Submitted To

Md Samsuddoha

Assistant Professor,

Dept.Of CSE

University Of Barishal

Submitted By

Reshma Akter

Batch:Database-2

Roll:05-002-02

1.Designing (Entity Relationship)ER Diagram

Steps of Drawing ERD :

1. **Identify the Entities Required**
2. **Identify the Attributes and Primary key for each Entity**
3. **Identify the Relationship needed**
4. **Identify the Cardinality Ratio and Participation**
5. **Draw the Diagram**

Project: Virtual plant map management system.

A **Virtual Plant Map Management System (VPMMS)** is an online application designed to manage and visualize the layout of industrial plants or manufacturing facilities. This system can track various components such as equipment, machinery, inventory, rooms, zones, and utilities in a plant, allowing users to interact with a digital representation of the facility.

Key Features:

- **Visualization of plant layout**: Create and display digital maps of plants.
- **Asset management**: Track and manage equipment and machinery.
- **Inventory management**: Keep track of spare parts and materials.
- **Maintenance tracking**: Schedule and monitor the status of equipment maintenance.
- **User roles and access management**: Define different roles like Admin, Engineer, and Manager with different access levels.
- **Interactive dashboard**: Allow users to click on areas of the plant map for detailed information and data.

Step-1: Identify the Entities Required.

For the **Virtual Plant Map Management System**, the primary entities involved would include:

1. **Plant**
2. **Room**
3. **Zone**
4. **Equipment**
5. **Inventory**
6. **Maintenance**
7. **User**
8. **Asset**

Step-2: Identify the Attributes and Primary key for each Entity.

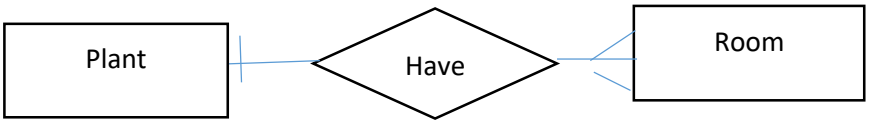
1. **Plant:** Plant_ID (PK), Name, Location, Area, Description.
2. **Room:** Room_ID (PK), Room_Name, Room_Type, Dimensions, Plant_ID (FK).
3. **Zone:** Zone_ID (PK), Zone_Name, Description, Plant_ID (FK).
4. **Equipment:** Equipment_ID (PK), Equipment_Name, Equipment_Type, Manufacturer, Status, Room_ID (FK).
5. **Inventory:** Inventory_ID (PK), Item_Name, Item_Type, Quantity, Location (FK).
6. **Maintenance:** Maintenance_ID (PK), Equipment_ID (FK), Maintenance_Type, Scheduled_Date, Completion_Date, Status.
7. **User:** User_ID (PK), Username, Password, Role, Full_Name, Email, Plant_ID (FK).
8. **Asset:** Asset_ID (PK), Asset_Name, Asset_Type, Plant_ID (FK), Equipment_ID (FK), Value.

Step-3: Identify the Relationship needed

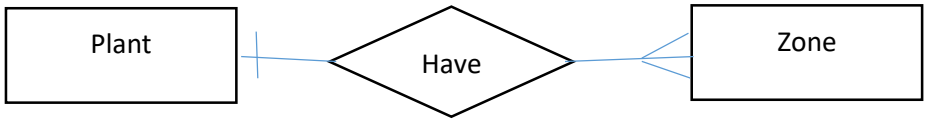
1. **Plant to Room**: A one-to-many relationship (One plant can have multiple rooms).
2. **Plant to Zone**: A one-to-many relationship (One plant can have multiple zones).
3. **Room to Equipment**: A one-to-many relationship (One room can contain many pieces of equipment).
4. **Zone to Equipment**: A many-to-many relationship (A zone can have multiple equipment, and equipment can be used across multiple zones).
5. **Equipment to Maintenance**: A one-to-many relationship (Each piece of equipment may have multiple maintenance records).
6. **Inventory to Equipment**: A one-to-many relationship (Inventory items are used to support equipment maintenance).
7. **User to Plant**: A many-to-one relationship (Users belong to a specific plant).
8. **Asset to Equipment**: A one-to-one or one-to-many relationship (An asset can be linked to one or more pieces of equipment).

Step-4: Identify the Cardinality Ratio and Participation.

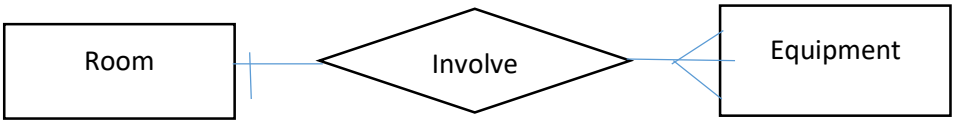
1.Plant to Room: A one-to-many relationship.



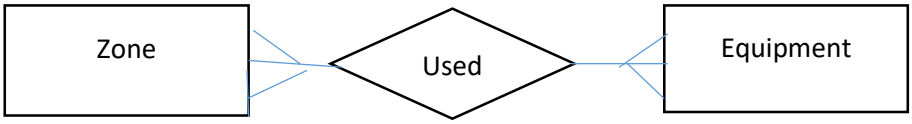
2.Plant to Zone: A one-to-many relationship.



3.Room to Equipment: A one-to-many relationship.

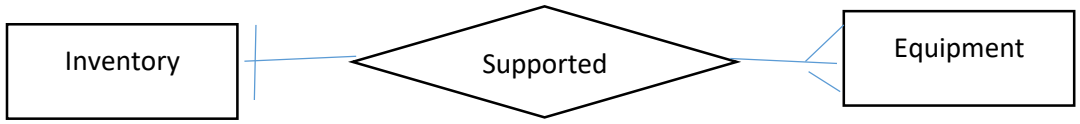


4.Zone to Equipment: A many-to-many relationship.

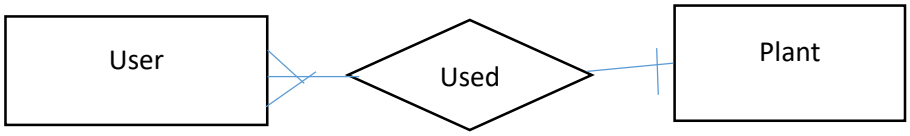


5.Equipment to Maintenance: A one-to-many relationship.

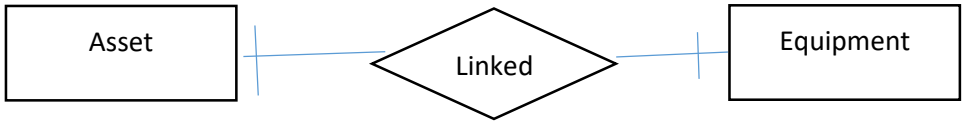
6.Inventory to Equipment: A one-to-many relationship.



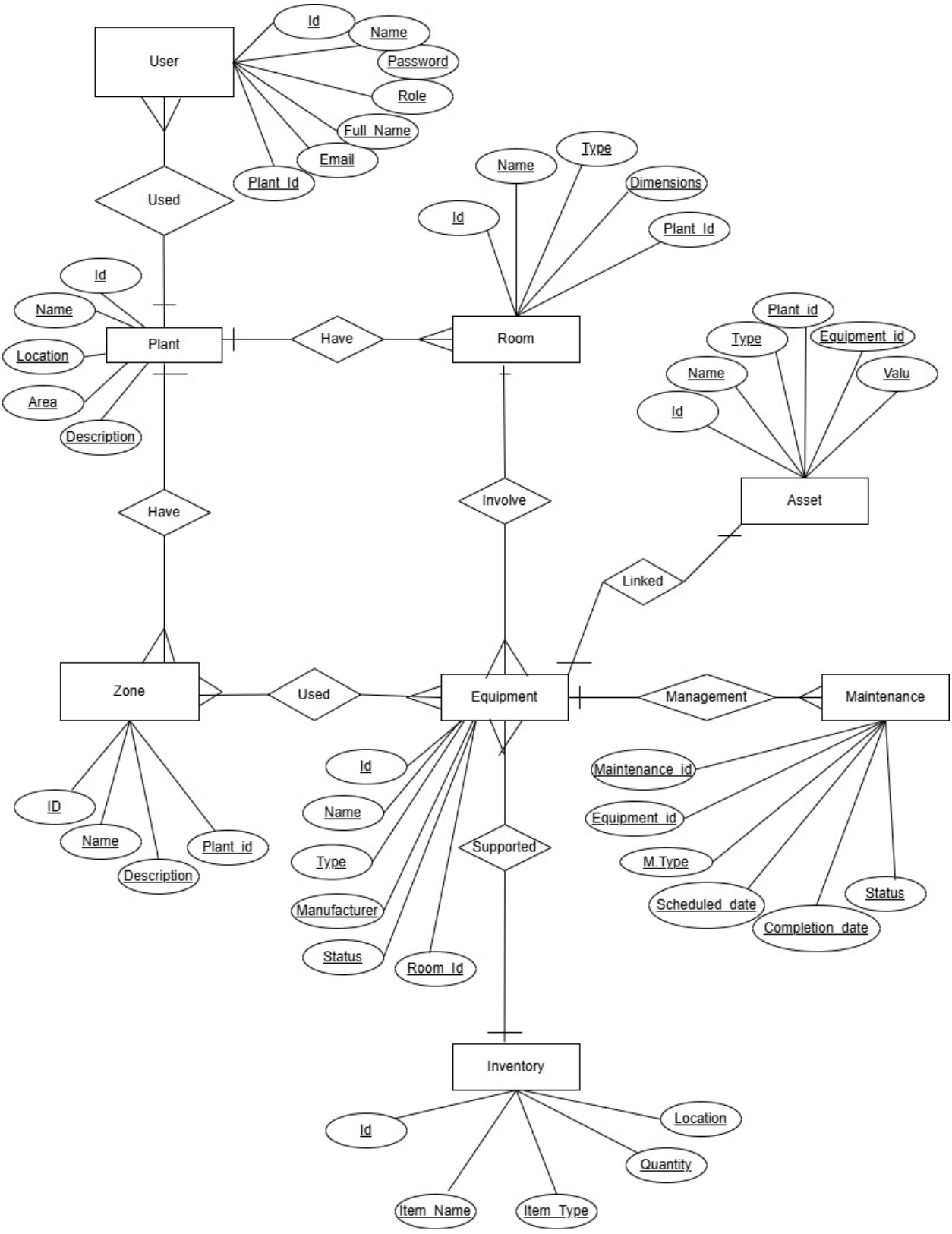
7.User to Plant: A many-to-one relationship.



8.Asset to Equipment: A one-to-one or one-to-many relationship.



Step-5: Draw the Diagram.



2. Reduction to database schema:

1.Plant: Plant_ID (PK), Name, Location, Area, Description.

2.Room: Room_ID (PK), Room_Name, Room_Type, Dimensions, Plant_ID (FK).

3.Zone: Zone_ID (PK), Zone_Name, Description, Plant_ID (FK).

4.Equipment: Equipment_ID (PK), Equipment_Name, Equipment_Type, Manufacturer, Status, Room_ID (FK).

5.Inventory: Inventory_ID (PK), Item_Name, Item_Type, Quantity, Location (FK).

6.Maintenance: Maintenance_ID (PK), Equipment_ID (FK), Maintenance_Type, Scheduled_Date, Completion_Date, Status.

7.User: User_ID (PK), Username, Password, Role, Full_Name, Email, Plant_ID (FK).

8.Asset: Asset_ID (PK), Asset_Name, Asset_Type, Plant_ID (FK), Equipment_ID (FK), Value.

3.Implementing the database in MySQL:

All tables with sample data:

1.Plant: Plant_ID (PK), Name, Location, Area, Description.

```
CREATE TABLE Plant (
    Plant_ID INT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    Location VARCHAR(255),
    Area DECIMAL(10, 2),  -- Assuming area is in square meters or square feet
```

Description TEXT

);

2.Room: Room_ID (PK), Room_Name, Room_Type, Dimensions, Plant_ID (FK).

```
CREATE TABLE Room (
    Room_ID INT PRIMARY KEY,
    Room_Name VARCHAR(255) NOT NULL,
    Room_Type VARCHAR(100),
    Dimensions VARCHAR(100),  -- Can store dimensions like "10x20 meters"
    Plant_ID INT,
    FOREIGN KEY (Plant_ID) REFERENCES Plant(Plant_ID) ON DELETE CASCADE
);
```

3.Zone: Zone_ID (PK), Zone_Name, Description, Plant_ID (FK).

```
CREATE TABLE Zone (
    Zone_ID INT PRIMARY KEY,
    Zone_Name VARCHAR(255) NOT NULL,
    Description TEXT,
    Plant_ID INT,
    FOREIGN KEY (Plant_ID) REFERENCES Plant(Plant_ID) ON DELETE CASCADE
);
```

4.Equipment: Equipment_ID (PK), Equipment_Name, Equipment_Type, Manufacturer, Status, Room_ID (FK).

```
CREATE TABLE Equipment (
    Equipment_ID INT PRIMARY KEY,
    Equipment_Name VARCHAR(255) NOT NULL,
    Equipment_Type VARCHAR(100),
    Manufacturer VARCHAR(255),
    Status VARCHAR(50),  -- Status like "Active", "Under Maintenance"
    Room_ID INT,
    FOREIGN KEY (Room_ID) REFERENCES Room(Room_ID) ON DELETE CASCADE
);
```

5.Inventory: Inventory_ID (PK), Item_Name, Item_Type, Quantity, Location (FK).

```
CREATE TABLE Inventory (
    Inventory_ID INT PRIMARY KEY,
    Item_Name VARCHAR(255) NOT NULL,
    Item_Type VARCHAR(100),
    Quantity INT,
    Location VARCHAR(255),  -- Location could refer to a room or zone
    FOREIGN KEY (Location) REFERENCES Room(Room_ID)  -- Assuming Location refers to Room_ID for simplicity
);
```

6.Maintenance: Maintenance_ID (PK), Equipment_ID (FK), Maintenance_Type, Scheduled_Date, Completion_Date, Status.

```
CREATE TABLE Maintenance (
    Maintenance_ID INT PRIMARY KEY,
    Equipment_ID INT,
    Maintenance_Type VARCHAR(100),
    Scheduled_Date DATE,
    Completion_Date DATE,
    Status VARCHAR(50),
    FOREIGN KEY (Equipment_ID) REFERENCES Equipment(Equipment_ID) ON DELETE CASCADE
);
```

7.User: User_ID (PK), Username, Password, Role, Full_Name, Email, Plant_ID (FK).

```
CREATE TABLE User (
    User_ID INT PRIMARY KEY,
    Username VARCHAR(255) NOT NULL,
    Password VARCHAR(255) NOT NULL,
    Role VARCHAR(50),  -- e.g., Admin, Manager, Engineer
    Full_Name VARCHAR(255),
```

```
    Email VARCHAR(255),
    Plant_ID INT,
    FOREIGN KEY (Plant_ID) REFERENCES Plant(Plant_ID) ON DELETE CASCADE
);
```

8.Asset: Asset_ID (PK), Asset_Name, Asset_Type, Plant_ID (FK), Equipment_ID (FK), Value.

```
CREATE TABLE Asset (
    Asset_ID INT PRIMARY KEY,
    Asset_Name VARCHAR(255) NOT NULL,
    Asset_Type VARCHAR(100),
    Plant_ID INT,
    Equipment_ID INT,
    Value DECIMAL(15, 2),  -- Monetary value of the asset
    FOREIGN KEY (Plant_ID) REFERENCES Plant(Plant_ID) ON DELETE CASCADE,
    FOREIGN KEY (Equipment_ID) REFERENCES Equipment(Equipment_ID) ON DELETE CASCADE
);
```