Mini-Project Report

on

Restaurant Management System

By

Hiteshree Panchal **Roll number:27**

Riddhi Pandya **Roll number:28**

Neha Parepalli **Roll number: 30**

Anuja Patel **Roll number:31**

TE EXTC B

Course: DBMS

AY: 2019-20

Table of Contents

I. Storyline

1. This restaurant management system is for a small-scale restaurant with only one location.

2. The restaurant has the record of each employee working in the restaurant:
   2.1 The employees have different role such as manager, waiter, chef and employee.
   2.2 Each manager manages the employees.
   2.3 Employee's hire-date is also mentioned to calculate the no. of years of a particular employee has worked in the restaurant.

3. The database must keep track of the menu used by the restaurant:
   3.1 Each item has an item-no. and a name.
   3.2 The price of the items ordered by a customer.
   3.3 The amount of quantity ordered by each customer.

4. The restaurant to keep a record of its customers:
   4.1 Every customer has an ID, Name, Address, phn number.
   4.2 We keep track of the customer's order.
   4.3 We also keep track of each waiter serving to each customer.

5. The restaurant has an inventory which oversees the all requirement raw materials:
   5.1 We keep track of the ID, name and quantity-left and last-date-of-entry of each raw materials available in the inventory.

6. The restaurant needs to keep a record of all the bills:
   6.1 Every customer must pay bills after his/her meal.
   6.2 If a customer is not yet completed his/her meal, then his/her bill cannot be made.
   6.3 Each customer is given an option to pay the bill with google-pay, debit, credit, Paytm or cash.
   6.4 The restaurant charges every customer as per their orders. Hence, the amount depends on the total sum of the prices of their ordered items.

II. Components of Database Design

To effectively design a database, we need to have a clear understanding of the entity sets and the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

In this mini project, a database management system for a small-scale hospital is being made.

The entities and their respective attributes required are as follows:

1) **Employees** – The restaurant needs a huge work force of competent and efficient staff to ensure the effective management of the restaurant. The restaurant must keep track of each employee's ID, name, phone number and hire-date.

Attributes:

- Employee ID (number) e.g. 5683223
- Employee Name (varchar) e.g. Vikrant
- Phone number (number) e.g. 9999988888
- Hire-Date(date) e.g. 6/6/2015

The restaurant hires these employees and differentiates them into 3 categories based on their work:

a) **Manager** – The manager are the employees who manages and instructs waiters. The restaurant needs to store, for every manager, a personal number and contact number where they can be reached in case of emergency.

b) Chef – The chef are the members of the work staff whose job is to cook the meal and look after the customer's requirements as per their desired cuisine.

c) Waiter – They are the employees who manage the day-to-day activities of the restaurant. They are an integral part of the efficient management of the restaurant affairs.

d) Employee – They are the employees who assist chefs and also assist waiters.

2) **Customer–** The customers are the people who come to the restaurant to eat delicious meal as per their desire. The restaurant needs to store customer ID, Name, phone number, address.

Attributes:

- Customer ID (number) e.g. 5683223
- Name (varchar) e.g. Meghan
- Address (varchar) e.g. 14 pq rd, Kandivali
- Phone number (number) e.g. 9989998989

3) **Inventory**– The restaurant needs a variety of raw materials to prepare the items as per the customer. They need to store the details of every inventory such as inventory number, name, quantity left, last date of entry. They also need to keep a record of when a particular inventory was bought.

Attributes:

- Inventory no. (number) e.g. 27
- Instrument Item name (varchar) e.g. Tomato
- Quantity left (number) e.g. – 4 kg
- Last date of Entry (date) e.g. 2020-02-15

4) **Menu** – The restaurant has the menu list where the manager can get the details of the food items. The restaurant needs to keep a record of the details of each item like ID, name, price and quantity. The restaurant needs to also keep a record of the items ordered by each customer.

Attributes:

- Item-no (number) e.g. 43
- Item name (varchar) e.g. Shahi paneer
- Price (number) e.g. 343
- Quantity (number) e.g. 2

5) **Bill**– Every customer need to pay bills after his or her meal. The restaurant needs to keep records of the bill ID, amount, method of payment.

Attributes:

- Bill ID (number) e.g. 99
- Amount (number) e.g. Rs 650
- Method of payment e.g. Google-pay

Now that we are done describing the all the entities in detail, we need to examine the **relationships** between each entity. But before we can examine the relationships, we need to be aware of a concept called cardinality.

A **cardinality** notation defines the attributes of the relationship between the entities. Cardinalities can denote that an entity is optional (for example, a sales rep could have no customers or could have many) or mandatory (for example, there must be at least one product listed in an order.)

The three main cardinalities are:

1. A one-to-one relationship (1:1). For example, if each customer in a database is associated with one mailing address.
2. A one-to-many relationship (1:M). For example, a single customer might place an order for multiple products. The customer is associated with multiple entities, but all those entities have a single connection back to the same customer.
3. A many-to-many relationship (M: N). For example, at a company where all call center agents work with multiple customers, each agent is associated with multiple customers, and multiple customers might also be associated with multiple agents.

Relationships And Cardinality

1) The entities **Customer** and **Employee(waiter)** are connected by a relation called **service**, since employees serves to customer. Each customer is served by one specific employee(waiter). An employee(waiter) can serve various customer. Hence it is a **one to many relationships**. A waiter cannot exist in the hotel without serving customer. Hence there is **total participation** on customer side.

2) The entities **Manager** and **Employee** are connected by a relation called **manages**. Each manager manages many employees. Hence it is a **one to many relationships**. There is **partial participation** since both entities can exist without each other.

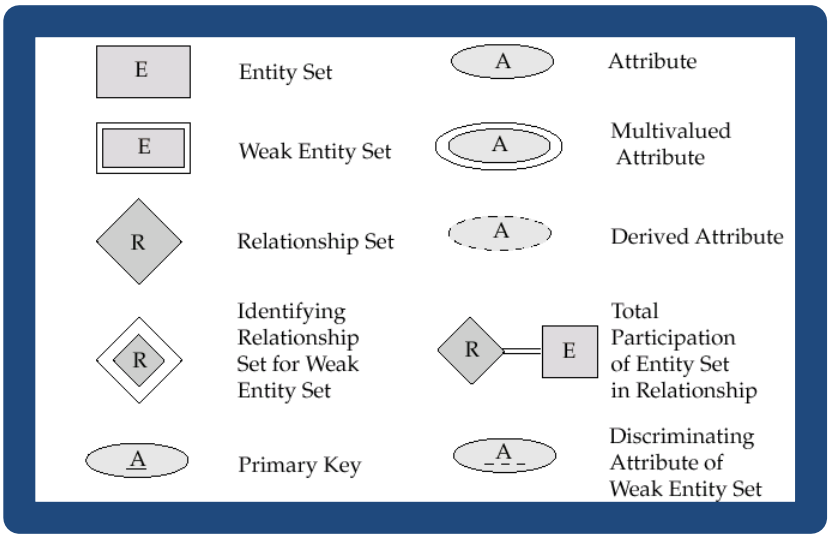
3) The entities **Customer** and **Bill** are connected by a relation called **payment** since a customer pays a bill. Each customer pays one bill. One bill is paid by one customer. Hence it is a **one to one relationship**. A customer cannot pay money if he there is no bill. Hence, there is **total participation** on Bill.

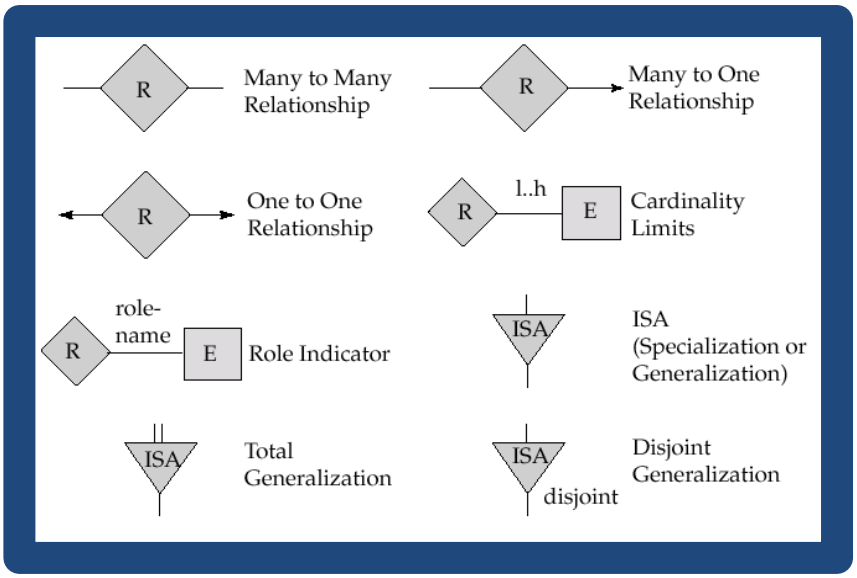
4) The entities **Menu** and **Customer** are connected by a relation **order.** Each customer orders many items and each item are ordered by many customers. Its cardinality is **many to many**. Since for existence of menu, a customer is not required but for a customer to have a meal, a menu is required. Therefore, **Menu** has partial participation and **Customer** has total participation.

5) The entities **Menu** and **Inventory** are connected by a relation called **supply**. 1 menu can have many inventory items and 1 inventory can have many menu items hence it is a **many to many relationships**. There is **partial participation** since both entities cannot exist without each other.

III. Entity Relationship Diagram

Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. Before starting the creation of ER Diagram, we should know these conventional symbols used in it.





Organizing the ERD in a logical way is incredibly important to increase comprehension. The main purpose of entity-relationship diagrams is to model a complex database, so learning how to create simple, logical ERDs is the key.

Entity-relationship diagrams are incredibly useful, and the following simple steps are needed to create an ER diagram:

1) **Determine the entities** – The entities for this ER Diagram are Employee, Customer, Inventory, Menu, Bill.
   Here Employee will be the higher-level entity and Manager, Waiter, Chef and Employees will be lower level entities. There is an 'is a' relationship. Hence, we use specialization to divide the employees into Manager, Waiter, Chef and Employee. All attributes common to all 3 lower-level entities will become attributes of employee.

2) **Next, we need to determine the primary key for each entity**

The primary key for employee, customer, inventory, menu, bill is emp-id, cust-id, inv-no, item-no, bill-no respectively.

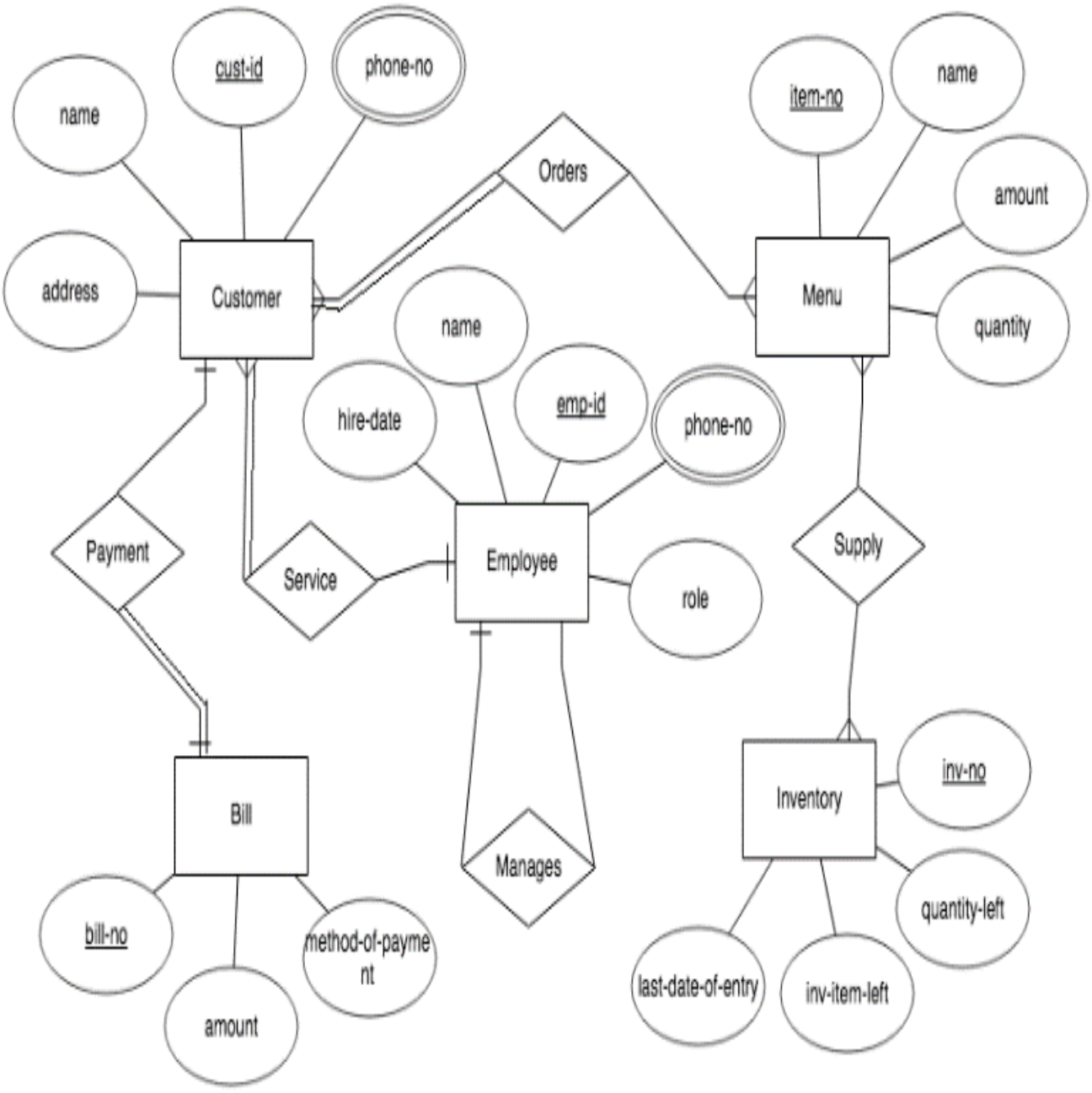
3) **Identify the relationships: Relationships highlight how entities interact with each other.**

This step was already done under the topic relationships and cardinality. According to the cardinality, we make the arrows when connecting two entities.

4) **Add attributes: Attributes show specific characteristics of an entity, refining what information is important to the model.**

Add the attributes of all the entities and finalize the ER Diagram.

ER Diagram


cust-id
phone-no
name
Orders
item-no
name
amount
address
Customer
Menu
quantity
name
hire-date
emp-id
phone-no
Payment
Service
Employee
role
Supply
Bill
Manages
Inventory
inv-no
quantity-left
bill-no
method-of-payment
amount
last-date-of-entry
inv-item-left

IV. Schema

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It is the database designers who design the schema to help programmers understand the database and make it useful.

The schema depends upon the entities, their attributes and the relations between those entities and the cardinality of the relations between the entities.

We make a table for each entity and then examine the relationships between entities.

Specialization

In this project Employee is specialized into entities Manager, Waiter and Chefs. Hence, we make a separate table for these subclass entities and include all attributes of that subclass entity set and attributes of the superclass entity set Employee.

For one-to-one relationship with one entity set having total participation

Augment one extra column on the right side of the table of the entity set with total participation, put in there the primary key of the entity set without complete participation as per to the relationship.

In this project Customer and Bill have a one to one relationship with total participation on the billing side. Hence, we used the above-mentioned steps to add PID as a foreign key in billing.

For one-to-many relationship without total participation

Same thing as one-to-one

This relationship is between employee (manager) to employee where both are partial. In this one to many relationships the primary key of one i.e. manager is foreign key to the many side i.e. employee.

For one-to-many/many-to-one relationship with one entity set having total participation on "many" side

Augment one extra column on the right side of the table of the entity set on the "many" sides, put in there the primary key of the entity set on the "one" side as per to the relationship.

In this project employee and customer have a one to many relationships with total participation on many sides. Hence, we add emp-id as a foreign key to mentioned entity.

For many-to-many relationship

Same thing as one-to-one relationship without total participation. Primary key of this new schema is the union of the foreign keys of each entity sets.

In this project, menu with customer and menu inventory have a many to many relationships. Hence, we make a new relation uses which will have item-no and cust-id as foreign keys from menu and customer respectively. We make a new relation supply which will have item-no and inv-no as foreign keys from menu and inventory respectively.

Schema

employee (emp-id, name, phone-no, role, hire-date, manager-id*)

bill (bill-no, amount, method-of-payment, customer-id*)

customer (customer-id, name, phn-no, address, emp-id*)

menu (item-no, name, amount, quantity)

inventory (inv-no, quantity-left, inv-item-name, last-date-of-entry)

orders (item-no*, cust-id*)

supply (item-no*, inv-no*)

V. SQL Queries

Based on the relational schema, create the database and tables in MySQL. Populate the tables with some suitable values.

Write at least 15 SQL queries covering most of the concepts studied in the class for Module 5.

1) **Show the bill details of the customers who has their amount greater than Rs.300**

```
mysql> select *
    -> from bill
    -> where amount > 300;
+---------+--------+--------------------+-------------+
| bill_no | amount | method_of_payment  | customer_id |
+---------+--------+--------------------+-------------+
|       2 |    941 | debit              |     6734826 |
|       4 |    387 | google_pay         |     6003358 |
|       5 |    929 | google_pay         |     6160390 |
|       7 |    681 | google_pay         |     7238390 |
|       8 |    369 | paytm              |     6615632 |
|       9 |    762 | debit              |     6500977 |
|      10 |    638 | credit             |     5989537 |
+---------+--------+--------------------+-------------+
7 rows in set (0.04 sec)
```

2) Insert new item-name Pav Bhaji with amount Rs 200 and quantity 2.

```
mysql> insert into menu values('64','Pav Bhaji','200','2');
Query OK, 1 row affected (0.16 sec)

mysql> select *
    -> from menu;
+---------+-------------+--------+----------+
| item_no | item_name   | amount | quantity |
+---------+-------------+--------+----------+
|       3 | Manchurian  | 863.21 |        3 |
|       8 | Dum_Aloo    | 809.49 |        2 |
|       9 | Fried_Rice  | 848.65 |        4 |
|      15 | Sizzler     | 542.79 |        4 |
|      17 | Burger      | 490.06 |        3 |
|      19 | Pizza       | 160.95 |        5 |
|      20 | French_Fries| 844.09 |        9 |
|      34 | Pasta       | 415.90 |        8 |
|      36 | Sandwich    | 240.27 |        1 |
|      40 | Shahi_Paneer| 343.28 |        2 |
|      64 | Pav Bhaji   | 200.00 |        2 |
+---------+-------------+--------+----------+
11 rows in set (0.00 sec)
```

3) Change the price of Pizza to Rs 200.

```
mysql> update menu
    -> set amount=200
    -> where item_name="Pizza";
Query OK, 1 row affected (0.18 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

BEFORE:

```
mysql> select * from menu;
+---------+-------------+--------+----------+
| item_no | item_name   | amount | quantity |
+---------+-------------+--------+----------+
|       3 | Manchurian  | 863.21 |        3 |
|       8 | Dum_Aloo    | 809.49 |        2 |
|       9 | Fried_Rice  | 848.65 |        4 |
|      15 | Sizzler     | 542.79 |        4 |
|      17 | Burger      | 490.06 |        3 |
|      19 | Pizza       | 160.95 |        5 |
|      20 | French_Fries| 844.09 |        9 |
|      34 | Pasta       | 415.90 |        8 |
|      36 | Sandwich    | 240.27 |        1 |
|      40 | Shahi_Paneer| 343.28 |        2 |
+---------+-------------+--------+----------+
10 rows in set (0.00 sec)
```

AFTER:

```
mysql> select * from menu;
+---------+--------------+--------+----------+
| item_no | item_name    | amount | quantity |
+---------+--------------+--------+----------+
|       3 | Manchurian   | 863.21 |        3 |
|       8 | Dum_Aloo     | 809.49 |        2 |
|       9 | Fried_Rice   | 848.65 |        4 |
|      15 | Sizzler      | 542.79 |        4 |
|      17 | Burger       | 490.06 |        3 |
|      19 | Pizza        | 200.00 |        5 |
|      20 | French_Fries | 844.09 |        9 |
|      34 | Pasta        | 415.90 |        8 |
|      36 | Sandwich     | 240.27 |        1 |
|      40 | Shahi_Paneer | 343.28 |        2 |
|      64 | Pav Bhaji    | 200.00 |        2 |
+---------+--------------+--------+----------+
11 rows in set (0.00 sec)
```

4) Show all the customers names whose bill amount is less than Rs 500.

```
mysql> select distinct customer_name
    -> from customer,bill
    -> where customer.customer_id=bill.customer_id
    -> and amount < 500
    -> order by customer_name;
+---------------+
| customer_name |
+---------------+
| Eve           |
| Gautam        |
| Jack          |
| Jill          |
| Shubham       |
+---------------+
5 rows in set (0.03 sec)
```

5) **Show all the customers names who has “a” letter in their name.**

```
mysql> select customer_name
    -> from customer
    -> where customer_name like '%a%';
+---------------+
| customer_name |
+---------------+
| Shubham       |
| Deepak        |
| Gautam        |
| Rakesh        |
| Jack          |
| Avijit        |
| Santosh       |
+---------------+
7 rows in set (0.05 sec)
```

6) **Find all the employees who are not chef.**

```
mysql> select emp_name
    -> from employee
    -> where role <> "chef";
+------------+
| emp_name   |
+------------+
| Hiteshree  |
| Rishi      |
| Rachit     |
| Riddhi     |
| Dhairya    |
| Neha       |
| Atharva    |
| Anuja      |
+------------+
8 rows in set (0.00 sec)
```

7) **Find the total number of items whose quantities are left.**

```
mysql> select count(quantity_left)
    -> from inventory;
+-----------------------+
| count(quantity_left) |
+-----------------------+
|                    10 |
+-----------------------+
1 row in set (0.04 sec)
```

8) **Find the names of the inventories that are being supplied.**

```
mysql> select inv_item_name
    -> from inventory as i,supply as s
    -> where i.inv_no=s.inv_no
    -> group by inv_item_name;
+---------------+
| inv_item_name |
+---------------+
| Potato        |
| Onion         |
| Spinach       |
| Butter        |
| Carrot        |
| Baby_Corn     |
| Bread         |
| Tomato        |
| Capsicum      |
| Cheese        |
+---------------+
10 rows in set (0.00 sec)
```

9) Find names of items where average amount is greater than Rs 500.

```
mysql> select item_name,avg(amount)
    -> from menu
    -> group by item_name
    -> having avg(amount) > 500;
+---------------+-------------+
| item_name     | avg(amount) |
+---------------+-------------+
| Manchurian    |  863.210000 |
| Dum_Aloo      |  809.490000 |
| Fried_Rice    |  848.650000 |
| Sizzler       |  542.790000 |
| French_Fries |  844.090000 |
+---------------+-------------+
5 rows in set (0.00 sec)
```

10) Find customers who are being served by Neha.

```
mysql> select customer_name
    -> from customer as c,employee as e
    -> where emp_name="Neha" and
    -> e.emp_id=c.emp_id;
+---------------+
| customer_name |
+---------------+
| Santosh       |
+---------------+
1 row in set (0.00 sec)
```

11) Show the name of the customer whose residence is at F-328_VAjira_Andheri and paid his bill by Google Pay.

```
mysql> select customer_name
    -> from customer as c
    -> where c.address="F-328_Vajira_Andheri" and
    -> c.customer_id in (select customer_id
    -> from bill
    -> where method_of_payment="google_pay");
+---------------+
| customer_name |
+---------------+
| Santosh       |
+---------------+
1 row in set (0.00 sec)
```

12) Delete the records of Onion from restaurant.

```
mysql> delete from supply
    -> where inv_no in
    -> (select inv_no
    -> from inventory
    -> where inv_item_name="Onion");
Query OK, 1 row affected (0.19 sec)

mysql> delete from inventory
    -> where inv_item_name="Onion";
Query OK, 1 row affected (0.16 sec)
```

BEFORE:-

```
mysql> select * from supply;
+----------+---------+
| item_no  | inv_no  |
+----------+---------+
|        3 |      60 |
|        8 |       4 |
|        9 |      11 |
|       15 |      27 |
|       17 |      14 |
|       19 |      26 |
|       20 |      30 |
|       34 |       5 |
|       36 |      10 |
|       40 |      17 |
+----------+---------+
10 rows in set (0.00 sec)
```

```
mysql> select * from inventory;
+-------+--------------+--------------+--------------------+
| inv_no | quantity_left | inv_item_name | last_date_of_entry |
+-------+--------------+--------------+--------------------+
|      4 |           35 | Potato        | 2020-03-08         |
|      5 |            6 | Onion         | 2020-02-11         |
|     10 |            7 | Spinach       | 2020-03-06         |
|     11 |           24 | Butter        | 2020-01-15         |
|     14 |           15 | Carrot        | 2020-03-18         |
|     17 |           20 | Baby_Corn     | 2020-03-14         |
|     26 |           17 | Bread         | 2020-02-06         |
|     27 |            4 | Tomato        | 2020-02-15         |
|     30 |           40 | Capsicum      | 2020-03-18         |
|     60 |           34 | Cheese        | 2020-02-29         |
+-------+--------------+--------------+--------------------+
10 rows in set (0.00 sec)
```

AFTER:-

```
mysql> select * from supply;
+---------+--------+
| item_no | inv_no |
+---------+--------+
|       3 |     60 |
|       8 |      4 |
|       9 |     11 |
|      15 |     27 |
|      17 |     14 |
|      19 |     26 |
|      20 |     30 |
|      36 |     10 |
|      40 |     17 |
+---------+--------+
9 rows in set (0.03 sec)
```

```
mysql> select * from inventory;
+--------+---------------+--------------+--------------------+
| inv_no | quantity_left | inv_item_name | last_date_of_entry |
+--------+---------------+--------------+--------------------+
|      4 |            35 | Potato       | 2020-03-08         |
|     10 |             7 | Spinach      | 2020-03-06         |
|     11 |            24 | Butter       | 2020-01-15         |
|     14 |            15 | Carrot       | 2020-03-18         |
|     17 |            20 | Baby_Corn    | 2020-03-14         |
|     26 |            17 | Bread        | 2020-02-06         |
|     27 |             4 | Tomato       | 2020-02-15         |
|     30 |            40 | Capsicum     | 2020-03-18         |
|     60 |            34 | Cheese       | 2020-02-29         |
+--------+---------------+--------------+--------------------+
9 rows in set (0.04 sec)
```

13) **Find the inventory brought before 2020-02-27.**

```
mysql> select inv_item_name
    -> from inventory
    -> where last_date_of_entry < "2020-02-27";
+---------------+
| inv_item_name |
+---------------+
| Butter        |
| Bread         |
| Tomato        |
+---------------+
3 rows in set (0.01 sec)
```

14) **Find names of the employees who are not manager and arrange them in descending order.**

```
mysql> select emp_name
    -> from employee
    -> where manager_id="null"
    -> order by emp_name desc;
+------------+
| emp_name   |
+------------+
| Vikrant    |
| Rishi      |
| Riddhi     |
| Rachit     |
| Neha       |
| Hiteshree  |
| Diksha     |
| Dhairya    |
+------------+
8 rows in set (0.00 sec)
```

15)Update the phone number of Eve who lives at A-32_Yam_Road_Dadar to 1234567890.

```
mysql> update customer
    -> set phn_no=1234567890
    -> where customer_name="Eve" and
    -> address="A-32_Yam_Road_Dadar";
Query OK, 1 row affected (0.23 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

BEFORE:

```
mysql> select * from customer;
+-------------+---------------+------------+--------------------------+---------+
| customer_id | customer_name | phn_no     | address                  | emp_id  |
+-------------+---------------+------------+--------------------------+---------+
|     5816967 | Shubham       | 9941434328 | A-32_Yam_Road_Dadar      | 5846757 |
|     5989537 | Deepak        | 9940917415 | F-328_Vajira_Andheri     | 6404075 |
|     6003358 | Gautam        | 9938208823 | E-65_Bhatia_Road_Dahisar | 5762513 |
|     6160390 | Eve           | 9902055540 | E-20_Swami_Road_Virar    | 5980172 |
|     6310256 | Eve           | 9950906733 | A-32_Yam_Road_Dadar      | 6604861 |
|     6500977 | Rakesh        | 9920769189 | A-57_Ambadi_road_Vasai   | 6474086 |
|     6615632 | Jill          | 9950235956 | A-520_Balaji_road_Bandra | 7150565 |
|     6645127 | Jack          | 9893392469 | B-43_Navghar_Road_Parle  | 5683223 |
|     6734826 | Avijit        | 9903028079 | E-65_Bhatia_Road_Dahisar | 7156878 |
|     7238390 | Santosh       | 9949380158 | F-328_Vajira_Andheri     | 6704120 |
+-------------+---------------+------------+--------------------------+---------+
10 rows in set (0.00 sec)
```

AFTER:

```
mysql> select * from customer;
+-------------+---------------+------------+--------------------------+---------+
| customer_id | customer_name | phn_no     | address                  | emp_id  |
+-------------+---------------+------------+--------------------------+---------+
|     5816967 | Shubham       | 9941434328 | A-32_Yam_Road_Dadar      | 5846757 |
|     5989537 | Deepak        | 9940917415 | F-328_Vajira_Andheri     | 6404075 |
|     6003358 | Gautam        | 9938208823 | E-65_Bhatia_Road_Dahisar | 5762513 |
|     6160390 | Eve           | 9902055540 | E-20_Swami_Road_Virar    | 5980172 |
|     6310256 | Eve           | 1234567890 | A-32_Yam_Road_Dadar      | 6604861 |
|     6500977 | Rakesh        | 9920769189 | A-57_Ambadi_road_Vasai   | 6474086 |
|     6615632 | Jill          | 9950235956 | A-520_Balaji_road_Bandra | 7150565 |
|     6645127 | Jack          | 9893392469 | B-43_Navghar_Road_Parle  | 5683223 |
|     6734826 | Avijit        | 9903028079 | E-65_Bhatia_Road_Dahisar | 7156878 |
|     7238390 | Santosh       | 9949380158 | F-328_Vajira_Andheri     | 6704120 |
+-------------+---------------+------------+--------------------------+---------+
10 rows in set (0.00 sec)
```