**Project Documentation for SDG Problem Solution Using Data**

**The problem we are addressing is food insecurity in Kenya, especially in rural areas. Despite small-scale farmers producing enough maize, there are inefficiencies in the distribution process. This leads to wastage, low profits for farmers, and limited access to affordable food for schools. Our project aims to develop a data-driven solution to streamline the distribution process, ensuring that maize is efficiently collected from farmers and supplied to schools. This solution will help reduce food wastage, increase farmers' profits, and provide schools with affordable maize, ultimately improving food security for school feeding programs.**

**Specific Problem Statement: "Addressing inefficiencies in maize distribution from rural farmers to schools to improve food security and reduce wastage."**

**Part 2: Database Design**

**ERD (Entity Relationship Diagram):** The ERD outlines the main entities and relationships in the maize distribution system. Below is the breakdown of entities:

1. **Farmers**: Details about maize producers.
2. **Schools**: Information about the schools receiving the maize.
3. **Transactions**: Records of maize supply transactions between farmers and schools.
4. **Supply Chains**: Describes the logistics of moving maize from farmers to schools.
5. **Maize Stock**: Tracks the quantity of maize held by each farmer and school.

**Entity Descriptions:**

* **Farmers**: Represents each farmer, including their name, location, and contact information.
* **Schools**: Represents each school, including school name, location, and enrollment size.
* **Transactions**: Captures each sale or transfer of maize, including the date, the quantity transferred, and the farmer and school involved.
* **Supply Chains**: Manages logistics entities such as transportation companies, routes, and delivery schedules.
* **Maize Stock**: Holds the quantity of maize in stock at various points in the distribution chain.

**ERD Structure Example:**

| **Entity** | **Attributes** |
| --- | --- |
| **Farmers** | FarmerID (PK), Name, Location, ContactInfo |
| **Schools** | SchoolID (PK), Name, Location, EnrollmentSize |
| **Transactions** | TransactionID (PK), FarmerID (FK), SchoolID (FK), Quantity, Date |
| **Maize Stock** | StockID (PK), FarmerID (FK), SchoolID (FK), Quantity |
| **Supply Chains** | SupplyChainID (PK), Route, DeliverySchedule |

**ERD Diagram:**  
*Illustrate the relationships between Farmers, Schools, Transactions, Maize Stock, and Supply Chains.*

**Database Sschema:** The SQL scripts below create the relational database tables.

-- Create a table for Farmers

CREATE TABLE Farmers (

FarmerID INT PRIMARY KEY,

Name VARCHAR(255),

Location VARCHAR(255),

ContactInfo VARCHAR(255)

);

-- Create a table for Schools

CREATE TABLE Schools (

SchoolID INT PRIMARY KEY,

Name VARCHAR(255),

Location VARCHAR(255),

EnrollmentSize INT

);

-- Create a table for Transactions

CREATE TABLE Transactions (

TransactionID INT PRIMARY KEY,

FarmerID INT,

SchoolID INT,

Quantity INT,

Date DATE,

FOREIGN KEY (FarmerID) REFERENCES Farmers(FarmerID),

FOREIGN KEY (SchoolID) REFERENCES Schools(SchoolID)

);

-- Create a table for Maize Stock

CREATE TABLE MaizeStock (

StockID INT PRIMARY KEY,

FarmerID INT,

SchoolID INT,

Quantity INT,

FOREIGN KEY (FarmerID) REFERENCES Farmers(FarmerID),

FOREIGN KEY (SchoolID) REFERENCES Schools(SchoolID)

);

-- Create a table for Supply Chains

CREATE TABLE SupplyChains (

SupplyChainID INT PRIMARY KEY,

Route VARCHAR(255),

Delivery Schedule VARCHAR(255)

);

**Sample Data Insertion:**

-- Insert sample Farmers data

INSERT INTO Farmers (FarmerID, Name, Location, ContactInfo)

VALUES (1, 'John Doe', 'Rural Area A', '0722000001'),

(2, 'Jane Smith', 'Rural Area B', '0722000002');

-- Insert sample Schools data

INSERT INTO Schools (SchoolID, Name, Location, EnrollmentSize)

VALUES (1, 'Green Valley School', 'Urban Area X', 500),

(2, 'Blue Ridge School', 'Urban Area Y', 300);

-- Insert sample Transaction data

INSERT INTO Transactions (TransactionID, FarmerID, SchoolID, Quantity, Date)

VALUES (1, 1, 1, 1000, '2023-01-10'),

(2, 2, 2, 800, '2023-01-15');

-- Insert sample Maize Stock data

INSERT INTO MaizeStock (StockID, FarmerID, SchoolID, Quantity)

VALUES (1, 1, NULL, 500),

(2, 2, NULL, 600);

**Part 3: SQL Programming**

**Data Retrieval Queries:** Here are SQL queries to retrieve data that help solve the defined problem:

1. **Total Maize Supplied to Schools:**

SELECT SchoolID, SUM(Quantity) AS TotalQuantity

FROM Transactions

GROUP BY SchoolID;

1. **List of Farmers Who Have Supplied Maize in January:**

SELECT Farmers. Name, Transactions.Quantity, Transactions.Date

FROM Farmers

JOIN Transactions ON Farmers.FarmerID = Transactions.FarmerID

WHERE Transactions.Date BETWEEN '2023-01-01' AND '2023-01-31';

1. **Current Maize Stock per Farmer:**

SELECT Farmers. Name, MaizeStock.Quantity

FROM Farmers

JOIN MaizeStock ON Farmers.FarmerID = MaizeStock.FarmerID;

**Data Analysis Queries:** Using advanced SQL to analyze trends and derive actionable insights:

1. **Average Maize Supply per School:**

SELECT SchoolID, AVG(Quantity) AS AvgSupply

FROM Transactions

GROUP BY SchoolID;

1. **Identify Schools with Insufficient Supply (Threshold: 500 units):**

SELECT SchoolID, SUM(Quantity) AS TotalQuantity

FROM Transactions

GROUP BY SchoolID

HAVING SUM(Quantity) < 500;