Food Delivery Management

CONTENTS: **PAGE NO:**

INTRODUCTION:

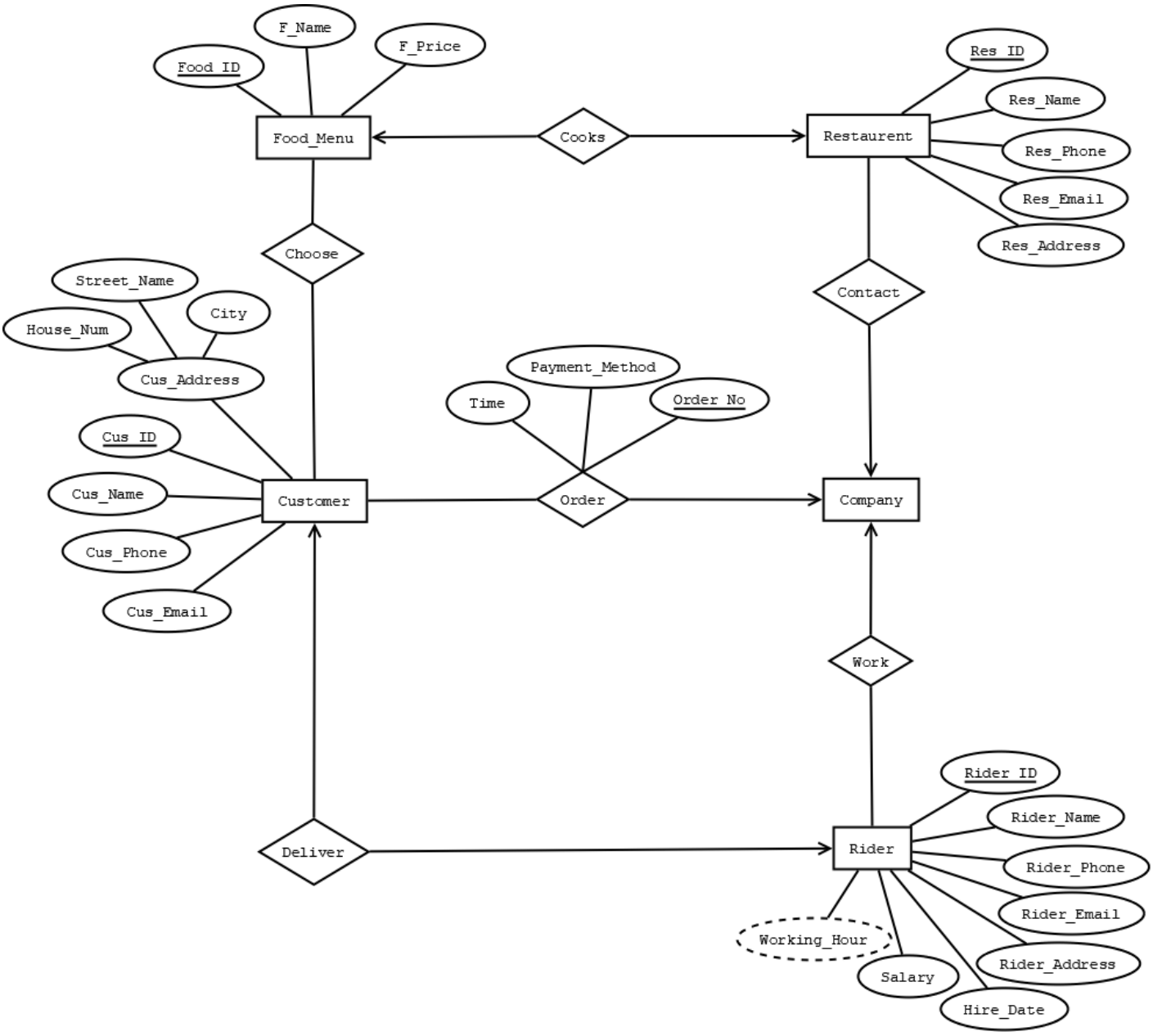
A relational database management system (RDBMS) is a system software for creating and managing databases. The RDBMS provides users and programmers with a systematic way to create, retrieve, update and manage data. A RDBMS makes it possible for end users to create, read, update and delete data in a database.

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

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:


F_Name
F_Price
Food_ID
Food_Menu
Cooks
Restaurent
Res_ID
Res_Name
Res_Phone
Res_Email
Res_Address
Choose
Contact
Street_Name
City
House_Num
Cus_Address
Payment_Method
Time
Order_No
Cus_ID
Cus_Name
Customer
Order
Company
Cus_Phone
Cus_Email
Work
Rider_ID
Rider_Name
Deliver
Rider
Rider_Phone
Rider_Email
Working_Hour
Rider_Address
Salary
Hire_Date

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:

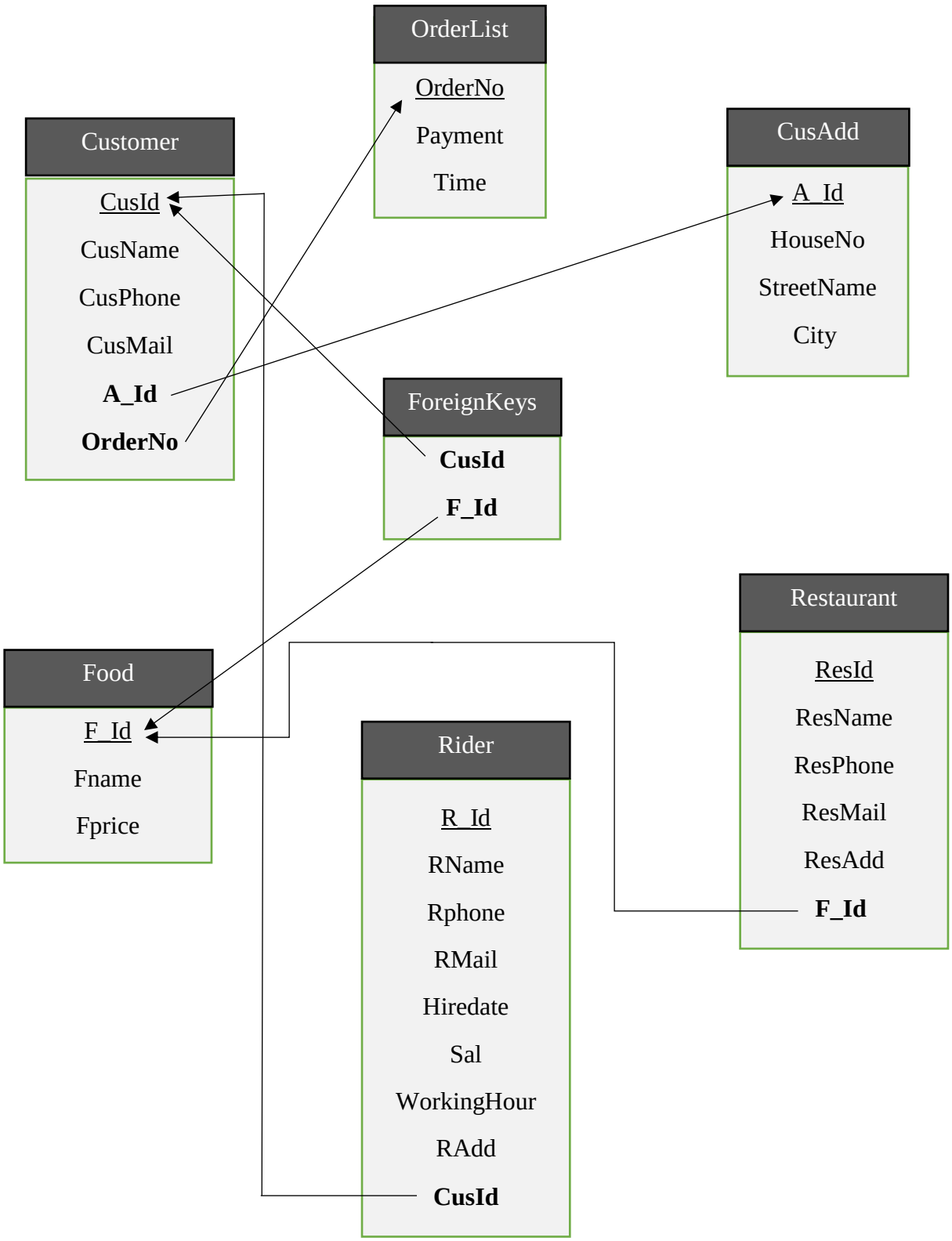

OrderList
OrderNo
Payment
Time
Customer
CusId
CusName
CusPhone
CusMail
A_Id
OrderNo
CusAdd
A_Id
HouseNo
StreetName
City
ForeignKeys
CusId
F_Id
Restaurant
ResId
ResName
ResPhone
ResMail
ResAdd
F_Id
Food
F_Id
Fname
Fprice
Rider
R_Id
RName
Rphone
RMail
Hiredate
Sal
WorkingHour
RAdd
CusId

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));

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **CUSTOMER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CUSID	Number	-	10	0	1	-	-	-
	CUSNAME	Varchar2	20	-	-	-	✓	-	-
	CUSPHONE	Number	-	11	0	-	✓	-	-
	CUSMAIL	Varchar2	40	-	-	-	✓	-	-
	AID	Number	-	10	0	-	✓	-	-
	ORDERNO	Number	-	10	0	-	✓	-	-

1 - 6

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

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **ORDERLIST**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDERLIST	ORDERNO	Number	-	10	0	1	-	-	-
	TIME	Varchar2	20	-	-	-	✓	-	-
	PAYMENT	Varchar2	6	-	-	-	✓	-	-

1 - 3

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

Results Explain **Describe** Saved SQL History

Object Type **TABLE** Object **CUSADD**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSADD	A_ID	Number	-	10	0	1	-	-	-
	CITY	Varchar2	20	-	-	-	✓	-	-
	STREETNAME	Varchar2	20	-	-	-	✓	-	-
	HOUSENO	Number	-	10	0	-	✓	-	-

1 - 4

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

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **FOOD**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FOOD	FNAME	Varchar2	20	-	-	-	✓	-	-
	FPRICE	Number	-	10	0	-	✓	-	-
	FID	Number	-	10	0	1	-	-	-

1 - 3

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

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **FOREIGNKEYS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FOREIGNKEYS	CUSID	Number	-	10	0	-	✓	-	-
	FID	Number	-	10	0	-	✓	-	-

1 - 2

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

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **RESTAURANT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RESTAURANT	RESNAME	Varchar2	20	-	-	-	✓	-	-
	RESID	Number	-	10	0	1	-	-	-
	RESPHONE	Number	-	11	0	-	✓	-	-
	RESMAIL	Varchar2	20	-	-	-	✓	-	-
	RESADD	Varchar2	20	-	-	-	✓	-	-
	FID	Number	-	10	0	-	✓	-	-

1 - 6

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));

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **RIDER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RIDER	RNAME	Varchar2	20	-	-	-	✓	-	-
	R_ID	Number	-	10	0	1	-	-	-
	RPHONE	Number	-	11	0	-	✓	-	-
	RMAIL	Varchar2	20	-	-	-	✓	-	-
	HIREDATE	Date	7	-	-	-	✓	-	-
	SAL	Number	-	10	0	-	✓	-	-
	WHOUR	Number	-	10	0	-	✓	-	-
	R_ADD	Varchar2	20	-	-	-	✓	-	-
	CUSID	Number	-	10	0	-	✓	-	-

1 - 9

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');

Results Explain Describe Saved SQL History

CUSID	CUSNAME	CUSPHONE	CUSMAIL	AID	ORDERNO
1101	Chandler	1714445555	chandler@gmail.com	1201	1
1102	Joey	2223331111	joey@gmail.com	1202	2
1103	Ross	4455552233	ross@gmail.com	1203	3
1104	Monica	6666887799	monica@gmail.com	1204	4
1105	Denver	9999666777	denver@gmail.com	1205	5

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');

Results Explain Describe Saved SQL History

ORDERNO	TIME	PAYMENT
1	12/30/2020-10AM	COD
2	01/04/2021-10:20AM	Card
3	01/13/2021-11AM	Card
4	02/27/2021-12:30PM	COD
5	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');

Results Explain Describe Saved SQL History

A_ID	CITY	STREETNAME	HOUSENO
1201	New York	Broadway	42
1202	New York	Madison	78
1203	New York	Houston	23
1204	New York	Canal	7
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');

Results Explain Describe Saved SQL History

FNAME	FPRICE	FID
Margherita Pizza	899	6001
Meatbox	299	6002
Chicken Burger	299	6003
Beef Steak	1799	6004
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');

Results Explain Describe Saved SQL History

CUSID	FID
1101	6002
1102	6001
1103	6003
1104	6005
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');

Results Explain Describe Saved SQL History

RESNAME	RESID	RESPHONE	RESMAIL	RESADD	FID
Dominos	1901	1719568426	support@dominos.com	New York	6001
Khubzun	1902	71235478523	support@khubzun.com	New York	6002
Chillox	1903	6521489257	support@chillox.com	New York	6003
Grill House	1904	5492318764	support@grill.com	New York	6004
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');

Results Explain Describe Saved SQL History

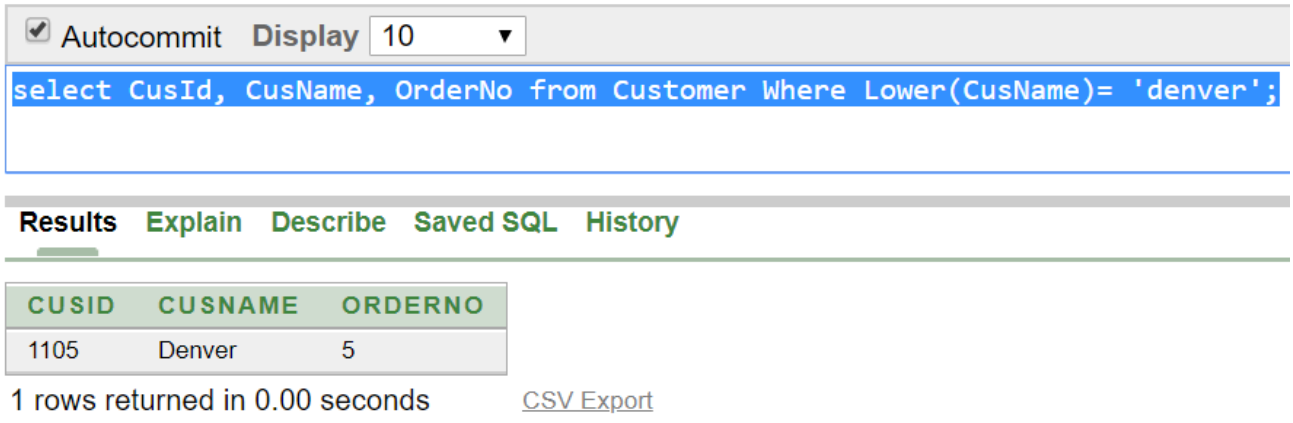
RNAME	R_ID	RPHONE	RMAIL	HIREDATE	SAL	WHOUR	R_ADD	CUSID
Sergio	2001	356741256	sergio@gmail.com	02-MAY-17	1900	-	Houston	1101
Silene	2002	6542549534	silene@gmail.com	16-OCT-17	1500	-	Wall Street	1102
Andres	2003	8546219547	andres@gmail.com	19-JUN-19	1400	-	Madison	1103
Anibal	2004	7965482358	anibal@gmail.com	03-APR-20	1000	-	Canal	1104
Raquel	2005	2459631258	raquel@gmail.com	10-NOV-21	800	-	Broadway	1105

QUERY WRITING:

Single Row Function:

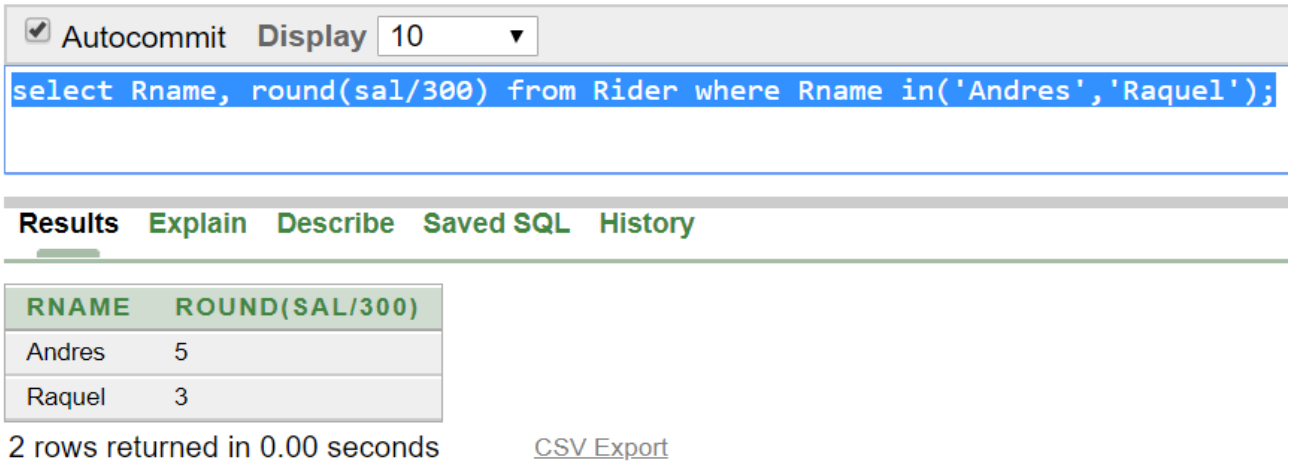
Ques: Display the Customer Id, Name and Order No for Customer Denver.

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.

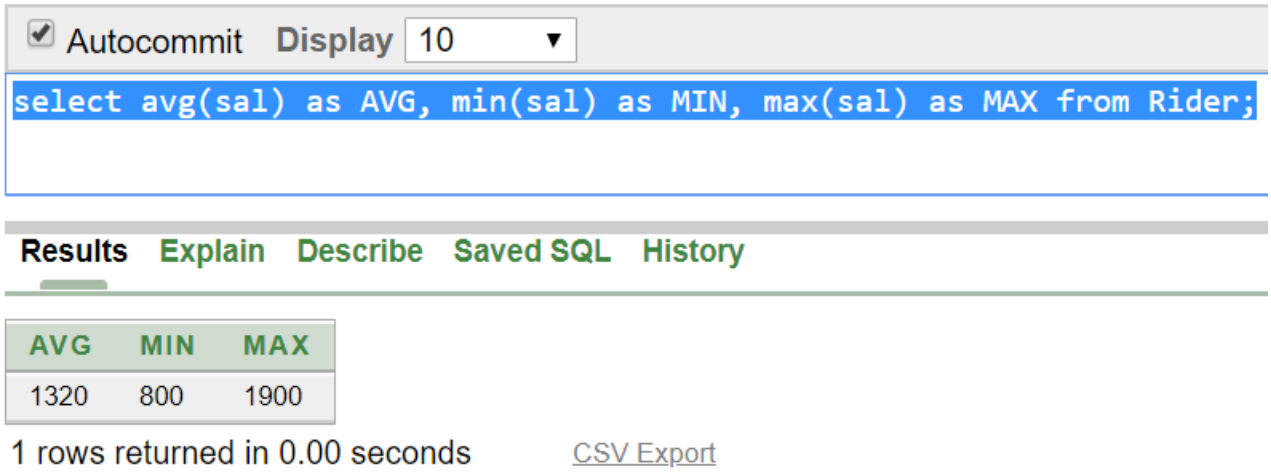
Ans: select Rname, round(sal/300) from Rider where Rname in('Andres','Raquel');



Group Function:

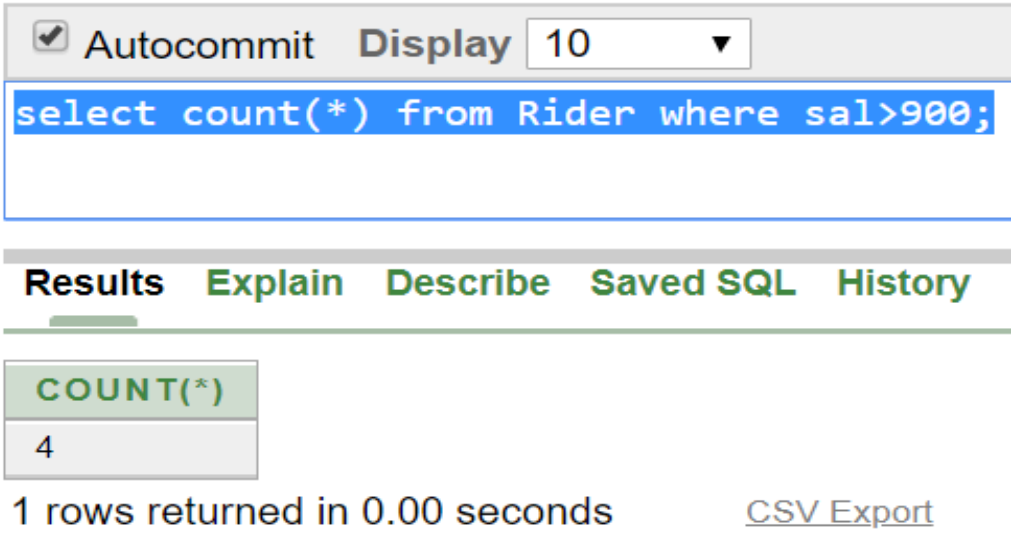
Ques: Find the average, minimum and maximum salary of the Riders. Label the columns AVG, MIN and MAX respectively.

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.

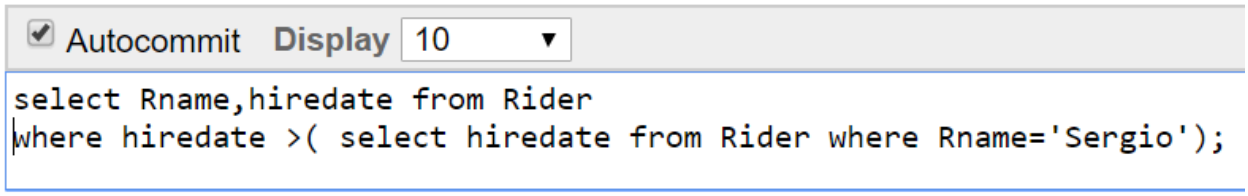
Ans: select count (*) from Rider where sal>900;



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');



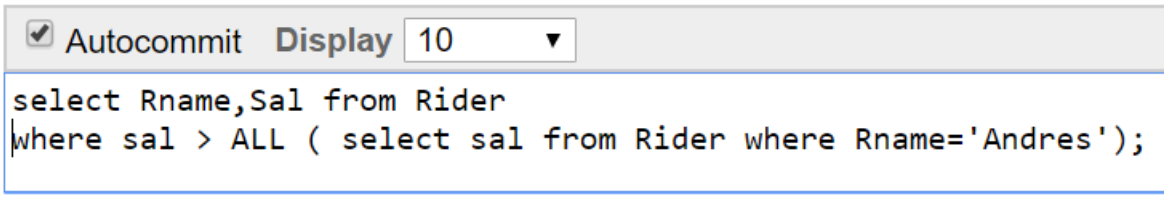
Results Explain Describe Saved SQL History

RNAME	HIREDATE
Silene	16-OCT-17
Andres	19-JUN-19
Anibal	03-APR-20
Raquel	10-NOV-21

4 rows returned in 0.00 seconds CSV Export

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');



Results Explain Describe Saved SQL History

RNAME	SAL
Sergio	1900
Silene	1500

2 rows returned in 0.02 seconds CSV Export

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';

Autocommit Display 10

```
SELECT Customer.CusName,CusAdd.City FROM Customer,CusAdd
WHERE Customer.Aid=CusAdd.A_id AND CusAdd.City='New York';
```

Results Explain Describe Saved SQL History

CUSNAME	CITY
Chandler	New York
Joey	New York
Ross	New York
Monica	New York
Denver	New York

5 rows returned in 0.00 seconds CSV Export

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;

Autocommit Display 10

```
SELECT Customer.CusName,CusAdd.City,CusAdd.StreetName,CusAdd.HouseNo
FROM Customer,CusAdd WHERE Customer.Aid=CusAdd.A_id;
```

Results Explain Describe Saved SQL History

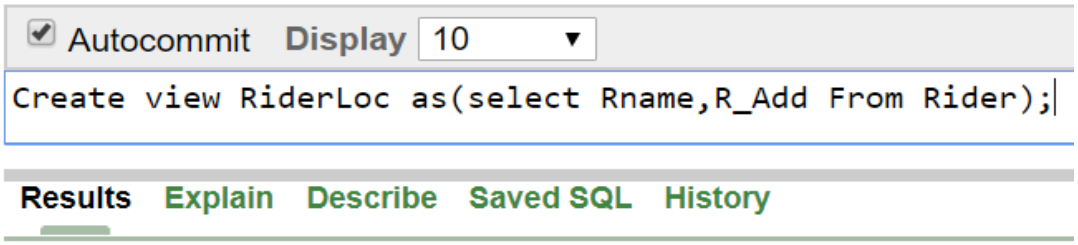
CUSNAME	CITY	STREETNAME	HOUSENO
Chandler	New York	Broadway	42
Joey	New York	Madison	78
Ross	New York	Houston	23
Monica	New York	Canal	7
Denver	New York	Wall Street	90

5 rows returned in 0.00 seconds CSV Export

View:

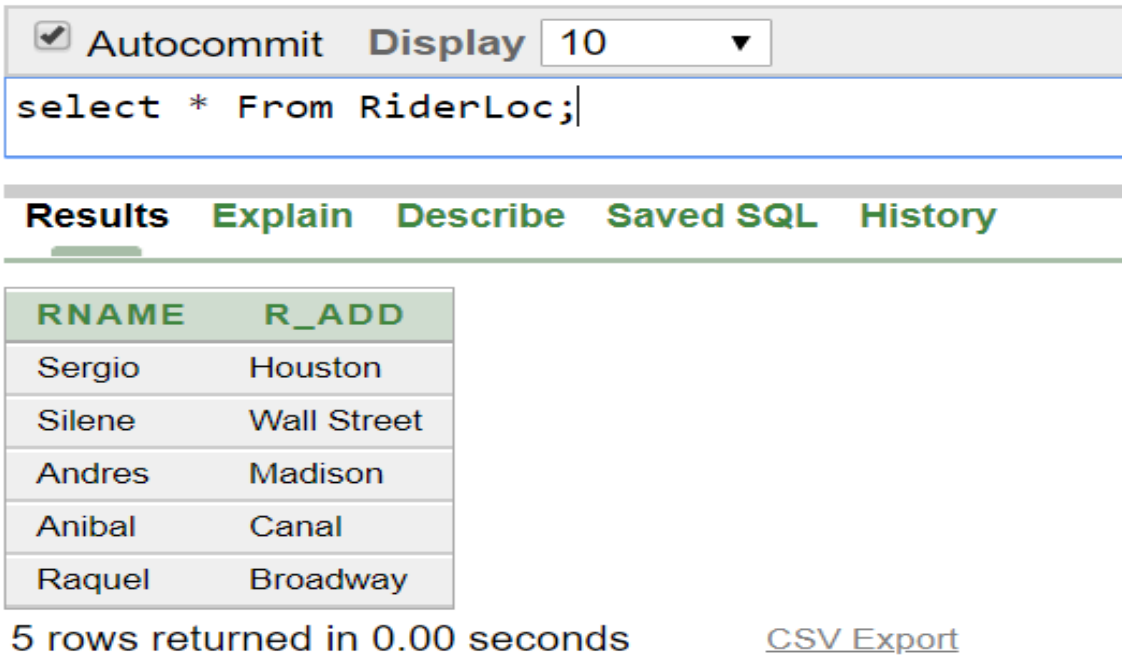
Ques: Create a view called RiderLoc based on the Rname and R_add from the Rider table.

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: $\prod_{\text{CusName}} (\sigma_{\text{CusId= “1103”}} (\text{Customer}))$

Ques: Find the rider who lives in Houston.

Ans: $\prod_{\text{Rname}} (\sigma_{\text{R\_Add= “Houston”}} (\text{Rider}))$

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

Ans: $\prod_{\text{Fname}} (\sigma_{\text{price<“1000”}} (\text{Food}))$

Ques: Find the name of all customer.

Ans: $\prod_{\text{CusName}} (\text{Customer}))$

Ques: Find the Sal of rider Andres.

Ans: . $\prod_{\text{Sal}} (\sigma_{\text{Rname= “Andres”}} (\text{Rider}))$