**NoSQL Project Proposal**

Problem Statement:

Farmers operating extensive ventures face several challenges in effectively managing their tasks, produce and workforce.They must be prepared to deal with emergencies and have a plan in place for when disasters strike.

Integrating diverse farm operations into a streamlined process has become difficult, affecting the overall efficiency of farm operations. To address this issue, our solution is a user-friendly centralized system that allows farmers to efficiently store, access, and manage activities and operations concerning their farms.

This system can help farmers to streamline their operations, improve their productivity, and increase their profits.By using farm management database, farmers can become strategic and efficient in their daily farm-related tasks and responsibilities.

Functional Requirements

Supply Chain Tracking:

The system should facilitate real-time tracking of incoming supplies, including seeds, fertilizers, and equipment.

It must allow for the recording and monitoring of the quantity, type, and source of each supply.

Produce and Livestock Outflow Management:

The system should enable farmers to track the outflow of produce and livestock, supporting the recording of quantity, destination, and type of each outgoing item.

Farm Employee Data Management:

The system should provide a comprehensive database for managing farm employee information.Storing features for storing and updating employee details, such as personal information, roles.

Role Assignments:

The system must support the assignment and management of specific roles to farm employees.

It should allow for the modification of roles based on changing requirements.

Customer (Company) Management:

The system should include a module for managing customer information, particularly companies purchasing farm products.

It must store details such as company name, contact information, and transaction history.

Sales and Orders Tracking (Expenses and Earnings):

The system must have functionality to record and track sales transactions, including orders and expenses. It should generate reports summarizing financial transactions, providing insights into both expenses and earnings.

Capital/Equipment Management:

The system must have a module for managing capital and equipment, including tracking their usage, and maintenance schedules.

Specific Data Retrieval:

The system should allow users to retrieve specific data efficiently, using filters and search functionalities.It must support the extraction of detailed information on various aspects, such as employee performance and inventory levels.

User Authentication and Authorization:

The system should implement secure user authentication to ensure authorized access.

It must include role-based access controls to manage different levels of permission for users (farm employees).

Current relations

1. \*Workers:\*

- WorkerID (PK) (Varchar)

- FirstName (Varchar)

- LastName (Varchar)

- Address (Varchar)

- Position (enum)

- Salary (double )

2. \*Works\_On:\*

- WorkerID (FK- WorkerID)

- PlotID (FK - PlotID )

- PenID (FK - PenID)

3. \*Livestock:\*

- LivestockID (PK) (Varchar)

- AnimalType

- Breed (Varchar)

- QuantityAvailable (double)

- Unit\_price

- DateOfBirth (Date time)

- PurchaseDate (datetime)

- Animal notes (doc)

4. \*Crop\_Plot:\*

- PlotID (PK)

- CropID (FK- CropID)

- SoilType (enum)

- Size (Varchar)

- Location (Varchar)

- notes (doc)

5. \*Equipment:\*

- EquipmentID (PK) (Varchar)

- EquipmentName (Varchar)

- QuantityAvailable (double)

- PurchaseDate (Date time)

- WarrantyExpirationDate (Date time)

- MaintenanceDate (Date time)

6. \*Crops:\*

- CropID (PK) (Varchar)

- CropName (varchar)

- CropType (Enum)

- QuantityAvailable (double)

- Unit\_price

- PlantingDate (datetime)

- HarvestDate (Date time)

- QuantityPlanted (int)

- QuantityHarvested (int)

- CropStatus ( Growing, Harvested)

- PlantNotes

7. \*Customer:\*

- CustomerID (PK) (varchar)

- FirstName (Varchar)

- LastName(Varchar)

- ContactNumber (Varchar)

- Email (Varchar)

8. \*Order:\*

- OrderID (Pk)

- CustomerID (FK - customer)

- Date (date time)

- Total\_amount (double)

9. OrderItems

* ItemID (PK) (varchar)
* orderID (FK - OrderID)
* ProduceID (FK - CropID)
* LivestockID (Fk - LivestockID)
* Quantity (decimal)
* Amount (decimal)

10. \*Tools\_In\_Use:\*

- ToolUseID (FK - Workers)

- EquipmentID (FK - EquipmentID)

- TimeAcquired (Date time)

- TimeReturned (Date time)

11. \*Expenses\_Record:\*

- ExpenseID (PK) (Varchar)

- ExpenseCategoryID (enum type)

- Amount (double)

- Notes (doc)

12. \*Transactions:\*

- TransactionID (PK) (Varchar)

- AuthID ( WorkerID)

- OrderID ( FK-OrderID)

- Worker\_payID (FK-WorkerID)

- ExpenseID ( FK - ExpenseID)

- TransactionType varchar

- Amount (double)

- Reciept\_photo (png)

- TransactionDate (Date time)

13. \*Livestock\_Pens:\*

- PenID (PK)

- LivestockID (FK)

- Size (Varchar)

- location (varchar)

Relationships

- Workers (1:1)—-----------(1:M) Works\_On

A worker is allowed to work on several plots at given points in time

A plot or pen assignment is involves at maximum and minimum one worker

- Works\_On(M:0)—------(1:1) Crop\_Plot

An assignment to a plot has maximum and minimum one Crop plot involved in the relationship

A single crop plot can either have no assignment to it or have many assigned to it.

- Works\_On(M:0)—------ (1:1) LiveStock\_Pens

An assignment to a livestock pen has maximum and minimum one Crop livestock pen involved in the relationship

A single Livestock pen can either have no assignments to it or have many assignments to it.

- Livestock(1:0) —------(0 : M) LiveStock\_Pens

A single livestock type can either be house in no livestock pen or be housed in many

A livestock pen can either house no livestock or house 1 livestock type at the most

- Crops(1:0)—-------(0 : M) Crop\_Plots

A single crop type can either be cultivated in no crop\_plot or be cultivated on many plots.

A crop plot can either cultivate no livestock or cultivate at most 1 crop type.

- Equipments (1:1) —----- (0 : M) Tools\_In\_Use

An inventory record can either not be in use or can be in use by multiple records

A tool\_in\_use record has to have at most and at least a corresponding inventory record

- Workers(1:1) —--------(0:M )Tools\_In\_Use

A worker can either not be using a tool or be using either one or many tools at a go

A tool in use has to have a corresponding worker record

- Transactions(1:1) —----------- (0 : M) Order

A transaction record can either not be an order or can be many orders

An order has to map on to a given transaction.

- Worker(1 : 0) —-----(has a payment transaction)------ (0 : M) Transactions

A worker record can either map onto no transactions or many

A transaction can either map onto no workers or at most 1

- Worker(1 : 1) —--(authorized transaction)--------- (0 : M) Transactions

A worker record can either authorize no transactions or many

A transaction can has to be authorized by a worker

- Transactions(1:1) —----------- (0 : 1) Expenses\_Record

A transaction record can either not be an expense record or can be a single record

An expense record has to map on to a given transaction.