NU Brewery – Final Report

# Problem Statement

A brewery deals with a large amount of data, such as information on ingredients, production processes, inventory levels, sales data, and customer information. Managing this data effectively is essential for a brewery to operate efficiently and make informed decisions.   
A database is crucial for a brewery company because it allows for the organization and management of data in a structured and efficient manner. For example, a brewery can use its database to manage inventory levels and ensure that it has enough raw materials to meet production demands without overstocking or wasting resources.   
This can also be used to optimize production schedules and track sales data across multiple branches, allowing the brewery to analyze trends and make informed decisions about pricing, marketing, and distribution. Furthermore, a DBMS can help the brewery to manage customer information and personalize marketing efforts to increase customer loyalty and drive sales.   
Overall, a database system is essential for a brewery company because it allows for the effective management and organization of large volumes of data, enabling the company to make informed decisions and operate efficiently. It ensures consistency, eliminates data redundancy, and provides efficient execution of queries and high-performance applications for the domain.

## Functionality:

The Brewery Database is designed to store and manage information pertaining to diverse aspects of the brewery, including its various branches, employees, suppliers, and product offerings.

This database will enable users to retrieve required information easily and quickly through efficient queries from a well-designed and easily maintainable database.

# Entities

1. Employee
2. Customer
3. Orders
4. Branch
5. Raw Material
6. Supplier
7. Batch
8. Product

***Associative Entities:***

1. Served
2. Visit
3. Stock
4. Production
5. Bill

***Disjoint Entities***

1. Brewer
2. Server

# Relationship between Entities

|  |  |  |
| --- | --- | --- |
| Employee ⇄ Employee | Employee ⇄ Brewer |  |
| Employee ⇄ Branch | Branch ⇄ Visit |  |
| Served ⇄ Server | Served ⇄ Customer |  |
| Customer⇄ Visit | Customer⇄ Orders |  |
| Product ⇄ Batch | Batch ⇄ Stock |  |
| Batch⇄ Production | Stock⇄ Brewer |  |
| Production⇄ Raw Material | Raw Material ⇄ Bill |  |
| Bill⇄ Supplier | Employee ⇄ Server |  |
|  |  |  |

# Cardinalities of Relationships among Entities

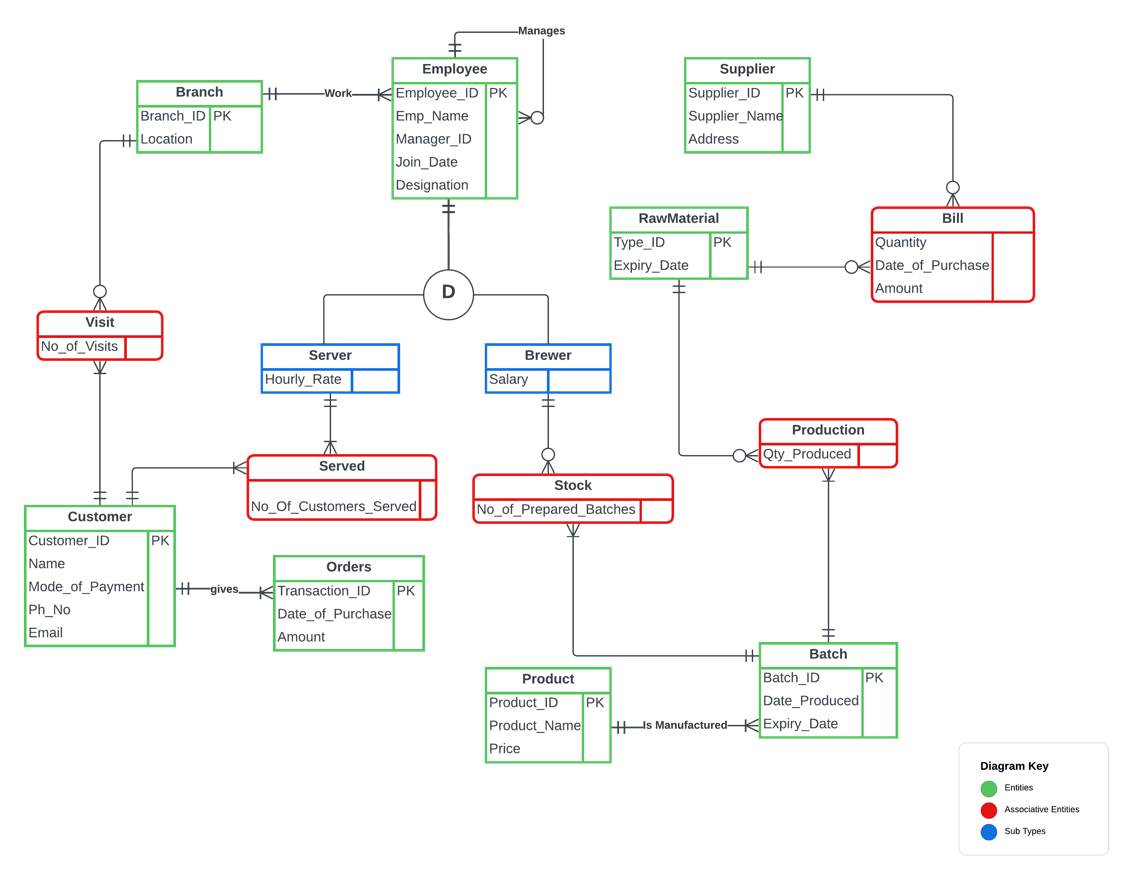
|  |  |
| --- | --- |
| Employee (Mandatory One) ⇄ Employee (Mandatory Many) | Product (Mandatory One) ⇄ Batch (Mandatory Many) |
| Employee (Mandatory One) ⇄ Brewer | Batch (Mandatory One) ⇄ Stock (Mandatory Many) |
| Employee (Mandatory Many) ⇄ Branch (Mandatory One) | Batch (Mandatory One) ⇄ Production (Mandatory Many) |
| Branch (Mandatory One) ⇄ Visit (Optional Many) | Stock (Optional Many) ⇄ Brewer (Mandatory One) |
| Served (Mandatory Many) ⇄ Server Mandatory One) | Production (Optional Many) ⇄ Raw Material (Mandatory One) |
| Served (Mandatory Many) ⇄ Customer (Mandatory One) | Raw Material (Mandatory One) ⇄ Bill (Optional Many) |
| Customer (Mandatory One) ⇄ Visit (Mandatory Many) | Bill (Optional Many) ⇄ Supplier (Mandatory One) |
| Customer (Mandatory One) ⇄ Orders (Mandatory Many) | Employee (Mandatory One) ⇄ Server |

# Attributes of All Entities

Below are the attributes, we as a group thought it is important to get an overall picture for running a Brewery and get good insights.

|  |  |
| --- | --- |
| Table | Attributes |
| Employee | Employee ID, Emp Name, Manager ID, Join Date, Designation (Discriminator) |
| Server | B\_Employee ID, Hourly Rate |
| Brewer | S\_Employee ID, Salary |
| Branch | Branch ID, Location |
| Customer | Customer ID, Name, Mode of Payment, Ph. No, Email |
| Orders | Transaction ID, Date of Purchase, Amount |
| Product | Product ID, Product Name, Price |
| Batch | Batch ID, Date Produced, Expiry Date |
| Raw Material | Type ID, Expiry Date |
| Supplier | Supplier ID, Supplier Name, Address |
| Production | Qty Produced, Qty of Raw Materials Used |
| Bill | Quantity, Date of Purchase, Amount |
| Stock | No. of Prepared Batches |
| Served | Customers Served |
| Visit | No. of Visits |

# Entity Relationship Diagram



# Relational Schema of NU Brewery

# Data Normalization

Partial Dependencies existed which were removed and converted into 3NF.

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

# Summary Table for each entity

|  |  |
| --- | --- |
| **Employee Table** Employee (Employee\_ID, Emp\_Name, Manager\_ID, Join\_Date, Designation, Branch\_ID)  Data Type: VARCHAR (6), VARCHAR (20), VARCHAR (6), DATE, VARCHAR (25) respectively.  Additional Details: Employee\_ID is Unique, and all fields are required. Manager\_ID & Branch\_ID are foreign keys. | **Customer Table** Customer (Customer\_ID, Name, Mode\_of\_Payment, Ph\_No, Email)  Data Type: VARCHAR (6), VARCHAR (20), VARCHAR (6), INTEGER (10), VARCHAR (30) respectively  Additional Details: Customer\_ID is Unique. |
| **Supplier Table** Supplier (Supplier\_ID, Supplier\_Name, Address)  Data Type: VARCHAR (6), VARCHAR (20), VARCHAR (50) respectively  Additional Details: Supplier\_ID is Unique, and all fields are required. | Orders Table Orders (Transaction\_ID, Date\_of\_Purchase, Amount, Customer\_ID)  Data Type: VARCHAR (10), DATE, INTEGER (5), VARCHAR (6) respectively  Additional Details: Transaction\_ID is Unique, and all fields are required. Customer\_ID is a foreign key. |
| **Branch Table** Branch (Branch\_ID, Location)  Data Type: VARCHAR (5), VARCHAR (10) respectively.  Additional Details: Branch\_ID is Unique, and all fields are required. | Product Table Product (Product\_ID, Product\_Name, Price)  Data Type: VARCHAR (5), VARCHAR (20), INTEGER (4) respectively.  Additional Details: Product\_ID is Unique, and all fields are required. |
| **Batch Table** Batch (Batch\_ID, Date\_Produced, Expiry\_Date, Product\_ID)  Data Type: VARCHAR (7), DATE, DATE, VARCHAR (5) respectively.  Additional Details: Batch\_ID is Unique, and all fields are required. Product\_ID is a foreign key. | Raw Material Table Raw Material (Type\_ID, Expiry\_Date)  Data Type: VARCHAR (5), DATE respectively.  Additional Details: Type\_ID is Unique, and all fields are required. |
| **Production Table** Production (Qty\_Produced, Batch\_ID, Type\_ID)  Data Type: INTEGER (5), VARCHAR (7), VARCHAR (5) respectively.  Additional Details: Batch\_ID & Type\_ID are Unique, and all fields are required. Batch\_ID & Type\_ID are foreign keys. | Bill Table Bill (Quantity, Date\_of\_Purchase, Amount, Type\_ID, Supplier\_ID)  Data Type: INTEGER (5), DATE, INTEGER (5), VARCHAR (5), VARCHAR (6)  Additional Details: Type\_ID & Supplier\_ID together are Unique, and all fields are required. Type\_ID & Supplier\_ID are foreign keys. |
| **Stock Table** Stock (No\_of\_Prepared\_Batches, Batch\_ID, BEmployee\_ID)  Data Type: INTEGER (3), VARCHAR (7), VARCHAR (6)  Additional Details: Batch\_ID & BEmployee\_ID together are Unique, and all fields are required. Batch\_ID & BEmployee\_ID are foreign keys. | Served Table Served (No\_of\_Customers\_Served, Customer\_ID, SEmployee\_ID)  Data Type: INTEGER (3), VARCHAR (6), VARCHAR (6)  Additional Details: Customer\_ID & SEmployee\_ID together are Unique, and all fields are required. Customer\_ID & SEmployee\_ID are foreign keys. |
| **Visit Table** Visit (Number\_of\_Visits, Customer\_ID, Branch\_ID)  Data Type: INTEGER (3), VARCHAR (6), VARCHAR (5)  Additional Details: Customer\_ID & Branch\_ID together are Unique, and all fields are required. Customer\_ID & Branch\_ID are foreign key | Brewer Table Brewer (BEmployee\_ID, Salary)  Data Type: VARCHAR (6), INTEGER (8)  Additional Details: BEmployee\_ID is Unique, and all fields are required. BEmployee\_ID is a foreign key. |
| **Server Table** Server (SEmployee\_ID, Hourly\_Rate)  Data Type: VARCHAR (6), INTEGER (3)  Additional Details: SEmployee\_ID is Unique, and all fields are required. SEmployee\_ID is a foreign key. |  |

# Creation of Tables

|  |  |
| --- | --- |
| Employee Table Employee (Employee\_ID, Emp\_Name, Manager\_ID, Join\_Date, Designation, Branch\_ID)  Data Type: VARCHAR (6), VARCHAR (20), VARCHAR (6), DATE, VARCHAR (25) respectively  Additional Details: Employee\_ID is Unique, and all fields are required. Manager\_ID & Branch\_ID are foreign keys.    DROP TABLE IF EXISTS Employee;  CREATE TABLE Employee  (  Employee\_ID VARCHAR (6) NOT NULL,  Emp\_Name VARCHAR (20) NOT NULL,  Join\_Date DATE NOT NULL,  Designation VARCHAR (25) NOT NULL,  Branch\_ID VARCHAR (5) NOT NULL,  Manager\_ID VARCHAR (6),  CONSTRAINT Employee\_ID PRIMARY KEY (Employee\_ID),  FOREIGN KEY (Branch\_ID)  REFERENCES Branch(Branch\_ID),  FOREIGN KEY (Manager\_ID)  REFERENCES Employee (Employee\_ID)  ); | Brewer Table Brewer (BEmployee\_ID, Salary)  Data Type: VARCHAR (6), INTEGER (8)  Additional Details: BEmployee\_ID is Unique, and all fields are required. BEmployee\_ID is a foreign key.    DROP TABLE IF EXISTS Brewer;  CREATE TABLE Brewer  (  BEmployee\_ID VARCHAR (6) NOT NULL,  Salary INT (8) NOT NULL,  CONSTRAINT BEmployee\_ID PRIMARY KEY (BEmployee\_ID),  FOREIGN KEY (BEmployee\_ID)  REFERENCES Employee (Employee\_ID)  ); |
| Server Table Server (SEmployee\_ID, Hourly\_Rate)  Data Type: VARCHAR (6), INTEGER (3)  Additional Details: SEmployee\_ID is Unique, and all fields are required. SEmployee\_ID is a foreign key.    DROP TABLE IF EXISTS Server;  CREATE TABLE Server  (  SEmployee\_ID VARCHAR (6) NOT NULL,  Hourly\_Rate INT (3) NOT NULL,  CONSTRAINT SEmployee\_ID PRIMARY KEY (SEmployee\_ID),  FOREIGN KEY (SEmployee\_ID)  REFERENCES Employee (Employee\_ID)  ); | Customer Table Customer (Customer\_ID, Name, Mode\_of\_Payment, Ph\_No, Email)  Data Type: VARCHAR (6), VARCHAR (20), VARCHAR (6), INTEGER (10), VARCHAR (30) respectively  Additional Details: Customer\_ID is Unique.    DROP TABLE IF EXISTS Customer;  CREATE TABLE Customer  (  Customer\_ID VARCHAR (6) NOT NULL,  Name VARCHAR (20) NOT NULL,  Mode\_of\_Payment VARCHAR (6) NOT NULL,  Ph\_No VARCHAR (10) NOT NULL,  Email VARCHAR (30),  CONSTRAINT Customer\_ID PRIMARY KEY (Customer\_ID)  ); |
| Supplier Table Supplier (Supplier\_ID, Supplier\_Name, Address)  Data Type: VARCHAR (6), VARCHAR (20), VARCHAR (50) respectively  Additional Details: Supplier\_ID is Unique, and all fields are required.    DROP TABLE IF EXISTS Supplier;  CREATE TABLE Supplier  (  Supplier\_ID VARCHAR (6) NOT NULL,  Supplier\_Name VARCHAR (20) NOT NULL,  Address VARCHAR (50) NOT NULL,  CONSTRAINT Supplier\_ID PRIMARY KEY (Supplier\_ID)  ); | Branch Table Branch (Branch\_ID, Location)  Data Type: VARCHAR (5), VARCHAR (10) respectively.  Additional Details: Branch\_ID is Unique, and all fields are required.    DROP TABLE IF EXISTS Branch;  CREATE TABLE Branch  (  Branch\_ID VARCHAR (5) NOT NULL,  Location VARCHAR (10) NOT NULL,  CONSTRAINT Branch\_ID PRIMARY KEY (Branch\_ID)  ); |
| Product Table Product (Product\_ID, Product\_Name, Price)  Data Type: VARCHAR (5), VARCHAR (20), INTEGER (4) respectively.  Additional Details: Product\_ID is Unique, and all fields are required.    DROP TABLE IF EXISTS Product;  CREATE TABLE Product  (  Product\_ID VARCHAR (5) NOT NULL,  Product\_Name VARCHAR (20) NOT NULL,  Price INT (4) NOT NULL,  CONSTRAINT Product\_ID PRIMARY KEY (Product\_ID)  ); | Batch Table Batch (Batch\_ID, Date\_Produced, Expiry\_Date, Product\_ID)  Data Type: VARCHAR (7), DATE, DATE, VARCHAR (5) respectively.  Additional Details: Batch\_ID is Unique, and all fields are required. Product\_ID is a foreign key.    DROP TABLE IF EXISTS Batch;  CREATE TABLE Batch (  Batch\_ID VARCHAR (7) NOT NULL,  Date\_Produced DATE NOT NULL,  Expiry\_Date DATE NOT NULL,  Product\_ID VARCHAR (5) NOT NULL,  CONSTRAINT Batch\_ID PRIMARY KEY (Batch\_ID),  FOREIGN KEY (Product\_ID)  REFERENCES Product (Product\_ID)  ); |
| Raw Material Table Raw Material (Type\_ID, Expiry\_Date)  Data Type: VARCHAR (5), DATE respectively.  Additional Details: Type\_ID is Unique, and all fields are required.    DROP TABLE IF EXISTS RawMaterial;  CREATE TABLE RawMaterial (  Type\_ID VARCHAR (5) NOT NULL,  Expiry\_Date DATE NOT NULL,  CONSTRAINT Type\_ID PRIMARY KEY (Type\_ID)  ); | Production Table Production (Qty\_Produced, Batch\_ID, Type\_ID)  Data Type: INTEGER (5), VARCHAR (7), VARCHAR (5) respectively.  Additional Details: Batch\_ID & Type\_ID are Unique, and all fields are required. Batch\_ID & Type\_ID are foreign keys.    DROP TABLE IF EXISTS Production;  CREATE TABLE Production (  Batch\_ID VARCHAR (7) NOT NULL,  Type\_ID VARCHAR (5) NOT NULL,  Qty\_Produced INT (5) NOT NULL,  CONSTRAINT BT\_ID PRIMARY KEY (Batch\_ID, Type\_ID),  FOREIGN KEY (Batch\_ID)  REFERENCES Batch (Batch\_ID),  FOREIGN KEY (Type\_ID)  REFERENCES RawMaterial (Type\_ID)  ); |
| Bill Table Bill (Quantity, Date\_of\_Purchase, Amount, Type\_ID, Supplier\_ID)  Data Type: INTEGER (5), DATE, INTEGER (5), VARCHAR (5), VARCHAR (6)  Additional Details: Type\_ID & Supplier\_ID together are Unique, and all fields are required. Type\_ID & Supplier\_ID are foreign keys.    DROP TABLE IF EXISTS Bill;  CREATE TABLE Bill (  Date\_Of\_Purchase DATE NOT NULL,  Quantity INT(5) NOT NULL,  Amount INT(5) NOT NULL,  Type\_ID VARCHAR(5) NOT NULL,  Supplier\_ID VARCHAR(6) NOT NULL,  CONSTRAINT TS\_ID PRIMARY KEY (Type\_ID , Supplier\_ID),  FOREIGN KEY (Type\_ID)  REFERENCES RawMaterial (Type\_ID),  FOREIGN KEY (Supplier\_ID)  REFERENCES Supplier (Supplier\_ID)  ); | Stock Table Stock (No\_of\_Prepared\_Batches, Batch\_ID, BEmployee\_ID)  Data Type: INTEGER (3), VARCHAR (7), VARCHAR (6)  Additional Details: Batch\_ID & BEmployee\_ID together are Unique, and all fields are required. Batch\_ID & BEmployee\_ID are foreign keys.    DROP TABLE IF EXISTS Stock;  CREATE TABLE Stock (  No\_Of\_Prepared\_Batches INT(3) NOT NULL,  Batch\_ID VARCHAR(7) NOT NULL,  BEmployee\_ID VARCHAR(6) NOT NULL,  CONSTRAINT Stock PRIMARY KEY (Batch\_ID , BEmployee\_ID),  FOREIGN KEY (Batch\_ID)  REFERENCES Batch (Batch\_ID),  FOREIGN KEY (BEmployee\_ID)  REFERENCES Brewer (BEmployee\_ID)  ); |
| Served Table Served (No\_of\_Customers\_Served, Customer\_ID, SEmployee\_ID)  Data Type: INTEGER (3), VARCHAR (6), VARCHAR (6)  Additional Details: Customer\_ID & SEmployee\_ID together are Unique, and all fields are required. Customer\_ID & SEmployee\_ID are foreign keys.    DROP TABLE IF EXISTS Served;  CREATE TABLE Served (  No\_Of\_Customers\_Served INT(3) NOT NULL,  SEmployee\_ID VARCHAR(12) NOT NULL,  Customer\_ID VARCHAR(6) NOT NULL,  CONSTRAINT SC\_ID PRIMARY KEY (SEmployee\_ID , Customer\_ID),  FOREIGN KEY (SEmployee\_ID)  REFERENCES Server (SEmployee\_ID),  FOREIGN KEY (Customer\_ID)  REFERENCES Customer (Customer\_ID)  ); | Visit Table Visit (Number\_of\_Visits, Customer\_ID, Branch\_ID)  Data Type: INTEGER (3), VARCHAR (6), VARCHAR (5)  Additional Details: Customer\_ID & Branch\_ID together are Unique, and all fields are required. Customer\_ID & Branch\_ID are foreign keys.    DROP TABLE IF EXISTS Visit;  CREATE TABLE Visit (  Branch\_ID VARCHAR(5) NOT NULL,  Customer\_ID VARCHAR(6) NOT NULL,  No\_Of\_Visits INT(3) NOT NULL,  CONSTRAINT BC\_ID PRIMARY KEY (Branch\_ID , Customer\_ID),  FOREIGN KEY (Branch\_ID)  REFERENCES Branch (Branch\_ID),  FOREIGN KEY (Customer\_ID)  REFERENCES Customer (Customer\_ID)  ); |
| Orders Table Orders (Transaction\_ID, Date\_of\_Purchase, Amount, Customer\_ID)  Data Type: VARCHAR (10), DATE, INTEGER (5), VARCHAR (6) respectively  Additional Details: Transaction\_ID is Unique, and all fields are required. Customer\_ID is a foreign key.  DROP TABLE IF EXISTS Orders; CREATE TABLE Orders (  Transaction\_ID VARCHAR (10) NOT NULL,  Date\_Of\_Purchase DATE NOT NULL,  Amount INT (5) NOT NULL,  Customer\_ID VARCHAR (6) NOT NULL,  CONSTRAINT Transaction\_ID PRIMARY KEY (Transaction\_ID),  FOREIGN KEY (Customer\_ID)  REFERENCES Customer (Customer\_ID) ); |  |

# Insertion of Data in Tables

|  |  |
| --- | --- |
| Employee Table INSERT INTO Employee (Employee\_ID, Emp\_Name, Manager\_ID, Join\_Date, Designation, Branch\_ID)  VALUES  ('E00001', 'John Doe', NULL, '2020-01-01', 'Manager', 'B00001'),  ('E00002', 'Jane Smith', 'E00001', '2020-01-01', 'Assistant Manager', 'B00001'),  ('E00003', 'David Lee', 'E00001', '2020-02-01', 'Supervisor', 'B00001'),  ('E00004', 'Amy Johnson', 'E00003', '2020-02-01', 'Team Lead', 'B00002'),  ('E00005', 'Sarah Kim', 'E00003', '2020-03-01', 'Team Lead', 'B00002'),  ('E00006', 'Adam Park', 'E00002', '2020-04-01', 'Server', 'B00001'),  ('E00007', 'Peter Chen', 'E00002', '2020-04-01', 'Server', 'B00001'),  ('E00008', 'Mary Wong', 'E00003', '2020-05-01', 'Server', 'B00002'),  ('E00009', 'Linda Wang', 'E00003', '2020-05-01', 'Brewer', 'B00002'),  ('E00010', 'Tom Chang', 'E00002', '2020-06-01', 'Brewer', 'B00001'); | Customer Table INSERT INTO Customer (Customer\_ID, Name, Mode\_Of\_Payment, Ph\_No, Email)  VALUES  ('C00001', 'Emily Chen', 'Cash', '2345678901', '[emily.chen@example.com](mailto:emily.chen@example.com)'),  ('C00002', 'Jake Kim', 'Credit', '3456789012', '[jake.kim@example.com](mailto:jake.kim@example.com)'),  ('C00003', 'Grace Park', 'Cash', '4567890123', '[grace.park@example.com](mailto:grace.park@example.com)'),  ('C00004', 'Lucy Lee', 'Credit', '5678901234', '[lucy.lee@example.com](mailto:lucy.lee@example.com)'),  ('C00005', 'Michael Smith', 'Cash', '6789012345', '[michael.smith@example.com](mailto:michael.smith@example.com)'),  ('C00006', 'Olivia Johnson', 'Credit', '7890123456', '[olivia.johnson@example.com](mailto:olivia.johnson@example.com)'),  ('C00007', 'Henry Wong', 'Cash', '8901234567', '[henry.wong@example.com](mailto:henry.wong@example.com)'),  ('C00008', 'Ava Wang', 'Credit', '9012345678', '[ava.wang@example.com](mailto:ava.wang@example.com)'),  ('C00009', 'Ethan Liu', 'Cash', '1234567890', '[ethan.liu@example.com](mailto:ethan.liu@example.com)'),  ('C00010', 'Sophia Chang', 'Credit', '2345678901', '[sophia.chang@example.com](mailto:sophia.chang@example.com)'); |
| Supplier Table INSERT INTO Supplier (Supplier\_ID, Supplier\_Name, Address)  VALUES  ('S00001', 'ABC Company', '123 Main St'),  ('S00002', 'XYZ Corporation', '456 Elm St'),  ('S00003', 'Acme Inc.', '789 Oak St'),  ('S00004', 'Smith & Sons', '1010 Maple Ave'),  ('S00005', 'Jones Industries', '1111 Cedar Blvd'),  ('S00006', 'Globex Corporation', '1313 Mockingbird Ln'),  ('S00007', 'Initech', '555 Office Park Dr'),  ('S00008', 'Vandelay Industries', '2468 Broadway'),  ('S00009', 'Stark Industries', '10880 Malibu Point'),  ('S00010', 'Wayne Enterprises', '1007 Mountain Dr'); | Orders Table INSERT INTO Orders (Transaction\_ID, Date\_of\_Purchase, Amount, Customer\_ID)  VALUES  ('T000001', '2022-01-01', 50, 'C00001'),  ('T000002', '2022-01-02', 75, 'C00002'),  ('T000003', '2022-01-03', 100, 'C00003'),  ('T000004', '2022-01-04', 125, 'C00004'),  ('T000005', '2022-01-05', 150, 'C00005'),  ('T000006', '2022-01-06', 175, 'C00006'),  ('T000007', '2022-01-07', 200, 'C00007'),  ('T000008', '2022-01-08', 225, 'C00008'),  ('T000009', '2022-01-09', 250, 'C00009'),  ('T000010', '2022-01-10', 275, 'C00002'),  ('T000011', '2022-01-11', 375, 'C00010'),  ('T000012', '2022-01-12', 515, 'C00005'),  ('T000013', '2022-01-13', 200, 'C00005'),  ('T000014', '2022-01-14', 750, 'C00001'),  ('T000015', '2022-01-15', 398, 'C00004'); |
| Branch Table INSERT INTO Branch (Branch\_ID, Location)  VALUES  ('B0001', 'Boston'),  ('B0002', 'London'); | Product Table INSERT INTO Product (Product\_ID, Product\_Name, Price)  VALUES  ('P0001', 'IPA', 15),  ('P0002', 'Lager', 14),  ('P0003', 'Stout', 16),  ('P0004', 'Wheat Beer', 15),  ('P0005', 'Pilsner', 12),  ('P0006', 'Brown Ale', 20),  ('P0007', 'Porter', 19),  ('P0008', 'Belgian Tripel', 10),  ('P0009', 'Hefeweizen', 13),  ('P0010', 'Sour Ale', 17); |
| Batch Table INSERT INTO Batch (Batch\_ID, Date\_Produced, Expiry\_Date, Product\_ID)  VALUES  ('B001001', '2022-01-01', '2023-01-01', 'P0010'),  ('B001002', '2022-01-01', '2023-01-01', 'P0001'),  ('B002003', '2022-01-01', '2023-02-01', 'P0002'),  ('B002004', '2022-01-01', '2022-07-01', 'P0006'),  ('B001005', '2022-01-01', '2023-01-01', 'P0003'),  ('B002006', '2022-01-01', '2023-01-01', 'P0007'),  ('B001007', '2022-01-01', '2022-09-31', 'P0004'),  ('B002008', '2022-01-01', '2023-01-01', 'P0008'),  ('B001009', '2022-01-01', '2023-01-01', 'P0005'),  ('B002010', '2022-01-01', '2023-01-01', 'P0009'); | Raw Material Table INSERT INTO RawMaterial (Type\_ID, Expiry\_Date)  VALUES  ('RM001', '2023-05-20'),  ('RM002', '2023-07-15'),  ('RM003', '2023-06-30'),  ('RM004', '2023-08-01'),  ('RM005', '2023-07-05'),  ('RM006', '2023-09-10'),  ('RM007', '2023-07-30'),  ('RM008', '2023-08-25'),  ('RM009', '2023-09-20'),  ('RM010', '2023-10-15'); |
| Brewer Table INSERT INTO Brewer (BEmployee\_ID, Salary)  VALUES  ('E00009', 50000),  ('E00010', 60000); | Server Table INSERT INTO Server (SEmployee\_ID, Hourly\_Rate)  VALUES  ('E00006', 30),  ('E00008',25),  ('E00007', 20); |
| Production Table INSERT INTO Production (Qty\_Produced, Batch\_ID, Type\_ID)    VALUES  (1000, 'B001001', 'RM001'),  (2000, 'B001002', 'RM002'),  (1500, 'B002003', 'RM003'),  (3000, 'B002004', 'RM004'),  (2500, 'B001005', 'RM005'),  (500, 'B002006', 'RM006'),  (800, 'B001007', 'RM007'),  (1200, 'B002008', 'RM008'); | Bill Table INSERT INTO Bill (Quantity, Date\_of\_Purchase, Amount, Type\_ID, Supplier\_ID)  VALUES  (1000, '2021-09-15', 2000, 'RM001','S00001'),  (715, '2022-02-10', 1500, 'RM002', 'S00001'),  (250, '2022-02-12', 400, 'RM001', 'S00002'),  (320, '2022-03-21', 800, 'RM002', 'S00003'),  (1570, '2022-03-01', 3200, 'RM001', 'S00010'); |
| Served Table INSERT INTO Served (No\_of\_Customers\_Served, Customer\_ID, SEmployee\_ID)  VALUES  (3,'C00001', 'E00006'),  (2,'C00002', 'E00006'),  (4,'C00003', 'E00007'),  (1,'C00004', 'E00007'),  (2,'C00005', 'E00008'),  (2,'C00006', 'E00008'),  (6,'C00006', 'E00006'); | Visit Table INSERT INTO Visit (No\_of\_Visits, Customer\_ID, Branch\_ID)  VALUES  (4, 'C00001', 'B0001'),  (2, 'C0002', 'B0002'),  (2, 'C0003', 'B0002'),  (2, 'C0004', 'B0001'),  (2, 'C0005', 'B0002'),  (2, 'C0006', 'B0001'),  (2, 'C0006', 'B0002'); |
| Stock Table INSERT INTO Stock (No\_of\_Prepared\_Batches, Batch\_ID, BEmployee\_ID)  VALUES  (1, 'B001001', 'E00009'),  (2, 'B001002', ' E00009'),  (1, 'B002003', ' E00010'),  (3, 'B002004', ' E00010'); |  |

# Data loaded into Database

|  |  |
| --- | --- |
| Table  Description automatically generated | Graphical user interface, application, table  Description automatically generated |
| Graphical user interface, text, application, table  Description automatically generated | Table  Description automatically generated |
| Table  Description automatically generated | Table  Description automatically generated |
| Graphical user interface, application, table  Description automatically generated | Table  Description automatically generated |
| Table  Description automatically generated | Graphical user interface, application, table  Description automatically generated |
| Graphical user interface, text, application, email  Description automatically generated | Graphical user interface, application, table  Description automatically generated |
| Graphical user interface, application  Description automatically generated | Graphical user interface, application, table  Description automatically generated |
| Table  Description automatically generated |  |

# Querying in the Database

1. **Which Branch of NU Brewery has the most visited customers?**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, table

Description automatically generated

London Branch has the most visited customers.

1. **Get the names of Suppliers and the Total Amount where Quantity ordered is more than 500 from each.**

Graphical user interface, text, application

Description automatically generated

ABC Company & Wayne Enterprises.

1. **Find the most underperforming Server with his details?**

Graphical user interface, text, application, email

Description automatically generated

Mary Wong from London Branch is the underperforming Server.

# Assumptions

1. Only one Phone Number is taken from a customer, and it is not a multivalued attribute.
2. All other employees other than Brewer and Server do not have Salary. They have Stake in the company.

# Learnings from Project

* 1. Proper Plan and entity establishment must be done before we continue to Relational Model.
  2. Attribute names should not have space in between. We re-did the ERD Charts & Relational Model after we got error in SQL.
  3. Sequence of Table creation is crucial. So, we thought of creating Tables without Foreign key first and then create those tables which are dependent.
  4. It takes time, effort and especially Team work to build a database and design where data must be stored.
  5. Normalizing the data into 3NF really showed us an effective way to store the data.