University of Asia Pacific

Department of Computer Science & Engineering

Course Code: CSE - 212

Course Title: Database Systems Lab

Project Report

Project Name: **Restaurant Management System.**

Submitted By:

Name:	**Student ID:**
Maisha Sameha	22201266
Shuhurat Fariha	22201252

Section: E-2

Semester: 2nd year 2nd semester

Submitted To:

Alif Ruslan, Lecturer

Department Of CSE

University of Asia Pacific.

Table of Contents

#Task - 1

#Task – 2

#Task – 3

#Task – 4

1.Project Description

Project Name: Restaurant Database Management System.

Description: Our project aims to provide a dependable and user-friendly database management solution designed for restaurant operations. This system will organize and handle key data, such as customer information, menu items and categories, orders and payment information, table bookings, and staff roles and responsibilities. It will also handle inventory management to guarantee that food service operations run smoothly and without shortages. Using past data, the database will allow for real-time order tracking, better reservation management, and individualized customer support. By centralizing these features, the system hopes to improve the restaurant's operational efficiency, eliminate errors, and improve customer dining experience. This project will provide a robust technology backbone for restaurant enterprises, allowing them to achieve modern service requirements while remaining profitable.

2.Database Name

The major goal of the restaurant_management_db is to properly manage data linked to restaurant operations, delivering a consistent dining experience for guests while optimizing staff workflows and business procedures. The database's name appropriately represents its objective, which is to manage and organize many aspects of restaurant operations, such as staff, menu, orders, bookings, and payments. Its structure allows for complete understanding of essential entities such as customers, waiters, kitchen staff, food items, and payment processes, ensuring the

restaurant can provide high-quality service while remaining operationally efficient.

3.Entities/Tables

There are 10 tables in this database -

1. Waiter Table
2. KitchenStaff Table
3. Customer Table
4. TableOrder Table
5. Reservation Table
6. Menu Table
7. FoodItems Table
8. OrderTable
9. OrderItems Table
10. Payment Table

All the tables hold very important information about the enterprise.

4.Table Columns/Attributes

1. Waiter (WaiterId, WaiterName, PhoneNo, Salary, Age)

2. KitchenStaff (StaffId, StaffName, Age, Salary, ContactNo)

3. Customer (CustomerId, CustomerName, Age, Email, ContactNo)

4. TableOrder (TableNo, SeatCapacity)

5. Reservation (reserveId, Customer_id, TableNo, NoOfGuests)

6. Menu (menuId, foodType)

7. FoodItems (menuId, foodId, foodName, foodSize, quantity, price)

8. Order (orderId, StaffId, CustomerId, WaiterId)

9. OrderItems (orderNo, orderId, foodId)

10. Payment (paymentId, orderId, totalBill, totalAmount, method)

5.Primary key, Foreign key, Relations & Description

1. Waiter:

a. WaiterId (Primary Key)

b. WaiterName

c. PhoneNo

d. Salary

e. Age

This table is created to manage information and personal details about the restaurant's waiters who serves a particular order. The attribute 'WaiterId' is the primary key. There are no foreign keys.

2. KitchenStaff:

a. StaffId (Primary Key)

b. StaffName

c. Age

d. Salary

e. ContactNo

This table is created to manage the personal details of kitchen staff responsible for preparing food according to the order. The attribute 'StaffId' is the primary key. There are no foreign keys.

3. Customer:

a. CustomerId (Primary Key)

b. CustomerName

c. Age

d. Email

e. ContactNo

This table is created to store information about customers who gives order. The attribute 'CustomerId' is the primary key. There are no foreign keys.

4. TableOrder:

a. TableNo (Primary Key)

b. SeatCapacity

This table is created to manage table information in the restaurant which helps manage seating arrangements and maximize space efficiency. The attribute 'TableNo' is the primary key. There are no foreign keys.

5. Reservation:

a. **ReserveId (Primary Key)**

b. **CustomerId(Foreign key Referencing Customer(CustomerId))**

c. **TableNo(Foreign key Referencing tableOrder(TableNo))**

d. **NoOfGuests**

This table is created to manage reservations made by customers which is linked to TableOrder in order to track which table is reserved. The attribute 'ReserveId' is the primary key. The attributes 'CustomerId' & 'tableNo' are foreign keys.

6. Menu:

a. **menuId (Primary Key)**

b. **foodType**

This table is created to classify food into categories such as salads, main courses, desserts, etc. The attribute 'menuId' is the primary key. There is no foreign key.

7. FoodItems:

a. **menuId(Foreign Key Referencing Menu(menuId))**

b. **FoodId(Primary Key)**

c. **foodName**

d. **foodSize**

e. **quantity**

f. **price**

This table is created to list all the food items available in the restaurant. This is linked to Menu table to categorized food items and also linked to OrderItems table to track which food items are ordered. The attribute 'foodId' is the primary key. The attribute 'menuId' is the foreign key.

8. Order:

a. orderId (Primary Key)

b. CustomerId (Foreign Key Referencing Customer(CustomerId))

c. StaffId (Foreign Key Referencing KitchenStaff(StaffId))

d. WaiterId (Foreign Key Referencing Waiter(WaiterId))

This table is created to manage customer orders. This is a Central table linking customers, waiters, and kitchen staff. The attribute 'orderId' is the primary key. The attributes 'CustomerId', 'StaffId' & Waiterid are foreign keys.

9. OrderItems:

a. orderNo(Primary Key)

b. orderId (Foreign Key Referencing OrderTable(orderId))

c. foodId (Foreign Key Referencing FoodItems(foodId))

This table is created to detail the food items in each order. This table connects OrderTable and FoodItems Table to break down orders into individual items. The attribute 'orderNo' is the primary key. The attributes 'orderId' & foodId are foreign keys.

10. Payment:

a. paymentId (Primary Key)

b. orderId (Foreign Key Referencing Ordertable(orderId))

c. totalBill

d. totalAmount

e. method

This table is created to manage payment details for orders. This table connects OrderTable to track payments for orders of customers. The attribute 'paymentId' is the primary key. The attribute 'orderId' is the foreign key.

6.Relationships Between Each Entities

Explaining entity relationships between multiple tables in a database management system (DBMS) involves describing how entities (tables) are related to each other through their attributes and keys. There are several types of relationships, including one-to-one, one-to-many, and many-to-many. There are also several kinds of entities: strong & weak entities. They Also can participate in different types of way total participation or partial participation. The Relations between all the tables are –

1. Entity name: Order (strong entity) & Payment (strong entity)

Relationship: An order is ***paid*** *for* by a payment. Each order is associated with exactly one payment, and each payment corresponds to a single order.

Relationship type: One to One

2. Entity name: Reservation (strong entity) & TableOrder (strong entity)

Relationship: A reservation is ***for*** a specific table. Each reservation is linked to a specific table with specific seating capacity and each table order can have only one reservation.

Relationship type: One to One

3. Entity name: Menu (strong entity) & FoodItems (strong entity)

Relationship: A menu ***contains*** multiple food items. One menu contains multiple food items, but each food item belongs to only one menu.

Relationship type: One to Many

4. Entity name: Customer (strong entity) & Reservation (weak entity)

Relationship: A customer ***makes*** many reservations. A customer can make multiple reservations, but each reservation is linked to only one customer.

Relationship type: One to Many

5. Entity name: Order (strong entity) & OrderItems (weak entity)

Relationship: An order ***contains*** multiple order items. One order can contain multiple order items, but each order item belongs to a single order.

Relationship type: One to Many

6. Entity name: Waiter (strong entity) & Order (Strong entity)

Relationship: A waiter ***takes*** many orders. A single waiter can handle multiple orders. Each order, however, is assigned to only one specific waiter

Relationship type: One to Many

7. Entity name: OrderItems (weak entity) & FoodItems (strong entity)

Relationship: Many order items ***contain*** one Food item. Each order item is linked with a single food item, but many order items may be connected to the same food item.

Relationship type: Many to One

8. Entity name: Order (strong entity) & Customer (strong entity)

Relationship: A customer ***requests*** many orders. Many orders can be requested by the same customer, but each order is linked to one specific customer.

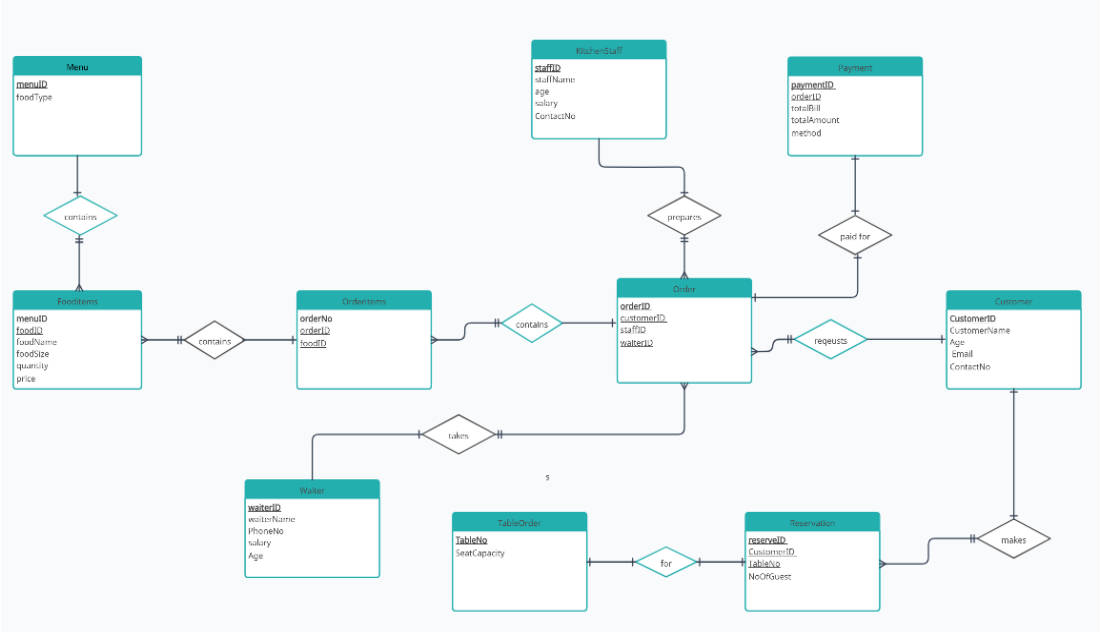
Relationship type: Many to One

9. Entity name: Order (strong entity) & KitchenStaff (strong entity)

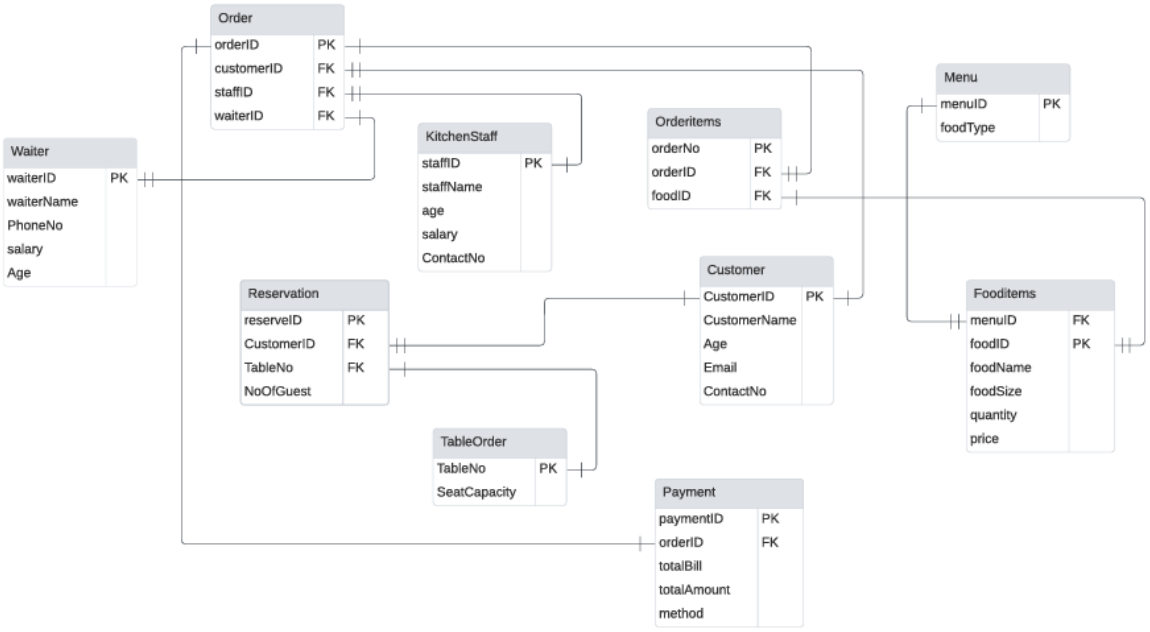
Relationship: Kitchen staff ***prepares*** multiple orders. Many orders can be prepared by one or more kitchen staff members and a single kitchen staff member can prepare multiple orders.

Relationship type: Many to One

7.ER Diagram



8.Schema Diagram



9.SQL Queries

Arithmetic operation based Queries

1.Annual salary of the waiters

```
select waiterID, waiterName, salary*12 as "Annual salary" from
 Waiter;
```

Result Grid | Filter Rows: | Export:

waiterID	waiterName	Annual salary
101	Raha	240000.00
102	Maya	216000.00
103	Raiyan	264000.00
104	Ihan	228000.00
105	Ruhul	252000.00
106	Tanvir	276000.00
107	Nadia	252000.00
108	Zahin	288000.00
109	Adib	240000.00
110	Samiha	264000.00
111	Nayeem	300000.00
112	Sara	234000.00
113	Raihan	252000.00
114	Ayesha	282000.00
115	Farhan	258000.00

2. Annual salary of the staffs

select staffID, staffName, salary*12 as "Annual salary" from Kitchenstaff;

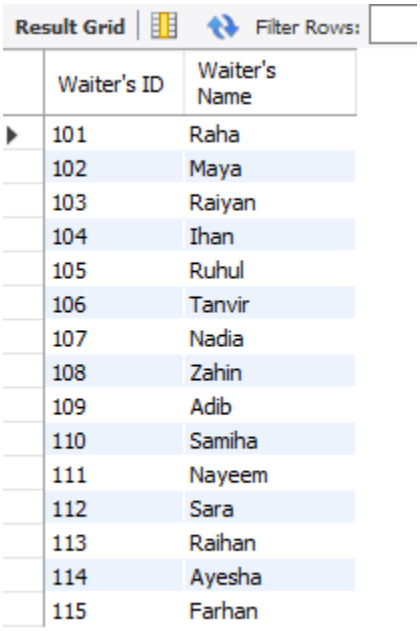
Result Grid | Filter Rows:

staffID	staffName	Annual salary
201	Ahmed	360000.00
202	Sara	384000.00
203	Hasan	336000.00
204	Mina	420000.00
205	Kamal	432000.00
206	Laila	348000.00
207	Rahim	312000.00
208	Zara	372000.00
209	Fariha	396000.00
210	Ihan	324000.00
211	Rima	384000.00
212	Tarek	420000.00
213	Nasrin	372000.00
214	Imran	336000.00
215	Arif	444000.00
216	Zainab	432000.00
217	Farid	372000.00
218	Anika	324000.00
219	Jamil	384000.00
220	Sakib	336000.00

String Function based Queries

3.Name and ID of the waiters

select waiterID "Waiter's ID",waiterName "Waiter's Name" from Waiter;



Result Grid | Filter Rows:

Waiter's ID	Waiter's Name
101	Raha
102	Maya
103	Raiyan
104	Ihan
105	Ruhul
106	Tanvir
107	Nadia
108	Zahin
109	Adib
110	Samiha
111	Nayeem
112	Sara
113	Raihan
114	Ayesha
115	Farhan

4. Name and ID of the staffs

select staffID "Staff's ID", staffName "Staff's Name" from Kitchenstaff;

Result Grid | Filter Rows:

Staff's ID	Staff's Name
201	Ahmed
202	Sara
203	Hasan
204	Mina
205	Kamal
206	Laila
207	Rahim
208	Zara
209	Fariha
210	Ihan
211	Rima
212	Tarek
213	Nasrin
214	Imran
215	Arif
216	Zainab
217	Farid
218	Anika
219	Jamil
220	Sakib

5.Staffs name that starts with ‘A’

```
select staffID "Staff's ID",staffName "Staff's Name" from
Kitchenstaff where staffName like "A%";
```

Result Grid | Filter Rows:

Staff's ID	Staff's Name
201	Ahmed
215	Arif
218	Anika

6.Waiter’s name that starts with ‘R’

```
select waiterID "Waiter's ID",waiterName "Waiter's Name" from
Waiter where waiterName like "R%";
```

Result Grid | Filter Rows:

Waiter's ID	Waiter's Name
101	Raha
103	Raiyan
105	Ruhul
113	Raihan

7.Food items in Uppercase

```
select foodID "food ID", Upper(foodName) "Dish Name" from
Fooditems ;
```

Result Grid | Filter Rows:

food ID	Dish Name
101	SPRING ROLLS
102	CHICKEN WINGS
103	BEEF CURRY
104	VEGETABLE BIRYANI
105	CHEESECAKE
106	CHOCOLATE BROWNIE
107	ICED TEA
108	LATTE
109	GREEK SALAD
110	FRUIT SALAD
111	CHICKEN SOUP
112	MINESTRONE SOUP
113	MOZZARELLA STICKS
114	STUFFED MUSHROOMS
115	MARGHERITA PIZZA
116	BBQ CHICKEN PIZZA
117	SPAGHETTI BOLOGNESE
118	FETTUCCINE ALFREDO
119	GRILLED LAMB CHOPS
120	GRILLED SALMON

8.Dish names with its’ prices

```
select concat (foodName ,'-' ,price) as "Dish Name - Dish Price"
from Fooditems ;
```

Result Grid | Filter Rows:

Dish Name - Dish Price
Spring Rolls-120.00
Chicken Wings-150.00
Beef Curry-350.00
Vegetable Biryani-280.00
Cheesecake-250.00
Chocolate Brownie-200.00
Iced Tea-100.00
Latte-150.00
Greek Salad-220.00
Fruit Salad-180.00
Chicken Soup-160.00
Minestrone Soup-190.00
Mozzarella Sticks-220.00
Stuffed Mushrooms-240.00
Margherita Pizza-400.00
BBQ Chicken Pizza-450.00
Spaghetti Bolognese-300.00
Fettuccine Alfredo-350.00
Grilled Lamb Chops-600.00
Grilled Salmon-550.00

Condition based Queries (WHERE, BETWEEN, IN)

9.staffs whose age is less than 35

SELECT * FROM KitchenStaff WHERE age < 35;

staffID	staffName	age	salary	ContactNo
203	Hasan	28	28000.00	+8801415636872
204	Mina	33	35000.00	+8801415668725
206	Laila	30	29000.00	+8801415668745
207	Rahim	25	26000.00	+8801412668732
210	Ihan	27	27000.00	+8801415624732
211	Rima	31	32000.00	+8801415661532
213	Nasrin	29	31000.00	+8801445668732
214	Imran	27	28000.00	+8801655668732
218	Anika	25	27000.00	+8801415458732
219	Jamil	30	32000.00	+8801415458732
220	Sakib	29	28000.00	+8801411268732
NULL	NULL	NULL	NULL	NULL

10. Customers with IDs between 301 and 310.

```
select * from Customer where customerID>= '301' AND
customerID<= '310';
```

Result Grid | Filter Rows: | Edit: | Export/Import:

CustomerID	CustomerName	Age	Email	ContactNo
301	Ara	30	Ara@example.com	+8801825664772
302	Zamia	25	Zamia@example.com	+8801812564772
303	Raisa	35	Raisa@example.com	+8801815664472
304	Zaima	28	Zaima@example.com	+8801845664772
305	Maya	32	Maya@example.com	+8801815664852
306	Mredula	29	Mredula@example.com	+8801415664772
307	Mina	26	Mina@example.com	+8801815654772
308	Mia	31	mia@example.com	+8801812464772
309	Emon	33	Emon@example.com	+8801812664772
310	Sneha	27	Sneha@example.com	+8801814564772
NULL	NULL	NULL	NULL	NULL

11. Food items priced between 150 and 300.

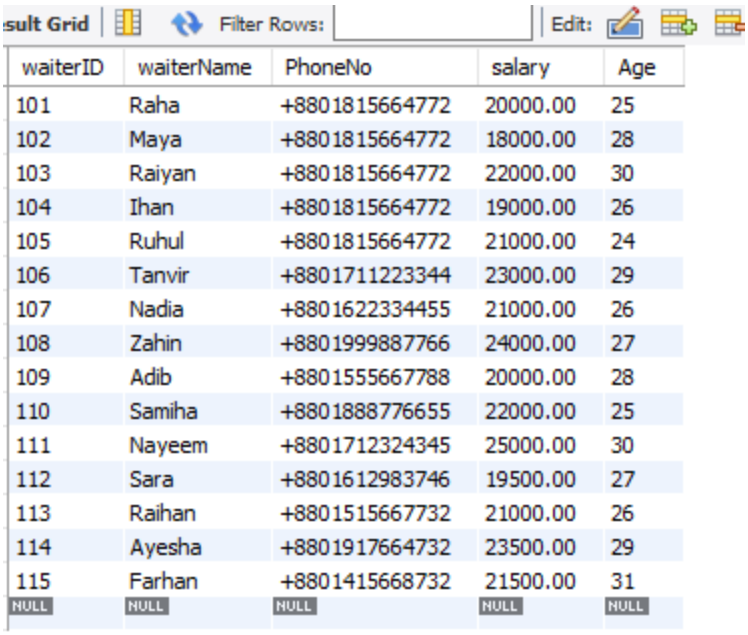
```
select * from Fooditems where price>= '150' AND price<= '300' ;
```

esult Grid | Filter Rows: | Edit: | E

menuID	foodID	foodName	foodSize	quantity	price
1	102	Chicken Wings	Medium	40	150.00
2	104	Vegetable Biryani	Medium	20	280.00
3	105	Cheesecake	Medium	25	250.00
3	106	Chocolate Brownie	Small	35	200.00
4	108	Latte	Medium	30	150.00
5	109	Greek Salad	Large	15	220.00
5	110	Fruit Salad	Small	20	180.00
6	111	Chicken Soup	Medium	25	160.00
6	112	Minestrone Soup	Large	18	190.00
7	113	Mozzarella Sticks	Small	50	220.00
7	114	Stuffed Mushrooms	Medium	35	240.00
9	117	Spaghetti Bolognese	Medium	30	300.00
NULL	NULL	NULL	NULL	NULL	NULL

12. Waiters with salaries between 15,000 and 30,000.

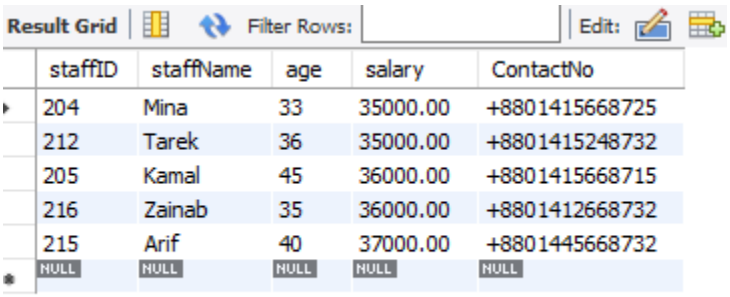
```
select * from Waiter where salary>= '15000' AND salary<=
'30000';
```



waiterID	waiterName	PhoneNo	salary	Age
101	Raha	+8801815664772	20000.00	25
102	Maya	+8801815664772	18000.00	28
103	Raiyan	+8801815664772	22000.00	30
104	Ihan	+8801815664772	19000.00	26
105	Ruhul	+8801815664772	21000.00	24
106	Tanvir	+8801711223344	23000.00	29
107	Nadia	+8801622334455	21000.00	26
108	Zahin	+8801999887766	24000.00	27
109	Adib	+8801555667788	20000.00	28
110	Samiha	+8801888776655	22000.00	25
111	Nayeem	+8801712324345	25000.00	30
112	Sara	+8801612983746	19500.00	27
113	Raihan	+8801515667732	21000.00	26
114	Ayesha	+8801917664732	23500.00	29
115	Farhan	+8801415668732	21500.00	31
NULL	NULL	NULL	NULL	NULL

13. Kitchen staff earning greater than or equal 35,000

select * from Kitchenstaff where salary>= '35000' order by salary;



staffID	staffName	age	salary	ContactNo
204	Mina	33	35000.00	+8801415668725
212	Tarek	36	35000.00	+8801415248732
205	Kamal	45	36000.00	+8801415668715
216	Zainab	35	36000.00	+8801412668732
215	Arif	40	37000.00	+8801445668732
NULL	NULL	NULL	NULL	NULL

14. Waiters with IDs between 101 and 105.

select * from Waiter where waiterID>= '101' && waiterID<= '105';

Result Grid | Filter Rows: | Edit:

waiterID	waiterName	PhoneNo	salary	Age
101	Raha	+8801815664772	20000.00	25
102	Maya	+8801815664772	18000.00	28
103	Raiyan	+8801815664772	22000.00	30
104	Ihan	+8801815664772	19000.00	26
105	Ruhul	+8801815664772	21000.00	24
NULL	NULL	NULL	NULL	NULL

15. Waiters with IDs 107, 108, or 110.

select * from Waiter where waiterID IN (107,108,110);

Result Grid | Filter Rows: | Edit:

waiterID	waiterName	PhoneNo	salary	Age
107	Nadia	+8801622334455	21000.00	26
108	Zahin	+8801999887766	24000.00	27
110	Samiha	+8801888776655	22000.00	25
NULL	NULL	NULL	NULL	NULL

16. Payments with total bills greater than 500

SELECT PaymentID,totalBill from Payment HAVING totalBill>500 ;

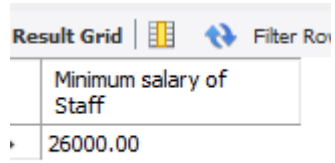
Result Grid | Filter Rows:

PaymentID	totalBill
3	570.00
4	720.00
10	1250.00
13	700.00
14	720.00
15	550.00

Aggregate Function-Based Queries:

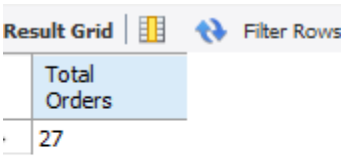
17. Minimum salary of kitchen staff.

select min(salary) "Minimum salary of Staff" from Kitchenstaff;



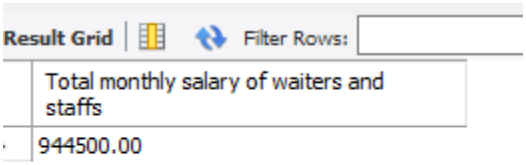
18. Total orders

select count(orderno) as "Total Orders" from Orderitems ;



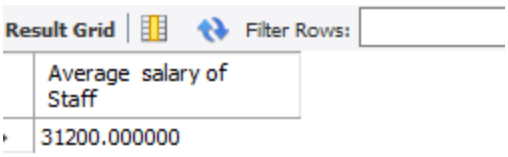
19. Monthly salary of all waiters and kitchen staff

select (select sum(salary) from Waiter)
+ (select sum(salary) from KitchenStaff) "Total monthly salary of waiters and staffs";



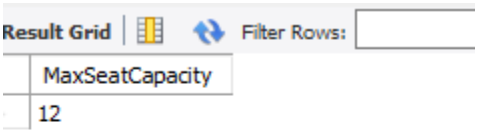
20. Average salary of kitchen staff.

select AVG(salary) "Average salary of Staff" from Kitchenstaff;



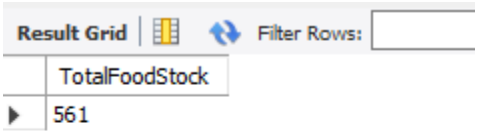
21. Maximum seating capacity among tables.

```
SELECT MAX(SeatCapacity) AS MaxSeatCapacity FROM
TableOrder;
```



22. Total stock of all food items.

```
SELECT SUM(quantity) AS TotalFoodStock FROM FoodItems;
```



Group By and Order By Queries:

23. Food items priced between 200 and 400, sorted by price.

select foodID,foodName,price from Fooditems where price>= '200' AND price<= '400' order by price ;

Result Grid | Filter Rows:

foodID	foodName	price
106	Chocolate Brownie	200.00
109	Greek Salad	220.00
113	Mozzarella Sticks	220.00
114	Stuffed Mushrooms	240.00
105	Cheesecake	250.00
104	Vegetable Biryani	280.00
117	Spaghetti Bolognese	300.00
103	Beef Curry	350.00
118	Fettuccine Alfredo	350.00
115	Margherita Pizza	400.00
NULL	NULL	NULL

24. Total orders per customer

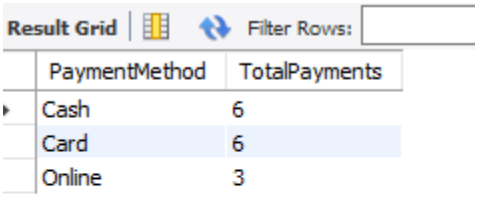
select orderID,count(orderno) "Total Orders from individual customers" from Orderitems group by orderID order by count(orderno) asc;

Result Grid | Filter Rows: | Export:

orderID	Total Orders from individual customers
5	1
6	1
8	1
9	1
11	1
12	1
2	2
3	2
7	2
13	2
14	2
15	2
1	3
4	3
10	3

25. Number of payments done by different payment methods.

```
select
method AS PaymentMethod,
 COUNT(*) AS TotalPayments
FROM Payment
GROUP BY method;
```

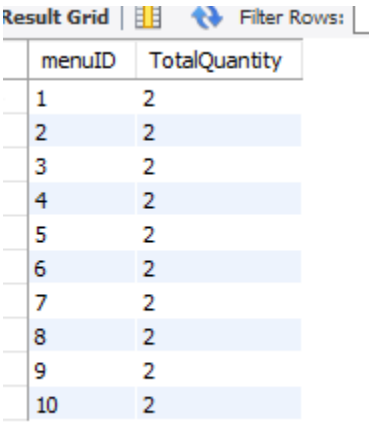


Result Grid | Filter Rows:

PaymentMethod	TotalPayments
Cash	6
Card	6
Online	3

27. Number of food items of the menu types.

```
SELECT menuID, COUNT(*) AS TotalQuantity FROM FoodItems GROUP
BY menuID;
```



Result Grid | Filter Rows:

menuID	TotalQuantity
1	2
2	2
3	2
4	2
5	2
6	2
7	2
8	2
9	2
10	2

Join Queries:

28. Food types and their corresponding dishes.

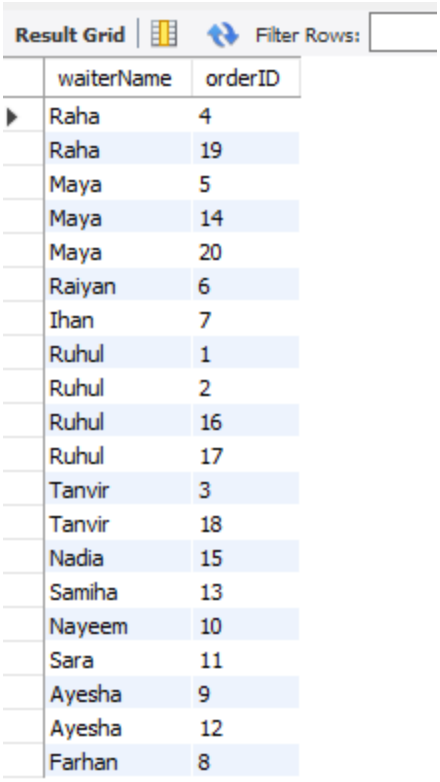
SELECT Menu.foodType,FoodItems.foodName FROM FoodItems JOIN Menu ON FoodItems.menuID = Menu.menuID;

Result Grid | Filter Rows:

foodType	foodName
Starter	Spring Rolls
Starter	Chicken Wings
Main Course	Beef Curry
Main Course	Vegetable Biryani
Dessert	Cheesecake
Dessert	Chocolate Brownie
Beverage	Iced Tea
Beverage	Latte
Salad	Greek Salad
Salad	Fruit Salad
Soup	Chicken Soup
Soup	Minestrone Soup
Appetizer	Mozzarella Sticks
Appetizer	Stuffed Mushrooms
Pizza	Margherita Pizza
Pizza	BBQ Chicken Pizza
Pasta	Spaghetti Bolognese
Pasta	Fettuccine Alfredo
Grill	Grilled Lamb Chops
Grill	Grilled Salmon

29. The order IDs dealt by each waiters

SELECT waiterName,orderID FROM Waiter JOIN Order_ ON Waiter.waiterID = Order_.waiterID;

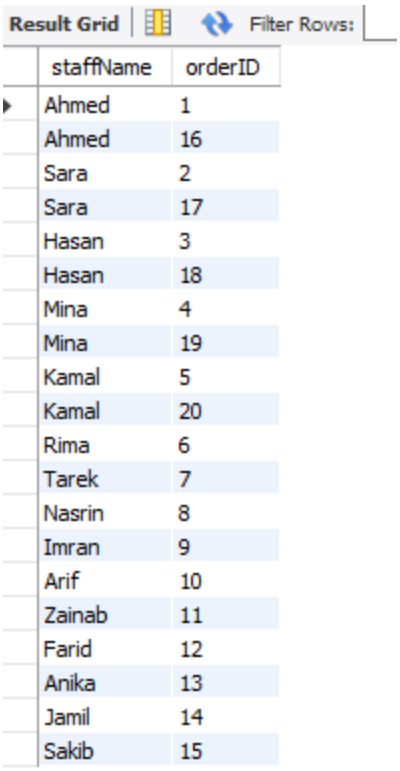


Result Grid | Filter Rows:

waiterName	orderID
Raha	4
Raha	19
Maya	5
Maya	14
Maya	20
Raiyan	6
Ihan	7
Ruhul	1
Ruhul	2
Ruhul	16
Ruhul	17
Tanvir	3
Tanvir	18
Nadia	15
Samiha	13
Nayeem	10
Sara	11
Ayesha	9
Ayesha	12
Farhan	8

30. The order IDs dealt by each Kitchen staffs

SELECT staffName,orderID FROM KitchenStaff JOIN Order_ ON KitchenStaff.staffID = Order_.staffID;



Result Grid | Filter Rows:

staffName	orderID
Ahmed	1
Ahmed	16
Sara	2
Sara	17
Hasan	3
Hasan	18
Mina	4
Mina	19
Kamal	5
Kamal	20
Rima	6
Tarek	7
Nasrin	8
Imran	9
Arif	10
Zainab	11
Farid	12
Anika	13
Jamil	14
Sakib	15

31. Customers with their table numbers and number of guests.

SELECT Customer.CustomerName, Reservation.TableNo, Reservation.NoOfGuest FROM Reservation JOIN Customer ON Reservation.CustomerID = Customer.CustomerID;

Result Grid | Filter Rows:

CustomerName	TableNo	NoOfGuest
Ara	1	4
Zamia	2	6
Raisa	3	8
Zaima	4	4
Zaima	5	4
Maya	21	10
Mredula	6	2
Mina	7	4
Mia	8	8
Emon	9	12
Sneha	10	6
Moni	11	5
Sadia	12	6
Raha	13	4
Jayma	14	10
Zarin	15	8
Ela	16	4
Kashfia	17	6
Luna	18	8
Shelly	19	2
Azad	20	12

Advanced Queries:

32. Customers with reservations for more than 6 guests.

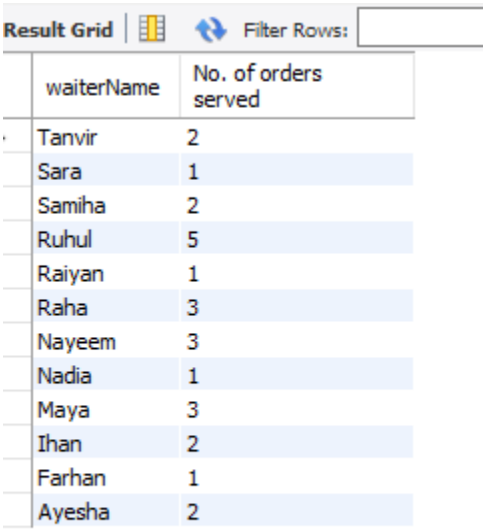
SELECT Customer.CustomerName, Reservation.NoOfGuest FROM Reservation JOIN Customer ON Reservation.CustomerID = Customer.CustomerID WHERE Reservation.NoOfGuest > 6;

Result Grid | Filter Rows:

CustomerName	NoOfGuest
Raisa	8
Mia	8
Emon	12
Jayma	10
Zarin	8
Luna	8
Azad	12
Maya	10

33. Amount of orders served by each waiter.

SELECT Waiter.waiterName, SUM(Payment.totalAmount) "No. of orders served" FROM Waiter JOIN Order_ ON Waiter.waiterID = Order_.waiterID JOIN Payment ON Order_.orderID = Payment.orderID GROUP BY Waiter.waiterName ORDER BY waiterName DESC;


Result Grid | Filter Rows:

waiterName	No. of orders served
Tanvir	2
Sara	1
Samiha	2
Ruhul	5
Raiyan	1
Raha	3
Nayeem	3
Nadia	1
Maya	3
Ihan	2
Farhan	1
Ayesha	2

34.Bill of each OrdersIDs

SELECT orderID, SUM(f.price) AS totalBill FROM OrderItems JOIN FoodItems f ON OrderItems.foodID = f.foodID GROUP BY orderID ORDER BY orderID;

Result Grid | Filter Rows:

orderID	totalBill
1	470.00
2	370.00
3	570.00
4	720.00
5	250.00
6	160.00
7	380.00
8	220.00
9	240.00
10	1250.00
11	450.00
12	300.00
13	700.00
14	720.00
15	1100.00

10.CEP Mapping

1. How Knowledge Profiles (K's) are addressed through the project and mapping among K's, COs and POs –

K's	Attributes	How K's are Addressed Through Our Project	CO	PO
K2	Mathematics	Conceptual mathematics and numerical analysis are utilized in calculating total order, total bills, total amount of foods, applying discounts efficiently.	CO2, CO3, CO4	PO1, PO3, PO6, PO7, PO8
K3	Engineering Fundamentals	Fundamental database principles are applied to design and operate the restaurant database, including the formulation of	CO1, CO2, CO3	PO1

		relationships, normalization, and integrity constraints to ensure proper data management.		
K5	Engineering Design	The project involves designing a structured ER diagram and schema diagram, effectively representing the data flow between entities like orders, customers, payments, and reservations.	CO3, CO4	PO3 PO5
K6	Engineering Practice	We've implemented our design in MySQL. We created tables, inserted data with DDL and ran queries with DML. We've used MySQL workbench software as a tool for implementing our design.	CO1, CO2, CO5	PO5, PO8

2. How Complex Engineering problem solving (P's) are addressed through the project and mapping among P's, COs and POs –

P's	**Attributes**	**How P's are Addressed Through Our Project**	**CO**	**PO**
P1	Depth of knowledge required	The project requires us to study one of the fundamentals of engineering which is DBMS design (K3). It also requires designing ER & Schema Diagram(K5) and implementing the designs (K6).	CO3, CO4, CO5	PO1, PO2, PO3, PO7
P3	Depth of analysis required	Functional queries and schema optimization are developed using abstract thinking and in-depth analysis to handle customer	CO3, CO5	PO3

		orders and payments quickly and accurately.		
P6	Extent of stakeholder involvement and conflicting requirements	The system is designed to accommodate the unique data requirements of stakeholders, such as customers, servers, and kitchen staff members, maintaining smooth cooperation and satisfaction.	CO3, CO5	PO2, PO7
P7	Interdependence	High levels of interdependence are displayed by components like OrderItems, Payment, and TableOrder; these interactions guarantee proper data connection and reporting.	CO3, CO4, CO5	PO2, PO7

3. How Complex Engineering Activities (A's) are addressed through the project and mapping among A's, COs and POs -

A's	**Attributes**	**How A's are Addressed Through Our Project**	**CO**	**PO**
A1	Range of resources	The project utilizes resources such as customer data, payment information, food orders, and staff records to manage restaurant operations effectively.	CO1, CO2 CO8	PO1, PO3, PO5, PO6, PO7, PO8
A2	Level of interaction	High interaction between stakeholders is modeled using relationships like	CO7 CO8 CO9	PO1 PO3 PO5 PO6

		Waiter taking orders, Customer making reservations, and KitchenStaff preparing food.		
A3	Innovation	Innovations include query optimization and database triggers to automate order updates, managing table reservation, and payment processes, ensuring real-time accuracy.	CO3 CO5 CO6 CO7	PO3 PO6 PO7 PO8
A5	Familiarity	The project builds on previously established database concepts, such as relational models and normalization, and applies them to the specific context of restaurant management. It addresses real-world challenges like minimizing data redundancy, resolving reservation conflicts, and ensuring accurate financial record-keeping, showcasing familiarity with the practical application of database systems.	CO5 CO6 CO7	PO10 PO11 PO12

11.Conclusion

The Restaurant Management Database System project successfully demonstrates the practical application of database design principles to solve real-world problems in the restaurant industry. By leveraging entity-relationship (ER) modeling, relational database concepts, and query optimization, the system effectively manages key operations such as menu management, order processing, staff allocation, table reservations, and payment handling. This project showcases how a well-structured database can improve operational efficiency, enhance customer satisfaction, and provide insights for informed decision-making. Through the use of SQL and advanced database techniques, the

system ensures data integrity, minimizes redundancy, and supports scalability to meet the demands of a growing business. The project provides a solid, approachable, and effective solution for modern restaurant management while highlighting the value of creativity and familiarity with standard engineering principles in solving challenging issues.