Project Report on

Farmer Management System

By

MOHD SADIQ

23BCA10779

For

Course: Database Management Systems

Department: BCA

Institution: Chandigarh University

Date: April 2025

Table of Contents

1. Introduction

The **Farmer Management System** is a database project aimed at storing and managing farmer-related data efficiently. It helps in organizing critical information such as farmer details, land records, crops grown, agricultural tools used, and subsidies received from the government. The system is built to ensure smooth access, proper management, and improved service delivery to farmers.

---

2. Objective

- To create a centralized database for storing farmer-related data.
- To reduce manual work and increase efficiency in managing agricultural data.
- To simplify the tracking of land, crops, tools, and subsidies.
- To enable government and agricultural agencies to monitor support services.
- To design and implement an ER-based relational database system.

3. Tables

Table 1: Farmer

Column	Data Type
FarmerID	INT (Primary Key)
Name	VARCHAR(50)
Age	INT
Contact	VARCHAR(15)

Table 2: Land

Column	Data Type
LandID	INT (Primary Key)
FarmerID	INT (Foreign Key)
Location	VARCHAR(100)
Area	FLOAT

Table 3: Crop

Column	Data Type
CropID	INT (Primary Key)
LandID	INT (Foreign Key)
CropName	VARCHAR(50)

Column	Data Type
Season	VARCHAR(20)

Table 4: Tool

Column	Data Type
ToolID	INT (Primary Key)
FarmerID	INT (Foreign Key)
ToolName	VARCHAR(50)
Quantity	INT

Table 5: Subsidy

Column	Data Type
SubsidyID	INT (Primary Key)
FarmerID	INT (Foreign Key)
Type	VARCHAR(50)
Amount	DECIMAL(10,2)

4. ER Diagram

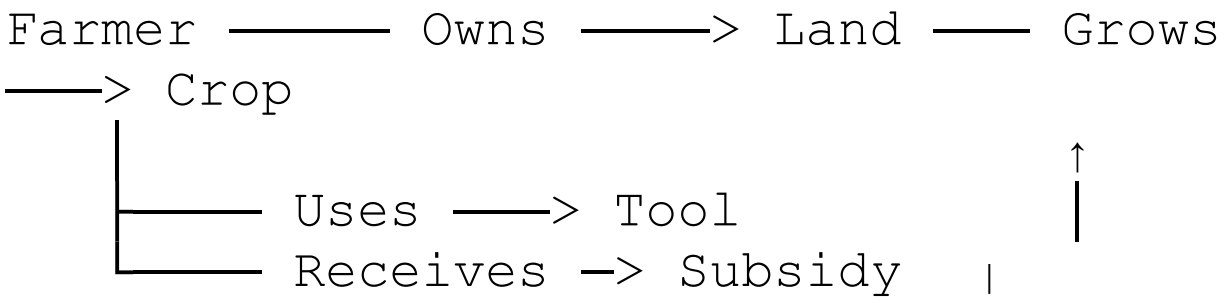
Entities:

- Farmer
- Land
- Crop
- Tool
- Subsidy

Relationships:

- A **Farmer** can own multiple **Lands**.
- A **Land** can have multiple **Crops**.
- A **Farmer** can have multiple **Tools**.
- A **Farmer** can receive multiple **Subsidies**.

ER Diagram Structure (text-based):



5. Technologies Used

Component	Technology
DBMS	MySQL / Oracle / SQLite
Backend Queries	SQL
Diagram Tool	Lucidchart / Draw.io
Development Platform	XAMPP / Localhost

6. Coding (SQL Commands)

```
 Creating Farmer Table
CREATE TABLE Farmer (
  FarmerID INT PRIMARY KEY,
  Name VARCHAR(50),
  Age INT,
  Contact VARCHAR(15)
);

 Creating Land Table
CREATE TABLE Land (
  LandID INT PRIMARY KEY,
  FarmerID INT,
  Location VARCHAR(100),
  Area FLOAT,
  FOREIGN KEY (FarmerID) REFERENCES Farmer(FarmerID)
);

Creating Crop Table
CREATE TABLE Crop (
  CropID INT PRIMARY KEY,
  LandID INT,
  CropName VARCHAR(50),
  Season VARCHAR(20),
  FOREIGN KEY (LandID) REFERENCES Land(LandID)
);

 Creating Tool Table
CREATE TABLE Tool (
  ToolID INT PRIMARY KEY,
  FarmerID INT,
  ToolName VARCHAR(50),
  Quantity INT,
  FOREIGN KEY (FarmerID) REFERENCES Farmer(FarmerID)
);

Creating Subsidy Table
CREATE TABLE Subsidy (
  SubsidyID INT PRIMARY KEY,
  FarmerID INT,
  Type VARCHAR(50),
  Amount DECIMAL(10,2),
```

```
  FOREIGN KEY (FarmerID) REFERENCES Farmer(FarmerID)
);
```

7. Output (Sample Data + Queries)

```
 Inserting Data
INSERT INTO Farmer VALUES (1, 'Raj Malhotra', 45,
'9876543210');
INSERT INTO Land VALUES (101, 1, 'Punjab', 2.5);
INSERT INTO Crop VALUES (201, 101, 'Wheat', 'Rabi');
INSERT INTO Tool VALUES (301, 1, 'Tractor', 1);
INSERT INTO Subsidy VALUES (401, 1, 'Fertilizer',
6000.00);
```

Query 1: Show farmer and land info

```
SELECT F.Name, L.Location, L.Area
FROM Farmer F
JOIN Land L ON F.FarmerID = L.FarmerID;
```

Query 2: List crops grown by each farmer

```
SELECT F.Name, C.CropName
FROM Farmer F
JOIN Land L ON F.FarmerID = L.FarmerID
JOIN Crop C ON L.LandID = C.LandID;
```

8. Conclusion

The **Farmer Management System** demonstrates the power of DBMS in organizing and managing real-world data efficiently. This system simplifies data handling for farmers, government officials, and agriculture-based organizations by providing a centralized solution for storing and retrieving data with ease. It enhances productivity and transparency in agricultural systems.

9. Learning Outcomes

- Gained knowledge of relational database design and normalization.
- Learned how to implement entity relationships using SQL.
- Practiced using constraints like primary and foreign keys.
- Developed skills in writing effective SQL queries.
- Understood how to apply database design in real-life scenarios like agriculture.