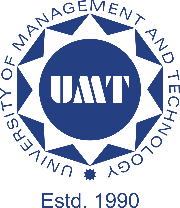
**Department of Informatics and Systems**

**University of Management and Technology**

**DataBase Systems**

**semester project**

**Submitted To:**

**Mam Maidah Sabir**

**Submitted By:**

**Zainab Zahid**

**F2022105062**

Table of Contents

**Project Description3**

Features and functionality3

Benefits4

**Objectives4**

**Goals4**

**Stakeholders5**

**Targeted customers6**

**Mapping6**

**ERD7**

**EERD8**

**Code and outputs 9**

# **Project description**

**Project Description:** Coffee Shop Management System

The Coffee Shop Management System is a comprehensive software solution designed to streamline and automate various aspects of operating a coffee shop. This project aims to provide an efficient and user-friendly platform for managing the day-to-day operations, enhancing customer service, and optimizing business processes in a coffee shop environment.

## **Features and Functionality:**

* **Menu Management**: The system allows coffee shop owners to create and manage their menu, including various coffee beverages, snacks, pastries, and other items. They can update prices, add new items, and categorize them for easy navigation.
* **Order Management:** Customers can place their orders through various channels, such as a mobile app, website, or in-store. The system captures and organizes the orders, ensuring accurate fulfillment. It enables the staff to view and process orders efficiently, reducing waiting times and improving customer satisfaction.
* **Inventory Management:** The system tracks inventory levels and automatically updates them as items are sold. It provides alerts when stock is running low, facilitating timely restocking. This feature helps in managing the supply chain and minimizing wastage.
* **Point of Sale (POS):** The coffee shop management system includes a POS interface that allows cashiers or staff members to process transactions. It supports different payment methods, such as cash, credit cards, or mobile payments, and generates receipts for customers.
* **Customer Relationship Management (CRM):** The system stores customer information and order history, enabling personalized services and loyalty programs. It helps in building customer profiles, tracking preferences, and sending targeted promotions or offers.
* **Staff Management:** The system allows coffee shop owners to manage their staff schedules, assign roles, and track attendance. It streamlines communication between staff members and provides a platform for internal messaging or announcements.
* **Reporting and Analytics:** The system generates comprehensive reports on various aspects of the coffee shop's operations, including sales, inventory, customer trends, and employee performance. These insights help in making data-driven decisions and identifying areas for improvement.
* **Integration with External Systems:** The coffee shop management system can integrate with external systems like accounting software, online ordering platforms, or loyalty program providers. This integration ensures smooth data flow and reduces manual data entry.

## **Benefits:**

* **Increased operational efficiency:** The automation of key processes reduces manual effort and streamlines operations, resulting in improved efficiency and reduced errors.
* **Enhanced customer experience:** The system enables faster order processing, personalized services, and convenient payment options, leading to a better customer experience.
* **Inventory optimization:** Real-time inventory tracking helps in reducing out of stock, minimizing wastage, and maintaining optimal stock levels.
* **Data-driven decision-making:** The reporting and analytics feature provides valuable insights into business performance, enabling informed decision-making for growth and profitability.
* **Scalability:** The system is designed to accommodate the growing needs of the coffee shop, allowing for the addition of new features, integration with external systems, and support for multiple locations.

# **Objectives**

The objective of the project on coffee shop management is to develop a comprehensive system that streamlines and optimizes various aspects of running a coffee shop. The system should cover areas such as customer management, inventory management, sales tracking, employee scheduling, and reporting.

# **Goals**

* **Efficient Customer Management:** Implement a customer relationship management (CRM) system that allows the coffee shop to track customer preferences, manage loyalty programs, and improve personalized customer service.
* **Inventory Management:** Develop an inventory management system that tracks the stock of various coffee beans, ingredients, supplies, and equipment. The system should provide alerts for low stock, automate reordering processes, and optimize inventory levels to minimize waste and maximize profitability.
* **Sales Tracking and Reporting**: Create a sales tracking system that records daily sales, identifies popular items, and analyzes sales trends. Generate comprehensive reports to gain insights into the coffee shop's performance and make informed business decisions.
* **Employee Scheduling and Performance Tracking:** Develop an employee scheduling system that optimizes shift planning, manages time-off requests, and ensures adequate staffing levels. Implement mechanisms to track employee performance, monitor productivity, and provide performance feedback.
* **Streamlined Point of Sale (POS) System:** Design a user-friendly POS system that allows seamless order taking, payment processing, and integration with other systems such as inventory management and customer management.
* **Enhanced Marketing and Promotion**: Implement tools to support marketing and promotional activities, such as email marketing campaigns, social media integration, and customer loyalty programs. Utilize data analytics to measure the effectiveness of marketing efforts.
* **Online Ordering and Delivery Integration:** Develop an online ordering platform or integrate with popular delivery services to provide customers with the convenience of ordering coffee and food items online for pick-up or delivery.
* **Sustainability and Waste Management:** Implement strategies to minimize waste generation, promote sustainable sourcing of coffee beans, and encourage environmentally friendly practices within the coffee shop.
* **Continuous Improvement and Adaptability**: Build the system with scalability and flexibility in mind to accommodate future growth and changing business needs. Regularly gather feedback from customers, employees, and stakeholders to identify areas for improvement and implement enhancements accordingly.
* **Cost Control and Profitability:** Optimize operational efficiency, reduce overhead costs, and maximize profitability by analyzing financial data, monitoring expenses, and identifying areas where cost savings can be achieved without compromising quality.

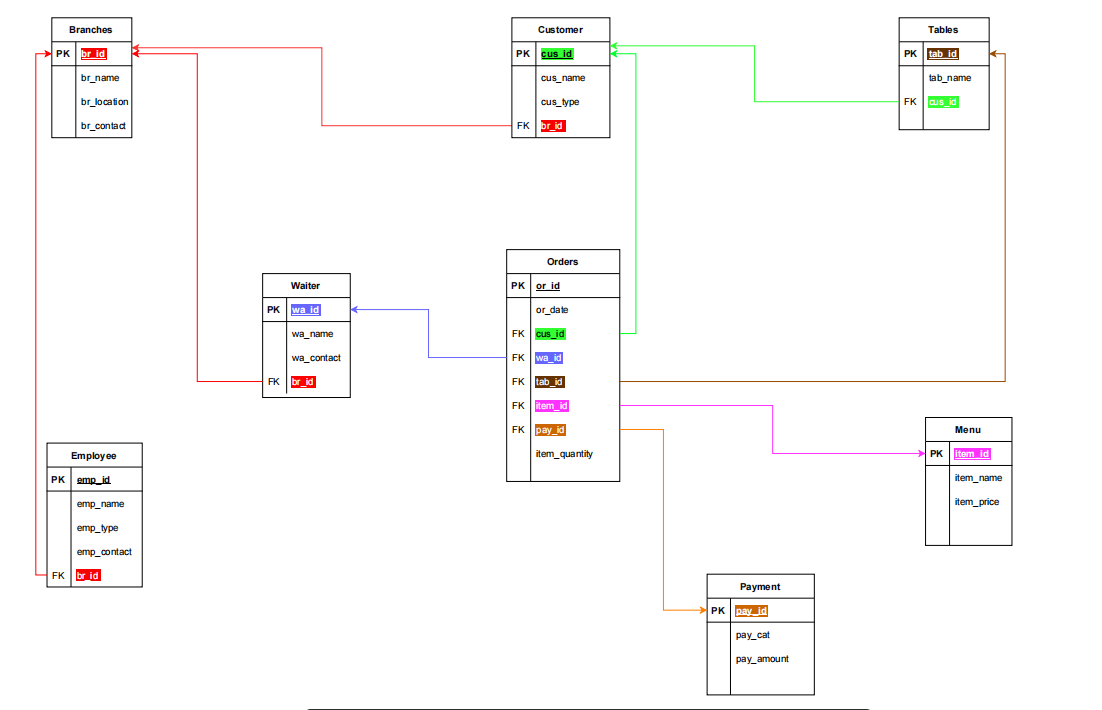
# **Stakeholders**

* **Owner(s) or Investor(s):** The individuals or group funding the coffee shop project and have a vested interest in its success.
* **Management Team:** The individuals responsible for overseeing the operations, financials, and overall performance of the coffee shop.
* **Employees:** Baristas, chefs, servers, and other staff members who work at the coffee shop and contribute to its daily functioning.
* **Suppliers:** Companies or individuals providing coffee beans, baked goods, dairy products, equipment, and other essential supplies to the coffee shop.
* **Local Community:** Residents, businesses, and organizations in the vicinity of the coffee shop who may be impacted by its presence and could potentially support it.

# **Targeted customers**

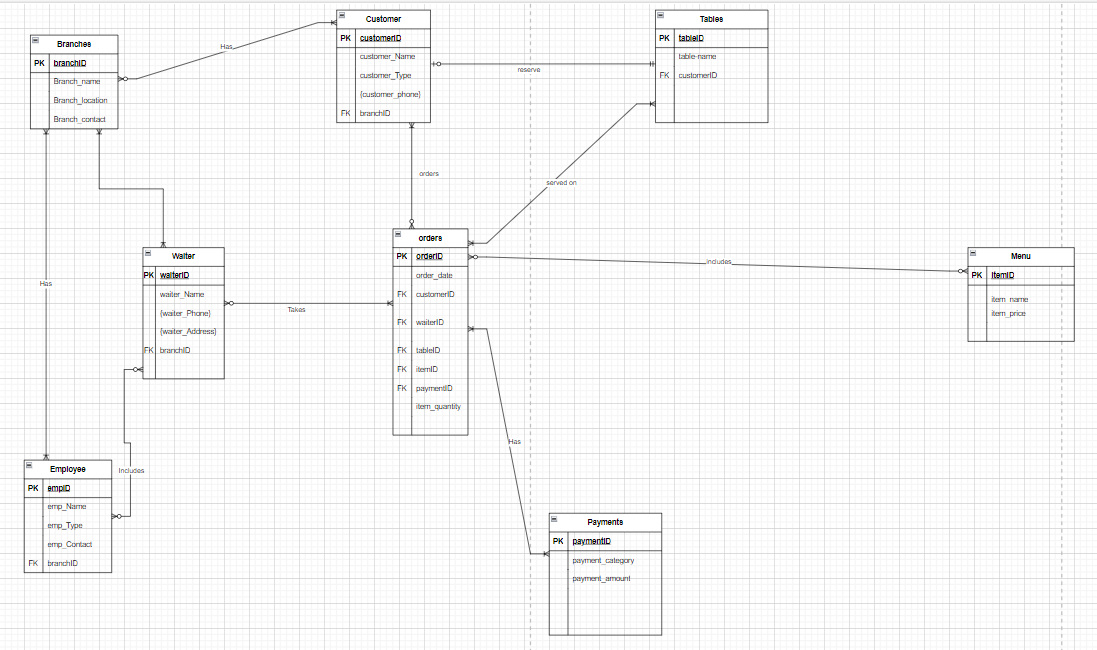
* **Coffee Enthusiasts:** Individuals who have a strong appreciation for high-quality coffee, different brewing methods, and unique flavors.
* **Office Workers:** Professionals who seek a convenient place to have meetings, work remotely, or grab a coffee during breaks.
* **Students:** College or university students who need a study-friendly environment and a caffeine boost.
* **Tourists:** Travelers visiting the area who are looking for local coffee experiences and a place to relax.
* **Local Residents:** People living nearby who may become regular customers, seeking a comfortable and welcoming place to socialize or enjoy a cup of coffee.

# **Mapping**



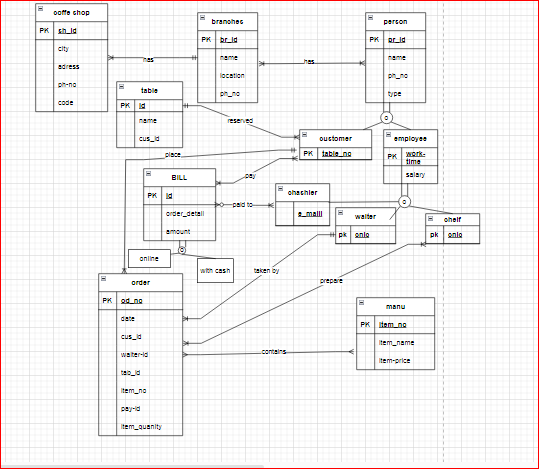
**Figure 1**

# **ERD**



**Figure 2**

# **EERD**



**Figure 3**

# **Code and Outputs:**

CREATE DATABASE LAB\_PROJECT

GO

USE LAB\_PROJECT

GO

**-- Create Table Of Branches**

CREATE TABLE Branches

(

Branch\_id INT,

Branch\_name VARCHAR(50),

Branch\_location VARCHAR(100),

Branch\_contact VARCHAR(20),

CONSTRAINT PK\_Branch\_id PRIMARY KEY(Branch\_id)

)

**-- Create Table Of Employee**

CREATE TABLE Employee

(

Employee\_Id INT,

Employee\_Name VARCHAR(50) UNIQUE,

Employee\_Type VARCHAR(50),

Employee\_contact VARCHAR(20),

Branch\_id INT,

CONSTRAINT PK\_Employee\_Id PRIMARY KEY(Employee\_Id),

CONSTRAINT FK\_Employee\_Branch\_id FOREIGN KEY (Branch\_id) REFERENCES Branches(Branch\_id)

)

**--Create Table Of Waiter**

CREATE TABLE Waiter

(

Waiter\_Id INT,

Waiter\_Name VARCHAR(50),

Waiter\_Address VARCHAR(100) DEFAULT 'LHR',

Waiter\_Phone VARCHAR(20),

Branch\_id INT,

CONSTRAINT PK\_Waiter\_Id PRIMARY KEY(Waiter\_Id),

CONSTRAINT FK\_Waiter\_Branch\_id FOREIGN KEY (Branch\_id) REFERENCES Branches(Branch\_id)

)

**-- Create Table Of Customer**

CREATE TABLE Customer

(

Customer\_Id INT,

Customer\_Name VARCHAR(50),

Customer\_Type VARCHAR(50),

Customer\_Phone VARCHAR(20),

Branch\_id INT,

CONSTRAINT PK\_Customer\_Id PRIMARY KEY(Customer\_Id),

CONSTRAINT FK\_Customer\_Branch\_id FOREIGN KEY (Branch\_id) REFERENCES Branches(Branch\_id)

)

**-- Create Table Of Tables**

CREATE TABLE Tables

(

Table\_Id INT,

Table\_Name VARCHAR(50),

Customer\_Id INT,

CONSTRAINT PK\_Table\_Id PRIMARY KEY(Table\_Id),

CONSTRAINT FK\_Table\_Customer\_id FOREIGN KEY (Customer\_Id) REFERENCES Customer(Customer\_Id)

)

**-- Create Table Of Menu**

CREATE TABLE Menu

(

Item\_Id INT,

Item\_name VARCHAR(50),

Item\_price MONEY NOT NULL,

CONSTRAINT PK\_Item\_Id PRIMARY KEY(Item\_Id)

)

**-- Create Table Of Payments**

CREATE TABLE Payments

(

Payment\_Id INT,

Payment\_category VARCHAR(50),

Payment\_amount MONEY CHECK (Payment\_amount<500),

CONSTRAINT PK\_Payment\_Id PRIMARY KEY(Payment\_Id)

)

**-- Create Table Of Order**

CREATE TABLE Orders

(

Order\_Id INT,

Order\_date DATE,

Customer\_Id INT,

Waiter\_Id INT,

Table\_Id INT,

Item\_Id INT,

Payment\_Id INT,

Item\_Quantity INT,

CONSTRAINT PK\_Order\_Id PRIMARY KEY(Order\_Id),

CONSTRAINT FK\_Orders\_Customer\_id FOREIGN KEY (Customer\_Id) REFERENCES Customer(Customer\_Id),

CONSTRAINT FK\_Orders\_Waiter\_Id FOREIGN KEY (Waiter\_Id) REFERENCES Waiter(Waiter\_Id),

CONSTRAINT FK\_Orders\_Table\_Id FOREIGN KEY (Table\_Id) REFERENCES Tables(Table\_Id),

CONSTRAINT FK\_Orders\_Item\_Id FOREIGN KEY (Item\_Id) REFERENCES Menu(Item\_Id),

CONSTRAINT FK\_Orders\_Payment\_Id FOREIGN KEY (Payment\_Id) REFERENCES Payments(Payment\_Id)

)

**-- Create Table Of ABC**

CREATE TABLE ABC

(

ID INT,

NAME VARCHAR(25),

CONSTRAINT PK\_Id PRIMARY KEY(Id)

)

INSERT INTO Branches VALUES (1, 'Branch 1', 'Location A', '123-456-7890')

INSERT INTO Branches VALUES (2, 'Branch 2', 'Location B', '987-654-3210')

INSERT INTO Branches VALUES (3, 'Branch 3', 'Location C', '555-123-4567')

INSERT INTO Branches VALUES (4, 'Branch 4', 'Location D', '111-222-3333')

INSERT INTO Branches VALUES (5, 'Branch 5', 'Location E', '444-555-6666')

INSERT INTO Employee VALUES (1, 'Farah Asif', 'Manager', '123-456-7890', 1)

INSERT INTO Employee VALUES (2, 'Horaain Fatima', 'Waiter', '956-693-3278', 2)

INSERT INTO Employee VALUES (3, 'Sarah Ali', 'Chef', '132-222-4527', 2)

INSERT INTO Employee VALUES (4, 'Ali Khan', 'Cashier', '844-1455-456', 2)

INSERT INTO Employee VALUES (5, 'Arfa Farhan', 'Waiter', '576-123-4327', 3)

INSERT INTO Employee VALUES (6, 'Waleed', 'Chef', '132-222-4527', 3)

INSERT INTO Employee VALUES (7, 'Zoha Younas', 'Cashier', '844-1455-456', 3)

INSERT INTO Employee VALUES (8, 'Alia Yousaf', 'Waiter', '567-897-343', 4)

INSERT INTO Employee VALUES (9, 'Usman Farooq', 'Chef', '908-4356-123', 4)

INSERT INTO Employee VALUES (10, 'Zohaib Tariq', 'Cashier', '5690-2456-8987', 4)

INSERT INTO Employee VALUES (11, 'Asifa Alam', 'Waiter', '342-676-327', 5)

INSERT INTO Employee VALUES (12, 'Minahil', 'Chef', '455-876-111', 5)

INSERT INTO Employee VALUES (13, 'Duaa Fatima', 'Cashier', '111-678-555', 5)

INSERT INTO Waiter VALUES (1, 'Abdullah Khan','', '123-456-7890', 1)

INSERT INTO Waiter VALUES (2, 'Waleed Dar', 'LHR DHA PHASE 1', '956-693-3278', 2)

INSERT INTO Waiter VALUES (3, 'Irtaza', 'LHR DHA PHASE 2', '132-222-4527', 3)

INSERT INTO Waiter VALUES (4, 'Usama', 'LHR DHA PHASE 1', '844-1455-456', 4)

INSERT INTO Waiter VALUES (5, 'Noman Saif', 'LHR DHA PHASE 2', '576-123-4327', 5)

INSERT INTO Customer VALUES (1, 'Sarah Azam', 'VIP', '987-654-3210', 1)

INSERT INTO Customer VALUES (2, 'Ali Khan', 'Regular', '555-123-4567', 2)

INSERT INTO Customer VALUES (3, 'Sarah Almas', 'Regular', '111-222-3333', 2)

INSERT INTO Customer VALUES (4, 'Zarnab Alyas', 'VIP', '444-555-6666', 3)

INSERT INTO Customer VALUES (5, 'Eman Fatima', 'Regular', '777-888-9999', 3)

INSERT INTO Customer VALUES (6, 'sidra Batool', 'VIP', '222-333-4444', 1)

INSERT INTO Customer VALUES (7, 'Ahmed Nadeem', 'Regular', '666-777-8888', 2)

INSERT INTO Tables VALUES (1, 'Table 1', 1)

INSERT INTO Tables VALUES (2, 'Table 2', 2)

INSERT INTO Tables VALUES (3, 'Table 3', 3)

INSERT INTO Tables VALUES (4, 'Table 4', 4)

INSERT INTO Tables VALUES (5, 'Table 5', 5)

INSERT INTO Menu VALUES (1, 'Espresso', 210.0)

INSERT INTO Menu VALUES (2, 'Cappuccino', 120.0)

INSERT INTO Menu VALUES (3, 'Latte', 130.0)

INSERT INTO Menu VALUES (4, 'Mocha',40.0 )

INSERT INTO Menu VALUES (5, 'Americano', 150.0)

INSERT INTO Payments VALUES (1, 'Cash', 10.0)

INSERT INTO Payments VALUES (2, 'Credit Card', 15.0)

INSERT INTO Payments VALUES (3, 'Online Payment', 20.00)

INSERT INTO Orders VALUES (1, '2023-05-17', 1, 2, 1, 1, 1, 2)

INSERT INTO Orders VALUES(2, '2023-05-18', 5, 1, 2, 3, 2, 1)

INSERT INTO Orders VALUES(3, '2023-05-18', 3, 4, 4, 2, 3, 3)

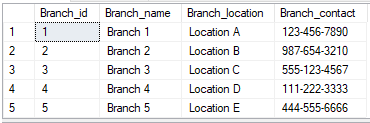
INSERT INTO Orders VALUES(4, '2023-05-19', 7, 3, 5, 4, 2, 2)

INSERT INTO Orders VALUES(5, '2023-05-20', 2, 5, 3, 5, 1, 1)

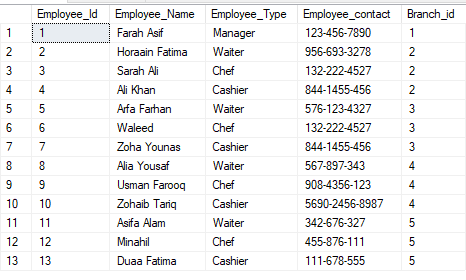
INSERT INTO ABC VALUES (1,'Ali')

INSERT INTO ABC VALUES (2,'Rehan')

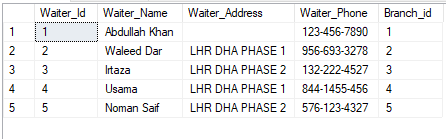
SELECT \*FROM Branches



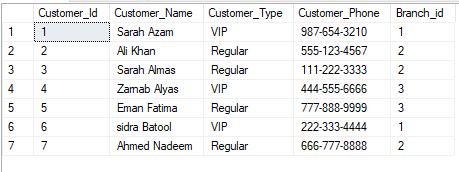
SELECT \*FROM Employee



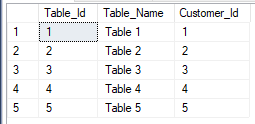
SELECT \*FROM Waiter



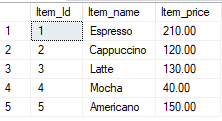
SELECT \*FROM Customer



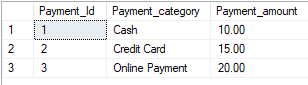
SELECT \*FROM Tables



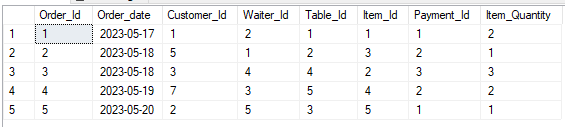
SELECT \*FROM Menu



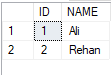
SELECT \*FROM Payments



SELECT \*FROM Orders

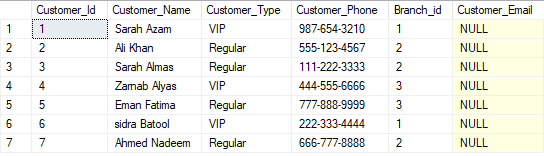


SELECT \*FROM ABC



**-- Add new column to the Customer table**

ALTER TABLE Customer ADD Customer\_Email VARCHAR(50)



**-- Truncate ABC table**

TRUNCATE TABLE ABC



**-- Add comment on the Customer table**

COMMENT ON TABLE Customer IS 'This table stores information about the customers.'

**-- Rename the ABC table to EXTRA\_TABLE**

RENAME ABC TO EXTRA

**--Add constraint to the Tables**

ALTER TABLE Employee ADD CONSTRAINT UQ\_Employee\_Name UNIQUE (Employee\_Name)

ALTER TABLE Menu ALTER COLUMN Item\_price Decimal(10,2) NOT NULL

ALTER TABLE Payments ADD CONSTRAINT CK\_Payment\_amount CHECK (Payment\_amount<100)

ALTER TABLE Waiter Add Constraint DF\_Waiter\_Address DEFAULT 'LHR' FOR Waiter\_Address

SELECT \* FROM Employee

SELECT \* FROM Menu

SELECT \* FROM Payments

SELECT \* FROM Waiter

SELECT \* FROM Employee WHERE Employee\_Type = 'Manager'



SELECT \* FROM Menu WHERE Item\_name = 'Espresso'



SELECT \* FROM Customer WHERE Customer\_Name = 'Sarah Azam'

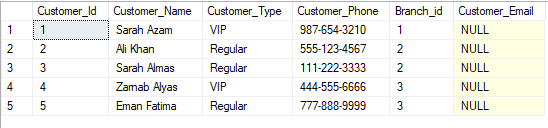


**--Sub Query**

SELECT \* FROM Menu WHERE Item\_Id IN (SELECT Item\_Id FROM Orders WHERE Waiter\_Id = 2)



SELECT \* FROM Customer WHERE Customer\_Id IN (SELECT Customer\_Id FROM Tables)



SELECT \* FROM Menu WHERE Item\_Id NOT IN (SELECT Item\_Id FROM Orders)



**-- Delete the record of employee**

DELETE FROM Employee WHERE Employee\_Id = 4

**-- Delete the record of menu**

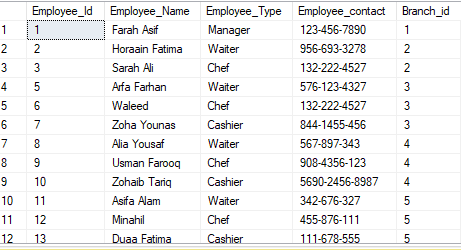
DELETE FROM Menu WHERE Item\_Id = 3

**-- Delete the record of customer**

DELETE FROM Customer WHERE Customer\_Id = 5

**-- Display the records after deletion**

SELECT \* FROM Employee



SELECT \* FROM Menu

SELECT \* FROM Customer

**-- Update the record of table employee**

UPDATE Employee SET Employee\_Name = 'John Smith' WHERE Employee\_Id = 1

**-- Update the record of table menu**

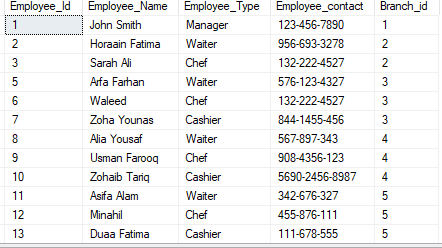
UPDATE Menu SET Item\_price = 150.0 WHERE Item\_Id = 1

**-- Update the record of table customer**

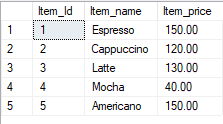
UPDATE Customer SET Customer\_Name = 'Bob Smith' WHERE Customer\_Id = 2

**-- Display the records after updating**

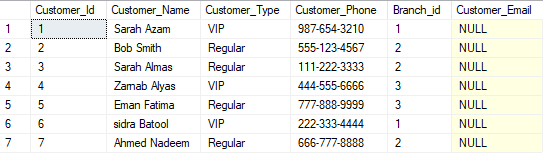
SELECT \* FROM Employee



SELECT \* FROM Menu



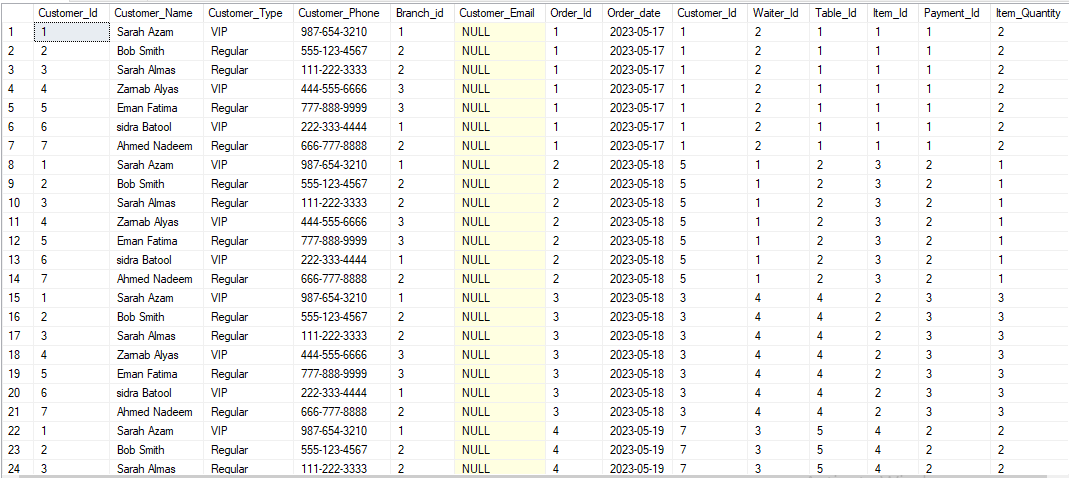
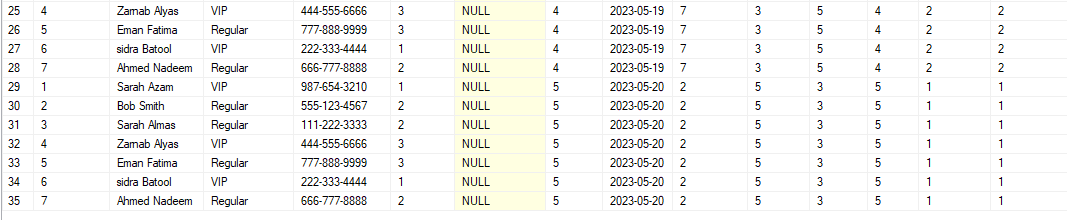
SELECT \* FROM Customer



**--Joins**

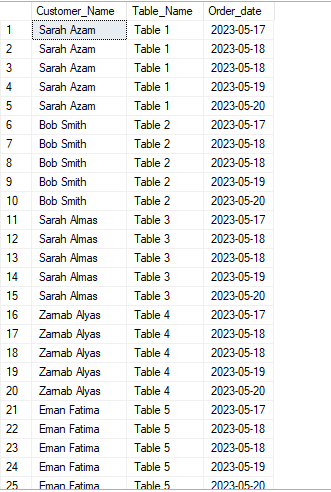
**--Cartesian**

SELECT \* FROM Customer , Orders

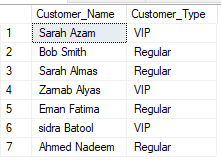
**--Inner/Equi**

SELECT c.Customer\_Name,t.Table\_Name,o.Order\_date FROM Customer c, Tables t, Orders o WHERE c.Customer\_Id =t.Customer\_Id



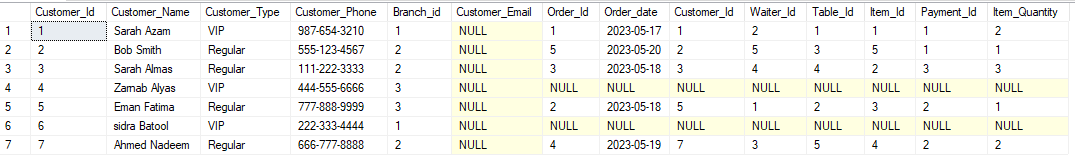
**--Self**

SELECT c1.Customer\_Name,c2.Customer\_Type FROM Customer c1,Customer c2 WHERE c1.Customer\_Id =c2.Customer\_Id

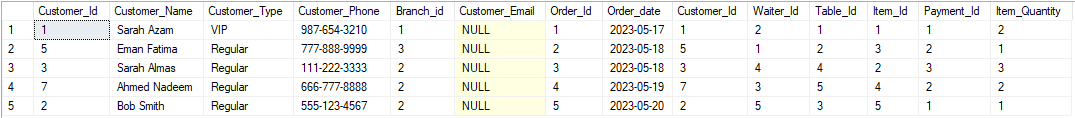


-**-Outer**

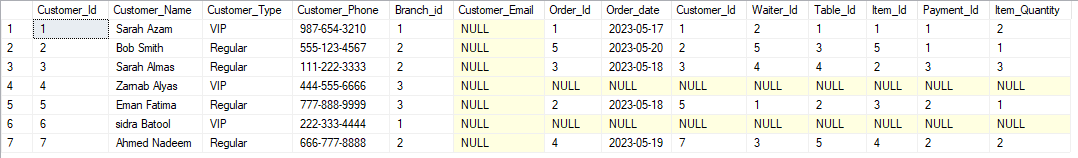
SELECT \* FROM Customer c left outer join Orders o ON c.Customer\_Id =o.Customer\_Id



SELECT \* FROM Customer c right outer join Orders o ON c.Customer\_Id =o.Customer\_Id



SELECT \* FROM Customer c FULL outer join Orders o ON c.Customer\_Id =o.Customer\_Id



**---Aggregate Functions**

SELECT COUNT(\*) AS 'No of branches'

FROM Branches



SELECT SUM(item\_price) AS 'total employee types'

FROM Menu



Select MAX(item\_price) AS 'total employee types'

FROM Menu



SELECT MIN(item\_price) AS 'total employee types'

FROM Menu



SELECT AVG(item\_price) AS 'total employee types'

FROM Menu



**--Sub Query**

SELECT COUNT(\*) AS 'No of branches'

FROM Branches t ,Orders o WHERE o.Payment\_Id >(SELECT MIN(item\_price) FROM Menu)



**--Top**

SELECT TOP(2) Employee\_name FROM Employee WHERE Employee\_Id=1



**--Distinct**

SELECT DISTINCT (Employee\_name) FROM Employee



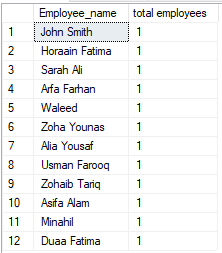
**--Group by**

SELECT e.Employee\_name , COUNT(\*) AS 'total employees'

FROM Employee e,Branches b

WHERE e.Branch\_id = b.Branch\_id

GROUP BY e.Employee\_Name

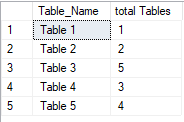


SELECT t.Table\_Name , SUM(o.Order\_id) AS 'total Tables'

FROM Tables t,orders o

WHERE t.Table\_Id = o.Table\_Id

GROUP BY t.Table\_Name

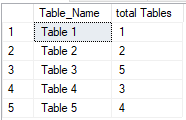


SELECT t.Table\_Name , MAX(o.Order\_id) AS 'total Tables'

FROM Tables t,orders o

WHERE t.Table\_Id = o.Table\_Id

GROUP BY t.Table\_Name

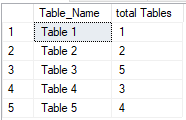


SELECT t.Table\_Name , MIN(o.Order\_id) AS 'total Tables'

FROM Tables t,orders o

WHERE t.Table\_Id = o.Table\_Id

GROUP BY t.Table\_Name

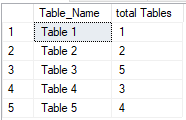


SELECT t.Table\_Name , AVG(o.Order\_id) AS 'total Tables'

FROM Tables t,orders o

WHERE t.Table\_Id = o.Table\_Id

GROUP BY t.Table\_Name



**--Order By**

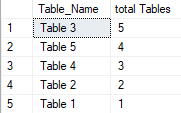
SELECT t.Table\_Name , SUM(o.Order\_id) AS 'total Tables'

FROM Tables t,orders o

WHERE t.Table\_Id = o.Table\_Id

GROUP BY t.Table\_Name

ORDER BY SUM(o.Order\_id) DESC



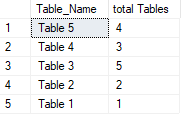
SELECT t.Table\_Name , MAX(o.Order\_id) AS 'total Tables'

FROM Tables t,orders o

WHERE t.Table\_Id = o.Table\_Id

GROUP BY t.Table\_Name

ORDER BY MAX(t.Table\_Name )DESC



**--Having**

SELECT t.Table\_id , SUM(o.Order\_id) AS 'total Tables'

FROM Tables t,orders o

WHERE t.Table\_Id = o.Table\_Id

GROUP BY t.Table\_id

HAVING t.Table\_Id=1



SELECT t.Table\_id , COUNT(o.Order\_id) AS 'total Tables'

FROM Tables t,orders o

WHERE t.Table\_Id = o.Table\_Id

GROUP BY t.Table\_id

HAVING COUNT(t.Table\_Id)>1

