**Part 1: SDG Selection and Problem Definition**

**Problem Definition: Inefficient Agricultural Supply Chains and Market Access**

In many regions, especially in the rural areas of developing countries, farmers face significant challenges in accessing fair markets and optimizing their supply chains. This inefficiency results in several issues:

* **Price Fluctuations:** Farmers often receive lower prices for their produce due to intermediaries and market inefficiencies.
* **Post-Harvest Losses:** Poor logistics and storage facilities lead to high levels of post-harvest waste.
* **Limited Market Access:** Smallholder farmers may lack direct access to larger markets or buyers, reducing their potential earnings.
* **Sustainability Issues:** Inefficient farming practices contribute to environmental degradation and resource waste.

**Data-Driven Solution:**

1. **Price Optimization:** Use data analytics to provide real-time pricing information and market trends, helping farmers set fair prices for their products and negotiate better deals.
2. **Logistics Management:** Implement AI and data-driven solutions to optimize logistics and reduce post-harvest losses by improving supply chain efficiency and storage practices.
3. **Market Access:** Create a platform that uses data to connect farmers directly with buyers, bypassing intermediaries and expanding market access.
4. **Sustainable Practices Insights:** Leverage AI to analyze data on farming practices and provide recommendations for more sustainable and efficient farming techniques.

By addressing these issues through data-driven solutions, an AI powered agribusiness project can enhance market efficiency, reduce waste, and promote sustainable agricultural practices, contributing to the goals of zero hunger and responsible consumption and production.

**Part 2: Database Design**

**Entities:**

1. **Farmers**
   * **farmer\_id** (Primary Key)
   * **name**
   * **contact\_info**
   * **location**
   * **farm\_size**
   * **crop\_types** (could be a separate table if detailed information is needed)
2. **Buyers**
   * **buyer\_id** (Primary Key)
   * **name**
   * **contact\_info**
   * **organization** (e.g., retailer, wholesaler)
   * **location**
3. **Products**
   * **product\_id** (Primary Key)
   * **name**
   * **category** (e.g., vegetables, fruits)
   * **unit** (e.g., kg, liters)
4. **Transactions**
   * **transaction\_id** (Primary Key)
   * **farmer\_id** (Foreign Key referencing Farmers)
   * **buyer\_id** (Foreign Key referencing Buyers)
   * **product\_id** (Foreign Key referencing Products)
   * **quantity**
   * **price**
   * **transaction\_date**
5. **Logistics**
   * **logistics\_id** (Primary Key)
   * **transaction\_id** (Foreign Key referencing Transactions)
   * **status** (e.g., pending, shipped, delivered)
   * **delivery\_date**
   * **transport\_mode** (e.g., truck, ship)
6. **SustainablePractices**
   * **practice\_id** (Primary Key)
   * **description**
   * **category** (e.g., water conservation, soil management)
   * **recommended** (Boolean indicating if it's recommended by AI)
7. **FarmerPractices**
   * **farmer\_practice\_id** (Primary Key)
   * **farmer\_id** (Foreign Key referencing Farmers)
   * **practice\_id** (Foreign Key referencing SustainablePractices)
   * **adopted\_date**

Based on the ERD provided, here are the SQL statements to create the database schema:

-- Create the Farmers table

CREATE TABLE Farmers (

farmer\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

contact\_info VARCHAR(255),

location VARCHAR(255),

farm\_size DECIMAL(10, 2) -- Assuming farm\_size is in hectares or similar unit

);

-- Create the Buyers table

CREATE TABLE Buyers (

buyer\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

contact\_info VARCHAR(255),

organization VARCHAR(255),

location VARCHAR(255)

);

-- Create the Products table

CREATE TABLE Products (

product\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

category VARCHAR(255),

unit VARCHAR(50) -- e.g., kg, liters

);

-- Create the Transactions table

CREATE TABLE Transactions (

transaction\_id INT AUTO\_INCREMENT PRIMARY KEY,

farmer\_id INT,

buyer\_id INT,

product\_id INT,

quantity DECIMAL(10, 2),

price DECIMAL(10, 2),

transaction\_date DATE,

FOREIGN KEY (farmer\_id) REFERENCES Farmers(farmer\_id),

FOREIGN KEY (buyer\_id) REFERENCES Buyers(buyer\_id),

FOREIGN KEY (product\_id) REFERENCES Products(product\_id)

);

-- Create the Logistics table

CREATE TABLE Logistics (

logistics\_id INT AUTO\_INCREMENT PRIMARY KEY,

transaction\_id INT,

status VARCHAR(50),

delivery\_date DATE,

transport\_mode VARCHAR(50),

FOREIGN KEY (transaction\_id) REFERENCES Transactions(transaction\_id)

);

-- Create the SustainablePractices table

CREATE TABLE SustainablePractices (

practice\_id INT AUTO\_INCREMENT PRIMARY KEY,

description TEXT,

category VARCHAR(255),

recommended BOOLEAN

);

-- Create the FarmerPractices table (many-to-many relationship between Farmers and SustainablePractices)

CREATE TABLE FarmerPractices (

farmer\_practice\_id INT AUTO\_INCREMENT PRIMARY KEY,

farmer\_id INT,

practice\_id INT,

adopted\_date DATE,

FOREIGN KEY (farmer\_id) REFERENCES Farmers(farmer\_id),

FOREIGN KEY (practice\_id) REFERENCES SustainablePractices(practice\_id)

);

To populate the database with relevant sample data, you can use the following SQL INSERT statements for each table:

-- Insert sample data into Farmers table

INSERT INTO Farmers (name, contact\_info, location, farm\_size)

VALUES

('Hellen Kairu, 'kairu96@gmail.com’, 'Rural Area A', 50.00),

('Muringe Jane, 'janemuringe@gmail.com’, 'Rural Area B', 35.00),

('Alice Johnson', 'alicejohnson54@gmail.com', 'Rural Area C', 40.00);

-- Insert sample data into Buyers table

INSERT INTO Buyers (name, contact\_info, organization, location)

VALUES

('Acme Corp', 'acme.corp@example.com', 'Acme Corporation', 'Urban Area A'),

('GreenGrocers', 'greengrocers@example.com', 'Green Grocers', 'Urban Area B'),

('FoodMart', 'foodmart@example.com', 'FoodMart Retailers', 'Urban Area C');

-- Insert sample data into Products table

INSERT INTO Products (name, category, unit)

VALUES

('Tomatoes', 'Vegetables', 'kg'),

('Apples', 'Fruits', 'kg'),

('Milk', 'Dairy', 'liters');

-- Insert sample data into Transactions table

INSERT INTO Transactions (farmer\_id, buyer\_id, product\_id, quantity, price, transaction\_date)

VALUES

(1, 1, 1, 100.00, 500.00, '2024-08-01'),

(2, 2, 2, 150.00, 450.00, '2024-08-05'),

(3, 3, 3, 200.00, 400.00, '2024-08-10');

-- Insert sample data into Logistics table

INSERT INTO Logistics (transaction\_id, status, delivery\_date, transport\_mode)

VALUES

(1, 'Delivered', '2024-08-02', 'Truck'),

(2, 'Shipped', '2024-08-06', 'Van'),

(3, 'Pending', NULL, 'Truck');

-- Insert sample data into SustainablePractices table

INSERT INTO SustainablePractices (description, category, recommended)

VALUES

('Drip irrigation to conserve water', 'Water Conservation', TRUE),

('Organic composting for soil health', 'Soil Management', TRUE),

('Crop rotation to maintain soil fertility', 'Soil Management', TRUE);

-- Insert sample data into FarmerPractices table

INSERT INTO FarmerPractices (farmer\_id, practice\_id, adopted\_date)

VALUES

(1, 1, '2024-06-01'),

(2, 2, '2024-07-15'),

(3, 3, '2024-08-01');

**Part 3: SQL Programming**

* a. **Data Retrieval:** Write SQL queries to retrieve relevant data based on your problem definition.

**1. Retrieve All Transactions for a Specific Farmer**

SELECT t.transaction\_id, b.name AS buyer\_name, p.name AS product\_name, t.quantity, t.price, t.transaction\_date

FROM Transactions t

JOIN Buyers b ON t.buyer\_id = b.buyer\_id

JOIN Products p ON t.product\_id = p.product\_id

WHERE t.farmer\_id = 1; -- Replace 1 with the desired farmer\_id

**2. Get Total Sales and Average Price for a Specific Product**

SELECT p.name AS product\_name,

SUM(t.quantity) AS total\_quantity\_sold,

SUM(t.price) AS total\_sales,

AVG(t.price) AS average\_price

FROM Transactions t

JOIN Products p ON t.product\_id = p.product\_id

WHERE p.product\_id = 1 -- Replace 1 with the desired product\_id

GROUP BY p.name;

**3. List All Farmers and Their Adopted Sustainable Practices**

SELECT f.name AS farmer\_name, sp.description AS practice\_description, fp.adopted\_date

FROM FarmerPractices fp

JOIN Farmers f ON fp.farmer\_id = f.farmer\_id

JOIN SustainablePractices sp ON fp.practice\_id = sp.practice\_id;

**4. Find Buyers and Their Purchased Products in a Given Time Range**

SELECT b.name AS buyer\_name, p.name AS product\_name, SUM(t.quantity) AS total\_quantity

FROM Transactions t

JOIN Buyers b ON t.buyer\_id = b.buyer\_id

JOIN Products p ON t.product\_id = p.product\_id

WHERE t.transaction\_date BETWEEN '2024-08-01' AND '2024-08-31' -- Replace with desired date range

GROUP BY b.name, p.name;

**5. Retrieve Current Status of Logistics for All Transactions**

SELECT t.transaction\_id, l.status, l.delivery\_date, l.transport\_mode

FROM Logistics l

JOIN Transactions t ON l.transaction\_id = t.transaction\_id;

**6. List Sustainable Practices and Farmers Who Adopted Them**

SELECT sp.description AS practice\_description, f.name AS farmer\_name

FROM SustainablePractices sp

JOIN FarmerPractices fp ON sp.practice\_id = fp.practice\_id

JOIN Farmers f ON fp.farmer\_id = f.farmer\_id;

**7. Get Farmers with the Largest Farm Sizes**

SELECT name, farm\_size

FROM Farmers

ORDER BY farm\_size DESC

LIMIT 5; -- Adjust the limit as needed

1. **Find the Most Recent Transactions for Each Farmer**

SELECT t.farmer\_id, f.name AS farmer\_name, t.transaction\_id, t.transaction\_date, p.name AS product\_name, t.quantity, t.price

FROM Transactions t

JOIN Farmers f ON t.farmer\_id = f.farmer\_id

JOIN Products p ON t.product\_id = p.product\_id

WHERE t.transaction\_date = (

SELECT MAX(transaction\_date)

FROM Transactions t2

WHERE t2.farmer\_id = t.farmer\_id

);

* b. **Data Analysis:** Write SQL queries to analyze data and generate insights related to your SDG problem.

**1. Analyze Total Sales by Farmer**

SELECT f.name AS farmer\_name,

SUM(t.quantity) AS total\_quantity\_sold,

SUM(t.price) AS total\_sales

FROM Transactions t

JOIN Farmers f ON t.farmer\_id = f.farmer\_id

GROUP BY f.name

ORDER BY total\_sales DESC;

**Insight:** This query provides an overview of which farmers generate the most sales, helping to identify top-performing farmers and understand their market impact.

**2. Calculate Average Price Per Product**

SELECT p.name AS product\_name,

AVG(t.price) AS average\_price

FROM Transactions t

JOIN Products p ON t.product\_id = p.product\_id

GROUP BY p.name

ORDER BY average\_price DESC;

**Insight:** This query helps determine the average selling price of each product, which can be used to analyze price trends and inform pricing strategies.

**3. Determine the Most Common Transport Mode Used**

SELECT l.transport\_mode,

COUNT(\*) AS count

FROM Logistics l

GROUP BY l.transport\_mode

ORDER BY count DESC;

**Insight:** This query reveals which transport modes are most frequently used, helping to assess logistics efficiency and identify potential areas for improvement.

**4. Identify Farmers with the Highest Post-Harvest Losses**

-- Assuming you have a field for post-harvest losses in the Logistics table or another related table

SELECT f.name AS farmer\_name,

SUM(t.quantity - COALESCE(l.quantity\_delivered, 0)) AS estimated\_post\_harvest\_losses

FROM Transactions t

JOIN Farmers f ON t.farmer\_id = f.farmer\_id

LEFT JOIN Logistics l ON t.transaction\_id = l.transaction\_id

GROUP BY f.name

ORDER BY estimated\_post\_harvest\_losses DESC;

**Insight:** This query estimates post-harvest losses by comparing quantities sold to quantities delivered, helping identify farmers who may need better logistics solutions.

**5. Evaluate Adoption of Sustainable Practices by Farmers**

SELECT f.name AS farmer\_name,

COUNT(fp.practice\_id) AS number\_of\_practices\_adopted

FROM FarmerPractices fp

JOIN Farmers f ON fp.farmer\_id = f.farmer\_id

GROUP BY f.name

ORDER BY number\_of\_practices\_adopted DESC;

**Insight:** This query shows which farmers are adopting sustainable practices, providing insight into the overall adoption rate and identifying leading adopters.

**6. Analyze Monthly Sales Trends**

SELECT DATE\_FORMAT(t.transaction\_date, '%Y-%m') AS month,

SUM(t.price) AS total\_sales

FROM Transactions t

GROUP BY month

ORDER BY month;

**Insight:** This query helps analyze sales trends over time, identifying peak sales months and trends that may be related to seasonal factors or market conditions.

**7. Compare Sales by Product Category**

SELECT p.category AS product\_category,

SUM(t.quantity) AS total\_quantity\_sold,

SUM(t.price) AS total\_sales

FROM Transactions t

JOIN Products p ON t.product\_id = p.product\_id

GROUP BY p.category

ORDER BY total\_sales DESC;

**Insight:** This query provides an overview of sales performance by product category, helping to identify which categories are performing well and which may need attention.

**8. Assess the Impact of Sustainable Practices on Sales**

SELECT sp.description AS practice\_description,

SUM(t.price) AS total\_sales

FROM FarmerPractices fp

JOIN SustainablePractices sp ON fp.practice\_id = sp.practice\_id

JOIN Transactions t ON fp.farmer\_id = t.farmer\_id

GROUP BY sp.description

ORDER BY total\_sales DESC;

**Insight:** This query assesses whether farmers who adopt sustainable practices experience higher sales, providing insight into the potential benefits of sustainable farming.

5. Document the process of importing data into Excel and ensuring consistency.

1. **Export Data:**

Transaction data is exported from the database as transactions.csv.

1. **Import Data into Excel:**

The CSV file is imported into Excel, with the data appearing in a new worksheet.

1. **Verify and Standardize Data:**

Date formats are checked and standardized to MM/DD/YYYY.

Currency fields are formatted to two decimal places with the $ symbol.

A check for duplicates using COUNTIF reveals no duplicate records.

1. **Apply Data Validation:**

Data validation is set up on the Category column to ensure only predefined categories are used.

1. **Document the Process:**

Any issues encountered (e.g., misaligned columns during import) are documented, along with how they were fixed.

1. **Final Review:**

A final review confirms that the data is consistent, accurate, and ready for use in PivotTables and charts.