CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

NAAC GRADE A+
Accredited University

UNIVERSITY INSTITUTE OF COMPUTING

CASE STUDY REPORT
ON
SUPERMART MANAGEMENT SYSTEM

Program Name: BCA

Subject Name/Code: Database Management System (23CAT-251)

Submitted By:

Name: Vansh Raizada

UID: 23BCA10476

Section: 4-'A'

Submitted To:

Name: Mr.Arvinder Singh

Designation: Assistant Professor

INTRODUCTION

The **SuperMart Database Management System** is designed to efficiently handle and streamline the operations of a retail supermarket. It facilitates the systematic storage, retrieval, and management of essential business data such as customer information, product details, supplier records, employee profiles, and order transactions.

This database supports key functionalities like placing and tracking orders, managing inventory, monitoring sales performance, and generating reports for business insights. By implementing a relational database structure with clearly defined entities and relationships, SuperMart ensures data integrity, reduces redundancy, and enhances operational efficiency.

The system is built using **MySQL**, incorporating various **SQL queries**, **ER modeling**, **relational constraints**, and **normalisation techniques** to maintain a robust and scalable architecture. This database serves as the backbone for automation in the supermarket, helping to simplify daily operations and support decision-making processes.

TECHNIQUES

The primary technology used in this project is MySQL, an open-source relational database management system. The following techniques have been implemented:

- **Entity-Relationship Modeling** for data structure visualisation.
- **Normalisation** to organise data efficiently and remove redundancy.
- **SQL Queries** for data manipulation and retrieval.
- **Use of Constraints** like PRIMARY KEY, FOREIGN KEY to enforce relationships.
- **Join operations** to combine data from multiple tables.
- **Aggregate Functions** to summarize and analyze data.
- **Filtering and Sorting** to extract meaningful insights from the dataset.
- **Stored Procedures and Views** (optional enhancements) for automation.

The goal is to simulate a real-time cinema database with multiple users accessing the system concurrently. Though our current system is simplified, it lays the foundation for large-scale enterprise software.

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

NAAC GRADE A+
Accredited University

SYSTEM CONFIGURATION

Hardware Requirements

- **Processor:** Intel i5 / Ryzen 5 or higher
- **RAM:** 8 GB minimum
- **Storage:** 256 GB SSD / 500 GB HDD
- **Display:** 14" or larger

Software Requirements

- **OS:** Windows 10/11 or Ubuntu 20.04+
- **DBMS:** MySQL Server 8.0+
- **Interface Tool:** MySQL Workbench / phpMyAdmin
- **ER Tool:** Draw.io / dbdiagram.io
- **Editor:** VS Code / Notepad++

Database Details

- **Name:** vansh_db
- **Tables:** Customers, Orders, OrderDetails, Products, Suppliers, Employees
- **Relations:** Primary & Foreign Keys, Constraints for integrity

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

NAAC GRADE A+
Accredited University

INPUT

The SuperMart database receives input from various entities involved in daily supermarket operations. These inputs are collected through forms, employee entries, or automated systems and are stored in structured tables within the database.

Customers Full Name, Email, Phone, Address

Orders Customer ID, Order Date, Total Amount, Payment Method, Status

Order Details Order ID, Product ID, Quantity, Unit Price

Products Product Name, Category, Brand, Price, Stock Quantity, Supplier ID

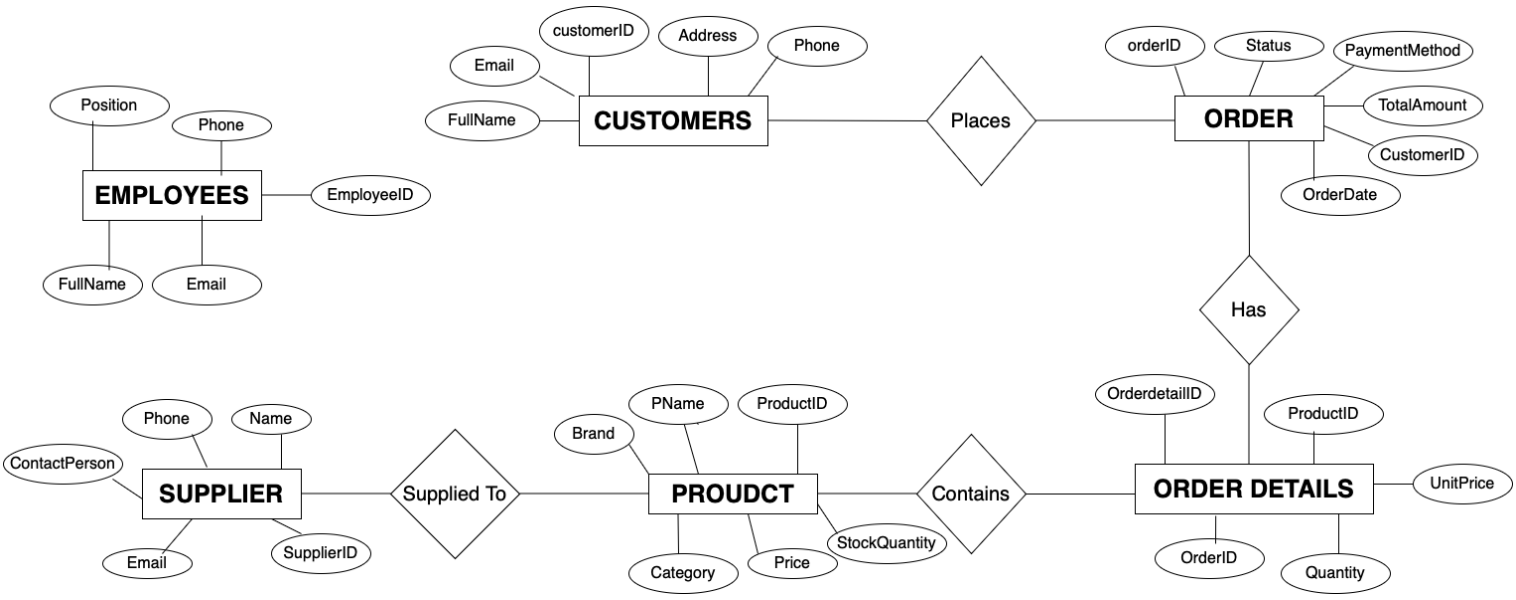
Suppliers Supplier Name, Contact Person, Phone, Email

Employees Full Name, Position, Phone, Email

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

NAAC GRADE A+
Accredited University

ENTITY-RELATIONSHIP DIAGRAM



The Entity-Relationship (ER) diagram outlines the structure and relationships among different entities of the hospital. It forms the blueprint for the actual database schema.

Each entity has clearly defined attributes and is connected using appropriate relationships like one-to-many and many-to-one, ensuring normalization and avoiding data redundancy.

RELATIONSHIP BETWEEN TABLES

These relationships ensure that the relational database mirrors real-world interactions within a SuperMart.

No.	Relationship Type	Parent Table	Child Table	Foreign key	Description
1	One-to-many	Customers	Orders	CustomerID	customer can place multiple orders
2	One-to-many	Orders	OrderDetails	OrderID	One order can have multiple products
3	One-to-many	Products	OrderDetails	ProductID	One product can appear in multiple orders
4	One-to-many	Suppliers	Products	SupplierID	One supplier supply multiple products
5	One-to-one	Customers	Orders	EmployeeID	One employee can manage multiple orders

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

NAAC GRADE A+
Accredited University

TABULAR FORMAT (SCHEMA)

Table Name	Primary Key	Foreign Key	Description
Customers	CustomerID	—	Stores customer info
Orders	OrderID	CustomerID	Records orders
OrderDetails	DetailID	OrderID,ProductID	Stores item details
Products	ProductID	SupplierID	Stores product info
Suppliers	SupplierID	—	Store supplier info
Employees	EmployeeID	—	Store employee info

TABLE CREATION

1. Customers Table:

```
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    FullName VARCHAR(100),
    Email VARCHAR(100),
    Phone VARCHAR(15),
    Address TEXT
);
```

```
INSERT INTO Customers VALUES
(201, 'Riya Sharma', 'riya@gmail.com', '9876543210', 'Chandigarh, Sector 35'),
(202, 'Aryan Kapoor', 'aryan@yahoo.com', '9988776655', 'Delhi, Rohini'),
(203, 'Meena Kumari', 'meena@outlook.com', '9123456780', 'Jaipur, Vaishali Nagar'),
(204, 'Rohit Verma', 'rohitv@gmail.com', '9001122334', 'Mumbai, Andheri'),
(205, 'Pooja Joshi', 'pooja_j@gmail.com', '9856231470', 'Pune, Kothrud'),
(206, 'Tanmay Singh', 'tanmay@gmail.com', '9834567821', 'Lucknow, Gomti Nagar'),
(207, 'Sneha Batra', 'sneha_batra@gmail.com', '9723456712', 'Bangalore, Indiranagar'),
(208, 'Aman Sethi', 'aman.sethi@yahoo.com', '9112233445', 'Hyderabad, Gachibowli');
```

2. Supplier Table:

```
CREATE TABLE Suppliers (
    SupplierID INT PRIMARY KEY,
    SupplierName VARCHAR(100),
    ContactPerson VARCHAR(100),
    Phone VARCHAR(15),
    Email VARCHAR(100)
);
```

```
INSERT INTO Suppliers VALUES
(101, 'Amul Distributors', 'Rajesh Patel', '9876001100', 'amul@distrib.com'),
(102, 'Grocers India', 'Suman Verma', '9823456712', 'grocers@india.com'),
(103, 'Hygiene Plus', 'Anil Mehra', '9900887766', 'hygieneplus@gmail.com'),
(104, 'Sweet Treats', 'Neha Jain', '9812345670', 'sweettreats@yahoo.com'),
(105, 'Bharat Retail', 'Sunil Singh', '9815566778', 'bharatretail@gmail.com'),
(106, 'D-Mart Wholesale', 'Reema Roy', '9798123456', 'dmart@wholesale.com'),
(107, 'Organic Fresh', 'Ajay Khurana', '9888011223', 'organic@fresh.com'),
(108, 'Daily Needs Pvt.', 'Kamal Bhatt', '9812349823', 'needs@daily.com');
```

3. Products Table

```sql
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100),
    Category VARCHAR(50),
    Brand VARCHAR(50),
    Price DECIMAL(10,2),
    StockQuantity INT,
    SupplierID INT,
    FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID)
);
```

```sql
INSERT INTO Products VALUES
(1, 'Milk 1L', 'Dairy', 'Amul', 55.00, 120, 101),
(2, 'Wheat Flour 5kg', 'Grocery', 'Aashirvaad', 250.00, 80, 102),
(3, 'Hand Wash 250ml', 'Personal Care', 'Dettol', 75.00, 60, 103),
(4, 'Basmati Rice 1kg', 'Grocery', 'India Gate', 110.00, 50, 102),
(5, 'Chocolate Bar', 'Snacks', 'Dairy Milk', 40.00, 200, 104),
(6, 'Toothpaste 150g', 'Personal Care', 'Colgate', 90.00, 90, 103),
(7, 'Butter 500g', 'Dairy', 'Amul', 230.00, 30, 101),
(8, 'Cooking Oil 1L', 'Grocery', 'Fortune', 180.00, 70, 102);
```

4. Orders Table

```sql
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    TotalAmount DECIMAL(10,2),
    PaymentMethod VARCHAR(50),
    Status VARCHAR(50),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

```
INSERT INTO Orders VALUES
(301, 201, '2025-04-01', 305.00, 'UPI', 'Delivered'),
(302, 202, '2025-04-02', 110.00, 'Card', 'Delivered'),
(303, 204, '2025-04-03', 90.00, 'Cash', 'Pending'),
(304, 203, '2025-04-03', 230.00, 'UPI', 'Delivered'),
(305, 205, '2025-04-04', 250.00, 'Card', 'Delivered'),
(306, 206, '2025-04-04', 125.00, 'Cash', 'Pending'),
(307, 207, '2025-04-05', 180.00, 'UPI', 'Delivered'),
(308, 208, '2025-04-05', 55.00, 'UPI', 'Delivered');
```

5. Order Details Table

```
CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    UnitPrice DECIMAL(10,2),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```

```
INSERT INTO OrderDetails VALUES
(401, 301, 1, 2, 55.00),
(402, 301, 5, 2, 40.00),
(403, 302, 4, 1, 110.00),
(404, 303, 6, 1, 90.00),
(405, 304, 7, 1, 230.00),
(406, 305, 2, 1, 250.00),
(407, 306, 3, 1, 75.00),
(408, 306, 5, 1, 50.00);
```

6. Employees Table

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    FullName VARCHAR(100),
    Position VARCHAR(50),
    Phone VARCHAR(15),
    Email VARCHAR(100)
);
```

```
INSERT INTO Employees VALUES
(501, 'Rahul Mehta', 'Manager', '9812345678', 'rahul@supermart.com'),
(502, 'Kiran Sharma', 'Cashier', '9823456789', 'kiran@supermart.com'),
(503, 'Deepak Malhotra', 'Inventory Head', '9834567890', 'deepak@supermart.com'),
(504, 'Aarti Chawla', 'Cashier', '9845678901', 'aarti@supermart.com'),
(505, 'Mohit Arora', 'Security', '9856789012', 'mohit@supermart.com'),
(506, 'Preeti Yadav', 'Sales Exec', '9867890123', 'preeti@supermart.com'),
(507, 'Nikhil Bansal', 'Stock Clerk', '9878901234', 'nikhil@supermart.com'),
(508, 'Sneha Taneja', 'Customer Care', '9889012345', 'sneha@supermart.com');
```

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

NAAC GRADE A+
Accredited University

SQL QUERIES (13 Queries)

```
SELECT * FROM Products
WHERE Price > 100;
```

ProductID	ProductName	Category	Brand	Price	StockQuanti...	SupplierID
2	Wheat Flour 5kg	Grocery	Aashirvaad	250.00	80	102
4	Basmati Rice 1kg	Grocery	India Gate	110.00	50	102
7	Butter 500g	Dairy	Amul	230.00	30	101
8	Cooking Oil 1L	Grocery	Fortune	180.00	70	102
NULL	NULL	NULL	NULL	NULL	NULL	NULL

- ```
  SELECT Orders.OrderID, Customers.FullName, Orders.OrderDate, Orders.TotalAmount
  FROM Orders
  JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
  ```

OrderID	FullName	OrderDate	TotalAmount
301	Riya Sharma	2025-04-01	305.00
302	Aryan Kapoor	2025-04-02	110.00
304	Meena Kumari	2025-04-03	230.00
303	Rohit Verma	2025-04-03	90.00
305	Pooja Joshi	2025-04-04	250.00
306	Tanmay Singh	2025-04-04	125.00
307	Sneha Batra	2025-04-05	180.00
308	Aman Sethi	2025-04-05	55.00
  ```

```sql
SELECT * FROM Employees
WHERE Position = 'Cashier';
```

EmployeeID	FullName	Position	Phone	Email
502	Kiran Sharma	Cashier	9823456789	kiran@supermart.com
504	Aarti Chawla	Cashier	9845678901	aarti@supermart.com
NULL	NULL	NULL	NULL	NULL

```sql
SELECT Category, COUNT(*) AS ProductCount
FROM Products
GROUP BY Category;
```

Category	ProductCount
Dairy	2
Grocery	3
Personal Care	2
Snacks	1

- ```
  SELECT SUM(TotalAmount) AS TotalSales
  FROM Orders;
  ```

TotalSales
1345.00

- ```
  SELECT p.ProductName, s.SupplierName
  FROM Products p
  JOIN Suppliers s ON p.SupplierID = s.SupplierID
  WHERE s.SupplierName = 'Grocers India';
  ```

ProductName	SupplierName
Wheat Flour 5kg	Grocers India
Basmati Rice 1kg	Grocers India
Cooking Oil 1L	Grocers India

```sql
SELECT * FROM Products
ORDER BY Price DESC
LIMIT 1;
```

ProductID	ProductName	Category	Brand	Price	StockQuanti...	SupplierID
2	Wheat Flour 5kg	Grocery	Aashirvaad	250.00	80	102
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
SELECT * FROM Products
ORDER BY Price DESC;
```

ProductID	ProductName	Category	Brand	Price	StockQuanti...	SupplierID
2	Wheat Flour 5kg	Grocery	Aashirvaad	250.00	80	102
7	Butter 500g	Dairy	Amul	230.00	30	101
8	Cooking Oil 1L	Grocery	Fortune	180.00	70	102
4	Basmati Rice 1kg	Grocery	India Gate	110.00	50	102
6	Toothpaste 150g	Personal Care	Colgate	90.00	90	103
3	Hand Wash 250ml	Personal Care	Dettol	75.00	60	103
1	Milk 1L	Dairy	Amul	55.00	120	101
5	Chocolate Bar	Snacks	Dairy Milk	40.00	200	104
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```sql
SELECT od.OrderID, p.ProductName, od.Quantity, od.UnitPrice
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID
WHERE od.OrderID = 301;
```

OrderID	ProductName	Quantity	UnitPrice
301	Milk 1L	2	55.00
301	Chocolate Bar	2	40.00

```sql
SELECT COUNT(*) AS DeliveredOrders
FROM Orders
WHERE Status = 'Delivered';
```

DeliveredOrders
6

```sql
SELECT DISTINCT c.FullName, o.OrderID
FROM Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID
WHERE o.OrderDate = '2025-04-03';
```

FullName	OrderID
Rohit Verma	303
Meena Kumari	304

```sql
SELECT c.FullName, COUNT(o.OrderID) AS OrderCount
FROM Customers c
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.FullName;
```

FullName	OrderCount
Riya Sharma	1
Aryan Kapoor	1
Meena Kumari	1
Rohit Verma	1
Pooja Joshi	1
Tanmay Singh	1
Sneha Batra	1
Aman Sethi	1

```
SELECT o.OrderID, c.FullName, o.OrderDate, o.TotalAmount
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
WHERE o.Status = 'Pending';
```

OrderID	FullName	OrderDate	TotalAmount
303	Rohit Verma	2025-04-03	90.00
306	Tanmay Singh	2025-04-04	125.00

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

NAAC GRADE A+
Accredited University

SUMMARY

The SuperMart Database Management System is a comprehensive solution designed to manage and streamline the core operations of a retail supermarket. This relational database effectively handles customer information, product inventory, supplier records, employee details, and the complete order lifecycle.

The system comprises six well-structured tables — **Customers, Orders, OrderDetails, Products, Suppliers, and Employees** — each connected through primary and foreign keys to maintain data integrity and avoid redundancy. Key relationships such as **Customer placing Orders**, **Orders containing Products**, and **Suppliers supplying Products** are modeled using one-to-many and many-to-many relationships.

The database was implemented using **MySQL**, with the support of tools like MySQL Workbench and ER modeling platforms. Through SQL queries, the system supports various operations including sales tracking, inventory management, and order history retrieval, providing essential insights into business performance.

Overall, the SuperMart database lays a strong foundation for automating supermarket operations, ensuring accurate record-keeping, efficient data management, and enhanced decision-making capabilities for both administrators and stakeholders.

CONCLUSION

Observations:

- The Hospital Management System database successfully demonstrates the organisation and management of hospital-related data such as patients, doctors, staff, departments, appointments, billing, and pharmacy.
- The use of SQL queries allows for effective data retrieval, patient tracking, appointment scheduling, and billing generation.
- Proper relational mapping using foreign keys ensures data consistency and integrity across all entities.
- The ER diagram and schema design provide a clear and normalised structure that supports both current hospital needs and future scalability.
- Complex queries like grouping, filtering, and joining across multiple tables have been efficiently implemented.

Limitations:

- This project is limited to backend database implementation and lacks a user-friendly frontend interface for hospital staff or patients.
- The pharmacy system is not directly linked to prescriptions in medical records, so medication usage tracking is not automated.
- There is no role-based access or login system for doctors, staff, or administrators.
- Real-time alerts for low medicine stock, upcoming appointments, or unpaid bills are not part of the current scope.
- Advanced features like data analytics, reporting dashboards, or integration with real-world hospital software (e.g., EHR systems) are not included.

Future Scope

- Development of a **web or mobile interface** for smooth and interactive user experience.
- Addition of **automated triggers** for stock alerts, billing, or order confirmations.
- Integration with **analytics platforms** (e.g., Power BI, Tableau) for better insights.
- Implementation of **user roles and permissions** to enhance system security.
- Deployment on **cloud platforms** for scalability, accessibility, and remote operations.