

```
-- CREATE DATABASE Restaurant;
```

```
-- USE Restaurant;
```

```
CREATE TABLE Waiter(  
waiterID int primary key,  
waiterName varchar(50),  
PhoneNo VARCHAR(15),  
salary DECIMAL(10,2),  
Age INT  
);
```

```
-- 2. KitchenStaff Table
```

```
CREATE TABLE KitchenStaff (  
staffID INT PRIMARY KEY,  
staffName VARCHAR(100),  
age INT,  
salary DECIMAL(10, 2),  
ContactNo VARCHAR(15)  
);
```

```
-- 3. Customer Table
```

```
CREATE TABLE Customer (  
CustomerID INT PRIMARY KEY,  
CustomerName VARCHAR(100),  
Age INT,  
Email VARCHAR(100),  
ContactNo VARCHAR(15)  
);
```

```
-- 4. TableOrder Table
```

```
CREATE TABLE TableOrder (  
    TableNo INT PRIMARY KEY,  
    SeatCapacity INT  
);
```

-- 5. Reservation Table

```
CREATE TABLE Reservation (  
    reserveID INT PRIMARY KEY,  
    CustomerID INT,  
    TableNo INT,  
    NoOfGuest INT,  
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),  
    FOREIGN KEY (TableNo) REFERENCES TableOrder(TableNo)  
);
```

-- 6. Menu Table

```
CREATE TABLE Menu (  
    menuID INT PRIMARY KEY,  
    foodType VARCHAR(100)  
);
```

-- 7. FoodItems Table

```
CREATE TABLE FoodItems (  
    menuID INT,  
    foodID INT PRIMARY KEY,  
    foodName VARCHAR(100),  
    foodSize VARCHAR(50),  
    quantity INT,  
    price DECIMAL(10, 2),
```

```
FOREIGN KEY (menuID) REFERENCES Menu(menuID)
);
-- 8. Order Table
CREATE TABLE Order_ (
    orderID INT PRIMARY KEY,
    customerID INT,
    staffID INT,
    waiterID INT,
    FOREIGN KEY (customerID) REFERENCES Customer(customerID),
    FOREIGN KEY (staffID) REFERENCES KitchenStaff(staffID),
    FOREIGN KEY (waiterID) REFERENCES Waiter(waiterID)
);
```

```
-- 9. OrderItems Table
CREATE TABLE OrderItems (
    orderNo INT,
    orderID INT,
    foodID INT,
    PRIMARY KEY (orderNo),
    FOREIGN KEY (orderID) REFERENCES Order_(orderID),
    FOREIGN KEY (foodID) REFERENCES FoodItems(foodID)
);
```

```
-- 10. Payment Table
CREATE TABLE Payment (
    paymentID INT PRIMARY KEY,
    orderID INT,
    totalBill DECIMAL(10, 2),
```

```
totalAmount INT,  
method VARCHAR(50),  
FOREIGN KEY (orderId) REFERENCES Order_(orderId)  
);
```

```
insert INTO Waiter values
```

```
(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);
```

```
INSERT INTO KitchenStaff VALUES
```

```
(201, 'Ahmed', 35, 30000.00, '+8801412568732'),  
(202, 'Sara', 40, 32000.00, '+8801415666332'),  
(203, 'Hasan', 28, 28000.00, '+8801415636872'),
```

(204, 'Mina', 33, 35000.00, '+8801415668725'),
(205, 'Kamal', 45, 36000.00, '+8801415668715'),
(206, 'Laila', 30, 29000.00, '+8801415668745'),
(207, 'Rahim', 25, 26000.00, '+8801412668732'),
(208, 'Zara', 38, 31000.00, '+8801415645732'),
(209, 'Fariha', 42, 33000.00, '+8801463668732'),
(210, 'Ihan', 27, 27000.00, '+8801415624732'),
(211, 'Rima', 31, 32000.00, '+8801415661532'),
(212, 'Tarek', 36, 35000.00, '+8801415248732'),
(213, 'Nasrin', 29, 31000.00, '+8801445668732'),
(214, 'Imran', 27, 28000.00, '+8801655668732'),
(215, 'Arif', 40, 37000.00, '+8801445668732'),
(216, 'Zainab', 35, 36000.00, '+8801412668732'),
(217, 'Farid', 38, 31000.00, '+8801445668732'),
(218, 'Anika', 25, 27000.00, '+8801415458732'),
(219, 'Jamil', 30, 32000.00, '+8801415458732'),
(220, 'Sakib', 29, 28000.00, '+8801411268732');

INSERT INTO Customer VALUES

(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'),
(311, 'Moni', 30, 'Moni@example.com', '+8801815454772'),
(312, 'Sadiah', 26, 'Sadiah@example.com', '+8801814264772'),
(313, 'Raha', 28, 'Raha@example.com', '+8801815664452'),
(314, 'Jayma', 32, 'Jayma@example.com', '+8801785664772'),
(315, 'Zarin', 29, 'Zarin@example.com', '+8801814564772'),
(316, 'Ela', 31, 'Ela@example.com', '+8801815669772'),
(317, 'Kashfia', 35, 'Kashfia@example.com', '+8801785664772'),
(318, 'Luna', 27, 'Luna@example.com', '+8801815424772'),
(319, 'Shelly', 36, 'Shelly@example.com', '+8801814564772'),
(320, 'Azad', 29, 'Azad@example.com', '+8801816364772');

INSERT INTO TableOrder VALUES

(1, 4),
(2, 6),
(3, 8),
(4, 4),
(5, 10),
(6, 2),
(7, 4),
(8, 8),

(9, 12),
(10, 6),
(11, 5),
(12, 6),
(13, 4),
(14, 10),
(15, 8),
(16, 4),
(17, 6),
(18, 8),
(19, 2),
(20, 12),
(21, 12);

INSERT INTO Reservation VALUES

(1, 301, 1, 4),
(2, 302, 2, 6),
(3, 303, 3, 8),
(4, 304, 4, 4),
(5, 304, 5, 4),
(21, 305, 21, 10),
(6, 306, 6, 2),
(7, 307, 7, 4),
(8, 308, 8, 8),
(9, 309, 9, 12),

(10, 310, 10, 6),
(11, 311, 11, 5),
(12, 312, 12, 6),
(13, 313, 13, 4),
(14, 314, 14, 10),
(15, 315, 15, 8),
(16, 316, 16, 4),
(17, 317, 17, 6),
(18, 318, 18, 8),
(19, 319, 19, 2),
(20, 320, 20, 12);

INSERT INTO Menu VALUES

(1, 'Starter'),
(2, 'Main Course'),
(3, 'Dessert'),
(4, 'Beverage'),

(5, 'Salad'),
(6, 'Soup'),
(7, 'Appetizer'),
(8, 'Pizza'),
(9, 'Pasta'),
(10, 'Grill'),
(11, 'Burger'),

(12, 'Seafood'),
(13, 'Vegan'),
(14, 'Juices');

INSERT INTO FoodItems (menuID, foodID, foodName, foodSize, quantity, price) VALUES

-- Starter

(1, 101, 'Spring Rolls', 'Small', 50, 120.00),
(1, 102, 'Chicken Wings', 'Medium', 40, 150.00),

-- Main Course

(2, 103, 'Beef Curry', 'Large', 30, 350.00),
(2, 104, 'Vegetable Biryani', 'Medium', 20, 280.00),

-- Dessert

(3, 105, 'Cheesecake', 'Medium', 25, 250.00),
(3, 106, 'Chocolate Brownie', 'Small', 35, 200.00),

-- Beverage

(4, 107, 'Iced Tea', 'Large', 40, 100.00),
(4, 108, 'Latte', 'Medium', 30, 150.00),

-- Salad

(5, 109, 'Greek Salad', 'Large', 15, 220.00),
(5, 110, 'Fruit Salad', 'Small', 20, 180.00),

-- Soup

(6, 111, 'Chicken Soup', 'Medium', 25, 160.00),
(6, 112, 'Minestrone Soup', 'Large', 18, 190.00),

-- Appetizer

```
(7, 113, 'Mozzarella Sticks', 'Small', 50, 220.00),  
(7, 114, 'Stuffed Mushrooms', 'Medium', 35, 240.00),  
-- Pizzas  
(8, 115, 'Margherita Pizza', 'Large', 20, 400.00),  
(8, 116, 'BBQ Chicken Pizza', 'Medium', 25, 450.00),  
-- Pasta  
(9, 117, 'Spaghetti Bolognese', 'Medium', 30, 300.00),  
(9, 118, 'Fettuccine Alfredo', 'Large', 20, 350.00),  
-- Grill  
(10, 119, 'Grilled Lamb Chops', 'Large', 15, 600.00),  
(10, 120, 'Grilled Salmon', 'Medium', 18, 550.00);
```

```
INSERT INTO Order_ VALUES
```

```
(1, 301, 201, 105),  
(2, 302, 202, 105),  
(3, 303, 203, 106),  
(4, 304, 204, 101),  
(5, 305, 205, 102),  
(16, 306, 201, 105),  
(17, 307, 202, 105),  
(18, 308, 203, 106),  
(19, 309, 204, 101),  
(20, 310, 205, 102),
```

(6, 311, 211, 103),
(7, 312, 212, 104),
(8, 313, 213, 115),
(9, 314, 214, 114),
(10,315, 215, 111),
(11, 316, 216, 112),
(12, 317, 217, 114),
(13, 318,218, 110),
(14, 319, 219, 102),
(15, 320, 220, 107);

INSERT INTO OrderItems VALUES

(1, 1, 101),
(16, 1, 105),
(17, 1, 107),
(2, 2, 113),
(18, 2, 102),
(20, 3, 113),
(21, 3, 103),
(4, 4, 104),
(22, 4, 104),
(23, 4, 111),
(5, 5, 105),

(6, 6, 111),
(7, 7, 112),
(24, 7, 112),
(25, 8, 113),
(9, 9, 114),
(10, 10, 115),
(26, 10, 115),
(27, 10, 116),
(11, 11, 116),
(12, 12, 117),
(13, 13, 118),
(28, 13, 118),
(14, 14, 119),
(29, 14, 101),
(15, 15, 120),
(3, 15, 120);

INSERT INTO Payment VALUES

(1, 1,470.00, 3, 'Cash'),
(2, 2, 370.00, 2, 'Card'),
(3, 3,570.00 ,2, 'Online'),
(4, 4,720.00 , 3, 'Cash'),
(5, 5, 250.00,1 , 'Card'),
(6, 6,160.00 , 1, 'Cash'),

```
(7, 7,380.00 , 2, 'Online'),  
(8, 8,220.00 , 1, 'Card'),  
(9, 9, 240.00 ,1, 'Cash'),  
(10, 10, 1250.00,3 , 'Card'),  
(11, 11, 450.00, 1, 'Cash'),  
(12, 12, 300.00, 1, 'Card'),  
(13, 13,700.00 ,2 , 'Online'),  
(14, 14, 720.00, 2, 'Cash'),  
(15, 15, 550.00, 1, 'Card');
```

-- Arithmetic operation based Queries

-- 1. Annual salary of the waiters

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

-- 2 . Annual salary of the staffs

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

-- String Function based Queries

-- 3. Name and ID of the waiters

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

-- 4. Name and ID of the staffs

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

-- 5. Staffs name that starts with 'A'

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

-- 6.Waiter's name that starts with 'R'

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

-- 7 .Food items in Uppercase

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

-- 8.Dish names with its' prices

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

-- Condition based Queries (WHERE, BETWEEN, IN)

-- 9.staffs whose age is less than 35

```
SELECT * FROM KitchenStaff WHERE age < 35;
```

-- 10. Customers with IDs between 301 and 310. select * from Customer where customerID>= '301' AND customerID<= '310';

-- 11. Food items priced between 150 and 300.

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

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

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

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

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

-- 14. Waiters with IDs between 101 and 105.

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

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

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

-- 16. Payments with total bills greater than 500

```
SELECT PaymentID,totalBill from Payment HAVING totalBill>500 ;
```

-- 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 ;
```

-- 24. Total orders per customer

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

-- 25. Number of payments done by different payment methods.

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

-- 27. Number of food items of the menu types.

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

-- Join Queries:

-- 28. Food types and their corresponding dishes.

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

-- 29. The order IDs dealt by each waiters

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

-- 30. The order IDs dealt by each Kitchen staffs

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

-- 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;
```

-- 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;
```

-- 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;

-- 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;

