CHAPTER 1

INTRODUCTION

A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques. A database management system refers to the technology for creating and managing databases. DBMS is a software tool to organize (create, retrieve, update and manage) data in a database. The main aim of a DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient. Computer scientists may classify database management systems according to the database models that they support. Relational databases became dominant in the 1980`s. These model data as rows and columns in a series of tables, and the vast majority use SQL for writing and querying data. In the 2000`s, non-relational databases became popular, referred to as no SQL because they use different query languages.

Characteristics of Database Management System

- Stores any kind of data: A database management system should be able to store any kind of data. It should not be restricted to the employee name, salary, and address. Any kind of data that exists in the real world can be stored in DBMS because we need to work with all kinds of data that is present around us.
- Support ACID properties: Any DBMS is able to support ACID (Accuracy, Completeness, Isolation, and Durability) properties.
- Represents complex relationship between data: Data stored in a database is connected with each other and a relationship is made in between data. DBMS should be able to represent the complex relationship between data to make the efficient and accurate use of data.
- Backup and recovery: There are many chances of failure of whole database. At that time no one will be able to get the database back and for sure company will be in a big loss. The only solution is to take backup of database and whenever it is needed, it can be stored back.

- Structures and described data: A database should not contains only the data but also all the structures and definitions of the data. These descriptions include the structure, types and format of data and relationship between them.
- Data integrity: This is one of the most important characteristics of database management system. Integrity ensures the quality and reliability of database system. It protects the unauthorized access of database and makes it more secure.
- Concurrent use of database: There are many chances that many users will be accessing the data at the same time. They may require altering the database system concurrently. At that time, DBMS supports them to concurrently use the database without any problem.

Architecture of Database Management System

- Internal level: It has an internal schema which describes physical storage structure of database. The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database.
- Conceptual level: It has an conceptual schema which has implementation details (relation) of database. The conceptual schema hides the details of physical storage structures. Usually, a representational data model is used to describe the conceptual schema when a database system is implemented.
- External or View level: It includes external schema or user views which describes the part of the database that a particular user group is interested and hides rest of the details. Each external schema is typically implemented using a representational data model, possibly based on an external schema design in a high-level conceptual data model.

DBMS Languages

A DBMS has appropriate languages and interfaces to express database queries and updates.

- Database languages can be used to read, store and update the data in the database.

Types of Database Languages

1. Data Definition Language
2. Data Manipulation Language

Data Definition Language

- DDL stands for Data Definition language. It is used to define Database structure.
- It is used to create Schema, tables, constraints etc.., in the database.
- Using the DDL statements, you can create the skeleton of the database.
- Data definition language is used to store the information of metadata like the number of tables, their names, indexes, columns in each table, constraints etc.
- Here are some commands that come under DDL:

i. CREATE: It is used to create objects in the database.
Syntax: CREATE TABLE <TABLE_NAME>;

ii. ALTER: It is used to alter the structure of the database.
Syntax: ALTER TABLE TABLE_NAME ADD(COLUMN_NAME1 DATATYPE1 DEFAULT SOME VALUE);

iii. TRUNCATE: It is used to remove all records from a table.
Syntax: TRUNCATE TABLE TABLE_NAME;

iv. RENAME: It is used to rename an object.
Syntax: RENAME TABLE OLD_TABLE_NAME TO NEW_TABLE_NAME;

v. DROP: It is used to delete objects from the database.
Syntax: DROP TABLE TABLE_NAME;

Data Manipulation Languages

- DML stands for data manipulation languages. It is used for accessing and manipulating data in a database. It handles user requests.
- Here are some commands that come under DML:

I. SELECT: It is used to retrieve data from a database.
Syntax: SELECT *FROM TABLE_NAME;

II. INSERT: It is used to insert data into a table.
Syntax: INSERT INTO TABLE_NAME VALUES (DATA1, DATA2…);

III. UPDATE: It is used to update existing data within a table.
Syntax: UPDATE TABLE_NAME SET COLUMN_NAME1=VALUE, COLUMN_NAME2= VALUE WHERE CONDITION;

ABOUT OUR PROJECT

Metro management system is the system where all the aspects related to the Proper management of Metro is done. These aspects involve managing Information about the various products, staff, managers, customers, billing etc. This system provides an efficient

way of managing the Metro information. Also allows the customer to purchase and pay for the items purchased.

This project is based on the sales transaction and billing of items in a Metro. The first activity is based on adding the items to the system along with the rate which are present in the Metro and the name of the items which the Metro will agree to sell. This authority is given only to admin (administrator). Any modifications to be done in the item name and the rate can be done

only by admin. He also has the right to delete any item. As the customer buys the products and comes to the billing counter, the user is supposed to enter the item name he purchased and the quantity of the item he had purchased. This is not a huge a task.

The system will display all the items whose name starts with the letter selected by the user. He can select out of those displayed. Finally a separate bill will be generated for each customer. This will be saved in the database. Any periodic records can be viewed at any time. If the stock is not available, the Metro orders and buys from a prescribed vendor. The amount will be paid by deducting the total amount acquired in the sales activity. Admin provides a unique username and password for each through which he can login.

1.1 Requirements Analysis

This phase entails gathering of requirements from users of the system. The requirements are collected in a requirements specification document. The project meets the expectation of the admin by allowing them to add products whatever the customer wants. The project also allows them to delete the products when it is unavailable. Each manager is given with unique login id and password. Only manager has the permission to check the details of customers, products, employees, and billing.

- **Employee:** When a new employee joins the company, his record is saved in the database.
- **Customers:** When a customer buys a product, his details have to be saved in the database.
- **Products:** Here the Admin can add any new items present in the supermarket. He also has the right to modify or delete it from the database.
- **Manager:** Can only view the items added by the admin into the management.
- **Login :** As soon as the employee joins the company, the admin provides unique username and password to him.
- **Product Entry:** The items bought from the vendor will be entered here and this will be added to the stock.
- **Sales Analysis:** This provides the report of the items sold for a particular month/ year and also gives the total amount acquired.

- **Billing:** This generates the total price the customers have to pay.
- **Display:** A user can view information regarding Items present in the supermarket.
- **Logout:** This module allows the user to Logout the application. Further operations cannot be performed after user exits.

The schema for keeping records of all the details of the metro wholesale management can be taken from the details given below

- The database needs to keep track of each Customer's details (name, gender, city, age) and his respective ids and display them.
- The database should display details about the employee (unique id, name, city, date of birth, salary, sex, department etc.) who works in the amusement park in his respective department.
- The database should store the basic information about different departments like groceries, fashion, sports and so on
- The system must also be allowed to update the database in case of new entries of employees or products or customers are added or deleted from the management.

CHAPTER 2

DESIGN

Design is the creation of a plan or convention for the construction of an object, system or measurable human interaction. Designing often necessitates considering the aesthetic, functional, economic and sociopolitical dimensions of both the design object and design process. Thus "design" may be a substantive referring to a categorical abstraction of a created thing or things (the design of something), or a verb for the process of creation. It is an act of creativity and innovation.

- **Conceptual Design:**

  The purpose of the conceptual design phase is to build a conceptual model based upon the previously identified requirements, but closer to the final physical model. A commonly-used conceptual model is called an entity-relationship model.Once all the requirements have been collected and analyzed, the next step is to create a conceptual schema for the database, using a high level conceptual data model. This phase is called conceptual design.

- **Logical Design:**

  Logical database design is the process of deciding how to arrange the attributes of the entities in a given business environment into database structures, such as the tables of a relational database. The goal of logical database design is to create well-structured tables. The tables will be able to store data about the entities in a non-redundant manner and foreign keys will be placed in the tables so that all the relationships among the entities will be supported.

2.1 Entity Relationship Diagram :

An entity–relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

Description of ER-Diagram

This ER-Diagram for this project has 10 entities:

1) Products

2) Departments
3) Employees
4) Customers
5) Billing
6) Admin

Sales Analysis **Relations between the various Entities are:**

1) ADMIN **adds** PRODUCTS. The relation between them is 1:M, one admin can add any number of products.
2) ADMIN **adds** EMPLOYEES. The relation between them is 1:M, one admin can add any number of employee.
3) ADMIN **adds** CUSTOMERS. The relation between them is 1:M, one admin can add any number of customer.
4) CUSTOMERS **buys** PRODUCTS, Here the relation between them is N:M.
5) EMPLOYEE **servres** CUSTOMERS to shop. Here the relation between them is N:M.
6) PRODUCTS **undergoes** BILLING for the customers. Here the relation between them is 1:1, a bill can be paid by only one counter.
7) BILLS are **analysed by** SALES ANALYSIS. Here the relation is M:N

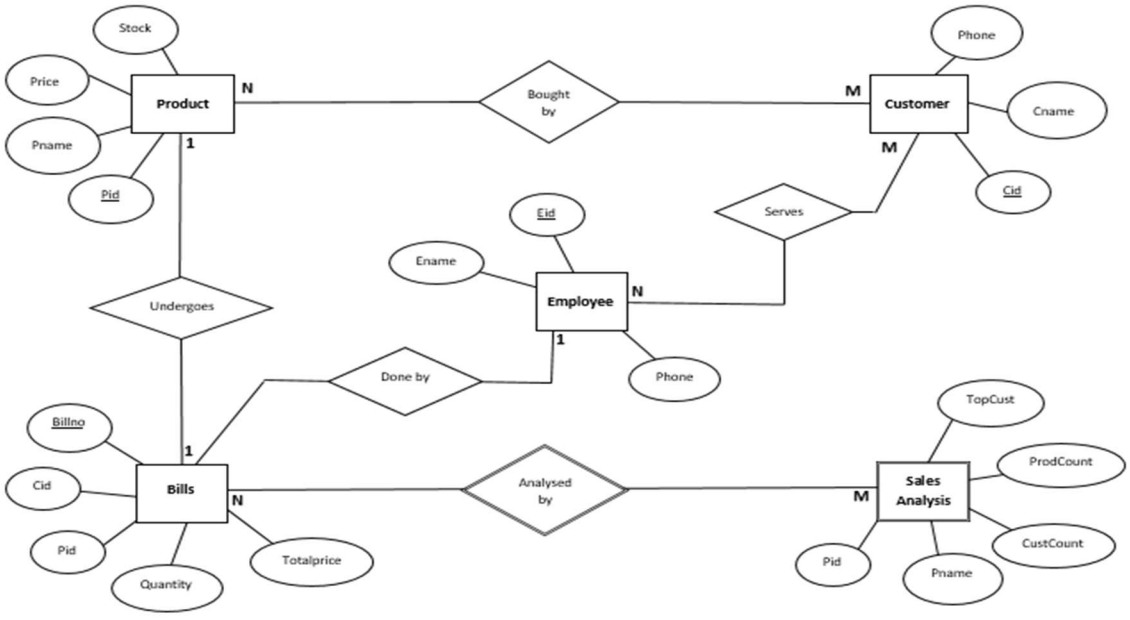

FIG.2.1 - ER Diagram of Metro Wholesale System

2.2 ER To Relational Mapping:

ER-to-Relational Mapping Algorithm:

Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relation Types

Step 4: Mapping of Binary 1:N Relationship Types.

Step 5: Mapping of Binary M:N Relationship Types.

Step 6: Mapping of Multivalued attributes.

Step 7: Mapping of N-ary Relationship Types.

Step 1: Mapping of Regular Entity Types.

1.1: For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E.

1.2 : Choose one of the key attributes of E as the primary key for R.

1.3 : If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of R.

Step 2: Mapping of Weak Entity Types

2.1: For each weak entity type W in the ER schema with owner entity type E, create a relation R & include all simple attributes (or simple components of composite attributes) of W as attributes of R.

2.2: Also, include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s).

2.3: The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any.

Step 3: Mapping of Binary 1:1 Relation Types

For each binary 1:1 relationship type R in the ER schema, identify the

Relations S and T that correspond to the entity types participating in R.

There are three possible approaches:

3.1: Foreign Key approach: Choose one of the relations-say S-and include a foreign key in S the primary key of T. It is better to choose an entity type with total participation in R in the role of S.

3.2: Merged relation option: An alternate mapping of a 1:1 relationship type is possible by merging the two entity types and the relationship into a single relation. This may be appropriate when both participations are total.

3.3: Cross-reference or relationship relation option: The third alternative is to set up a third relation R for the purpose of cross-referencing the primary keys of the two relations S and T representing the entity types.

Step 4: Mapping of Binary 1:N Relationship Types.

4.1: For each regular binary 1:N relationship type R, identify the relation S that . represent the participating entity type at the N-side of the relationship type

4.2: Include as foreign key in S the primary key of the relation T that represents the . other entity type participating in R.

4.3: Include any simple attributes of the 1:N relation type as attributes of S.

Step 5: Mapping of Binary M:N Relationship Types.

5.1: For each regular binary M:N relationship type R, create a new relation S to represent R.

5.2: Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S.

5.3: Also include any simple attributes of the M : N relationship type (or Simple components of composite attributes) as attributes of S.

Step 6: Mapping of Multivalued attributes.

6.1: For each multivalued attribute A, create a new relation R.

6.2: This relation R will include an attribute corresponding to A, plus the primary key attribute K-as a foreign key in R-of the relation that represents the entity type of relationship type that has A as an attribute.

6.3: The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

Step 7: Mapping of N-array Relationship Types.

7.1: For each n-array relationship type R, where n>2, create a new relationship S to represent R.

7.2: Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.

7.3: Also include any simple attributes of the n-array relationship type (or simple components of composite attributes) as attributes of S.

2.3 Relational Model:

A relational database schema is the tables, columns and relationships that make up a relational database. A relational database schema helps you to organize and understand the structure of a database. This is particularly useful when designing a new database, modifying an existing database to support more functionality, or building integration between databases. The Relational model contains all ent0ities and their relations with other entities. It also contains all relations having (m: n) cardinality.

In databases, relational schema may refer to

- database schema, in the relational paradigm
- (single) relation schema

Database schema. ... The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). The formal definition of a database schema is a set of formulas (sentences) called integrity constraints imposed on a database.

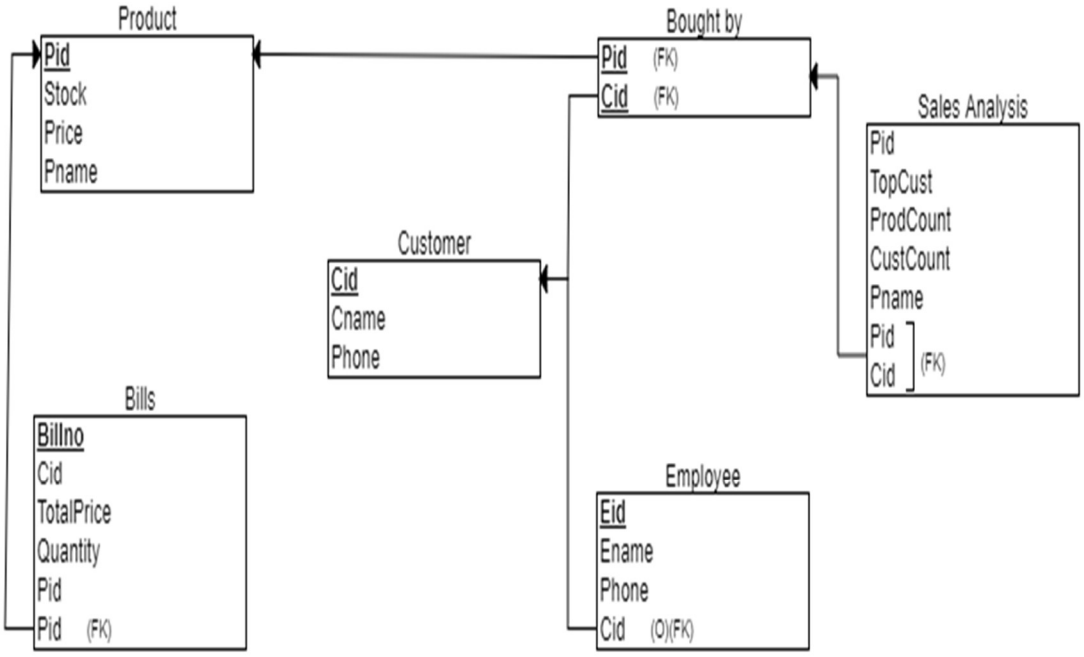

FIG.2.3 – Relational Model of Metro Wholesale System

2.4 Relational Database Schema

The database schema of a database system is its structure described in a formal language supported by the database management system (DBMS). The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database

tables in the case of relational databases). The formal definition of a database schema is a set of formulas (sentences) called integrity constraints imposed on a database.[citation needed] These integrity constraints ensure compatibility between parts of the schema.

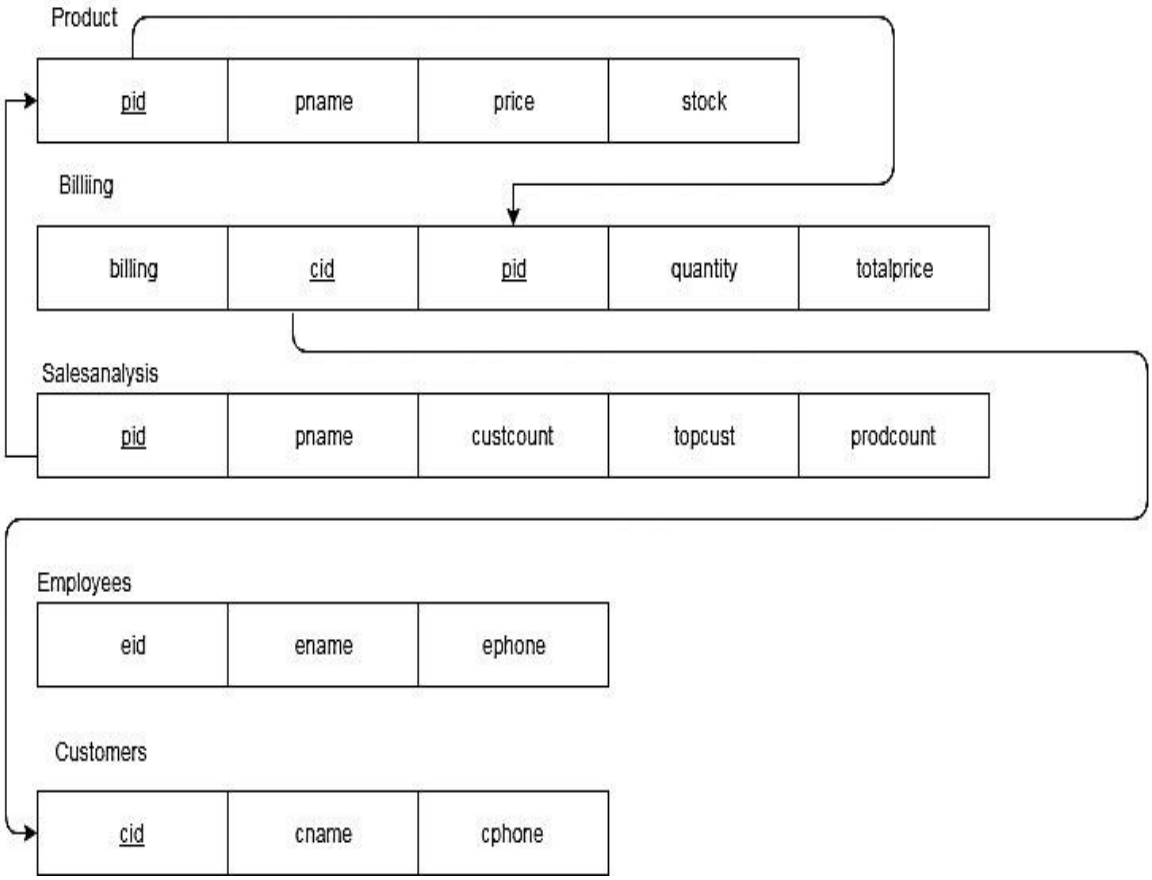

FIG.2.4 – Schema Diagram of Metro Wholesale System

2.5 Normalized Relational Schema

Normalization is a database design technique which organizes tables in a manner that reduces redundancy and dependency of data. It divides larger tables into smaller tables and links them using relationships. Theory of data Normalization in SQL is still being developed further. There are discussions on the 6th Normal Form. However, in most practical applications, normalization achieves its best in the 3rd NF.

The database is said to be in 1 NF if and only if

- Each table cell contains a single value
- Each record needs to be unique.

The database in said to be in 2 NF if and only if:

- It is in 1 NF
- No non-prime attribute functionally depends on a prime attribute (i.e. a part of the candidate key)

The database in said to be in 3 NF if and only if:

- It is in 1 NF and 2 NF
- Has no transitive functional dependencies

➢ **To prove that they are in 1NF:**

1NF is a property of a relation in a relational database. A relation is in first normal form if and only if the domain of each attribute contains only atomic (indivisible) values, and the value of each attribute contains only a single value from that domain.

First normal form is an essential property of a relation in a relational database. Database normalization is the process of representing a database in terms of relations in standard normal forms, where first normal is a minimal requirement.

First normal form enforces these criteria:

- Eliminate repeating groups in individual tables.
- Create a separate table for each set of related data.
- Identify each set of related data with a primary key

Products: pid, pcost, pdept, pname, pqty, poffer. Thus the relation is in 1NF.

Customers: cid, cname, cmob, cdept. Thus the relation is in 1NF.

Department: drptname, deptid. Thus the relation is in 1NF.

Employees: eid, ename, emob. Thus the relation is in 1NF.

Sales Analysis: pid. Pname, cost. Thus the relation is in 1NF.

Billing: cid, pid, soldqty, cost, total, billno, status. Thus the relation is in 1NF.

KEY ATTRIBUTES:

1) Products: pid
2) Customers: cid
3) Employees: eid
4) Billing: bill no
5) Department: dname
6) Sales analysis: sid

- **To prove that the tables are in 2NF:**

1. The customer table is in 1NF and based on the functional dependencies defined above, there is no prime attribute that determines a non-prime attribute. Hence the table is in 2NF.
2. The employee table is in 1NF and based on the functional dependencies defined above, there is no prime attribute that determines a non-prime attribute. Hence the table is in 2NF.
3. The product table is in 1NF and based on the functional dependencies defined above, there is no prime attribute that determines a non-prime attribute. Hence the table is in 2NF.
4. The sales analysis table is in 1NF and based on the functional dependencies defined above, there is no prime attribute that determines a non prime attribute. Hence the table is in 2NF.
5. The billing table is in 1NF and based on the functional dependencies defined above, there is no prime attribute that determines a non prime attribute. Hence the table is in 2NF.
6. The department table is in 1NF and based on the functional dependencies defined above, there is no prime attribute that determines a non prime attribute. Hence the table is in 2NF.

Therefore all the tables are in 2NF.

- **To prove that they are in 3NF**

The third normal form (3NF) is a normal form used in database normalization. Codd's definition states that a table is in 3NF if and only if both of the following conditions hold:

- The relation R (table) is in second normal form (2NF)
- Every non-prime attribute of R is non-transitively dependent on every key of R.

A non-prime attribute of R is an attribute that does not belong to any candidate key of R. A transitive dependency is a functional dependency in which $X \rightarrow Z$ (X determines Z) indirectly, by virtue of $X \rightarrow Y$ and $Y \rightarrow Z$ (where it is not the case that $Y \rightarrow X$).

A 3NF definition that is equivalent to C odd's, but expressed differently. This definition states that a table is in 3NF if and only if, for each of its functional dependencies $X \rightarrow A$, at least one of the following conditions holds:

- X contains A (that is, $X \rightarrow A$ is trivial functional dependency), or
- X is a super key, or
- Every element of A-X, the set difference between A and X, is a prime attribute (i.e., each attribute in A-X is contained in some candidate key)

1. The customer table is in both 1NF and 2NF. In this table there are no transitive dependencies i.e a non-prime attribute does not functionally determine another non-prime attribute. Hence according to the definition, the table is in 3NF.
2. The product table is in both 1NF and 2NF. In this table there are no transitive dependencies i.e a non-prime attribute does not functionally determine another non-prime attribute. Hence according to the definition, the table is in 3NF.
3. The employee table is in both 1NF and 2NF. In this table there are no transitive dependencies i.e a non-prime attribute does not functionally determine another non-prime attribute. Hence according to the definition, the table is in 3NF.
4. The sales analysis table is in both 1NF and 2NF. In this table there are no transitive dependencies i.e a non-prime attribute does not functionally determine another non-prime attribute. Hence according to the definition, the table is in 3NF.
5. The billing table is in both 1NF and 2NF. In this table there are no transitive dependencies i.e a non-prime attribute does not functionally determine another non-prime attribute. Hence according to the definition, the table is in 3NF.
6. The department table is in both 1NF and 2NF. In this table there are no transitive dependencies i.e a non-prime attribute does not functionally determine another non-prime attribute. Hence according to the definition, the table is in 3NF.

Therefore all the tables are in 3NF.

2.6 Key Attributes

1. PRODUCTS

   Primary Key: Pid

   Foreign Key: dept

2. CUSTOMERS

   Primary Key: cid

3. EMPLOYEES

   Primary Key: eid

   Foreign Key: dept

4. BILLING

   Primary Key: bill no

   Foreign Key: pid, cid

5. DEPARTMENT

   Primary Key: dept name

6. SALES ANALYSIS

   Primary Key: sid

   ForeignKey:pid,pcost

CHAPTER 3

SYSTEM REQUIREMENTS

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time.

3.1 Hardware and Software requirements:

- **Software Requirements:-**

Operating System : Windows 7 and above

Browsers : Chrome,IE

Database : MySQL /PHP/XAMPP

Technology : Servlets, JSP, TCP/IP Protocol suite.

Web Server : XAMPP

Software's : MySQL

Web Technologies : HTML, CSS

- **Hardware Requirements:-**

Processor : Intel i3 or i5, dual core,64 bit.

RAM : 4GB

Hard Disk : Minimum 1GB

CD Drive

- **User Interface:-**

Front end :- HTML

Back end :- PHP

- **Design Tools :-**

ERD Plus (Online diagram software for making ER Diagrams)

Draw.io (Online diagram software for making schema Diagrams)

3.2 Tools Used:

1) MySQL – MySQL is an open-source relational database management system. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a central component of the LAMP open-source web application software stack. LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, and Drupal.

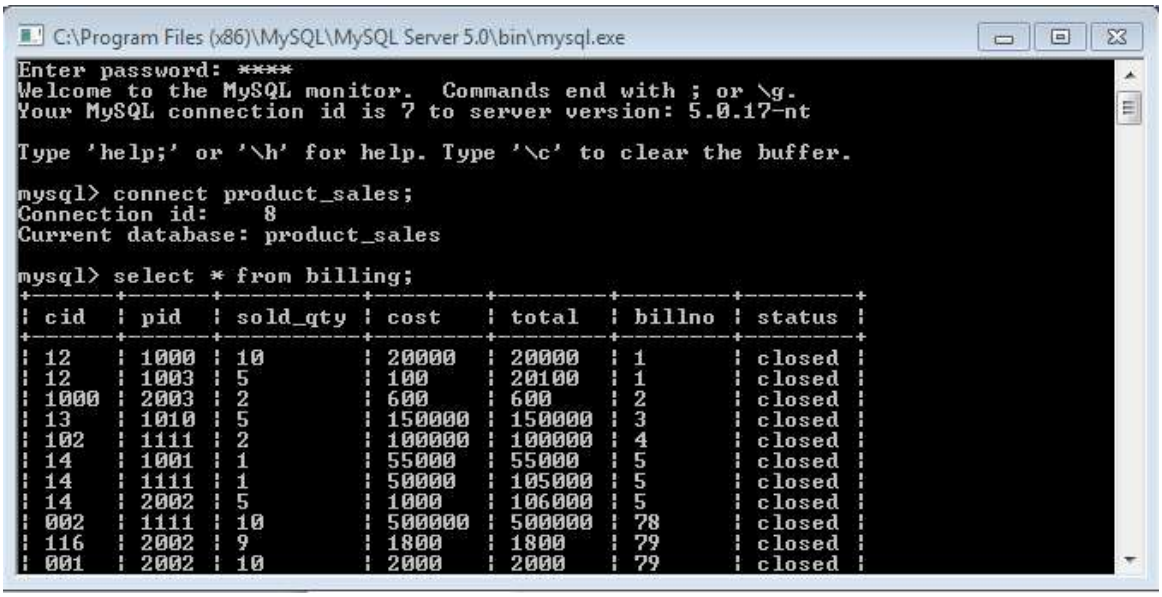

Fig.3.1 – MySQL command prompt.

2) XAMPP - XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, Maria DB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

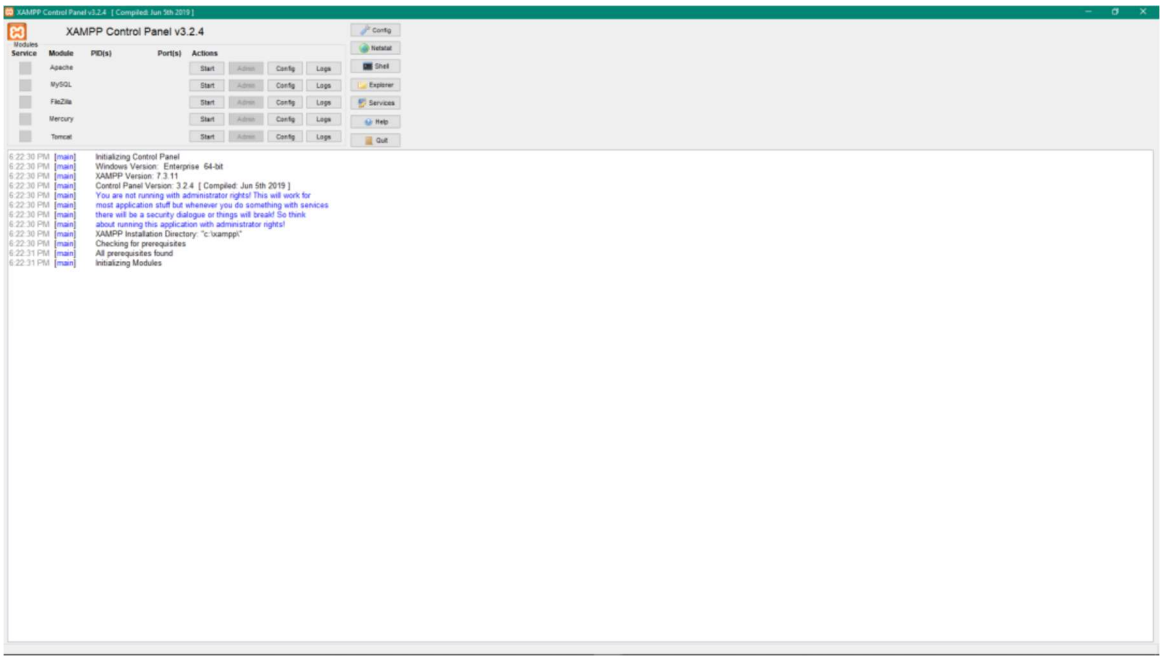

Fig.3.2 – XAMPP SERVER VERSION 3.2.4

3) **VISUAL STUDIO CODE –** Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It provides basic support for HTML, CSS, PHP, Java Script ,etc programming out of the box. It is opensource software.

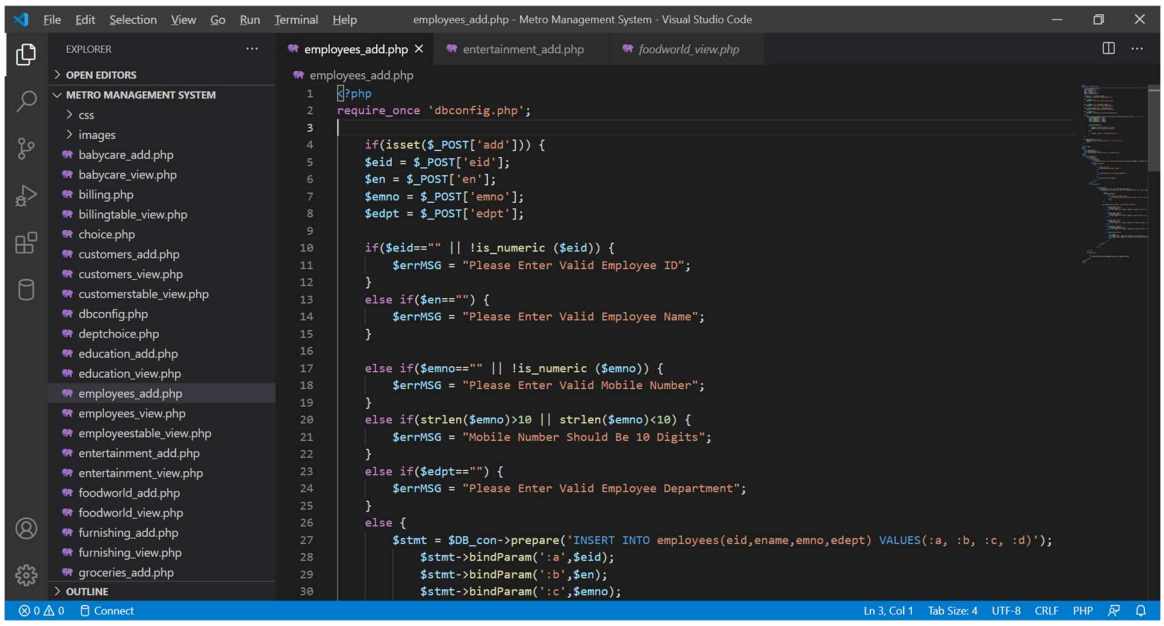

Fig 3.3 – Visual Studio Code Editor

4) Web Browser - A web browser is a software application for retrieving, presenting and traversing information resources on the World Wide Web. An information resource is identified by a Uniform Resource Identifier (URI/URL) that may be a web page, image, video or other piece of content. Hyperlinks present in resources enable users easily to navigate their browsers to related resources. Although browsers are primarily intended to use the World Wide Web, they can also be used to access information provided by web servers in private networks or files in file systems. The most popular web browsers are Chrome, Safari, Opera and Firefox.

5) ERD PLUS and DRAW.IO –ERD Plus is a web-based database modelling tool that lets you quickly and easily create

Entity Relationship Diagrams (ERDs)

Relational Schemas (Relational Diagrams)

Star Schemas (Dimensional Models)

Schema Diagrams.

CHAPTER 4

IMPLEMENTATION

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy.

- **Back end Implementation in MySQL:**

- **Create table queries:**

```
1) CREATE TABLE `billing` (
  `cid` varchar(500),
  `pid` varchar(500),
  `sold_qty` varchar(500),
  `cost` varchar(500),
  `total` varchar(500),
  `billno` varchar(500),
  `status` varchar(500));
2) CREATE TABLE `customers` (
  `cid` varchar(500),
  `cname` varchar(500),
  `mno` varchar(200));
3) CREATE TABLE `employees` (
  `eid` varchar(500),
  `ename` varchar(500),
  `emno` varchar(200),
  `edept` varchar(200));
4) CREATE TABLE `dept` (
  `dept` varchar(200));
5) CREATE TABLE `products` (
  `pid` varchar(500),
  `pname` varchar(500),
  `pcost` varchar(500),
```

```
`pqty` varchar(500) ,
`poffer` varchar(200),
`pdept` varchar(500));
```

6) CREATE TABLE `sales` (

```
`cid` varchar(200),
`pcost` varchar(200),
`pqty` varchar(200));
```

- **Insert tables queries:**

1) insert into products (pid,pname,pcost,pqty) values(:a, :b, :c, :d) ;

```
$stmt->bindParam(':a',$pid);
$stmt->bindParam(':b',$pn);
$stmt->bindParam(':c',$pc);
$stmt->bindParam(':d',$pq);
```

2) Insert into billing(cid,pid,sold_qty,cost,total,billno,status) VALUES(:a, :b, :c, :d, :e, :f, :g);

```
$stmt->bindParam(':a',$cid);
$stmt->bindParam(':b',$pid);
$stmt->bindParam(':c',$pq);
$stmt->bindParam(':d',$pc);
$stmt->bindParam(':e',$main_total);
$stmt->bindParam(':f',$bill_no);
$stmt->bindParam(':g',$sta);
```

3) insert into employees(eid,ename,emno) VALUES(:a, :b, :c);

```
$stmt->bindParam(':a',$eid);
$stmt->bindParam(':b',$en);
$stmt->bindParam(':c',$emno);
```

4) insert into customers(cid,cname,mno) VALUES(:a, :b, :c)'

$stmt->bindParam(':a',$cid);

$stmt->bindParam(':b',$cn);

$stmt->bindParam(':c',$mno);

5) insert into department(deptid,deptname) VALUES(:a, :b)'

$stmt->bindParam(':a',$deptid);

$stmt->bindParam(':b',$deptname);

- **SQL Queries**

1) SELECT * FROM billing ;
2) SELECT * FROM products ;
3) DELETE FROM products WHERE pid =:pid ;
4) SELECT * FROM products ORDER BY pid DESC ;
5) SELECT * FROM employees ORDER BY eid DESC ;
6) SELECT * FROM customers ORDER BY cid DESC;
7) SELECT * FROM products where pdept="education" ORDER BY pid DESC;
8) SELECT * FROM products where pdept="groceries" ORDER BY pid DESC;
9) SELECT * FROM products where pdept="sport" ORDER BY pid DESC;
10) SELECT * FROM products where pdept="menfashion" ;
11) SELECT * FROM products where pdept="babycare" ORDER BY pid DESC;
12) SELECT * FROM products where pdept="milk" ORDER BY pid DESC;
13) SELECT * FROM products where pdept="health" ORDER BY pid DESC;

- **Describe table**

Field	Type	Null	Key	Default	Extra
cid	varchar(500)	YES		NULL	
pid	varchar(500)	YES		NULL	
sold_qty	varchar(500)	YES		NULL	
cost	varchar(500)	YES		NULL	
total	varchar(500)	YES		NULL	
billno	varchar(500)	YES		NULL	
status	varchar(500)	YES		NULL	

Table.4.1-desc billing

Field	Type	Null	Key	Default	Extra
pid	varchar(500)	YES		NULL	
pname	varchar(500)	YES		NULL	
pcost	varchar(500)	YES		NULL	
pqty	varchar(500)	YES		NULL	
poffer	varchar(200)	YES		NULL	
pdept	varchar(200)	YES		NULL	

Table.4.2-desc product

Field	Type	Null	Key	Default	Extra
eid	varchar(500)	YES		NULL	
ename	varchar(500)	YES		NULL	
emno	varchar(500)	YES		NULL	
edept	varchar(200)	YES		NULL	

Table.4.3-desc employee

Field	Type	Null	Key	Default	Extra
cid	varchar(500)	YES		NULL	
cname	varchar(500)	YES		NULL	
mno	varchar(500)	YES		NULL	
cdept	varchar(200)	YES		NULL	

Table.4.4-desc customers

Field	Type	Null	Key	Default	Extra
dept	varchar(200)	YES		NULL	

Table.4.5-desc department

- **Update statement:**

1) UPDATE billing SET status=:sta WHERE billno=:bno

- **Alter statements:**

2) ALTER table products add pdeptname varchar(200) references dept(deptname);
3) ALTER table employees add edeptname varchar(200) references dept(deptname);

- **Triggers:**

  Delimiter$$

  Create trigger t1

  After update on billing

  For each row

  Begin

  Set new.total=new.sold_qty*new.cost;

  End$$

cid	pid	sold_qty	cost	total	billno	status
12	1000	10	20000	20000	1	closed
12	1003	5	100	20100	1	closed
1000	2003	2	600	600	2	closed
13	1010	5	150000	150000	3	closed
102	1111	2	100000	100000	4	closed
14	1001	1	55000	55000	5	closed
14	1111	1	50000	105000	5	closed
14	2002	5	1000	106000	5	closed

Triggers table

- **Procedures:**

To calculate the total turnover .

CREATE DEFINER=`root`@`localhost`

PROCEDURE ` METROPROC `( )

NOT DETERMINISTIC NO SQL SQL SECURITY DEFINER

SELECT PRODUCT(`pcost`.`pqty`) as total FROM `product `;

```
mysql> call metroproc();
+----------+
| TURNOVER |
+----------+
|  1132650 |
+----------+
1 row in set (0.08 sec)

Query OK, 0 rows affected (0.08 sec)
```

Procedure table

- **Connectivity between SQL and PHP is done using the following code:**

```
<?php
$DB_HOST = 'localhost';
$DB_USER = 'root';
$DB_PASS = ' ';
$DB_NAME = 'product_sales';
try{
$DB_con = new PDO("mysql:host={$DB_HOST};dbname={$DB_NAME}",$DB_USER,$DB_PASS);
$DB_con->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}
catch(PDOException $e){
echo $e->getMessage();
}
?>
```

Front End Implimentation:

On opening the website registered admin can login to his account with unique username and password so that he can see some of the information like products, employee, customers, what the current month's bill management and manager details etc…

Same as admin, manager also having an account where he can view every details like products, employees, customers, billing, and so on.

The pseudo code for various parts of the html page is given below:

1) The **buttons** are implemented using the form tag with the input type as submit.

```
<form action="first_page.php">
```

<input type=”submit” name=”submit” value=”admin” id=”btn”>

</form>

The ‘action’ specifies the page which should be opened when the submit button is clicked. ‘value’ specifies the text to be displayed on the button. ‘id’ specifies a unique value which can be used in the css. Similarly text, radio, checkbox etc. can be added in a form in the ‘type’ field.

1) The **background images** are set in the body tag of the page by specifying its url.

```
<style>
 body { background-image: url("image-name.jpg");
 </style>
```

Here ‘image-name’ is the name of the image.

2) A **text** can be added using <label> or headers ranging from <h1> to <h6>.

```
<p>
<label> TEXT </label>
<h1> TEXT </h1>
</p>
```

Here the paragraph tag <p> must be within <body> tag.

5. To display **data fetched** from the backend in a table form, table row is defined by <tr> tag. Table Header is defined by <th> tag. Table contents are defined by <td> tag. For ex.

```
<tr>
<th> ID </th>  <th> Name </th>
</tr>
<tr>
<td> 2001 </td>  <td> Sandeep</td>
</tr>
```

CHAPTER 5 RESULTS

SNAPSHOTS

1) Home Page :-

This page gives a brief idea of the project and how it can be used, there is a user login button that lets the user sign into the system, so that the data can be modified.

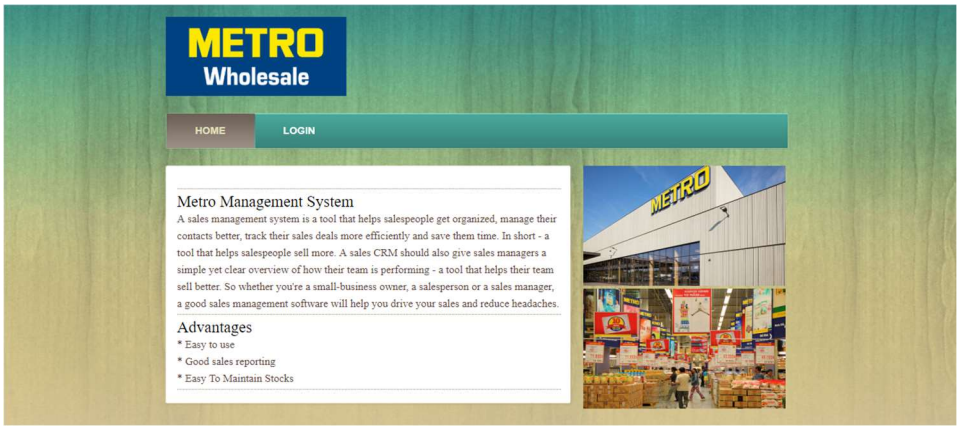

Fig.5.1- Home Page

2) Admin Login Page:

This is a login page for the admin or the manager or for billing.

They have to give their respective username and password.

Admin can add products, new employees and customers to the management.

Whereas manager can only view them.

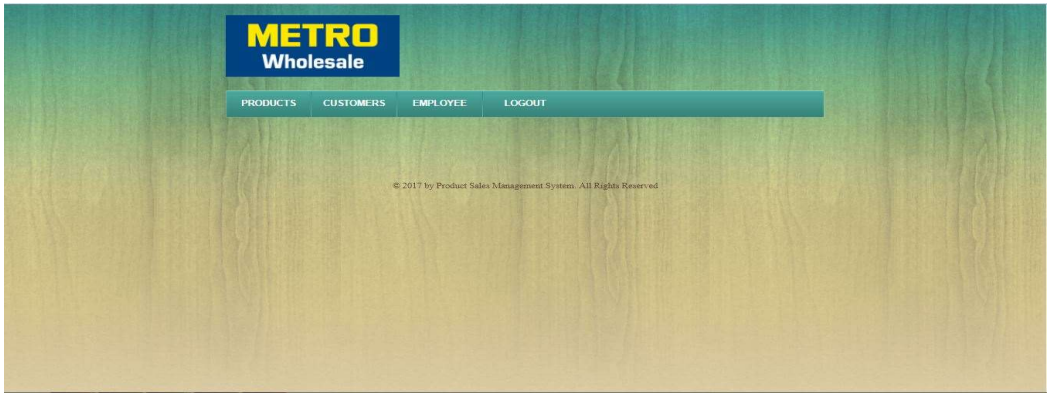

Fig.5.2- Login Page

3) Admin page:

Admin can add, view and also delete new products, employees and customers to the management.

He has to first choose who he is adding in this page i.e. product or employee or customers.

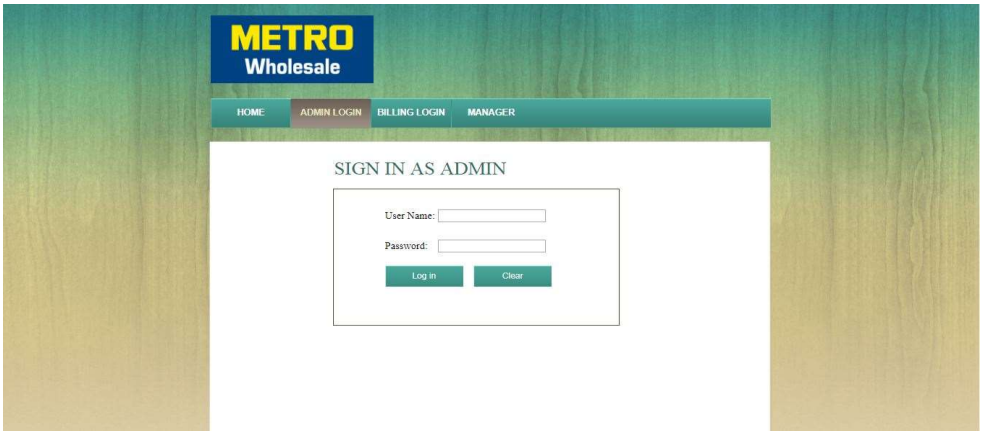

Fig 5.3- Admin Page

4) Department page:

This page is directed when the admin selects ”product” in the previous page. In this page the admin has to select the department of the product which he will add.

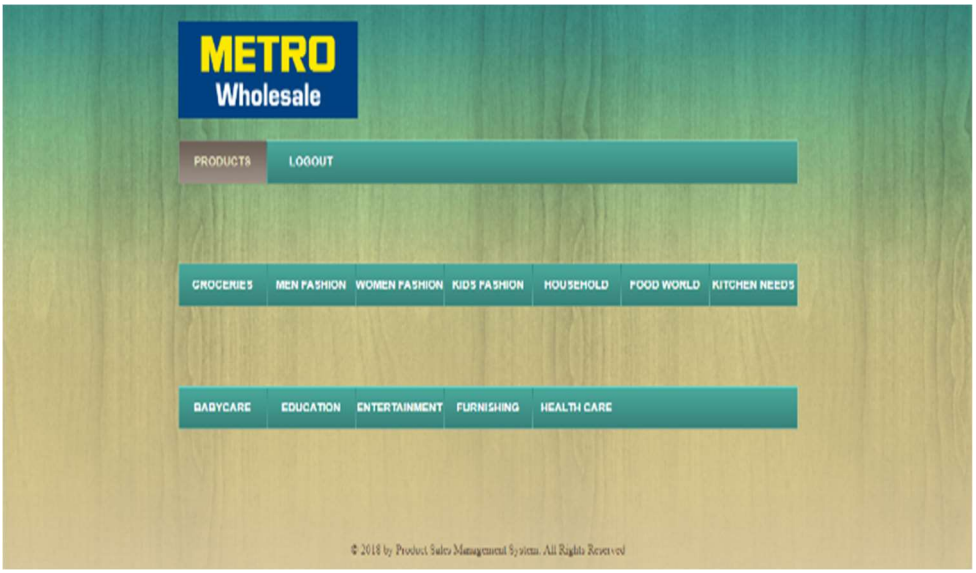

Fig.5.4- Department Page

5) Add product :

The admin has to give proper asked details and add the products into their respective department

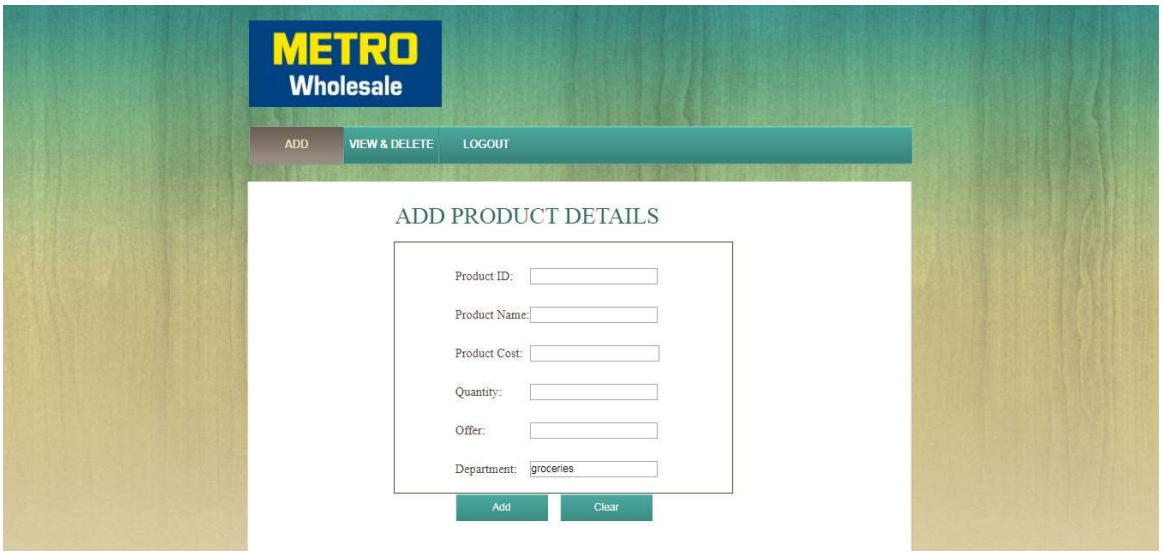

Fig.5.5- Product add Page

6) View and delete items:

Here the admin can either delete or just view the added items.

Fig.5.6 - view and delete products.

7) Billing Page:

The person should login using his username and password.

He has to bill the items bought by the customers.

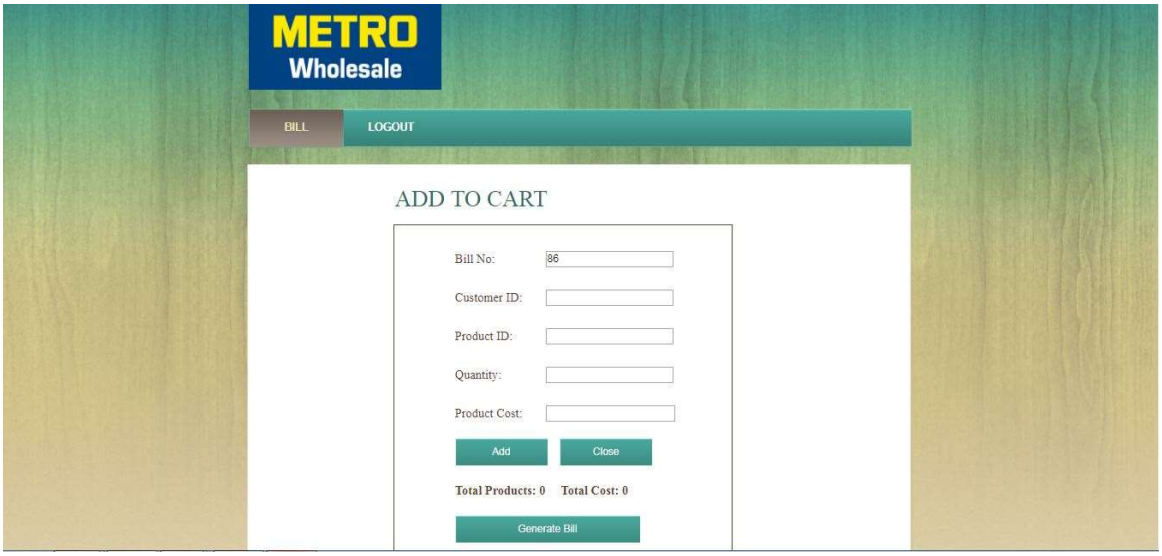

Fig.5.7 – Billing page.

9) View and delete employees:

Here the admin can either delete or just view the added employees.

Fig.5.9 – View and delete employees.

10) Manager product table:

The manager can only view the products, customers and employees added by the admin. products, customers and employees added by the admin.

Fig.5.10– Product view page.

11) Sales Analysis:

The manager can only view the products, customers and employees added by the admin.

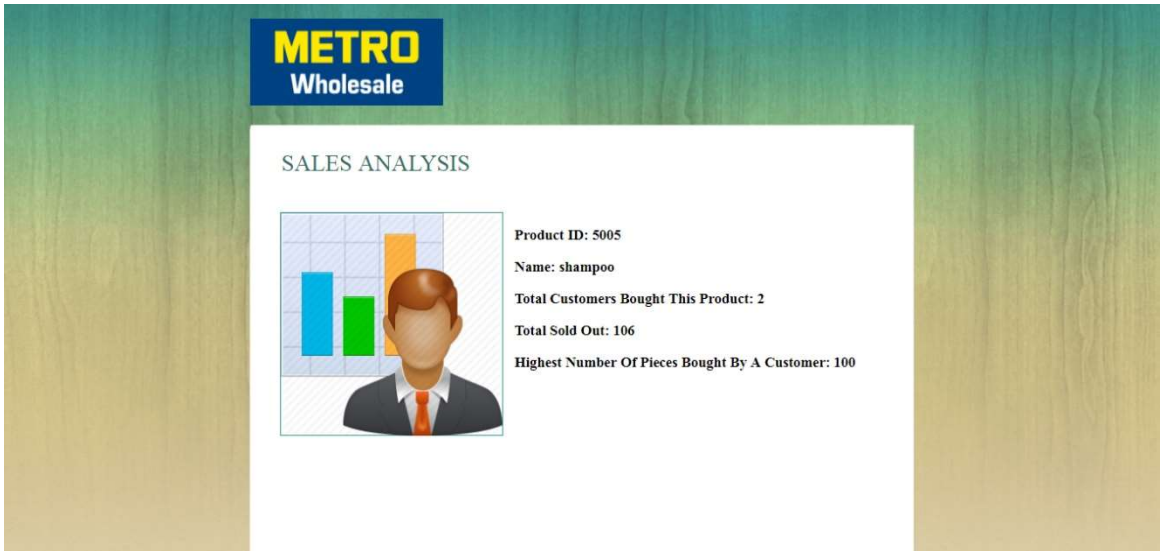

Fig. 5.11 – Sales Analysis.

CHAPTER 6

CONCLUSION

A sales management system is a tool that helps salespeople get organized, manage their contacts better, track their sales deals more efficiently and save them time. In short - a tool that helps salespeople sell more. A sales CRM should also give sales managers a simple yet clear overview of how their team is performing - a tool that helps their team sell better. So whether you're a small-business owner, a salesperson or a sales manager, a good sales management software will help you drive your sales and reduce headaches.

- Advantages

1) Easy to use
2) Good sales reporting
3) Easy To Maintain Stocks

- Limitations:

1) Detailed information and gathering has to be done to obtain satisfactory results.
2) Implementing the software requires change in business practices.
3) Implementation and maintenance costs run very high.
4) The manager must add a new employee and later his login details can't be communicated through this project and must be communicated personally.

REFERENCES

- **Textbooks:**

1) Database systems Models, languages, design and application programming, Ramez Elmasri and Shamkant B, 6th Edition, Pearson.
2) Database management systems, Ramkrishna and Gehrke, 3rd edition, 2014, McGraw Hill.
3) Herbert Schildt: JAVA the Complete Reference, 7th/9th Edition, Tata McGraw Hill 2007.
3) Jim Keogh: J2EE-TheCompleteReference, McGraw Hill, 2007.

- **Websites:**

1) www.w3schools.com
2) Google Chrome.
3) YouTube.
4) www.draw.io
5) www.mysql.com
6) www.apachefriends.org

APPENDIX

SOURCE CODE

Back end Implementation in MySQL:

- **Create table queries:**

```
1) CREATE TABLE `billing` (
  `cid` varchar(500),
  `pid` varchar(500),
  `sold_qty` varchar(500),
  `cost` varchar(500),
  `total` varchar(500),
  `billno` varchar(500),
  `status` varchar(500));
```

- **Connection:**

```
<?php
$DB_HOST = 'localhost';
$DB_USER = 'root';
$DB_PASS = ' ';
$DB_NAME = 'product_sales';
try{
$DB_con=new
PDO("mysql:host={$DB_HOST};dbname={$DB_NAME}",$DB_USER,$DB_PASS);
$DB_con->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}
catch(PDOException $e){
echo $e->getMessage();
}
```

?>

Front End Implementation:

- **INDEX**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Metro Wholesale Management System</title>
<link rel="stylesheet" href="css/style.css" type="text/css">
</head>
<body>
<div id="background">
<div id="page">
<div id="header">
<div id="logo">
<a href="index.html"><img src="images/metrologo.png" alt="LOGO" height="112"
width="250"></a>
</div>
<div id="navigation">
<ul>
<li class="selected">
<a href="index.html"> Home </a>
</li>
<li>
<a href="loginadmin.php"> Login </a>
</li>
</ul>
</div>
</div>
```

```
<div id="contents">
<div id="main">
<div class="box">
<div>
<div>
<h3></h3>
<ul>
<li>
<h4>Metro WareHouse System</h4>
<p>
A sales management system is a tool that helps salespeople get organized, manage their contacts better, track their sales deals more efficiently and save them time.
</p>
</li>
<li>
<h4>Advantages</h4>
</li>
</ul>
</div>
</div>
</div>
</div>
<div id="sidebar">
<div class="section">
<img src="images/metrobuilding.jpg" alt="Img" height="170" width="280">
<img src="images/metroinside.png" alt="Img" height="170" width="280">
</div>
</div>
</div>
</div>
<div id="footer">
```

</body>
</html>