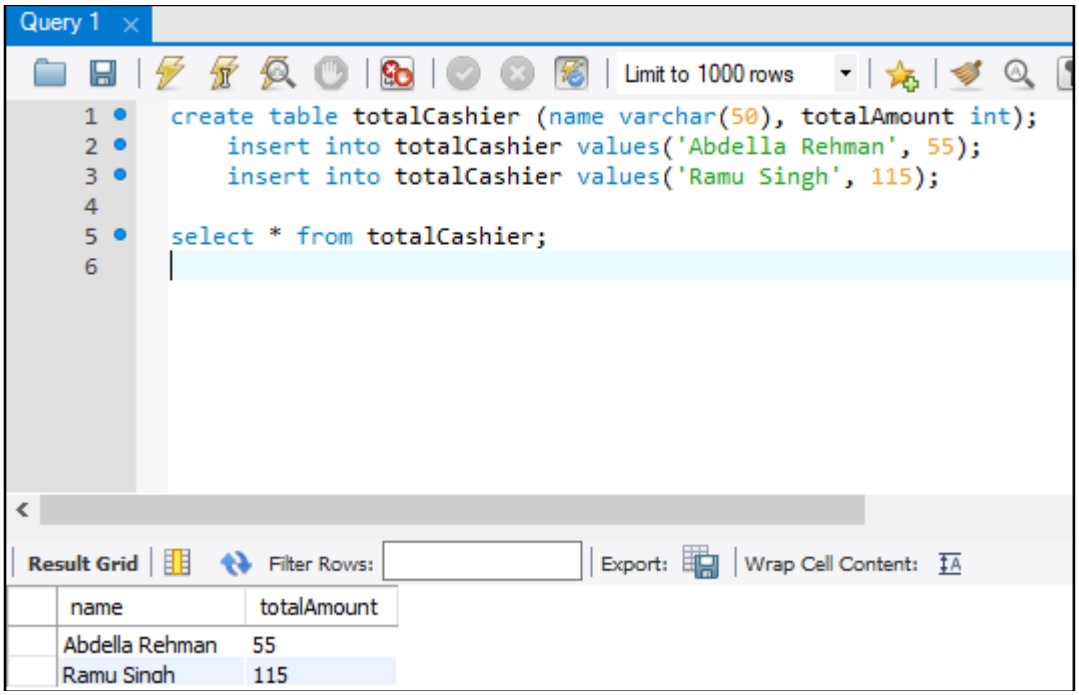


```
delimiter |
create procedure giveDiscount (in inp double)
    begin
        declare done int default 0;
        declare bid int;
        declare amnt int;
        declare billrec cursor for select billNo, amount from bill;
        declare continue handler for not found set done = 1;
        open billrec;
        repeat
            fetch billrec into bid, amnt;
            update bill
                set amount = amount - inp
                where billNo = bid;
        until done
        end repeat;
end |
delimiter ;
call giveDiscount(2);
select * from bill;
```

orderDetail	billNo	amount	customerID
Hummus and Pita	1	18	1
Shish Tawook	2	28	2
Foul meddamas	3	23	3
Baba ghanoush	4	18	4
Hummus and Pita	7	18	7
Falafel	5	13	5
Cheese Manakisk	6	18	6
Hummus and Pita	8	18	8
Hummus	9	8	9
Mega Falafel	10	38	10
So Falafel	11	18	11
So Hummus	12	11	12

Procedure to update the total cashed in amount by a particular cashier in the totalCashier table :

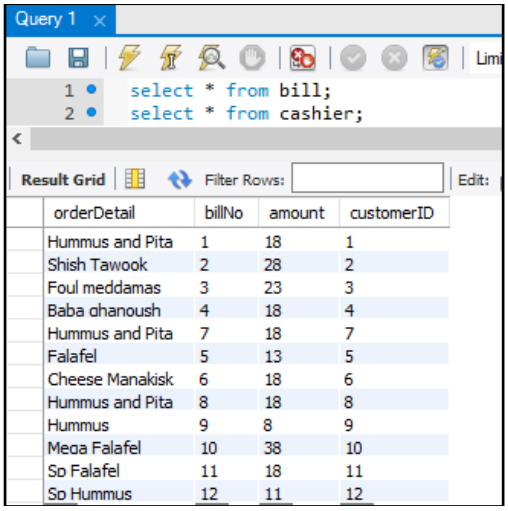
Creation of totalCashier table :



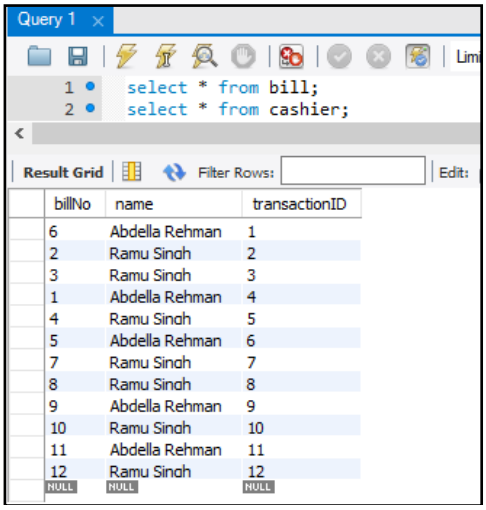
```
create table totalCashier (name varchar(50), totalAmount int);
    insert into totalCashier values('Abdella Rehman', 55);
    insert into totalCashier values('Ramu Singh', 115);

select * from totalCashier;
```

name	totalAmount
Abdella Rehman	55
Ramu Singh	115

Status of tables after few insertions :

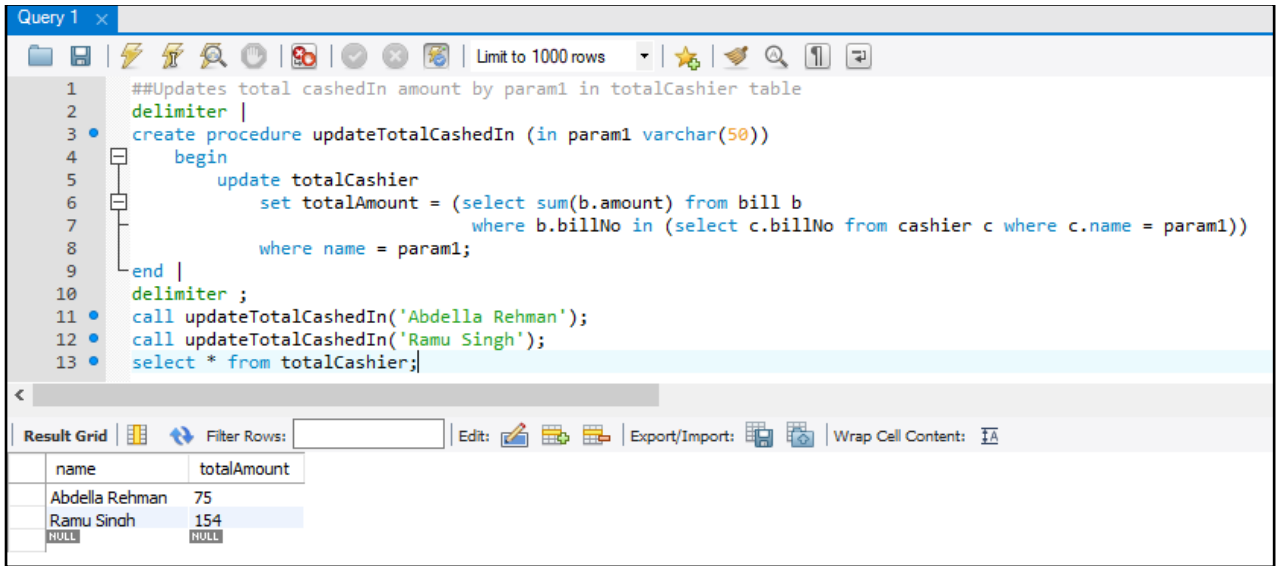


orderDetail	billNo	amount	customerID
Hummus and Pita	1	18	1
Shish Tawook	2	28	2
Foul meddamas	3	23	3
Baba ghanoush	4	18	4
Hummus and Pita	7	18	7
Falafel	5	13	5
Cheese Manakisk	6	18	6
Hummus and Pita	8	18	8
Hummus	9	8	9
Mega Falafel	10	38	10
Sp Falafel	11	18	11
Sp Hummus	12	11	12

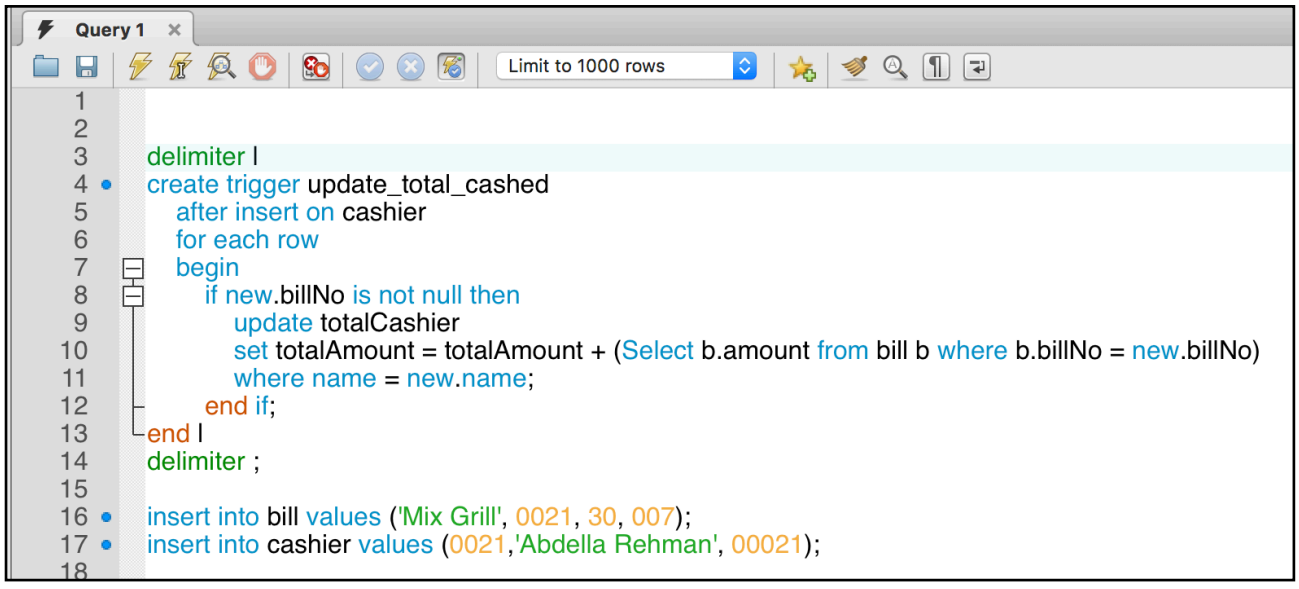


billNo	name	transactionID
6	Abdella Rehman	1
2	Ramu Singh	2
3	Ramu Singh	3
1	Abdella Rehman	4
4	Ramu Singh	5
5	Abdella Rehman	6
7	Ramu Singh	7
8	Ramu Singh	8
9	Abdella Rehman	9
10	Ramu Singh	10
11	Abdella Rehman	11
12	Ramu Singh	12
NULL	NULL	NULL

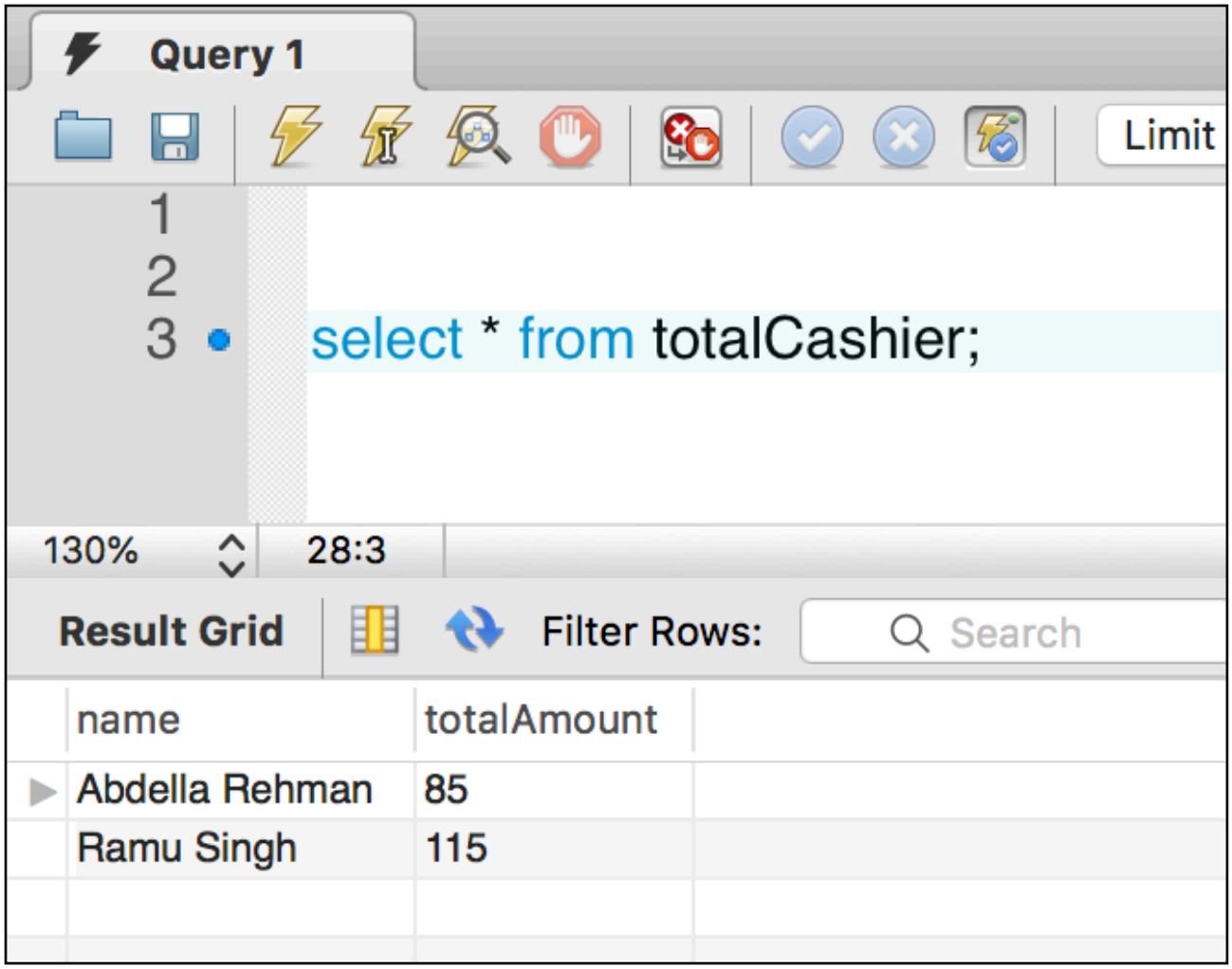
After creating and calling Procedure ;



Trigger to update the total cashed in amount by all the cashiers in the totalCashier table whenever a new bill is generated and added in the cashier register table :



Values updated in table totalCashier;



THANK YOU