

# **BASICS OF DATABASES**

## **ASSIGNMENT**

### **Restaurant Management System**

---

**BY:**

**VRIDHI WADHAWAN**

**PRN 18021021354**

**BATCH 2018-21**

**SUBMITTED TO:**

**DR.NIDHI NATRAJAN**

# CONTENTS

A. THE ENTITY RELATIONSHIP MODEL .....	3
B. BUSINESS SCENARIO .....	3
C. NEED FOR AUTOMATION .....	3
D. ENTITIES AND ATTRIBUTES .....	5
I. ENTITIES .....	6
THE MAIN ENTITIES OF THE RESTURANT MANAGEMENT SYSTEM.....	6
II. ATTRIBUTES.....	6
RESTURANT MANAGEMENT SYSTEM ENTITIES WITH THEIR ATTRIBUTES .....	6
E. E-R DIAGRAM FOR RESTURANT MANAGEMENT SYSTEM .....	8
F. DATABASE TABLES .....	9
I. CUSTOMERS .....	9
II. EMPLOYEES .....	10
III. FOOD ITEMS .....	11
IV. ORDERS .....	12
G. QUERIES AND RESULTS .....	13
I. DISPLAY THE NAME, ADDRESS AND GENDER OF ALL CUSTOMERS WHO ARE MALE .....	13
II. DISPLAY THE NAME, CONTACT NUMBER AND GENDER OF ALL FEMALE CUSTOMERS .....	13
III. DISPLAY THE NAME, AGE, GENDER, MOBILE NUMBER, DESIGNATION AND SALARY OF EMPLOYEES HAVING A SALARY GREATER THAN EQUAL TO 3000 .....	14
IV. COUNT THE TOTAL NUMBER OF EMPLOYEES .....	14
V. DISPLAY ALL FOOD ITEMS WITH PRICE WHEN THE PRICE IS GREATER THAN 150 .....	15
VI. DISPLAY ORDER ID, ORDERING DATE, NAME OF THE ITEM ORDERED WITH ITS QUANTITY AND THE TOTAL BILL AMOUNT WHERE THE BILL AMOUNT IS GREATER THAN RS.500 .....	15
VII.UPDATE THE PRICE OF SANDWICH IN FOOD ITEMS TO RS. 300.....	16
VIII.DISPLAY THE COUNT OF ALL EMPLOYEES GROUPED BY THEIR GENDER .....	16
IX. DISPLAY THE ORDER ID AND THE CUSTOMER'S NAME WHO PLACED THAT ORDER.....	17
X. DISPLAY THE CUSTOMER ID, ORDERING DATE, CUSTOMER'S NAME, MOBILE, ITEMS ORDERED WITH THEIR QUANTITY AND TOTAL BILL AMOUNT.....	17

# THE ENTITY RELATIONSHIP MODEL

At a basic level, databases store information about distinct objects, or entities, and the associations, or relationships, between these entities. A popular approach to conceptual design uses the Entity Relationship (ER) model, which helps transform the requirements into a formal description of the entities and relationships that appear in the database.

## BUSINESS SCENARIO

With improvement in the technology, the life is becoming easier and smarter. People are using the applications that will solve their problems within some time. Restaurant is the place where the people come together to have the food of their choice. A restaurant business is a one which is quite complex as it requires continuous communication from various people involved from the customers to the chefs. Communication is vital component of a restaurant management system.

As a customer has to go to the restaurant or make a phone call earlier to reserve their meal. When they walk in the restaurant, they have to be greeted and shown to their table depending on weather they have prior reservation or not and the free tables available. After settling in, the customer has to place their order from the menu of that particular restaurant and it is the duty of the employees to take the order correctly and communicate the same to the chef who then prepares the order and hands it back to the employee (in this case the waiter) who then serves the same to the customer. After the meal the customer has to be given the total bill amount for his order and has to pay that amount which also includes taxes and a tip for the server.

Thus due to its very dynamic nature of functioning, communication between the servers and the customer is an integral as if there is a gap in communication the consequences of the same can be drastic and thus can bring the whole system down for a customer and can result in loss of sales for the restaurant. And hence it plays a very integral to develop an automated and digital database system which will be able to store the details about the employees working and the roles that they have for the restaurant to run smoothly and thus better the service being provided and increase the revenue for the same while leaving happy customers. Thus a restaurant can benefit from it greatly.

## NEED FOR AUTOMATION

A conventional system of record keeping has many drawbacks which includes difficulty in immediate retrievals of the relevant documents, large information storage problems which further include the problem of their physical storage and difficulty in maintain and updating the same. It also delays calculations as they have to be done manually and thus there is a high probability of error in the same calculations. Thus it makes it difficult to prepare accurate and prompt reports with the required information.

In the conventional system in a restaurant management system, a waiter would have booked the order on paper so there is probability of loss and duplicates customer information. Also the manual system of paper makes it difficult for restaurant administrator to find the waiter information and thus a probability of missing the paper containing information and difficult to arrange the schedule. There also comes a problem of customer getting connected over the phone would be harder when the restaurant is very popular and busy. Waiter and customer information is important to restaurant administrator for reference in the future and the traditional method doesn't offer immediate information when it might be required. Also the chances of committing mistakes at the restaurant in providing a menu list for a specific time would be more if there is no automation. Furthermore, restaurant needs management in the menu as it is important for the waiters to manage the menu. As customer won't be able to order without a menu list with them as they won't remember the entire list with price and come to a decision. There might be some communication problems in the manual system as it is not effective and efficient anymore. As entire booking has to be done manually at the restaurant end, the chances of occurrence of mistakes is high as well. The manual system cannot save, manage and monitor the restaurant waiter information, menu information, customer ordering information and generate report well. Even assigning orders to a specific cook can be cumbersome if it is done parallel with the bookings of the order.

This is where an automated data base management system comes to play as it has many advantages over the conventional system of record keeping. These advantages include Data sharing or transferring as this allows the user to share the given data with number of people without creating a physical copy or duplication. As in case of a restaurant, when taking an order one can directly send an automated copy of what the customer wants thus eliminating the need for the employee to go to the kitchen to give the order and also eliminating the chance of giving the wrong order to the kitchen.

Automation also provides an added advantage of data security as it protects the data from piracy and keeps the important data safe and makes it available only for those to see who have the right authorization to do the same. As in case of a restaurant not every employee will be authorized to see the credit card of the customer but only the manager so that there is no chance of misuse of information and thus helps in keeping accountability from the owner's side. This can also be done for when the restaurant employees are to be given salary not everyone will be given access to employee's bank account details and their social security number.

Another advantage that one sees is that data base management system ensures data integration which means that all the data is well managed and synchronized making handling the data easier. So in a restaurant in case of a security breach of say theft one will know who all were in the premises and not only that but it helps to make the menu well synchronized by removing old dishes which the restaurant is no longer serving and adding new dishes which the restaurant has introduced. This further helps to remove the data inconsistency as it removes the incorrect data thus minimising inconsistent entries and redundancies. Especially in restaurant database system as it removes say duplication of records of a customer, employees or even food items.

Not only that but DBMS also helps in providing data access to the right people who can run apt queries to answer their questions and thus make the right decisions based on the updated data to help improve the business further by enabling the apt information at a much accurate and faster rate to help with the business decisions. In a restaurant by having the right information at the right time can help to further drive the business up as say one particular item is selling like hotcakes in a week so the manager having the apt and required information can make an informed decision of driving the prices up for that particular food item thus driving the profits and revenue up for the restaurant. Thus automated DBMS enables the manager to better manage data and access to it generates higher quality information based on which one can make better informed decisions as the manager did when driving the prices up of a food item when it sales increased in the past week. Lastly DBMS also provides a simplistic approach towards data as it provides a clear and logical view of data thus making inconsistencies easily detectable and thus makes the data manageable. As in the case of restaurant all the information regarding the customers and employees are stored in one place maybe even cloud.

Restaurant management system puts the order in a queue with specific priority according to time and quantity, and then a cook is assigned for the specific order to complete it. Also, it can be used by employees in the restaurant to handle the customers, their orders and help them easily find free tables or place orders. The services that can be provided are plenty which includes but is not limited to ordering food, making reservation for a table by the customer through the system online, managing customer and employee information and managing the menu information and reporting it. The web application of this system may be used to maintain restaurant database such as membership and menu maintenance, manage reservation, ordering food, making payment, kitchen maintenance and reporting.

The restaurant management system allows the owner of the restaurant to manage all the details of the employees without much effort. It reduces the pen paper work that one needs to do. It will be very useful for the restaurants having more number of employees. It can also store the details of the available time and money for managing the restaurants well. It can give some suggestions to manage in managing the restaurants well and how to run it smoothly and efficiently. In some restaurants where there are many employees, the higher authorities will not know about the job roles that the person is working for. This will help in knowing the job roles of the varied persons. One will be able to feed in the number of employees working and also the job roles that one is working for.

## ENTITIES AND ATTRIBUTES

An ER diagram shows the relationship among entities. An entity set is a group of similar entities and these entities have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.

## I. ENTITIES

An entity is an object that exists. In database administration, an entity can be a single thing, person, place, or object. Data can be stored about such entities. In database administration, only those things about which data will be captured or stored are considered an entity.

### *THE MAIN ENTITIES OF THE RESTURANT MANAGEMENT SYSTEM*

- ❖ Customer
- ❖ Order
- ❖ Food Items
- ❖ Employee

## II. ATTRIBUTES

An attribute defines the information about the entity that needs to be stored. An entity will have zero or more attributes, and each of those attributes apply only to that entity. Attributes also have further refinements, such as domain and key. The domain of an entity describes the possible values of attributes. In the entity, each attribute will have only one value, which could be blank or it could be a number, text, a date, or a time. The key is the unique identifier that identifies the entity. A key is also a domain because it will have values. These values are unique to each record, and so it's a special type of domain. A key isn't always required but is good to have as it makes keeping record easier.

### *RESTURANT MANAGEMENT SYSTEM ENTITIES WITH THEIR ATTRIBUTES*

**Attributes of Food Items Entity are:**

**(WEAK ENTITY)**

- ❖ f\_name                      Food Item's Name
- ❖ f\_price                      Food Item's Price

**Attributes of Customer Entity are:**

**(STRONG ENTITY)**

- ❖ c\_id                      Customer ID                      **(Primary Key)**
- ❖ c\_name                      Customer's Name                      **(Combined Attribute)**
- ❖ c\_sex                      Customer's Sex
- ❖ c\_mobile                      Customer's Mobile Numbers                      **(Multi-valued)**
- ❖ c\_add                      Customer's Address

**Attributes of Order Entity are:****(STRONG ENTITY)**

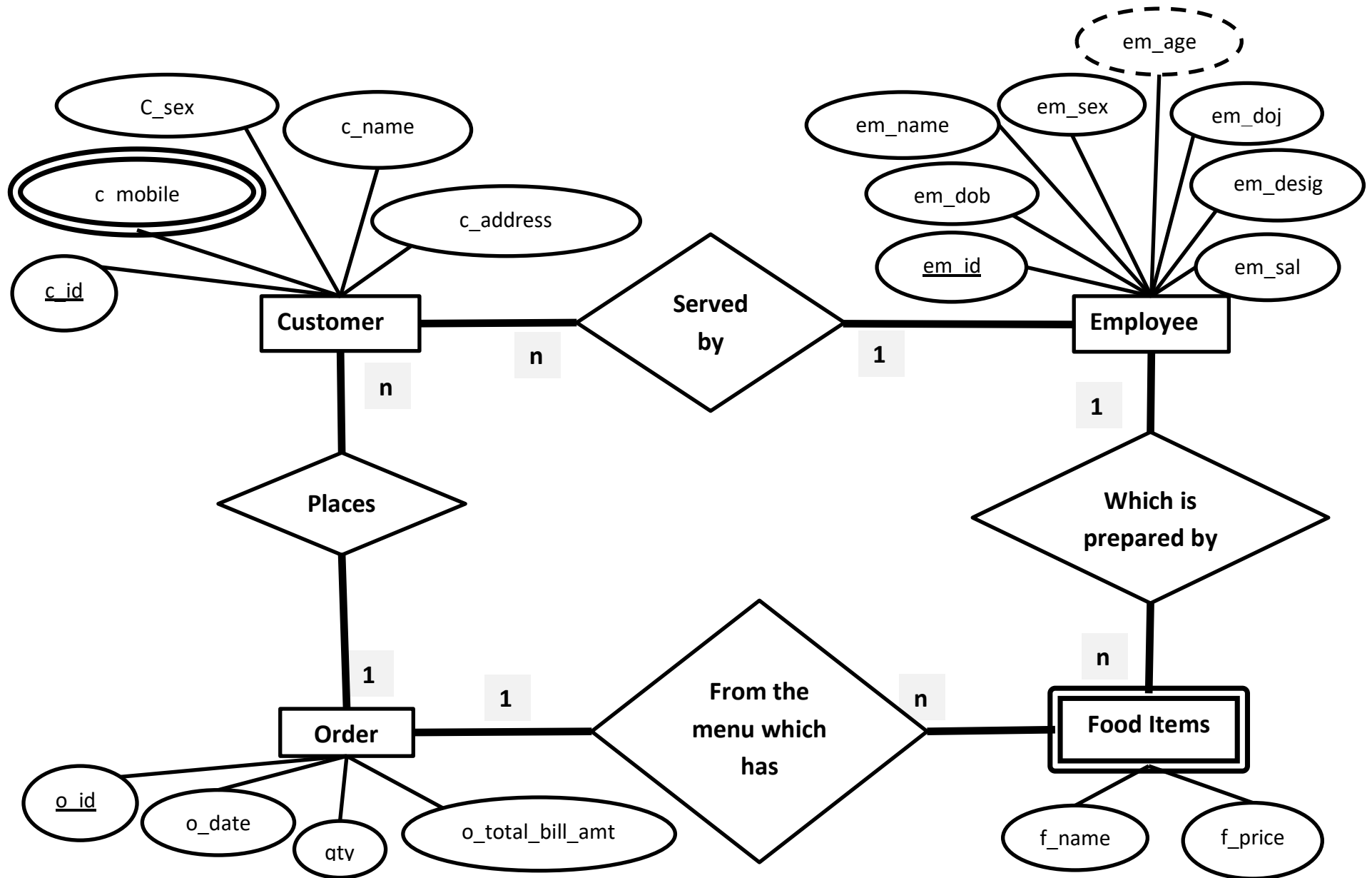
❖ o_id	Order ID	<b>(Primary Key)</b>
❖ o_date	Order's Date	
❖ o_total_bill	Order's Total Bill	
❖ qty	Order Quantity	
❖ c_id	Customer ID	<b>(Foreign Key)</b>
❖ em_id	Employee ID	<b>(Foreign Key)</b>

**Attributes of Employee Entity are:****(STRONG ENTITY)**

❖ em_id	Employee ID	<b>(Primary Key)</b>
❖ em_name	Employee's Name	
❖ em_mobile	Employee's Mobile Numbers	<b>(Multi-valued)</b>
❖ em_sex	Employee's Sex	
❖ em_dob	Employee's Date Of Birth	
❖ em_age	Employee's Age	<b>(Derived Attribute)</b>
❖ em_doj	Employee's Date Of Joining	
❖ em_deig	Employee's Designation	
❖ em_sal	Employee's Salary	

*Above 'c' stands for customer, 'o' stands for order, 'em' stands for employee and 'f' stands for food items.*

## E-R DIAGRAM FOR RESTURANT MANAGEMENT SYSTEM





# DATABASE TABLES

## I. CUSTOMERS

c_id	c_name	c_mobile	c_sex	c_add
1	Alex	6057477201	F	Berkeley Gardens London
2	Piper	7019339087	F	South House Queensbridge London
3	Eric	3078788895	M	Boulevard Charonne London
4	Mike	5165586300	M	Fauntleroy Circus London
5	Robert	6623185527	M	Rue Lauriston London

```
/* Create a table */
CREATE TABLE CUSTOMER(c_id integer(5) not null, c_name char(20) not null, c_mobile integer(15) not null,
c_sex char(2) not null, c_add char(50) not null);
/* Create few records in this table */
INSERT INTO CUSTOMER VALUES(1,'Alex','F','Berkeley Gardens London');
INSERT INTO CUSTOMER VALUES (2,'Piper','F','South House Queensbridge London');
INSERT INTO CUSTOMER VALUES (3,'Eric','M','Boulevard Charonne London');
INSERT INTO CUSTOMER VALUES (4,'Mike','M','Fauntleroy Circus London');
INSERT INTO CUSTOMER VALUES (5,'Robert','M','Rue Lauriston London');
/* Display all the records from the table */
SELECT c_id, c_name, c_mobile, c_sex, c_add FROM CUSTOMER;
```

### SQL Statement:

```
SELECT * FROM [CUSTOMER]
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL >

### Result:

Number of Records: 5

c_id	c_name	c_mobile	c_sex	c_add
1	Alex	6057477201	F	Berkeley Gardens London
2	Piper	7019339087	F	South House Queensbridge London
3	Eric	3078788895	M	Boulevard Charonne London
4	Mike	5165586300	M	Fauntleroy Circus London
5	Robert	6623185527	M	Rue Lauriston London

## II. EMPLOYEES

em_id	em_name	em_mobile	em_sex	em_dob	em_age	em_doj	em_desig	em_salary
1	Susan Brown	6757477201	F	02-04-1988	32	01-06-2010	Head Chef	5000
2	Jim Kern	328578951	M	24-06-1990	30	16-08-1999	General Manager	4500
3	Ellen Owens	5267786305	F	04-07-1995	25	19-02-2017	Counter Service	3000
4	Henry Perkins	6423687427	M	19-01-1993	27	19-03-2016	Waiter	1500
5	Fred Campbell	4621154539	M	17-03-1996	24	01-04-2016	Waiter	1500

```
/* Create a table */
CREATE TABLE WORKING_EMPLOYEES_DETAILS(em_id integer(5) not null, em_name char(20) not null,
em_mobile integer(15) not null, em_sex char(2) not null, em_dob char(10) not null, em_age integer(3) not null,
em_doj char(10) not null, em_desig char(50) not null, em_salary integer(5) not null);
/* Create few records in this table */
INSERT INTO WORKING_EMPLOYEES_DETAILS VALUES (1,'Susan Brown',6757477201,'F','02-04-1988',32,'01-06-
2010','Head Chef',5000);
INSERT INTO WORKING_EMPLOYEES_DETAILS VALUES (2,'Jim Kern',328578951,'M','24-06-1990',30,'16-08-
1999','General Manager',4500);
INSERT INTO WORKING_EMPLOYEES_DETAILS VALUES (3,'Ellen Owens',5267786305,'F','04-07-1995',25,'19-02-
2017','Counter Service',3000);
INSERT INTO WORKING_EMPLOYEES_DETAILS VALUES (4,'Henry Perkins',6423687427,'M','19-01-1993',27,'19-
03-2016','Waiter',1500);
INSERT INTO WORKING_EMPLOYEES_DETAILS VALUES (5,'Fred Campbell',4621154539,'M','17-03-1996',24,'01-
04-2016','Waiter',1500);
/* Display all the records from the table */
SELECT em_id, em_name, em_mobile, em_sex, em_dob, em_age, em_doj, em_desig, em_salary FROM
WORKING_EMPLOYEES_DETAILS;
```

SQL Statement:

```
SELECT * FROM [WORKING_EMPLOYEE_DETAILS]
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL >

Result:

Number of Records: 5

em_id	em_name	em_mobile	em_sex	em_dob	em_age	em_doj	em_desig	em_salary
1	Susan Brown	6757477201	F	02-04-1988	32	01-06-2010	Head Chef	5000
2	Jim Kern	328578951	M	24-06-1990	30	16-08-1999	General Manager	4500
3	Ellen Owens	5267786305	F	04-07-1995	25	19-02-2017	Counter Service	3000
4	Henry Perkins	6423687427	M	19-01-1993	27	19-03-2016	Waiter	1500
5	Fred Campbell	4621154539	M	17-03-1996	24	01-04-2016	Waiter	1500

### III. FOOD ITEMS

f_name	f_price
Fresh Salad	200
Soup Of The Day	150
Sandwich	350
Grilled Steak	600
Cheeseburger	250
Broccoli	100
French Fries	150
Soda	125
Coffee	85

```
/* Create a table */
```

```
CREATE TABLE FOOD_ITEMS (f_name char(20) not null, f_price integer(10) not null);
```

```
/* Create few records in this table */
```

```
INSERT INTO FOOD_ITEMS VALUES ('Fresh Salad', 200);
```

```
INSERT INTO FOOD_ITEMS VALUES ('Soup Of The Day', 150);
```

```
INSERT INTO FOOD_ITEMS VALUES ('Sandwich', 350);
```

```
INSERT INTO FOOD_ITEMS VALUES ('Grilled Steak', 600);
```

```
INSERT INTO FOOD_ITEMS VALUES ('Cheeseburger', 250);
```

```
INSERT INTO FOOD_ITEMS VALUES ('Broccoli', 100);
```

```
INSERT INTO FOOD_ITEMS VALUES ('French Fries', 150);
```

```
INSERT INTO FOOD_ITEMS VALUES ('Soda', 125);
```

```
INSERT INTO FOOD_ITEMS VALUES ('Coffee', 85);
```

```
/* Display all the records from the table */
```

```
SELECT f_name ,f_price FROM FOOD_ITEMS;
```

SQL Statement:

```
SELECT * FROM [FOOD_ITEMS]
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL

Result:

Number of records: 9

f_name	f_price
Fresh Salad	200
Soup Of The Day	150
Sandwich	350
Grilled Steak	600
Cheeseburger	250
Broccoli	100
French Fries	150
Soda	125
Coffee	85

## IV. ORDERS

or_id	or_date	f_name	qty	or_total_bill_amt	c_id	em_id
1	7-11-20	Cheeseburger	4	1000	3	5
2	8-11-20	French Fries	2	300	4	1
3	9-11-20	Coffee	6	510	1	3
4	10-11-20	Fresh Salad	1	200	5	2
5	11-11-20	Soup Of The Day	2	300	2	4
6	12-11-20	Cheeseburger	3	750	3	5
7	13-11-20	Soda	5	625	5	3
8	14-11-20	Grilled Steak	1	600	1	4

```
/* Create a table */
```

```
CREATE TABLE RES_ORDERS (or_id integer(5) not null, or_date char(10) not null, f_name char(20) not null, qty integer(5) not null, or_total_bill_amt integer(50) not null, c_id integer(5) not null, em_id integer(5) not null);
```

```
/* Create few records in this table */
```

```
INSERT INTO RES_ORDERS VALUES (1,'7-11-20','Cheeseburger', 4, 1000, 3, 5);
```

```
INSERT INTO RES_ORDERS VALUES (2,'8-11-20','French Fries', 2, 300, 4, 1);
```

```
INSERT INTO RES_ORDERS VALUES (3,'9-11-20','Coffee', 6, 510, 1, 3);
```

```
INSERT INTO RES_ORDERS VALUES (4,'10-11-20','Fresh Salad', 1, 200, 5, 2);
```

```
INSERT INTO RES_ORDERS VALUES (5,'11-11-20','Soup Of The Day', 2, 300, 2, 4);
```

```
INSERT INTO RES_ORDERS VALUES (6,'12-11-20','Cheeseburger', 3, 750, 3, 5);
```

```
INSERT INTO RES_ORDERS VALUES (7,'13-11-20','Soda', 5, 625, 5, 3);
```

```
INSERT INTO RES_ORDERS VALUES (8,'14-11-20','Grilled Steak', 1, 600, 1, 4);
```

```
/* Display all the records from the table */
```

```
SELECT or_id, or_date, f_name, qty, or_total_bill_amt, c_id, em_id FROM RES_ORDERS;
```

SQL Statement:						
SELECT * FROM [RES_ORDERS]						
Edit the SQL Statement, and click "Run SQL" to see the result.						
Run SQL >						
Result:						
Number of Records: 8						
or_id	or_date	f_name	qty	or_total_bill_amt	c_id	em_id
1	7-11-20	Cheeseburger	4	1000	3	5
2	8-11-20	French Fries	2	300	4	1
3	9-11-20	Coffee	6	510	1	3
4	10-11-20	Fresh Salad	1	200	5	2
5	11-11-20	Soup Of The Day	2	300	2	4
6	12-11-20	Cheeseburger	3	750	3	5
7	13-11-20	Soda	5	625	5	3
8	14-11-20	Grilled Steak	1	600	1	4

## QUERIES AND RESULTS

### I. DISPLAY THE NAME, ADDRESS AND GENDER OF ALL CUSTOMERS WHO ARE MALE

SQL Statement:

```
SELECT c_name, c_add, c_sex
FROM [CUSTOMER]
WHERE c_sex = 'M';
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL >

Result:

Number of Records: 3

c_name	c_add	c_sex
Eric	Boulevard Charonne London	M
Nike	Pavilion Circus London	M
Robert	Rue Lauriston London	M

```
SELECT c_name, c_add, c_sex
FROM [CUSTOMER]
WHERE c_sex = 'M';
```

### II. DISPLAY THE NAME, CONTACT NUMBER AND GENDER OF ALL FEMALE CUSTOMERS

SQL Statement:

```
SELECT c_name, c_mobile, c_sex
FROM [CUSTOMER]
WHERE c_sex = 'F';
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL >

Result:

Number of Records: 2

c_name	c_mobile	c_sex
Alex	6037477201	F
Pper	7019339087	F

```
SELECT c_name, c_mobile, c_sex
FROM [CUSTOMER]
WHERE c_sex = 'F';
```

### III. DISPLAY THE NAME, AGE, GENDER, MOBILE NUMBER, DESIGNATION AND SALARY OF EMPLOYEES HAVING A SALARY GREATER THAN EQUAL TO 3000

SQL Statement:

```
SELECT em_name,em_age, em_sex, em_mobile, em_desig, em_salary
FROM [WORKING_EMPLOYEE_DETAILS]
GROUP BY em_salary
HAVING em_salary >=3000;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL

Result:

Number of Records: 3

em_name	em_age	em_sex	em_mobile	em_desig	em_salary
Ellen Owens	25	F	5347786305	Counter Service	3000
Jim Kern	30	M	328578951	General Manager	4500
Susan Brown	32	F	6757477201	Head Chef	9000

```
SELECT em_name,em_age, em_sex, em_mobile, em_desig, em_salary
FROM [WORKING_EMPLOYEE_DETAILS]
GROUP BY em_salary
HAVING em_salary >=3000;
```

### IV. COUNT THE TOTAL NUMBER OF EMPLOYEES

SQL Statement:

```
SELECT COUNT(em_name)
FROM [WORKING_EMPLOYEE_DETAILS];
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL

Result:

Number of Records: 1

COUNT(em_name)
5

```
SELECT COUNT(em_name)
FROM [WORKING_EMPLOYEE_DETAILS] ;
```

## V. DISPLAY ALL FOOD ITEMS WITH PRICE WHEN THE PRICE IS GREATER THAN 150

SQL Statement:

```
SELECT f_name,f_price
FROM [FOOD_ITEMS]
WHERE f_price > 150;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL

Result:

Number of Records: 4

f_name	f_price
Fresh Salad	200
Sandwich	300
Grilled Steak	600
Cheeseburger	250

```
SELECT f_name,f_price
FROM [FOOD_ITEMS]
WHERE f_price > 150;
```

## VI. DISPLAY ORDER ID, ORDERING DATE, NAME OF THE ITEM ORDERED WITH ITS QUANTITY AND THE TOTAL BILL AMOUNT WHERE THE BILL AMOUNT IS GREATER THAN RS.500

SQL Statement:

```
SELECT or_id,or_date,f_name,qty,or_total_bill_amt
FROM [RES_ORDERS]
WHERE or_total_bill_amt > 500;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL

Result:

Number of Records: 5

or_id	or_date	f_name	qty	or_total_bill_amt
1	7-11-20	Cheeseburger	4	1000
3	9-11-20	Coffee	6	510
6	12-11-20	Cheeseburger	3	750
7	13-11-20	Soda	5	625
8	14-11-20	Grilled Steak	1	600

```
SELECT or_id,or_date,f_name,qty,or_total_bill_amt
FROM [RES_ORDERS]
WHERE or_total_bill_amt > 500;
```

## VII. UPDATE THE PRICE OF SANDWICH IN FOOD ITEMS TO RS. 300

SQL Statement:

```
UPDATE [food_items]
SET f_price = 300
WHERE f_name = 'Sandwich';
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL

Result:

Number of Records: 9

f_name	f_price
Fresh Salad	200
Soup Of The Day	150
Sandwich	300
Grilled Steak	600
Cheeseburger	250
Broccoli	100
French Fries	150
Soft	125
Coffee	85

```
UPDATE [FOOD_ITEMS]
SET f_price = 300
WHERE f_name = 'Sandwich';
```

## VIII. DISPLAY THE COUNT OF ALL EMPLOYEES GROUPED BY THEIR GENDER

SQL Statement:

```
SELECT COUNT(em_sex), em_sex
FROM [WORKING_EMPLOYEE_DETAILS]
GROUP BY em_sex;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL

Result:

Number of Records: 2

COUNT(em_sex)	em_sex
2	F
3	M

```
SELECT COUNT(em_sex), em_sex
FROM [WORKING_EMPLOYEE_DETAILS]
GROUP BY em_sex;
```



## IX. DISPLAY THE ORDER ID AND THE CUSTOMER'S NAME WHO PLACED THAT ORDER

SQL Statement:

```
SELECT RES_ORDERS.or_id, CUSTOMER.c_name
FROM RES_ORDERS
INNER JOIN CUSTOMER
ON RES_ORDERS.c_id = CUSTOMER.c_id;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL. >

Result:

Number of Records: 8

or_id	c_name
1	Eric
2	Mike
3	Alex
4	Robert
5	Piper
6	Eric
7	Robert
8	Alex

```
SELECT RES_ORDERS.or_id, CUSTOMER.c_name
FROM RES_ORDERS
INNER JOIN CUSTOMER
ON RES_ORDERS.c_id = CUSTOMER.c_id;
```

## X. DISPLAY THE CUSTOMER ID, ORDERING DATE, CUSTOMER'S NAME, MOBILE, ITEMS ORDERED WITH THEIR QUANTITY AND TOTAL BILL AMOUNT

SQL Statement:

```
SELECT RES_ORDERS.c_id, or_date, c_name, c_mobile, f_name, qty, or_total_bill_amt
FROM [RES_ORDERS], [CUSTOMER]
WHERE RES_ORDERS.c_id = CUSTOMER.c_id;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL. >

Result:

Number of Records: 8

c_id	or_date	c_name	c_mobile	f_name	qty	or_total_bill_amt
3	7-11-20	Eric	3078788895	Cheeseburger	4	1000
4	8-11-20	Mike	5185586300	French Fries	2	300
1	9-11-20	Alex	6057477201	Coffee	6	510
8	10-11-20	Robert	862188527	Fresh Salad	1	200
2	11-11-20	Piper	7019339087	Soup Of The Day	2	300
3	12-11-20	Eric	3078788895	Cheeseburger	3	750
8	13-11-20	Robert	862188527	Soda	5	625
1	14-11-20	Alex	6057477201	Grilled Steak	1	600

```
SELECT RES_ORDERS.c_id, or_date, c_name, c_mobile, f_name, qty, or_total_bill_amt
FROM [RES_ORDERS], [CUSTOMER]
WHERE RES_ORDERS.c_id = CUSTOMER.c_id;
```