

Project Documentation of

“Inventory Management System”

Course Teacher:

Sir M. Malik Ali

Lab Instructor:

Miss Salas Akbar

Students:

Fatima Butt (02-134201-037)

Muhammad Nawfal Burhan (02-134201-035)

Mariam Reem Siddiqui(02-134201-030)

# **Table of Contents:**

1. **Acknowledgment** ------------------------------------page 4

Chapter 1: introduction

1. Introduction to the system ------------------------------------page 5
2. Background of the system ------------------------------------page 5
3. Motivation ------------------------------------page 6
4. Methodology ------------------------------------page 7
5. Brief System Description ------------------------------------page 7
6. Tools used/ System Specification ------------------------------------page 8

Chapter 2: Designed System

1. Complete Description of the work ------------------------------------page 9
2. Normalization Description ------------------------------------page 9
3. ER Diagram of the Mgt system ------------------------------------page 12
4. Schema Of the Mgt System ------------------------------------page 13
5. Screenshots of the system with proper explanation ----------------------page 14

Chapter 3: Future work and conclusion

1. Problems Faced ------------------------------------page 20
2. Future Work ------------------------------------page 20
3. Conclusion ------------------------------------page 20

# **Acknowledgment:**

We have taken serious efforts in this project. It would not have been possible without the support of our teachers “Sir Malik”, and “Miss Salas”. We are grateful for the excellent quality of lectures they have provided us that made us achieve this project. We would also like to thank many YouTube channels that are sincerely passionate to educate the young minds of our generation without any cost. We would also like to express our special gratitude towards our parents for always supporting us from the beginning until the end. Lastly, we are grateful to each other for becoming a great team and completing this project and it would not have been easy without each other’s help.

Chapter 1: Introduction

# **Introduction to the system:**

Inventory Management comprises the storage and processing of raw materials, components, and end products, as well as the administration of raw materials, components, and final products.

An inventory system’s main function is to keep track of your products and supplies. When you buy inventory, you need to keep track of when you bought it, when you sold it, and how much you have on hand.

To make a change and provide easiness in these activities, a database of computerized inventory management should be done. Through this, every inventory management would be much easier and will not require a lot of effort. This will also secure every record and inventory transaction for important purposes.

# **Background of the system:**

The earliest form of inventory management dates back over 50,000 years in which people used “tally sticks” to count. Over time, inventory management developed into slightly more accurate systems of accounting and record-keeping, particularly in ancient Greek and Egyptian societies.

The start of the second industrial revolution brought incredible breakthroughs for inventory management. Herman Hollerith, an American inventor, developed the first modern automatic computation machine. Replacing pen and paper and saving countless hours, the tabulator and sorter machine was specially designed to record information using punch cards. These punch cards allowed people to record many types of data, including inventory.

Since the mid-1970s, the barcode has predominately been many industries’ primary inventory management tool. In the late 1940s, Norman Woodland invented the barcode due to a request from a distraught grocery store owner who needed help keeping track of inventory.

The first to take advantage of the technology included the National Association of Food Chains, which used them to decrease check-out times.

# **Motivation**

1. Accomplishing Organizational Goals:

All members of an organization strive to achieve objectives and work efficiently regardless of the circumstances, our motivation is to help achieve those goals for companies.

1. Best Utilization of Resources:

Instead of taking up a lot of resources like pen and paper to store all information, a database such as this one can reduce the utilization of resources and save space. All information can be stored in the size of a bag.

1. Reduction in Labor Problems:

All the members try to concentrate their efforts to achieve the objectives of the organization and carry out the plans by the policies and programs laid down by the organization if the management introduces motivational plans. It reduces the labor problems like management confusion, loss of tracking information, etc.

1. Acceptance of Organizational Change:

An organization must incorporate changes that are to occur, to cope with the requirement of the time. If people are effectively motivated, they gladly accept, introduce, and implement these changes without reserving any resistance to change and negative attitude, thus, keeping the organization on the right trade of progress. With our database management system, we hope to bring acceptance in this fast-evolving era of technology

1. Improvement Upon Skill and Knowledge:

All the members will try to be as efficient as possible and will try to improve upon their skill and knowledge so that they may be able to contribute to the progress of the organization as much as possible because they know that they, in turn, will get what has been promised and ultimately, they will be able to satisfy their needs

# **Methodology:**

**Phase 1: An Idea:**

The first step was to research the project which is currently running in the market. We analyzed the database application inflow and got an idea of what is required. Reverse engineering of the system was done to break down it into further tables.

**Phase 2: Basic Model and planning:**

Firstly, tables calculation was done and required tables were formed. An ERD was then formed having relationships specified between each table which were normalized to prevent future disturbance between tables.

**Phase 3: Implementation of the system**

In the implementation phase, we firstly created single nondependent tables (Vendor, retailer, retailer order, customer, customer order, products, end products, shipper, employee, taxing, godown ) and then created bridge tables between required tables keeping in mind the normalization.

**Phase 4: Testing**

After the implementation of tables queries were created to run a test between several tables of the database.

**Phase 5: Report writing**

After four successful phases of testing and implementation which were conducted on the database, a final report explaining in detail the inventory management system was created.

# **Brief System Description:**

Separate tables are as follows:

• **vendor**: vendor is the main person who provides the parts in bulk to the retailer.

• **retailer**: the retailer is basically a whole seller who will be further selling products to customer

• **retailerOrder**: retailer order is the order which a retailer places to vendor

• **Customer**: the customer is the person who purchases from the retailer. Customer data is stored in this table.

• **CustomerOrder**: this table has the order details of a customer.

• **products**: this table has the basic parts which can be purchased by customers or vendors.

• **End Products**: This table has finished products made from products which a customer orders

• **employee**: Employee is the person who works in a godown and collets order for the retailer from the shipper

• **shipper**: shipper is the shipping company employee that delivers the order from vendor to retailer.

• **godown**: godown is the storage area where products and end products are stored.

• **Taxing**: taxing is the table that tells the percentage tax was in a specific year.

* **Location**: The location will store all the location details of the respective orders.

# **Tools used/ System Specification**

The software we have used to create our management system is Microsoft SQL Server Management Studio 18.

1. **System specifications:**
2. Operating System (Windows)

We have chosen Windows operating system for its best support and user-friendliness.

1. Database (SSMS)

To save records, we have chosen SQL server management studio.

1. **Requirements:**

-OS: Windows 7 or later

-1.8 GHz or faster x86 (Intel, AMD) processor. Dual-core or better --recommended 2 GB of RAM; 4 GB of RAM recommended (2.5 GB minimum if running on a virtual machine)

-Hard disk space: Minimum of 2 GB up to 10 GB of available space

Chapter 2: Designed System

# **Complete Description of the work:**

The working of this system is easy to understand. The database starts from a vendor who supplies a product to a retailer product in bulk which is ordered by a retailer. In this management system, we are considered to be retailers that order from vendors and deliver products to our customers. After the retailer orders the product and receives it, a customer can order a product or an end product which we retailers can deliver to them. There is a shipper who ships products and end products to a retailer or a customer. Every Retailer and vendor has a godown. There is a location for a godown and the address where a product needs to be delivered. There is a taxing scheme for every customer’s purchase. The Employees work at a godown and assemble an end product to be delivered.

# **Normalization Description:**

**Tables:**

Vendor, Product, Ret\_Order, godown, Taxing, VenProdRet\_OrdTaxBridge, Retailer, RetRet\_ordBridge, CustomerOrder, RetProdCust\_OrdTaxBridge, cust, Locations, Employee, LocCustomer\_ordShipBridge, Shipper, End\_product, CustCust\_OrdBridge, LocShipperEmpRet\_OrdBridge, CustomerOrdProdEnd\_ProdBridge, End\_ProdEmployeeGodownBridge, ProductGodownBridge, RetailerLocationGodownBridge, VendorLocationGodownBridge, VenRetail\_TRANSACTION\_Bridge

**1NF:**

**VendorID, ProductID, RetOrderID, RetailerID, CustomerOrderID, CustomerID, PostalCode, EmployeeID, ShipperID, End\_ProductID, GodownID, TaxingID**, TaxingDate, TaxingPercentage Capacity, status End\_ProductName, Category, Description, Cost, Sale\_Price, Quantity, Weight, length, width, height ShipperName, shipperPhone\_no, ShipperLoadLimit Name, Address, Phone\_no, Email, Status, Gender, Designation, EmployeeSalary country, city, area, street, plot\_no, flat\_no custName, custPhone\_No, custEmail OrderStatus, CustomerOrderPlacedDate, Description RetailerName, RetailerPhone\_No, RetailerEmail, RetailerWebsite,

OrderStatus, RetOrderPlacedDate, Description ProductName, Category, Description, Cost, Sale\_Price, Quantity, Weight, length, width, height vendorName, vendorPhone\_no, vendorEmail, Vendorwebsite

**2NF:**

**Vendor** ( VendorID, vendorName, vendorPhone\_no, vendorEmail, Vendorwebsite)

Product ( ProductID, ProductName, Category, Description, Cost, Sale\_Price, Quantity, Weight, length, width, height)

**Ret\_order** ( RetOrderID, OrderStatus, RetOrderPlacedDate, Description)

Retailer ( RetailerID, RetailerName, RetailerPhone\_No, RetailerEmail, RetailerWebsite)

**C.order** ( CustomerOrderID, OrderStatus, CustomerOrderPlacedDate, Description)

**Cust** (CustomerID, custName, custPhone\_No, custEmail)

**Location** ( PostalCode, country, city, area, street, plot\_no, flat\_no)

**Employee** ( EmployeeID, Name, Address, Phone\_no, Email, Status, Gender, Designation, EmployeeSalary)

**Shipper** (ShipperID, ShipperName, shipperPhone\_no, ShipperLoadLimit)

End\_Product (End\_ProductID, End\_ProductName, Category, Description, Cost, Sale\_Price, Quantity, Weight, length, width, height)

**Godown** ( GodownID, Capacity(volume), status) //status means is it on rent or owned

**Taxing** (TaxingID, TaxingDate, TaxingPercentage)

**3NF:**

**VenProdRet\_OrdTaxBridge** (VendorID, RetOrderID, ProductID, TaxingID,UnitPrice, remarks, Discount, Quantity, PaymentTerms, subtotal)

**RetRet\_ordBridge** (RetailerID, RetailerOrderID)

**RetProdCust\_ordTaxBridge** (CustomerOrderID, RetailerID, ProductID, TaxingID, UnitPrice, remarks, Discount, Quantity, PaymentTerms, subtotal)

**CustCust\_OrdBridge** (CustomerID, CustomerOrderID)

**LocShipperEmpRet\_ordBridge** ( PostalCode, EmployeeID, RetailOrderID, ShipperID, ReceivedDate, shippingCharges, Total, Balance, Credit)

**LocCustomer\_ordShipBridge** ( PostalCode ,CustomerOrderID, custAddress, ReceivedDate, shippingCharges)

**CustomerOrdProdEnd\_ProdBridge** ( CustomerOrderID, ProductID, End\_ProductID, servicesCharges)

**End\_ProdEmployeeGodownBridge** (EmployeeID, End\_ProductID, GodownID)

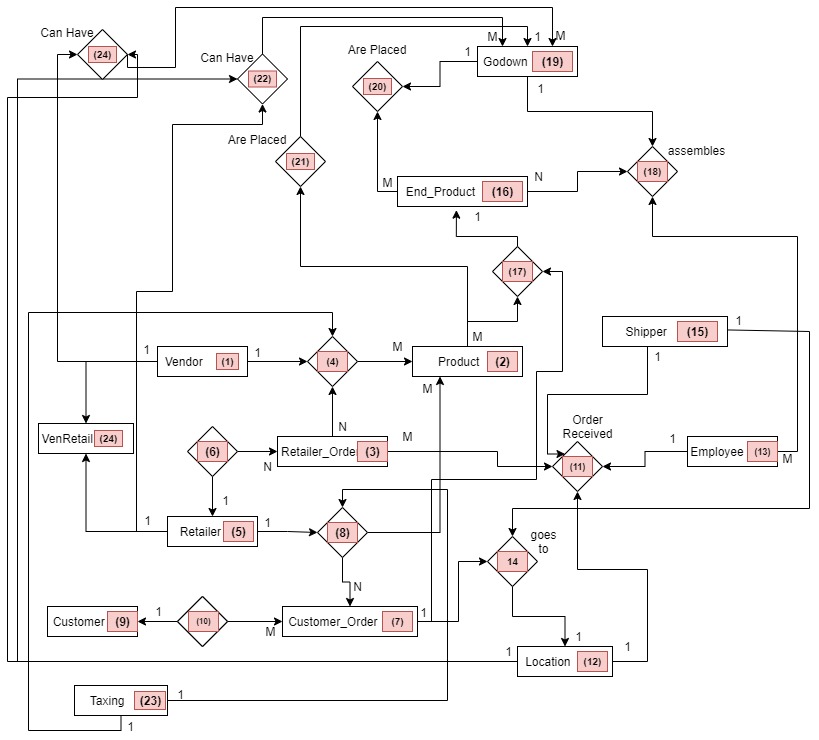
**ProductGodownBridge** (ProductID, GodownID)

**RetailerLocationGodownBridge** (RetailerID, PostalCode, GoDownID)

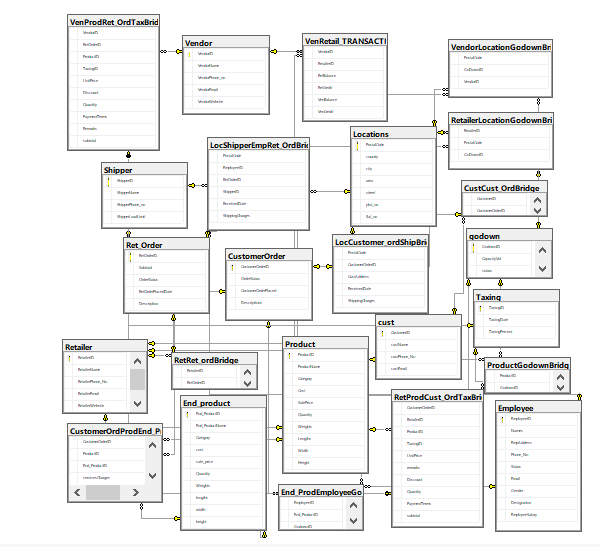
**VendorLocationGodownBridge** (VendorID, PostalCode, GoDownID)

**VenRetail\_TRANSACTION\_Bridge**(VendorID, RetailerID, RetBalance, RetCredit, VenBalance, VenCredit)

# **ERD of the Management System:**



# **Schema of the Management System:**



# **Screenshots of the system with Explanation:**

**Customer:** A customer is a person who purchases from the retailer

Graphical user interface, text

Description automatically generated

**Customer Order:** This table has the order details of a customer.

Graphical user interface, text

Description automatically generated

**Locations:** The location table stores all the location details of the respective orders

Graphical user interface, text

Description automatically generated

**Employee:** Employee is the person who works in a godown and collets order for the retailer from a shipper

Graphical user interface, text

Description automatically generated

**Godown:** Godown is the storage area where products and end products are stored.

Graphical user interface, text

Description automatically generated

**End\_Product:** This table has finished products made from products that a customer orders

Graphical user interface, text

Description automatically generated

**Product:** This table has the basic parts which can be purchased by customers or vendors.

Graphical user interface, text

Description automatically generated with medium confidence

**Retailer Order:** Retailer order is the order which a retailer places to vendor

Graphical user interface, text

Description automatically generated

**Retailer:** A retailer is basically a whole seller who will be further selling products to the customer

Graphical user interface, text

Description automatically generated

**Shipper:** A shipper is the shipping company employee that delivers the order from vendor to retailer.

Graphical user interface, text

Description automatically generated

**Taxing:** Taxing is the table that tells the percentage tax was in a specific year.

Graphical user interface, text

Description automatically generated

**Vendor:** The vendor is the main person who provides the parts in bulk to a retailer.

Graphical user interface, text

Description automatically generated

**CustCust\_OrdBridge:** This will work as a bridge table between customer and customer order bridge **.**

**Graphical user interface, text, application

Description automatically generated**

**CustomerOrdProdEnd\_ProdBridge:** This table will work as a bridge table between customer, order, product, end\_product.

**A screenshot of a computer

Description automatically generated**

**End\_ProdEmployeeGodownBridge:** This table will work as a bridge table between end\_product, employee, and godown.

**Graphical user interface, text, application

Description automatically generated**

**LocCustomer\_ordShipBridge:** This table will have all the information to connect location, customer\_order, shipper.

**A screenshot of a computer

Description automatically generated**

**LocShipperEmpRet\_OrdBridge:** This table will have the keys to connect Location, shipper, employee, retailer\_Order and will work as a bridge table.

**Graphical user interface, application

Description automatically generated**

**ProductGodownBridge:** This table will will connect product, and Godown.

**Graphical user interface, application

Description automatically generated**

**RetailerLocationGodownBridge:** The following table will be a bridge between Retailer, Location, and godown tables.

**Graphical user interface, application

Description automatically generated**

**RetProdCust\_OrdTaxBridge:** This table will form a bridge between Perailer, Product, Customer\_order, and taxing table.

**A screenshot of a computer

Description automatically generated**

**RetRet\_ordBridge:** The following table will connect retailer, and retailer\_order tables.

**Graphical user interface, text, application

Description automatically generated**

**VendorLocationGodownBridge:** The following table will have the keys to connect vendor, locations, and godown table.

**Graphical user interface, text, application

Description automatically generated**

**venProdRet\_OrdTaxBridge:** The following table will be a bridge between Vendor, product, retailer\_order, and taxing.

**A screenshot of a computer

Description automatically generated**

**VenRetail\_TRANSACTION\_Bridge:** This table will have all the information to connect vendor, retailer tables.

**Graphical user interface, text, application

Description automatically generated**

Chapter 3: Future Work and Conclusion

# **Problems Faced:**

* One of the biggest concerns and problems faced while creating this project was optimizing the d management system database in such a way that it remains flexible for not a specific client but as an optimal management system that can be used as a service for many clients.
* Another problem faced was where we had to test out the scenarios based on our ERD diagram, we had to keep in mind many hazards and cases which can create a problem with no solution and for that, we did create solutions.
* We have no front-end functionality due to which operating with a database will be difficult for the time being until a front-end functionality is not added.

# **Future work:**

We strongly believe this project has great potential in the work field and it can be improved a lot, but due to lack of some knowledge in our project, we are unable to progress further. Firstly, front-end functionality will be added in the future for a user-friendly interface that clients can work with.

Secondly, more features can be added to improve the inventory management system, like adding a tracking system for every order.

# **Conclusion:**

As the database has no flaws in normalization and has fully functional queries, this management system can be applied to real-world problems. The management system can be used in small businesses as well because the queries are not too difficult, but very simple.