

FarmConnect – A Farmer-to-Market Supply Chain

Submitted By: Rutuja Nagnath Kamble

ID: 28285

To: SR Umesh Sir

Date: July 2025

Abstract

FarmConnect is a database-integrated Java console application developed to digitize the agricultural product supply chain. The system enables farmers to upload and manage their products, and allows buyers to search, place, and track orders. It features real-time inventory updates, robust exception handling, and a structured MySQL database backed by JDBC. This project simulates real-world backend operations, ensuring data consistency, usability, and extensibility.

Introduction

The Indian agriculture ecosystem faces a disconnect between farmers and markets. FarmConnect aims to bridge this gap using technology. By offering a console-based Java interface backed by a relational database, farmers can directly list products, and buyers can place orders and view histories. The system is designed with scalability and real-world constraints in mind.

Objectives

- To digitize the interaction between farmers and buyers.
- To manage product inventories and orders in real-time.
- To implement proper database design using MySQL.
- To simulate enterprise-level backend development using JDBC and Java.
- To use exception handling, transactions, and modular code practices.

System Requirements

Hardware:

- RAM: 4 GB or higher
- Processor: Intel i3 or above
- Disk: 500 MB

Software:

- Java JDK 8+
- MySQL Server & Workbench
- IDE (Eclipse / IntelliJ / NetBeans)
- MySQL Connector JAR
- Command-line or Terminal

Technology Stack

Language: Java

Database: MySQL

DB Access: JDBC

Design: OOP, Exception Handling, Collections

Interface: Console (Scanner)

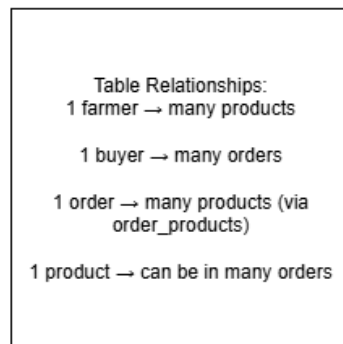
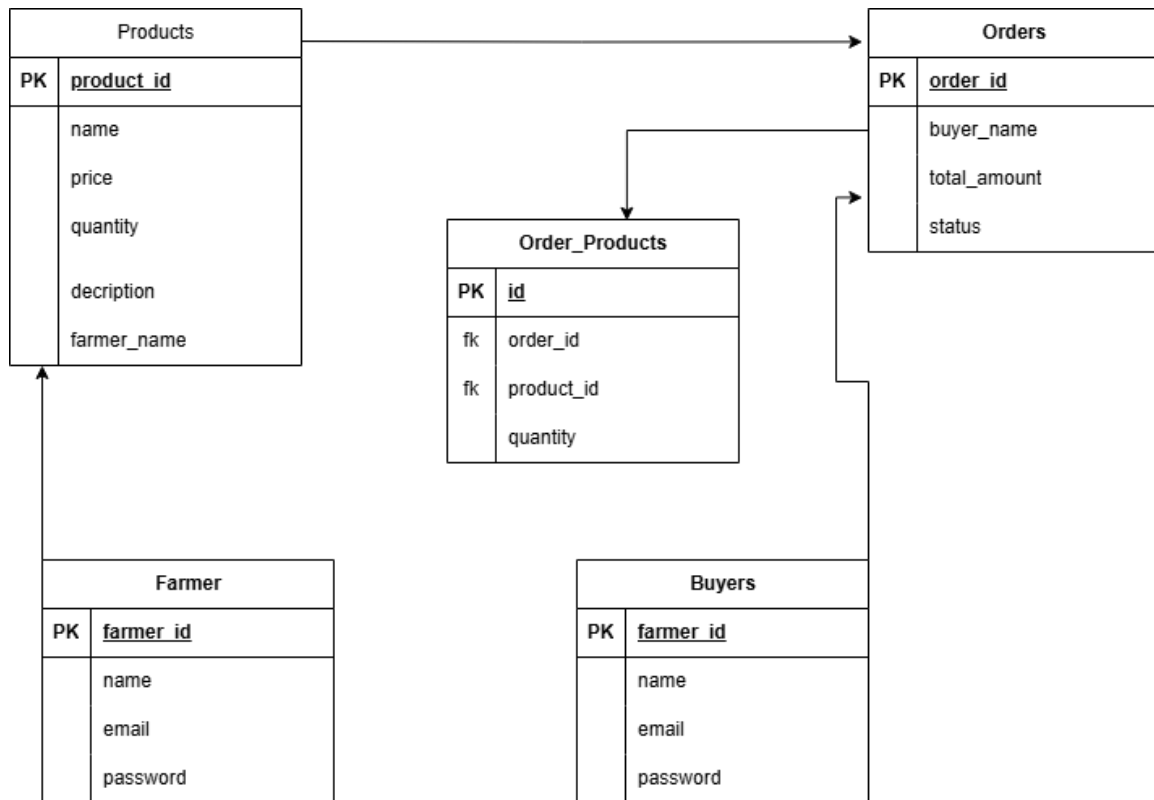
Tools Used: MySQL Workbench, Java IDE

System Design

Architecture: Java-based backend, JDBC middleware, MySQL as data store

Use Case Flow: Farmer – Add/Update/Delete/View Products, View Orders | Buyer – View/Search Products, Add to Cart, Place Order, View History

ER Diagram



Database Tables

- farmers: id, name, email, password (optional)
- buyers: id, name, email, password (optional)
- products: product_id, name, price, quantity, farmer_name
- orders: order_id, buyer_name, total_amount, status
- order_products: order_id, product_id, quantity

Implementation

Features:

- Farmer: Add/Update/Delete/View Products, View Orders
- Buyer: View/Search Products, Place Orders, View History
- Order Management: Auto-update inventory, status updates
- Exception Handling: Invalid input, stock checks, rollback
- DB Operations: Insert, Join, Transactions, Stored Procedure

Stored Procedure

```
CREATE PROCEDURE GetBuyerOrders (IN buyerName VARCHAR(100))
BEGIN
    SELECT o.order_id, o.total_amount, o.status, p.name AS product_name, op.quantity
    FROM orders o
    JOIN order_products op ON o.order_id = op.order_id
    JOIN products p ON op.product_id = p.product_id
    WHERE o.buyer_name = buyerName;
END;
```

Trigger (Optional)

```
CREATE TRIGGER update_stock_after_order
AFTER INSERT ON order_products
FOR EACH ROW
BEGIN
    UPDATE products
    SET quantity = quantity - NEW.quantity
    WHERE product_id = NEW.product_id;
END;
```

Testing & Validation

- Manual test cases for all modules
- Validations:
 - Out-of-stock conditions
 - Invalid product ID
 - Empty cart
- Verifications:
 - Order tracking
 - Stock reduction
 - DB consistency with rollback

Conclusion

The FarmConnect project demonstrates how Java and MySQL can be combined to simulate a real-world backend application. From product listing to order management and stock tracking, the system handles all critical operations. It is modular, scalable, and production-ready with minimal modifications.

Future Scope

- Add login/authentication using farmers and buyers tables
- Create a GUI using Java Swing or a web front-end
- Integrate payment simulation
- Generate PDF invoices or reports

References

- <https://dev.mysql.com/doc/>
- <https://docs.oracle.com/javase/tutorial/>
- Stack Overflow
- GeeksForGeeks
- JavaTPoint