

**AMERICAN INTERNATION UNIVERSITY OF BANGLADESH**

**(AIUB)**

Project Name : Food Delivery Management System

Course Name : Introduction To Database.

Course Instructor : Juena Ahmed Noshin.

Section : A.

|  |  |
| --- | --- |
| Group Members Name | ID |
| MD. Rahamatullah | 19-40946-2 |
| Maizul Islam | 19-41002-2 |
| Tasnova Tabassum Aroni | 19-41081-2 |

**CONTENTS:** **PAGE NO:**

* **Introduction 3**
* **Scenario 4**
* **ER Diagram 5**
* **Normalization 6**
* **Schema Diagram 18**
* **Table Creation 19**
* **Data Insertion 27**
* **Query Writing 34**
* **Relational Algebra 39**
* **Conclusion 40**

**INTRODUCTION:**

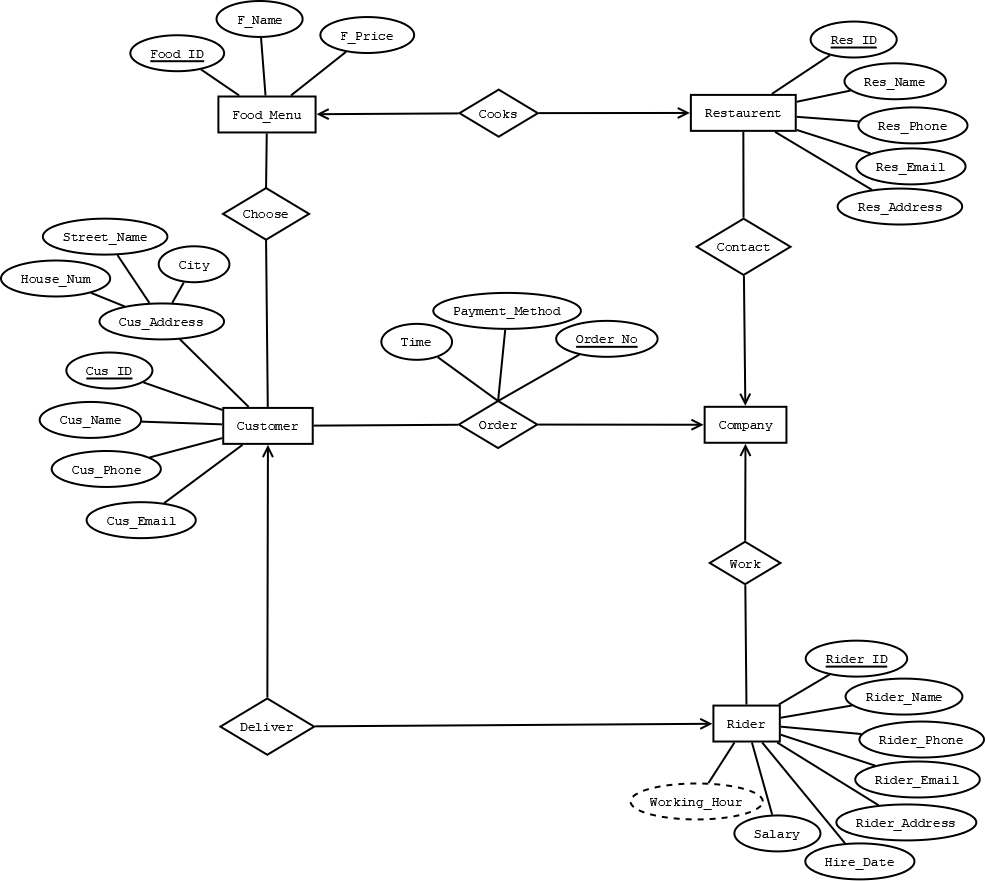
A database management system (DBMS) is a system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data. A DBMS makes it possible for end users to create, read, update and delete data in a database.

In our project (Food Delivery Management System) was created by the concept of DBMS.

**SCENARIO:**

In a “Food Delivery” management system, a Customer may order food from one Restaurant. One Restaurant may take order from many Customer. A Customer is identified by a Customer ID. The system also stores customers Name, Address, E-mail, Phone Number. Customers address is composed of House Number, Street name and City. A Restaurant is identified by Restaurant ID, Restaurant Name, Address, E-mail, Phone Number. Each Restaurant has a Food Menu. To identify a Food, the system stores Food ID along with Food Name. Food Price is also stored. While ordering to find the priority of order the Date and Time of order along with Order Number and Payment-Type is stored. Every Food is delivered by a Rider. A Rider can work only in one Company. The Company may have many Riders. A Rider is identified by Rider ID, Name, Address, E-mail and Phone Number, The System also stores Riders Salary, Join-Date and Working Hours.

**ER DIAGRAM:**



**NORMALIZATION:**

**CUSTOMER CHOOSE FOOD MENU (Many to Many)**

**Unnormalized Form (UNF) :**

Choose (Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail, House\_No, Street\_name, City, Food\_Id, F\_Name, F\_Price).

**1NF ( 1st Normalized Form ) :**

There is no multi valued attribute. Relation already in **1NF.**

(Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail, House\_No, Street\_name, City, Food\_Id, F\_Name, F\_Price).

**2NF ( 2nd Normalized Form ) :**

* Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail, House\_No, Street\_name, City.
* Food\_Id, F\_Name, F\_Price.

**3NF (3rd Normalized Form):**

* Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail.
* House\_No, Street\_name, City.
* Food\_Id, F\_Name, F\_Price.

**TABLE CREATION:**

* Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail, **A\_Id.**
* A\_Id, House\_No, Street\_name, City.
* Food\_Id, F\_Name, F\_Price.
* **Cus\_Id, Food\_Id.**

**CUSTOMER ORDER COMPANY(Many to One)**

**Unnormalized Form (UNF) :**

Order ( Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail, House\_No, Street\_name, City, Time, Payment\_Method, Order\_No).

**1NF ( 1st Normalized Form ) :**

There is no multi valued attribute. Relation already in **1NF.**

(Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail, House\_No, Street\_name, City, Time, Payment\_Method, Order\_No).

**2NF ( 2nd Normalized Form ) :**

* Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail, House\_No, Street\_name, City.
* Time, Payment\_Method, Order\_No.

**3NF (3rd Normalized Form):**

* Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail.
* House\_No, Street\_name, City.
* Time, Payment\_Method, Order\_No.

**TABLE CREATION:**

* Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail,**A\_Id**, **Order\_No**.
* A\_Id,House\_No, Street\_name, City.
* Time, Payment\_Method, Order\_No.

**COMPANY CONTACT RESTAURENT(One To Many)**

**Unnormalized Form (UNF) :**

Contact (Res\_Id, Res\_Name, Res\_Phone, Res\_E-mail, Res\_Add)

**1NF ( 1st Normalized Form ) :**

There is no multi valued attribute. Relation already in **1NF.**

(Res\_Id, Res\_Name, Res\_Phone, Res\_E-mail, Res\_Add).

**2NF ( 2nd Normalized Form ) :**

There is no Partial Dependency. Relation already in **2NF**.

* Res\_Id, Res\_Name, Res\_Phone, Res\_E-mail, Res\_Add.

**3NF (3rd Normalized Form):**

There is no transitive dependency. Relation already in **3NF**.

* Res\_Id, Res\_Name, Res\_Phone, Res\_E-mail, Res\_Add.

**TABLE CREATION:**

* Res\_Id, Res\_Name, Res\_Phone, Res\_E-mail, Res\_Add**.**

**RESTAURENT COOKS FOOD\_MENU(One to One)**

**Unnormalized Form (UNF) :**

Cook (Res\_Id, Res\_Name, Res\_Phone, Res\_E-mail, Res\_Add, Food\_Id, F\_Name, F\_Price).

**1NF ( 1st Normalized Form ) :**

There is no multi valued attribute. Relation already in **1NF.**

(Res\_Id, Res\_Name, Res\_Phone, Res\_E-mail, Res\_Add, Food\_Id, F\_Name, F\_Price).

**2NF ( 2nd Normalized Form ) :**

* Res\_Id, Res\_Name, Res\_Phone, Res\_Add, Res\_Street.
* Food\_Id, F\_Name, F\_Price.

**3NF (3rd Normalized Form):**

* Res\_Id, Res\_Name, Res\_Phone, Res\_E-mail, Res\_Add.
* Food\_Id, F\_Name, F\_Price.

**TABLE CREATION:**

* Res\_Id, Res\_Name, Res\_Phone, Res\_E-mail, Res\_Add**, Food\_Id.**
* Food\_Id, F\_Name, F\_Price.

**RIDER WORK CMPANY (Many to One)**

**Unnormalized Form (UNF) :**

Work (R\_Name, R\_Id, R\_Phone, R\_E-mail, R\_Add, H\_Date, Sal, Working\_H).

**1NF ( 1st Normalized Form ) :**

There is no multi valued attribute. Relation already in **1NF.**

(R\_Name, R\_Id, R\_Phone, R\_E-mail, R\_Add, H\_Date, Sal, Working\_H).

**2NF ( 2nd Normalized Form ) :**

There is no Partial Dependency. Relation already in **2NF**.

* R\_Name, R\_Id, R\_Phone, R\_E-mail, R\_Add, H\_Date, Sal, Working\_H.

**3NF (3rd Normalized Form):**

There is no transitive dependency. Relation already in **3NF**.

* R\_Name, R\_Id, R\_Phone, R\_E-mail, H\_Date, Sal, Working\_H, R\_Add.

**TABLE CREATION:**

* R\_Name, R\_Id, R\_Phone, R\_E-mail, H\_Date, Sal, Working\_H, R\_Add.

**RIDER DELIVER CUSTOMER (One to One)**

**Unnormalized Form (UNF) :**

Deliver (R\_Name, R\_Id, R\_Phone, R\_E-mail, R\_Add, H\_Date, Sal, Working\_H, Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail, House\_No, Street\_name, City).

**1NF ( 1st Normalized Form ) :**

There is no multi valued attribute. Relation already in **1NF.**

(R\_Name, R\_Id, R\_Phone, R\_E-mail, R\_Add, H\_Date, Sal, Working\_H, Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail, House\_No, Street\_name, City).

**2NF ( 2nd Normalized Form ) :**

* R\_Name, R\_Id, R\_Phone, R\_E-mail, R\_Add, H\_Date, Sal, Working\_H.
* Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail, House\_No, Street\_name, City.

**3NF (3rd Normalized Form):**

* R\_Name, R\_Id, R\_Phone, R\_E-mail, H\_Date, Sal, Working\_H, R\_Add.
* Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail.
* House\_No, Street\_name, City.

**TABLE CREATION:**

* R\_Name, R\_Id, R\_Phone, R\_E-mail, H\_Date, Sal, Working\_H,R\_Add, **Cus\_Id.**
* Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail, **A\_Id**
* A\_Id, House\_No, Street\_name, City.

**TEMPORARY TABLES:**

* ~~Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail,~~ **~~A\_Id.~~**
* ~~A\_Id, House\_No, Street\_name, City.~~
* ~~Food\_Id, F\_Name, F\_Price.~~
* **Cus\_Id, Food\_Id.**
* Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail,**A\_Id**, **Order\_No**.
* A\_Id,House\_No, Street\_name, City.
* Time, Payment\_Method, Order\_No.
* ~~Res\_Id, Res\_Name, Res\_Phone, Res\_E-mail, Res\_Add~~**~~.~~**
* Res\_Id, Res\_Name, Res\_Phone, Res\_E-mail, Res\_Add**, Food\_Id.**
* Food\_Id, F\_Name, F\_Price.
* ~~R\_Name, R\_Id, R\_Phone, R\_E-mail, H\_Date, Sal, Working\_H, R\_Add.~~
* R\_Name,R\_Id,R\_Phone,R\_E-mail,H\_Date,Sal,Working\_H,R\_Add, **Cus\_Id.**
* ~~Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail,~~ **~~A\_Id.~~**
* ~~A\_Id, House\_No, Street\_name, City.~~

**FINAL TABLE:**

* Cus\_Id, Cus\_Name, Cus\_Phone, Cus\_E-mail,**A\_Id**, **Order\_No**.
* Time, Payment\_Method, Order\_No.
* A\_Id,House\_No, Street\_name, City.
* Food\_Id, F\_Name, F\_Price.
* **Cus\_Id, Food\_Id.**
* Res\_Id, Res\_Name, Res\_Phone, Res\_E-mail, Res\_Add**, Food\_Id.**
* R\_Name,R\_Id,R\_Phone,R\_E-mail,H\_Date,Sal,Working\_H,R\_Add, **Cus\_Id.**

**SCHEMA DIAGRAM:**

CusId

CusName

CusPhone

CusMail

**A\_Id**

**OrderNo**

Customer

OrderNo

Payment

Time

OrderList

A\_Id

HouseNo

StreetName

City

CusAdd

F\_Id

Fname

Fprice

Food

**CusId**

**F\_Id**

ForeignKeys

ResId

ResName

ResPhone

ResMail

ResAdd

**F\_Id**

Restaurant

R\_Id

RName

Rphone

RMail

Hiredate

Sal

WorkingHour

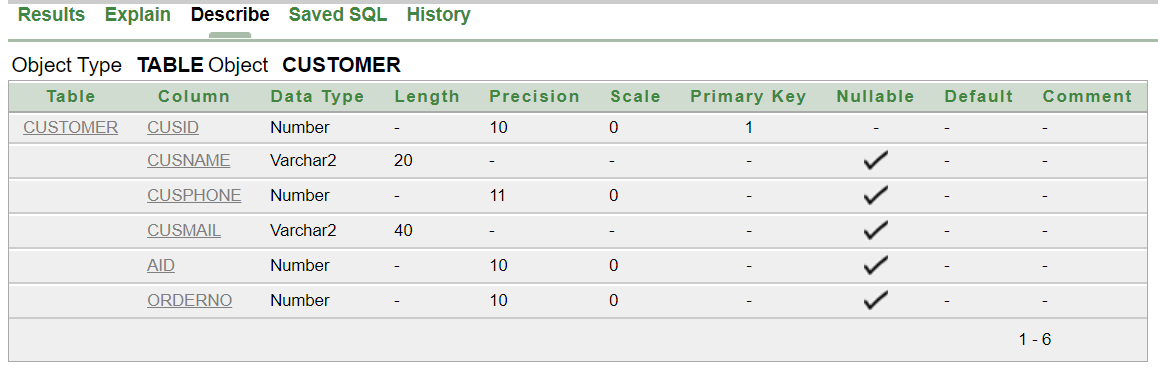
RAdd

**CusId**

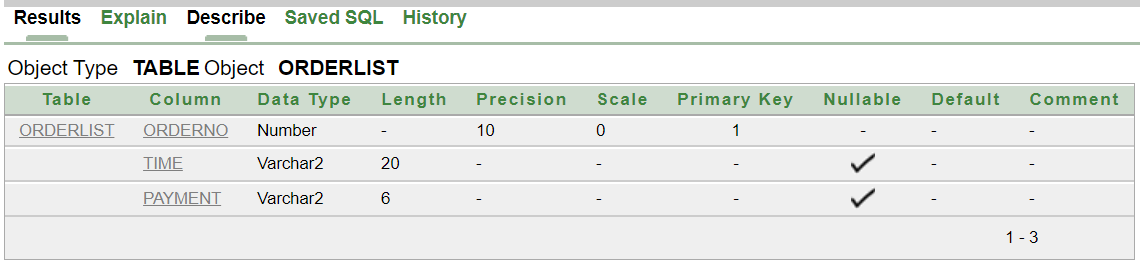
Rider

**TABLE CREATION:**

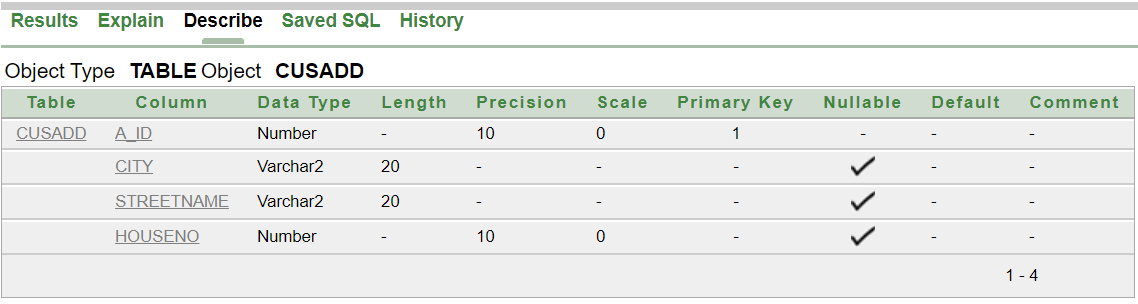
1. create table Customer (CusId number (10) Primary key,CusName varchar2(20),CusPhone number(11),CusMail varchar2(40),AId number(10), OrderNo number(10));



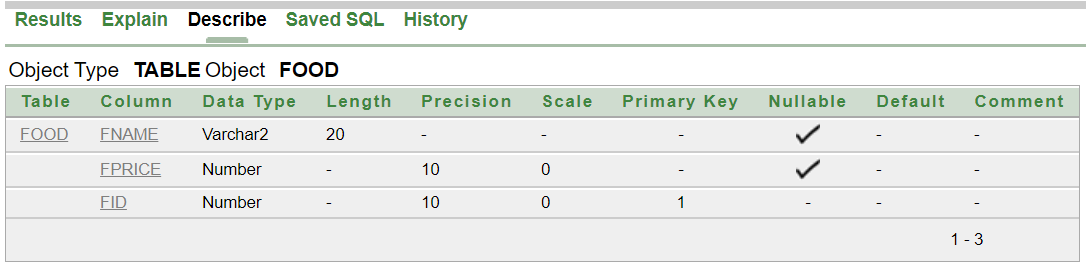
2. create table OrderList (OrderNo number (10) Primary Key, Time varchar2(20), Payment varchar2(6) check (Payment='COD' or Payment='Card'));



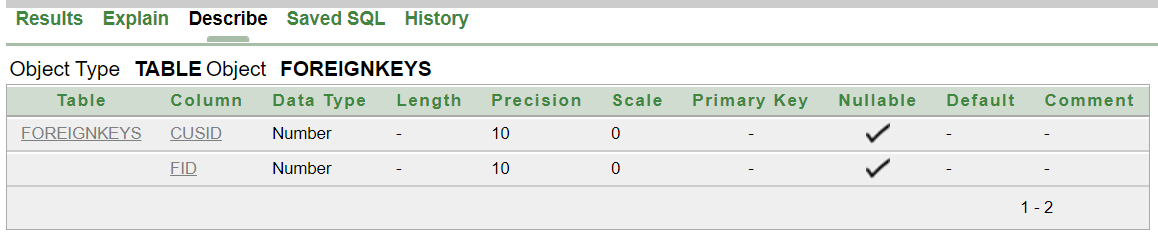
3. Create table CusAdd (A\_Id number (10) Primary Key, City varchar2(20), StreetName varchar2(20), HouseNo number (10));



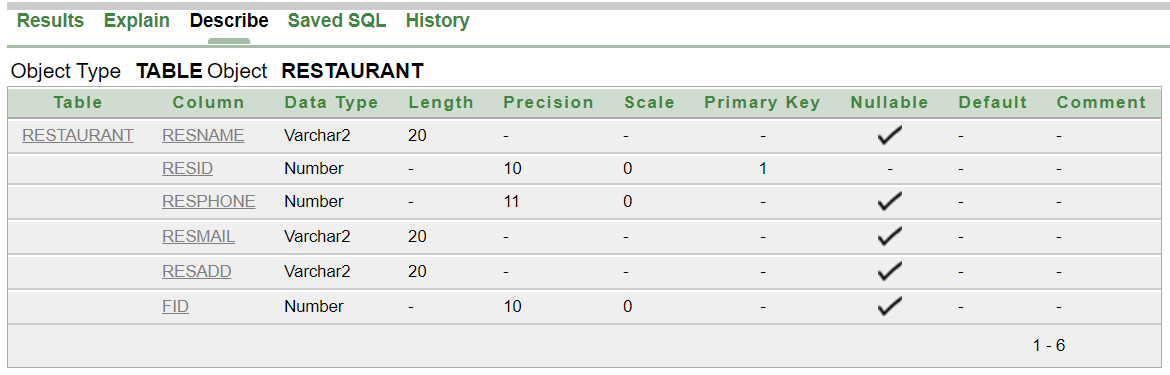
4. create table Food (Fname varchar2(20), Fprice number (10), Fid number (10) Primary Key);



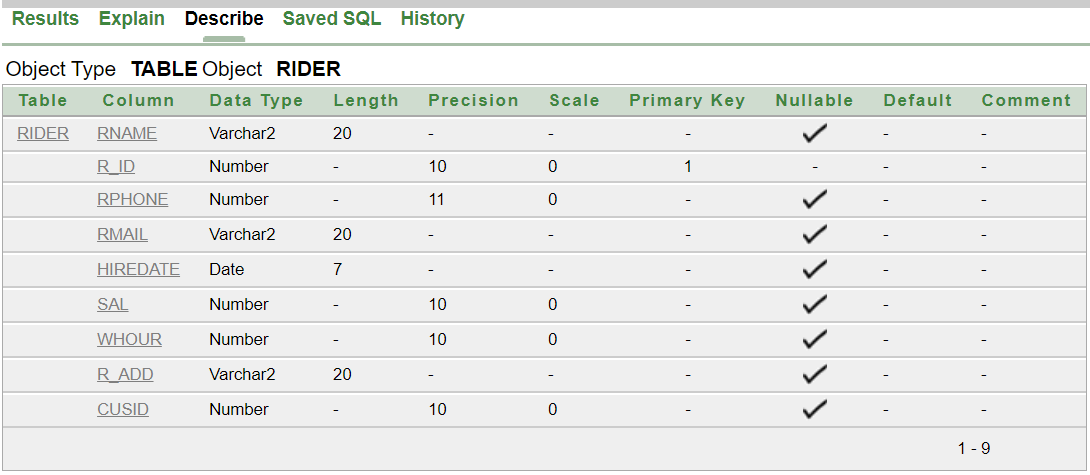
5. Create table ForeignKeys (CusId number (10), Fid number (10));



6. Create table Restaurant (ResName varchar2(20), ResId number (10) Primary Key, ResPhone number (11), ResMail varchar2(20), ResAdd varchar2(20), Fid number (10));



7. Create table Rider (Rname varchar2(20), R\_Id number (10) Primary Key, Rphone number (11), Rmail varchar2(20), HireDate Date, sal number (10), Whour number (10), R\_Add varchar2(20), CusId number (10));



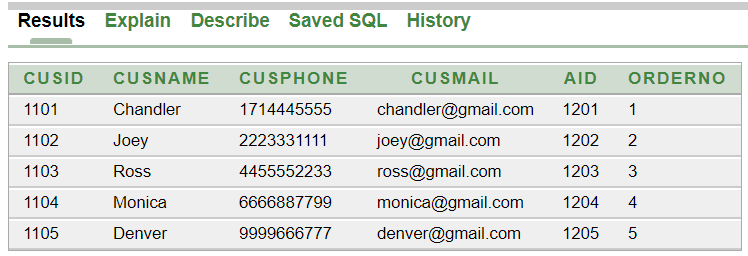
**CONTRAINTS:**

1. Alter table ForeignKeys Add Constraint FK1 Foreign Key(CusId) References Customer(CusId);
2. Alter table ForeignKeys Add Constraint FK2 Foreign Key(Fid) References Food(Fid);
3. Alter table Customer Add Constraint FK3 Foreign Key(AId) References CusAdd(A\_Id);
4. Alter table Customer Add Constraint FK4 Foreign Key(OrderNo) References OrderList(OrderNo);
5. Alter table Restaurant Add Constraint FK5 Foreign Key(Fid) References Food(Fid);
6. Alter table Rider Add Constraint FK6 Foreign Key(CusId) References Customer(CusId);

**DATA INSERTION:**

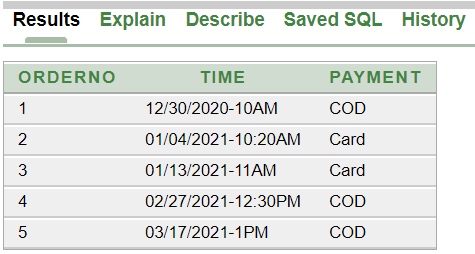
**CUSTOMER TABLE:**

* insert into customer values('1101','Chandler','1714445555','chandler@gmail.com','1201','001');
* insert into customer values('1102','Joey','2223331111','joey@gmail.com','1202','002');
* insert into customer values('1103','Ross','4455552233','ross@gmail.com','1203','003');
* insert into customer values('1104','Monica','6666887799','monica@gmail.com','1204','004');
* insert into customer values('1105','Denver','9999666777','denver@gmail.com','1205','005');



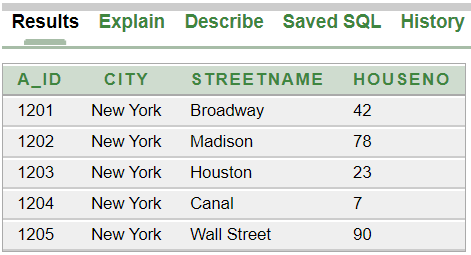
**ORDERLIST TABLE:**

* insert into orderlist values('001','12/30/2020-10AM','COD');
* insert into orderlist values('002','01/04/2021-10:20AM','Card');
* insert into orderlist values('003','01/13/2021-11AM','Card');
* insert into orderlist values('004','02/27/2021-12:30PM','COD');
* insert into orderlist values('005','03/17/2021-1PM','COD');



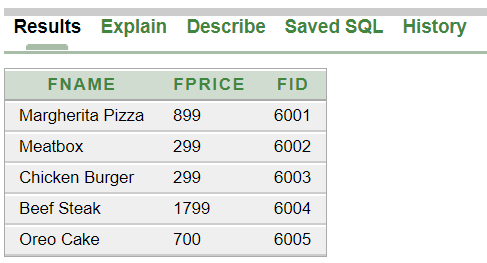
**CUSADD TABLE:**

* insert into cusadd values( '1201','New York','Broadway','42');
* insert into cusadd values( '1202','New York','Madison','78');
* insert into cusadd values( '1203','New York','Houston','23');
* insert into cusadd values( '1204','New York','Canal','7');
* insert into cusadd values( '1205','New York','Wall Street','90');



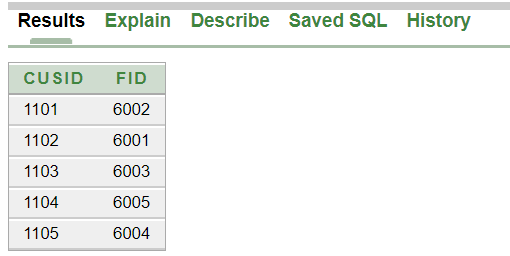
**FOOD TABLE:**

* insert into food values ('Margherita Pizza','899','6001');
* insert into food values ('Meatbox','299','6002');
* insert into food values ('Chicken Burger','299','6003');
* insert into food values ('Beef Steak','1799','6004');
* insert into food values ('Oreo Cake','700','6005');



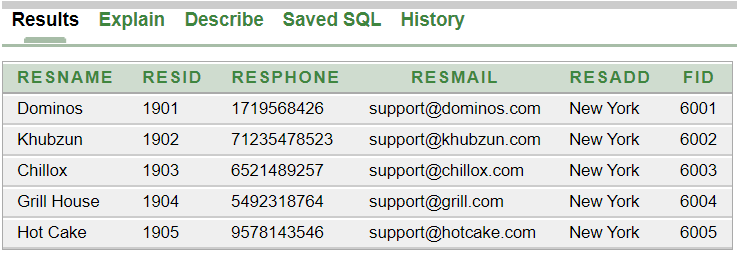
**FOREIGNKEYS TABLE:**

* insert into foreignkeys values('1101','6002');
* insert into foreignkeys values('1102','6001');
* insert into foreignkeys values('1103','6003');
* insert into foreignkeys values('1104','6005');
* insert into foreignkeys values('1105','6004');



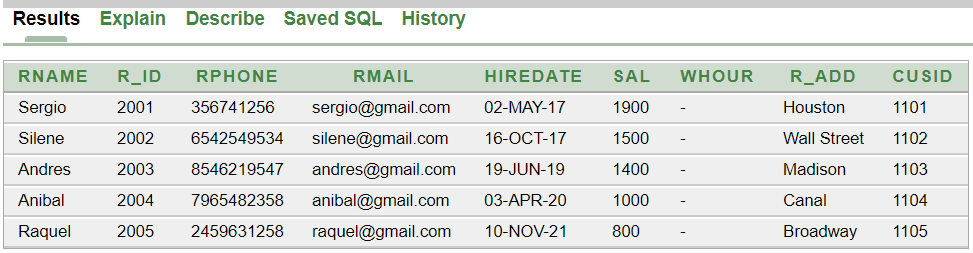
**REATAURANT TABLE:**

* insert into restaurant values('Dominos','1901','1719568426','support@dominos.com','New York','6001');
* insert into restaurant values('Khubzun','1902','71235478523','support@khubzun.com','New York','6002');
* insert into restaurant values('Chillox','1903','6521489257','support@chillox.com','New York','6003');
* insert into restaurant values('Grill House','1904','5492318764','support@grill.com','New York','6004');
* insert into restaurant values('Hot Cake','1905','9578143546','support@hotcake.com','New York','6005');



**RIDER TABLE:**

* insert into rider values('Sergio','2001','356741256','sergio@gmail.com',to\_date('02-05-2017','dd-mm-yyyy'),'1900','','Houston','1101');
* insert into rider values('Silene','2002','6542549534','silene@gmail.com',to\_date('16-10-2017','dd-mm-yyyy'),'1500','','Wall Street','1102');
* insert into rider values('Andres','2003','8546219547','andres@gmail.com',to\_date('19-06-2019','dd-mm-yyyy'),'1400','','Madison','1103');
* insert into rider values('Anibal','2004','7965482358','anibal@gmail.com',to\_date('03-04-2020','dd-mm-yyyy'),'1000','','Canal','1104');
* insert into rider values('Raquel','2005','2459631258','raquel@gmail.com',to\_date('10-11-2021','dd-mm-yyyy'),'800','','Broadway','1105');

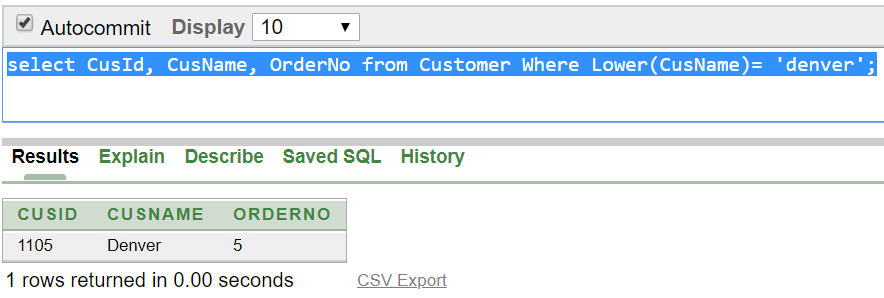


**QUERY WRITING:**

**Single Row Function:**

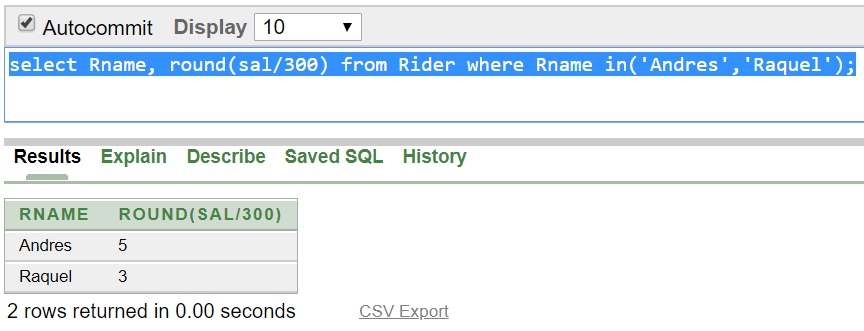
**Ques:** Display the Customer Id, Name and Order No for Customer Denver.

**Ans:** select CusId, CusName, OrderNo from Customer Where Lower (CusName)= 'denver';



**Ques:**  Calculate and display the rounded salary of Rider Andres and Raquel after dividing salary by 300.

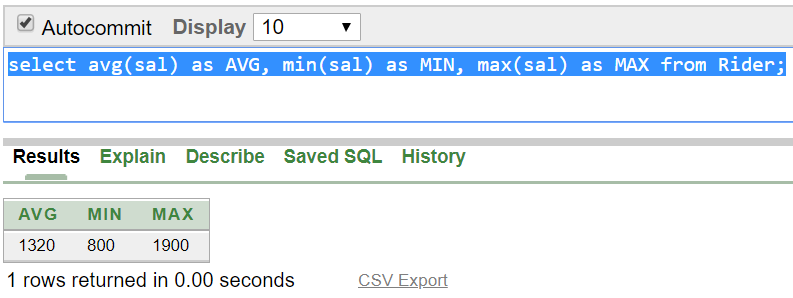
**Ans:** select Rname, round(sal/300) from Rider where Rname in('Andres','Raquel');



**Group Function:**

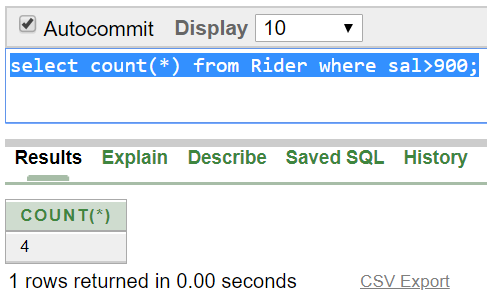
**Ques:** Find the average, minimum and maximum salary of the Riders. Label the columns AVG, MIN and MAX respectively.

**Ans:** select avg(sal) as AVG, min(sal) as MIN, max(sal) as MAX from Rider;



**Ques:** Display the number of Rider whose Sal is greater than 900 Tk.

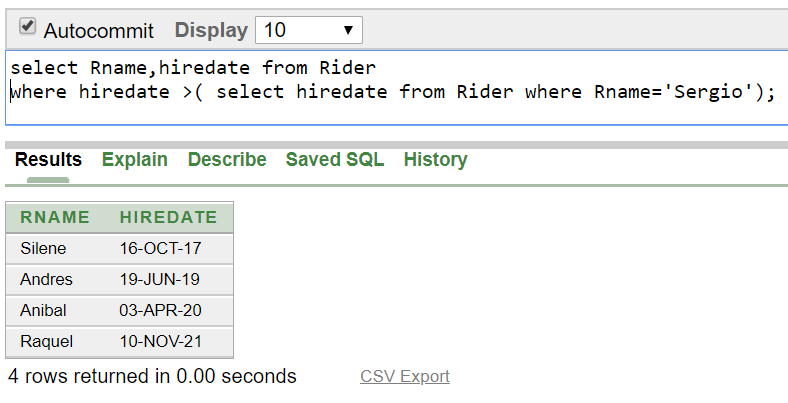
**Ans:** select count (\*) from Rider where sal>900;



**Subquery:**

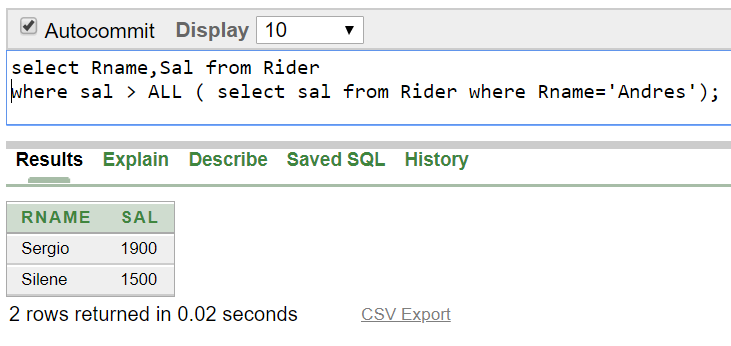
**Ques:** Display the Rider names and hire date who joined after Sergio.

**Ans:**  select Rname,hiredate from Rider where hiredate >( select hiredate from Rider where Rname='Sergio');



**Ques:**  Display the Rider names and salary that earn a salary that is higher than the salary of Andres.

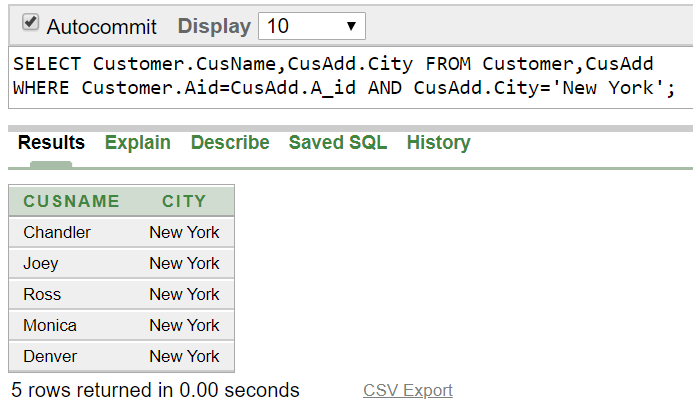
**Ans:**  select Rname,Sal from Rider where sal > ALL ( select sal from Rider where Rname='Andres');



**Joining:**

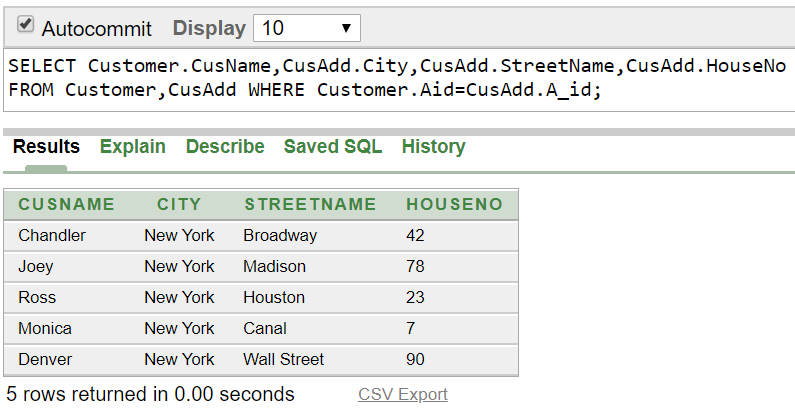
**Ques:** Display the name of all the customers who lives in New York.

**Ans:** SELECT Customer.CusName,CusAdd.City FROM Customer,CusAdd WHERE Customer.Aid=CusAdd.A\_id AND CusAdd.City='New York';



**Ques:** Write a query to display the Customer name, city,HouseNo and Street name for all Customer.

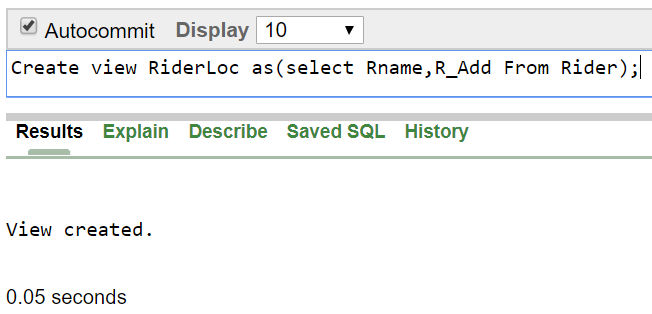
**Ans:**selectCustomer.CusName,CusAdd.City,CusAdd.StreetName,CusAdd.HouseNo FROM Customer,CusAdd WHERE Customer.Aid=CusAdd.A\_id;



**View:**

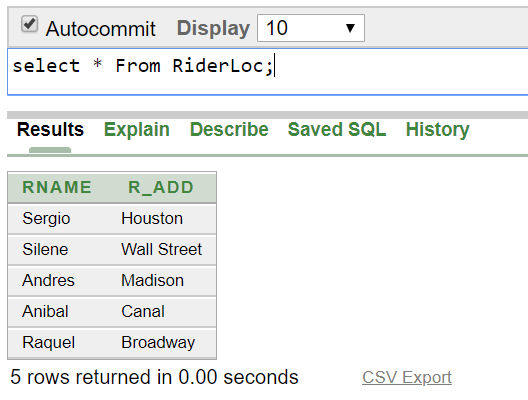
**Ques:** Create a view called RiderLoc based on the Rname and R\_add from the Rider table.

**Ans:** Create view RiderLoc as (select Rname,R\_Add From Rider);



**Ques:**  Display all data From the RiderLoc View.

**Ans:** Select \* From RiderLoc;



**Relational Algebra:**

Ques: Find the name of the customer which Id is 1103.

Ans: ∏CusName (σCusId= “1103” (Customer))

Ques: Find the rider who lives in Houston.

Ans: ∏Rname (σR\_Add= “Houston” (Rider))

Ques: Find the name of food which price is less than 1000.

Ans: ∏Fname (σprice<“1000” (Food))

Ques: Find the name of all customer.

Ans: ∏CusName (Customer))

Ques: Find the Sal of rider Andres.

Ans: . ∏Sal (σRname= “Andres” (Rider))

**Conclusion:**

After a lot of hard work together, we finally created a Food Delivery Management System. Considering current circumstances it was really tough for us to communicate with each other but finally we finished it. Hopefully in future we will be able to create a real Database Management System.