Database Design Guide

This guide will help the owner to create a database on the Inventory Management System. It will help to manage the below functionalities.

- Supplier Details
- Product Details
- Customer Details
- Sales Details
- Defective Products

We will use MySQL as the DBMS to create the database and its related operations.

1. Introduction to MySQL:

MySQL is an open-source relational database management system (RDBMS) that uses structured query language (SQL) to manage and manipulate data in a database. It is widely used for various applications, from small web applications to large enterprise systems.

MySQL's key features include:

- Scalability: Capable of handling large amounts of data and concurrent connections.
- Flexibility: Supports various data types and storage engines.
- Performance: Optimized for speed and efficiency.
- Reliability: Known for its stability and robustness.

2. Installation of MySQL:

MySQL can be installed on various operating systems, including Windows, macOS, and Linux. Here are the general steps to install MySQL:

Windows:

- Download the MySQL installer from the official website. https://dev.mysql.com/downloads/installer/
- Run the installer and follow the on-screen instructions.
- Choose the installation type (Typical, Complete, or Custom). Recommended Custom.
- Set a root password for the MySQL server.

3. E-R Diagram (ERD):

An Entity-Relationship Diagram (ERD) is a visual representation of the data model that shows the entities, attributes, relationships between entities, and cardinality. ERDs are commonly used in database design to help developers and stakeholders understand the structure and relationships within a database.

i) Identify Entities:

- Start by identifying the main entities in your system. These are the objects or concepts about which you want to store data.
- Each entity should correspond to a table in your database.

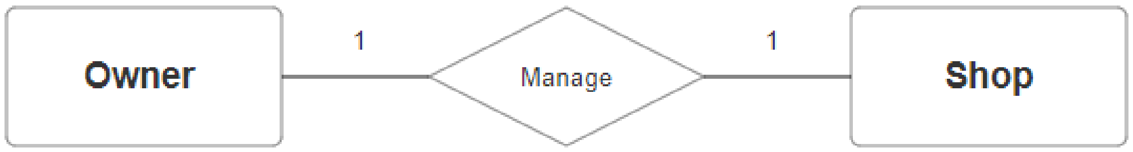
ii) Define Attributes:

- For each entity, list the attributes (properties or fields) that describe it.
- These attributes will become columns in the corresponding database table.

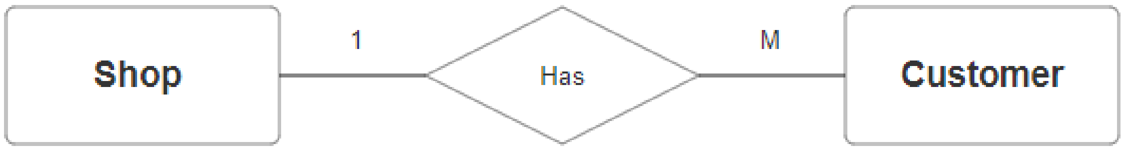
iii) Identify Relationships:

- Determine how entities are related to each other. There are three types of relationships: one-to-one (1:1), one-to-many (1:M), and many-to-many (M:M).
- Represent these relationships using lines connecting the entities.
- Let's see a few examples of relationships:

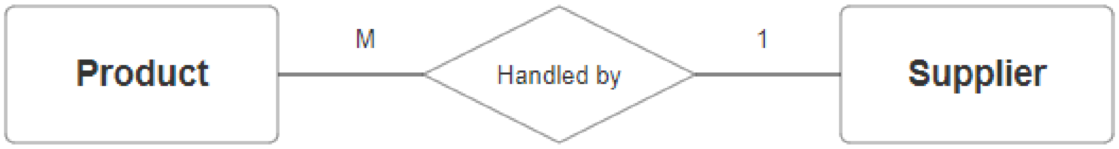
One to One



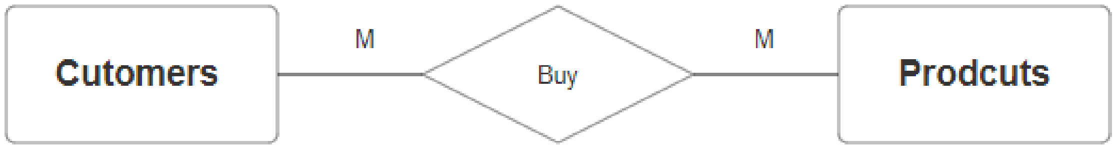
One to Many



Many to One



Many to Many



iv) Optional:

a) **Add Attributes and Constraints:**
   - Include additional information in your ERD, such as primary keys, foreign keys, and constraints (e.g., unique constraints).

b) **Create the Diagram**:
   - Use specialized diagramming software or tools (e.g., Lucidchart, draw.io, or even pen and paper) to create your ERD.

c) **Refine and Review:**
   - Review your ERD with stakeholders and team members to ensure it accurately represents the data model and relationships. Make any necessary refinements.

4. Let's identify the attributes and relationships of each entities for the Inventory Management System:

1) <u>Supplier:</u>

- **Attributes:**
  SupplierId (Primary Key)
  SupplierName
  SupplierAddress
  SupplierMobileNumber

- **Relationships:**

One **Supplier** can have multiple **products (One-to-Many)**

2) <u>Products:</u>

- **Attributes**:
  ProductId (Primary Key)
  ProductName
  ProductDescription
  ProductCategory
  ProductQuantity
  ProductPrice

- **Relationships:**
  Many **Products** can be in single **category** (**Many-to-One**)
  Many **Products** can have single **supplier (One-to-Many)**

3) <u>Customer:</u>

- **Attributes:**
  CustomerId (Primary Key)
  CustomerName
  CustomerAddress
  CustomerMobileNumber

- **Relationships:**
  One **Customer** can buy multiple **products** (**One-to-Many**)
  One **Customer** can buy single **product** also **(One-to-One)**

4) <u>Sales:</u>

- **Attributes:**
  SaleId (Primary Key)
  PurchaseDate
  TotalAmount
  ProductId (Foreign key)
  CustomerId (Foreign Key)

- **Relationships:**
  One **Sale** can placed by multiple **customer** (**One-to-Many**)
  Many **Sale** can placed through single **customer** (**Many-to-One**)

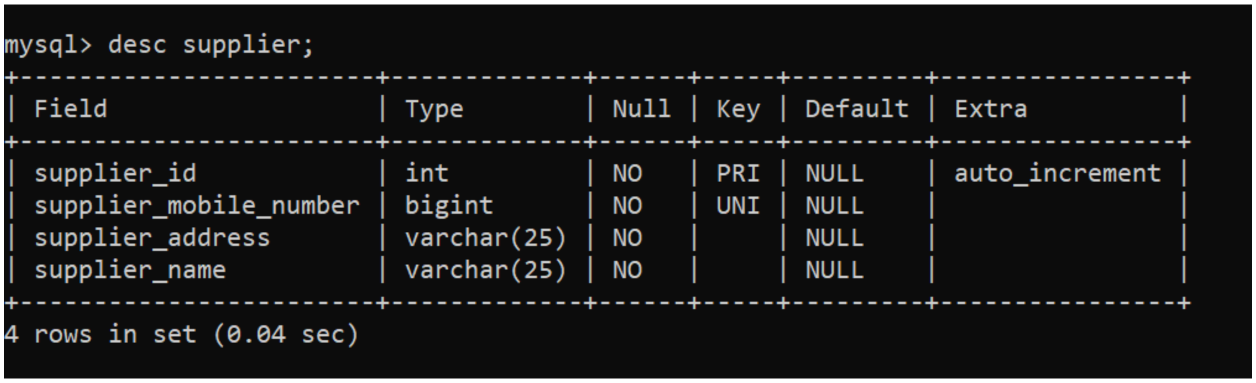
5) Defective:

- **Attributes**:
  DefectiveItemsId (Primary key)
  DefectiveQuantity
  ProductId (Foreign key)
  SupplierId (Foreign key)

- **Relationships**:
  Many **Defective** quantity will be there of single **Product** (**One-to-Many**)

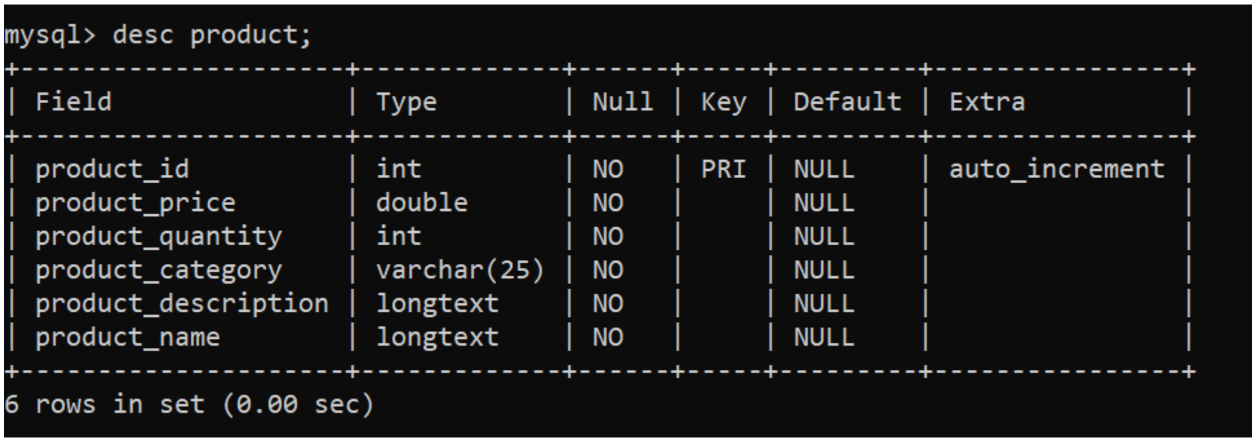
5. Table Structure:

1) Supplier Details:



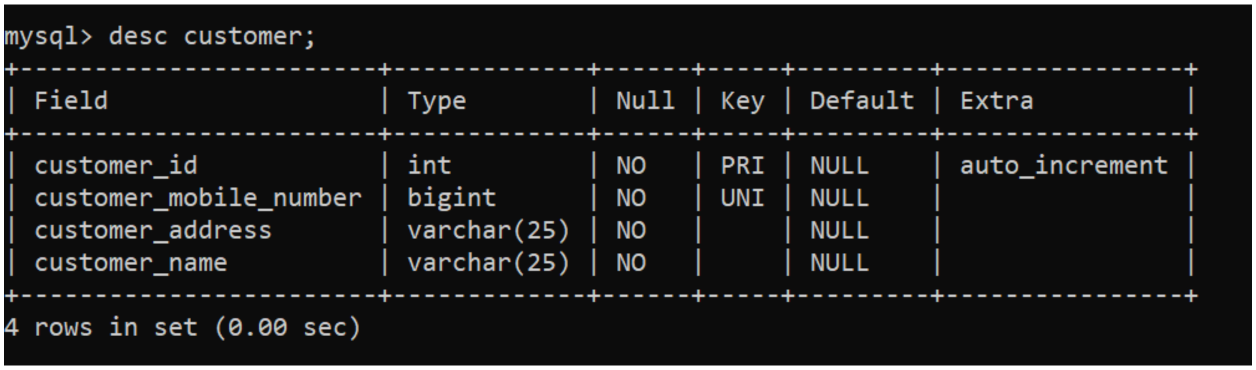
```
mysql> desc supplier;
+------------------------+-------------+------+-----+---------+----------------+
| Field                  | Type        | Null | Key | Default | Extra          |
+------------------------+-------------+------+-----+---------+----------------+
| supplier_id            | int         | NO   | PRI | NULL    | auto_increment |
| supplier_mobile_number | bigint      | NO   | UNI | NULL    |                |
| supplier_address       | varchar(25) | NO   |     | NULL    |                |
| supplier_name          | varchar(25) | NO   |     | NULL    |                |
+------------------------+-------------+------+-----+---------+----------------+
4 rows in set (0.04 sec)
```

2) Product Details:

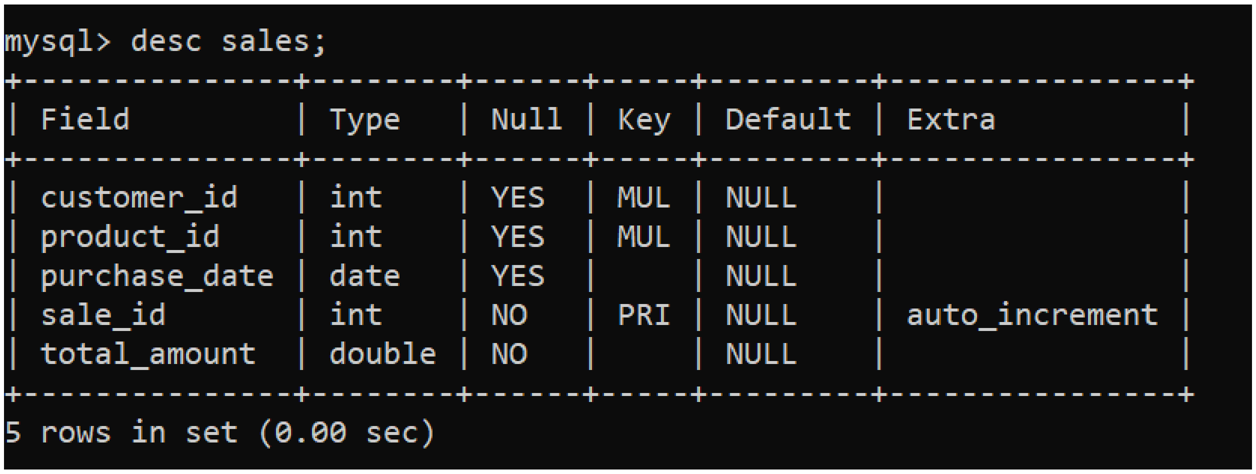


```
mysql> desc product;
+---------------------+-------------+------+-----+---------+----------------+
| Field               | Type        | Null | Key | Default | Extra          |
+---------------------+-------------+------+-----+---------+----------------+
| product_id          | int         | NO   | PRI | NULL    | auto_increment |
| product_price       | double      | NO   |     | NULL    |                |
| product_quantity    | int         | NO   |     | NULL    |                |
| product_category    | varchar(25) | NO   |     | NULL    |                |
| product_description | longtext    | NO   |     | NULL    |                |
| product_name        | longtext    | NO   |     | NULL    |                |
+---------------------+-------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

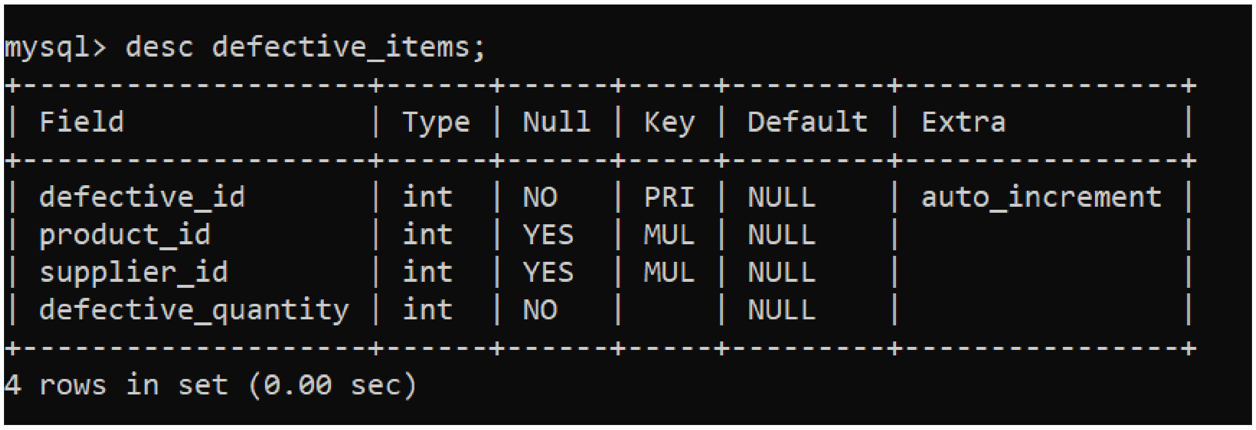
3) Customer Details:


```
mysql> desc customer;
+------------------------+-------------+------+-----+---------+----------------+
| Field                  | Type        | Null | Key | Default | Extra          |
+------------------------+-------------+------+-----+---------+----------------+
| customer_id            | int         | NO   | PRI | NULL    | auto_increment |
| customer_mobile_number | bigint      | NO   | UNI | NULL    |                |
| customer_address       | varchar(25) | NO   |     | NULL    |                |
| customer_name          | varchar(25) | NO   |     | NULL    |                |
+------------------------+-------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

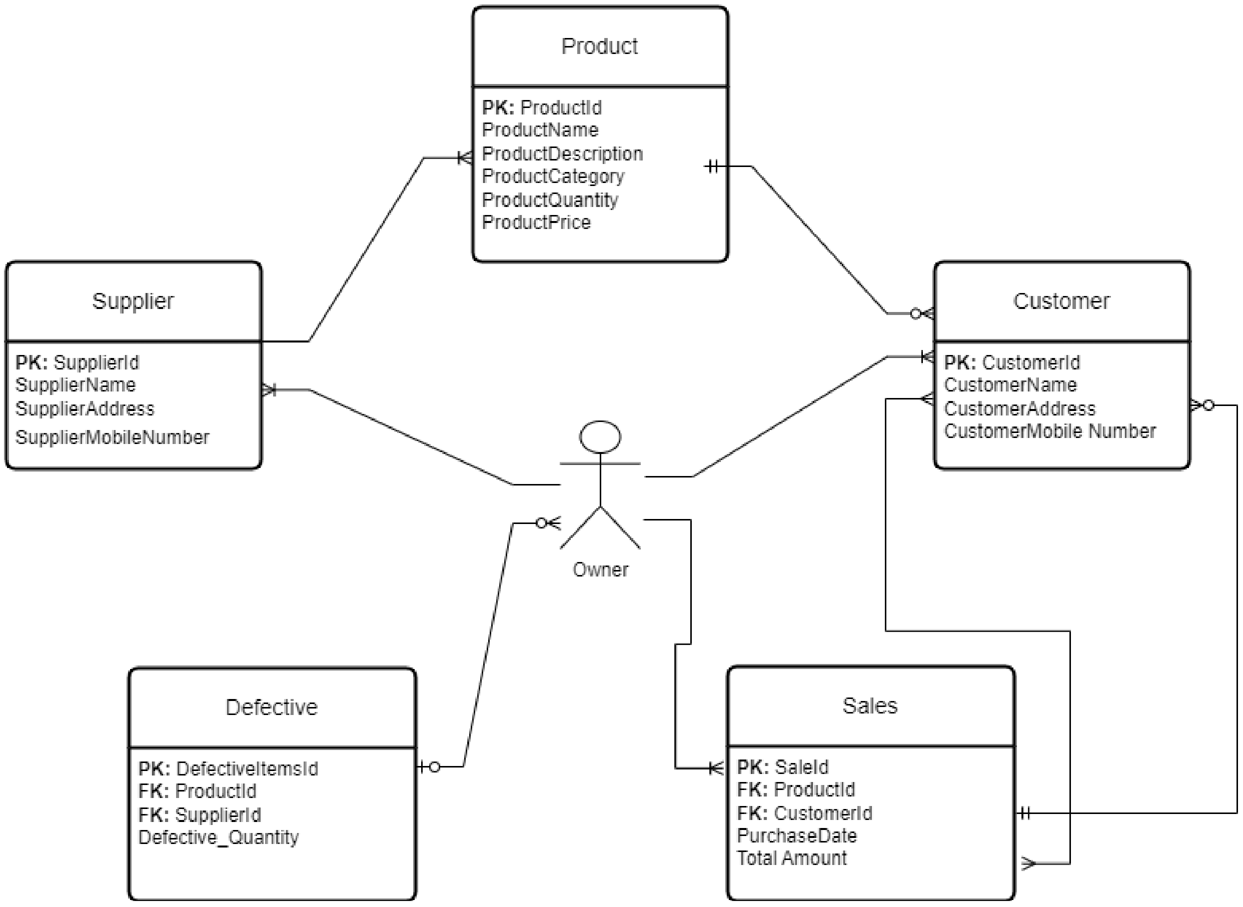
4) Sales Details:


```
mysql> desc sales;
+---------------+--------+------+-----+---------+----------------+
| Field         | Type   | Null | Key | Default | Extra          |
+---------------+--------+------+-----+---------+----------------+
| customer_id   | int    | YES  | MUL | NULL    |                |
| product_id    | int    | YES  | MUL | NULL    |                |
| purchase_date | date   | YES  |     | NULL    |                |
| sale_id       | int    | NO   | PRI | NULL    | auto_increment |
| total_amount  | double | NO   |     | NULL    |                |
+---------------+--------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

5) Defective Products:


```
mysql> desc defective_items;
+--------------------+------+------+-----+---------+----------------+
| Field              | Type | Null | Key | Default | Extra          |
+--------------------+------+------+-----+---------+----------------+
| defective_id       | int  | NO   | PRI | NULL    | auto_increment |
| product_id         | int  | YES  | MUL | NULL    |                |
| supplier_id        | int  | YES  | MUL | NULL    |                |
| defective_quantity | int  | NO   |     | NULL    |                |
+--------------------+------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

6. **Let's create the ER diagram to visually represent the entities and relationships:**



- **In this ERD:**
  1. Owner have multiple supplier and multiple customer, creating a many-to-many relationship.
  2. Multiple products deliver by on supplier (many-to-one relationship).
  3. Each Customer buy multiple products (one-to-many relationship).
  4. Customer can order one or more than one product.
  5. One or Many products may be defective.

7. **Creating a Database:**
Using MySQL server, create a new database for your Inventory management system. You can do this with SQL commands or through the graphical interface.

```
CREATE DATABASE IMS;
```

8. **Using a Database:**
   Before performing any operations on a database, you need to select it using the USE statement:

```
USE IMS;
```

9. **Creating the tables for each entity:**

```
CREATE TABLE Supplier_Details
(Supplier_Id INT(10) PRIMARY KEY,
Supplier_Name VARCHAR(100),
Supplier_Address VARCHAR(100),
Supplier_MobileNumber BIGINT(10));

CREATE TABLE Products_Details
(Product_Id INT(10) PRIMARY KEY,
Product_Name VARCHAR(100),
Product_Description VARCHAR(100),
Product_Category VARCHAR(100),
Product_Quantity BIGINT(50),
Product_Price BIGINT(50));

CREATE TABLE Customer_Details
(Customer_Id INT(10) PRIMARY KEY,
Customer_Name VARCHAR(100),
Customer_Address VARCHAR(200),
Customer_MobileNumber BIGINT(10));

CREATE TABLE Sales_Details
(Sale_Id INT(10) PRIMARY KEY,
Purchase_Date DATE,
Total_Amount BIGINT(50),
Product_Id INT(10),
Customer_Id INT(10)
FOREIGN KEY (Product_Id) REFERENCES Product_Details (Product_Id),
FOREIGN KEY (Customer_Id) REFERENCES Customer_Details (Customer_Id));

CREATE TABLE Defective_Items
(Defective_Items_Id INT(10) PRIMARY KEY,
Defective_Quantity BIGINT(25),
Product_Id VARCHAR(10),
Supplier_Id VARCHAR(10),
FOREIGN KEY (Product_Id) REFERENCES Product_Details (Product_Id),
FOREIGN KEY (Supplier_Id) REFERENCES Supplier_Details (Supplier_Id));
```

10. Insert records:

Add data to your tables to work with. This step helps you test your database.

```
INSERT INTO Supplier_Details
(Supplier_Id, Supplier_Name, Supplier_Address, Supplier_MobileNumber)
VALUES
('S01', 'Kartik_Pawar', 'Babhaleshwar', '7840975898'),
('S02', 'Ishwar Mhase', 'Rahuri', '7746894589');

INSERT INTO Product_Details
(Product_Id, Product_Name, Product_Description, Product_Category, Product_Quantity,
Product_Price)
VALUES
('P01', 'IQOO Z6 5G', 'Snapdragon processor', 'Mobile', '20', '18999'),
('P02', 'Thomson Android Smart TV', 'Android 11 5000+ Apps Support', 'Appliances', '05',
'25000');

INSERT INTO Customer_Details
(Customer_Id, Customer_Name, Customer_Address, Customer_MobileNumber)
VALUES
('C01', 'Bipin_Pawar', 'Shrirampur', '8446435328'),
('C02', 'Sujit Shete', 'Shirdi', '8974943569');

INSERT INTO Sales_Details
(Sale_Id, Purchase_Date, Total_Amount, Product_Id, Customer_Id)
VALUES
('S01', '05/12/2023', '18999', 'P01', 'C02'),
('S02', '05/12/2023', '25000', 'P02', 'C01');

INSERT INTO Defective_Items
(Defective_Items_Id, Defective_Quantity, Product_Id, Supplier_Id)
VALUES
('D01', '2', 'P02', 'S01');
```

11. Select records:

Write SQL queries to retrieve and manage data.

Retrieve all customer:

Select * FROM Customer_Details;

Retrieve a specific product:
Select * from Product_Details where Product_Name=' IQOO Z6 5G';
Select Product_Quantity from Product_Details where Product_Name=' IQOO Z6 5G';

Now try similar Select queries with other tables

12. Update records:

Write SQL statements to update record(s) when needed. For example:

Update supplier mobile number:
Update Supplier_Details SET Supplier_MobileNumber = '7868456795' Where Supplier_Name = 'Kartik_Pawar';

13. Delete records:

Write SQL statements to delete record(s) when needed.

Delete FROM Product_Details Where Product_Name = ‘Thomson Android Smart TV’;

PN: Ideally no data should be deleted from any tables. You can use an additional column to set the status of that record to ‘Active/Inactive’, etc. Or you can use an Archive table to move the unnecessary records out of the main table.