



**Project Name: Coffee Shop Management System**

<u>Name</u>	<u>ID</u>
Talukder, Noor Mohammad	19-40649-1
Nibras, Shamit	19-40117-1
Marfy, Ariful, Islam	19-40116-1
Islam, Md, Naymul	19-40125-1
Islam, Mohammad Maruful	19-40135-1
Jesey, Dilruba Khanam	19-39979-1

**Course Name: INTRODUCTION TO DATABASE**

**Section: H**

# **CONTENTS**

Sl.	Topic	Page
a.	<b>Introduction</b> Coffee Shop Management System Desktop-based application developed using SQL DATABASE SYSTEM.	1
b.	<b>Scenario Description</b> Address is composed of house no street name and city While customer placed order date and customer id is stored.	1
c.	<b>ER Diagram</b> With proper annotations using DIA	1-2
d.	<b>Normalization</b> Organizing Data Elements	2-6
e.	<b>Schema Diagram</b>	6
f.	<b>Table Creation</b> <ul style="list-style-type: none"> <li>a. Users, assign roles and grant privileges</li> <li>b. Constraints</li> <li>c. Sequence of the queries</li> <li>d. Screenshots of the queries</li> </ul>	6-10
g.	<b>Data Insertion</b> <ul style="list-style-type: none"> <li>a. Insert the data</li> <li>b. Screenshots of the tables after inserting data</li> </ul>	10-15
h.	<b>Query Writing</b> <ul style="list-style-type: none"> <li>a. Subquery</li> <li>b. Joining</li> <li>c. View</li> </ul>	15-18
i.	<b>Relational Algebra</b>	18
j.	<b>Conclusion</b> Summarizing & future plan of the project	18-19

## a. INTRODUCTION:

Our Coffee Shop Management System is based on a concept to maintain orders and management of a particular coffee shop. This project is mainly divided into two sections in this project, they are Order Placing and Admin panel. By using this system, one can maintain ordering records of a day. By selecting Menu Order the system displays a list of Available items and the user has to place an order with item quantity. The role of the administrator is to maintain employee information including operations like modifying, deleting, updating the employee records and customer order records in the system.

By using SQL the database and required query of Coffee Shop Management System is written. Our admin panel is responsible for receiving all processing orders from customers. We also provide delivery to customer in less time. It is an attractive option for many customers. It is convenient and cost effective for our customers.

## b. Scenario Description

In a coffee management a customer may be served by one coffee shop but one coffee shop may serve to many customers. A customer is identified by customer id. The system also stores customer name, phone number. A coffee shop is identified with shop id. Shop name and address is also stored. Address is composed of house number, street name and city.

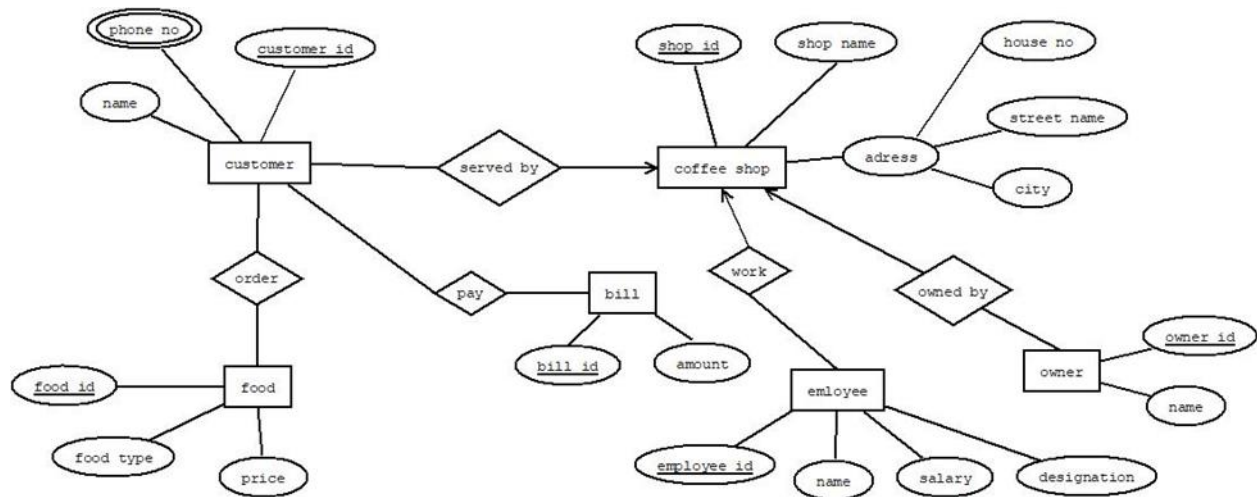
A customer can also order many foods and a food may be ordered by many customers. Food is identified with food id. In this process food type and price also stored.

Bill may be paid by many customers and a customer can pay many bills. Bill is identified by biller id. Amount of bill is also stored in the process.

A coffee shop may be owned many owners and a owner may owned a coffee shop. Owner is identified with owner id and owner's name also stored in this process.

Many employees can work in one coffee shop and in one coffee shop many employees can work. Employee is identified with employee id. Name, salary, designation also stored in this process.

## c. ER Diagram:



## d. Normalization:

### Served by

#### UNF

Served by (customer\_id, customer\_name, phone no, shop\_id, shop\_name, house\_no, street\_name, city)

#### 1NF

phone no is a multi-value attribute

1. customer\_id, customer\_name, phone no, shop\_id, shop\_name, house\_no, street\_name, city

#### 2NF

1. customer\_id, customer\_name, phone no
2. shop\_id, shop\_name, house\_no, street\_name, city

#### 3NF

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

1. customer\_id, customer\_name, phone no
2. shop\_id, shop\_name, house\_no, street\_name, city

## Table Creation

1. customer\_id, customer\_name, phone no, **shop\_id**
2. shop\_id, shop\_name, house\_no, street\_name, city

## Order

### UNF

Order (food\_id, food\_type, price, customer\_id, customer\_name, phone no)

### 1NF

phone no is a multi-value attribute

1. Food\_id, food\_type, price, customer\_id, customer\_name, phone no

### 2NF

1. food\_id, food\_type, price
2. customer\_id, customer\_name, phone no

### 3NF

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

1. food\_id, food\_type, price
2. customer\_id, customer\_name, phone no

## Table Creation

1. food\_id, food\_type, price
2. customer\_id, customer\_name, phone no
3. **food\_id, customer\_id**

## Pay

### UNF

Pay (bill\_id, amount, customer\_id, customer\_name, phone no)

### 1NF

phone no is a multi-value attribute

bill\_id, amount, customer\_id, customer\_name, phone no

## 2NF

1. bill\_id, amount
2. customer\_id, customer\_name, phone no

## 3NF

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

1. bill\_id, amount
2. customer\_id, customer\_name, phone no

## Table Creation

1. bill\_id, amount
2. customer\_id, customer\_name, phone no
3. bill\_id, customer\_id

## work

## UNF

work (shop\_id, shop\_name, house\_no, street\_name, city, employee\_id, employee\_name, salary, designation)

## 1NF

There is no multi valued attribute.

1. shop\_id, shop\_name, house\_no, street\_name, city, employee\_id, employee\_name, salary, designation

## 2NF

1. shop\_id, shop\_name, house\_no, street\_name, city
2. employee\_id, employee\_name, salary, designation

## 3NF

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

1. shop\_id, shop\_name, house\_no, street\_name, city
2. employee\_id, employee\_name, salary, designation

## TABLE CREATION

1. shop\_id, shop\_name, house\_no, street\_name, city
2. employee\_id, employee\_name, salary, designation, **shop\_id**

## Owned by

### UNF

Owned by (shop\_id, shop\_name, house\_no, street\_name, city, owner\_id, owner\_name)

### 1NF

There is no multi valued attribute.

1. shop\_id, shop\_name, house\_no, street\_name, city, owner\_id, owner\_name

### 2NF

1. shop\_id, shop\_name, house\_no, street\_name, city
2. owner\_id, owner\_name

### 3NF

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

1. shop\_id, shop\_name, house\_no, street\_name, city
2. owner\_id, owner\_name

## TABLE CREATION

1. shop\_id, shop\_name, house\_no, street\_name, city
2. owner\_id, owner\_name, **shop\_id**

## Temporally Table

1. customer\_id, customer\_name, phone no, **shop\_id**
2. shop\_id, shop\_name, house\_no, street\_name, city
3. food\_id, food\_type, price
4. ~~customer\_id, customer\_name, phone no~~
5. **food\_id, customer\_id**
6. bill\_id, amount
7. ~~customer\_id, customer\_name, phone no~~

8. bill\_id, customer\_id

9. ~~shop\_id, shop\_name, house\_no, street\_name, city~~

10. employee\_id, employee\_name, salary, designation, **shop\_id**

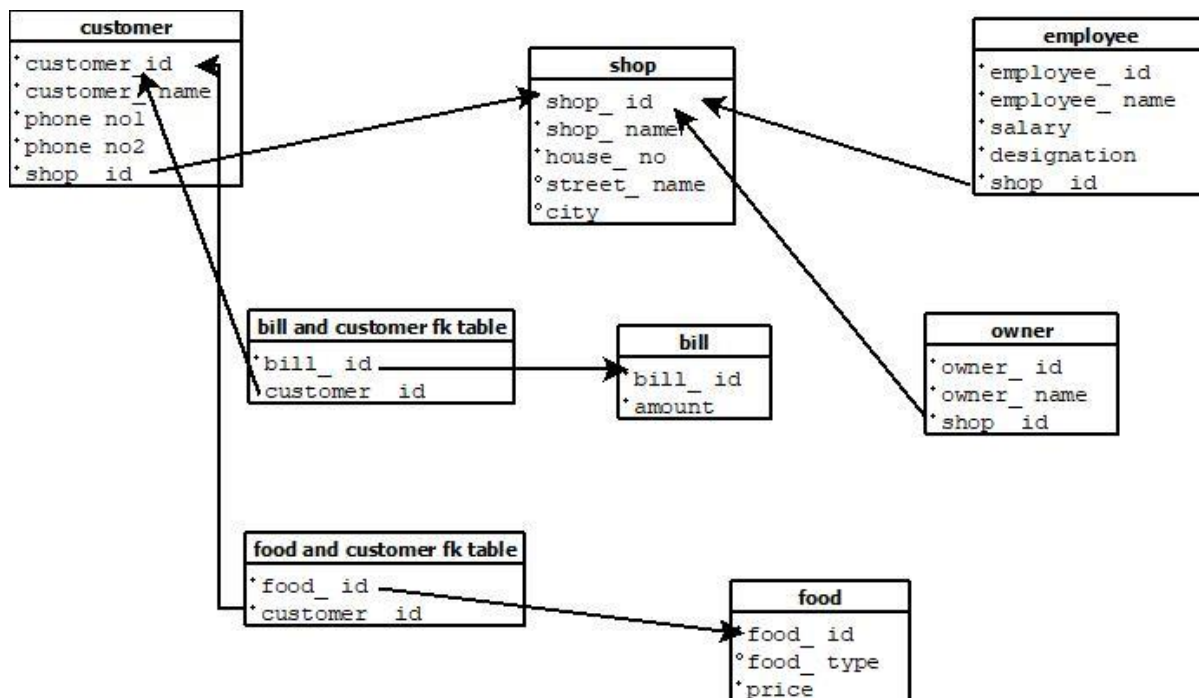
11. ~~shop\_id, shop\_name, house\_no, street\_name, city~~

12. owner\_id, owner\_name, **shop\_id**

Final Table

1. customer\_id, customer\_name, phone no1, phone no2, phone no3, ..... phone no n  
**shop\_id**
2. shop\_id, shop\_name, house\_no, street\_name, city
3. food\_id, food\_type, price
4. food\_id, customer\_id
5. bill\_id, amount
6. bill\_id, customer\_id
7. employee\_id, employee\_name, salary, designation, **shop\_id**
8. owner\_id, owner\_name, **shop\_id**
- 9.

e. Schema Diagram:



f. Table Creation

➤ CREATE USER Isopropyl IDENTIFIED BY net23xyz;



- GRANT UNLIMITED TABLESPACE TO Isopropyl;
- CREATE ROLE Accounts;
- GRANT create table, create view, create sequence to Accounts;
- GRANT Accounts to Isopropyl;
- ALTER USER Isopropyl IDENTIFIED BY net345abc;

### 1. For Shop Table

create table shop (shop\_id number (3) primary key,shop\_name varchar2(50) not null,house\_no varchar2(50) not null,street\_name varchar2(50) not null,city varchar2(50) not null);

Describe shop;

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **SHOP**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SHOP	SHOP_ID	Number	-	3	0	1	-	-	-
	SHOP_NAME	Varchar2	50	-	-	-	-	-	-
	HOUSE_NO	Varchar2	50	-	-	-	-	-	-
	STREET_NAME	Varchar2	50	-	-	-	-	-	-
	CITY	Varchar2	50	-	-	-	-	-	-

1 - 5

### 2. For Customer Table

create table customer (customer\_id number(3) primary key,customer\_name varchar2(50) not null,phone\_no1 number(15) not null,phone\_no2 number(15) not null,shop\_id number(3) not null,foreign key(shop\_id) references shop(shop\_id));

Describe customer;

Results Explain Describe Saved SQL History

Object Type TABLE Object CUSTOMER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CUSTOMER_ID	Number	-	3	0	1	-	-	-
	CUSTOMER_NAME	Varchar2	50	-	-	-	-	-	-
	PHONE_NO1	Number	-	15	0	-	-	-	-
	PHONE_NO2	Number	-	15	0	-	-	-	-
	SHOP_ID	Number	-	3	0	-	-	-	-

1 - 5

### 3. For Employee Table

create table employee (employee\_id number (3) primary key,employee\_name varchar2(50) not null,salary number(20) not null,designation varchar2(50) not null,shop\_id number(3) not null,foreign key(shop\_id) references shop(shop\_id));

Describe employee;

Results Explain Describe Saved SQL History

Object Type TABLE Object EMPLOYEE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE	EMPLOYEE_ID	Number	-	3	0	1	-	-	-
	EMPLOYEE_NAME	Varchar2	50	-	-	-	-	-	-
	SALARY	Number	-	20	0	-	-	-	-
	DESIGNATION	Varchar2	50	-	-	-	-	-	-
	SHOP_ID	Number	-	3	0	-	-	-	-
1 - 5									

Application Express 2.1.0.00.39

#### 4. For Owner Table

create table owner (owner\_id number (3) primary key,owner\_name varchar2(50) not null,shop\_id number(3) not null,foreign key(shop\_id) references shop(shop\_id));

Describe owner;

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **OWNER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>OWNER</u>	<u>OWNER_ID</u>	Number	-	3	0	1	-	-	-
	<u>OWNER_NAME</u>	Varchar2	50	-	-	-	-	-	-
	<u>SHOP_ID</u>	Number	-	3	0	-	-	-	-

1 - 3

#### 5. For Bill Table

create table bill (bill\_id number (3) primary key,amount varchar2(50) not null);

Describe bill;

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **BILL**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<a href="#">BILL</a>	<a href="#">BILL_ID</a>	Number	-	3	0	1	-	-	-
	<a href="#">AMOUNT</a>	Varchar2	50	-	-	-	-	-	-

1 - 2

#### 6. For Food Table

create table food (food\_id number (3) primary key,food\_type varchar2(50) unique,price varchar2(50) not null);

Describe food;

Results Explain Describe Saved SQL History

Object Type TABLE Object FOOD

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FOOD	FOOD_ID	Number	-	3	0	1	-	-	-
	FOOD_TYPE	Varchar2	50	-	-	-	✓	-	-
	PRICE	Varchar2	50	-	-	-	-	-	-

1 - 3

## 7. For Food and customer fk Table

Create table food\_and\_customer\_fk\_table(food\_id number(3)not null,foreign key(food\_id)references food(food\_id),customer\_id number(3) not null,foreign key(customer\_id) references customer(customer\_id),constraint t\_pk primary key(food\_id,customer\_id));

Describe food\_and\_customer\_fk\_table;

Results Explain Describe Saved SQL History

Object Type TABLE Object FOOD\_AND\_CUSTOMER\_FK\_TABLE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FOOD_AND_CUSTOMER_FK_TABLE	FOOD_ID	Number	-	3	0	1	-	-	-
	CUSTOMER_ID	Number	-	3	0	2	-	-	-

1 - 2

## 8. For bill and customer fk Table

create table bill\_and\_customer\_fk\_table ( bill\_id number(3) not null,foreign key(bill\_id)references bill(bill\_id),customer\_id number(3) not null,foreign key(customer\_id) references customer (customer\_id) ,constraint t\_pk1 primary key (bill\_id,customer\_id));

Describe bill\_and\_customer\_fk\_table;

Results Explain Describe Saved SQL History

Object Type TABLE Object BILL\_AND\_CUSTOMER\_FK\_TABLE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BILL AND CUSTOMER FK TABLE	BILL_ID	Number	-	3	0	1	-	-	-
	CUSTOMER_ID	Number	-	3	0	2	-	-	-
1 - 2									

## Sequence

- create sequence shop\_id\_seq increment by 1 start with 100 maxvalue 999;
- create sequence customer\_id\_seq increment by 1 start with 200 maxvalue 999;
- create sequence employee\_id\_seq increment by 1 start with 300 maxvalue 999;

- create sequence owner\_id\_seq increment by 1 start with 400 maxvalue 999;
- create sequence bill\_id\_seq increment by 1 start with 500 maxvalue 999;
- create sequence food\_id\_seq increment by 1 start with 600 maxvalue 999;

## g. Data Insertion

### 1. For Shop Table

- insert into shop (shop\_id,shop\_name ,house\_no,street\_name,city)values (shop\_id\_seq.NEXTVAL,'ABC','55/B','Baily Road','DHAKA');
- insert into shop (shop\_id,shop\_name ,house\_no,street\_name,city)values (shop\_id\_seq.NEXTVAL,'DEF','64/A','Shohid Bir Uttam Road','DHAKA');
- insert into shop (shop\_id,shop\_name ,house\_no,street\_name,city)values (shop\_id\_seq.NEXTVAL,'XYZ','191/A','Abdullah Khan Road','CHITTAGONG');
- insert into shop (shop\_id,shop\_name ,house\_no,street\_name,city)values (shop\_id\_seq.NEXTVAL,'MNO','109/L','Abu Mia Road','PABNA');
- insert into shop (shop\_id,shop\_name ,house\_no,street\_name,city)values (shop\_id\_seq.NEXTVAL,'JKL','76/B','Merajnagar Road','DHAKA');

SELECT \*FROM SHOP;

SHOP_ID	SHOP_NAME	HOUSE_NO	STREET_NAME	CITY
100	ABC	55/B	Baily Road	DHAKA
101	DEF	64/A	Shohid Bir Uttam Road	DHAKA
102	XYZ	191/A	Abdullah Khan Road	CHITTAGONG
103	MNO	109/L	Abu Mia Road	PABNA
104	JKL	76/B	Merajnagar Road	DHAKA

### 2. For Customer Table

- INSERT INTO customer(customer\_id, customer\_name,phone\_no1,phone\_no2,shop\_id) values(customer\_id\_seq.NEXTVAL,'RAJ','8801711318734','8801552392801','100');
- INSERT INTO customer(customer\_id,customer\_name,phone\_no1,phone\_no2,shop\_id) values(customer\_id\_seq.NEXTVAL,'SAMIUN','8801723210623','88019107499455','101');
- INSERT INTO customer(customer\_id,customer\_name,phone\_no1,phone\_no2,shop\_id) values(customer\_id\_seq.NEXTVAL,'BADOL','8801847265122','8801313054598','103');
- INSERT INTO customer(customer\_id,customer\_name,phone\_no1,phone\_no2,shop\_id) values(customer\_id\_seq.NEXTVAL,'SIMRAN','8801552349494','8801722629237','104');

- INSERT INTO customer(customer\_id,customer\_name,phone\_no1,phone\_no2,shop\_id) values(customer\_id\_seq.NEXTVAL,'RIYA','8801747440554','8801955997675','102');  
SELECT \*FROM CUSTOMER;

CUSTOMER_ID	CUSTOMER_NAME	PHONE_NO1	PHONE_NO2	SHOP_ID
200	RAJ	8801711318734	8801552392801	100
201	SAMIUN	8801723210623	88019107499455	101
202	BADOL	8801847265122	8801313054598	103
203	SIMRAN	8801552349494	8801722629237	104
204	RIYA	8801747440554	8801955997675	102

### 3. For Employee Table

- insert into employee (employee\_id, employee\_name, salary, designation, shop\_id) values (employee\_id\_seq.NEXTVAL, 'SIMRAN', 5000, 'CHEF', '101');
- insert into employee (employee\_id, employee\_name, salary, designation, shop\_id) values (employee\_id\_seq.NEXTVAL, 'RANA', 6500, 'MANAGER', '101');
- insert into employee (employee\_id, employee\_name, salary, designation, shop\_id) values (employee\_id\_seq. NEXTVAL, 'RONY', 3000, 'WAITER', '101');
- insert into employee (employee\_id, employee\_name, salary, designation, shop\_id) values (employee\_id\_seq. NEXTVAL, 'RAKHI', 6500, 'MANAGER', '103');
- insert into employee (employee\_id, employee\_name, salary, designation, shop\_id) values (employee\_id\_seq. NEXTVAL, 'JONY', 3000, 'WAITER', '103');
- insert into employee (employee\_id, employee\_name, salary, designation, shop\_id) values (employee\_id\_seq .NEXTVAL, 'KAMRUL', 5000, 'CHEF', '103');
- insert into employee (employee\_id, employee\_name, salary, designation, shop\_id) values (employee\_id\_seq.NEXTVAL,'BADON',2500,'WAITER','102');
- insert into employee (employee\_id,employee\_name,salary,designation,shop\_id)values (employee\_id\_seq.NEXTVAL,'ATIK',4000,'CHEF','102');
- insert into employee (employee\_id,employee\_name,salary,designation,shop\_id)values (employee\_id\_seq.NEXTVAL,'ALVI',6000,'MANAGER','102');
- insert into employee (employee\_id,employee\_name,salary,designation,shop\_id)values (employee\_id\_seq.NEXTVAL,'MINHAZ',3000,'WAITER','104');
- insert into employee (employee\_id,employee\_name,salary,designation,shop\_id)values (employee\_id\_seq.NEXTVAL,'BORNA',5500,'CHEF','104');
- insert into employee (employee\_id,employee\_name,salary,designation,shop\_id)values (employee\_id\_seq.NEXTVAL,'SHAKIL',6000,'MANAGER','100');
- insert into employee (employee\_id,employee\_name,salary,designation,shop\_id)values (employee\_id\_seq.NEXTVAL,'RONY',3000,'WAITER','100');
- insert into employee (employee\_id,employee\_name,salary,designation,shop\_id)values

(employee\_id\_seq. NEXTVAL,'FATEMA',5000,'CHEF','100');

- insert into employee (employee\_id,employee\_name,salary,designation,shop\_id)values (employee\_id\_seq. NEXTVAL,'SHIRAJ',3500,'CASHIER','101');
- insert into employee (employee\_id,employee\_name,salary,designation,shop\_id)values (employee\_id\_seq. NEXTVAL,'SAKIB',3500,'CASHIER','102');
- insert into employee (employee\_id,employee\_name,salary,designation,shop\_id)values (employee\_id\_seq. NEXTVAL,'RAISA',3500,'CASHIER','104');
- insert into employee (employee\_id, employee\_name, salary, designation, shop\_id) values (employee\_id\_seq. NEXTVAL,'ANIKA',3500,'CASHIER','100');

SELECT \*FROM EMPLOYEE;

EMPLOYEE_ID	EMPLOYEE_NAME	SALARY	DESIGNATION	SHOP_ID
300	SIMRAN	5000	CHEF	101
301	RANA	6500	MANAGER	101
302	RONY	3000	WAITER	101
303	RAKHI	6500	MANAGER	103
304	JONY	3000	WAITER	103
305	KAMRUL	5000	CHEF	103
306	BADON	2500	WAITER	102
307	ATIK	4000	CHEF	102
308	ALVI	6000	MANAGER	102
309	MINHAZ	3000	WAITER	104
310	BORNA	5500	CHEF	104
311	SHAKIL	6000	MANAGER	100
312	RONY	3000	WAITER	100
313	FATEMA	5000	CHEF	100
314	SHIRAJ	3500	CASHIER	101
315	SAKIB	3500	CASHIER	102
316	RAISA	3500	CASHIER	104
317	ANIKA	3500	CASHIER	100

#### 4. For Owner Table

- insert into owner (owner\_id, owner\_name, shop\_id) values (owner\_id\_seq.NEXTVAL, 'ALAM',100 );
- insert into owner (owner\_id, owner\_name, shop\_id) values (owner\_id\_seq.NEXTVAL, 'RAGIB',104 );

- insert into owner (owner\_id, owner\_name, shop\_id) values (owner\_id\_seq.NEXTVAL, 'TOMA',102 );
- insert into owner (owner\_id, owner\_name, shop\_id) values (owner\_id\_seq.NEXTVAL, 'NISHAN',103 );
- insert into owner (owner\_id, owner\_name, shop\_id) values (owner\_id\_seq.NEXTVAL, 'TANJILA',101 );

SELECT \*FROM OWNER;

OWNER_ID	OWNER_NAME	SHOP_ID
400	ALAM	100
401	RAGIB	104
402	TOMA	102
403	NISHAN	103
404	TANJILA	101

## 5. For Bill Table

- INSERT INTO bill (bill\_id,amount)VALUES(bill\_id\_seq.NEXTVAL,'450');
- INSERT INTO bill (bill\_id,amount)VALUES(bill\_id\_seq.NEXTVAL,'250');
- INSERT INTO bill (bill\_id,amount)VALUES(bill\_id\_seq.NEXTVAL,'350');
- INSERT INTO bill (bill\_id,amount)VALUES(bill\_id\_seq.NEXTVAL,'200');
- INSERT INTO bill (bill\_id,amount)VALUES(bill\_id\_seq.NEXTVAL,'90');

SELECT \*FROM BILL;

BILL_ID	AMOUNT
500	450
501	250
502	350
503	200
504	90

## 6. For Food Table

- INSERT INTO food (food\_id, food\_type, price) VALUES (food\_id\_seq.NEXTVAL, 'PASTRY', '300');
- INSERT INTO food (food\_id,food\_type,price) VALUES (food\_id\_seq.NEXTVAL, 'AFFOGATO','350');
- INSERT INTO food (food\_id, food\_type, price) VALUES (food\_id\_seq.NEXTVAL, 'ICED LATTE', '150');



- INSERT INTO food (food\_id, food\_type, price) VALUES (food\_id\_seq.NEXTVAL, 'CAFFÈ MOCHA', '550');
- INSERT INTO food (food\_id, food\_type, price) VALUES (food\_id\_seq.NEXTVAL, 'CAPPUCCINO', '200');
- INSERT INTO food (food\_id, food\_type, price) VALUES (food\_id\_seq.NEXTVAL, 'SANDWICH', '50');
- INSERT INTO food (food\_id, food\_type, price) VALUES (food\_id\_seq.NEXTVAL, 'COOKIES', '90');

SELECT \*FROM FOOD;

FOOD_ID	FOOD_TYPE	PRICE
600	PASTRY	300
601	AFFOGATO	350
602	ICED LATTE	150
603	CAFFÈ MOCHA	550
604	CAPPUCCINO	200
605	SANDWICH	50
606	COOKIES	90

## 7. For Food and customer fk Table

- INSERT INTO food\_and\_customer\_fk\_table(food\_id ,customer\_id)VALUES('600','201');
- INSERT INTO food\_and\_customer\_fk\_table(food\_id ,customer\_id)VALUES('604','204');
- INSERT INTO food\_and\_customer\_fk\_table(food\_id ,customer\_id)VALUES('602','201');
- INSERT INTO food\_and\_customer\_fk\_table(food\_id ,customer\_id)VALUES('605','204');
- INSERT INTO food\_and\_customer\_fk\_table(food\_id ,customer\_id)VALUES('601','203');
- INSERT INTO food\_and\_customer\_fk\_table(food\_id ,customer\_id)VALUES('606','200');
- INSERT INTO food\_and\_customer\_fk\_table(food\_id ,customer\_id)VALUES('604','202');

SELECT \*FROM food\_and\_customer\_fk\_table;

FOOD_ID	CUSTOMER_ID
600	201
604	204
602	201
605	204
601	203
606	200
604	202

## 8. For Bill and customer fk Table

- INSERT INTO bill\_and\_customer\_fk\_table(bill\_id ,customer\_id)VALUES('500','201');
- INSERT INTO bill\_and\_customer\_fk\_table(bill\_id ,customer\_id)VALUES('501','204');
- INSERT INTO bill\_and\_customer\_fk\_table(bill\_id ,customer\_id)VALUES('503','202');
- INSERT INTO bill\_and\_customer\_fk\_table(bill\_id ,customer\_id)VALUES('504','203');



- INSERT INTO bill\_and\_customer\_fk\_table(bill\_id ,customer\_id)VALUES('502','200');  
SELECT \*FROM bill\_and\_customer\_fk\_table;

BILL_ID	CUSTOMER_ID
500	201
501	204
503	202
504	203
502	200

## h. Query Writing

### 1. Sub Query

**Question:** Display the employee names who earn more than employee SIMRAN.

**ANSWER:** select employee\_name from employee where salary>(select salary from employee where employee\_name='SIMRAN');

EMPLOYEE_NAME
RANA
RAKHI
ALVI
BORNA
SHAKIL

5 rows returned in 0.00 seconds

**Question:** Display the employee names who have the same designation as employee SHIRAJ.

**ANSWER:** select employee\_name from employee where designation=(select designation from employee where employee\_name='SHIRAJ');

EMPLOYEE_NAME
SHIRAJ
SAKIB
RAISA
ANIKA

### 2. Joining

**Question:** Display the name of all the customers who eat in shop ABC.

**ANSWER:** SELECT c.customer\_name FROM customer c, shop s WHERE c.shop\_id=s.shop\_id and shop\_name='ABC';

CUSTOMER_NAME
RAJ

**Question:** Display the name of all the employees who work in shop XYZ.

**ANSWER:** SELECT e.employee\_name FROM employee e, shop s WHERE e.shop\_id=s.shop\_id and shop\_name='XYZ';

EMPLOYEE_NAME
BADON
ATIK
ALVI
SAKIB

### 3. View

**Question:** Create a view called **EMPView** based on the Employee\_Name and Salary from the Employee table.

**ANSWER:** CREATE VIEW EMPView AS SELECT Employee\_Name , salary FROM Employee;

select \* from EMPView;

EMPLOYEE_NAME	SALARY
SIMRAN	5000
RANA	6500
RONY	3000
RAKHI	6500
JONY	3000
KAMRUL	5000
BADON	2500
ATIK	4000
ALVI	6000
MINHAZ	3000
BORNA	5500
SHAKIL	6000
RONY	3000
FATEMA	5000
SHIRAJ	3500
SAKIB	3500
RAISA	3500
ANIKA	3500

**Question:** Create a view called **OWNERView** based on the OWNER\_Name from the OWNER table.

**ANSWER:** CREATE VIEW OWNERView AS SELECT OWNER\_NAME FROM OWNER;

select \* from OWNERView;

OWNER_NAME
ALAM
RAGIB
TOMA
NISHAN
TANJILA

## i. Relational Algebra

Q1. Find the employee name where salary is greater than 3500.

Answer:  $\Pi_{EMPLOYEE\_NAME} (\sigma_{SALARY > 3500} (EMPLOYEE))$

Q2. Find the shop name where shop location is in Dhaka.

Answer:  $\Pi_{SHOP\_NAME} (\sigma_{CITY="DHAKA"} (SHOP))$

Q3. Find the customer name where customer id is 200.

Answer:  $\Pi_{CUSTOMER\_NAME} (\sigma_{CUSTOMER\_ID=200} (CUSTOMER))$

Q4. Find the employee name where designation is manager.

Answer:  $\Pi_{EMPLOYEE\_NAME} (\sigma_{DESIGNATION="MANAGER"} (EMPLOYEE))$

Q5. Find the owner name where owner id is 400.

Answer:  $\Pi_{OWNER\_NAME} (\sigma_{OWNER\_ID=400} (OWNER))$

## j. Conclusion

Our project topic is Coffee Shop Management System. Our Coffee Shop Management System is based on a concept to maintain orders and management of a particular coffee shop. By selecting Menu Order the system displays a list of Available items and the user has to place an order with item quantity. Our admin panel is responsible for receiving all processing orders

from customers. In our system shop name and address is stored. We also stored employee name, salary, designation and also stored customer name and phone numbers.

At first, we have drawn ER diagram using DIA. Then we normalize the diagram for final table. After normalization we have drawn schema diagram to find primary key and foreign key before table creation. We have created a user's name Isopropyl identified by net123xyz. We have created and grant role accounts to Isopropyl. Grant create table, create view and create sequence to accounts. Alter user Isopropyl identified by net345abc. We have created table following our final table which we have get from normalization and insert necessary data into table and also insert screenshot of all of these.

In query writing at first, we have done sub-query for employee table then we have done joining in customer table and employee table. After that we have assign view in employee table and owner table. Then we have done necessary relational algebra. We have done query writing and relational algebra by creating some question and by answering them.

In future we will invent an app using java, java script and c#. It will be very helpful for our future business plan. By using the app owners/ users can actuate any update in the system. We can make sure best security for the system.