**Database Design Guide**

This guide will help the student to create a database on the Farmer’s Buddy.

It is allow user to buy the products online.It will help to manage the below functionalities.

**High Level Functionality:-**

**1.Admin**

**Functionality:**

* **Manage the system:** Add,update,delete product . ship the orders**.**

**2.User Details**

**Functionality:**

* **User Registration and Authentication**: Secure user registration, login, and password management.
* **User Roles :** User will buy the products. Availabe on the website.
* **Admin Roles**: Admin will add the products. For selling.

**3.Product Details**

**Functionality:**

* **Product Catalog**: Display a searchable and filterable list of products.
* **Product Management**: Allow admin to add, update, and delete products.

**4. Cart**

**Functionality:**

* **Add to Cart**: Allow users to add products to their cart.
* **View Cart**: Users can view items in their cart, update quantities, and remove items.

**5 .Payment**

**Functionality:**

* User can do the payment of the products. Fill the necessary details.

**6. Order Details**

**Functionality:**

* **Order History**: Users can view their past orders.
* **Order Status**: Provide Status of the order.

### 7.Advisory Posts

#### Functionality:

* Posts on topics such as solutions for the diseases, pest control,etc.

**8. Feedback**

**Functionality:**

* **Feedback Submission**: Allow users to submit feedback on products, and overall experience of the system.
* **Feedback Management**: Admin can view the feedback given by the user.

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

**Front-end :-** angular

**Back-end :-** springboot

**Database :-** mysql

**Entities**

 **User Details**

 **Product Details**

 **Cart**

 **Order**

 **Payment**

 **Feedback**

**Define Attributes**

* **User**:
  + farmersId (PK)
  + fName
  + password
  + email - (username)
  + phonNumber
  + address
* **Product**:
  + productId (PK)
  + name
  + descriptions
  + price
  + quantity
  + image
* **Cart**:
  + cartId (PK)
  + farmersId (FK)
  + productId (FK)
  + quantity

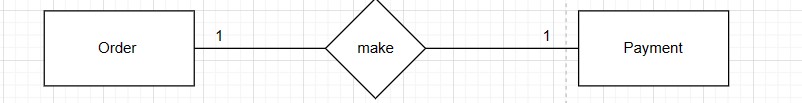
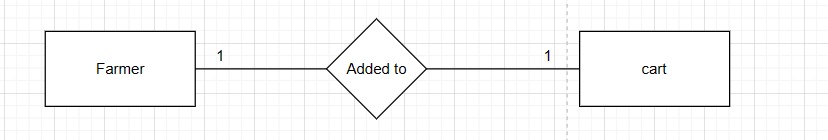
 **Orders**:

* + orderId (PK)
  + farmersId (FK)
  + productId (FK)
  + paymentId (FK)
  + orderDate
  + totalAmount
  + status
  + quantity
  + price
* **Payment**:
  + paymentId (PK)
  + farmerId(FK)
  + orderId (FK)
  + cardholderName
  + cardNumbr
  + paymentDate
  + amount
* **Feedback**:
  + feedbackId (PK)
  + farmersId (FK)
  + rating
  + comment

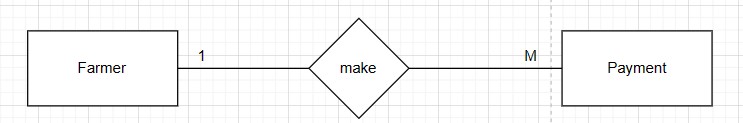
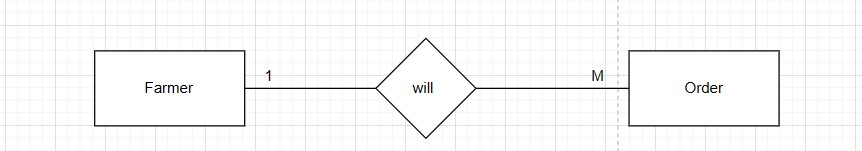
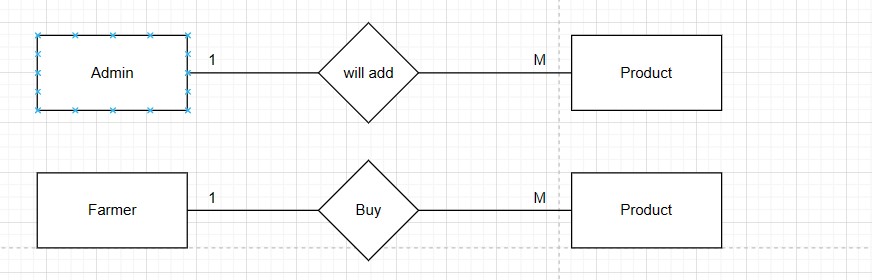
**Identify Relationships**

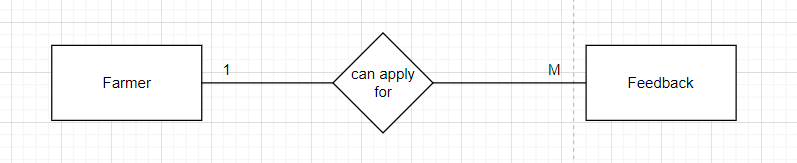
Let’s see a few examples of relationships:

**One to One**

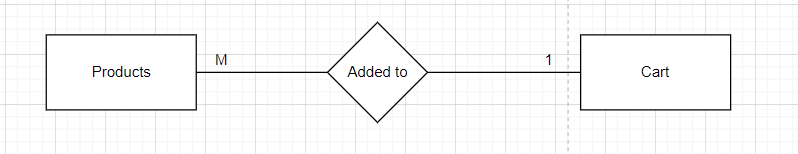


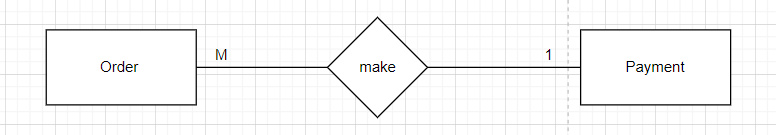
**One to Many**





**Many to One**





\*\*\* Now let’s identify the attributes and relationships of each entity for Farmer’s Buddy system.

* **User**:
  + farmersId (PK)
  + username
  + password
  + email
  + phonNumber
  + address

**Relationship:**

 One user can have many orders (one-to-many with Orders table).

 One user can have many feedback entries (one-to-many with Feedback table).

* **Product**:
  + productId (PK)
  + name
  + descriptions
  + price
  + quantity
  + image

**Relationship:**

 One product can be in many carts (one-to-many with Cart table).

 One product can be in many orders (one-to-many with Orders table).

* **Cart**:
  + cartId (PK)
  + farmersId (FK)
  + productId (FK)
  + quantity

**Relationship:**

 many products in one user cart(many-to-one with Users table).

 A Cart can have multiple Products. (one-to-many with Products table).

 **Orders**:

* + orderId (PK)
  + farmersId (FK)
  + productId (FK)
  + paymentId (FK)
  + order\_date
  + total\_amount
  + status
  + quantity
  + price

**Relationship:**

 Many orders can be done by one user (many-to-one with users table)

 One Order can contain multiple Products.(One-to-many with products)

 One Order can have one Payment.(one-to-many with payment)

* **Payment**:
  + paymentId (PK)
  + username(FK)
  + orderId (FK)
  + cardholderName
  + accountNumbr
  + paymentDate
  + amount

**Relationship:**

 many Payment belongs to one User.(many-to-one with users)

 one Payment is associated with one Order.(one-to-one with order)

* **Feedback**:
  + feedbackId (PK)
  + farmersId (FK)
  + rating
  + comment

**Relationship:**

 many Feedback belongs to one User.(many-to-one with users)

Now, let’s create the ER diagram to visually represent the entities and relationships.

**ERD Diagram**

