# CHAPTER 1-

# INTRODUCTION

The WAREHOUSE SUPPLY CHAIN is a web based system, designed for a production sector, which gives information related to the customers and the product details they ordered. This product develops a system that can be used by the company management to keep track of all the sales, vendors and its customers. It automates the communication between all these 3 sectors.

This project has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by the existing system. Moreover this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is required for the user to use this system. By this it proves its user friendly. Warehouse Supply Chain, as described above can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather than concentrating on record keeping. Thus it will help organization in better utilization of resources.

Every organization whether big or small has challenges to overcome and managing the information of product, customer, orders, delivery, and shipment. Every supply chain has different customer needs; therefore we design exclusive employee management systems that are adaptable to our managerial requirements.

This is designed to assist in strategic planning, and will help us ensure that our organization is equipped with the right level of information and details for our future goals. Also, for those busy executives who are always on the go, our systems come with all remote access features which will allow us to manage our workforce anytime. These systems will ultimately allow us to better manage our resources.

The functionalities provided by the present warehouse supply chain include:

* Provides the searching facilities based on various factors such as customers, Product Company, orders, shipments.
* It manages the delivery details for order details and shipment details.
* Manages the information of the products.
* Shows the details of the customers, vendors and transporters.
* Keeps track of the available quantities.
* Supports any number of product addition.
* Can support the access of multiple vendors and customers at one instant.
* All the update and delete operations are recorded for audit purposes.

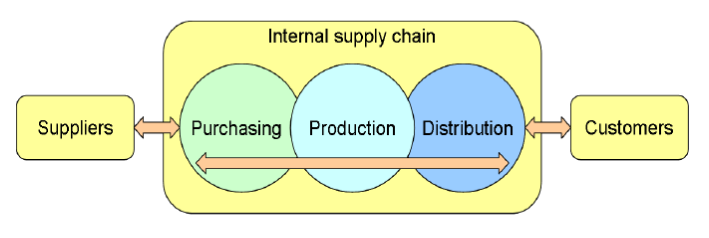


Figure 1.1- Pictorial view of the Supply chain

# CHAPTER 2-

# PROBLEM STATEMENT

A small consignment shop is in need of a system to manage the inventory of their store. They have an extremely limited budget, and do not want to purchase an expensive software suite in such a risky economic environment. In a single software package they will need the ability to handle in-store transactions, mange inventory, create reports, and create an online presence.

Purpose: To create an affordable inventory management system for a small, family-operated consignment shop as an alternative to similar high dollar systems on the market today. To stay competitive, an online storefront will also be implemented to reach a wider customer base and maximize product turnover. The system will be implemented using Python, PHP, and MySQL.

# OBJECTIVES

The main objective of the project is to find out the supply chain management system. Supply chain is a lengthy process. Basically their supply chain is totally ware-house based. In supply chain, at first they store the entire product from different manufacturer in their warehouse. The transportation cost is totally taken care by the warehouse company. Then the stored products are distributed through their own vehicles to its several outlets. There is no intermediary between warehouse and manufacturing company. No product directly comes to the outlet without the hands of warehouse.

The other objectives include:

* To gather practical knowledge of supply chain management.
* Reduce manpower and manual paper works for maintaining the records offline.
* Maintain the accuracy, integrity and consistency of the data.
* To maintain all the products, customers, employees, vendors and orders details in one database.

Manage end to end supply chain management from one portal

# DATABASE ASSUMPTIONS

A warehouse basically has 4 important elements: products, transporters, orders and suppliers. Each element must be tracked based on its location and quantity. The product on hand is updated by tracking incoming shipments and outgoing orders. Order alerts can be set to trigger when inventory levels fall below custom-defined minimum levels.

* A warehouse has goods that are either supplied by some supplier or can be delivered to the customers.
* Every good has a unique article id, description of the good, denomination of quantity (DOQ), cost per piece and total cost (cost per piece\*quantity).
* One good can belong to maximum two transactions (one of the supplier to the warehouse and other one to the customer from warehouse) and one transaction can involve one or many goods.
* Transaction has details like order number, consignee name, date, quantity and transaction value.
* Goods are sent to warehouse by one or many suppliers and a supplier or indent can send one or many goods to the warehouse.
* Every indent has a unique supplier id, name, DOQ, quantity, cost per piece and amount.
* A customer has a unique customer id, mobile no., address, DOQ, quantity and amount.
* Goods can be sent to customers from the warehouse with its transaction information (ternary).
* A vehicle has a unique vehicle no., driver name, date of registration, date of incorporation, insurance and expiry date.
* A transporter is the one providing vehicle for the warehouse, who has a unique transporter id, name and age
* A transporter can have one or many vehicles but each vehicle belongs to only one transporter.

# CHAPTER 3-

# FEASIBILITY STUDY

Once the system objectives have been ascertained by initial investigation, we need to spell the various possible solutions to meet the various objectives. The feasibility study is conducted to check whether the candidate system is feasible. The system which is selected to be the best against the criteria is thereafter designed and developed. The feasibility study takes into consideration. The risks involved in the project development beforehand. Feasibility study includes seven distinct but inters related type of feasibility. All these feasibility study used are used.

Focus is on establishing whether the technology needed for the proposed system is available and how this technology can be integrated within the organization. Technologies include are:

• Hardware

• Software

• Application developed environment

# ECONOMIC FEASIBILITY-

It is concerned with the returns or benefits of the organization are likely to derive from investment in the system. Estimated costs of new system development and operation must be balanced against projected tangible as well as intangible benefits.

# OPERATIONAL FEASIBILITY-

It is an evaluation to determine whether a system is operationally acceptable. Two impotent dimensions to be accessed are ability and motivation to use the system.

# MANAGERIAL FEASIBILITY-

It determines whether a proposed system will be acceptable to the people or not. It also determines from the management.

# CHAPTER 4-

# SYSTEM ANALYSIS

* The system runs on apache server. So it’s needed that the server must have an apache server version 2.0 available.
* MySQL database has been used for storing the data of the website.
* HTML has been used for creating the layout of the web application.
* CSS has been used for creating and designing of the webpage.

# HARDWARE INTERFACE

1. Processor - Intel Core i5
2. Speed - 1.8 GHz
3. RAM - 256 MB
4. Hard Disk - 10 GB

# SOFTWARE INTERFACE

1. Operating System - Windows 7
2. Database used - MySQL
3. Scripting Language - Javascript

# CHAPTER 5-

# DATABASE DESIGN

# RELATIONAL MODEL

Relational data model is the primary data model, which is used widely around the world for data storage and processing. This model is simple and it has all the properties and capabilities required to process data with storage efficiency.

In the relational model, all data must be stored in relations (tables), and each relation consists of rows and columns. Each relation must have a header and body. The header is simply the list of columns in the relation. The body is the set of data that actually populates the relation, organized into rows. You can extrapolate that the junction of one column and one row will result in a unique value - this value is called a tuple.

The second major characteristic of the relational model is the usage of keys. These are specially designated columns within a relation, used to order data or relate data to other relations. One of the most important keys is the primary key, which is used to uniquely identify each row of data. To make querying for data easier, most relational databases go further and physically order the data by the primary key. Foreign keys relate data in one relation to the primary key of another relation. Besides defining how the data are to be structured as discussed above, the relational model also lays down a set of rules to enforce data integrity, known as integrity constraints. It also defines how the data are to be manipulated (relational calculus). In addition, the model defines a special feature termed normalization to ensure efficient data storage.

1. **ENTITY**

An entity is an object that has its existence in the real world. It includes all those “things” about which data is collected. An entity may be a tangible object such as a student, a place or a part. It may also be non-tangible such as an event, a job title or a customer account. For example, if we say that a customer buys goods, it means customer and goods are entities. Diagrammatically, entities are represented in rectangles.

1. **AN ENTITY SET**

It is a set of entities of the same type that share the same properties, or attributes. The set of all persons who are customers at a given bank, example, can be defined as the entity set customer.

1. **ATTRIBUTES**

Attributes are units that describe the characteristics or properties of entities. In a database, entities are represented by tables and attributes by columns. For example, a customer entity might have numerous attributes such as code, name and addresses. Similarly, the goods entity may have attributes like code and price. They drawn in elliptical shapes along with the entity rectangles.

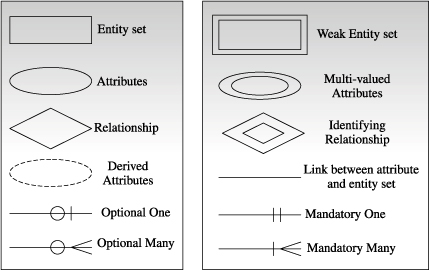


Figure 5.1.1- Database Design

# ENTITY RELATIONSHIP DIAGRAM

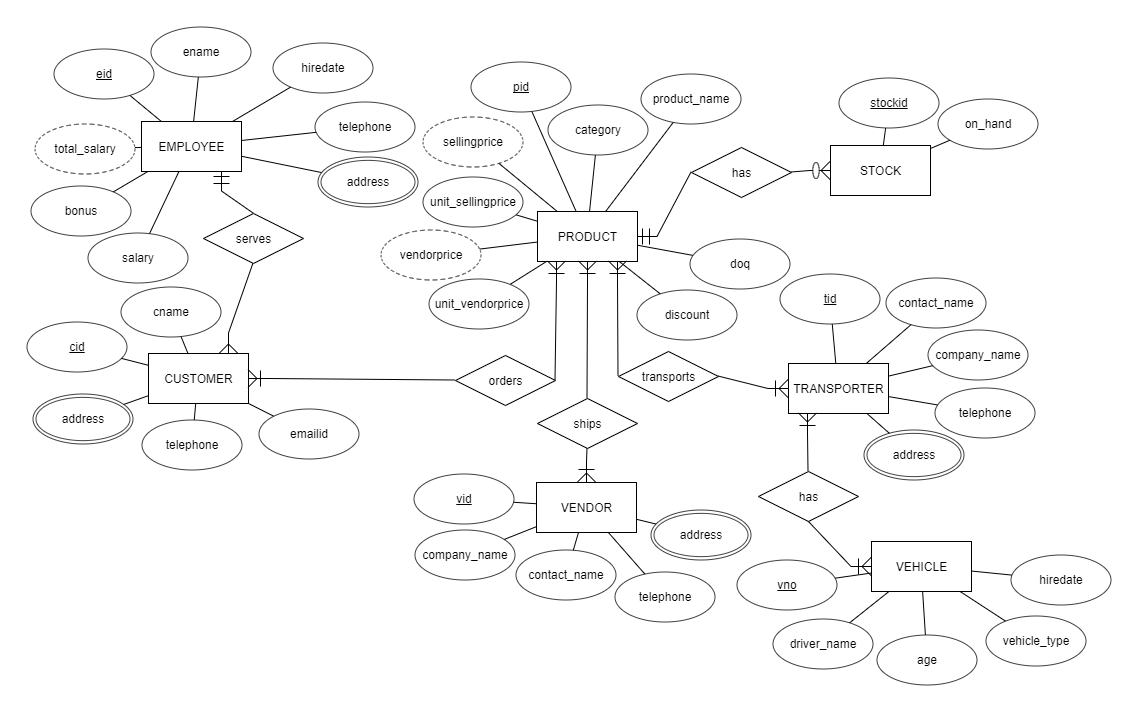
Entity Relationship Diagrams (ERDs) illustrate the logical structure of databases. It is a piece of data-an object or concept about which data is stored.

Figure 5.2.1- Entity Relationship Diagram

# SCHEMA

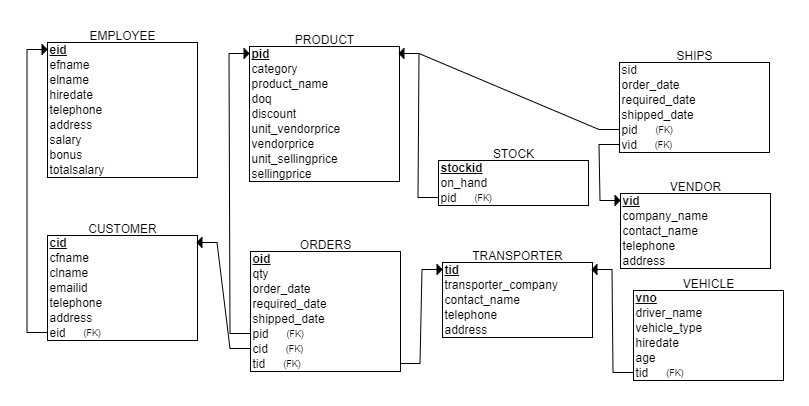
A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams.

Figure 5.3.1- Schema

# CHAPTER 6-

# IMPLEMENTATION

The modules of this application include:

* CUSTOMER- Used for storing the customer information.
* PRODUCT- Used for storing the product details along with its stock.
* STOCK- Stores the number of products available in hand.
* ORDERS- Used for ordering the products along with the customer id who ordered.
* VENDOR- Used for storing the information of the sellers who are providing the products to the warehouse.
* TRANSPORTER- Stores the transporter information.
* VEHICLE- Stores the vehicle and the driver information who are entitled to deliver the product to the customer.

# TABLE CREATIONS

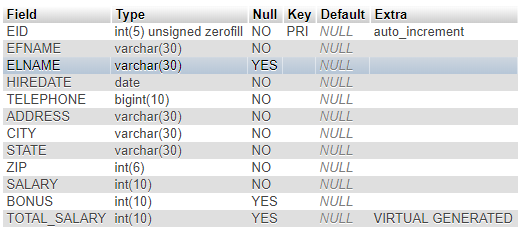


Figure 6.1.1- Employee table description

1. **EMPLOYEE TABLE**
2. **CUSTOMER TABLE**

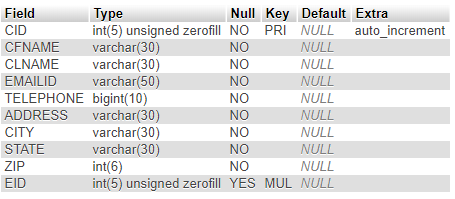
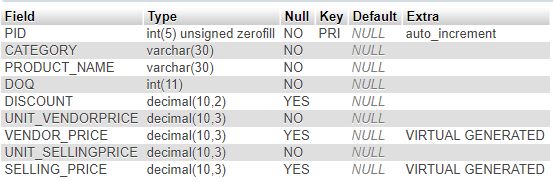


Figure 6.1.2- Customer table description

1. **PRODUCT TABLE**



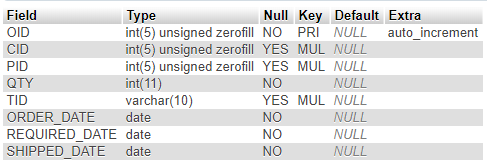
Figure 6.1.3- Product table description

Figure 6.1.4- Orders Table description

1. **ORDERS TABLE**
2. **VENDOR TABLE**

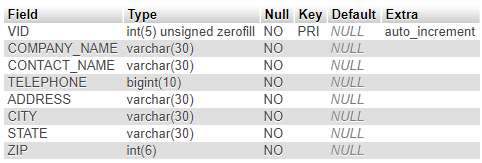


Figure 6.1.5- Vendor table description

1. **TRANSPORTER TABLE**

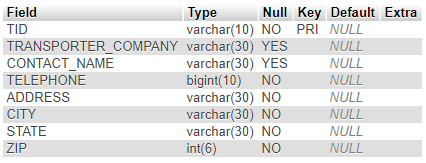


Figure 6.1.6- Transporter table description

1. **VEHICLE TABLE**

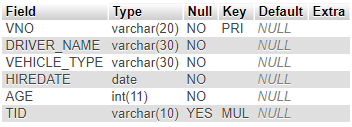


Figure 6.1.7- Vehicle table description

# DDL COMMANDS

1. **Add unique constraint to the product name**

mysql> ALTER TABLE PRODUCT ADD CONSTRAINT UQ UNIQUE(PRODUCT\_NAME);

Query OK, 0 rows affected (0.58 sec)

Records: 0 Duplicates: 0 Warnings: 0

1. **Add a new column age to the vehicle table**

mysql> ALTER TABLE VEHICLE ADD AGE INT(11);

Query OK, 0 rows affected (0.45 sec)

Records: 0 Duplicates: 0 Warnings: 0

1. **Set default value for salary as 10,000 in Employee table**

mysql> ALTER TABLE EMPLOYEE MODIFY SALARY INT(10) DEFAULT 10000;

Query OK, 0 rows affected (0.26 sec)

Records: 0 Duplicates: 0 Warnings: 0

1. **Set NOT NULL constraint for address in Transporter table**

mysql> ALTER TABLE TRANSPORTER MODIFY ADDRESS VARCHAR(30) NOT NULL;

Query OK, 0 rows affected (1.57 sec)

Records: 0 Duplicates: 0 Warnings: 0

# AGGREGATIONS

1. **Display the average bonus salary given to an employee.**

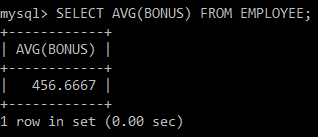
****

Figure 6.3.1- Avg() function

1. **Display the minimum and maximum cost of a product based on each of its category.**

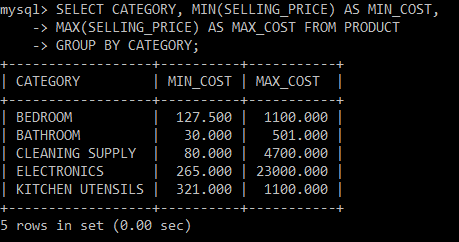
****

Figure 6.3.2- Min() and Max() functions

1. **Display the total salary that needs to be allocated to the Employee department.**

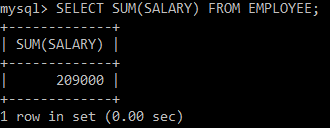
****

Figure 6.3.3- Sum() function

# JOINS

1. **Display the customer along with the employee who is serving him/her.**

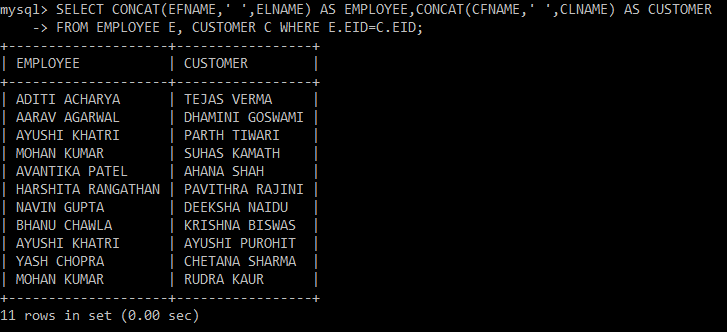


Figure 6.4.1- Join query 1

1. **Display the driver details along with the transporter company he is working for.**

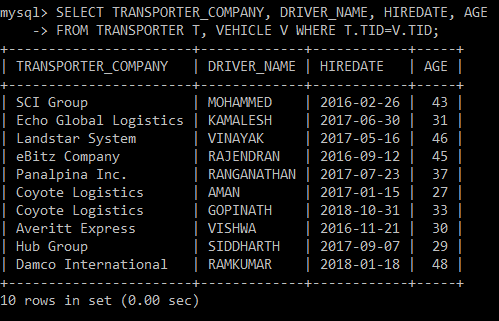


Figure 6.4.2- Join query 2

# NESTED AND CORRELATED QUERIES

1. **Display the employee names who are not active members in the warehouse.**

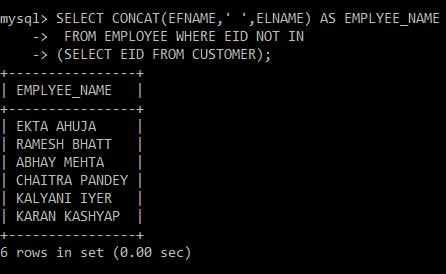


Figure 6.5.1- Nested query

1. **Display the product names along with their cost that are sold until now.**

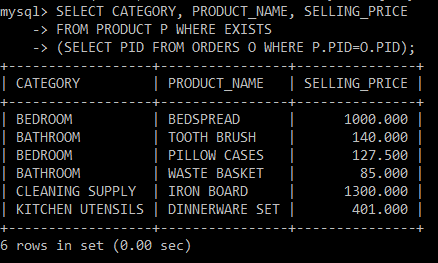


Figure 6.5.2- Correlated query

# VIEWS

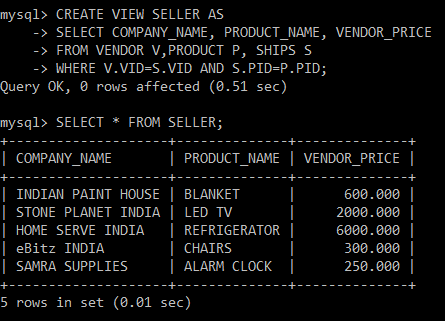
1. **Create a view to store the product name and its cost along with the company who is supplying it.**

Figure 6.6.1- View SELLER

1. **Create a view to store the product name along with the customer who bought it.**

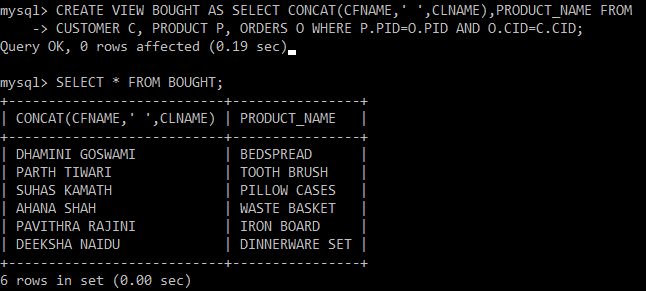


Figure 6.6.2- View BOUGHT

# TRIGGERS

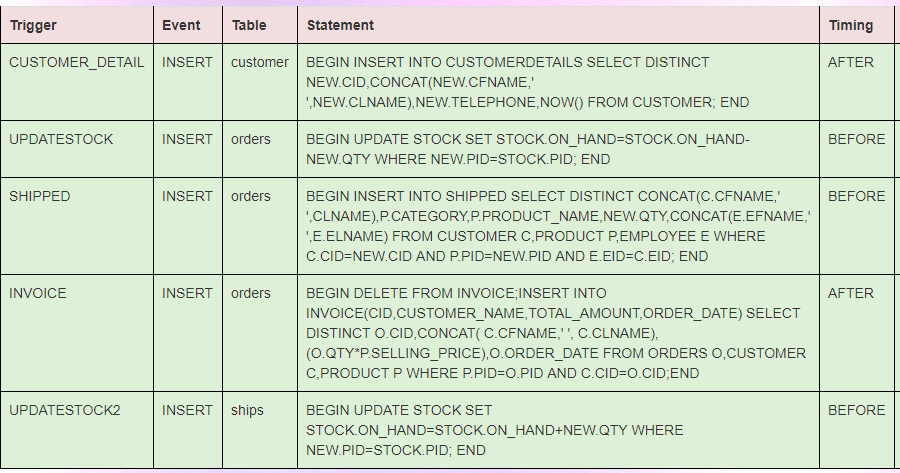


Figure 6.7.1- Show triggers

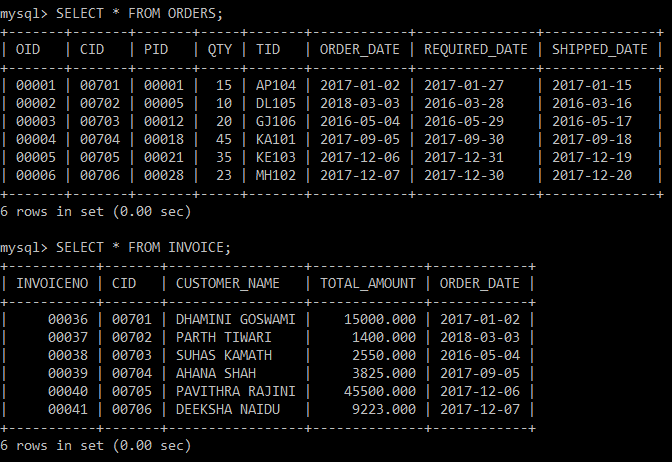
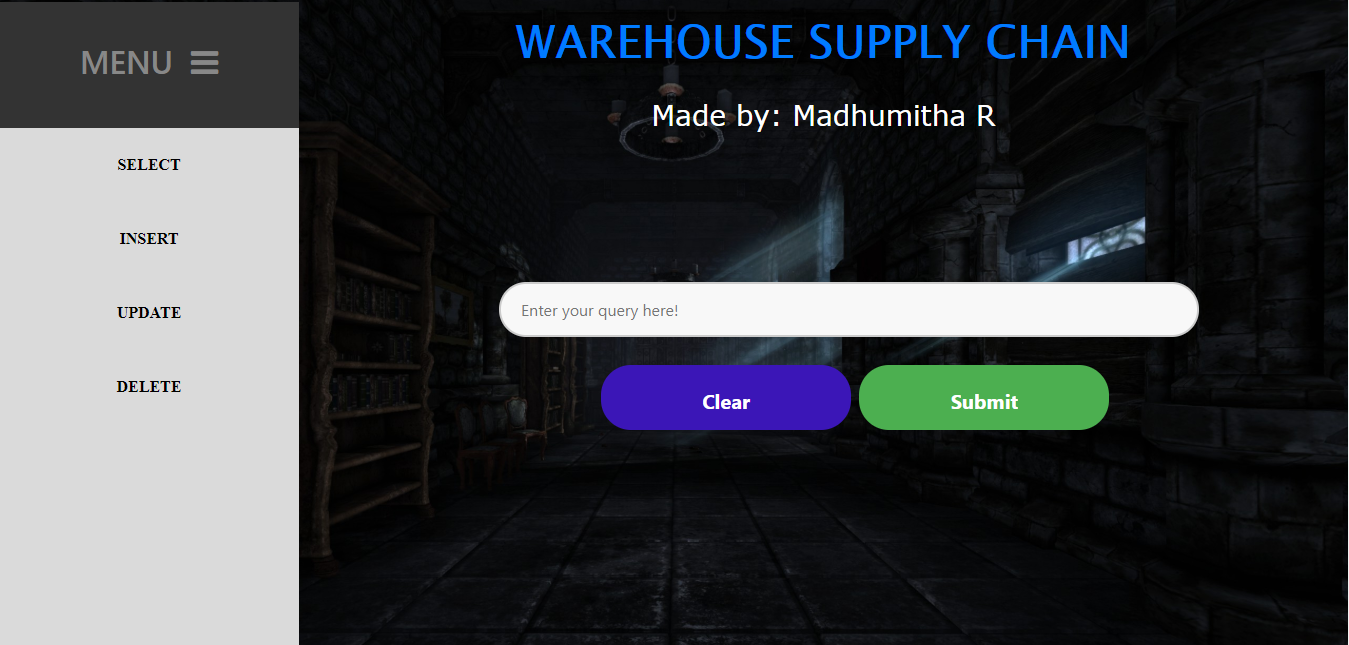


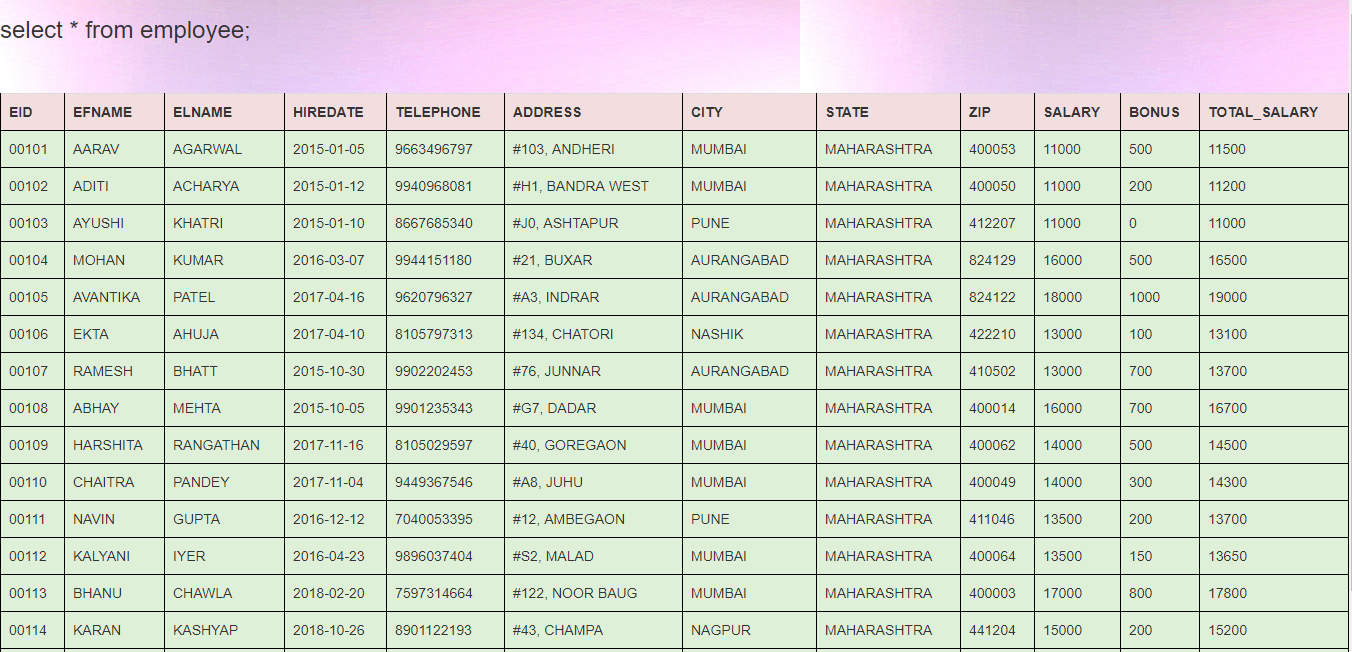
Figure 6.7.2- Effect of Invoice due to Orders

# CHAPTER 7-

# SNAPSHOTS

1. **Front end to enter the queries**



Figure 7.1- Front end

1. **Displaying the table**

Figure 7.2-Table display

1. **Insert page**

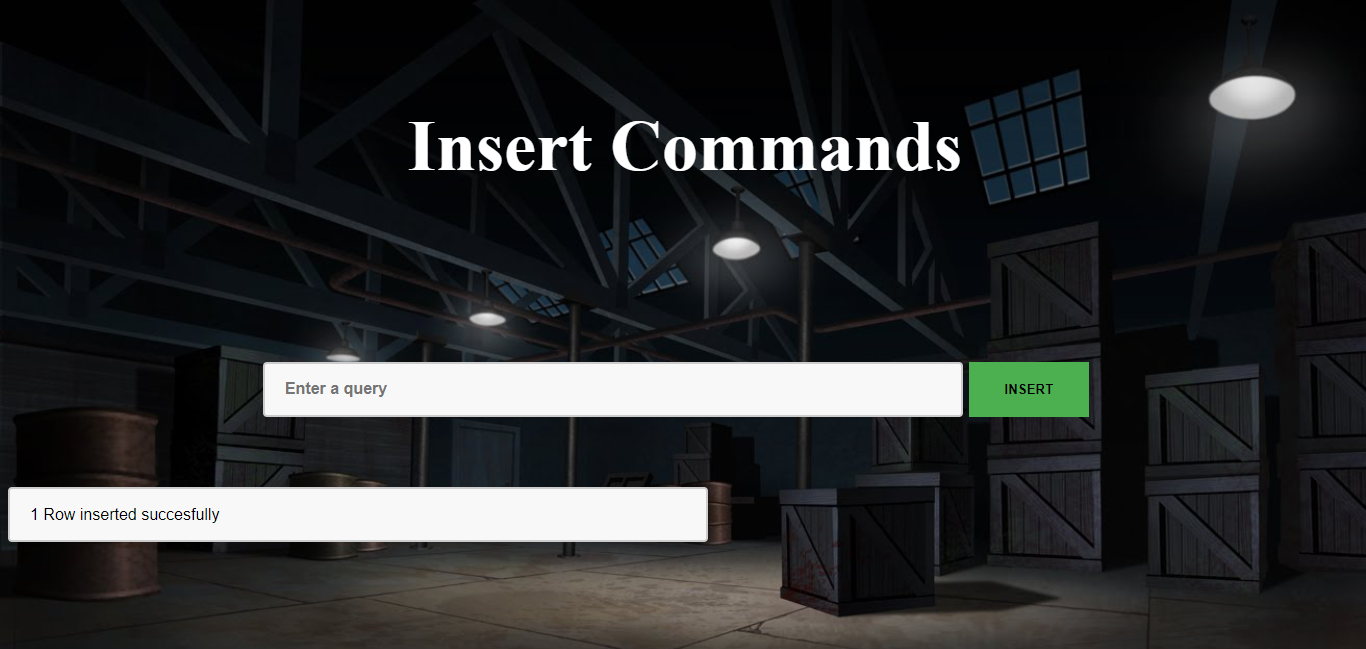


Figure 7.3- Insert page

1. **Update page**



Figure 7.4- Update page

# CHAPTER 8-

# CONCLUSION

* This project has more scope in the future and can be integrated further.
* This project is successfully implemented with all the features mentioned earlier.
* Deployment of this application will help the user to reduce the unnecessary wastage of time in going and ordering the product manually.
* Therefore, we are successfully able o reach the goals and target of the project.

# FUTURE SCOPE

In a nutshell it can be summarized that the future scope of the project circles around maintaining information regarding:

* We can add a printer.
* We can give for advance front end facilities.
* We can host the platform on online servers to make it accessible worldwide.
* Create a master and a slave database structure to reduce the overload of the database queries.
* Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers.

# CHAPTER 9-

# BIBLIOGRAPHY

* <https://en.wikipedia.org/wiki/Supply_chain>
* <https://scm.ncsu.edu/scm-articles/article/what-is-supply-chain-management-scm>
* <https://www.investopedia.com/terms/s/supplychain.asp>
* <https://whatis.techtarget.com/definition/supply-chain>
* <https://erdplus.com>
* <https://www.geeksforgeeks.org/relational-model/>
* <https://www.tutorialspoint.com/dbms/dbms_data_schemas.htm>