

Project Name:

Course Name: Introduction to Database

Course Instructor: Juena Ahmed Noshin

Section: K

Group Members

SL NO	NAME	ID
1	ASHRAF,MOHAMMED	17-34496-2
2	JOBAYER, MD. ABDULLAH AL	18-37647-1
3		
4		

CONTENT LIST

Introduction

Scenario

ER Diagram

Normalization

Schema Diagram

Table Creation

Data Insertion

Query Writing

Conclusion

INTRODUCTION

Database Management system (DBMS) is a collection of programs for managing data and simultaneously it support different type of users to create, manage, retrieve, update and store information. The vital functions of the database are that it not only manages database engine which is used to access the data but also the database schema which is used to define the logical structure of a database.

We used the concept of DBMS in our project (Ice-cream Parlor Management System).

SCENARIO:

Ice cream parlor is a place where various kinds of ice cream can be found. The customer can choose from various flavor that has been offered and place their order to the staff. The staff then makes the product and deliver it to the customer. One staff can serve multiple order. There are various types of flavor to choose, Chocolate, Vanilla, Mint Chocolate Chip, Buttered Pecan, Cookie Dough and Strawberry. Order contains unique order id, customer id, flavor of ice-cream, quantity and delivery date. One customer can place multiple order and choose multiple flavor. A customer has unique id, name, phone number, address and email. An owner owns the Ice cream parlor and he/she employees a manager. The manager manages the staff and the shifts. Multiple staff can work on one shift. One manager can manage one to multiple staff. The manager has id, name, address, phone number and email address as his/her attribute, where a staff has Staff_id, name, address, phone number and email as their attribute. There are two shifts, morning and night. One owner can only have one manager. Ice- cream parlor is a place where various kinds of ice cream can be found. The customer can choose from various flavor that has been offered and place their order to the staff. The staff then makes the product and deliver it to the customer. One staff can serve multiple order. There are various types of flavor to choose, Chocolate, Vanilla, Mint Chocolate Chip, Buttered Pecan, Cookie Dough and Strawberry. Order contains unique order id, customer id, flavor of ice-cream, quantity and delivery date. One customer can place multiple order and choose multiple flavor. A customer has unique id, name, phone number, address and email. An owner owns the Ice cream parlor and he/she employees a manager. The manager manages the staff and the shifts. Multiple staff can work on one shift. One manager can manage one to multiple staff. The manager has id, name, address, phone number and email address as his/her attribute, where a staff has Staff_id, name, address, phone number and email as their attribute. There are two shifts, morning and night. One owner can only have one manager.

NORMALIZATION

Owner__1__Owns____1__Ice_cream parlor

UNF:

Owns(O_PHNUMBER, O_name, O_address, O_email,
IP_NAME, IP_email, IP_address)

1NF:

O_PHNUMBER are multivalued attribute.

1. O_PHNUMBER, O_name, O_address, O_email, IP_NAME, IP_email, IP_address.

2NF:

1. O_name, O_address, O_email, O_PHNUMBER
2. IP_NAME, IP_email, IP_address.

3NF:

No transitive dependency.

1. O_PHNUMBER, O_name, O_address, O_email.
2. IP_NAME, IP_address, IP_email.

Table Creation:

1. O_PHNUMBER, O_name, O_address, O_email.
2. IP_NAME, IP_address, IP_email, **O_email**.

Owner____1____Employ____1____Manager

UNF

Employ (M_PHNUMBER, M_Name, M_Id, M_address, M_Email, O_PHNUMBER, O_name, O_address, O_email)

1NF:

M_PHNUMBER and O_PHNUMBER are multivalued attribute.

1. M_PHNUMBER, M_Name, M_Id, M_address, M_Email, O_PHNUMBER, O_name, O_address, O_email.

2NF:

1. M_PHNUMBER, M_Name, M_Id, M_address, M_Email.
2. O_PHNUMBER, O_name, O_address, O_email.

3NF:

No transitive dependency.

1. M_PHNUMBER, M_Name, M_Id, M_address, M_Email.
2. O_PHNUMBER, O_name, O_address, O_email.

Table Creation:

1. M_PHNUMBER, M_Name, M_Id, M_address, M_Email.
2. O_PHNUMBER, O_name, O_address, O_email, M_Id.

Manager__1__Manage_____*__Staff

UNF

Manage (S_PHNUMBER, S_Name, S_Id, S_Address, S_Email, M_PHNUMBER, M_Name, M_Id, M_address, M_Email)

1NF

S_PHNUMBER and M_PHNUMBER are multivalued attribute

1. S_PHNUMBER, S_Name, S_Id, S_Address, S_Email, M_PHNUMBER, M_Name, M_Id, M_address, M_Email.

2NF

1. S_PHNUMBER, S_Name, S_Id, S_Address, S_Email.
2. M_PHNUMBER, M_Name, M_Id, M_address, M_Email.

3NF

No transitive dependency.

1. S_PHNUMBER, S_Name, S_Id, S_Address, S_Email.
2. M_PHNUMBER, M_Name, M_Id, M_address, M_Email.

Table Creation:

1. S_PHNUMBER, S_Name, S_Id, S_Address, S_Email.
2. M_PHNUMBER, M_Name, M_Id, M_address, M_Email.
3. **M_id, s_id.**

Staff____*____Serve_____*____Order_O

UNF

Serve(Odr_Id, Odr_Quantity, S_PHNUMBER, S_Name, S_Id, S_Address, S_Email)

1NF

S_PHNUMBER are multivalued attribute.

1. Odr_Id, Odr_Quantity, S_PHNUMBER, S_Name, S_Id, S_Address, S_Email.

2NF

1. Odr_Id, Odr_Quantity.
2. S_PHNUMBER, S_Name, S_Id, S_Address, S_Email.

3NF

No transitive dependency.

1. Odr_Id, Odr_Quantity.
2. S_Id, S_Name, S_PHNUMBER, S_Address, S_Email.

Table Creation:

1. Odr_Id, Odr_Quantity.
2. S_Id, S_Name, S_PHNUMBER, S_Address, S_Email.
3. **Odr_Id , S_Id.**

Order_O ____ * ____ Order Info ____ * ____ Customer

UNF

Order Info(Cus_Id, Cus_Name, Cus_Email, CUS_PHNUMBER, Cus_Address, Odr_Id, Odr_Quantity)

1NF

CUS_PHNUMBER are multivalued attribute

1. Cus_Id, Cus_Name, Cus_Email, CUS_PHNUMBER, Cus_Address, Odr_Id, Odr_Quantity.

2NF

1. Cus_Id, Cus_Name, Cus_Email, CUS_PHNUMBER, Cus_Address.
2. Odr_Id, Odr_Quantity.

3NF

No transitive dependency.

1. Cus_Id, Cus_Name, Cus_Email, CUS_PHNUMBER, Cus_Address
2. Odr_Id, Odr_Quantity.

Table Creation:

1. Cus_Id, Cus_Name, Cus_Email, CUS_PHNUMBER, Cus_Address.
2. Odr_Id, Odr_Quantity
3. **Odr_id, cus_id.**

Order_O ____ * ____ Contains ____ * ____ Flavour

UNF

Contains(F_Id, F_Name, Odr_Id, Odr_Quantity)

1NF

There is no multivalued attribute. Relation already in 1NF.

1. F_Id, F_Name, Odr_Id, Odr_Quantity.

2NF

1. F_Id, F_Name.
2. Odr_Id, Odr_Quantity.

3NF

No transitive dependency.

1. F_Id, F_Name.
2. Odr_Id, Odr_Quantity

Table Creation:

1. F_Id, F_Name.
2. Odr_Id, Odr_Quantity, **F_Id**.

Table Sorting

- ~~1. O_PHNUMBER, O_name, O_address, O_email.~~
2. IP_NAME, IP_address, IP_email, **O_email**.
- ~~3. M_PHNUMBER, M_Name, M_Id, M_address, M_Email.~~
4. O_PHNUMBER, O_name, O_address, **M_ID**
- ~~5. S_PHNUMBER, S_Name, S_Id, S_Address, S_Email.~~
6. M_PHNUMBER, M_Name, M_Id, M_address, M_Email.
7. **M_id, S_id.**
- ~~8. Odr_Id, Odr_Quantity.~~
9. S_Id, S_Name, S_PHNUMBER, S_Address, S_Email.
10. **Odr_Id, S_Id.**
11. Cus_Id, Cus_Name, Cus_Email, CUS_PHNUMBER, Cus_Address.
12. **Odr_id, cus_id.**
13. ~~Odr_Id, Odr_Quantity~~
14. F_Id, F_Name.
15. Odr_Id, Odr_Quantity, **F_Id**.

Final Table

1. IP_NAME, IP_address, IP_email, **O_email**.
2. O_PHNUMBER ,O_name, O_address, O_email ,**M_ID**.
3. M_PHNUMBER, M_Name, M_Id, M_address, M_Email.
4. **M_id**,**S_id**.
5. S_Id, S_Name, S_PHNUMBER, S_Address, S_Email.
6. **Odr_Id , S_Id**.
7. Cus_Id, Cus_Name, Cus_Email, CUS_PHNUMBER, Cus_Address.
8. Odr_Id, Odr_Quantity, **F_ID**
9. **Odr_id**, **cus_id**.
10. F_Id, F_Name.

SCHEMA DIAGRAM

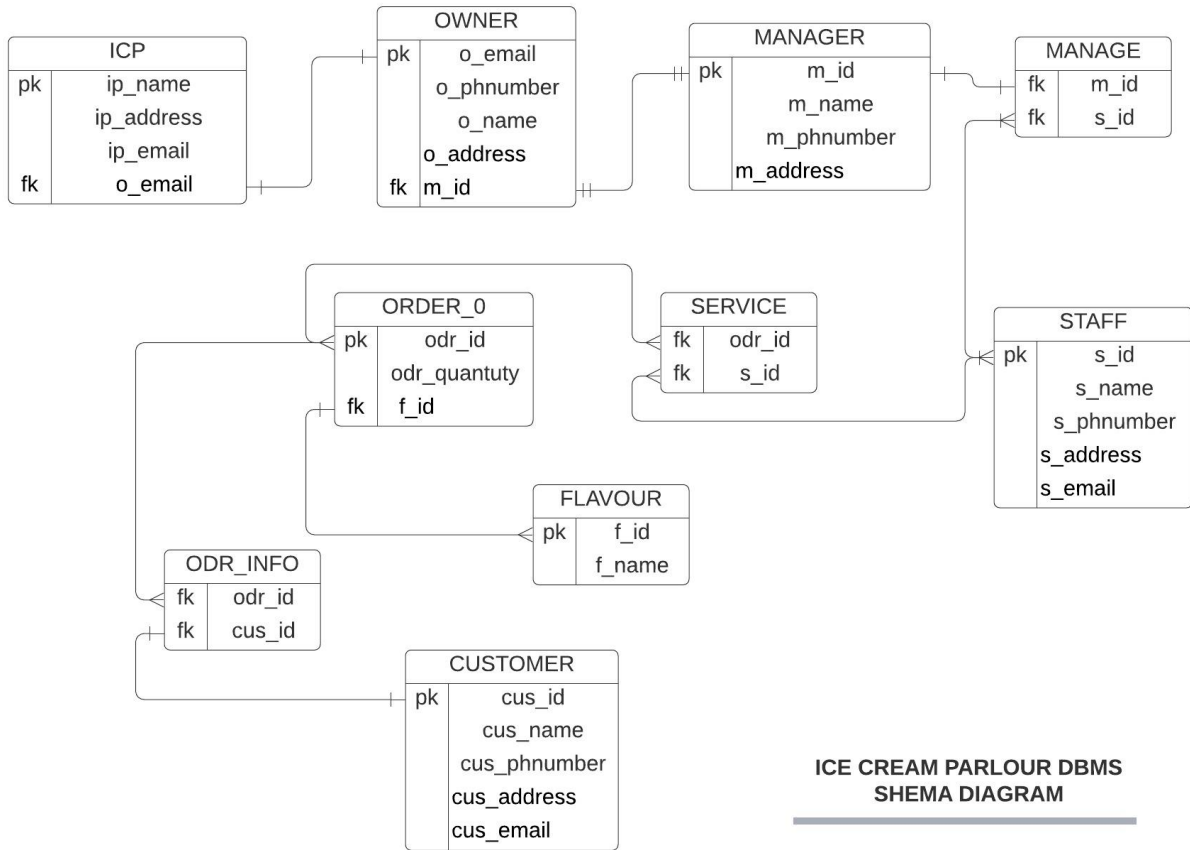


TABLE CREATION QUERY

1. ICP :

```
create table icp(ip_name varchar2(30),ip_address
varchar2(20),ip_email varchar2(50),o_email varchar2(50),
constraint cons_ippk primary key(ip_email));
```

```
ALTER TABLE ICP
add constraint cons_icpfk foreign key(o_email)
references owner(o_email);
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **ICP**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ICP	IP_NAME	Varchar2	30	-	-	-	✓	-	-
	IP_ADDRESS	Varchar2	20	-	-	-	✓	-	-
	IP_EMAIL	Varchar2	50	-	-	1	-	-	-
	O_EMAIL	Varchar2	50	-	-	-	✓	-	-

1 - 4

2. OWNER:

```
CREATE TABLE OWNER (
o_email varchar(30),
o_phnumber number(14),
o_name varchar(20),
o_address varchar(20),m_id NUMBER(10),constraint
con_ownerpk primary key(o_email));
```

```
ALTER TABLE owner
add constraint cons_ownfk foreign key(m_id)
references manager(m_id);
```

Results Explain Describe Saved SQL History

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

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
OWNER	O_EMAIL	Varchar2	30	-	-	1	-	-	-
	O_PHNUMBER	Number	-	14	0	-	✓	-	-
	O_NAME	Varchar2	20	-	-	-	✓	-	-
	O_ADDRESS	Varchar2	20	-	-	-	✓	-	-
	M_ID	Number	-	10	0	-	✓	-	-

1 - 5

3. MANAGER:

```
CREATE TABLE MANAGER (
  m_id NUMBER(10),
  m_name VARCHAR2(20),
  m_phnumber NUMBER(10),
  m_address VARCHAR2(20),constraint con_managerpk
primary key(m_id) );
```

Results Explain Describe Saved SQL History

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

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>MANAGER</u>	<u>M_ID</u>	Number	-	10	0	1	-	-	-
	<u>M_NAME</u>	Varchar2	20	-	-	-	✓	-	-
	<u>M_PHNUMBER</u>	Number	-	10	0	-	✓	-	-
	<u>M_ADDRESS</u>	Varchar2	20	-	-	-	✓	-	-

4. MANAGE:

```
CREATE TABLE MANAGE (
  m_id NUMBER(10),
  s_id NUMBER(10),constraint cons_mngfk foreign
key(m_id) references manager(m_id), constraint
cons_mngfk2 foreign key(s_id) references staff(s_id) );
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **MANAGE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>MANAGE</u>	<u>M_ID</u>	Number	-	10	0	-	✓	-	-
	<u>S_ID</u>	Number	-	10	0	-	✓	-	-

1 - 2

5. STAFF:

```
CREATE TABLE STAFF (
  s_id NUMBER(10),
  s_name VARCHAR2(20),
  s_phnumber NUMBER(14),
  s_address VARCHAR2(30),
  s_email VARCHAR2(20),
  constraint con_staffpk primary key(s_id));
```

Results Explain Describe Saved SQL History

Object Type TABLE Object STAFF

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STAFF	S_ID	Number	-	10	0	1	-	-	-
	S_NAME	Varchar2	20	-	-	-	✓	-	-
	S_PHNUMBER	Number	-	14	0	-	✓	-	-
	S_ADDRESS	Varchar2	30	-	-	-	✓	-	-
	S_EMAIL	Varchar2	20	-	-	-	✓	-	-

1 - 5

6. CUSTOMER:

```
CREATE TABLE CUSTOMER (
  cus_id NUMBER(10),
  cus_name VARCHAR2(20),
  cus_phnumber NUMBER(10),
  cus_address VARCHAR2(30),
  cus_email VARCHAR2(30),constraint con_cuspk primary
  key(cus_id));
```


Results Explain Describe Saved SQL History

Object Type TABLE Object CUSTOMER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CUS_ID	Number	-	10	0	1	-	-	-
	CUS_NAME	Varchar2	20	-	-	-	✓	-	-
	CUS_PHNUMBER	Number	-	10	0	-	✓	-	-
	CUS_ADDRESS	Varchar2	30	-	-	-	✓	-	-
	CUS_EMAIL	Varchar2	30	-	-	-	✓	-	-

1 - 5

7. FLAVOUR:

```
CREATE TABLE "FLAVOUR" (
  f_id NUMBER(10),
  f_name VARCHAR(20),constraint con_flavpk primary
key(f_id)
);
```

Results Explain Describe Saved SQL History

Object Type TABLE Object FLAVOUR

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FLAVOUR	F_ID	Number	-	10	0	1	-	-	-
	F_NAME	Varchar2	20	-	-	-	✓	-	-
									1 - 2

8: ORDER_O:

```
CREATE TABLE ORDER_O (
  odr_id NUMBER(10),
  odr_quantuty NUMBER(10),
  f_id NUMBER(10),constraint con_odrpk primary key(odr_id),
```

constraint cons_odrfk1 foreign key(f_id) references
flavour(f_id));

Results Explain Describe Saved SQL History

Object Type TABLE Object ORDER_O

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDER_O	ODR_ID	Number	-	10	0	1	-	-	-
	ODR_QUANTUTY	Number	-	10	0	-	✓	-	-
	F_ID	Number	-	10	0	-	✓	-	-

1 - 3

9.ODR_INFO:

```
CREATE TABLE ODR_INFO(
  odr_id NUMBER(10),
  cus_id NUMBER(10),
  constraint cons_odrinfk1 foreign key(odr_id) references
ORDER_O(odr_id),
  constraint cons_odrinfk2 foreign key(cus_id) references
CUSTOMER(cus_id)
);
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ODR_INFO

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ODR_INFO	ODR_ID	Number	-	10	0	-	✓	-	-
	CUS_ID	Number	-	10	0	-	✓	-	-

1 - 2

10.SERVICE:

```
CREATE TABLE SERVICE (
```

```

odr_id NUMBER(10),
s_id NUMBER(10),
constraint cons_srvfk1 foreign key(odr_id) references
ORDER_O(odr_id),
constraint cons_srvfk2 foreign key(s_id) references
STAFF(s_id)
);

```

Results Explain Describe Saved SQL History									
Object Type		TABLE		Object		SERVICE			
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SERVICE	ODR_ID	Number	-	10	0	-	✓	-	-
	S_ID	Number	-	10	0	-	✓	-	-
1 - 2									
Application Express 2.1.0.00.39									
Copyright © 1999, 2006, Oracle. All rights reserved.									
Language: en-us									

INSERTION QUERY

ICP:

1.insert into icp values('SUB
ZERO','Gazipur,Dhaka','subzero@gmail.com','abc@gmail.com') ;

OWNER:

1. insert into owner
values('abc@gmail.com',01248524,'GAZi','Gazipur,Dhaka',120) ;

MANAGER:

1. insert into owner values(120,'david' ,'Gazipur,Dhaka',154785) ;

STAFF:

1.insert into staff

values(011,'joy',0154785,'dhaka','joy@gmail.com') ;

2.insert into staff

3.values(012,'ashiq',01485785,'cumilla','asiq@gmail.com') ;

4.insert into staff

values(013,'jackman',01267585,'gulshan','jack@gmail.com') ;

Service:

1. INSERT INTO service values(2201,11);

MANAGE:

1.INSERT INTO manage(m_id,s_id)

SELECT m_id,s_id FROM manager, staff

Customer:

1.insert into customer

values(03471,'robi',014656875,'dhaka','robi@gmail.com') ;

Flavour:

1. INSERT INTO flavour values(001,'vanilla');

2.INSERT INTO flavour values(002,'mango');

3.INSERT INTO flavour values(003,'chocolate');

Order_o:

1. INSERT INTO order_o values(2201,2,003);

Odr_info:

1. INSERT INTO odr_info values(2201,03471);