SQL Project Report

Created by:

Somnath Banerjee

Projects:

[‘A’-Dairy Management System Research Project]

[‘B’-Loan Database Management System]

Table of Content : List of my projects

Author Name : **Somnath Banerjee**

Github Profile : https://github.com/Somnath342000

Gmail : somnathbanerjee342000@gmail.com

Phone ***:91+ 6290693785***

LinkedIN Profile : *https://www.linkedin.com/in/somnath-banerjee-9a91391a4*

Python Machine Learning	***Power Consumption Projectn Time Series Forecasting) -*** https://github.com/Somnath342000/Python-ML-project-of-Power-Consumption-Forecasting-state-wise-in-India.git
	Healthcare Research Project (Unsupervised ML Project for categorical target variable)- https://github.com/Somnath342000/Python-Unsupervised-ML-Classification-project-on-Healthcare-Research.git
	Stock Market fundamental Analysis (Unsupervised ML Project for continuous target variable)- https://github.com/Somnath342000/Python-Unsupervised-ML-Regression-project-on-Fundamental-Analysis.git
Power BI	***American Coffee Taste Analysis Project -*** https://github.com/Somnath342000/Power-BI-American-Coffee-Taste-Analysis-Project.git
SQL	***Dairy Management Information Project for Market Research Survey -*** https://github.com/Somnath342000/Dairy-Information-System-Management-Project-SQL_Python-.git
	Loan Database Analysis System project- https://github.com/Somnath342000/SQL-Python-Financial-Health-Tracker.git
Tableau	***Pizza Sales Tableau Analysis project - Tableau Community*** https://public.tableau.com/views/PIZZASALES_17376436150760/Dashboard1?:language=en-GB&publish=yes&:sid=&:redirect=auth&:display_count=n&:origin=viz_share_link ***Github Link -*** https://github.com/Somnath342000/Tableau-pizza-Sales-Analysis.git
Excel & VBA	***Super store Sales MIS Dashboard Report-*** https://github.com/Somnath342000/Excel-SuperStore-Sales-Report-Dashboard.git
	Automatic Grade Checker – https://github.com/Somnath342000/Excel-VBA-Automatic-Grade-Checker.git

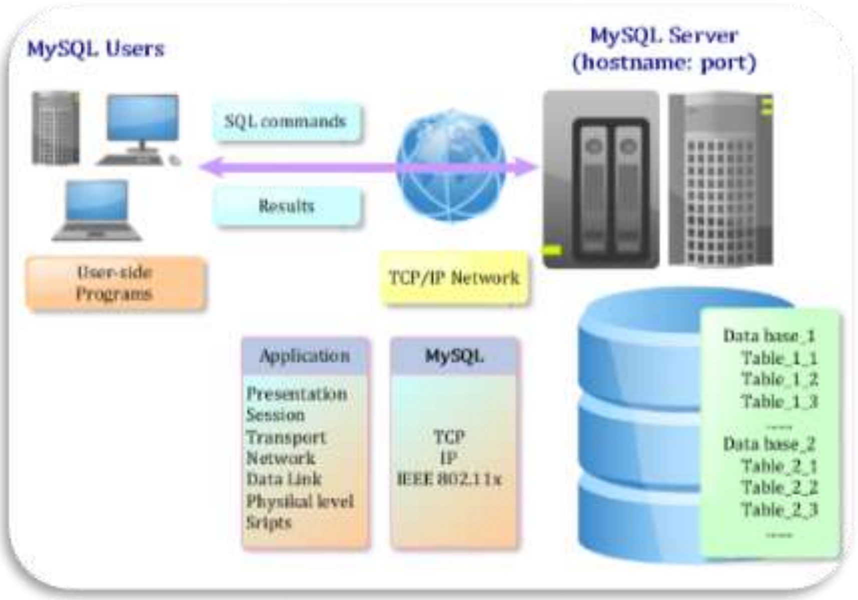
Introduction to SQL

- **What is SQL (Structured Query Language)?**
  - SQL is a standard programming language used to manage and manipulate relational databases. It allows users to query, insert, update, and delete data from databases.
  - SQL is used to interact with relational database management systems (RDBMS).
- **Importance of SQL in Database Management**
  - SQL is the foundation for managing databases in industries such as finance, healthcare, education, and more.
- **SQL Commands Overview**
  - **DDL (Data Definition Language):** Commands like `CREATE`, `ALTER`, `DROP` for defining and modifying database structures.
  - **DML (Data Manipulation Language):** Commands like `SELECT`, `INSERT`, `UPDATE`, `DELETE` for manipulating data.
  - **DCL (Data Control Language):** Commands like `GRANT`, `REVOKE` for permissions.
  - **TCL (Transaction Control Language):** Commands like `COMMIT`, `ROLLBACK` for handling database transactions.

Introduction to SQL and Its Usage in Industry

SQL (Structured Query Language) is a standard programming language used to manage and manipulate databases. SQL allows users to perform a wide range of operations on relational databases, including querying data, inserting, updating, and deleting records, and creating and modifying database structures.

SQL is widely used in various industries and sectors to handle data effectively, streamline business operations, and make informed decisions based on data. Its simplicity, efficiency, and flexibility have made it a go-to choice for database management across multiple fields.



Core Features of SQL

1. **Data Retrieval**: SQL allows users to retrieve specific information from large datasets using powerful querying capabilities. The SELECT statement is used to filter, sort, and retrieve data from one or more tables in a database.
2. **Data Manipulation**: SQL enables users to perform CRUD (Create, Read, Update, Delete) operations:
   - **INSERT**: Add new records into a table.
   - **UPDATE**: Modify existing records.
   - **DELETE**: Remove records from a table.
3. **Data Definition**: SQL is used to define the structure of a database using Data Definition Language (DDL). This includes:
   - **CREATE**: Create new database objects (e.g., tables, views).
   - **ALTER**: Modify existing objects (e.g., adding columns).
   - **DROP**: Delete objects (e.g., dropping a table).
4. **Data Control**: SQL provides commands to control access to data and manage user permissions using:
   - **GRANT**: Grant specific privileges to users.
   - **REVOKE**: Remove user privileges.

5. **Transaction Control**: SQL supports managing transactions, which are groups of operations that are executed as a single unit. Key commands include:
   - **COMMIT**: Save changes made by a transaction.
   - **ROLLBACK**: Undo changes made by a transaction if something goes wrong.
   - **SAVEPOINT**: Set a savepoint in a transaction to allow partial rollbacks.

SQL in Different Industries

SQL is used across industries to manage vast amounts of structured data efficiently. Below are the primary use cases of SQL across various sectors:

1. Banking and Finance

- **Data Storage and Retrieval**: Banks and financial institutions store vast amounts of data, including customer accounts, transaction records, and loan histories. SQL is used to manage this data and generate reports such as account balances, transaction history, and financial statements.
- **Fraud Detection**: SQL queries help identify unusual patterns in transaction data that may indicate fraudulent activity, allowing quick responses to prevent losses.
- **Risk Management**: SQL is used to analyze historical data to predict and manage financial risks by analyzing patterns and correlations.

2. Retail and E-Commerce

- **Inventory Management**: SQL is essential for managing inventory data, tracking stock levels, and ensuring product availability. Retailers use SQL to monitor the movement of goods and generate reports for inventory replenishment.
- **Sales Analysis**: SQL queries help analyze sales data, understand customer preferences, and track product performance across different regions. This is vital for optimizing pricing, promotions, and product offerings.
- **Customer Relationship Management (CRM)**: Retailers use SQL to manage customer data, personalize marketing campaigns, and improve customer retention by analyzing purchasing behavior.

3. Healthcare

- **Patient Records Management**: SQL is widely used in healthcare systems to store and retrieve patient data, such as medical histories, prescriptions, appointments, and test results. Databases are structured in a way that allows fast retrieval of critical medical information.
- **Clinical Research**: SQL allows healthcare researchers to query and analyze medical data for patterns that help in developing new treatments, improving care delivery, and understanding disease progression.
- **Billing and Insurance**: SQL plays a key role in managing billing information, insurance claims, and financial records, ensuring timely payments and compliance with regulations.

4. Education

- **Student Records Management**: Schools and universities use SQL databases to store student data, such as enrollment information, grades, course registrations, and attendance. SQL is essential for generating academic reports, transcripts, and tracking student progress.
- **Curriculum Management**: Educational institutions use SQL to manage course offerings, track student enrollments, and generate schedules, ensuring optimal resource allocation.
- **Research and Analysis**: SQL helps in querying large datasets for research purposes, particularly in areas like student performance analysis, educational trends, and institutional assessments.

5. Telecommunications

- **Customer Data Management**: Telecommunications companies use SQL to store and manage customer data, such as account details, call records, and service subscriptions. SQL helps analyze customer behavior and optimize customer service operations.
- **Billing and Usage Tracking**: SQL is used to track customer usage of services like data, voice, and SMS, and manage billing processes based on usage.

- **Network Management**: SQL databases store information about network infrastructure, including data traffic, bandwidth usage, and service uptime, enabling telecom companies to ensure service reliability.

6. Manufacturing

- **Production Monitoring**: SQL is used to track production processes, monitor inventory levels of raw materials, and ensure timely delivery of finished products. This helps in optimizing supply chain and production efficiency.
- **Quality Control**: Manufacturers use SQL to manage data related to product quality, defect tracking, and testing results, enabling them to maintain product standards.
- **Employee and Resource Management**: SQL is used to track employee work schedules, labor costs, and equipment utilization, improving operational efficiency.

7. Logistics and Transportation

- **Inventory and Shipment Tracking**: Logistics companies use SQL to manage inventory records, track shipments, and optimize routes. SQL helps in improving delivery times and reducing shipping costs.
- **Fleet Management**: SQL is used to monitor vehicle locations, maintenance schedules, and driver information to ensure efficient fleet management.
- **Supply Chain Optimization**: SQL helps analyze supply chain data, allowing businesses to optimize their distribution networks, reduce bottlenecks, and improve overall efficiency.

8. Entertainment and Media

- **Content Management**: Media companies use SQL to store and manage digital content, such as movies, music, and articles. SQL helps retrieve and deliver content to users efficiently.
- **Subscription and User Management**: SQL is used to track user subscriptions, preferences, and activity. Media services like Netflix and Spotify use SQL to personalize content recommendations.
- **Audience Analysis**: SQL helps analyze user data and track engagement metrics to optimize content creation and marketing strategies.

9. Government

- **Public Administration**: Governments use SQL databases to store records related to public services, citizen registrations, permits, and licenses. This allows governments to manage records efficiently and ensure transparency.
- **Census Data Analysis**: SQL is used to analyze population data and track demographic trends, which helps inform policy decisions and public resource allocation.
- **E-Governance**: SQL is central to managing e-governance services such as tax filings, welfare programs, and legal documentation.

Benefits of Using SQL in Industry

1. **Data Integrity and Accuracy**: SQL ensures data integrity by using constraints (e.g., primary keys, foreign keys) to enforce data relationships and consistency across different tables.
2. **Data Security**: SQL allows fine-grained access control, ensuring that only authorized users can access or modify certain data, protecting sensitive information.
3. **Scalability**: SQL databases can handle massive amounts of data and scale easily as businesses grow, ensuring high performance even under large datasets.
4. **Real-Time Reporting and Analytics**: SQL provides the ability to generate reports and analyze data in real time, supporting quick decision-making in various industries.
5. **Cost-Effective**: SQL databases are often open-source and widely supported, making them a cost-effective solution for managing and analyzing data.
6. **Support for Complex Queries**: SQL allows for complex data manipulation, such as joins, unions, and aggregations, which are essential for deep data analysis

Educational Report on SQL using SQLite3 and MySQL

Overview of SQLite3

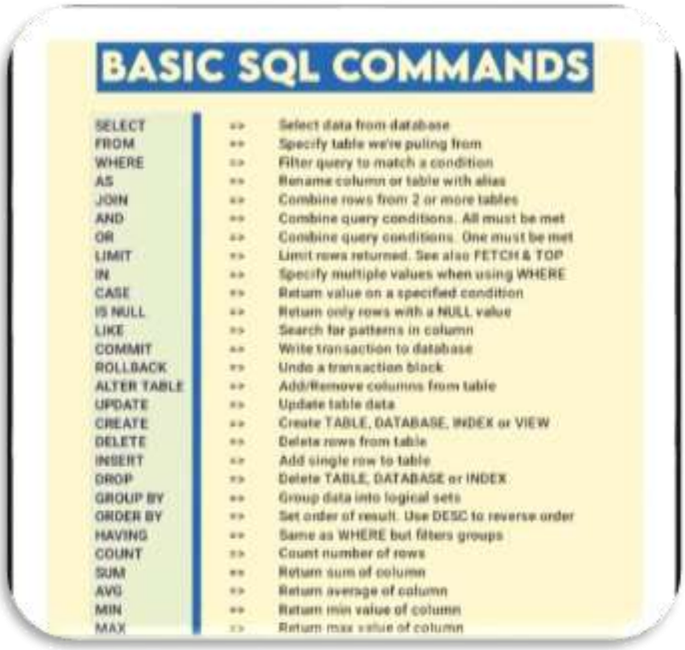
- **What is SQLite3?**
  - SQLite3 is a self-contained, serverless, and zero-configuration SQL database engine. It stores data in a single file on the disk and is ideal for embedded applications.
- **Key Features of SQLite3:**
  - Lightweight and fast.
  - Local, file-based database engine.
  - Requires no server setup, configuration, or network.
- **When to Use SQLite3:**
  - Mobile apps, small web applications, testing environments.
  - Scenarios where simplicity and speed are more important than scalability.
- **Basic SQLite3 Commands:**
  - **Create Database:**

    ```
    CREATE DATABASE my_database;
    ```
  - **Create Table:**

    ```
    CREATE TABLE users (
        id INTEGER PRIMARY KEY,
        name TEXT,
        email TEXT
    );
    ```
  - **Insert Data:**

    ```
    INSERT INTO users (name, email)
    VALUES ('John Doe',
    'john@example.com');
    ```
  - **Select Data:**

    ```
    SELECT * FROM users;
    ```



BASIC SQL COMMANDS

SELECT	=>	Select data from database
FROM	=>	Specify table we're puling from
WHERE	=>	Filter query to match a condition
AS	=>	Rename column or table with alias
JOIN	=>	Combine rows from 2 or more tables
AND	=>	Combine query conditions. All must be met
OR	=>	Combine query conditions. One must be met
LIMIT	=>	Limit rows returned. See also FETCH & TOP
IN	=>	Specify multiple values when using WHERE
CASE	=>	Return value on a specified condition
IS NULL	=>	Return only rows with a NULL value
LIKE	=>	Search for patterns in column
COMMIT	=>	Write transaction to database
ROLLBACK	=>	Undo a transaction block
ALTER TABLE	=>	Add/Remove columns from table
UPDATE	=>	Update table data
CREATE	=>	Create TABLE, DATABASE, INDEX or VIEW
DELETE	=>	Delete rows from table
INSERT	=>	Add single row to table
DROP	=>	Delete TABLE, DATABASE or INDEX
GROUP BY	=>	Group data into logical sets
ORDER BY	=>	Set order of result. Use DESC to reverse order
HAVING	=>	Same as WHERE but filters groups
COUNT	=>	Count number of rows
SUM	=>	Return sum of column
AVG	=>	Return average of column
MIN	=>	Return min value of column
MAX	=>	Return max value of column

Overview of MySQL

- **What is MySQL?**
  - MySQL is a widely-used open-source relational database management system (RDBMS) that stores and manages large volumes of data in a multi-user environment. It typically operates on client-server architecture.
- **Key Features of MySQL:**
  - Supports large-scale data management with a multi-user environment.
  - Relational database that supports complex queries and transactions.
  - Scalable and suitable for both small and large applications.
- **When to Use MySQL:**
  - Websites, enterprise applications, data warehousing, and large systems that require scalability and reliability.
- **Basic MySQL Commands:**
  - **Create Database:**

    ```
    CREATE DATABASE my_database;
    ```
  - **Create Table:**

    ```
    CREATE TABLE users (
        id INT PRIMARY KEY AUTO_INCREMENT,
        name VARCHAR(100),
        email VARCHAR(100)
    );
    ```
  - **Insert Data:**

    ```
    INSERT INTO users (name, email) VALUES ('Jane Doe', 'jane@example.com');
    ```
  - **Select Data:**

    ```
    SELECT * FROM users;
    ```

Comparison between SQLite3 and MySQL

- **Architecture:**
  - SQLite3 is serverless and stores data in a single file, making it ideal for lightweight applications and embedded systems.
  - MySQL is a client-server RDBMS, suitable for large applications, and typically uses a network-based architecture.
- **Scalability:**
  - SQLite3 is not built for handling large-scale applications with heavy concurrent access.
  - MySQL supports high-concurrency environments and large datasets, making it suitable for enterprise-level applications.
- **Use Cases:**
  - SQLite3: Mobile apps, single-user applications, testing environments, small web applications.
  - MySQL: Large web apps, multi-user systems, online platforms with high traffic, and enterprise software.
- **Performance:**
  - SQLite3 is fast and efficient for small-scale applications but does not support complex queries or transactions as effectively as MySQL.
  - MySQL, while requiring more resources, is optimized for large-scale applications and handles more complex workloads.
- **Installation and Setup:**
  - SQLite3 does not require installation or server setup; just create a database file and start working.
  - MySQL requires setting up a server (local or remote) and user authentication before use.

SQL Syntax Examples and Usage

- **Basic Queries in Both SQLite3 and MySQL:**
  - **Creating a Database and Table:**

    ```
    CREATE DATABASE example_db;
    CREATE TABLE employees (
        id INT AUTO_INCREMENT PRIMARY KEY,
        name VARCHAR(50),
        role VARCHAR(50),
        salary DECIMAL(10, 2)
    );
    ```
  - **Inserting Data:**

    ```
    INSERT INTO employees (name, role, salary)
    VALUES ('Somnath', 'Analyst', 75000.00);
    ```
  - **Selecting Data:**

    ```
    SELECT * FROM employees WHERE role = 'Analyst';
    ```
  - **Updating Data:**

    ```
    UPDATE employees SET salary = 80000 WHERE name = 'Somnath';
    ```
  - **Deleting Data:**

    ```
    DELETE FROM employees WHERE name = 'Somnath';
    ```

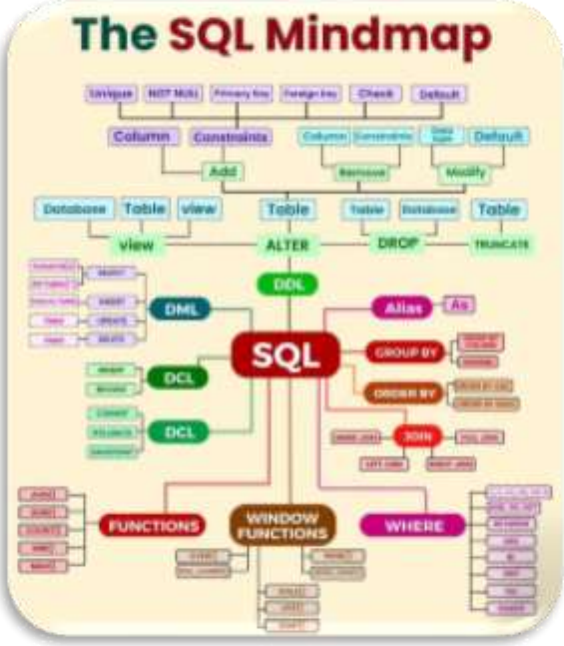
Advantages and Disadvantages of SQLite3 vs. MySQL

- **SQLite3 Advantages:**
  - No setup required, lightweight, simple to use.
  - Best suited for mobile apps, desktop apps, or small web applications.
  - Excellent for testing and prototyping.
- **SQLite3 Disadvantages:**
  - Not suitable for multi-user, high-concurrency applications.
  - Limited scalability.
  - Lacks advanced features found in enterprise-level databases like MySQL.
- **MySQL Advantages:**
  - High scalability and support for large datasets.
  - Strong support for multi-user environments and concurrent access.
  - Reliable, with strong community and commercial support.
- **MySQL Disadvantages:**

- Requires server setup and more resources.
- Complexity increases with large applications.

Practical Applications of SQL with SQLite3 and MySQL

- **SQLite3 Use Case Example:**
  - **Mobile App Database:** A simple mobile app can use SQLite3 to store local user data (e.g., an app for managing personal notes).
- **MySQL Use Case Example:**
  - **E-commerce Website:** MySQL can be used to store and manage product, customer, and order data for an online store, handling high traffic and concurrent users.



---

10. References

- List books, websites, and documentation you referenced while preparing the report. For example:
  - MySQL Documentation: https://dev.mysql.com/doc/
  - SQLite Documentation: https://www.sqlite.org/docs.html
  - SQL Tutorials: https://www.w3schools.com/sql/

Project Report: Dairy Management Information System (DMIS) for the Indian Dairy Industry -(A)

GitHub Link - https://github.com/Somnath342000/Dairy-Information-System-Management-Project-SQL_Python-.git

---

1. Introduction

The **Dairy Management Information System (DMIS)** is designed to centralize and automate the management of key aspects of the dairy industry, including milk production, sales tracking, and inventory management. This SQL-based project uses **SQLite3** and **MySQL** to create a robust database system for dairy farms and their sales operations. The project specifically focuses on **India's dairy industry**, spanning the years **2018-2022**, and analyzing data from two primary tables: **Dairy Farm** and **Dairy Sales**.

This project aims to gain insights into **consumer preferences**, **production efficiencies**, and **market dynamics** in the dairy industry, with a focus on understanding the regional variations in demand and supply of dairy products.

---

2. Problem Statement

The dairy industry in India is vast and complex, with thousands of dairy farms, multiple brands, and diverse consumer preferences across different states. However, managing data related to milk production, sales, inventory, and customer preferences has traditionally been done manually, leading to inefficiencies and errors. The key problems addressed in this project include:

1. **Fragmented Data**: Data about production and sales is often stored separately, making it hard to get a holistic view of the dairy operations.
2. **Lack of Real-Time Data**: Without a centralized system, stakeholders are unable to access real-time data for decision-making.
3. **Regional Differences**: Different regions in India exhibit varying demand and supply dynamics for dairy products. Understanding these differences can help optimize production and sales.
4. **Inefficiencies in Sales and Inventory**: Poor tracking of inventory and production can result in overproduction or stockouts, leading to financial losses.

The project aims to design a **centralized database system** to improve efficiency, provide real-time data access, and help stakeholders optimize dairy operations.

---

3. Database Design and Structure

Tables Overview

The database is composed of two primary tables:

1. **Dairy Farm Table**: This table stores production-related data, such as milk quantity, farm size, and cow details.
2. **Dairy Sales Table**: This table records sales transactions, including product details, sales volumes, revenue, and customer locations.

---

4. SQL Query Development & Data Analysis

Total Milk Production by State

To analyze the total milk production across different states, we can use the following SQL query:

```
SELECT Production_State, SUM(Quantity_Produced) AS Total_Production
FROM Dairy_Farm
GROUP BY Production_State;
```

This query will provide insights into the total milk produced in each state.

Sales Analysis by Product

To analyze the sales of different dairy products by brand, we use the following query:

```
SELECT Product_Name, Brand, SUM(Quantity_Sold) AS Total_Sold, SUM(Total_Revenue) AS
Total_Revenue
FROM Dairy_Sales
GROUP BY Product_Name, Brand;
```

This will give insights into which dairy products are performing well and the revenue generated from different brands.

Regional Sales Trends

To compare sales by region, the following query is used:

```
SELECT Customer_State, SUM(Total_Revenue) AS Total_Sales
FROM Dairy_Sales
GROUP BY Customer_State
ORDER BY Total_Sales DESC;
```

This provides a comparison of sales revenue across different states in India.

---

6. Results and Findings

From the analysis of the **Dairy Farm** and **Dairy Sales** tables, the following findings were observed:

1. **Production Trends**:
   - Delhi and Chandigarh showed the highest production of milk, as these regions have more farms with larger land areas and cow populations.
2. **Sales Performance**:
   - **Amul** and **Mother Dairy** were the dominant brands, with **Amul** leading in milk sales and **Mother Dairy** excelling in curd sales.
3. **Consumer Preferences**:
   - **Curd** emerged as the most preferred product, especially in states like Delhi and Chandigarh, reflecting regional dietary habits.
4. **Regional Demand**:
   - States like Delhi, Chandigarh, and Maharashtra had higher sales figures compared to others, indicating strong demand for dairy products in urbanized and economically developed regions.

---

7. Conclusion

This SQL-based project demonstrates the power of data in the dairy industry, specifically in managing production and sales data efficiently. By creating a centralized database using **SQLite3** and **MySQL**, this project has streamlined the tracking of **milk production**, **inventory**, and **sales transactions**. The results highlight key insights into **consumer preferences**, **regional demand**, and **market trends** in India's dairy industry. This information can guide strategic decisions for dairy farm operations, brand marketing, and regional expansion.

By implementing this **Dairy Management Information System (DMIS)**, stakeholders can improve efficiency, reduce operational costs, and make data-driven decisions to optimize production, inventory, and sales strategies across the Indian dairy market.

Thank You

GitHub Link-https://github.com/Somnath342000/Dairy-Information-System-Management-Project-SQL_Python-.git

GitHub Link-https://github.com/Somnath342000/SQL-Python-Financial-Health-Tracker.git