**StockPulse: Real-Time Inventory Insights**

**CS-603-B**

**Abhishek Dhanani**

12/07/2024

**Project Description**

**"StockTrack: An Inventory Management System"** is a relational database designed to streamline the management of inventory data. This system provides a robust and efficient way to manage products, suppliers, stores, stock levels, and sales transactions. With features like tracking stock movements, analysing sales performance, and maintaining supplier relationships, **StockTrack** ensures accuracy and accessibility for all inventory-related operations.

The system includes user-specific views tailored for different roles, such as store managers, suppliers, warehouse supervisors, sales analysts, and pricing teams, making it easy to extract meaningful insights. By leveraging normalized database structures and user access controls, **StockTrack** is built to enhance operational efficiency and decision-making for inventory management.

**Key Features**:

* Organized tracking of products, suppliers, and stores.
* Streamlined monitoring of stock levels.
* Comprehensive sales analysis for performance tracking.
* Simplified data access for users with tailored views.
* Scalable database design for efficient management.

This system provides a reliable foundation for maintaining a streamlined inventory and supports the organization's goals of accuracy, efficiency, and scalability.

**User Views**

**Store manager view**

This view provides details about the products available in each store, including their categories, stock quantities, and the store locations. Helps store managers track product availability and manage inventory efficiently.

|  |  |  |  |
| --- | --- | --- | --- |
| Date: 11/26/2024 | | Page: 1 | |
| **Store inventory report** | | | |
| Store: 201 | | Location: New York | |
| ProductID | Product Name | Category | StockQuantity |
| 1001 | Apple | Fruit | 100 |
| 1002 | Orange | Fruit | 150 |

**Supplier View**

Displays information about suppliers, the products they supply, order quantities, unit prices, and total order prices. Assists in supplier relationship management and tracking supply orders.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date: 22/24/2024 | |  | Page: 1 | |
| **Supplier delivery report** | | | | |
| SupplierId: 301 | |  | Name: FreshFarm | |
| ProductId | ProductName | OrderQuantity | UnitPrice | Total |
| 1 | Apple | 50 | 2 | 100 |

**Sales Analysis Report**

Shows sales data for each product, including the units sold, product categories, and revenue generated. Provides insights into product performance and helps identify top-selling items.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date: 11/23/2024 | |  | Page:1 | |
| **Sales Performance Report** | | | | |
| ProductId | ProductName | Category | Unitsold | Revenue |
| 1001 | Apple | Fruits | 150 | $300.00 |
| 1002 | Orange | Fruits | 120 | $300.00 |

**Warehouse Manager View**

Focuses on inventory data, showing stock quantities for products at various store locations. Enables warehouse managers to monitor and manage stock distribution across stores.

|  |  |  |  |
| --- | --- | --- | --- |
| Date:11/26/2024 | | | Page:1 |
| **Warehouse Stock Report** | | | |
| ProductId | ProducatName | StockQuantity | StoreLocation |
| 1001 | Apple | 100 | New York |
| 1002 | Orange | 150 | Los Anagles |

**Inventory Auditor View**

Provides an overview of the inventory, showing stock quantities for products along with the associated store IDs. Supports auditors in verifying inventory accuracy and ensuring stock levels are maintained.

|  |  |  |  |
| --- | --- | --- | --- |
| Date: 11/23/2024 | | Page: 1 | |
| **Inventory verification Report** | | | |
| ProductId | ProductName | Stockquantity | StoreId |
| 1001 | Apple | 100 | 201 |
| 1002 | Orange | 150 | 202 |

**Pricing team**

Displays product pricing information, including current prices for all items. Assists the pricing team in analysing and updating product prices.

|  |  |  |
| --- | --- | --- |
| Date: 11/23/2024 |  | Page:1 |
| **Product Pricing Report** | | |
| ProductId | ProductName | CurrentPrice |
| 1001 | Apple | $2.00 |
| 1002 | Orange | $2.50 |

**Normalization**

**Store Manager View**

**1 NF: (**ProductId, ProductName, Category, StockQuantity, StoreID, StoreLocation**)**

**2 NF:** Product (ProductID, ProductName, Category)

Store (StoreID**,** StoreLocation)

Inventory (ProductId, StoreId, StockQuantity)

**3 NF:** there are no transitive dependencies thus, all the tables are already in 3 NF.

Product (**ProductID**, ProductName, Category)

Store (**StoreID,** StoreLocation)

Inventory (**ProductId, StoreId**, StockQuantity)

**Supplier View**

**1 NF:** (SupplierId, SupplierName, ProductId, ProductName, OrderQuantity, UnitPrice, Total)

**2 NF:** Supplier (SupplierId, SupplierName)

Product (ProductId, ProductName, UnitPrice)

ProductSupplier (ProductId, SupplierId, OrderQuantity, Total)

**3 NF:** there are no transitive dependencies thus, all the tables are already in 3 NF.

Supplier (**SupplierId**, SupplierName)

Product (**ProductId**, ProductName)

ProductSupplier (**ProductId, SupplierId**, OrderQuantity)

**Sales Analysis View**

**1 NF:** (ProdcutId, ProductName, category, unitsold, Revenue)

**2 NF, 3NF:** we don’t have any partial dependency or transitive dependency. So, it is in 3 NF.

Product (**ProdcutId**, ProductName, category, unitsold, Revenue)

**Warehouse Manager View**

**1 NF:** (ProductId, ProductName, StockQuantity, StoreLocation)

**2 NF:** Product (**ProductId**, ProductName)

Inventory (**ProductId**, storeLocation, StockQuantity)

**3 NF:** there are no transitive dependencies thus, all the tables are already in 3 NF.

**Inventory Auditor**

**1 NF:** (ProductId, ProductName, Stockquantity, StoreId)

**2 NF:** Product (**ProductID**, ProductName, Category)

Store (**StoreID**, StoreLocation)

Inventory (**ProductId, StoreId**, StockQuantity)

**3 NF:** there are no transitive dependencies thus, all the tables are already in 3 NF.

**Pricing Team**

**1 NF:** Product(ProductId, ProductName, CurrentPrice)

**2 NF, 3 NF:** we don’t have any partial dependency or transitive dependency. So, it is in 3 NF.

Product(**ProductId**, ProductName, CurrentPrice)

**Final Tables:**

Products (ProductID, ProductName, category, CurrentPrice)

Stores (StoreID, StoreLocation)

Inventory (ProductID, StoreID, StockQuantity)

Suppliers (SupplierID, SupplierName)

ProductSupplier (ProductID, SupplierID, OrderQuantity)

Sales (ProductID, Unitsold)

**ER Diagram**

**(1, 1)**

**(1, M)**

**(1, M)**

**(M, 1)**

**(1, M)**

**Product supplier**

**Products**

**Inventory**

**Stores**

**Sales**

**Suppliers**

**ER Diagram Explanation**

**Products → Inventory (One-to-Many):** A single product can be stored in multiple stores (or inventory locations), but each inventory record corresponds to only one product.

**Stores → Inventory (One-to-Many):** A single store can have multiple inventory records, each for a different product, but each inventory record belongs to only one store.

**Suppliers → ProductSupplier (One-to-Many):** A single supplier can supply multiple products, but each product-supplier combination is recorded separately in the ProductSupplier table.

**Products → ProductSupplier (One-to-Many):** A single product can be supplied by multiple suppliers, but each product-supplier combination is recorded separately in the ProductSupplier table.

**Products → Sales (One-to-One):** Each product can have only one sales record, and each sales record corresponds to only one product.

**ATTRIBUTE DOMAIN**

**Products Table**

|  |  |
| --- | --- |
| **Attributes** | **Data type** |
| ProductID | INTEGER |
| ProductName | VARCHAR (50) |
| Category | VARCHAR (30) |
| CurrentPrice | DECIMAL (10, 2) |

**Stores Table**

|  |  |
| --- | --- |
| **Attributes** | **Data type** |
| StoreID | INTERGER |
| StoreLocation | VARCHAR (50) |

**Inventory Table**

|  |  |
| --- | --- |
| **Attributes** | **Data type** |
| ProductID | INTERGER |
| StoreID | INTERGER |
| StockQuantity | INTERGER |

**Supplier Table**

|  |  |
| --- | --- |
| **Attributes** | **Data type** |
| SupplierID | INTEGER |
| SupplierName | VARCHAR (50) |

**ProductSupplier Table**

|  |  |
| --- | --- |
| **Attributes** | **Data type** |
| ProductID | INTEGER |
| SupplierID | INTEGER |
| OrderQuantity | INTEGER |

**Sales Table**

|  |  |
| --- | --- |
| **Attributes** | **Data type** |
| ProductID | INTEGER |
| UnitSold | INTEGER |

**Attribute – User view Matrix**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Attributes** | **Store manager (User view 1)** | **Supplier (User view 2)** | **Sales Analysis (User view 3)** | **Warehouse Manager (User view 4)** | **Inventory Auditor (User view 5)** | **Pricing Team (User view 6)** |
| **ProductID** | **Close with solid fill** | **Close with solid fill** | **Close with solid fill** | **Close with solid fill** | **Close with solid fill** | **Close with solid fill** |
| **ProductName** | **Close with solid fill** | **Close with solid fill** | **Close with solid fill** | **Close with solid fill** | **Close with solid fill** | **Close with solid fill** |
| **Category** | **Close with solid fill** |  | **Close with solid fill** |  |  |  |
| **CurrentPrice** |  | **Close with solid fill** | **Close with solid fill** |  |  | **Close with solid fill** |
| **StoreID** | **Close with solid fill** |  |  |  | **Close with solid fill** |  |
| **StoreLocation** | **Close with solid fill** |  |  | **Close with solid fill** |  |  |
| **StockQuantity** | **Close with solid fill** |  |  | **Close with solid fill** | **Close with solid fill** |  |
| **SupplierID** |  | **Close with solid fill** |  |  |  |  |
| **SupplierName** |  | **Close with solid fill** |  |  |  |  |
| **OrderQuantity** |  | **Close with solid fill** |  |  |  |  |
| **UnitPrice** |  | **Close with solid fill** |  |  |  | **Close with solid fill** |
| **TotalOrderPrice** |  | **Close with solid fill** |  |  |  |  |
| **UnitSold** |  |  | **Close with solid fill** |  |  |  |
| **Revenue** |  |  | **Close with solid fill** |  |  |  |

**Note:** UnitPrice, TotalOrderQuantity, Revenue these attributes are calculate at the time of select query so they are not in tables.

**Business Rules**

* Quantity on hand (StockQuantity) must not be less than 0.
* Unit price (CurrentPrice) must be at least $0.01 and not exceed $10,000.
* Order quantity (OrderQuantity) must be greater than 0.
* Units sold (UnitsSold) must not be less than 0.
* Revenue and total order price should always be dynamically calculated based on sales and order quantities, respectively.