



## **DATABASE (PROJECT)**

**Course Name:** INTRODUCTION TO DATABASE

**Course Instructor:** RIFAT TASNIM ANANNYA

**Semester:** SPRING (2018-2019)

**Department:** CSE

**Sec:** I

**Project Name:** RESTAURANT MANAGEMENT  
SYSTEM

**Group Member:**

No	Name	ID
01	FOYSAL, MD ABU ZEHAD	18-37514-1
02	NOWRIN MUHAJIB SHAILEE	18-37259-1
03	SAFINA ISLAM	18-36395-1
04	CHOWDHURY, RIDWAN	18-37598-1

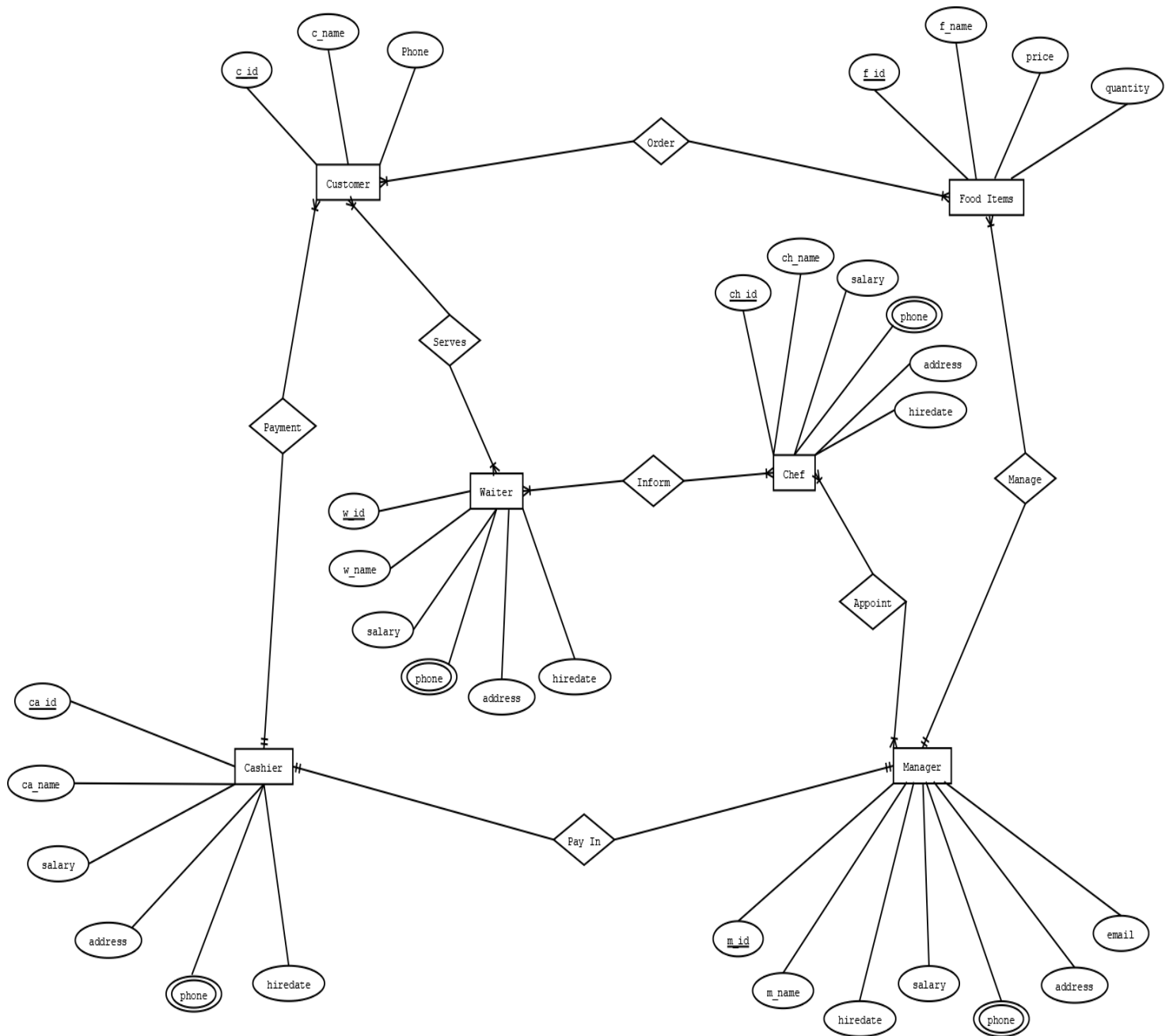
# CONTENTS

No	Name	Page No.
01	Scenario	03
02	E-R Diagram	04
03	Normalization	05-09
04	Creation	10-16
05	Insertion	16-22
06	Constraints	23-31
07	Query Question	31-32

## **Scenario**

**An owner of a restaurant wants to design their database system. By which they can manage a small restaurant properly. The database primarily focused on the customers and interaction between the customers and the restaurant elements i.c waiter, food, bill, chef etc. The database collects information on customers who gives order for meal. This includes customers name, id and phone number. Customer orders food items. Restaurant has waiter identified by waiter id, name, salary, hire date. Waiter serves food items. Chefs who are identified by name, id, salary etc. are appointed by manager. Manager has name, id, email, phone number etc. in the information system. Cashiers who collects payment from customers, pay in manager. There are different types of food items with food name, food id, price and quantity in this restaurant management system.**

# E-R Diagram



# Normalization

## 1. UNF:

Order (c\_id, c\_name, phone, f\_id, f\_name, price, quantity)

1NF: No multivalued attribute.

2NF: c\_id, c\_name, phone  
f\_id, f\_name, price, quantity

3NF: The data is already in 3NF form.

Table for Order:

1. c\_id, c\_name, phone
2. f\_id, f\_name, price, quantity
3. n\_id, c\_id, f\_id

## 2. UNF:

Payment (c\_id, c\_name, phone, ca\_id, ca\_name, hiredate, salary, phone, address)

1NF: Phone is a multivalued attribute.

2NF: c\_id, c\_name, phone  
Ca\_id, ca\_name, hiredate, salary, phone, address

3NF: The data is already in 3NF form.

### Table for Payment:

1. c\_id, c\_name, phone, address, ca\_id
2. ca\_id, ca\_name, hiredate, salary, phone, address

### **3. UNF:**

Pay in (ca\_id, ca\_name, phone, salary, hiredate, address, m\_id, m\_name, salary, phone, address, hiredate, email)

1NF: Phone is a multivalued attribute.

2NF: ca\_id, ca\_name, phone, salary, hiredate, address

m\_id, m\_name, salary, phone, address, hiredate, email

3NF: The data is already in 3NF form.

### Table for Pay in:

1. ca\_id, ca\_name, phone, salary, address, hiredate, m\_id
2. m\_id, m\_name, salary, phone, address, hiredate, email

### **4. UNF:**

Serve (c\_id, c\_name, phone, w\_id, w\_name, salary, phone, address, hiredate)

1NF: Phone is a multivalued attribute

2NF: c\_id, c\_name, phone

w\_id, w\_name, salary, phone, address, hiredate

3NF: The data is already in 3NF form.

Table for serve:

1. c\_id, c\_name, phone

2. w\_id, w\_name, salary, phone, address, hiredate

3. x\_id, c\_id, w\_id

## **5. UNF:**

Inform ( w\_id, w\_name, salary, phone, address, hiredate,  
ch\_id, ch\_name, salary, phone, address, hiredate)

1NF: Phone is a multivalued attribute.

2NF: w\_id, w\_name, salary, phone, address, hiredate

ch\_id, ch\_name, salary, phone, address, hiredate

3NF: The data is already in 3NF form.

Table for inform:

1. w\_id, w\_name, salary, phone, address, hiredate

2. ch\_id, ch\_name, salary, phone, address, hiredate

3. a\_id, w\_id, ch\_id

## Final Tables:

1. c\_id, c\_name, phone, ca\_id
2. n\_id, c\_id, f\_id
3. ca\_id, ca\_name, phone, salary, address, hiredate, m\_id
4. m\_id, m\_name, salary, phone, address, hiredate, email
5. w\_id, w\_name, salary, phone, address, hiredate
6. x\_id, c\_id, w\_id
7. a\_id, w\_id, ch\_id
8. f\_id, f\_name, price, quantity, m\_id
9. ch\_id, ch\_name, salary, phone, address, hiredate, m\_id

1.

<u>c_id</u>	c_name	phone	<u>ca_id</u>
pk			fk

2.

<u>n_id</u>	<u>c_id</u>	<u>f_id</u>
pk	fk	fk

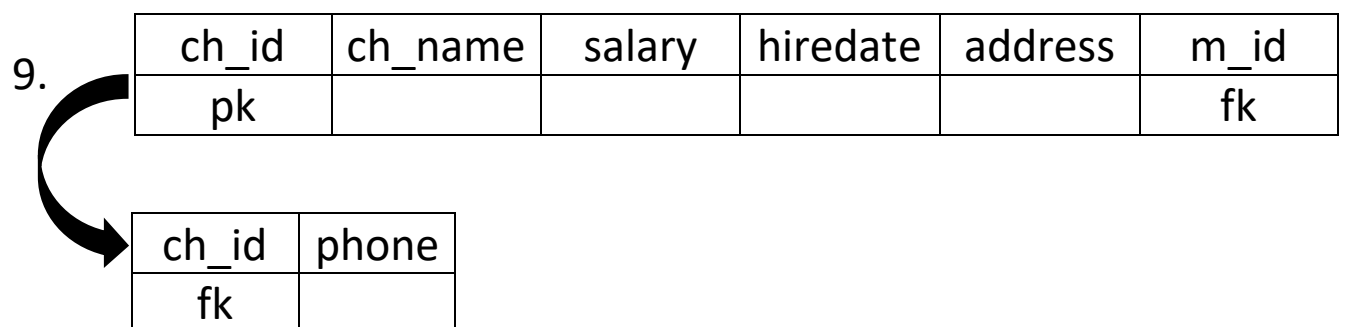
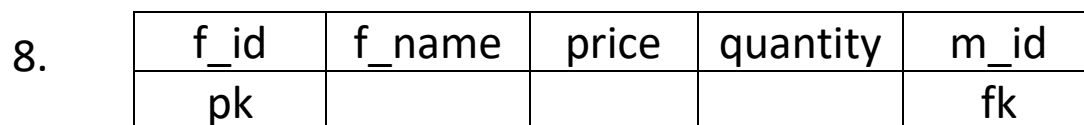
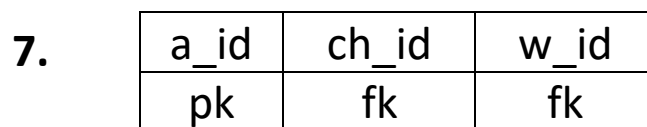
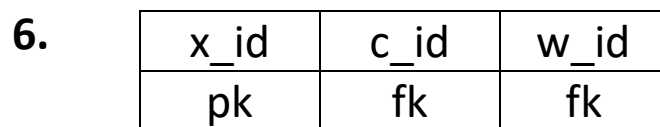
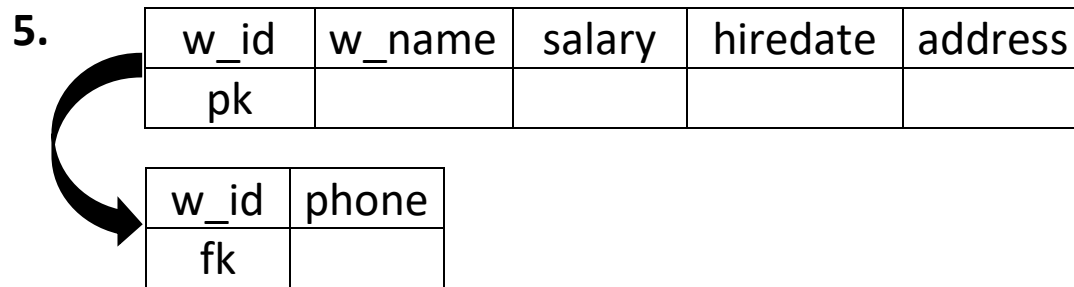
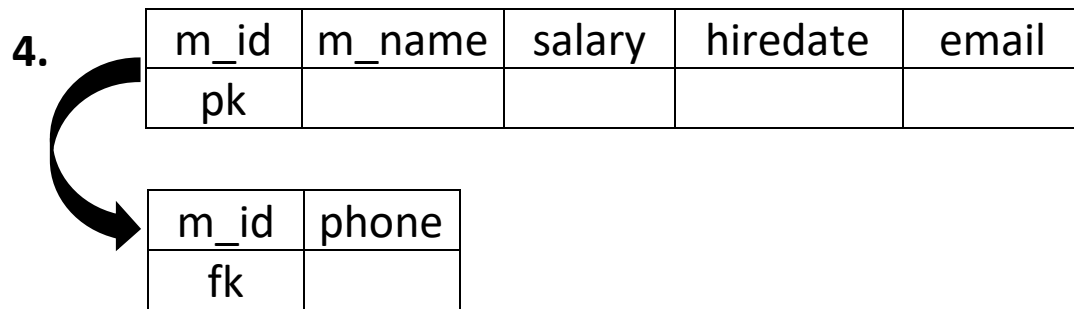
3.

<u>ca_id</u>	ca_name	salary	address	hiredate	<u>m_id</u>
pk					fk

phone	<u>Ca_id</u>
	fk





# Table Creation

## 1. Cashier Table:

```
desc cashier
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **CASHIER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>CASHIER</u>	<u>CA_ID</u>	Number	-	10	0	1	-	-	-
	<u>CA_NAME</u>	Varchar2	20	-	-	-	✓	-	-
	<u>SALARY</u>	Number	-	10	0	-	✓	-	-
	<u>ADDRESS</u>	Varchar2	20	-	-	-	✓	-	-
	<u>HIREDATE</u>	Date	7	-	-	-	✓	-	-
	<u>M_ID</u>	Number	-	10	0	-	✓	-	-
1 - 6									

## 2. Cashier\_phone Table:

```
desc cashier_phone
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **CASHIER\_PHONE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>CASHIER_PHONE</u>	<u>CA_ID</u>	Number	-	10	0	-	✓	-	-
	<u>PHONE</u>	Number	-	20	0	-	✓	-	-
1 - 2									

### 3. Chef Table:

```
desc chef
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Object Type **TABLE** Object **CHEF**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CHEF	CH_ID	Number	-	10	0	1	-	-	-
	CH_NAME	Varchar2	20	-	-	-	✓	-	-
	SALARY	Number	-	10	0	-	✓	-	-
	ADDRESS	Varchar2	20	-	-	-	✓	-	-
	HIREDATE	Date	7	-	-	-	✓	-	-
	M_ID	Number	-	10	0	-	✓	-	-
1 - 6									

### 4. Chef\_Phone Table:

```
desc chef_phone
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Object Type **TABLE** Object **CHEF\_PHONE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CHEF_PHONE	CH_ID	Number	-	10	0	-	✓	-	-
	PHONE	Number	-	20	0	-	✓	-	-
1 - 2									

## 5. Chefwaiter Table:

```
desc chefwaiter
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **CHEFWAITER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>CHEFWAITER</u>	<u>A_ID</u>	Number	-	10	0	1	-	-	-
	<u>CH_ID</u>	Number	-	10	0	-	✓	-	-
	<u>W_ID</u>	Number	-	10	0	-	✓	-	-
									1 - 3

## 6. Customer Table:

```
desc customer
```

Results Explain Describe Saved SQL History

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

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>CUSTOMER</u>	<u>C_ID</u>	Number	-	10	0	1	-	-	-
	<u>C_NAME</u>	Varchar2	20	-	-	-	✓	-	-
	<u>PHONE</u>	Number	-	20	0	-	✓	-	-
	<u>CA_ID</u>	Number	-	10	0	-	✓	-	-
									1 - 4

## 7. Customerfood Table:

```
desc customerfood
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **CUSTOMERFOOD**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERFOOD	N_ID	Number	-	10	0	1	-	-	-
	C_ID	Number	-	10	0	-	✓	-	-
	F_ID	Number	-	10	0	-	✓	-	-
1 - 3									

## 8. Customerwaiter Table:

```
desc customerwaiter
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **CUSTOMERWAITER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERWAITER	X_ID	Number	-	10	0	1	-	-	-
	C_ID	Number	-	10	0	-	✓	-	-
	W_ID	Number	-	10	0	-	✓	-	-
1 - 3									

## 9. Fooditems Table:

```
desc fooditem
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **FOODITEM**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FOODITEM	F_ID	Number	-	10	0	1	-	-	-
	F_NAME	Varchar2	20	-	-	-	✓	-	-
	PRICE	Number	-	10	0	-	✓	-	-
	QUANTITY	Number	-	10	0	-	✓	-	-
	M_ID	Number	-	10	0	-	✓	-	-
1 - 5									

## 10. Manager Table:

```
desc manager
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **MANAGER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGER	M_ID	Number	-	10	0	1	-	-	-
	M_NAME	Varchar2	20	-	-	-	✓	-	-
	SALARY	Number	-	10	0	-	✓	-	-
	ADDRESS	Varchar2	20	-	-	-	✓	-	-
	HIREDATE	Date	7	-	-	-	✓	-	-
	EMAIL	Varchar2	20	-	-	-	✓	-	-
1 - 6									

## 11. Manager\_phone Table:

```
desc manager_phone
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Object Type **TABLE** Object **MANAGER\_PHONE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGER_PHONE	M_ID	Number	-	10	0	-	✓	-	-
	PHONE	Number	-	20	0	-	✓	-	-
1 - 2									

## 12. Waiter Table:

```
desc waiter
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Object Type **TABLE** Object **WAITER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
WAITER	W_ID	Number	-	10	0	1	-	-	-
	W_NAME	Varchar2	20	-	-	-	✓	-	-
	SALARY	Number	-	10	0	-	✓	-	-
	ADDRESS	Varchar2	20	-	-	-	✓	-	-
	HIREDATE	Date	7	-	-	-	✓	-	-
1 - 5									

## 13. Waiter\_phone Table:

```
desc waiter_phone
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **WAITER\_PHONE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
WAITER_PHONE	W_ID	Number	-	10	0	-	✓	-	-
	PHONE	Number	-	20	0	-	✓	-	-
1 - 2									

## Insertion

### 1. Cashier:

```
select * from cashier
```

Results Explain Describe Saved SQL History

CA_ID	CA_NAME	SALARY	ADDRESS	HIREDATE	M_ID
1	Rohim Islam	500	Comilla	08-JAN-04	1

1 rows returned in 0.00 seconds

[CSV Export](#)



## 2. Cashier\_phone:

```
select * from cashier_phone
```

**Results** Explain Describe Saved SQL History

CA_ID	PHONE
1	903379
1	1927463530

2 rows returned in 0.00 seconds

[CSV Export](#)

## 3. Chef:

```
select * from chef
```

**Results** Explain Describe Saved SQL History

CH_ID	CH_NAME	SALARY	ADDRESS	HIREDATE	M_ID
1	Rafa	700	Chittagong	09-JUL-04	1
2	Sworna	700	Mymensingh	12-JUL-04	1
3	Hasan	700	Dhaka	12-MAY-04	1
4	Robi	700	Dhaka	20-MAY-04	1

4 rows returned in 0.00 seconds

[CSV Export](#)

#### 4. Chef\_phone:

```
select * from chef_phone
```

**Results** Explain Describe Sav

CH_ID	PHONE
1	236171808
1	91234657
2	82234657
3	12454657
3	88244657
4	12093617

6 rows returned in 0.00 seconds

#### 5. Chefwaiter:

```
select * from chefwaiter
```

**Results** Explain Describe Save

A_ID	CH_ID	W_ID
1	2	1
2	1	2
3	4	4
4	3	3

4 rows returned in 0.00 seconds

## 6. Customer:

```
select * from customer
```

**Results** Explain Describe Saved SQL History

C_ID	C_NAME	PHONE	CA_ID
1	Nowrin	1712251728	1
2	Shafina	1717351728	1
3	Foysal	1717390256	1
4	Ridwan	1712783645	1

4 rows returned in 0.00 seconds

[CSV Export](#)

## 7. Customerfood:

```
select * from customerfood
```

**Results** Explain Describe Save

N_ID	C_ID	F_ID
1	1	4
3	2	1
2	3	3
4	4	2

4 rows returned in 0.00 seconds

## 8. Customerwaiter:

```
select * from customerwaiter
```

**Results** Explain Describe Save

X_ID	C_ID	W_ID
1	2	1
2	1	2
3	3	4
4	4	3

4 rows returned in 0.00 seconds

## 9. Fooditems:

```
select * from fooditem
```

**Results** Explain Describe Saved SQL History

F_ID	F_NAME	PRICE	QUANTITY	M_ID
1	Pasta	200	1	1
2	Pizza	800	3	1
3	Coffe	80	1	1
4	Rice	100	1	1
5	Frech fries	100	1	1

5 rows returned in 0.00 seconds

[CSV Export](#)

## 10. Manager:

```
select * from manager|
```

**Results** Explain Describe Saved SQL History

M_ID	M_NAME	SALARY	ADDRESS	HIREDATE	EMAIL
1	Korim Hasan	800	Dhaka	11-FEB-00	kh@gmail.com

1 rows returned in 0.00 seconds

[CSV Export](#)

## 11. Manager\_phone:

```
select * from manager_phone|
```

**Results** Explain Describe Saved SQL History

M_ID	PHONE
1	1873927464
1	1947927464

2 rows returned in 0.00 seconds

[CSV Export](#)

## 12. Waiter:

```
select * from waiter
```

**Results** Explain Describe Saved SQL History

W_ID	W_NAME	SALARY	ADDRESS	HIREDATE
1	Jamal	300	Sunamgonj	02-JAN-02
2	Kazi	300	Naraongonj	22-JAN-03
3	Wazi	300	Dhaka	22-MAY-03
4	Pulok	300	Dhaka	22-JUN-03

4 rows returned in 0.00 seconds

[CSV Export](#)

## 13. Waiter\_phone:

```
select * from waiter_phone
```

**Results** Explain Describe Saved

W_ID	PHONE
1	1902718623
3	11267389
1	926178
2	1267389
4	135699008

5 rows returned in 0.00 seconds

# Constraints

1.

```
alter table Customer add constraint con1 primary key(c_id)
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

Table altered.

0.03 seconds

2.

```
alter table Cashier add constraint con2 primary key(ca_id)
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

Table altered.

0.03 seconds

3.

```
alter table Customer add constraint con3 foreign key(ca_id) reference Cashier (ca_id)
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

Table altered.

0.03 seconds

4.

```
alter table Cashier_phone add constraint con4 foreign key(ca_id) references Cashier (ca_id)
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

Table altered.

0.03 seconds



5.

```
alter table Manager add constraint con5 primary key(m_id)
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

Table altered.

0.03 seconds

6.

```
alter table Manager_phone add constraint con6 foreign key(m_id) references Manager (m_id)
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

Table altered.

0.03 seconds

7.

```
alter table Fooditem add constraint con7 primary key(f_id)
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

Table altered.

0.03 seconds

8.

```
alter table Chef add constraint con9 primary key(ch_id)
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

Table altered.

0.03 seconds

9.

```
alter table Chef add constraint con9 primary key(ch_id)
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

Table altered.

0.03 seconds

10.

```
alter table chef add constraint con10 foreign key(m_id) references Manager (m_id)
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

Table altered.

0.03 seconds

11.

```
alter table Chef_phone add constraint con11 foreign key(ch_id) references Chef (ch_id)
```

---

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

---

Table altered.

0.03 seconds

12.

```
alter table Waiter add constraint con12 primary key(w_id)
```

---

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

---

Table altered.

0.03 seconds

13.

```
alter table Waiter_phone add constraint con13 foreign key(w_id) references Waiter (w_id)
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

Table altered.

0.03 seconds

14.

```
alter table Customerfood add constraint con14 primary key(n_id)
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

Table altered.

0.03 seconds

15.

```
alter table Customerfood add constraint con15 foreign key(f_id) references Fooditem (f_id)
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

Table altered.

0.03 seconds

16.

```
alter table Customerfood add constraint con16 foreign key(f_id) references Fooditem (f_id)
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

Table altered.

0.03 seconds

17.

```
alter table Customerwaiter add constraint con17 primary key(x_id)
```

**Results** Explain Describe Saved SQL History

Table altered.

0.03 seconds

## Query Question

### Group function:

1. Display the food name, food number and the price of all food which price is equal to the minimum price.
2. Display Chef id and average salary who was hired in the date of 12.

### Subquery:

1. Display the employee's details who has joined after HASAN.
2. Display waiter salary and hire date for all waiter in the same address as JAMAL.

### Join:

1. Display customer name and waiter id using equijoin.
2. Display f\_name, m\_name where food item's price is between 80 and 100.

### View:

1. Create a view called waiter\_ve based on the waiter id and salary from waiter table.
2. Create a view called customer\_vu based on customer id and phone number from customer table with no update.

## Conclusion

After finishing the project, the basic concept of relational database management system is quite clear to us. Now each of the group member can try by their own self to make this kind of various management system in their future.