TED UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

CMPE 232

2024 Fall

Term Project Design Report for

Relational Model and Database Structures

Info

Group Name:TED CowCode Innovators

Project Name: Tracking and optimizing milk production and distribution

Group Members (Name Surname - ID):

İrem Özbağcı 22219520070

Ece Sezginer 10201193074

Mert Efe Şensoy 11650180158

Tardu Yüce Yavaş 10759069142

I. Introduction

II. Relational Model

a. Definition:

The relational model used in this project organizes data into tables that represent entities critical to tracing milk from farms to retail outlets. For instance, tables like Products, Inventory, Orders, and Shipments capture essential details about the milk supply chain. Each table is uniquely identified by a primary key, such as ProductID or OrderID, and relationships between tables are established using foreign keys (e.g., ProductID in Inventory links to Products). This structure allows for tracking product details, monitoring inventory levels, and managing supplier and customer interactions while ensuring data consistency through constraints and relationships.

b. Database Schema:

Table	Description
Products	Stores information about the milk products, including name, brand, flavor, packaging size, and price.
Inventory	Tracks the availability of products in warehouses, including the location and quantity available.
Suppliers	Maintains records of suppliers, including their contact information and address.
Orders	Logs customer orders, including the total amount and date of the order.
Customers	Stores customer details such as contact information, address, and loyalty points.
Shipments	Tracks shipments associated with orders, including shipment dates and estimated delivery times.
Employees	Contains information about employees responsible for managing operations.

1) Relationships:

- **One-to-One Relationship (Products to Inventory):**

Each product corresponds to a single inventory entry, representing its availability. This relationship is established using the foreign key ProductID in the Inventory table referencing the Products table.

- **Many-to-Many Relationship (Products to Suppliers):**

A single product can be supplied by multiple suppliers, and a supplier can provide multiple

products. This is an identifying relationship as SupplierProducts is a weak entity that depends on both Suppliers and Products for its identification. The composite primary key in SupplierProducts consists of SupplierID and ProductID, linking each record to specific suppliers and products.

- **Many-to-One Relationship (Orders to Customers):**

Each order is placed by one customer, but a customer can place multiple orders over time. This is implemented with a foreign key (CustomerID) in the Orders table referencing the Customers table.

- **Many-to-Many Relationship (Orders to Products through OrderDetails):**

An order can include multiple products, and a product can appear in multiple orders. This relationship is handled through the OrderDetails table, which includes foreign keys for both OrderID (referencing Orders) and ProductID (referencing Products). This is another identifying relationship.

- **One-to-One Relationship (Shipments to Orders):**

Each order is associated with a single shipment, and each shipment corresponds to one order. This is enforced using a foreign key (OrderID) in the Shipments table referencing the Orders table.

- **One-to-Many Relationship (Employees to Orders):**

Each employee is responsible for processing multiple orders, but each order is handled by one employee. This relationship is established using a foreign key (EmployeeID) in the Orders table referencing the Employees table.

2) Normalization:

It is demonstrated here that all attributes depend directly on the primary key, ensuring the tables are in 3NF.

1. Products

- **Schema**: {ProductID, Name, Brand, Flavor, PackSize, Price, LastModified}
- Normalized to 3NF. Each product attribute depends solely on the primary key (ProductID) with no other dependencies.

2. Inventory

- **Schema**: {InventoryID, ProductID, QuantityAvailable, WarehouseLocation, LastModified}
- Normalized to 3NF. Attributes like QuantityAvailable and WarehouseLocation depend only on InventoryID, which is the primary key.

3. Suppliers

- **Schema**: {SupplierID, Name, ContactInfo, Address, LastModified}
- Normalized to 3NF. All attributes, including ContactInfo and Address, are directly related to SupplierID.

4. Orders

- **Schema**: {OrderID, CustomerID, OrderDate, TotalAmount, LastModified}
- Normalized to 3NF. Attributes such as OrderDate and TotalAmount depend solely on the primary key (OrderID).

5. Customers

- **Schema**: {CustomerID, Name, ContactInfo, Address, LoyaltyPoints, LastModified}
- Normalized to 3NF. Customer attributes like Name and LoyaltyPoints are directly dependent on CustomerID.

6. Shipments

- **Schema**: {ShipmentID, OrderID, ShipmentDate, EstimatedDeliveryDate, LastModified}
- Normalized to 3NF. All shipment-related attributes are directly dependent on the primary key (ShipmentID).

7. Employees

- **Schema**: {EmployeeID, Name, Role, ContactInfo, LastModified}
- Normalized to 3NF. Employee details like Name and Role are linked to EmployeeID without redundant data.

8. Additional junction table: ProductSuppliers

- **Schema**: {ProductID, SupplierID}
- Normalized to 3NF. Both ProductID and SupplierID are the primary keys, and there are no non-key attributes in this table. There are no partial dependencies, and it ensures that each product-supplier relationship is uniquely defined.

9. Additional junction table: OrderDetails

- **Schema**: {OrderID, ProductID, Quantity, Price}
- Normalized to 3NF. Attributes such as Quantity and Price depend only on the combination of OrderID and ProductID. There are no partial dependencies because the primary key is composite (composed of both OrderID and ProductID).

By applying these principles, the database is designed to reduce redundancy, ensure data integrity, and enhance performance.

III. Database Structures

This section presents vast knowledge and understanding of the database structures that were applied in the Tracking and Optimizing Milk Production and Distribution Project. It encompasses the structure of tables, indices, views and triggers that improve the integrity and performance of the system.

Table Structures:

1. **Products**
    - **Columns**:
        - ProductID (INT, AUTO_INCREMENT, PRIMARY KEY)
        - Name (VARCHAR (255), NOT NULL)
        - Brand (VARCHAR (255))
        - Flavor (VARCHAR (255))
        - PackSize (VARCHAR (50))
        - Price (DECIMAL (10,2), NOT NULL, CHECK (Price >= 0))
        - LastModified (DATETIME, DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP)
    - **Constraints**:
        - ProductID is primary key.
        - Price must be non negative.

- **Relationships**:
  - One to one with Inventory.
  - Many to many with Suppliers.

2. **Inventory**
   - **Columns**:
     - InventoryID (INT, AUTO_INCREMENT, PRIMARY KEY)
     - ProductID (INT, FOREIGN KEY REFERENCES Products (ProductID))
     - QuantityAvailable (INT, NOT NULL, DEFAULT 0, CHECK (QuantityAvailable >= 0))
     - WarehouseLocation (VARCHAR (255))
     - LastModified (DATETIME, DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP)
   - **Relationships**:
     - ProductID provides a direct link to products.
3. **Orders**
   - **Columns**:
     - OrderID (INT, AUTO_INCREMENT, PRIMARY KEY)
     - CustomerID (INT, FOREIGN KEY REFERENCES Customers (CustomerID))
     - OrderDate (DATETIME, NOT NULL)
     - TotalAmount (DECIMAL (10,2), CHECK (TotalAmount >= 0))
     - LastModified (DATETIME, DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP)
   - **Constraints**:
     - OrderDate is required.
   - **Relationships**:
     - Many to Many with Products through OrderDetails.
4. **OrderDetails**
   - **Columns**:
     - OrderDetailID (INT, AUTO_INCREMENT, PRIMARY KEY)
     - OrderID (INT, FOREIGN KEY REFERENCES Orders (OrderID))
     - ProductID (INT, FOREIGN KEY REFERENCES Products (ProductID))
     - Quantity (INT, NOT NULL, CHECK (Quantity > 0))
     - UnitPrice (DECIMAL (10,2), CHECK (UnitPrice >= 0))
     - LastModified (DATETIME, DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP)
   - **Relationships**:
     - Aggregates Orders and Products.

Like Customers and Suppliers tables, a Shipments table and an Employees table also observe the same conventions with relevant relationships created for data accuracy and consistency.

Indices:

• Primary Indices: Automatically created on primary keys for all tables.
• Secondary Indices:
On the LastModified columns for efficient audit queries.
On OrderDetails' ProductID and OrderID for fast access to information in aggregate queries.

Views:

• ProductSummaryView: Presents products and their stock available for order including the suppliers of the products.

```
CREATE VIEW ProductSummaryView AS
SELECT p.ProductID, p.Name, p.Brand, i.QuantityAvailable, s.Name AS SupplierName
FROM Products p
LEFT JOIN Inventory i ON p.ProductID = i.ProductID
LEFT JOIN SupplierProducts sp ON p.ProductID = sp.ProductID
LEFT JOIN Suppliers s ON sp.SupplierID = s.SupplierID;
```

- OrderSummaryView: Combines ordering data and order customers with final amounts.

```
CREATE VIEW OrderSummaryView AS
SELECT o.OrderID, c.Name AS CustomerName, o.OrderDate, o.TotalAmount
FROM Orders o
INNER JOIN Customers c ON o.CustomerID = c.CustomerID;
```

Triggers:

- Trigger for Inventory Updates:

```
CREATE TRIGGER UpdateInventoryAfterOrder
AFTER INSERT ON OrderDetails
FOR EACH ROW
BEGIN
  UPDATE Inventory
  SET QuantityAvailable = QuantityAvailable - NEW.Quantity
  WHERE ProductID = NEW.ProductID;
END;
```

- Trigger for Order Total Calculation: Ensures accurate totals for Manage Orders.

```
CREATE TRIGGER CalculateOrderTotal
AFTER INSERT OR UPDATE ON OrderDetails
FOR EACH ROW
BEGIN
  UPDATE Orders
  SET TotalAmount = (
    SELECT SUM (Quantity * UnitPrice)
    FROM OrderDetails
    WHERE OrderID = NEW.OrderID
  )
  WHERE OrderID = NEW.OrderID;
END;
```

IV. UI Design

1. Login/Registration Page

Purpose: Allow users (admin, employees, or customers) to log in and register.

Features:

User roles selection during registration (e.g., Admin, Quality Control, Inventory Manager, Customer).

Input fields: Username, Password (with show/hide toggle), Confirm Password.

Buttons: "Login", "Register", "Forgot Password".

Feedback: Error messages for incorrect credentials or incomplete fields.

Design Concept: Use a clean and professional interface with your project's branding.

2. Admin Dashboard

Purpose: Central hub for administrators to manage the system.

Features:

Panels for quick statistics: Total Products, Active Shipments, Quality Checks Pending.

Links to:

Manage Users

Manage Inventory

Manage Suppliers

View Reports

Notification panel for alerts (e.g., inventory low, quality issues).

Design Concept: Intuitive layout with a side navigation menu and widgets for quick actions.

3. Manage Inventory

Purpose: Track and update inventory across warehouses.

Features:

Table view of inventory items with fields: Product Name, Quantity, Location, Last Updated.

Actions: Edit, Add New, Delete.

Search and filter options (e.g., by location or product type).

Design Concept: Tabular data with collapsible filters for clarity.

4. Quality Control

Purpose: Monitor and log quality checks.

Features:

Forms for inputting temperature and humidity data.

History view of past quality checks by lot.

Alerts for deviations from acceptable thresholds.

Design Concept: Bright color coding for status (e.g., green for acceptable, red for issues).

5. Manage Orders

Purpose: Oversee customer orders and track shipments.

Features:

List of orders with fields: Order ID, Customer Name, Status (Pending, Shipped, Delivered), Total.

Detailed view for individual orders: Items, Quantities, Delivery Status.

Options to update status and assign shipments.

Design Concept: Use progress indicators for shipment status.

6. Manage Shipments

Purpose: Optimize routes and track shipment statuses.

Features:

Input for shipment details: Order ID, Route, Driver, Estimated Delivery.

View real-time tracking and timestamps for key stages (Dispatched, In-Transit, Delivered).

Alerts for delays or rerouting suggestions.

Design Concept: Map integration for route visualization.

7. Customer Interface

Purpose: Simplify customer interactions.

Features:

View orders and shipment tracking.

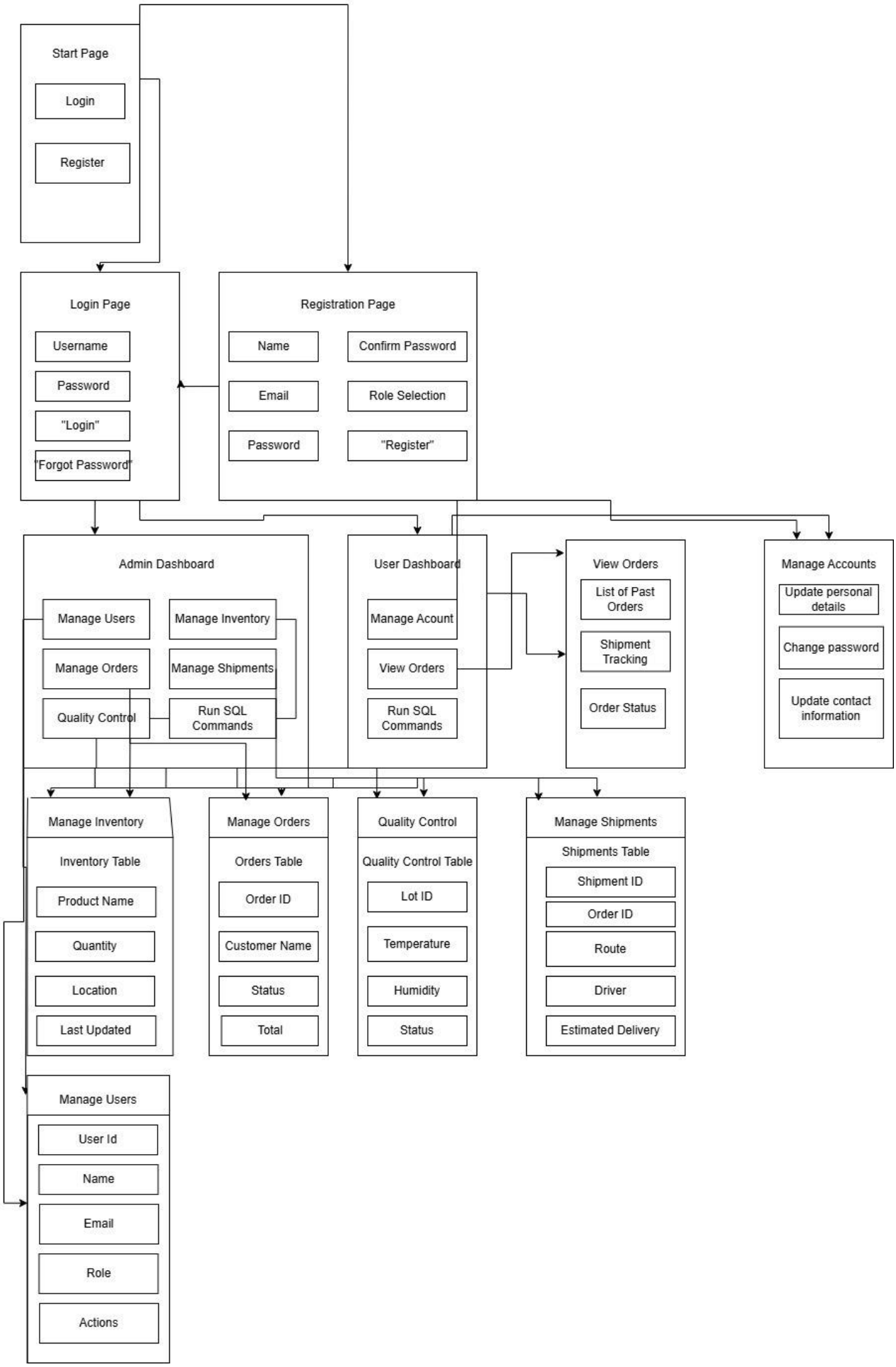
Loyalty points and promotional offers.

Contact support or place new orders.

Design Concept: User-friendly with minimal navigation

UI Design Flow

Create a UI Design diagram like the example below:


Start Page
Login
Register
Login Page
Username
Password
"Login"
"Forgot Password"
Registration Page
Name
Confirm Password
Email
Role Selection
Password
"Register"
Admin Dashboard
Manage Users
Manage Inventory
Manage Orders
Manage Shipments
Quality Control
Run SQL Commands
User Dashboard
Manage Acount
View Orders
Run SQL Commands
View Orders
List of Past Orders
Shipment Tracking
Order Status
Manage Accounts
Update personal details
Change password
Update contact information
Manage Inventory
Inventory Table
Product Name
Quantity
Location
Last Updated
Manage Orders
Orders Table
Order ID
Customer Name
Status
Total
Quality Control
Quality Control Table
Lot ID
Temperature
Humidity
Status
Manage Shipments
Shipments Table
Shipment ID
Order ID
Route
Driver
Estimated Delivery
Manage Users
User Id
Name
Email
Role
Actions

V. Workload Division

All parts of the project were distributed to the group members and eventually everyone ended up helping each other and everyone took part in most of the steps of the project.

VI. References

Ataseven , Z. Y. (2023). *Süt Ve süt ürünleri durum Tahmin Raporu 2023*. Tarımsal Ekonomi ve Politika Geliştirme Enstitüsü . https://arastirma.tarimorman.gov.tr/tepge/Belgeler/PDF Durum-Tahmin Raporları/2023 Durum-Tahmin Raporları/Süt ve SüT Ürünleri Durum Tahmin Raporu 2023-372 TEPGE.pdf

Chen, C., Zhang, J., & Delaurentis, T. (2013). Quality control in food supply chain management: An analytical model and case study of the adulterated milk incident in China. *International Journal of Production Economics*, *152*, 188–199. https://doi.org/10.1016/j.ijpe.2013.12.016

Xiu, C., & Klein, K. (2010). Melamine in milk products in China: Examining the factors that led to deliberate use of the contaminant. *Food Policy*, *35*(5), 463–470. https://doi.org/10.1016/j.foodpol.2010.05.001