

Online Grocery Store Management System Using SQL

A Case Study Submitted to

**DEPARTMENT
of
COMPUTER SCIENCE AND SYSTEMS ENGINEERING**

Submitted by

**S.Naziya Fathima
P.Vallusha
S.Feeza
K.Bhavani**

**20121A2951
20121A2953
20121A2952
20121A2927**

Under the Guidance of
M. Ramu, M.Tech
Assistant Professor
Dept. of CSSE, SVEC



Department of Computer Science and Systems Engineering
Sree Vidyanikethan Engineering College
(Autonomous)

Sree Sainath Nagar, Tirupati – 517 102
(2021-2022)

DEPARTMENT OF COMPUTER SCIENCE AND SYSTEMS ENGINEERING

VISION

To become a centre of excellence in Computer Sciences and Systems Engineering through teaching, training, research and innovation to create quality engineering professionals who can solve the growing complex problems of the society.

MISSION

- ✓ Established with the cause of development of technical education in advanced computer sciences and engineering with applications to systems there by serving the society and nation.
- ✓ Transfer of Knowledge through contemporary curriculum and fostering faculty and student development.
- ✓ Create keen interest for research and innovation among students and faculty by understanding the needs of the society and industry.
- ✓ Skill development among diversity of students in technical domains and profession for development of systems and processes to meet the demands of the industry and research.
- ✓ Imbibing values and ethics in students for prospective and promising engineering profession and develop a sense of respect for all.

PROGRAM EDUCATIONAL OBJECTIVES

1. Demonstrate competencies in the Computer Science domain and Management with an ability to comprehend, analyze, design and create software systems for pursuing advanced studies in the areas of interest.
2. Evolve as entrepreneurs or be employed by acquiring required skill sets for developing computer systems and solutions in multi-disciplinary areas.
3. Exhibit progression and professional skill development in Computer programming and systems development with ethical attitude through life-long learning.

PROGRAM SPECIFIC OUTCOMES

PSO1: Employ Systems Approach to model the solutions for real life problems, design and develop software systems by applying Modern Tools.

PSO2: Develop solutions using novel algorithms in High Performance Computing and Data Science.

PSO3: Use emerging technologies for providing security and privacy to design, deploy and manage network systems.

PROGRAM OUTCOMES

1. Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

II B. Tech. – II Semester

(20BT40531) DATABASE MANAGEMENT SYSTEMS LAB

COURSE OUTCOMES

CO1. Analyze the requirements of a given database problem and design viable ER-Models for implementation of database.

CO2. Create database schemas, select and apply suitable integrity constraints for querying databases using SQL interface.

CO3. Develop and interpret PL/SQL blocks to centralize database applications for maintainability and reusability.

CO4. Develop database applications for societal applications such as ticket reservation system, employee payroll system using modern tools.

CO5. Work independently and communicate effectively in oral and written forms.



SREE VIDYANIKETHAN ENGINEERING COLLEGE

(AUTONOMOUS)

Sree Sainath Nagar, Tirupati

**DEPARTMENT OF COMPUTER SCIENCE AND SYSTEMS
ENGINEERING**

CERTIFICATE

This is to certify that the case study report entitled

Online Grocery Store Management System Using SQL

is the Bonafide work done by

**S.Naziya Fathima
P.Vallusha
S.Feeza
K.Bhavani**

**20121A2951
20121A2953
20121A2952
20121A2927**

in the Department of **Computer Science and Systems Engineering**, and submitted to Computer Science and Systems Engineering during the academic year 2021-2022. This work has been carried out under my supervision.

Guide:

**M. Ramu, M. Tech
Assistant Professor
Dept. of CSSE**

Head:

**Dr. K. Ramani
Professor & Head
Dept. of CSSE**

INTERNAL EXAMINER

EXTERNAL EXAMINER

TABLE OF CONTENTS

| Title | Page No. |
|--|-----------------|
| ABSTRACT..... | 1 |
| CHAPTER 1. INTRODUCTION | |
| 1.1 Introduction to the topic..... | 2 |
| 1.2 Problem Statement..... | 3 |
| 1.3 Objectives..... | 3 |
| CHAPTER 2. DATABASE DESIGN | |
| 2.1 List of Attributes, entities and relationship..... | 4-7 |
| 2.2 E-R Diagram..... | 8 |
| CHAPTER 3. RELATIONAL MODEL | |
| 3.1 Database languages..... | 9-10 |
| 3.2 Table Description..... | 11-12 |
| 3.3 Relational Database Scheme..... | 13 |
| 3.4 Relational Queries..... | 14-34 |
| CHAPTER 4. CONCLUSION AND FUTUREWORK | |
| 5.1 Conclusion..... | 35 |
| 5.2 Future Work..... | 36 |

ABSTRACT

The online Grocery Store documents a purchaser to submit online commands for items and facilities from a store that distributes both walk-in clients and online customers. The online Store system grants an online display of all the matters they want to wholesale from store. This web grounded application assists customers to select their products. Customers provide their all detail regarding address and contact and they get their chosen products in their home. Shopping days may be check at any time, and their substances can be modified or detached at the option of the customer. Once the customer adopts to submit a buying order, the purchaser may print the insides of the shopping carrier in order to gain a hard copy record of the deal. This Web application saves lots of time of customers and give the more advantages to customer. E-commerce has seen a tremendous growth in the past decade. An important feature of an online grocery system is to arise up with suitable recommendations, which can help the user make quick decisions, so that they dont have to spend additional time, browsing the website. Generating visual needs and reports acts as another plus point, for these websites. In order to implement this, almost all major online shopping sites use Recommendation systems. The main persistence of this broadside is to give an review of a smart cart application which is a predictive model application and which aims to provide germane item recommendations grounded on purchase olden times and user interests based on a dataset. This paper also describes the application that analyses the customers grocery purchasing habits and predicts which previously purchased products will be in a customers next order. The application will also have a reference scheme that predicts the items hence making the users purchasing experience more.

Keywords: E- Commerce, Grocery, Recommendation System, Neural Networks.

1.INTRODUCTION

1.1 Introduction to the topic

The main goal for developing this project where customer can purchase an order on groceries. The structure is very convenient for customer. They can easily buy the grocery products from home through internet. The system decrease a much of work load for customer. The product is directly delivered customer address by system online grocery shopping. The system functionality of products an orders is stored on the admin side in web service. this project provides a lot of feature to manage the product in well manner. This project contains details advance module that can make the backend system very powerful. Online grocery market is a method of E-Commerce that allows customer to buy a product form a seller over internet. There had been a increasing demand for e commerce sites, in the past decades. Online grocers, especially have growing inpopularity. Each of this sites are using recommendation system and algorithms. Internet of Everything or Network of Everything is additionally known as Internet of Things (IoT). When physical articles or things are embedded with physics, sensors and software then the network called IOT is formed. This network has property to change objects to exchange knowledge with the assembly, operator and/or different connected devices supported the infrastructure of International Telecommunication Union's international Standards Initiative. IOT allows the discernment of objects and controlling them remotely across existing network infrastructure. Thus a network makes a extent for a lot of direct integration between the physical world and computer based systems. This brings to betterment in accuracy, potency and economic profit. Every factoris clearly acknowledgeable through its embedded system however is in a position to interoperate among the present internet infrastructure.

1.2 PROBLEM STATEMENT

Online grocers face number of challenges. The major challenge is lack of handy experience in consumer demands. Online market has developed its space in virtual world but is this market worth for all kinds of products specially the perishable grocery products.

1.3 Objectives

The objective of the project is to make an application in android platform to purchase items in an existing shop. In order to build such an application complete web support need to be provided. A complete and efficient web application which can provide the online shopping experience is the basic objective of the project. The web application can be implemented in the form of an android application with web view.

- i. To workshop while in the luxury of your own home, without having to period out of the door.**
- ii. Sell at lesser rate due to a lesser amount of overhead.**
- iii. Provide home delivery free of cost.**

2.DATABASE DESIGN

2.1 List of Attributes,entities and relationship

Entity Name : Categories

| Column Name | Data Type | Remarks |
|---------------|-------------|-------------|
| Category_ID | Integer | Primary Key |
| Category_Name | varchar(30) | Not Null |
| Category_Type | Varchar(30) | Not Null |

Entity Name : Customers

| Column Name | Data Type | Remarks |
|-------------|-------------|-------------|
| Customer_ID | Integer | Primary Key |
| Name | varchar(30) | Not Null |
| Contact_NO | Integer | Not Null |
| Address | varchar(30) | Not Null |
| time_spent | integer | Not Null |

Entity Name : Shopping Order

| Column Name | Data Type | Remarks |
|-------------|--------------|-------------|
| Order_ID | IntegerOrder | Primary Key |
| Customer_ID | Integer | Foreign Key |
| S_Date | Date | Not Null |
| Address | Varchar (30) | Not Null |
| Contact_NO | Integer | Not Null |

Entity Name : Delivery

| Column Name | Data Type | Remarks |
|-------------|-----------|-------------|
| Delivery_ID | Integer | Primary Key |
| Customer_ID | interger | Foreign Key |
| D_Date | Date | Not Null |

Entity Name : Products

| Column Name | Data Type | Remarks |
|---------------|-------------|-------------|
| Product_ID | Integer | Primary Key |
| Category_ID | Integer | Foreign Key |
| Product_Name | Varchar(30) | Not Null |
| Product_Price | Integer | Not Null |

Entity Name : Seller

| Column Name | Data Type | Remarks |
|-------------|-------------|-------------|
| Seller_ID | Integer | Primary_Key |
| Product_ID | Integer | Foreign Key |
| S_Name | Varchar(30) | Not Null |

Entity Name : Payment

| Column Name | Data Type | Remarks |
|-------------|-----------|-------------|
| Payment_ID | Integer | Primary_Key |
| Category_ID | Integer | Foreign_Key |
| P_Date | Date | Not Null |

Entity Name : Transaction Reports

| Column Name | Data Type | Remarks |
|-------------|-----------|-------------|
| Report_ID | Integer | Primary Key |
| Customer_ID | Integer | Foreign Key |
| Order_ID | Integer | Foreign Key |
| Product_ID | Integer | Foreign Key |
| Payment_ID | Integer | Foreign Key |

Entity Name : Login

| Column Name | Data Type | Remarks |
|-------------|-------------|-------------|
| U_User_ID | Integer | Primary Key |
| U_User_Name | Varchar(30) | Not Null |
| Password | Varchar(30) | Not Null |

Entity Name : User Details

| Column Name | Data Type | Remarks |
|--------------|--------------|-------------|
| U_User_ID | Integer | Primary Key |
| U_User_Name | Varchar(30) | Not Null |
| Email_ID | Varchar(30) | Not Null |
| U_Contact_NO | Integer | Not Null |

Entity Name : Cart details

| Column Name | Data Type | Remarks |
|------------------|-----------|-------------|
| Cart_ID | Integer | Primary Key |
| Product_ID | Integer | Foreign Key |
| Product_Quantity | Integer | Not Null |
| Product_Price | Integer | Not Null |

Entity Name : Grocery_Visit

| Column Name | Data Type | Remarks |
|--------------|-----------|----------|
| Date | date | Text |
| time_spent | Integer | Not null |
| amount_spent | Real | Not null |

Entity Name : Grocery_list

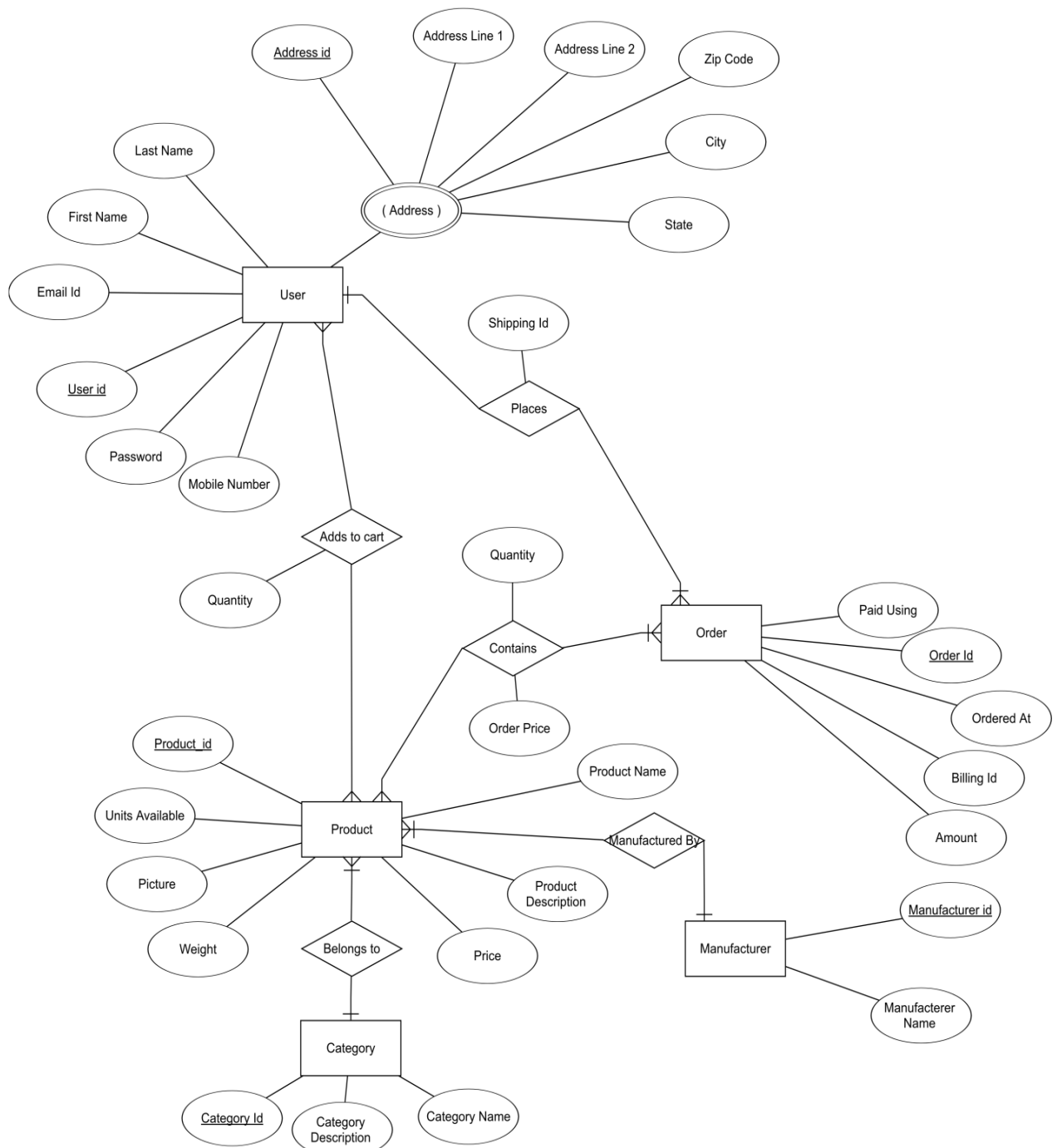
| Column Name | Data Type | Remarks |
|------------------|-------------|----------|
| Date | date | Text |
| Product_Name | Varchar(30) | Not null |
| Product_Category | Varchar(30) | Not null |

Entity Name : dept_store_visit

| Column Name | Data Type | Remarks |
|--------------|-----------|----------|
| entered_date | date | Text |
| time_spent | Integer | Not null |
| amount_spent | Real | Not null |

2.2 ER DIAGRAM

To know in detail about the types of relationships that exist between all the entities and to know the different attributes that describes about the entity we design an Entity Relation Diagram.



3.RELATIONAL MODEL

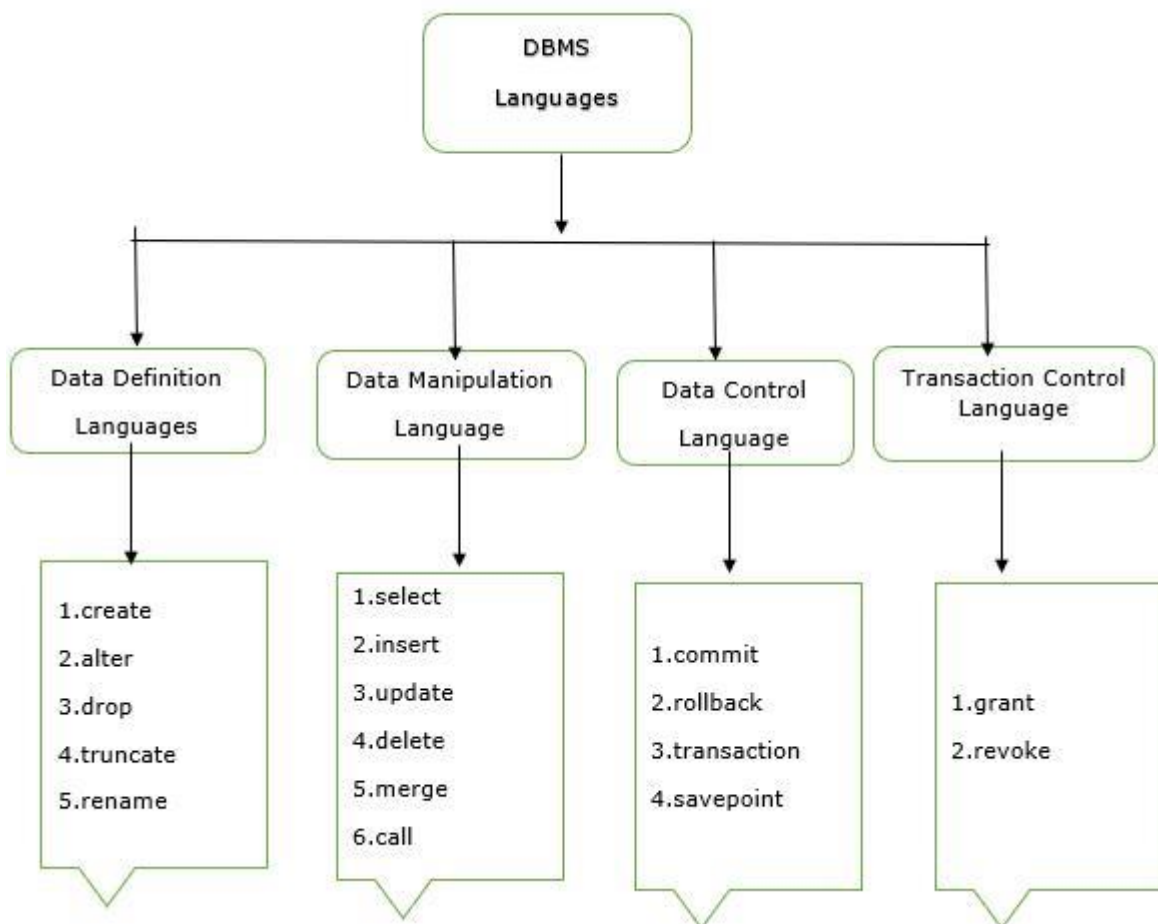
3.1 Database Languages

Once data is stored or filled it requires manipulation like insertion, deletion, updating, and modification of data. For these operations a set of languages are provided by the database management system (DBMS). So, the database languages are used to read, update and store data in the database.

The different types of DBMS languages are as follows –

- ❖ Data Definition Language (DDL) – Create, Drop, Truncate, Rename.
- ❖ Data Manipulation language (DML) – Select, Insert, Delete, Update.
- ❖ Data Control Language (DCL) – Revoke, Grant.
- ❖ Transaction Control Language (TCL) – Rollback, Commit.

The DBMS languages are pictorially represented as follows :



Data Definition Language (DDL)

It is a language that allows the user to define the data and their relationship to other types of data. The DDL commands are: Create, Alter, Rename, Drop, Truncate.

Data Manipulation Language (DML)

It is a language that provides a set of operations to support the basic data manipulation operation on data held in the database. The DML commands are: Insert, delete, update, select, merge, call.

Data Control Language (DCL)

DCL is used to access the stored data. It is mainly used for revoke and grant the user access to a database. The DCL commands are: Grant, Revoke.

Transaction Control Language (TCL)

TCL is a language which manages the transactions within the database. It is used to execute the changes made by the data manipulation language statements. The TCL commands are: Commit, Rollback.

3.2 TABLE DESCRIPTION

1. **Categories** : This table consists of details about the various categories of groceries in the grocery store. The information stored in the table includes category_ID, category_name, category_type.

Constraint: Categories entered should contain C only as a prefix and must exist in **ALL_CATEGORIES** table.

2. **Customers** : The information stored in this table includes Customer_ID, Cu_Name, contact_no, Address,time_spent.

Constraint: Customer entered should contain Cu only as a prefix and must exist in **ALL_Customers** table.

The customer id will be as the primary key constraint for the customer info relation in the database and also the customer id will be considered as the foreign key constraint.

3. **Shopping_order** : This table consists of details about the various categories of groceries in the grocery store. The information stored in the table includes order_ID, customer_ID, date, Address,contact_no.

Constraint:The Order_ID will be considered as a Primary key and the Customer_ID as a Foreign Key.

4. **Delivery** : The information stored in this table includes Delivery_ID, Customer_ID, date.

Constraint:The Delivery_ID will be considered as a Primary key and the Customer_ID as a Foreign Key.

5. **Products** : The information stored in this table includes product_ID, category_ID, product_name,product_category and product_price .

Constraint:The Product_ID will be considered as a Primary key and the Category_ID as a Foreign Key.

6. **Seller** : This table consists of details about the seller_ID,product_ID,S_name.

Constraint:The Seller_ID will be considered as a Primary key and the Product_ID as a Foreign Key.

7. **Payment** : This table contains information about Payment_ID, category_ID, date.

Constraint:The Payment_ID will be considered as a Primary key and the Category_ID as a Foreign Key.

8. **Transaction_Reports** : This transaction table contains the information about the Report_ID, customer_ID, order_ID,product_ID,payment_ID.

Constraint:The Report_ID will be considered as a Primary key and the rest of attributes can be considered as a Foreign Key.

9. **Login** : A login is set of credentials used to authenticate a user. This login consists of a user_id,user_name and password.

Constraint: Only the User_ID will be considered as a Primary key.

10. **User Details** : This table consists of user_id ,user_name, email_id, contact_no.

Constraint: Only the User_ID will be considered as a Primary key.

11. **Cart Details:** This table consists of details about the cart_ID, product_ID,product_name,quantity,product_price.

Constraint:The Cart_ID will be considered as a Primary key and the Product_ID as a Foreign Key

12. **Grocery_visit** : The information stored in this table includes date, time_spent_min, amount_spent.

13. **Grocery_list** : The information stored in this table includes date,product_name, product_category.

14. **Dept_store_visit:** The information stored in this table includes date, time_spent_min, amount_spent.

3.3 RELATIONAL DATABASE SCHEMA

The relational database schema for Online Grocery Store database is as follows :

1. **CATEGORIES** (CAREGORY_ID, CATEGORY_NAME, CATEGORY_TYPE)
2. **CUSTOMERS** (CUSTOMER_ID,Cu_NAME,CONTACT_NO,ADDRESS,TIME_SPENT)
3. **SHOPPING_ORDER** (ORDER_ID, CUSTOMER_ID, ADDRESS,CONTACT_NO, DATE)
4. **DELIVERY** (DELIVERY_ID, CUSTOMER_ID,DATE)
5. **PRODUCTS** (PRODUCT_ID,
CATEGORY_ID,PRODUCT_NAME,PRODUCT_CATEGORY,
PRODUCT_PRICE)
6. **SELLER** (SELLER_ID,PRODUCT_ID,S_NAME)
7. **PAYMENT** (PAYMENT_ID,CATEGORY_ID,DATE)
8. **TRANSACTION_REPORTS** (REPORT_ID, CUSTOMER_ID, ORDER_ID,
PRODUCT_ID,PAYMENT_ID)
9. **LOGIN** (U_USER_ID, U_USER_NAME,PASSWORD)
10. **USERDETAILS** (U_USER_ID,U_USER_NAME,EMAIL_ID,CONTACT_NO)
11. **CARTDETAILS** (
CART_ID,PRODUCT_ID,PRODUCT_NAME,QUANTITY,PRODUCT_PRICE)
12. **Grocery_visit** (DATE,TIME_SPENT_MIN, AMOUNT_SPENT)
13. **Grocery_list**(DATE,PRODUCT_NAME<PRODUCT_CATEGORY)
14. **DEPT_STORE_VISIT** (DATE,TIME_SPENT_MIN, AMOUNT_SPENT)

3.4 RELATIONAL QUERIES

1.Categories Relation

create table categories

```
(
    CAREGORY_ID          int          primary key,
    CATEGORY_NAME        varchar(30)  not null,
    CATEGORY_TYPE        varchar(20)  not null
);
```

Tuples insertion for the table:

```
insert into categories values(1,'Fruits','Eatables');
```

```
insert into categories values(2,'Vegetables','Eatables');
```

```
insert into categories values(3,'Dairy','Eatables');
```

```
insert into categories values(4,'Canned Goods','Eatables');
```

```
insert into categories values(5,'snacks','Eatables');
```

```
insert into categories values(6,'Hair_Products','Daily Essentials');
```

```
insert into categories values(7,'Washing_Poducts','Daily Essentials');
```

```
insert into categories values(8,'Beverages','Drinks');
```

```
select* from categories;
```

OUTPUT:

| CAREGORY_ID | CATEGORY_NAME | CATEGORY_TYPE |
|-------------|-----------------|------------------|
| 1 | Fruits | Eatables |
| 2 | Vegetables | Eatables |
| 3 | Dairy | Eatables |
| 4 | Canned Goods | Eatables |
| 5 | snacks | Eatables |
| 7 | Washing_Poducts | Daily Essentials |
| 8 | Beverages | Drinks |

2.Customer Relation

create table customers

```
(
    Customer_ID          int          primary key,
    Name                 varchar(30)  not null,
```

```

        contact_no    int                not null,
        address       varchar(30)      not null,
        time_spent    integer          not null
    );

```

Tuples insertion for the table:

```

insert into customers values(11,'Fathima',90932411,'kadapa',50);
insert into customers values(12,'Feeza',76533342,'Rayachoty',25);
insert into customers values(13,'Bhavani',96543241,'Chittoor',30);
insert into customers values(14,'Vallusha',94542264,'Tirupati',45);
insert into customers values(15,'Alekhya',94542265,'Kadapa',20);
insert into customers values(16,'Ananya',9454226,'Delhi',10);
insert into customers values(17,'Kaira',94542267,'Kolkata',40);
insert into customers values(18,'Nikhitha',94542268,'Tirupati',60);
select * from customers;

```

OUTPUT:

| Customer_ID | Name | contact_no | address | time_spent |
|-------------|----------|------------|-----------|------------|
| 11 | Fathima | 90932411 | kadapa | 50 |
| 12 | Feeza | 76533342 | Rayachoty | 25 |
| 13 | Bhavani | 96543241 | Chittoor | 30 |
| 14 | Vallusha | 94542264 | Tirupati | 45 |
| 15 | Alekhya | 94542265 | Kadapa | 20 |
| 16 | Ananya | 9454226 | Delhi | 10 |
| 17 | Kaira | 94542267 | Kolkata | 40 |
| 18 | Nikhitha | 94542268 | Tirupati | 60 |

3.Shopping_order

create table Shopping_order

```

(
    order_ID    int primary key,
    customer_ID int foreignkey(Customer_ID)references customer(customer_ID),
    Address     varchar(30)      not null,
    contact_no  int              not null,

```

```

s_date date not null
);

```

Tuples insertion for the table:

```

insert into shopping_order values(100,11,'kadapa',90932411,'25-Jul-2022');
insert into shopping_order values(101,12,'Rayachoty',76533342,'25-Jul-2022');
insert into shopping_order values(102,13,'Chittor',96543241,'28-Jul-2022');
insert into shopping_order values(103,14,'Tirupati',94542264,'29-Jul-2022');
insert into shopping_order values(104,15,'Kadapa',94542265,'30-Jul-2022');
insert into shopping_order values(105,16,'Delhi',94542266,'10-Aug-2022');
insert into shopping_order values(106,17,'Kolkata',94542267,'11-Aug-2022');
insert into shopping_order values(107,18,'Tirupati',94542268,'12-Aug-2022');
select * from Shopping_order;

```

OUTPUT:

| order_ID | customer_ID | Address | contact_no | s_date |
|----------|-------------|-----------|------------|-------------|
| 100 | 11 | kadapa | 90932411 | 25-Jul-2022 |
| 101 | 12 | Rayachoty | 76533342 | 25-Jul-2022 |
| 102 | 13 | Chittor | 96543241 | 28-Jul-2022 |
| 103 | 14 | Tirupati | 94542264 | 29-Jul-2022 |
| 104 | 15 | Kadapa | 94542265 | 30-Jul-2022 |
| 105 | 16 | Delhi | 94542266 | 10-Aug-2022 |
| 106 | 17 | Kolkata | 94542267 | 11-Aug-2022 |
| 107 | 18 | Tirupati | 94542268 | 12-Aug-2022 |

4.Delivery Relation

create table Delivery

```

(
    Delivery_id          int          primary key,
    customer_ID          int foriegn key(customer_ID) references customer(customer_ID),
    d_date               date        not null
);

```

Tuples insertion for the table:

```

insert into delivery values(50,11,'26-Jul-2022');

```

```

insert into delivery values(53,14,'30-Jul-2022');
insert into delivery values(54,15,'31-Jul-2022');
insert into delivery values(55,16,'11-Aug-2022');
insert into delivery values(56,17,'12-Aug-2022');
select * from Delivery;

```

OUTPUT:

| Delivery_id | customer_ID | d_date |
|-------------|-------------|-------------|
| 50 | 11 | 26-Jul-2022 |
| 53 | 14 | 30-Jul-2022 |
| 54 | 15 | 31-Jul-2022 |
| 55 | 16 | 11-Aug-2022 |
| 56 | 17 | 12-Aug-2022 |

5.Products Relation

create table Products

```

(
    product_ID    int primary key,
    CAREGORY_ID   int          foreign key(CAREGORY_ID) references
categories(CAREGORY_ID),
    product_name  varchar(30)    not null,
    product_price int            not null,
);

```

Tuples insertion for the table:

```

insert into product values(200,1,'Mango',50);
insert into product values(201,2,'Tomato',30);
insert into product values(202,3,'Mick',30);
insert into product values(203,4,'Soup',10);
insert into product values(204,5,'Chocolate',100);
insert into product values(205,6,'Shampoo',500);
insert into product values(206,7,'Surf Excel',300);
insert into product values(207,8,'Pepsi',60);
select * from Products;

```


OUTPUT:

| product_ID | CAREGORY_ID | product_name | product_price |
|------------|-------------|--------------|---------------|
| 200 | 1 | Mango | 50 |
| 201 | 2 | Tomato | 30 |
| 202 | 3 | Mick | 30 |
| 203 | 4 | Soup | 10 |
| 204 | 5 | Chocolate | 100 |
| 205 | 6 | Shampoo | 500 |
| 206 | 7 | Surf Excel | 300 |
| 207 | 8 | Pepsi | 60 |

6.Seller

create table Seller

```
(
    seller_ID    int           primary key,
    product_ID   int           foreign    key(product_ID)      references
Products(product_ID),
    S_name       varchar(30)    not null
);
```

Tuples insertion for the table:

```
insert into seller values(10,200,'Naziya');
insert into seller values(12,201,'Naziya');
insert into seller values(112,201,'Naziya');
insert into seller values(19,203,'Naziya');
insert into seller values(29,204,'Nikkam');
insert into seller values(11,205,'Nikkam');
insert into seller values(15,206,'Nikkam');
insert into seller values(5,207,'Nikkam');
select * from Seller;
```

OUTPUT:

| seller_ID | product_ID | S_name |
|------------------|-------------------|---------------|
| 10 | 200 | Naziya |
| 12 | 201 | Naziya |
| 112 | 201 | Naziya |
| 19 | 203 | Naziya |
| 29 | 204 | Nikkam |
| 11 | 205 | Nikkam |
| 15 | 206 | Nikkam |
| 5 | 207 | Nikkam |

7.Payment Relation

create table Payment

```
(
    Payment_ID int primary key,
    CAREGORY_ID int foreign key(CAREGORY_ID) references
categories(CAREGORY_ID),
    p_date date not null
);
```

Tuples insertion for the table:

```
insert into payment values(1000,1,'13-jul-2022');
```

```
insert into payment values(1001,2,'20-jul-2022');
```

```
insert into payment values(1002,3,'6-jul-2022');
```

```
insert into payment values(1003,4,'7-jul-2022');
```

```
select * from Payment;
```

OUTPUT:

| Payment_ID | CATEGORY_ID | p_date |
|------------|-------------|-------------|
| 1000 | 1 | 13-jul-2022 |
| 1001 | 2 | 20-jul-2022 |
| 1002 | 3 | 6-jul-2022 |
| 1003 | 4 | 7-jul-2022 |

8.Transaction_Reports Relation

create table Transaction_Reports

```
(  
    Report_ID    int primary key,  
    customer_ID  int foreign key(customer_ID) references customers(customer_ID),  
    order_ID     int foreign key(order_ID) references Shopping_order(order_ID),  
    product_ID   int foreign key(product_ID) references Products(product_ID),  
    Payment_ID   int foreign key(Payment_ID) references Payment(Payment_ID)  
);
```

Tuples insertion for the table:

```
insert into transaction_reports values(20,11,100,200,1000);
```

```
insert into transaction_reports values(21,12,101,201,1001);
```

```
insert into transaction_reports values(22,13,102,202,1002);
```

```
insert into transaction_reports values(23,14,103,203,1003);
```

```
select * from Transaction_Reports;
```

OUTPUT:

| Report_ID | customer_ID | order_ID | product_ID | Payment_ID |
|-----------|-------------|----------|------------|------------|
| 20 | 11 | 100 | 200 | 1000 |
| 21 | 12 | 101 | 201 | 1001 |
| 22 | 13 | 102 | 202 | 1002 |
| 23 | 14 | 103 | 203 | 1003 |

9.Login Relation

create table Login

```
(
    l_user_id          int          primary key,
    l_user_name        varchar(30)  not null,
    password           varchar(20)  not null
);
```

Tuples insertion for the table:

```
insert into login values(30,'S_Fathima','fa12');
insert into login values(31,'S_Feeza','fe13');
insert into login values(32,'K_Bhavani','bha14');
insert into login values(33,'P_Vallusha','val15');
select * from Login;
```

OUTPUT:

| L_USER_ID | L_USER_NAME | PASSWORD |
|-----------|-------------|----------|
| 30 | S_Fathima | fa12 |
| 31 | S_Feeza | fe13 |
| 32 | K_Bhavani | bha14 |
| 33 | P_Vallusha | val15 |

10. user_detailsRelation

create table user_details

```
(
    u_user_id      int primary key,
    u_user_name    varchar(30)  not null,
    emailid        varchar(30)  not null,
    u_contact_no   int          not null
);
```

Tuples insertion for the table:

```
insert into user_details values(30,'S_Fatima','fathima@gmail.com',90932411);
insert into user_details values(31,'S_Feeza','feeza@gmail.com',76533342);
```

```
insert into user_details values(32,'K_Bhavani','bhavani@gmail.com',96543241);
insert into user_details values(33,'P_Vallusha','vallusha@gmail.com,94542264);
select*from user_details;
```

OUTPUT:

| U_USER_ID | U_USER_NAME | EMAILID | U_CONTACT_NO |
|-----------|-------------|-------------------|--------------|
| 30 | S_Fatima | fathima@gmail.com | 90932411 |
| 31 | S_Feeza | feeza@gmail.com | 76533342 |
| 32 | K_Bhavani | bhavani@gmail.com | 96543241 |

11.Cartdetails Relation

create table Cartdetails

```
(
    cart_ID      int          primary key,
    product_ID   int          foreign      key(product_ID)      references
Products(product_ID),
    product_quantity  int not null,
    product_price     int          not null
);
```

Tuples insertion for the table:

```
insert into cart_details values(41,200,'Mango','1/2kg',50);
insert into cart_details values(42,201,'Tomato','1kg',30);
insert into cart_details values(43,202,'Milk','1lit',30);
insert into cart_details values(44,203,'Soup','5kg',10);
insert into cart_details values(45,205,'Chaco','2kg',100);
insert into cart_details values(46,206,'Surf Excel','1kg',300);
insert into cart_details values(47,207,'Apple','1kg',80);
insert into cart_details values(48,208,'Potato','2kg',40);
Select * From cart_details;
```

OUTPUT:

| cart_ID | product_ID | product_name | product_quantity | product_price |
|---------|------------|--------------|------------------|---------------|
| 41 | 200 | Mango | 1/2kg | 50 |
| 42 | 201 | Tomato | 1kg | 30 |
| 43 | 202 | Milk | 1lit | 30 |
| 44 | 203 | Soup | 5 | 10 |

| | | | | |
|----|-----|------------|---|-----|
| 45 | 205 | Chaco | 2 | 100 |
| 46 | 206 | Surf Excel | 1 | 300 |
| 47 | 207 | Apple | 1 | 80 |
| 48 | 208 | Potato | 2 | 40 |

12.Grocer_visit Relation

create table Grocer_visit

```
(
    date TEXT,
    time_spent_min int,
    amount_spent real
);
```

Tuples insertion for the table:

```
insert into Grocer_visit values ('2020-12-01', 40, 45.99);
```

```
insert into Grocer_visit values ('2020-12-15', 25, 37.30);
```

```
insert into Grocer_visit values ('2020-12-19', 20, 15.82);
```

```
select* From Grocery_visit
```

Output:

| date | time_spent_min | amount_spent |
|------------|----------------|--------------|
| 2020-12-01 | 40 | 45.99 |
| 2020-12-15 | 25 | 37.3 |
| 2020-12-19 | 20 | 15.82 |

13.Grocer_list

create table Grocer_list

```
(
    date TEXT,
    product_name TEXT,
    product_category TEXT
);
```

Tuples insertion for the table:

```
insert into grocer_list values("2020-12-03", "Hamburger patties", "Meat and Fish");
insert into grocer_list values("2020-12-03", "Chips", "Pantry");
insert into grocer_list values("2020-12-03", "Avocado", "Fruits and Vegetables");
insert into grocer_list values("2020-12-03", "Lime", "Fruits and Vegetables");
insert into grocer_list values("2020-12-03", "Tomato", "Fruits and Vegetables");
insert into grocer_list values("2020-12-15", "Rice cakes", "Pantry");
insert into grocer_list values("2020-12-15", "Graham crackers", "Pantry");
insert into grocer_list values("2020-12-15", "Toothpaste", NULL)
```

select * From Grocer_list

OUTPUT:

| date | product_name | product_category |
|------------|-------------------|-----------------------|
| 2020-12-03 | Hamburger patties | Meat and Fish |
| 2020-12-03 | Chips | Pantry |
| 2020-12-03 | Avocado | Fruits and Vegetables |
| 2020-12-03 | Lime | Fruits and Vegetables |
| 2020-12-03 | Tomato | Fruits and Vegetables |
| 2020-12-15 | Rice cakes | Pantry |
| 2020-12-15 | Graham crackers | Pantry |
| 2020-12-15 | Toothpaste | null |

14.dept_store_visit

```
create table dept_store_visit
(
    entered_date    Date,
    time_spent_min int,
    amount_spent   real
);
```

Tuples insertion for the table:

```
insert into dept_store_visit values ('2020-12-01', 40, 45.99);
```

```
insert into dept_store_visit values ('2020-12-15', 25, 37.30);
```

```
insert into dept_store_visit values ('2020-12-19', 20, 15.82);
```

```
select * From dept_store_visit
```

Output:

| entered_date | time_spent_min | amount_spent |
|---------------------|-----------------------|---------------------|
| 2020-12-01 | 40 | 45.99 |
| 2020-12-15 | 25 | 37.3 |
| 2020-12-19 | 20 | 15.82 |

QUERIES: RELATED TO ONLINE GROCERY STORE

--Query: Find the total number of count of different category_name.

```
SELECT COUNT(DISTINCT CATEGORY_NAME)
FROM CATEGORY;
```

Output:

```
COUNT(DISTINCTCATEGORY_NAME)
7
```

--Query:Write a query by using where clause and filter the output to only include items that have a null category.

```
SELECT *
FROM CATEGORY
WHERE CATEGORY_NAME IS NULL;
```

Output:

```
no data found
```

--Query:Write a query to display different categories.

```
SELECT DISTINCT CATEGORY_NAME
FROM CATEGORY_TABLE;
```

Output:

```
CATEGORY_NAME

Dairy
snacks
Washing_Poduc
ts
Vegetables
Beverages
Fruits
```

--Query: Write a query using in operator which will filter the dataset based on the list of items and the list contain only 'Eatables' and 'Daily Essentials'.

```
SELECT *  
FROM CATEGORY  
WHERE CATEGORY_TYPE IN('Eatables','Daily Essentials');
```

Output:

| CATEGORY_ID | CATEGORY_NAME | CATEGORY_TYPE |
|-------------|------------------|------------------|
| 1 | Fruits | Eatables |
| 2 | Vegetables | Eatables |
| 3 | Dairy | Eatables |
| 4 | Canned Goods | Eatables |
| 5 | snacks | Eatables |
| 7 | Washing_Products | Daily Essentials |

--Query: Find the total number of count of different customer_id.

```
SELECT COUNT(DISTINCT CUSTOMER_ID)  
FROM CUSTOMERS;
```

Output:

```
COUNT(DISTINCTCUSTOMER_ID)  
4
```

--Query: Find all the names of customers?

```
SELECT NAME as Customers  
FROM CUSTOMER;
```

Output:

| CUSTOMERS |
|-----------|
| Fathima |
| Feeza |
| Bhavani |
| Vallusha |

--Query: Write a query to find average spent by customers to the grocery store.

```
SELECT AVG(time_spent) AS AVERAGE_TIME_SPENT  
FROM CUSTOMER;
```

Output:

| AVERAGE_TIME_SPENT |
|--------------------|
| 36.25 |

--Query: Write a query to find max time spent made by Vallusha to the grocery store.

```
SELECT MAX(TIME_SPENT) AS max_time_spent  
FROM CUSTOMER;
```

Output:

| MAX_TIME_VISIT |
|----------------|
| 50 |

--Query: Write a query to find max time spent by customers the grocery store.

```
SELECT Min(TIME_SPENT) AS min_time_spent  
FROM CUSTOMER;
```

Output:

| MIN_TIME_SPENT |
|----------------|
| 25 |

--Query: Sales made between 25-07-22 and 29-07-22.

```
SELECT *  
FROM DELIVERY  
WHERE D_DATE BETWEEN "25/07/22" AND "29/07/22";
```

Output:

| Delivery_id | customer_ID | d_date |
|-------------|-------------|--------------|
| 50 | 11 | 26-July-2022 |

--Query: Write the product names whose product price is between 50 to 100.

```
SELECT PRODUCT_NAME
FROM PRODUCTS
WHERE PRODUCT_PRICE BETWEEN 50 AND 100;
```

Output:

product_name

Mango

Chocolate

Pepsi

--Query: Find the name and highest price of the products.

```
SELECT PRODUCT_NAME, MAX(PRODUCT_PRICE)
FROM PRODUCTS;
```

Output:

| Product_name | Max(Product_price) |
|--------------|--------------------|
|--------------|--------------------|

| | |
|---------|-----|
| Shampoo | 500 |
|---------|-----|

--Query: A query to find the average price of the products.

```
SELECT AVG(Product_PRICE)
FROM PRODUCT;
```

Output:

AVG(Product_PRICE)

135

--Query: Write a sql statement that lists the number of users who have logged in with their user names.

```
SELECT COUNT(L_USER_ID), L_USER_NAME
FROM LOGIN
GROUP BY L_USER_NAME;
```

Output:

| COUNT(L_USER_ID) | I_user_name |
|------------------|-------------|
| 1 | A_Kaira |
| 1 | K_Bhavani |
| 1 | P_Alekhya |
| 1 | P_Nikhitha |
| 1 | P_Vallusha |
| 1 | S_Fathima |
| 1 | S_Feeza |
| 1 | T_Ananya |

--Query:Write the names of the seller from seller table without duplicates.

```
SELECT DISTINCT S_NAME
FROM SELLER;
```

Output:

| S_name |
|--------|
| Naziya |
| Nikkam |

--Query:Write a sql CASE ststatement on products table printing product_id,product_name,product_price and stating the price greater, lesser and equal to 100 using conditions.

```
SELECT PRODUCT_ID,PRODUCT_NAME,PRODUCT_PRICE,
CASE
  WHEN PRODUCT_PRICE>100 THEN 'PRICE IS GREETHER THAN 100'
  WHEN PRODUCT_PRICE=100 THEN 'PRICE IS 100'
  WHEN PRODUCT_PRICE<100 THEN 'PRICE IS UNDER 100'
END AS PRICETEXT
FROM PRODUCT;
```

Output:

| product_ID | product_name | product_price | PRICETEXT |
|------------|--------------|---------------|--------------------------|
| 200 | Mango | 50 | PRICE IS UNDER 100 |
| 201 | Tomato | 30 | PRICE IS UNDER 100 |
| 202 | Mick | 30 | PRICE IS UNDER 100 |
| 204 | Chocolate | 100 | PRICE IS 100 |
| 205 | Shampoo | 500 | PRICE IS GRETER THAN 100 |
| 206 | Surf Excel | 300 | PRICE IS GRETER THAN 100 |
| 207 | Pepsi | 60 | PRICE IS UNDER 100 |

--Query:Write a sql statement that returns the products name(duplicate value also)from both the products and cart details tables with limit = 5.

```
SELECT PRODUCT_NAME
FROM PRODUCT
UNION ALL
SELECT PRODUCT_NAME
FROM CART_DETAILS;
```

Output:

product_name

Mango

Tomato

Mick

Soup

Chocolate

--Query:Write a sql statement that selects all shopping order details that has d_date same in the delivery table.

```
SELECT *  
FROM SHOPPING_ORDERS  
WHERE S_DATE IN (SELECT D_DATE FROM DELIVERY);
```

Output:

| order_ID | customer_ID | Address | contact_no | s_date |
|----------|-------------|----------|------------|-------------|
| 106 | 17 | Kolkata | 94542267 | 11-Aug-2022 |
| 107 | 18 | Tirupati | 94542268 | 12-Aug-2022 |

--Query:Write a sql statement which selects all the customers with a customer name ending with a.

```
SELECT *  
FROM CUSTOMER  
WHERE NAME LIKE '%a';
```

Output:

| Customer_ID | Name | contact_no | address | time_spent |
|-------------|----------|------------|-----------|------------|
| 11 | Fathima | 90932411 | kadapa | 50 |
| 12 | Feeza | 76533342 | Rayachoty | 25 |
| 14 | Vallusha | 94542264 | Tirupati | 45 |
| 15 | Alekhya | 94542265 | Kadapa | 20 |
| 16 | Ananya | 9454226 | Delhi | 10 |
| 17 | Kaira | 94542267 | Kolkata | 40 |
| 18 | Nikhitha | 94542268 | Tirupati | 60 |

--Query: Write a sql statement which selects all the customers with address starting with i.

```
SELECT *  
FROM CUSTOMER  
WHERE ADDRESS LIKE '%i';
```

Output:

| Customer_ID | Name | contact_no | address | time_spent |
|-------------|----------|------------|----------|------------|
| 14 | Vallusha | 94542264 | Tirupati | 45 |
| 16 | Ananya | 9454226 | Delhi | 10 |
| 18 | Nikhitha | 94542268 | Tirupati | 60 |

--Query: Find the total number of count of different category_name.

```
SELECT COUNT(DISTINCT CATEGORY_NAME)  
FROM CATEGORIES;
```

Output:

```
COUNT(DISTINCT CATEGORY_NAME)  
8
```

--Query: Find customer name of customer where address=kadapa.

```
SELECT NAME  
FROM CUSTOMER  
WHERE Customer_ID= 15;
```

Output:

```
Name  
Alekhya
```

--Query: Write the customers table in ascending order according to the Address.

```
SELECT *  
FROM CUSTOMERS  
ORDER BY ADDRESS;
```


Output:

| Customer_ID | Name | contact_no | address | time_spent |
|--------------------|-------------|-------------------|----------------|-------------------|
| 13 | Bhavani | 96543241 | Chittoor | 30 |
| 16 | Ananya | 9454226 | Delhi | 10 |
| 15 | Alekhya | 94542265 | Kadapa | 20 |
| 17 | Kaira | 94542267 | Kolkata | 40 |
| 12 | Feeza | 76533342 | Rayachoty | 25 |
| 14 | Vallusha | 94542264 | Tirupati | 45 |
| 18 | Nikhitha | 94542268 | Tirupati | 60 |
| 11 | Fathima | 90932411 | kadapa | 50 |

--Query:Find and write the first three rows from login table.

```
SELECT L_User_Name,L_User_ID,password
FROM LOGIN limit 3;
```

Output:

| l_user_name | l_user_id | password |
|--------------------|------------------|-----------------|
| S_Fathima | 30 | Fa12 |
| S_Feeza | 31 | Fe13 |
| K_Bhavani | 32 | Bha14 |

--Query:Find the data of first four rows from Grocery_list..

```
SELECT *
FROM GROCER_LIST LIMIT 4
```

Output:

| date | product_name | product_category |
|-------------|---------------------|-------------------------|
| 2020-12-03 | Hamburger patties | Meat and Fish |
| 2020-12-03 | Chips | Pantry |
| 2020-12-03 | Avocado | Fruits and Vegetables |
| 2020-12-03 | Lime | Fruits and Vegetables |

4.CONCLUSION AND FUTUREWORK

4.1 Conclusion :

The project entitled ' Online Grocery Store Management System ' is very convenient for the Computer Companies. This system is very convenient for customer or users to buy online computer products. It can be observe that the information can be obtained easily and accurately. The online grocery shopping Software is made more user friendly to the users, so that anyone can run the software. Then this software provide permission to enter to the system via the login password credentials to the user who use this system . This project manages all the details about Computer Products.

4.2 Future Work :

In Future users can buy there identical products using mobile phones. This organization is very beneficial for both users and companies. This product has great future scope. Online grocery project established using web based technology and for Windows too future versions of environments. This project also delivers security by using a security credentials like user id as well as password, so that any illegal users cannot practice your account. The only Lawful person that will consume proper admittance authority can use the online grocery shopping software.

Upcoming work or related work for smart grocery system is we can directly place order by android application in our smart phones, so by using Python developing a Web application to automatically placing order to grocery shops is further execution in this system.

REFERENCE :

"Grocer". Oxford English Dictionary (2nd ed.). 1989.

"Food prices rising across the world", CNN. 24 March 2008

"Grocery Spending Survey". Stretcher.com. 2006-06-20. Retrieved 2014-04-22.