C.V. RAMAN GLOBAL UNIVERSITY

BHUBANESWAR, ODISHA-752054

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

A Case Study Report of DBE LAB

on

Mess Management System

Submitted By:

1. **Sachin Kumar (Leader)**
2. **Roshan Parida**
3. **Prithvi Som**
4. **Anish Kumar Sahoo**
5. **Smaran Kar**

Outline

1. Team Details: -

Name	Regd No.	Roll No.
Roshan Parida	20010001	CSE20002
Sachin Kumar	20010009	CSE20010
Anish Kumar Sahoo	20010026	CSE20031
Prithvi Som	20010029	CSE20034
Smaran Kar	20010054	CSE20004

2. Abstract: -

The main aim of Mess Management System is that the mess should provide clean and fresh food to the customers of the organization and maintain the records of customers availing the mess services including the billing and type of meals availed.

Existing System:

In many organizations, entire mess management and billing calculations are done manually till date. It is very time consuming and increases the chances of performing calculation mistakes. It would be possible to do the same work within a short period of time and without using many efforts and manpower if there existed a software/ database for the same.

Thus, there arises a need to create a database for the same. Such a database would make the entire Mess related management through ER-Model.

Proposed System:

Mess management system using database helps you effectively manage the mess in smooth manner. This reduces the burden on staff as well as reduces paperwork and keeps all student records, their meals, breakfast, requirements and other essentials up to date.

This module is specially designed to deal with mess management, including inventory details, payments, food specifications and details, billing, etc

3. **Introduction**: -

The Mess Management System helps the user to access all the functionalities of the mess without having to visit the mess physically and to apply for leave. It enables the admin/owner to view the inventory and access customer and mess staff details.

The main aim behind this report is to get the current status of mess & meals per day, to manage details regarding the stocks of meals, customers availing the meal services and also the info about the mess staffs.

The database also provides the costing and monthly calculations of each type of meal for a customer and mess staffs. In the database we are maintaining the entire detail of the customer such of type of meal, billings, queries regarding the mess facility, etc. It also includes information about the mess staffs working, service behaviors, food quality, etc.

Using the information provided by all the mess management database, the admin can take decisions and the inventory for total mess can be managed efficiently.

3.1. Description: -

This data module helps to supervise the Mess Staffs' details including the job, salary. It also decides the type of Meal based on customer/ mess members choice and also store the details of Mess members including the Bills based on due and payment on daily or monthly basis and Feedback is given to mess staffs changes or modify the meal to improve the food quality. Mess members can also complaint in case of poor servicing of the mess staffs.

It generates customers records for mess and assigns them to mess staffs so that the mess organization can work concurrently without creating any hindrance. It also allows the admin/mess supervisor to manage & optimize the whole mess inventory.

Features:

- Publish daily meal calendars for students/ faculty/ staff.
- Record all customers who availed mess facility on daily basis.
- Maintain food/grocery stock of mess
- Reminders can be sent to maintain hygiene of mess-to-mess staff by sending complaints or receiving queries.

Benefits:

- Easy communication
- Minimize day to day operational costs
- Efficient tracking of mess products
- 100% Accurate data for maintaining mess accounts
- Saves time and Effort
- Real time check on inventory of mess
- Better Hygiene Maintenance

4. Problem Statement: -

1. A Mess Staff can supervise the information of multiple mess staffs.
2. Different types of Meals are decided by the individual Mess Staff.
   There is an Entity Meal which have some attributes as Item No., Item Name, Complimentary food along with the timings of the meals for breakfast, lunch, and dinner.
3. A type of meal is selected/ chosen by a mess member in the mess.
4. Then the member must pay the bill for the respective month after he finishes his completion of every month in the mess.

Bill details such as Date, Time, Month, Due Amount, Payment done is saved in the mess database.

5. Any member of mess can complain about the mess staff in case of bad meal, improper timing of meal or in case of bad behaviour of the mess Staff member.
6. Feedback is given by the members of mess for the mess service of meals, meal quality rating, behaviour of staff and so on.
   Feedback details is stored according to following date, time and rating of the food.
7. According to Feedback received the mess manages the meal based on improving meal rating, timing, etc.
8. Mess members stays in the room of the hostel. Allotment is done based on type of room they choose according to that room no is provided to the mess member.

4.1. <u>Relationship and Participation Constraint</u>

1. Relationship (N :1), Recursive Relationship both full participations.
2. Relationship (1: N), Meal is total and Manager is partial participation. There is an Entity Meal which have some attributes as Item No., Item Name, Complimentary food, Meal Type along with the timings of the meals for breakfast, lunch, and dinner.
3. Relationship (1: N) meal partial and mess member partial.
4. Relationship (1:1) both are Total participation. Bill details such as Date, Time, Month, Due Amount, Payment done is saved in saved in the mess database.
5. Relationship (M: N) partial participation for Mess Members and Mess Staff.
6. Feedback details is stored according to following date, time and rating of the food. Relationship (1: 1) full participation
7. Relationship (M: N). Feedback is Total participation and Meal is partial participation.
8. Relationship (M: 1). Mess Member and Room are both total participations.

5. <u>Feasibility Study and Requirement Analysis</u>: -

Feasibility Study-

Yes, it is **technically feasible** to design such a system. There is technology available which can be used to easily design Mess Management system. The data storage capability of our system is large i.e., it can store data without running out of space for many years to come. This system is very responsive, and the servers can be used to fetch data wherever you are in the world. The data required will be instantly delivered without any delay. It is a very reliable system and a very safe system where data security is of utmost importance. Our system is also **economically feasible**. It can be installed easily and the cost of implementing is not that high. The benefit of once implementing this system is immense and will be beneficial for coming number of years.

Requirement Analysis -

- **Usability Requirement:** The web site is designed for user friendly environment and ease of use.
- **Database Security:** Unauthorized person cannot access the panel and database, do not read and write the information.
- **Reliability Requirement:** The system provides a reliable environment for teachers. All requirements are reaching at the admin without any errors.

6. Table Description: -

Mess_Member

Field	Datatype	Length	Constraints
Member_ID	Number	5	Primary Key
Name	Varchar2	20	
Gender	Varchar2	7	
Address	Varchar2	30	
D.O.J. (Date Of Joining)	Date		
Bill_No	Number	5	Foreign Key

Room

Field	Datatype	Length	Constraints
Room_No	Varchar2	10	Partial Key
Memberid	Number	5	Foreign Key
Room Type	Varchar2	5	

Feedback

Field	Datatype	Length	Constraints
Fno	Varchar2	10	Primary Key
Memberid	Number	5	Foreign Key
Date	Date		
Rating	Number	2	

Mess Staffs

Field	Datatype	Length	Constraints
Eno	Varchar2	5	Primary Key
Name	Varchar2	20	
Gender	Varchar2	7	
Address	Varchar2	30	
D.O.J.	Date		
Salary	Number	8	
Mob_No	Number	10	

Meal

Field	Datatype	Length	Constraints
Item_No	Varchar2	5	Primary Key
Item Name	Varchar2	10	
Memberid	Number	5	Foreign Key
Meal Time	Varchar2	10	

Bill

Field	Datatype	Length	Constraints
Bill_No	Varchar2	10	Primary Key
Date	date		
Month	Varchar2	15	
Payment Status	Varchar2	8	
Dues Amount	Number	8	

Meal Complementary Food

Field	Datatype	Length	Constraints
COMPLEMENTARY_FOOD	Number	10	Primary Key
ITEM_NO	Number	2	Foreign Key

Changes

Field	Datatype	Length	Constraints
Item_No	Number	2	Primary Key
FNO	Number	5	Foreign Key
MemberId	Number	5	Foreign Key

Mess_Member_Mobile_No

Field	Datatype	Length	Constraints
MOBILE_NO	Number	10	Primary Key
MemberId	Number	5	Foreign Key

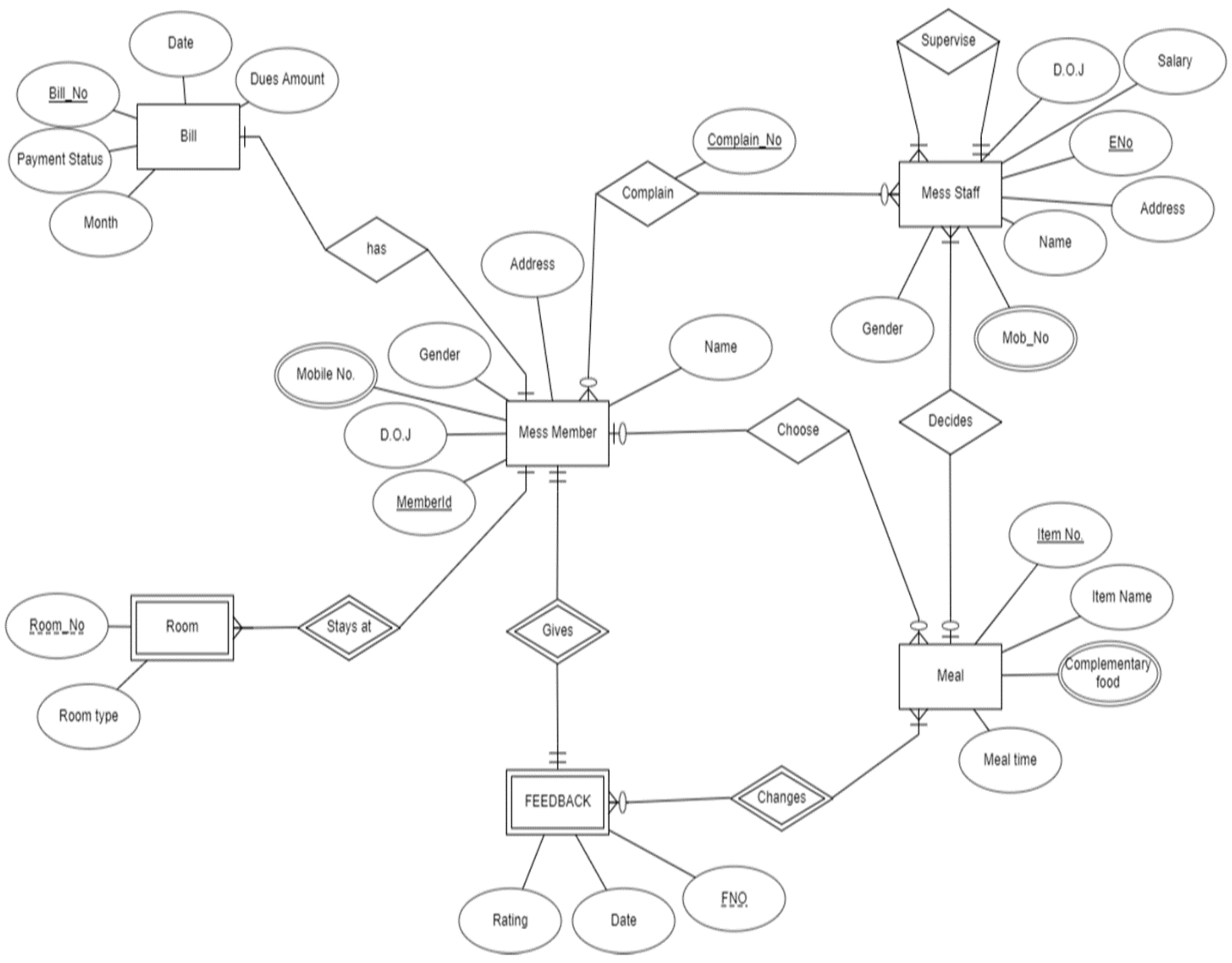
Mess_Staff_MobNo

Field	Datatype	Length	Constraints
MOBILE_NO	Number	10	Primary Key
ENo	Number	5	Foreign Key

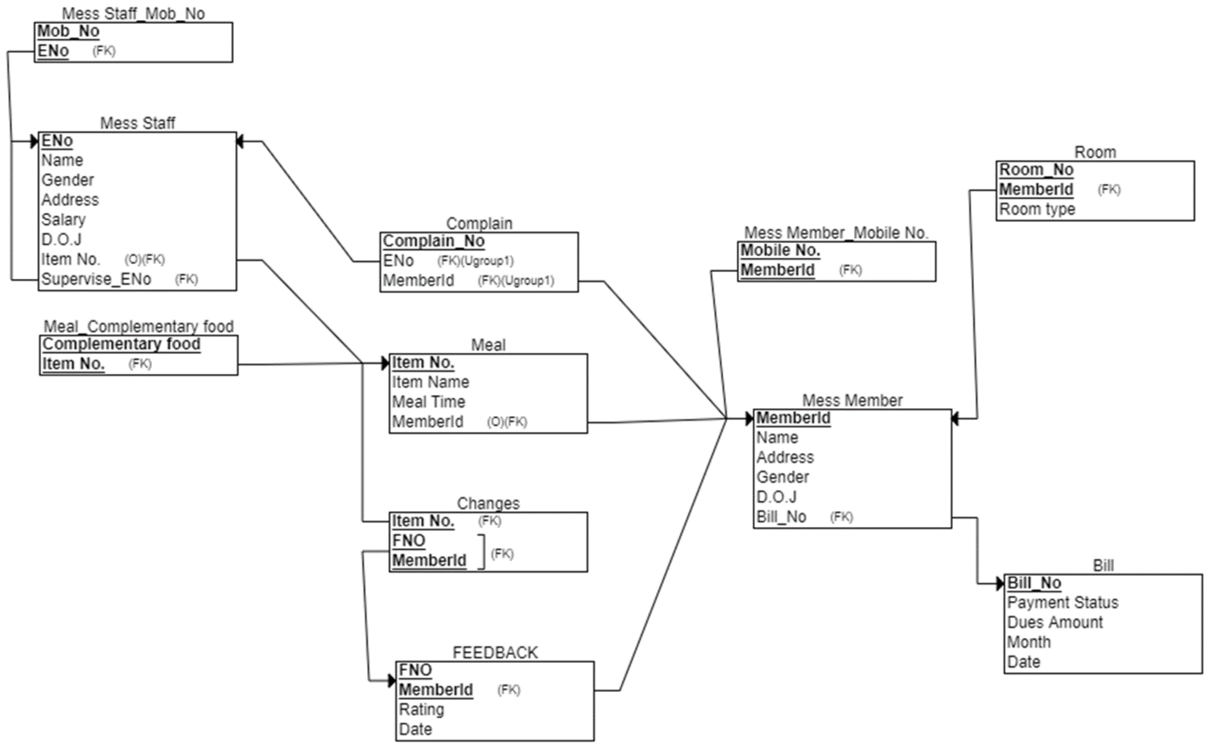
Complain

Field	Datatype	Length	Constraints
Complain_No	Number	5	Primary Key
FNO	Number	5	Foreign Key
MemberId	Number	5	Foreign Key

7. **ER Diagram: -**


Date
Dues Amount
Bill_No
Bill
Payment Status
Month
has
Supervise
D.O.J
Salary
Complain_No
ENo
Complain
Mess Staff
Address
Name
Address
Gender
Mob_No
Name
Gender
Mobile No.
D.O.J
Mess Member
Choose
Decides
MemberId
Item No.
Item Name
Room_No
Room
Stays at
Gives
Meal
Complementary food
Room type
Meal time
FEEDBACK
Changes
FNO
Rating
Date

8. Relational Database Schema: -



9. Normalized Table

Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion, and update anomalies.
So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.

First Normal Form

A relation is in first normal form if every attribute in that relation is singled valued attribute.
(or) A relation schema R is in 1NF, if it does not have any **composite** attributes, **multivalued** attribute or their combination.

1. First Approach

Rules:

- Each table cell should contain a single value.
- Each record needs to be unique.

We have taken a table from the Mess Management System:

<u>Mess_Member_Mobile_No</u>

Member_ID	**Mob_No**
22001	7895236475, 7008389689
22002	9955400877
22003	7488192958
22004	9861775097
22005	9931436435, 8114755079

Table Mess_Member_Mobile_No is not in 1NF because of mutivalued attributes of Mob_No. To make it into 1NF, we decomposed the table as follows:

<u>Mess_Member_Mobile_No</u>

Member_ID	**Mob_No**
22001	7895236475
22001	7008389689
22002	9955400877
22003	7488192958
22004	9861775097
22005	9931436435
22005	8114755079

But in this approach Primary key rule is violates and no. of rows is increased.

2. Second Approach

For multivalued attribute, we create extra column for the attribute so that we can store the extra values containing in each row/tuple.

So to store the multiple Mob_No we insert the values into separate attribute to store the mobile number as shown.

Mess_Member_Mobile_No

Member_ID	Mob_No1	Mob_No2
22001	7895236475	7008389689
22002	9955400877	NULL
22003	7488192958	NULL
22004	9861775097	NULL
22005	9931436435	8114755079

Here, only null values is increased in the second approach.

3.Third Approach

In this approach, we remove the multi-valued attribute that violates 1NF and place it in other table along with the Primary key.

So, we decompose the table into two tables:

Mess_Member_Mobile_No1

Member_ID	Mob_No1
22001	7895236475
22002	9955400877
22003	7488192958
22004	9861775097
22005	9931436435

Mess_Member_Mobile_No2

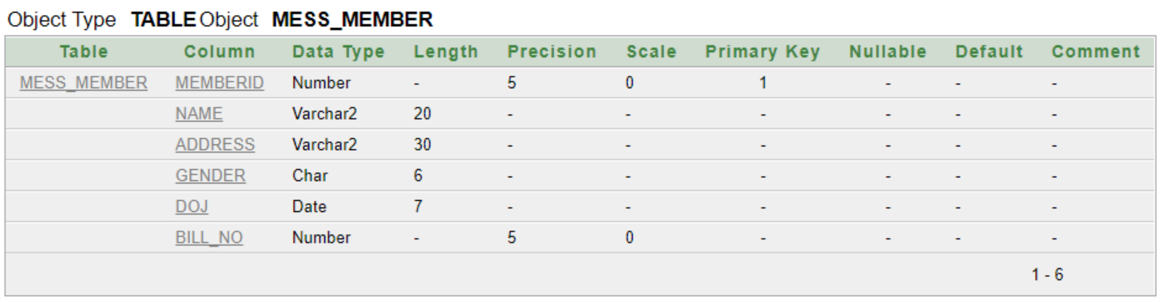
Member_ID	Mob_No2
22001	7008389689
22005	8114755079

This approach is the best in First Normal Form.

10. SQL Queries: -

10.1. Creating Tables

```
CREATE TABLE Mess_Member

(MemberId number(5) Primary Key,

Name varchar2(20) NOT NULL,

Address varchar2(30) NOT NULL,

Gender char(6) NOT NULL,

DOJ date NOT NULL,

Bill_No number(5) not null,

FOREIGN KEY (Bill_No) references Bill(Bill_No) on delete cascade

);

DESC MESS_MEMBER;
```

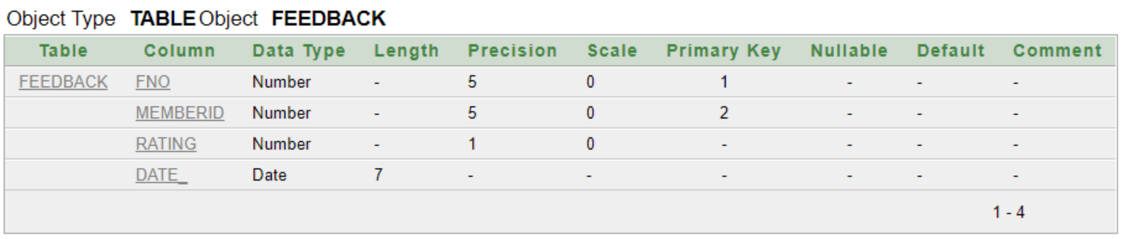


Object Type **TABLE** Object **MESS_MEMBER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MESS_MEMBER	MEMBERID	Number	-	5	0	1	-	-	-
	NAME	Varchar2	20	-	-	-	-	-	-
	ADDRESS	Varchar2	30	-	-	-	-	-	-
	GENDER	Char	6	-	-	-	-	-	-
	DOJ	Date	7	-	-	-	-	-	-
	BILL_NO	Number	-	5	0	-	-	-	-

1 - 6

```
CREATE TABLE FEEDBACK

( FNO number(5) NOT NULL,

  MemberId number(5) NOT NULL,

  Rating number(1) NOT NULL,

  Date_ date NOT NULL,

  PRIMARY KEY (FNO, MemberId),

  FOREIGN KEY (MemberId) REFERENCES Mess_Member(MemberId) on delete cascade

);

desc FEEDBACK;
```

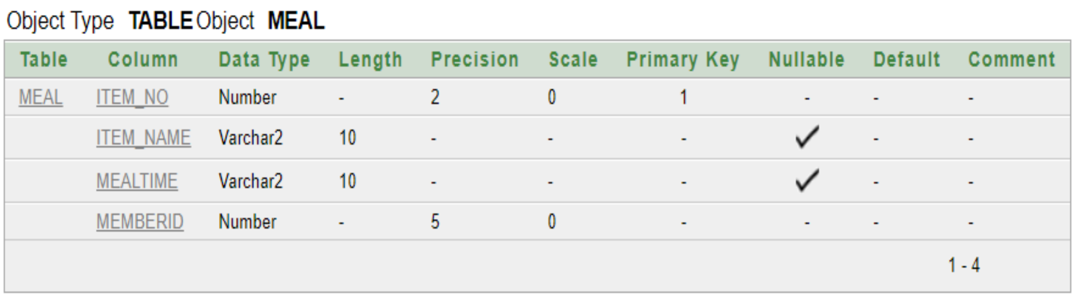


Object Type **TABLE** Object **FEEDBACK**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FEEDBACK	FNO	Number	-	5	0	1	-	-	-
	MEMBERID	Number	-	5	0	2	-	-	-
	RATING	Number	-	1	0	-	-	-	-
	DATE_	Date	7	-	-	-	-	-	-

1 - 4

```
CREATE TABLE Meal
(
  Item_No number(2) NOT NULL,
  Item_Name varchar2(10),
  Mealtime varchar2(10),
  MemberId number(5) not null,
  PRIMARY KEY (Item_No),
  FOREIGN KEY (MemberId) REFERENCES Mess_Member(MemberId) on delete cascade
);
desc Meal;
```

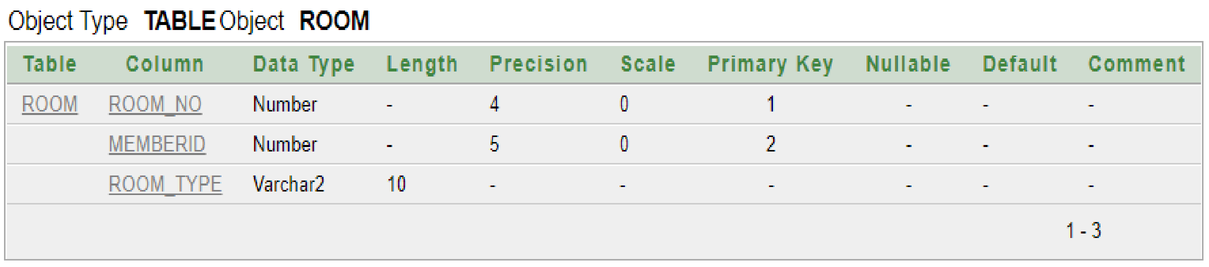


Object Type **TABLE** Object **MEAL**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MEAL	ITEM_NO	Number	-	2	0	1	-	-	-
	ITEM_NAME	Varchar2	10	-	-	-	✓	-	-
	MEALTIME	Varchar2	10	-	-	-	✓	-	-
	MEMBERID	Number	-	5	0	-	-	-	-

1 - 4

```
CREATE TABLE Room
(
  Room_No number(4) NOT NULL,
  MemberId number(5) NOT NULL,
Room_type varchar2(10) NOT NULL,
  PRIMARY KEY (Room_No, MemberId),
  FOREIGN KEY (MemberId) REFERENCES Mess_Member(MemberId) on delete cascade
);
Desc Room;
```

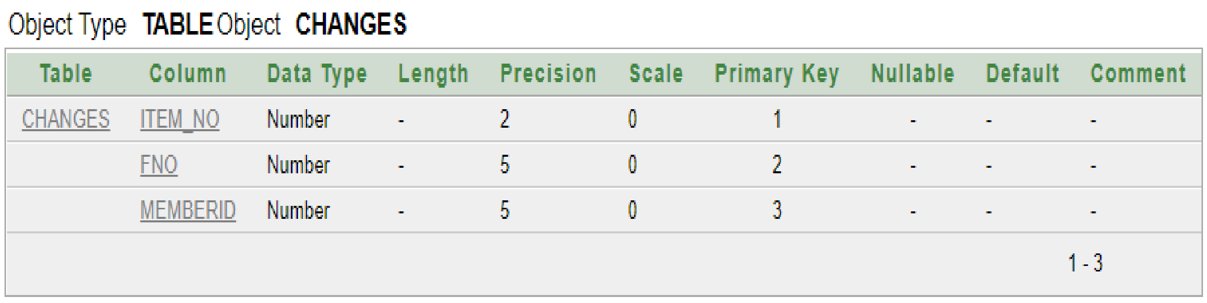


Object Type **TABLE** Object **ROOM**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ROOM	ROOM_NO	Number	-	4	0	1	-	-	-
	MEMBERID	Number	-	5	0	2	-	-	-
	ROOM_TYPE	Varchar2	10	-	-	-	-	-	-

1 - 3

```
CREATE TABLE Changes
(
  Item_No number(2) NOT NULL,
  FNO number(5) NOT NULL,
  MemberId number(5) NOT NULL,
  PRIMARY KEY (Item_No, FNO, MemberId),
  FOREIGN KEY (Item_No) REFERENCES Meal(Item_No),
  FOREIGN KEY (FNO, MemberId) REFERENCES FEEDBACK(FNO, MemberId)
);
desc Changes;
```

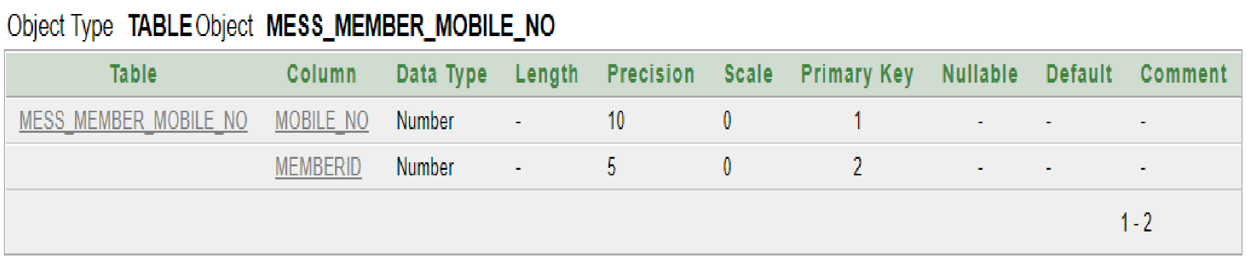


Object Type **TABLE** Object **CHANGES**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CHANGES	ITEM_NO	Number	-	2	0	1	-	-	-
	FNO	Number	-	5	0	2	-	-	-
	MEMBERID	Number	-	5	0	3	-	-	-

1 - 3

```
CREATE TABLE Mess_Member_Mobile_No
(
  Mobile_No number(10) NOT NULL,
  MemberId number(5) NOT NULL,
  PRIMARY KEY (Mobile_No, MemberId),
  FOREIGN KEY (MemberId) REFERENCES Mess_Member(MemberId)
);
Desc Mess_Member_Mobile_No
```

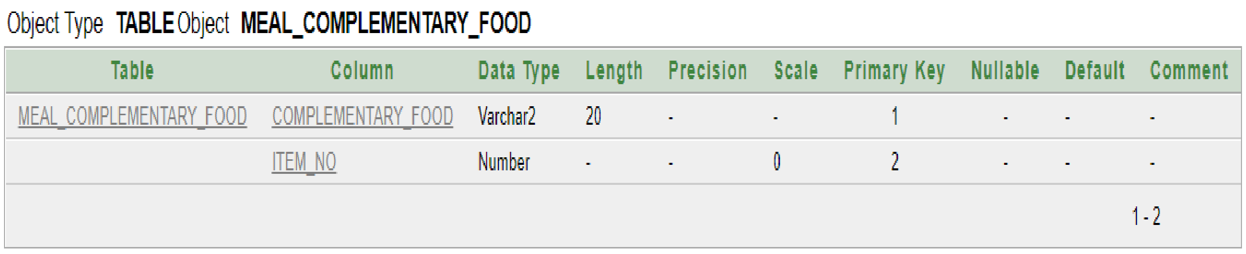


Object Type **TABLE** Object **MESS_MEMBER_MOBILE_NO**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MESS_MEMBER_MOBILE_NO	MOBILE_NO	Number	-	10	0	1	-	-	-
	MEMBERID	Number	-	5	0	2	-	-	-

1 - 2

```
CREATE TABLE Meal_Complementary_food
( Complementary_food varchar2(20) NOT NULL,
  Item_No INT NOT NULL,
  PRIMARY KEY (Complementary_food, Item_No),
  FOREIGN KEY (Item_No) REFERENCES Meal(Item_No)
);

Desc Meal_Complementary_food
```



Object Type TABLE Object MEAL_COMPLEMENTARY_FOOD

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MEAL_COMPLEMENTARY_FOOD	COMPLEMENTARY_FOOD	Varchar2	20	-	-	1	-	-	-
	ITEM_NO	Number	-	-	0	2	-	-	-

1 - 2

```
CREATE TABLE Mess_Staff
(
  ENo number(5),
  Name varchar2(20),
  Gender varchar2(6),
  Address varchar2(30),
  Salary number(7,2),
  DOJ date,
  Item_No number(2) NOT NULL,
  Supervise_ENo number(5) NOT NULL,
  PRIMARY KEY (ENo),
  FOREIGN KEY (Item_No) REFERENCES Meal(Item_No),
  FOREIGN KEY (Supervise_ENo) REFERENCES Mess_Staff(ENo) on delete cascade
);
Desc Mess_Staff
```

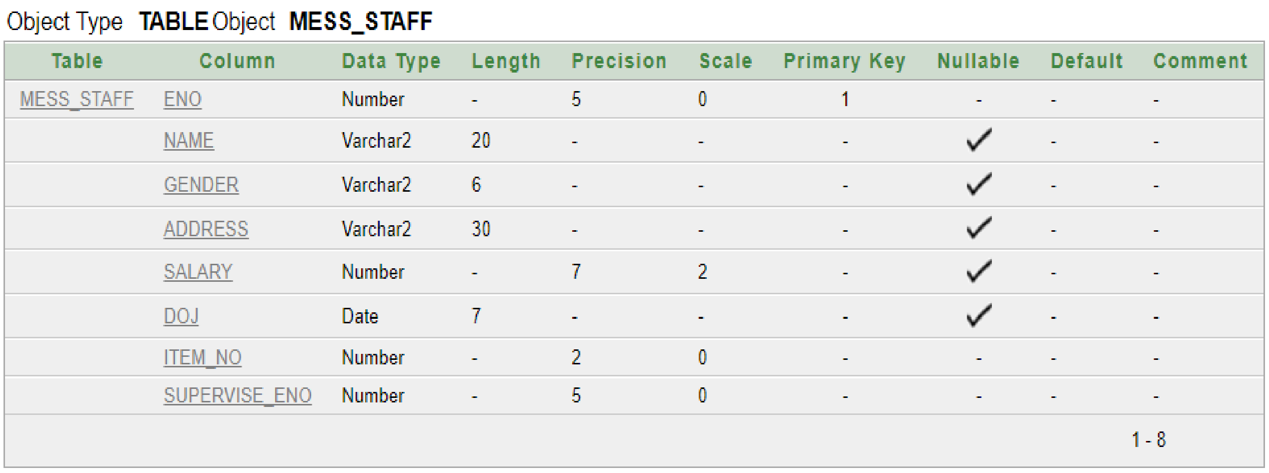

Object Type TABLE Object MESS_STAFF

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MESS_STAFF	ENO	Number	-	5	0	1	-	-	-
	NAME	Varchar2	20	-	-	-	✓	-	-
	GENDER	Varchar2	6	-	-	-	✓	-	-
	ADDRESS	Varchar2	30	-	-	-	✓	-	-
	SALARY	Number	-	7	2	-	✓	-	-
	DOJ	Date	7	-	-	-	✓	-	-
	ITEM_NO	Number	-	2	0	-	-	-	-
	SUPERVISE_ENO	Number	-	5	0	-	-	-	-

1 - 8

```
CREATE TABLE Complain
( Complain_No number(5) NOT NULL,
  ENo number(5) NOT NULL,
  MemberId number(5) NOT NULL,
  PRIMARY KEY (Complain_No),
  FOREIGN KEY (ENo) REFERENCES Mess_Staff(ENo),
  FOREIGN KEY (MemberId) REFERENCES Mess_Member(MemberId),
  UNIQUE (ENo, MemberId)
);
```

desc Complain

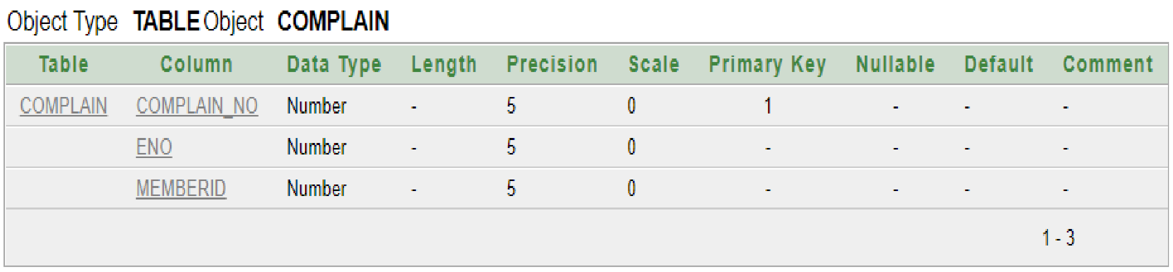
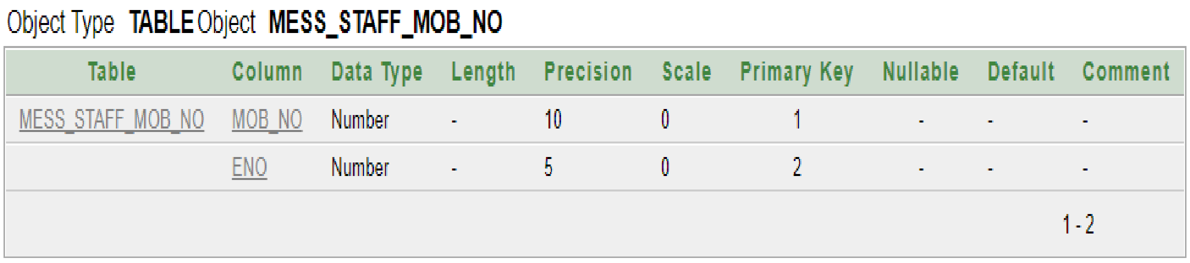

Object Type TABLE Object COMPLAIN

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COMPLAIN	COMPLAIN_NO	Number	-	5	0	1	-	-	-
	ENO	Number	-	5	0	-	-	-	-
	MEMBERID	Number	-	5	0	-	-	-	-

1 - 3

```
CREATE TABLE Mess_Staff_Mob_No
( Mob_No number(10) NOT NULL,
  ENo number(5) NOT NULL,
  PRIMARY KEY (Mob_No, ENo),
  FOREIGN KEY (ENo) REFERENCES Mess_Staff(ENo)
);
```

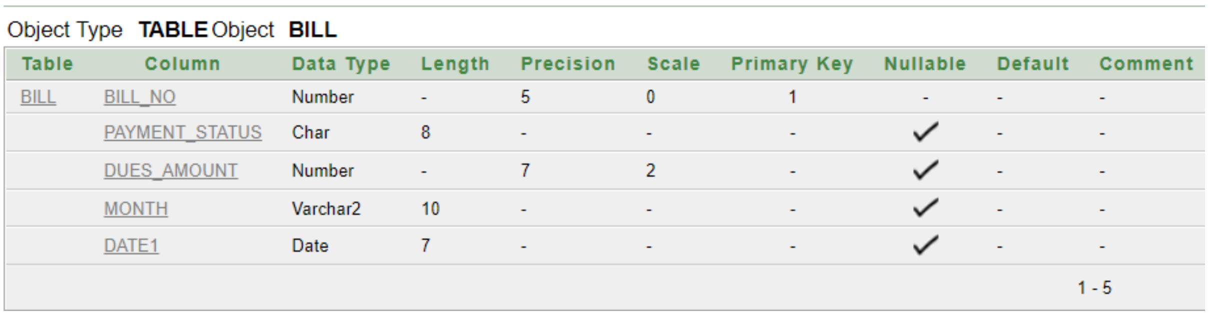
Desc Mess_Staff_Mob_No



Object Type TABLE Object MESS_STAFF_MOB_NO

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MESS_STAFF_MOB_NO	MOB_NO	Number	-	10	0	1	-	-	-
	ENO	Number	-	5	0	2	-	-	-
									1 - 2

```
CREATE TABLE Bill
(
  Bill_No number(5) Primary Key,
  Payment_Status char(8),
  Dues_Amount number(7,2),
  Month varchar2(10),
  Date1 date
);
Desc Bill;
```

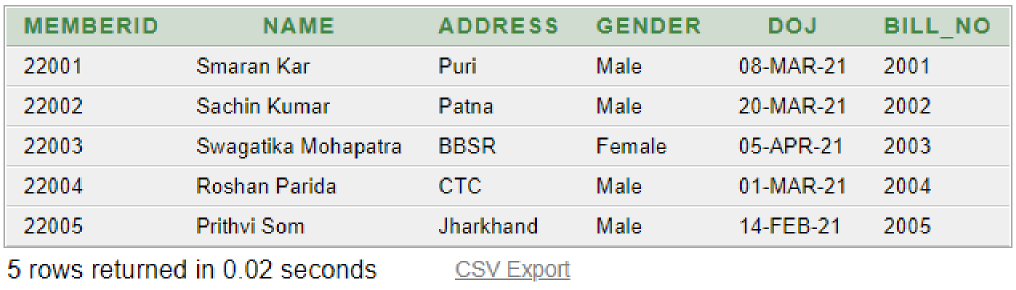


Object Type TABLE Object BILL

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BILL	BILL_NO	Number	-	5	0	1	-	-	-
	PAYMENT_STATUS	Char	8	-	-	-	✓	-	-
	DUES_AMOUNT	Number	-	7	2	-	✓	-	-
	MONTH	Varchar2	10	-	-	-	✓	-	-
	DATE1	Date	7	-	-	-	✓	-	-
									1 - 5

10.2. Inserting Data into Tables-

```
insert into mess_member values(22001,'Smaran Kar','Puri','Male','8-MAR-2021',2001);
insert into mess_member values(22002,'Sachin Kumar','Patna','Male','20-MAR-2021',2002);
insert into mess_member values(22003,'Swagatika Mohapatra','BBSR','Female','5-APR-
2021',2003);
insert into mess_member values(22004,'Roshan Parida','CTC','Male','1-MAR-2021',2004);
insert into mess_member values(22005,'Prithvi Som','Jharkhand','Male','14-FEB-2021',2005);
select * from mess_member;
```



MEMBERID	NAME	ADDRESS	GENDER	DOJ	BILL_NO
22001	Smaran Kar	Puri	Male	08-MAR-21	2001
22002	Sachin Kumar	Patna	Male	20-MAR-21	2002
22003	Swagatika Mohapatra	BBSR	Female	05-APR-21	2003
22004	Roshan Parida	CTC	Male	01-MAR-21	2004
22005	Prithvi Som	Jharkhand	Male	14-FEB-21	2005

5 rows returned in 0.02 seconds CSV Export

insert into Bill values(2001,'Paid',30000,'April','10-APR-2021');

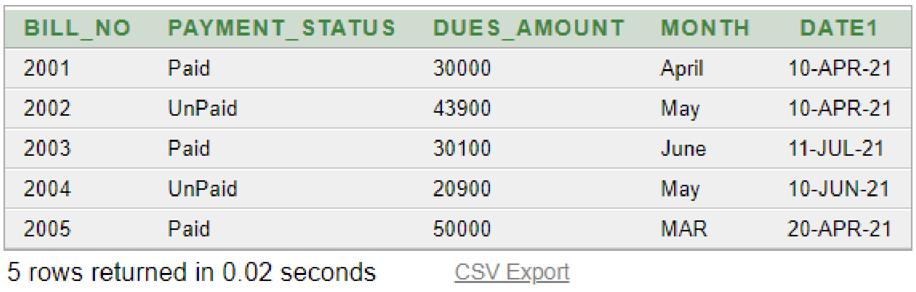
insert into Bill values(2002,'UnPaid',43900,'May','10-APR-2021');

insert into Bill values(2003,'Paid',30100,'June','11-JUL-2021');

insert into Bill values(2004,'UnPaid',20900,'May','10-JUN-2021');

insert into Bill values(2005,'Paid',50000,'MAR','20-APR-2021');

select * from bill;



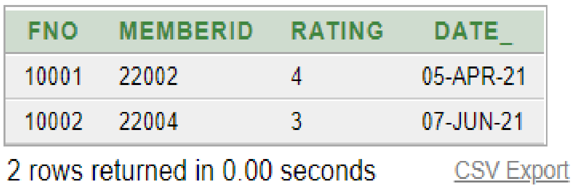
BILL_NO	PAYMENT_STATUS	DUES_AMOUNT	MONTH	DATE1
2001	Paid	30000	April	10-APR-21
2002	UnPaid	43900	May	10-APR-21
2003	Paid	30100	June	11-JUL-21
2004	UnPaid	20900	May	10-JUN-21
2005	Paid	50000	MAR	20-APR-21

5 rows returned in 0.02 seconds CSV Export

insert into feedback values(10001,22002,4,'5-APR-2021'):

insert into feedback values(10002,22004,3,'7-JUN-2021');

select * from feedback;



FNO	MEMBERID	RATING	DATE_
10001	22002	4	05-APR-21
10002	22004	3	07-JUN-21

2 rows returned in 0.00 seconds CSV Export

insert into Meal values(01,'Dal Tadka','Lunch',22001);

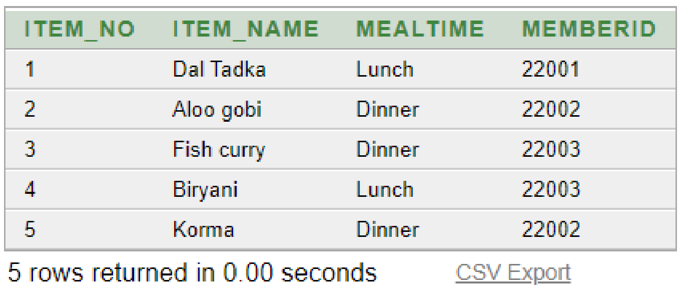
insert into Meal values(02,'Aloo gobi','Dinner',22002);

insert into Meal values(03,'Fish curry','Dinner',22003);

insert into Meal values(04,'Biryani','Lunch',22003);

insert into Meal values(05,'Korma','Dinner',22002);

select * from Meal;



ITEM_NO	ITEM_NAME	MEALTIME	MEMBERID
1	Dal Tadka	Lunch	22001
2	Aloo gobi	Dinner	22002
3	Fish curry	Dinner	22003
4	Biryani	Lunch	22003
5	Korma	Dinner	22002

5 rows returned in 0.00 seconds CSV Export

insert into Mess_Staff values(10001,'Ram','Male','Cuttack',75000,'1-JAN-2000',01,10001);

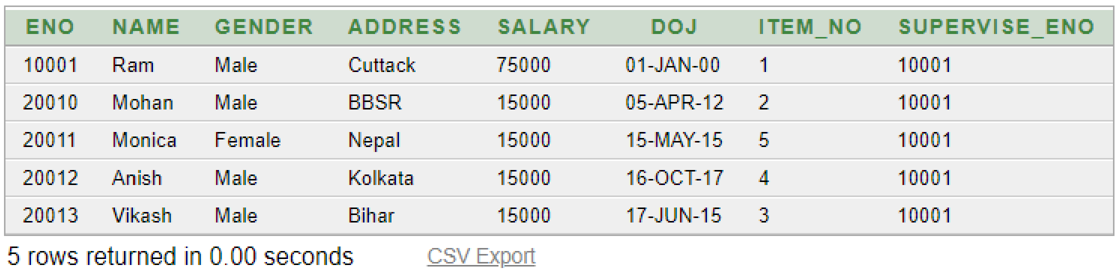
insert into Mess_Staff values(20010,'Mohan','Male','BBSR',15000,'5-APR-2012',02,10001);

insert into Mess_Staff values(20011,'Monica','Female','Nepal',15000,'15-MAY-2015',05,10001);

insert into Mess_Staff values(20012,'Anish','Male','Kolkata',15000,'16-OCT-2017',04,10001);

insert into Mess_Staff values(20013,'Vikash','Male','Bihar',15000,'17-JUN-2015',03,10001);

select * from Mess_Staff;



ENO	NAME	GENDER	ADDRESS	SALARY	DOJ	ITEM_NO	SUPERVISE_ENO
10001	Ram	Male	Cuttack	75000	01-JAN-00	1	10001
20010	Mohan	Male	BBSR	15000	05-APR-12	2	10001
20011	Monica	Female	Nepal	15000	15-MAY-15	5	10001
20012	Anish	Male	Kolkata	15000	16-OCT-17	4	10001
20013	Vikash	Male	Bihar	15000	17-JUN-15	3	10001

5 rows returned in 0.00 seconds CSV Export

insert into Mess_Member_Mobile_No values(7895236475,22001);

insert into Mess_Member_Mobile_No values(7008389689,22001);

insert into Mess_Member_Mobile_No values(9955400877,22002);

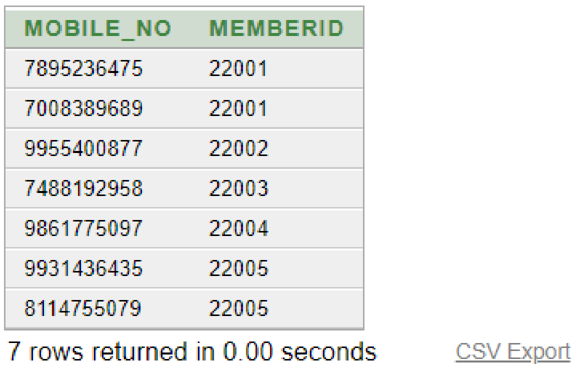
insert into Mess_Member_Mobile_No values(7488192958,22003);

insert into Mess_Member_Mobile_No values(9861775097,22004);

insert into Mess_Member_Mobile_No values(9931436435,22005);

insert into Mess_Member_Mobile_No values(8114755079,22005);

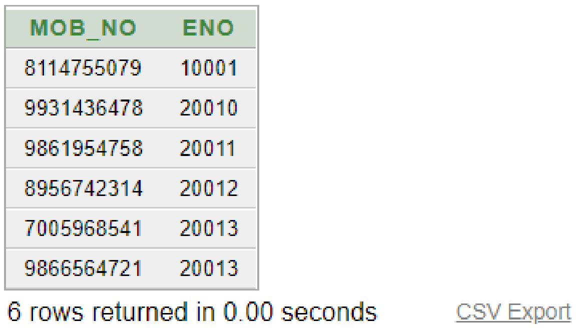
select * from Mess_Member_Mobile_No;



MOBILE_NO	MEMBERID
7895236475	22001
7008389689	22001
9955400877	22002
7488192958	22003
9861775097	22004
9931436435	22005
8114755079	22005

7 rows returned in 0.00 seconds CSV Export

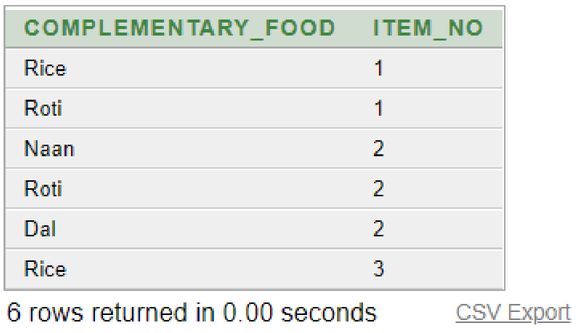
```
insert into Mess_Staff_Mob_No values(8114755079,10001);
insert into Mess_Staff_Mob_No values(9931436478,20010);
insert into Mess_Staff_Mob_No values(9861954758,20011);
insert into Mess_Staff_Mob_No values(8956742314,20012);
insert into Mess_Staff_Mob_No values(7005968541,20013);
insert into Mess_Staff_Mob_No values(9866564721,20013);

select * from Mess_Staff_Mob_No
```



MOB_NO	ENO
8114755079	10001
9931436478	20010
9861954758	20011
8956742314	20012
7005968541	20013
9866564721	20013

6 rows returned in 0.00 seconds CSV Export

```
insert into Meal_Complementary_food values('Rice',01);
insert into Meal_Complementary_food values('Roti',01);
insert into Meal_Complementary_food values('Naan',02);
insert into Meal_Complementary_food values('Roti',02);
insert into Meal_Complementary_food values('Dal',02);
insert into Meal_Complementary_food values('Rice',03);

select * from Meal_Complementary_food;
```



COMPLEMENTARY_FOOD	ITEM_NO
Rice	1
Roti	1
Naan	2
Roti	2
Dal	2
Rice	3

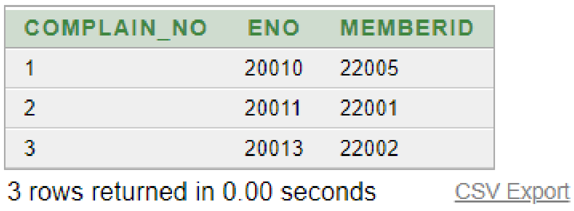
6 rows returned in 0.00 seconds CSV Export

insert into Complain values(01,20010,22005);

insert into Complain values(02,20011,22001);

insert into Complain values(03,20013,22002);

select * from Complain;

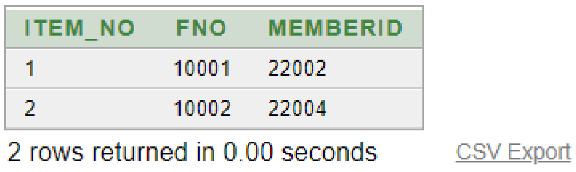


COMPLAIN_NO	ENO	MEMBERID
1	20010	22005
2	20011	22001
3	20013	22002

3 rows returned in 0.00 seconds CSV Export

insert into Changes values(01,10001,22002);

insert into Changes values(02,10002,22004);



ITEM_NO	FNO	MEMBERID
1	10001	22002
2	10002	22004

2 rows returned in 0.00 seconds CSV Export

insert into Room values(301,22002,'AC');

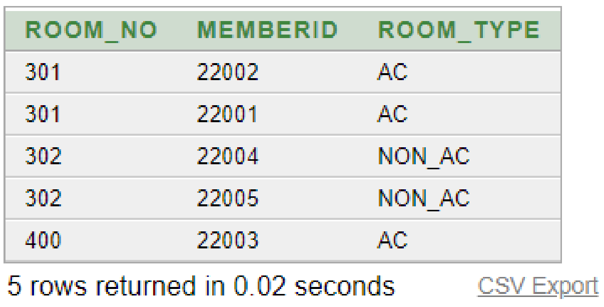
insert into Room values(301,22001,'AC');

insert into Room values(302,22004,'NON_AC');

insert into Room values(302,22005,'NON_AC');

insert into Room values(400,22003,'AC');

select * from Room;



ROOM_NO	MEMBERID	ROOM_TYPE
301	22002	AC
301	22001	AC
302	22004	NON_AC
302	22005	NON_AC
400	22003	AC

5 rows returned in 0.02 seconds CSV Export

11.Conclusion: -

We have successfully presented case study based on Mess Management System. During the development of this project, we first analysed how feasible this system is and what are its requirements. Then we created the table with all the constraints and restrictions.
With keeping view of the table description, we created the ER diagram and how various entities are related to each other. Then we show the schema diagram and how primary key and foreign key are related in each schema. Then we created the normalised forms of the table and completely removed the redundancy of the data and complexity of relations in the system.

With the help of ER- Diagram we create a fully functional system for the Mess Management. Creating it with the help of ER model will help us to fully implement in the back end of the system. This will hold of all the data that'll enter and exit the system. And there won't be any redundancy or inconsistency of data. All the relations are designed with keeping in view of the various relational protocols.

12.References: -

[1] www.Erdplus.com

[2] www.wikipedia.com/Mess_Management_System

[3] http://github.com/Mess_Management_System_SQL

[4] https://www.academia.edu/34352162/Mess_Management_System