INVENTORY MANAGEMENT SYSTEM

Submitted To:

Dr. Amritpal Singh
Asst. Professor, Dept. Of CSE
Dr. B. R. Ambedkar National Institute of Technology, Jalandhar

NOVEMBER 24, 2020

PRESENTED BY:

ARYAN GARG

19124018

IT(G-1)

PROJECT WORK

INVENTORY MANAGEMENT SYSTEM

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our professor **Dr. Amritpal Singh** as well as **Dr. Harsh Verma** who gave us the golden opportunity to do this wonderful project on the topic**:** **INVENTORY MANAGEMENT SYSTEM** , which also helped us in doing a lot of Research and we came to know about so many new things We are really thankful to them.

Also we would also like to thank them as they provided us great amount of guidance for the project.

"It Was a Very Nice Experience to Have Worked with You Sir in Such an Amazing Project and Subject."

Aryan Garg (19124018)

Ayan Gupta (19124019)

INDEX

PROJECT STARTS FROM HERE

DESCRIPTION OF PROJECT AND PROBLEM TO SOLVE

Brief Description

Inventory management is a systematic approach to sourcing, storing, and selling inventory—both raw materials (components) and finished goods (products).

In business terms, inventory management means the right stock, at the right levels, in the right place, at the right time, and at the right cost as well as price. Also inventory management system is also a logistics system which helps to keep things with themselves and also delivers the product to various other locations and to the various other companies.

As a part of your supply chain, inventory management includes aspects such as controlling and overseeing purchases — from suppliers as well as customers — maintaining the storage of stock, controlling the amount of product for sale, and order fulfilment. A system for identifying every inventory item and its associated information, such as barcode labels or asset tags.

Naturally, your company's precise inventory management meaning will vary based on the types of products you sell and the channels you sell them through. But as long as those basic ingredients are present, you'll have a solid foundation to build upon. Inventory management software, which provides a central database and point of reference for all inventory, coupled with the ability to analyze data, generate reports, forecast future demand, and more.

"Inventory Management System is also called

***Stock Management System**."*

Here in this inventory management system, we have used around 9 tables with capability of having the look into the items available with the company, product names and their id's along with the quantity available with the company with expected date of delivery and the stock expiry date in it.

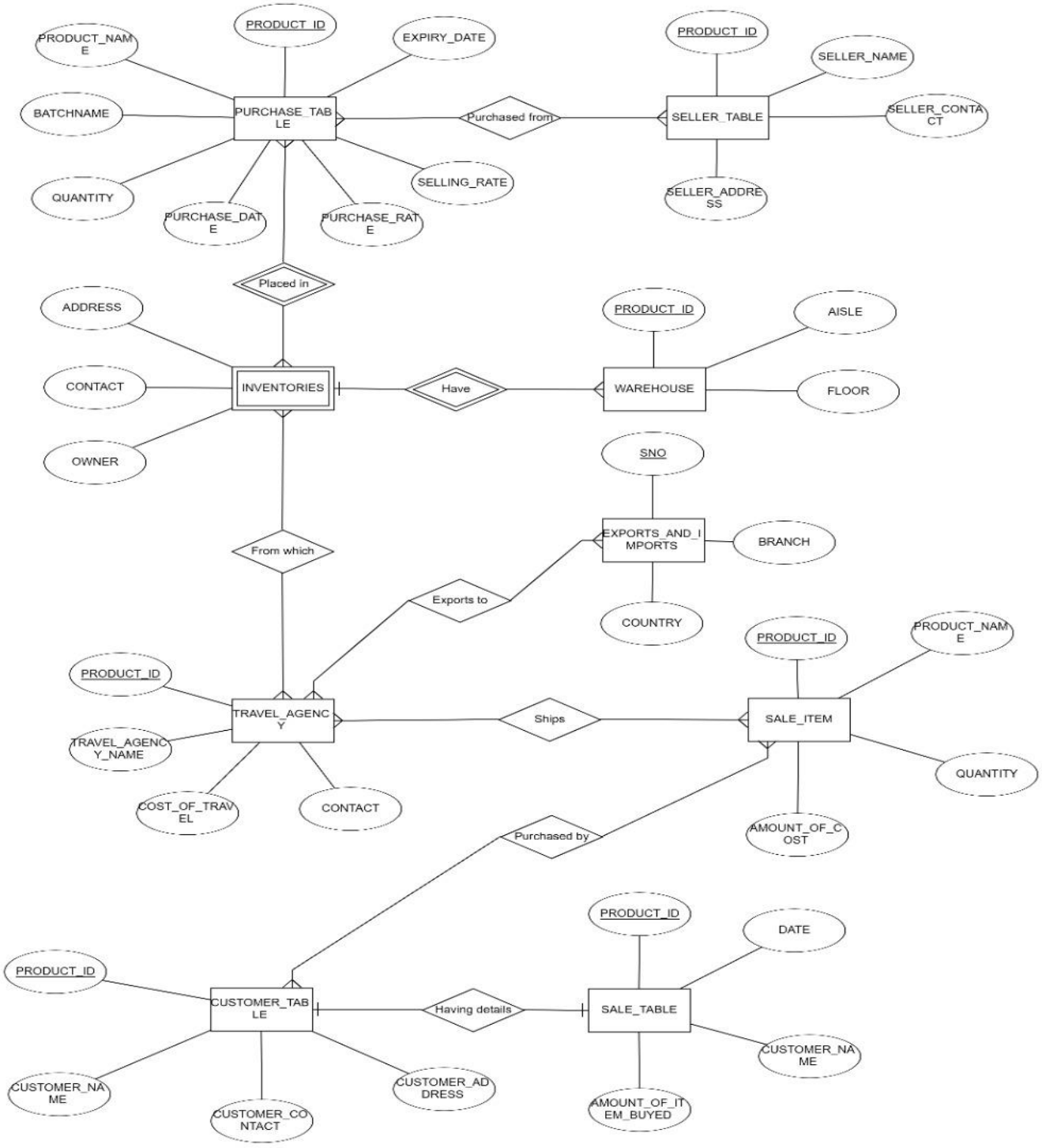
There are also various other tables including the warehouse information in which the company workers could easily search about the shelf and the specific location of the product that is a placed in it.

The system also provides the capability that allows to update information and delete the product from the database and the tables in it.

Look into further information for the specific constraints on the tables and there values present in the database.

Here we have included the inventory management system for managing the electric goods such as shackles, transformers, wires and many more and we have taken tables of inventory management such as purchase_table , warehouse , inventories and many more tables in our database Inventory.

DESCRIPTIVE FUNCTIONALITIES AND SCHEMA DESIGN



In this schema of the we have taken the combination of 9 tables which include PURCHASE, SELLER, CUSTOMER, SALE , SALE_ITEM , WAREHOUSE , INVENTORIES, EXPORTS_AND_IMPORTS ,TRAVEL_AGENCY.

So in this table schema we are including the relations in all of the tables and in this situation we are including the various other attributes or the entities in this. An **Entity–relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

So in this table we have included the Schema in which PRODUCT_ID is very much common in a lot of the tables situated in this. This all includes the entities which are situated in this schema design or the attributes of the tables which are used in this project of the inventory management system.

Here in this ER Diagram we include the many to one or the one to one relationship. In this we include many tables with their respective connection between themselves.

Entity–relationship modeling was developed for database and design by Peter Chen and published in a 1976 paper, with variants of the idea existing previously. Some ER models show super and subtype entities connected by generalization-specialization relationships, and an ER model can be used also in the specification of domain-specific ontologies.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.

In this the oval or the circular shapes with the data gives the table attributes in that table situated inside the tables. The rectangular shapes with the text show the table names inside our database. With that brick or crystal shaped text figure represents the relationship between the tables situated inside our database.

An entity is an object or component of data. An entity is represented as rectangle in an ER diagram.

An entity that cannot be uniquely identified by its own attributes and relies on the relationship with other entity is called weak entity. The weak entity is represented by a double rectangle.

An attribute describes the property of an entity. An attribute is represented as Oval in an ER diagram.

There are four types of attributes:

1.Key attribute
2.Composite attribute
3.Multivalued attribute
4.Derived attribute

A key attribute can uniquely identify an entity from an entity set. Key attribute is represented by oval same as other attributes however the **text of key attribute is underlined**.

An attribute that is a combination of other attributes is known as composite attribute.

There is also the partial participation of the relation inside the tables.

An attribute that can hold multiple values is known as multivalued attribute. It is represented with **double ovals** in an ER Diagram. A derived attribute is one whose value is dynamic and derived from another attribute. It is represented by **dashed oval** in an ER Diagram.

In our case we are having PRODUCT_ID As a key attribute as it is situated in many of the tables and the relation show all of the tables connectivity among each other.

INVENTORIES act a weak entity set as it is dependent on other tables and other entity sets including the entities in between the tables.

Rest the Properties we have mentioned are not being found here inside it but he many to one and the one to one relation could be seen very easily in it.

ER Diagrams are very much useful as they provide all the overview of the schema we are using inside the project and the database and the main functionalities that are being described in the table or the schema design.

So this is the final description of the ER Diagram with the Schema Design Available with us.

QUERIES STARTS FROM HERE

APPLICATIONS OF THE SQL QUERIES AND COMMANDS DESCRIBING ABOUT ALL THE TABLES USED.

First here the project has certain tables to show which uses the commands of creating the table and inserting the values inside the table with describe command will also be displayed in it. We Have Created 9 Tables Inside the Inventory Management System Project. Below ARE Shown All The 9 Tables used in this project .

- TABLE NAME – **PURCHASE_TABLE**

DESCRIBE PURCHASE_TABLE;

```
mysql> DESCRIBE PURCHASE_TABLE;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| PRODUCT_ID    | int         | NO   | PRI | NULL    |       |
| PRODUCT_NAME  | varchar(50) | YES  |     | NULL    |       |
| BATCHNAME     | varchar(50) | YES  |     | NULL    |       |
| QUANTITY      | varchar(50) | YES  |     | NULL    |       |
| PURCHASE_DATE | date        | YES  |     | NULL    |       |
| PURCHASE_RATE | double      | YES  |     | NULL    |       |
| SELLING_RATE  | double      | YES  |     | NULL    |       |
| EXPIRY_DATE   | date        | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
8 rows in set (0.01 sec)
```

```
mysql> SELECT * FROM PURCHASE_TABLE;
+------------+-----------------+-----------+----------+---------------+---------------+--------------+-------------+
| PRODUCT_ID | PRODUCT_NAME    | BATCHNAME | QUANTITY | PURCHASE_DATE | PURCHASE_RATE | SELLING_RATE | EXPIRY_DATE |
+------------+-----------------+-----------+----------+---------------+---------------+--------------+-------------+
|        112 | TRANSFORMERS    | A113      | 120      | 2021-03-12    |         14000 |        18000 | 2022-08-23  |
|        121 | ISOLATORS       | B221      | 250      | 2021-01-26    |         17000 |        40000 | 2023-02-24  |
|        131 | INSULATORS      | B221      | 210      | 2021-01-26    |         11000 |        43000 | 2023-02-24  |
|        132 | PANEL BOARDS    | B221      | 450      | 2022-01-26    |         14500 |        23000 | 2024-09-12  |
|        141 | SHACKLES        | B121      | 140      | 2020-11-26    |         11000 |        20000 | 2025-02-12  |
|        156 | METERS          | B112      | 212      | 2021-01-26    |         14500 |        27850 | 2025-07-13  |
|        159 | CIRCUIT BREAKERS | B121      | 245      | 2022-01-26    |         18500 |        27850 | 2026-01-24  |
|        198 | MCCB            | B142      | 912      | 2020-12-26    |           250 |          650 | 2026-11-25  |
+------------+-----------------+-----------+----------+---------------+---------------+--------------+-------------+
8 rows in set (0.00 sec)
```

- TABLE NAME – **SELLER_TABLE**

DESCRIBE SELLER_TABLE;

```
mysql> DESCRIBE SELLER_TABLE;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| PRODUCT_ID     | int         | YES  | MUL | NULL    |       |
| SELLER_NAME    | varchar(50) | YES  |     | NULL    |       |
| SELLER_CONTACT | varchar(50) | YES  |     | NULL    |       |
| SELLER_ADDRESS | varchar(50) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
4 rows in set (0.11 sec)
```

```
mysql> SELECT * FROM SELLER_TABLE;
+------------+----------------------+----------------+-------------------------------+
| PRODUCT_ID | SELLER_NAME          | SELLER_CONTACT | SELLER_ADDRESS                |
+------------+----------------------+----------------+-------------------------------+
|        112 | SAGAR ENTERPRISES    | 0123-4567578   | 78,MODEL TOWN INDUSTRIES,DELHI |
|        121 | RASAM ENTERPRISES    | 0134-5676784   | 112,SHALINI INDUSTRIES,MUMBAI  |
|        131 | DILAWAR ENTERPRISES  | 0154-5633784   | 12,BRAHAMAAND POINT,BHOPAL     |
|        132 | MAHAKSHAY ENTERPRISES | 0254-5623724   | 189,FOCAL INDUSTRIES,LUDHIANA  |
|        141 | RADHE ENTERPRISES    | 0184-5143724   | 234,RADHE INDUSTRIES,NEW DELHI |
|        156 | KHALSA ENTERPRISES   | 0175-5233774   | 1123,FOCAL POINT,PATIALA       |
|        159 | DHARMA ENTERPRISES   | 0172-4233174   | 23,RESHAM INDUSTRIES,KOLKATA   |
|        198 | CHANDAR ENTERPRISES  | 0212-7893134   | 53,SHIVAM INDUSTRIES,CHENNAI   |
+------------+----------------------+----------------+-------------------------------+
8 rows in set (0.00 sec)
```

- TABLE NAME – **CUSTOMER_TABLE**

DESCRIBE CUSTOMER_TABLE;

```
mysql> DESCRIBE CUSTOMER_TABLE;
+------------------+-------------+------+-----+---------+-------+
| Field            | Type        | Null | Key | Default | Extra |
+------------------+-------------+------+-----+---------+-------+
| PRODUCT_ID       | int         | YES  | UNI | NULL    |       |
| CUSTOMER_NAME    | varchar(50) | NO   |     | NULL    |       |
| CUSTOMER_CONTACT | varchar(50) | NO   |     | NULL    |       |
| CUSTOMER_ADDRESS | varchar(50) | NO   |     | NULL    |       |
+------------------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

```
mysql> SELECT * FROM CUSTOMER_TABLE;
+------------+----------------------+-----------------+-----------------------------------+
| PRODUCT_ID | CUSTOMER_NAME        | CUSTOMER_CONTACT | CUSTOMER_ADDRESS                  |
+------------+----------------------+-----------------+-----------------------------------+
|        112 | DHARAMRAJ CHERALATHAN | 67896-44445     | 12,KONADADERU INDUSTRIES,CHENNAI  |
|        121 | RAJESH PANDIT        | 78968-57678     | 112,SEEMANT SOCIETY,MUMBAI        |
|        131 | SHARIYAR KHAN        | 71248-45678     | 12,JEENAT MAHAL INDUSTRIES,LUCKNOW |
|        132 | ZAHIR KHAN           | 68148-56789     | 10,SHAM INDUSTRIES,JAIPUR         |
|        141 | MAHATHIR HASSAN      | 78668-12321     | 56,JANAH INSUSTRIES,GHAZIABAD     |
|        156 | SHARANJIT SINGH      | 87987-32466     | 12,SINGH ENTERPRISES,AMRITSAR     |
|        159 | HARMAN SINGH         | 98792-32465     | 45,RAKESH ENTERPRISES,HYDERABAD   |
|        198 | KISAV SHERRY         | 86821-75178     | 89,MAHANT ENTERPRISES,AHMEDABAD   |
+------------+----------------------+-----------------+-----------------------------------+
8 rows in set (0.00 sec)
```

- TABLE NAME – **SALE_TABLE**

DESCRIBE SALE_TABLE;

```
mysql> DESCRIBE SALE_TABLE;
+----------------------+--------------+------+-----+---------+-------+
| Field                | Type         | Null | Key | Default | Extra |
+----------------------+--------------+------+-----+---------+-------+
| PRODUCT_ID           | int          | YES  | UNI | NULL    |       |
| DATE                 | datetime     | YES  |     | NULL    |       |
| CUSTOMER_NAME        | varchar(150) | YES  |     | NULL    |       |
| AMOUNT_OF_ITEM_BUYED | double       | YES  |     | NULL    |       |
+----------------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM SALE_TABLE;
+------------+---------------------+----------------------+----------------------+
| PRODUCT_ID | DATE                | CUSTOMER_NAME        | AMOUNT_OF_ITEM_BUYED |
+------------+---------------------+----------------------+----------------------+
|        112 | 2021-04-15 12:30:12 | DHARAMRAJ CHERALATHAN |                19500 |
|        121 | 2022-05-11 11:30:12 | RAJESH PANDIT        |                42000 |
|        131 | 2022-09-21 10:30:42 | SHARIYAR KHAN        |                50000 |
|        132 | 2022-11-23 10:30:23 | ZAHIR KHAN           |                40000 |
|        141 | 2022-11-12 10:23:25 | MAHATHIR HASSAN      |                34600 |
|        156 | 2021-05-21 10:23:24 | SHARANJIT SINGH      |                31600 |
|        159 | 2023-12-22 12:10:24 | HARMAN SINGH         |                30500 |
|        198 | 2021-11-22 09:30:24 | KISAV SHERRY         |                 1200 |
+------------+---------------------+----------------------+----------------------+
8 rows in set (0.02 sec)
```

- TABLE NAME – **SALE_ITEM_TABLE**

  DESCRIBE SALE_ITEM_TABLE;

```
mysql> DESCRIBE SALE_ITEM_TABLE;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| PRODUCT_ID      | int         | YES  | UNI | NULL    |       |
| PRODUCT_NAME    | varchar(50) | YES  |     | NULL    |       |
| QUANTITY        | varchar(50) | YES  |     | NULL    |       |
| AMOUNT_OF_COST  | varchar(50) | YES  |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

```
mysql> SELECT * FROM SALE_ITEM_TABLE;
+------------+------------------+----------+----------------+
| PRODUCT_ID | PRODUCT_NAME     | QUANTITY | AMOUNT_OF_COST |
+------------+------------------+----------+----------------+
|        112 | TRANSFORMERS     | 80       | 19500          |
|        121 | ISOLATORS        | 125      | 42000          |
|        131 | INSULATORS       | 160      | 50000          |
|        132 | PANEL BOARDS     | 300      | 40000          |
|        141 | SHACKLES         | 65       | 34600          |
|        156 | METERS           | 165      | 31600          |
|        159 | CIRCUIT BREAKERS | 200      | 30500          |
|        198 | MCCB             | 600      | 1200           |
+------------+------------------+----------+----------------+
8 rows in set (0.00 sec)
```

- TABLE NAME – **WAREHOUSE**

  DESCRIBE WAREHOUSE;

```
mysql> DESCRIBE WAREHOUSE;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| PRODUCT_ID | int         | NO   | PRI | NULL    |       |
| AISLE      | varchar(20) | YES  |     | NULL    |       |
| FLOOR      | varchar(20) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

```
mysql> SELECT * FROM  WAREHOUSE;
+------------+-------+-------+
| PRODUCT_ID | AISLE | FLOOR |
+------------+-------+-------+
|        112 | 1     | 2     |
|        121 | 3     | 1     |
|        131 | 4     | 5     |
|        132 | 13    | 2     |
|        141 | 3     | 4     |
|        156 | 4     | 2     |
|        159 | 2     | 6     |
|        198 | 4     | 2     |
+------------+-------+-------+
8 rows in set (0.01 sec)
```

- TABLE NAME – **TRAVEL_AGENCY**
- DESCRIBE TRAVEL_AGENCY;

```
mysql> DESCRIBE TRAVEL_AGENCY;
+--------------------+------------+------+-----+---------+-------+
| Field              | Type       | Null | Key | Default | Extra |
+--------------------+------------+------+-----+---------+-------+
| PRODUCT_ID         | int        | YES  | UNI | NULL    |       |
| TRAVEL_AGENCY_NAME | varchar(50) | YES  |     | NULL    |       |
| COST_OF_TRAVEL     | varchar(50) | YES  |     | NULL    |       |
| CONTACT            | varchar(50) | YES  |     | NULL    |       |
+--------------------+------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

```
mysql> SELECT * FROM TRAVEL_AGENCY;
+------------+--------------------+---------------+-------------+
| PRODUCT_ID | TRAVEL_AGENCY_NAME | COST_OF_TRAVEL | CONTACT     |
+------------+--------------------+---------------+-------------+
|        112 | SS TRAVELS         | 5000           | 79887-67657 |
|        121 | JS TRAVELS         | 8000           | 71287-67347 |
|        131 | GS TRAVELS         | 9000           | 71787-67847 |
|        132 | MARSHALL TRAVELS   | 5600           | 88473-83739 |
|        141 | SANDHU TRAVELS     | 7800           | 68978-76821 |
|        156 | RAJDEV TRAVELS     | 10000          | 98978-79921 |
|        159 | SHARMA TRAVELS     | 9600           | 97827-79921 |
|        198 | GREWAL TRAVELS     | 9080           | 79927-79921 |
+------------+--------------------+---------------+-------------+
8 rows in set (0.00 sec)
```

- TABLE NAME – **INVENTORIES**

DESCRIBE INVENTORIES;

```
mysql> SELECT * FROM INVENTORIES;
+-----------------------+--------------------+-------------+
| ADDRESS               | OWNER              | CONTACT     |
+-----------------------+--------------------+-------------+
| 24,AJIT NAGAR,DELHI   | RADHE SHYAM TIWARI | 87897-77682 |
| 12,RAJAT NAGAR,DELHI  | SAJJAN SINGH       | 97921-87991 |
| 24,MODEL TOWN,PATIALA | RANJEET SINGH      | 87799-87991 |
+-----------------------+--------------------+-------------+
3 rows in set (0.00 sec)

mysql> DESCRIBE INVENTORIES;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| ADDRESS | varchar(50) | YES  |     | NULL    |       |
| OWNER   | varchar(50) | YES  |     | NULL    |       |
| CONTACT | varchar(50) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

- TABLE NAME – **EXPORTS_AND_IMPORTS**

DESCRIBE EXPORTS_AND_IMPORTS;

```
mysql> SELECT * FROM EXPORTS_AND_IMPORTS;
+-----+-----------+-----------+
| SNO | BRANCH    | COUNTRY   |
+-----+-----------+-----------+
|   1 | VANCOUVER | CANADA    |
|   2 | NEW DELHI | INDIA     |
|   3 | LONDON    | UK        |
|   4 | NEW YORK  | USA       |
|   5 | BRISBANE  | AUSTRALIA |
+-----+-----------+-----------+
5 rows in set (0.00 sec)

mysql> DESCRIBE EXPORTS_AND_IMPORTS;
+---------+-------------+------+-----+-----------+----------------+
| Field   | Type        | Null | Key | Default   | Extra          |
+---------+-------------+------+-----+-----------+----------------+
| SNO     | int         | NO   | PRI | NULL      | auto_increment |
| BRANCH  | varchar(50) | YES  |     | NEW DELHI |                |
| COUNTRY | varchar(50) | NO   |     | NULL      |                |
+---------+-------------+------+-----+-----------+----------------+
3 rows in set (0.01 sec)
```

APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB-1 & 2 IN PROJECT

1) The Command To Implement The Insert Query Of Inserting The Table In The Database. Here We Will create the database and select it.

USE INVENTORY;

```
mysql> USE INVENTORY;
Database changed
```

2) Now we use the command to create the table by using the create command or the query.

CREATE TABLE CUSTOMER_TABLE(PRODUCT_ID INT UNIQUE,CUSTOMER_NAME(50) NOT NULL,CUSTOMER_CONTACT VARCHAR(50) NOT NULL,CUSTOMER_ADDRESS VARCHAR(50) NOT NULL;

INSERT COMMAND WITH INSERTING VALUES.

```
mysql> CREATE TABLE CUSTOMER_TABLE(PRODUCT_ID INT UNIQUE , CUSTOMER_NAME VARCHAR(50) NOT NULL,CUSTOMER_CONTACT VARCHAR(50) NOT NULL,CUSTOMER_ADDRESS VARCHAR(50) NOT NULL);
Query OK, 0 rows affected (0.06 sec)

mysql> INSERT INTO CUSTOMER_TABLE VALUES(112,"DHARAMRAJ CHERALATHAN","67896-44445","12,KONADADERU INDUSTRIES,CHENNAI");
Query OK, 1 row affected (0.01 sec)
```

3) Now Here we Will use the query to insert the values inside the created table we have created inside our database.
INSERT COMMAND WITH INSERTING VALUES.

```
mysql> CREATE TABLE SELLER_TABLE(PRODUCT_ID INT , SELLER_NAME VARCHAR(50),SELLER_CONTACT VARCHAR(50),SELLER_ADDRESS VARCHAR(50));
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO SELLER_TABLE VALUES(112,"SAGAR ENTERPRISES","0123-4567578","78,MODEL TOWN INDUSTRIES,DELHI");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO SELLER_TABLE VALUES(121,"RASAM ENTERPRISES","0134-5676784","112,SHALINI INDUSTRIES,MUMBAI");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO SELLER_TABLE VALUES(131,"DILAWAR ENTERPRISES","0154-5633784","12,BRAHAMAAND POINT,BHOPAL");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SELLER_TABLE VALUES(132,"MAHAKSHAY ENTERPRISES","0254-5623724","189,FOCAL INDUSTRIES,LUDHIANA");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SELLER_TABLE VALUES(141,"RADHE ENTERPRISES","0184-5143724","234,RADHE INDUSTRIES,NEW DELHI");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO SELLER_TABLE VALUES(156,"KHALSA ENTERPRISES","0175-5233774","1123,FOCAL POINT,PATIALA");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO SELLER_TABLE VALUES(159,"DHARMA ENTERPRISES","0172-4233174","23,RESHAM INDUSTRIES,KOLKATA");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO SELLER_TABLE VALUES(198,"CHANDAR ENTERPRISES","0212-7893134","53,SHIVAM INDUSTRIES,CHENNAI");
Query OK, 1 row affected (0.01 sec)
```

4) Now here we will use the select command or the select query to select something from the table or select something specific from the table.

   SELECT * FROM PURCHASE_TABLE;

```
mysql> SELECT * FROM PURCHASE_TABLE;
+------------+-----------------+-----------+----------+---------------+---------------+--------------+-------------+
| PRODUCT_ID | PRODUCT_NAME    | BATCHNAME | QUANTITY | PURCHASE_DATE | PURCHASE_RATE | SELLING_RATE | EXPIRY_DATE |
+------------+-----------------+-----------+----------+---------------+---------------+--------------+-------------+
|        112 | TRANSFORMERS    | A113      | 120      | 2021-03-12    |         14000 |        18000 | 2022-08-23  |
|        121 | ISOLATORS       | B221      | 250      | 2021-01-26    |         17000 |        40000 | 2023-02-24  |
|        131 | INSULATORS      | B221      | 210      | 2021-01-26    |         11000 |        43000 | 2023-02-24  |
|        132 | PANEL BOARDS    | B221      | 450      | 2022-01-26    |         14500 |        23000 | 2024-09-12  |
|        141 | SHACKLES        | B121      | 140      | 2020-11-26    |         11000 |        20000 | 2025-02-12  |
|        156 | METERS          | B112      | 212      | 2021-01-26    |         14500 |        27850 | 2025-07-13  |
|        159 | CIRCUIT BREAKERS | B121      | 245      | 2022-01-26    |         18500 |        27850 | 2026-01-24  |
|        198 | MCCB            | B142      | 912      | 2020-12-26    |           250 |          650 | 2026-11-25  |
+------------+-----------------+-----------+----------+---------------+---------------+--------------+-------------+
8 rows in set (0.00 sec)
```

5) The command of the describe table will be used to describe whole of the table that what are the attributes we have used and what are the datatypes we have used inside our attributes alongside with the keys we have used and the extras we will use inside the table or the database.

DESCRIBE PURCHASE_TABLE;

```
mysql> DESCRIBE PURCHASE_TABLE;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| PRODUCT_ID    | int         | NO   | PRI | NULL    |       |
| PRODUCT_NAME  | varchar(50) | YES  |     | NULL    |       |
| BATCHNAME     | varchar(50) | YES  |     | NULL    |       |
| QUANTITY      | varchar(50) | YES  |     | NULL    |       |
| PURCHASE_DATE | date        | YES  |     | NULL    |       |
| PURCHASE_RATE | double      | YES  |     | NULL    |       |
| SELLING_RATE  | double      | YES  |     | NULL    |       |
| EXPIRY_DATE   | date        | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
8 rows in set (0.01 sec)
```

6) Here Again the create command will be used similarly to create another table.

CREATE AND THE INSERT COMMAND.

```
mysql> CREATE TABLE SELLER_TABLE(PRODUCT_ID INT , SELLER_NAME VARCHAR(50),SELLER_CONTACT VARCHAR(50),SELLER_ADDRESS VARCHAR(50));
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO SELLER_TABLE VALUES(112,"SAGAR ENTERPRISES","0123-4567578","78,MODEL TOWN INDUSTRIES,DELHI");
Query OK, 1 row affected (0.01 sec)
```

7) Here Again the insert command will be used similarly to insert values in another table.

INSERT COMMAND TO INSERT THE VALUES.

```
mysql> INSERT INTO CUSTOMER_TABLE VALUES(121,"RAJESH PANDIT","78968-57678","112,SEEMANT SOCIETY,MUMBAI");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CUSTOMER_TABLE VALUES(131,"SHARIYAR KHAN","71248-45678","12,JEENAT MAHAL INDUSTRIES,LUCKNOW");
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO CUSTOMER_TABLE VALUES(132,"ZAHIR KHAN","68148-56789","10,SHAM INDUSTRIES,JAIPUR");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CUSTOMER_TABLE VALUES(141,"MAHATHIR HASSAN","78668-12321","56,JANAH INSUSTRIES,GHAZIABAD");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CUSTOMER_TABLE VALUES(156,"SHARANJIT SINGH","87987-32466","12,SINGH ENTERPRISES,AMRITSAR");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CUSTOMER_TABLE VALUES(159,"HARMAN SINGH","98792-32465","45,RAKESH ENTERPRISES,HYDERABAD");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CUSTOMER_TABLE VALUES(198,"KISAV SHERRY","86821-75178","89,MAHANT ENTERPRISES,AHMEDABAD");
Query OK, 1 row affected (0.01 sec)
```

8) Here Again the describe command will be used similarly to describe another table.

DESCRIBE SALE_TABLE;

```
mysql> DESCRIBE SALE_TABLE;
+----------------------+--------------+------+-----+---------+-------+
| Field                | Type         | Null | Key | Default | Extra |
+----------------------+--------------+------+-----+---------+-------+
| PRODUCT_ID           | int          | YES  | UNI | NULL    |       |
| DATE                 | datetime     | YES  |     | NULL    |       |
| CUSTOMER_NAME        | varchar(150) | YES  |     | NULL    |       |
| AMOUNT_OF_ITEM_BUYED | double       | YES  |     | NULL    |       |
+----------------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

9) Here Again the select command will be used similarly to select another table values from the table or the entire table will be selected.

SELECT * FROM TRAVEL_AGENCY;

```
mysql> SELECT * FROM TRAVEL_AGENCY;
+------------+--------------------+----------------+-------------+
| PRODUCT_ID | TRAVEL_AGENCY_NAME | COST_OF_TRAVEL | CONTACT     |
+------------+--------------------+----------------+-------------+
|        112 | SS TRAVELS         | 5000           | 79887-67657 |
|        121 | JS TRAVELS         | 8000           | 71287-67347 |
|        131 | GS TRAVELS         | 9000           | 71787-67847 |
|        132 | MARSHALL TRAVELS   | 5600           | 88473-83739 |
|        141 | SANDHU TRAVELS     | 7800           | 68978-76821 |
|        156 | RAJDEV TRAVELS     | 10000          | 98978-79921 |
|        159 | SHARMA TRAVELS     | 9600           | 97827-79921 |
|        198 | GREWAL TRAVELS     | 9080           | 79927-79921 |
+------------+--------------------+----------------+-------------+
8 rows in set (0.00 sec)
```

10) Here we will use the update commands to update a value or set of values inside the table.

UPDATE PURCHASE_TABLE SET EXPIRY_DATE = “2024-09-12” WHERE PRODUCT_ID= 132;

```
mysql> UPDATE PURCHASE_TABLE SET EXPIRY_DATE="2024-09-12" WHERE PRODUCT_ID =132;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE PURCHASE_TABLE SET EXPIRY_DATE="2025-07-13" WHERE PRODUCT_ID =156;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB-3 IN PROJECT

1) Alter Command is used to add foreign key, primary key or modify the given table.

ALTER TABLE SALE_TABLE ADD CONSTRAINT FK_SALE FOREIGN KEY(PRODUCT_ID) REFERENCES PURCHASE_TABLE(PURCHASE_ID);

```
mysql> ALTER TABLE SALE_TABLE ADD CONSTRAINT FK_SALE FOREIGN KEY(PRODUCT_ID) REFERENCES PURCHASE_TABLE(PRODUCT_ID);
Query OK, 8 rows affected (0.16 sec)
Records: 8  Duplicates: 0  Warnings: 0

mysql> DESCRIBE SALE_TABLE;
+---------------------+-------------+------+-----+---------+-------+
| Field               | Type        | Null | Key | Default | Extra |
+---------------------+-------------+------+-----+---------+-------+
| PRODUCT_ID          | int         | YES  | UNI | NULL    |       |
| DATE                | datetime    | YES  |     | NULL    |       |
| CUSTOMER_NAME       | varchar(150)| YES  |     | NULL    |       |
| AMOUNT_OF_ITEM_BUYED| double      | YES  |     | NULL    |       |
+---------------------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

```
mysql> ALTER TABLE CUSTOMER_TABLE ADD CONSTRAINT FK_CUSTOMER FOREIGN KEY(PRODUCT_ID) REFERENCES PURCHASE_TABLE(PRODUCT_ID);
Query OK, 8 rows affected (0.10 sec)
Records: 8  Duplicates: 0  Warnings: 0
```

2) Drop Command Will be used to drop specific information from the table.

DROP TABLE CUSTOMER_TABLE;

```
mysql> DROP TABLE CUSTOMER_TABLE;
Query OK, 0 rows affected (0.04 sec)
```

3) Delete Command Will used to delete a row from the table .

DELETE FROM SELLER_TABLE WHERE SELLER_NAME = “ZAHIR KHAN”;

```
mysql> DELETE FROM SELLER_TABLE WHERE SELLER_NAME="ZAHIR KHAN";
Query OK, 1 row affected (0.01 sec)
```

4) Here We will be using the like and where command in select query which select product_name whose name starts from ‘TRANS’.

SELECT PRODUCT_NAME FROM PURCHASE_TABLE WHERE PRODUCT_NAME LIKE “TRANS%”;

```
mysql> SELECT PRODUCT_NAME FROM PURCHASE_TABLE WHERE PRODUCT_NAME LIKE "TRANS%";
+--------------+
| PRODUCT_NAME |
+--------------+
| TRANSFORMERS |
+--------------+
1 row in set (0.01 sec)
```

5) Select some information from the tables which allows us to use like command whose batch name starts from “BB”.

SELECT PURCHASE_RATE,BATCHNAME FROM PURCHASE_TABLE WHERE BATCHNAME LIKE “B2%”;

```
mysql> SELECT PURCHASE_RATE,BATCHNAME FROM PURCHASE_TABLE WHERE BATCHNAME LIKE "B2%";
+---------------+-----------+
| PURCHASE_RATE | BATCHNAME |
+---------------+-----------+
|         17000 | B221      |
|         11000 | B221      |
|         14500 | B221      |
+---------------+-----------+
3 rows in set (0.00 sec)
```

6) If we want to select seller_name, seller_contact from the table whose address is situated in “DELHI”.

SELECT SELLER_NAME, SELLER_CONTACT FROM seller_TABLE WHERE SELLER_ADDRESS LIKE “%DELHI%”;

```
mysql> SELECT SELLER_NAME,SELLER_CONTACT FROM SELLER_TABLE WHERE SELLER_ADDRESS LIKE "%DELHI%";
+------------------+----------------+
| SELLER_NAME      | SELLER_CONTACT |
+------------------+----------------+
| SAGAR ENTERPRISES | 0123-4567578   |
| RADHE ENTERPRISES | 0184-5143724   |
+------------------+----------------+
2 rows in set (0.00 sec)
```

7) Select The specific Information from the table whose city names consists of “BAD” consisting of various cities.

SELECT CUSTOMER_NAME,CUSTOMER_CONTACT,PRODUCT_ID FROM CUSTOMER_TABLE WHERE CUSTOMER_ADDRESS LIKE “%BAD%”;

```
mysql> SELECT CUSTOMER_NAME,CUSTOMER_CONTACT,PRODUCT_ID FROM CUSTOMER_TABLE WHERE CUSTOMER_ADDRESS LIKE "%BAD%";
+----------------+------------------+------------+
| CUSTOMER_NAME  | CUSTOMER_CONTACT | PRODUCT_ID |
+----------------+------------------+------------+
| MAHATHIR HASSAN | 78668-12321     |        141 |
| HARMAN SINGH    | 98792-32465     |        159 |
| KISAV SHERRY    | 86821-75178     |        198 |
+----------------+------------------+------------+
3 rows in set (0.00 sec)

mysql> SELECT CUSTOMER_NAME,CUSTOMER_CONTACT,PRODUCT_ID,CUSTOMER_ADDRESS FROM CUSTOMER_TABLE WHERE CUSTOMER_ADDRESS LIKE "%BAD%";
+----------------+------------------+------------+--------------------------------+
| CUSTOMER_NAME  | CUSTOMER_CONTACT | PRODUCT_ID | CUSTOMER_ADDRESS               |
+----------------+------------------+------------+--------------------------------+
| MAHATHIR HASSAN | 78668-12321     |        141 | 56,JANAH INSUSTRIES,GHAZIABAD  |
| HARMAN SINGH    | 98792-32465     |        159 | 45,RAKESH ENTERPRISES,HYDERABAD |
| KISAV SHERRY    | 86821-75178     |        198 | 89,MAHANT ENTERPRISES,AHMEDABAD |
+----------------+------------------+------------+--------------------------------+
3 rows in set (0.00 sec)
```

8) If We want to select specific information from the table which has the info to select some contact info from table which has first two digits having “71” in them.

SELECT TRAVEL_AGENCY_NAME, CONTACT FROM TRAVEL_AGENCY WHERE CONTACT LIKE “71%”;

```
mysql> SELECT TRAVEL_AGENCY_NAME,CONTACT FROM TRAVEL_AGENCY WHERE CONTACT LIKE "71%";
+--------------------+-------------+
| TRAVEL_AGENCY_NAME | CONTACT     |
+--------------------+-------------+
| JS TRAVELS         | 71287-67347 |
| GS TRAVELS         | 71787-67847 |
+--------------------+-------------+
2 rows in set (0.00 sec)
```

9) Select some info from the table from the table which have the cost ending from “0” in the table.

SELECT PRODUCT_NAME,QUANTITY,AMOUNT_OF_COST FROM SALE_ITEM_TABLE WHERE AMOUNT_OF_COST LIKE “_0%”;

```
mysql> SELECT PRODUCT_NAME,QUANTITY,AMOUNT_OF_COST FROM SALE_ITEM_TABLE WHERE AMOUNT_OF_COST LIKE "_0%";
+-----------------+----------+----------------+
| PRODUCT_NAME    | QUANTITY | AMOUNT_OF_COST |
+-----------------+----------+----------------+
| INSULATORS      | 160      | 50000          |
| PANEL BOARDS    | 300      | 40000          |
| CIRCUIT BREAKERS | 200     | 30500          |
+-----------------+----------+----------------+
3 rows in set (0.00 sec)
```

10) If We Want To select Aisle from warehouse which has floor “3” in the warehouse.

SELECT FLOOR FROM WAREHOUSE WHERE AISLE LIKE “3”;

```
mysql> SELECT FLOOR FROM WAREHOUSE WHERE AISLE LIKE "3";
+-------+
| FLOOR |
+-------+
| 1     |
| 4     |
+-------+
2 rows in set (0.00 sec)
```

APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB-4 IN PROJECT

1) A command to implement auto_increment , check ,primary key and various other constraint s inside the table.

CREATE TABLE EXPORTS_AND_IMPORTS(SNO INT AUTO_INCREMENT,BRANCH VARCHAR(50) DEFAULT 'NEW DELHI',COUNTRY VARCHAR(50) NOT NULL CHECK(COUNTRY IN("INDIA", "USA" , "UK" , "CANADA" , "AUSTRALIA")) , PRIMARY KEY(SNO));

```
mysql> CREATE TABLE EXPORTS_AND_IMPORTS(SNO INT AUTO_INCREMENT,BRANCH VARCHAR(50) DEFAULT 'NEW DELHI',COUNTRY VARCHAR(50) NOT NULL CHECK(COUNTRY IN ("INDIA","USA","UK","CAN
ADA","AUSTRALIA")),PRIMARY KEY(SNO));
Query OK, 0 rows affected (0.04 sec)
```

2) A command of implementing the alter command inside the table of adding of the foreign key inside the table.

ALTER TABLE CUSTOMER_TABLE ADD CONSTRAINT FK_CUSTOMER FOREIGN KEY(PRODUCT_ID) REFRENCES PURCHASE_TABLE(PRODUCT_ID);

```
mysql> ALTER TABLE CUSTOMER_TABLE ADD CONSTRAINT FK_CUSTOMER FOREIGN KEY(PRODUCT_ID) REFERENCES PURCHASE_TABLE(PRODUCT_ID);
Query OK, 8 rows affected (0.10 sec)
Records: 8  Duplicates: 0  Warnings: 0
```

3) A create command to implement the not null constraint on the table that will not accept the null values.

CREATE TABLE CUSTOMER_TABLE(PRODUCT_ID INT UNIQUE , CUSTOMER_NAME VARCHAR(50) NOT NULL,CUSTOMER_CONTACT VARCHAR(50) NOT NULL , CUSTOMER_ADDRESS VARCHAR(50) NOT NULL);

INSERT COMMAND TO INSERT VALUES IN TABLE.

```
mysql> CREATE TABLE CUSTOMER_TABLE(PRODUCT_ID INT UNIQUE , CUSTOMER_NAME VARCHAR(50) NOT NULL,CUSTOMER_CONTACT VARCHAR(50) NOT NULL,CUSTOMER_ADDRESS VARCHAR(50) NOT NULL);
Query OK, 0 rows affected (0.06 sec)

mysql> INSERT INTO CUSTOMER_TABLE VALUES(112,"DHARAMRAJ CHERALATHAN","67896-44445","12,KONADADERU INDUSTRIES,CHENNAI");
Query OK, 1 row affected (0.01 sec)
```

4) A describe command that will describe the table which will tell about the different datatypes present inside the table or the extras which are present inside the table.

ALTER TABLE WAREHOUSE ADD PRIMARY KEY(PRODUCT_ID);

```
mysql> ALTER TABLE WAREHOUSE ADD PRIMARY KEY(PRODUCT_ID);
Query OK, 0 rows affected (0.18 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE WAREHOUSE;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| PRODUCT_ID | int         | NO   | PRI | NULL    |       |
| AISLE      | varchar(20) | YES  |     | NULL    |       |
| FLOOR      | varchar(20) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.02 sec)
```

5) A command to implement distinct batch name from the table .

SELECT DISTINCT BATCHNAME FROM PURCHASE_TABLE;

```
mysql> SELECT DISTINCT BATCHNAME FROM PURCHASE_TABLE;
+-----------+
| BATCHNAME |
+-----------+
| A113      |
| B221      |
| B121      |
| B112      |
| B142      |
+-----------+
5 rows in set (0.00 sec)
```

6) A command to implement the Or in the sql commands which give us the conditional statements.

SELECT PRODUCT_ID, AMOUNT_OF_ITEM_BUYED FROM SALE_TABLE WHERE AMOUNT_OF_ITEM_BUYED>10000 OR AMOUNT_OF_ITEM_BUYED<35000 ;

```
mysql> SELECT PRODUCT_ID,AMOUNT_OF_ITEM_BUYED FROM SALE_TABLE WHERE AMOUNT_OF_ITEM_BUYED>10000 OR AMOUNT_OF_ITEM_BUYED <35000;
+------------+----------------------+
| PRODUCT_ID | AMOUNT_OF_ITEM_BUYED |
+------------+----------------------+
|        112 |                19500 |
|        121 |                42000 |
|        131 |                50000 |
|        132 |                40000 |
|        141 |                34600 |
|        156 |                31600 |
|        159 |                30500 |
|        198 |                 1200 |
+------------+----------------------+
8 rows in set (0.00 sec)
```

7) A command to implement the alias or the as command query inside our command and also finding the sum of selling_rate.

   SELECT SUM(SELLING_RATE) AS NET_SP,SUM(PURCHASE_RATE) AS NET_CP , SUM(SELLING_RATE) – SUM(PURCHASE_RATE) AS NET_EXPECTED_PROFIT FROM PURCHASE_TABLE;

```
mysql> SELECT SUM(SELLING_RATE) AS NET_SP,SUM(PURCHASE_RATE) AS NET_CP , SUM(SELLING_RATE)-SUM(PURCHASE_RATE) AS NET_EXPECTED_PROFIT FROM PURCHASE_TABLE;
+--------+--------+---------------------+
| NET_SP | NET_CP | NET_EXPECTED_PROFIT |
+--------+--------+---------------------+
| 200350 | 100750 |               99600 |
+--------+--------+---------------------+
1 row in set (0.00 sec)
```

8) Select command with and constraint or condition which imposes that to select purchase_rate and selling_rate in a range and also the use of between which specifies the range of given values .

SELECT PRODUCT_NAME FROM SALE_ITEM_TABLE WHERE QUANTITY BETWEEN 100 ANS 156;

```
mysql> SELECT PRODUCT_NAME FROM SALE_ITEM_TABLE WHERE QUANTITY BETWEEN 100 AND 156;
+--------------+
| PRODUCT_NAME |
+--------------+
| ISOLATORS    |
+--------------+
1 row in set (0.00 sec)
```

9) Select command with the in clause to select batchname from the given set of values given in the command.

SELECT * FROM PURCHASE_TABLE WHERE BATCHNAME IN (“B221” , “B112” , “B142”);

```
mysql> SELECT * FROM PURCHASE_TABLE WHERE BATCHNAME IN ("B221","B112","B142");
+------------+--------------+-----------+----------+---------------+---------------+--------------+-------------+
| PRODUCT_ID | PRODUCT_NAME | BATCHNAME | QUANTITY | PURCHASE_DATE | PURCHASE_RATE | SELLING_RATE | EXPIRY_DATE |
+------------+--------------+-----------+----------+---------------+---------------+--------------+-------------+
|        121 | ISOLATORS    | B221      | 250      | 2021-01-26    |         17000 |        40000 | 2023-02-24  |
|        131 | INSULATORS   | B221      | 210      | 2021-01-26    |         11000 |        43000 | 2023-02-24  |
|        132 | PANEL BOARDS | B221      | 450      | 2022-01-26    |         14500 |        23000 | 2024-09-12  |
|        156 | METERS       | B112      | 212      | 2021-01-26    |         14500 |        27850 | 2025-07-13  |
|        198 | MCCB         | B142      | 912      | 2020-12-26    |           250 |          650 | 2026-11-25  |
+------------+--------------+-----------+----------+---------------+---------------+--------------+-------------+
5 rows in set (0.00 sec)
```

10) Select command to implement the command which specifies the set of values not inside the set of values which we specify the info by saying that batch name not in set of values.

SELECT PRODUCT_ID , PRODUCT_NAME , SELLING_RATE FROM PURCHASE_TABLE WHERE BATCHNAME NOT IN (“A113” , “B112”, “B142” , “B121”);

```
mysql> SELECT PRODUCT_ID,PRODUCT_NAME,SELLING_RATE FROM PURCHASE_TABLE WHERE BATCHNAME NOT IN ("A113","B112","B142","B121");
+------------+--------------+--------------+
| PRODUCT_ID | PRODUCT_NAME | SELLING_RATE |
+------------+--------------+--------------+
|        121 | ISOLATORS    |        40000 |
|        131 | INSULATORS   |        43000 |
|        132 | PANEL BOARDS |        23000 |
+------------+--------------+--------------+
3 rows in set (0.00 sec)
```

11) Select command to implement the order by clause by use of descending order of values by ordering the selling_rate in descending order.

SELECT * FROM PURCHASE_TABLE ORDER BY SELLING_RATE DESC;

```
mysql> SELECT * FROM PURCHASE_TABLE ORDER BY SELLING_RATE DESC;
+------------+-----------------+-----------+----------+---------------+---------------+--------------+-------------+
| PRODUCT_ID | PRODUCT_NAME    | BATCHNAME | QUANTITY | PURCHASE_DATE | PURCHASE_RATE | SELLING_RATE | EXPIRY_DATE |
+------------+-----------------+-----------+----------+---------------+---------------+--------------+-------------+
|        131 | INSULATORS      | B221      | 210      | 2021-01-26    |         11000 |        43000 | 2023-02-24  |
|        121 | ISOLATORS       | B221      | 250      | 2021-01-26    |         17000 |        40000 | 2023-02-24  |
|        156 | METERS          | B112      | 212      | 2021-01-26    |         14500 |        27850 | 2025-07-13  |
|        159 | CIRCUIT BREAKERS | B121      | 245      | 2022-01-26    |         18500 |        27850 | 2026-01-24  |
|        132 | PANEL BOARDS    | B221      | 450      | 2022-01-26    |         14500 |        23000 | 2024-09-12  |
|        141 | SHACKLES        | B121      | 140      | 2020-11-26    |         11000 |        20000 | 2025-02-12  |
|        112 | TRANSFORMERS    | A113      | 120      | 2021-03-12    |         14000 |        18000 | 2022-08-23  |
|        198 | MCCB            | B142      | 912      | 2020-12-26    |           250 |          650 | 2026-11-25  |
+------------+-----------------+-----------+----------+---------------+---------------+--------------+-------------+
8 rows in set (0.00 sec)
```

12) Select command to implement the group by clause to display all of the values by grouping the data according to a specific attribute as we do here by selecting a set of values from table by grouping according to branch.

SELECT SNO,COUNTRY FROM EXPORTS_AND_IMPORTS GROUP BY BRANCH;

```
mysql> SELECT SNO,COUNTRY FROM EXPORTS_AND_IMPORTS GROUP BY BRANCH;
+-----+-----------+
| SNO | COUNTRY   |
+-----+-----------+
|   1 | CANADA    |
|   2 | INDIA     |
|   3 | UK        |
|   4 | USA       |
|   5 | AUSTRALIA |
+-----+-----------+
5 rows in set (0.01 sec)
```

13) Select command which makes the use of count clause as here we count the no. of rows in the purchase_table.

SELECT COUNT(*) FROM PURCHASE_TABLE;

```
mysql> SELECT COUNT(*) FROM PURCHASE_TABLE;
+----------+
| COUNT(*) |
+----------+
|        8 |
+----------+
1 row in set (0.01 sec)
```

APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB-5 IN PROJECT

1) Select command with the distinct constraint which selects distinct address, owner of inventories table.

SELECT DISTINCT ADDRESS,OWNER FROM INVENTORIES;

```
mysql> SELECT DISTINCT ADDRESS,OWNER FROM INVENTORIES ;
+-------------------------+---------------------+
| ADDRESS                 | OWNER               |
+-------------------------+---------------------+
| 24,AJIT NAGAR,DELHI     | RADHE SHYAM TIWARI  |
| 12,RAJAT NAGAR,DELHI    | SAJJAN SINGH        |
| 24,MODEL TOWN,PATIALA   | RANJEET SINGH       |
+-------------------------+---------------------+
3 rows in set (0.01 sec)
```

2) Select Command with the where and and clause which gives the specific range of values to be included of selling_rate.

SELECT PRODUCT_NAME,PRODUCT_ID,PURCHASE_DATE FROM PURCHASE_TABLE WHERE SELLING_RATE>12300 AND SELLING_RATE<43555;

```
mysql> SELECT PRODUCT_NAME,PRODUCT_ID,PURCHASE_DATE FROM PURCHASE_TABLE WHERE SELLING_RATE>12300 AND SELLING_RATE<43555;
+-----------------+------------+--------------+
| PRODUCT_NAME    | PRODUCT_ID | PURCHASE_DATE |
+-----------------+------------+--------------+
| TRANSFORMERS    |        112 | 2021-03-12   |
| ISOLATORS       |        121 | 2021-01-26   |
| INSULATORS      |        131 | 2021-01-26   |
| PANEL BOARDS    |        132 | 2022-01-26   |
| SHACKLES        |        141 | 2020-11-26   |
| METERS          |        156 | 2021-01-26   |
| CIRCUIT BREAKERS |       159 | 2022-01-26   |
+-----------------+------------+--------------+
7 rows in set (0.00 sec)
```

3) Select Command with the where and or clause which gives the specific range of values to be included of selling_rate.

SELECT PRODUCT_NAME,PURCHASE_RATE,SELLING_RATE,PURCHASE_DATE FROM PURCHASE_TABLE WHERE SELLING_RATE>10300 OR SELLING_RATE<4453;

```
mysql> SELECT PRODUCT_NAME,PURCHASE_RATE,SELLING_RATE,PURCHASE_DATE FROM PURCHASE_TABLE WHERE SELLING_RATE>10300 OR SELLING_RATE<4453;
+-----------------+---------------+--------------+---------------+
| PRODUCT_NAME    | PURCHASE_RATE | SELLING_RATE | PURCHASE_DATE |
+-----------------+---------------+--------------+---------------+
| TRANSFORMERS    |         14000 |        18000 | 2021-03-12    |
| ISOLATORS       |         17000 |        40000 | 2021-01-26    |
| INSULATORS      |         11000 |        43000 | 2021-01-26    |
| PANEL BOARDS    |         14500 |        23000 | 2022-01-26    |
| SHACKLES        |         11000 |        20000 | 2020-11-26    |
| METERS          |         14500 |        27850 | 2021-01-26    |
| CIRCUIT BREAKERS |         18500 |        27850 | 2022-01-26    |
| MCCB            |           250 |          650 | 2020-12-26    |
+-----------------+---------------+--------------+---------------+
8 rows in set (0.00 sec)
```

4) Concat command in select query uses the concatenation of various values of attribuyes of travel_agency table.

SELECT CONCAT(TRAVEL_AGENCY_NAME, " – " , CONTACT) AS TRAVEL_DETAILS FROM TRAVEL_AGENCY;

```
mysql> SELECT CONCAT(TRAVEL_AGENCY_NAME," - ",COST_OF_TRAVEL," - ",CONTACT)  AS TRAVEL_DETAILS FROM TRAVEL_AGENCY;
+--------------------------------------+
| TRAVEL_DETAILS                       |
+--------------------------------------+
| SS TRAVELS - 5000 - 79887-67657      |
| JS TRAVELS - 8000 - 71287-67347      |
| GS TRAVELS - 9000 - 71787-67847      |
| MARSHALL TRAVELS - 5600 - 88473-83739 |
| SANDHU TRAVELS - 7800 - 68978-76821  |
| RAJDEV TRAVELS - 10000 - 98978-79921 |
| SHARMA TRAVELS - 9600 - 97827-79921  |
| GREWAL TRAVELS - 9080 - 79927-79921  |
+--------------------------------------+
8 rows in set (0.00 sec)
```

5) Select command which uses as command in the use of between clause by the use of taking the range of values of selling_rate.

SELECT PRODUCT_ID,PRODUCT_NAME FROM PURCHASE_TABLE WHERE PURCHASE_RATE >12000 AND SELLING_RATE<30000;

```
mysql> SELECT PRODUCT_ID,PRODUCT_NAME FROM PURCHASE_TABLE WHERE PURCHASE_RATE>12000 AND SELLING_RATE<30000;
+------------+-----------------+
| PRODUCT_ID | PRODUCT_NAME    |
+------------+-----------------+
|        112 | TRANSFORMERS    |
|        132 | PANEL BOARDS    |
|        156 | METERS          |
|        159 | CIRCUIT BREAKERS |
+------------+-----------------+
4 rows in set (0.00 sec)
```

6) Select command which makes use of in clause to take a set of values in batchname.

SELECT * FROM PURCHASE_TABLE WHERE BATCHNAME IN (“B221”, “B112” , “B142”);

```
mysql> SELECT * FROM PURCHASE_TABLE WHERE BATCHNAME IN ("B221","B112","B142");
+------------+--------------+-----------+----------+---------------+---------------+--------------+-------------+
| PRODUCT_ID | PRODUCT_NAME | BATCHNAME | QUANTITY | PURCHASE_DATE | PURCHASE_RATE | SELLING_RATE | EXPIRY_DATE |
+------------+--------------+-----------+----------+---------------+---------------+--------------+-------------+
|        121 | ISOLATORS    | B221      | 250      | 2021-01-26    |         17000 |        40000 | 2023-02-24  |
|        131 | INSULATORS   | B221      | 210      | 2021-01-26    |         11000 |        43000 | 2023-02-24  |
|        132 | PANEL BOARDS | B221      | 450      | 2022-01-26    |         14500 |        23000 | 2024-09-12  |
|        156 | METERS       | B112      | 212      | 2021-01-26    |         14500 |        27850 | 2025-07-13  |
|        198 | MCCB         | B142      | 912      | 2020-12-26    |           250 |          650 | 2026-11-25  |
+------------+--------------+-----------+----------+---------------+---------------+--------------+-------------+
5 rows in set (0.00 sec)
```

7) Select command which makes use of not in clause to not take a set of values in batchname.

SELECT PRODUCT_ID,PRODUCT_NAME,SELLING_RATE FROM PURCHASE_TABLE WHERE BATCHNAME NOT IN (“A113” , “B112” , “B142”, “B121”);

```
mysql> SELECT PRODUCT_ID,PRODUCT_NAME,SELLING_RATE FROM PURCHASE_TABLE WHERE BATCHNAME NOT IN ("A113","B112","B142","B121");
+------------+--------------+--------------+
| PRODUCT_ID | PRODUCT_NAME | SELLING_RATE |
+------------+--------------+--------------+
|        121 | ISOLATORS    |        40000 |
|        131 | INSULATORS   |        43000 |
|        132 | PANEL BOARDS |        23000 |
+------------+--------------+--------------+
3 rows in set (0.00 sec)
```

8) Select command which makes the use of between and and clause which makes use of taking the set of values of quantity in a specific range of values and selects some info.

SELECT PRODUCT_NAME FROM SALE_ITEM_TABLE WHERE QUANTITY BETWEEN 100 AND 156;

```
mysql> SELECT PRODUCT_NAME FROM SALE_ITEM_TABLE WHERE QUANTITY BETWEEN 100 AND 156;
+--------------+
| PRODUCT_NAME |
+--------------+
| ISOLATORS    |
+--------------+
1 row in set (0.00 sec)
```

9) Select command which makes the use of not null values of the customer_table in it.

SELECT PRODUCT_ID , CUSTOMER_NAME FROM CUSTOMER_TABLE WHERE CUSTOMER_ADDRESS IS NOT NULL;

```
mysql> SELECT PRODUCT_ID,CUSTOMER_NAME FROM CUSTOMER_TABLE WHERE CUSTOMER_ADDRESS IS NOT NULL;
+------------+----------------------+
| PRODUCT_ID | CUSTOMER_NAME        |
+------------+----------------------+
|        112 | DHARAMRAJ CHERALATHAN |
|        121 | RAJESH PANDIT        |
|        131 | SHARIYAR KHAN        |
|        132 | ZAHIR KHAN           |
|        141 | MAHATHIR HASSAN      |
|        156 | SHARANJIT SINGH      |
|        159 | HARMAN SINGH         |
|        198 | KISAV SHERRY         |
+------------+----------------------+
8 rows in set (0.00 sec)
```

APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB-6 IN PROJECT

1) Select command with the ability of various other inbuilt functions of the max, min and avg applied to purchase_rate , selling_rate.

SELECT MAX(SELLING_RATE), PURCHASE_RATE,SELLING_RATE FROM PURCHASE_TABLE;

SELECT MIN(SELLING_RATE), PURCHASE_RATE,SELLING_RATE FROM PURCHASE_TABLE;

SELECT AVG(SELLING_RATE),AVG(SELLING_RATE) FROM PURCHASE_TABLE;

```
mysql> SELECT MAX(SELLING_RATE),PURCHASE_RATE,SELLING_RATE FROM PURCHASE_TABLE;
+-------------------+---------------+--------------+
| MAX(SELLING_RATE) | PURCHASE_RATE | SELLING_RATE |
+-------------------+---------------+--------------+
|             43000 |         14000 |        18000 |
+-------------------+---------------+--------------+
1 row in set (0.00 sec)

mysql> SELECT MIN(PURCHASE_RATE),PURCHASE_RATE,SELLING_RATE FROM PURCHASE_TABLE;
+--------------------+---------------+--------------+
| MIN(PURCHASE_RATE) | PURCHASE_RATE | SELLING_RATE |
+--------------------+---------------+--------------+
|                250 |         14000 |        18000 |
+--------------------+---------------+--------------+
1 row in set (0.00 sec)

mysql> SELECT AVG(PURCHASE_RATE),AVG(SELLING_RATE) FROM PURCHASE_TABLE;
+--------------------+-------------------+
| AVG(PURCHASE_RATE) | AVG(SELLING_RATE) |
+--------------------+-------------------+
|           12593.75 |          25043.75 |
+--------------------+-------------------+
1 row in set (0.00 sec)
```

2) Select command to implement the add some value to the amount_of_item_buyed.

SELECT (AMOUNT_OF_ITEM_BUYED+23532) AS INCREASED_AMOUNT, AMOUNT_OF_ITEM_BUYED FROM SALE_TABLE;

```
mysql> SELECT (AMOUNT_OF_ITEM_BUYED+23532) AS INCREASED_AMOUNT,AMOUNT_OF_ITEM_BUYED FROM SALE_TABLE;
+-----------------+---------------------+
| INCREASED_AMOUNT | AMOUNT_OF_ITEM_BUYED |
+-----------------+---------------------+
|           43032 |               19500 |
|           65532 |               42000 |
|           73532 |               50000 |
|           63532 |               40000 |
|           58132 |               34600 |
|           55132 |               31600 |
|           54032 |               30500 |
|           24732 |                1200 |
+-----------------+---------------------+
8 rows in set (0.00 sec)
```

3) Select Command to multiply some value to the amount_of_item_buyed in the command.

SELECT(AMOUNT_OF_ITEM_BUYED*2.5)AS MULTIPLIED_AMOUNT,AMOUNT_OF_ITEM_BUYED FROM SALE_TABLE;

```
mysql> SELECT (AMOUNT_OF_ITEM_BUYED*2.5) AS MULTIPLIED_AMOUNT,AMOUNT_OF_ITEM_BUYED FROM SALE_TABLE;
+-----------------+---------------------+
| MULTIPLIED_AMOUNT | AMOUNT_OF_ITEM_BUYED |
+-----------------+---------------------+
|            48750 |               19500 |
|           105000 |               42000 |
|           125000 |               50000 |
|           100000 |               40000 |
|            86500 |               34600 |
|            79000 |               31600 |
|            76250 |               30500 |
|             3000 |                1200 |
+-----------------+---------------------+
8 rows in set (0.00 sec)
```

4) Select Command to subtract some value from the amount_of_item_buyed in the command.

SELECT (AMOUNT_OF_ITEM_BUYED-1234) AS DECREASED_AMOUNT,AMOUNT_OF_ITEM_BUYED FROM SALE_TABLE;

```
mysql> SELECT (AMOUNT_OF_ITEM_BUYED-1234) AS DECREASED_AMOUNT,AMOUNT_OF_ITEM_BUYED FROM SALE_TABLE;
+-----------------+----------------------+
| DECREASED_AMOUNT | AMOUNT_OF_ITEM_BUYED |
+-----------------+----------------------+
|           18266 |                19500 |
|           40766 |                42000 |
|           48766 |                50000 |
|           38766 |                40000 |
|           33366 |                34600 |
|           30366 |                31600 |
|           29266 |                30500 |
|             -34 |                 1200 |
+-----------------+----------------------+
8 rows in set (0.00 sec)
```

5) Select Command to divide some value to the amount_of_item_buyed in the command.

SELECT (AMOUNT_OF_ITEM_BUYED/2) AS HALF_AMOUNT,AMOUNT_OF_ITEM_BUYED FROM SALE_TABLE;

```
mysql> SELECT (AMOUNT_OF_ITEM_BUYED/2) AS HALF_AMOUNT,AMOUNT_OF_ITEM_BUYED FROM SALE_TABLE;
+-------------+----------------------+
| HALF_AMOUNT | AMOUNT_OF_ITEM_BUYED |
+-------------+----------------------+
|        9750 |                19500 |
|       21000 |                42000 |
|       25000 |                50000 |
|       20000 |                40000 |
|       17300 |                34600 |
|       15800 |                31600 |
|       15250 |                30500 |
|         600 |                 1200 |
+-------------+----------------------+
8 rows in set (0.00 sec)
```

6) Select command which uses the like command clause which displays the product name starting from “TRANS”.

SELECT PRODUCT_NAME FROM PURCHASE_TABLE WHERE PRODUCT_NAME LIKE “TRANS%”;

```
mysql> SELECT PRODUCT_NAME FROM PURCHASE_TABLE WHERE PRODUCT_NAME LIKE "TRANS%";
+--------------+
| PRODUCT_NAME |
+--------------+
| TRANSFORMERS |
+--------------+
1 row in set (0.01 sec)
```

7) Select command which displays the batch name which have “BB” in it.

SELECT PURCHASE_RATE,BATCHNAME FROM PURCHASE_TABLE WHERE BATCHNAME LIKE “%BB%”;

```
mysql> SELECT PURCHASE_RATE,BATCHNAME FROM PURCHASE_TABLE WHERE BATCHNAME LIKE "%BB%";
Empty set (0.00 sec)

mysql> SELECT PURCHASE_RATE,BATCHNAME FROM PURCHASE_TABLE WHERE BATCHNAME LIKE "BB%";
Empty set (0.00 sec)
```

8) Select command which displays all the info from purchase_table which had date not having march month.

   SELECT * FROM PURCHASE_TABLE WHERE PURCHASE_DATE NOT LIKE “_____03%”;

```
mysql> SELECT * FROM PURCHASE_TABLE WHERE PURCHASE_DATE NOT LIKE "_____03%";
+------------+-----------------+-----------+----------+---------------+---------------+--------------+-------------+
| PRODUCT_ID | PRODUCT_NAME    | BATCHNAME | QUANTITY | PURCHASE_DATE | PURCHASE_RATE | SELLING_RATE | EXPIRY_DATE |
+------------+-----------------+-----------+----------+---------------+---------------+--------------+-------------+
|        121 | ISOLATORS       | B221      | 250      | 2021-01-26    |         17000 |        40000 | 2023-02-24  |
|        131 | INSULATORS      | B221      | 210      | 2021-01-26    |         11000 |        43000 | 2023-02-24  |
|        132 | PANEL BOARDS    | B221      | 450      | 2022-01-26    |         14500 |        23000 | 2024-09-12  |
|        141 | SHACKLES        | B121      | 140      | 2020-11-26    |         11000 |        20000 | 2025-02-12  |
|        156 | METERS          | B112      | 212      | 2021-01-26    |         14500 |        27850 | 2025-07-13  |
|        159 | CIRCUIT BREAKERS | B121      | 245      | 2022-01-26    |         18500 |        27850 | 2026-01-24  |
|        198 | MCCB            | B142      | 912      | 2020-12-26    |           250 |          650 | 2026-11-25  |
+------------+-----------------+-----------+----------+---------------+---------------+--------------+-------------+
7 rows in set (0.00 sec)
```

9) Select some info from the customer_table which has address not having the address city as “GHAZIABAD”.

SELECT CUSTOMER_NAME,CUSTOMER_ADDRESS FROM CUSTOMER_TABLE WHERE CUSTOMER_ADDRESS NOT LIKE “%GHAZIABAD%”;

```
mysql> SELECT CUSTOMER_NAME,CUSTOMER_ADDRESS FROM CUSTOMER_TABLE WHERE CUSTOMER_ADDRESS NOT LIKE "%GHAZIABAD%";
+----------------------+-----------------------------------+
| CUSTOMER_NAME        | CUSTOMER_ADDRESS                  |
+----------------------+-----------------------------------+
| DHARAMRAJ CHERALATHAN | 12,KONADADERU INDUSTRIES,CHENNAI  |
| RAJESH PANDIT        | 112,SEEMANT SOCIETY,MUMBAI        |
| SHARIYAR KHAN        | 12,JEENAT MAHAL INDUSTRIES,LUCKNOW |
| ZAHIR KHAN           | 10,SHAM INDUSTRIES,JAIPUR         |
| SHARANJIT SINGH      | 12,SINGH ENTERPRISES,AMRITSAR     |
| HARMAN SINGH         | 45,RAKESH ENTERPRISES,HYDERABAD   |
| KISAV SHERRY         | 89,MAHANT ENTERPRISES,AHMEDABAD   |
+----------------------+-----------------------------------+
7 rows in set (0.00 sec)
```

10) Select some info from the warehouse table which do not have the aisle as “3”.

SELECT * FROM WAREHOUSE WHERE AISLE NOT LIKE “%3%”;

```
mysql> SELECT * FROM WAREHOUSE WHERE AISLE NOT LIKE "%3%";
+------------+-------+-------+
| PRODUCT_ID | AISLE | FLOOR |
+------------+-------+-------+
|        112 | 1     | 2     |
|        131 | 4     | 5     |
|        156 | 4     | 2     |
|        159 | 2     | 6     |
|        198 | 4     | 2     |
+------------+-------+-------+
5 rows in set (0.00 sec)
```

APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB-7 IN PROJECT

1) Select the total purchase_rate of different types of the products in the purchase_table.

SELECT COUNT(PURCHASE_RATE) FROM PURCHASE_TABLE;

```
mysql> SELECT COUNT(PURCHASE_RATE) FROM PURCHASE_TABLE;
+----------------------+
| COUNT(PURCHASE_RATE) |
+----------------------+
|                    8 |
+----------------------+
1 row in set (0.00 sec)
```

2) Select the total no. of owners of different types of the products in the inventories.

SELECT COUNT(OWNER) FROM INVENTORIES;

```
mysql> SELECT COUNT(OWNER) FROM INVENTORIES;
+--------------+
| COUNT(OWNER) |
+--------------+
|            3 |
+--------------+
1 row in set (0.01 sec)
```

3) Select the total or the sum of the selling_rate in the purchase_table.

SELECT SUM(SELLING_RATE) FROM PURCHASE_TABLE;

```
mysql> SELECT SUM(SELLING_RATE) FROM PURCHASE_TABLE;
+-------------------+
| SUM(SELLING_RATE) |
+-------------------+
|            200350 |
+-------------------+
1 row in set (0.00 sec)
```

4) Select the total or the sum of the amount_of_item_buyed in the sale_table.

SELECT SUM(AMOUNT_OF_ITEM_BUYED) FROM SALE_TABLE;

```
mysql> SELECT SUM(AMOUNT_OF_ITEM_BUYED) FROM SALE_TABLE;
+---------------------------+
| SUM(AMOUNT_OF_ITEM_BUYED) |
+---------------------------+
|                    249400 |
+---------------------------+
1 row in set (0.00 sec)
```

5) Select Command which finds the avg(selling_rate) from the purchase_table.

SELECT AVG(SELLING_RATE) FROM PURCHASE_TABLE;

```
mysql> SELECT AVG(SELLING_RATE) FROM PURCHASE_TABLE;
+-------------------+
| AVG(SELLING_RATE) |
+-------------------+
|          25043.75 |
+-------------------+
1 row in set (0.00 sec)
```

6) Select the avg of the amount_of_item_buyed in the sale_table.

SELECT AVG(AMOUNT_OF_ITEM_BUYED) FROM SALE_TABLE;

```
mysql> SELECT AVG(AMOUNT_OF_ITEM_BUYED) FROM SALE_TABLE;
+----------------------------+
| AVG(AMOUNT_OF_ITEM_BUYED) |
+----------------------------+
|                      31175 |
+----------------------------+
1 row in set (0.00 sec)
```

7) Select command with the min clause which gives the min value of amount_of_item_buyed and selling_rate.

SELECT MIN(SELLING_RATE) FROM PURCHASE_TABLE;

```
mysql> SELECT MIN(SELLING_RATE) FROM PURCHASE_TABLE;
+--------------------+
| MIN(SELLING_RATE) |
+--------------------+
|                650 |
+--------------------+
1 row in set (0.00 sec)

mysql> SELECT MIN(AMOUNT_OF_ITEM_BUYED) FROM SALE_TABLE;
+----------------------------+
| MIN(AMOUNT_OF_ITEM_BUYED) |
+----------------------------+
|                       1200 |
+----------------------------+
1 row in set (0.00 sec)
```

8) Select command with the max clause which gives the min value of amount_of_item_buyed and selling_rate.

SELECT MAX(AMOUNT_OF_ITEM_BUYED) FROM SALE_TABLE;

```
mysql> SELECT MAX(AMOUNT_OF_ITEM_BUYED) FROM SALE_TABLE;
+----------------------------+
| MAX(AMOUNT_OF_ITEM_BUYED) |
+----------------------------+
|                      50000 |
+----------------------------+
1 row in set (0.00 sec)

mysql> SELECT MAX(SELLING_RATE) FROM PURCHASE_TABLE;
+--------------------+
| MAX(SELLING_RATE) |
+--------------------+
|              43000 |
+--------------------+
1 row in set (0.00 sec)
```

9) Select command with the group by clause which groups the data according to the branch inside the table.

SELECT SNO, COUNTRY FROM EXPORTS_AND_IMPORTS GROUP BY BRANCH;

```
mysql> SELECT SNO,COUNTRY FROM EXPORTS_AND_IMPORTS GROUP BY BRANCH;
+-----+-----------+
| SNO | COUNTRY   |
+-----+-----------+
|   1 | CANADA    |
|   2 | INDIA     |
|   3 | UK        |
|   4 | USA       |
|   5 | AUSTRALIA |
+-----+-----------+
5 rows in set (0.01 sec)
```

APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB-8 IN PROJECT

1) Select command to show the purchase_date from purchase_table.

SELECT PURCHASE_DATE FROM PURCHASE_TABLE;

```
mysql> SELECT PURCHASE_DATE FROM PURCHASE_TABLE;
+---------------+
| PURCHASE_DATE |
+---------------+
| 2021-03-12    |
| 2021-01-26    |
| 2021-01-26    |
| 2022-01-26    |
| 2020-11-26    |
| 2021-01-26    |
| 2022-01-26    |
| 2020-12-26    |
+---------------+
8 rows in set (0.00 sec)
```

2) Select command to show the current date and the system date of the system.

SELECT CURDATE();

SELECT SYSDATE();

```
mysql> SELECT CURDATE();
+------------+
| CURDATE()  |
+------------+
| 2020-11-07 |
+------------+
1 row in set (0.00 sec)

mysql> SELECT SYSDATE();
+---------------------+
| SYSDATE()           |
+---------------------+
| 2020-11-07 21:16:46 |
+---------------------+
1 row in set (0.00 sec)
```

3) Select command with the capability of displaying the date of the purchase_date.

SELECT DATE(PURCHASE_DATE),PURCHASE_DATE, EXPIRY_DATE FROM PURCHASE_TABLE;

```
mysql> SELECT DATE(PURCHASE_DATE),PURCHASE_DATE,EXPIRY_DATE FROM PURCHASE_TABLE;
+--------------------+---------------+-------------+
| DATE(PURCHASE_DATE) | PURCHASE_DATE | EXPIRY_DATE |
+--------------------+---------------+-------------+
| 2021-03-12         | 2021-03-12    | 2022-08-23  |
| 2021-01-26         | 2021-01-26    | 2023-02-24  |
| 2021-01-26         | 2021-01-26    | 2023-02-24  |
| 2022-01-26         | 2022-01-26    | 2024-09-12  |
| 2020-11-26         | 2020-11-26    | 2025-02-12  |
| 2021-01-26         | 2021-01-26    | 2025-07-13  |
| 2022-01-26         | 2022-01-26    | 2026-01-24  |
| 2020-12-26         | 2020-12-26    | 2026-11-25  |
+--------------------+---------------+-------------+
8 rows in set (0.00 sec)
```

4) Select command to display the day of the date from the table.

SELECT DAY(DATE), DATE FROM SALE_TABLE;

```
mysql> SELECT DAY(DATE),DATE FROM SALE_TABLE;
+-----------+---------------------+
| DAY(DATE) | DATE                |
+-----------+---------------------+
|        15 | 2021-04-15 12:30:12 |
|        11 | 2022-05-11 11:30:12 |
|        21 | 2022-09-21 10:30:42 |
|        23 | 2022-11-23 10:30:23 |
|        12 | 2022-11-12 10:23:25 |
|        21 | 2021-05-21 10:23:24 |
|        22 | 2023-12-22 12:10:24 |
|        22 | 2021-11-22 09:30:24 |
+-----------+---------------------+
8 rows in set (0.00 sec)
```

5) Select command to display the dayname of the expiry_date of the stock in purchase_table.

SELECT DAYNAME(EXPIRY_DATE),EXPIRY_DATE,PURCHASE_DATE FROM PURCHASE_TABLE;

```
mysql> SELECT DAYNAME(EXPIRY_DATE),EXPIRY_DATE,PURCHASE_DATE FROM PURCHASE_TABLE;
+----------------------+-------------+---------------+
| DAYNAME(EXPIRY_DATE) | EXPIRY_DATE | PURCHASE_DATE |
+----------------------+-------------+---------------+
| Tuesday              | 2022-08-23  | 2021-03-12    |
| Friday               | 2023-02-24  | 2021-01-26    |
| Friday               | 2023-02-24  | 2021-01-26    |
| Thursday             | 2024-09-12  | 2022-01-26    |
| Wednesday            | 2025-02-12  | 2020-11-26    |
| Sunday               | 2025-07-13  | 2021-01-26    |
| Saturday             | 2026-01-24  | 2022-01-26    |
| Wednesday            | 2026-11-25  | 2020-12-26    |
+----------------------+-------------+---------------+
8 rows in set (0.01 sec)
```

6) Select command to display the week of the day.

SELECT WEEK(DATE) , DATE FROM SALE_TABLE;

```
mysql> SELECT WEEK(DATE),DATE FROM SALE_TABLE;
+------------+---------------------+
| WEEK(DATE) | DATE                |
+------------+---------------------+
|         15 | 2021-04-15 12:30:12 |
|         19 | 2022-05-11 11:30:12 |
|         38 | 2022-09-21 10:30:42 |
|         47 | 2022-11-23 10:30:23 |
|         45 | 2022-11-12 10:23:25 |
|         20 | 2021-05-21 10:23:24 |
|         51 | 2023-12-22 12:10:24 |
|         47 | 2021-11-22 09:30:24 |
+------------+---------------------+
8 rows in set (0.00 sec)
```

7) Select command with the datediff clause which finds the no.of days between purchase_date and expiry_date.

SELECT DATEDIFF(EXPIRY_DATE,PURCHASE_DATE),PURCHASE_DATE,EXPIRY_DATE FROM PURCHASE_TABLE;

```
mysql> SELECT DATEDIFF(EXPIRY_DATE,PURCHASE_DATE),PURCHASE_DATE,EXPIRY_DATE FROM PURCHASE_TABLE;
+-----------------------------------+---------------+-------------+
| DATEDIFF(EXPIRY_DATE,PURCHASE_DATE) | PURCHASE_DATE | EXPIRY_DATE |
+-----------------------------------+---------------+-------------+
|                               529 | 2021-03-12    | 2022-08-23  |
|                               759 | 2021-01-26    | 2023-02-24  |
|                               759 | 2021-01-26    | 2023-02-24  |
|                               960 | 2022-01-26    | 2024-09-12  |
|                              1539 | 2020-11-26    | 2025-02-12  |
|                              1629 | 2021-01-26    | 2025-07-13  |
|                              1459 | 2022-01-26    | 2026-01-24  |
|                              2160 | 2020-12-26    | 2026-11-25  |
+-----------------------------------+---------------+-------------+
8 rows in set (0.00 sec)
```

8) Select command which takes the minutes from the date.

SELECT EXTRACT(MINUTE FROM DATE) FROM SALE_TABLE;

```
mysql> SELECT EXTRACT(MINUTE FROM DATE) FROM SALE_TABLE;
+---------------------------+
| EXTRACT(MINUTE FROM DATE) |
+---------------------------+
|                        30 |
|                        30 |
|                        30 |
|                        30 |
|                        23 |
|                        23 |
|                        10 |
|                        30 |
+---------------------------+
8 rows in set (0.00 sec)
```

9) Select command which displays the time to seconds that are displayed in the date from sale_table.

SELECT TIME_TO_SEC(DATE) , DATE FROM SALE_TABLE;

```
mysql> SELECT TIME_TO_SEC(DATE),DATE FROM SALE_TABLE;
+--------------------+---------------------+
| TIME_TO_SEC(DATE) | DATE                 |
+--------------------+---------------------+
|              45012 | 2021-04-15 12:30:12 |
|              41412 | 2022-05-11 11:30:12 |
|              37842 | 2022-09-21 10:30:42 |
|              37823 | 2022-11-23 10:30:23 |
|              37405 | 2022-11-12 10:23:25 |
|              37404 | 2021-05-21 10:23:24 |
|              43824 | 2023-12-22 12:10:24 |
|              34224 | 2021-11-22 09:30:24 |
+--------------------+---------------------+
8 rows in set (0.00 sec)
```

10) Select command that displays the no. of days displayed in the date from sale_table.

SELECT TIME_TO_SEC(DATE),DATE FROM SALE_TABLE;

```
mysql> SELECT TIME_TO_SEC(DATE),DATE FROM SALE_TABLE;
+--------------------+---------------------+
| TIME_TO_SEC(DATE) | DATE                 |
+--------------------+---------------------+
|              45012 | 2021-04-15 12:30:12 |
|              41412 | 2022-05-11 11:30:12 |
|              37842 | 2022-09-21 10:30:42 |
|              37823 | 2022-11-23 10:30:23 |
|              37405 | 2022-11-12 10:23:25 |
|              37404 | 2021-05-21 10:23:24 |
|              43824 | 2023-12-22 12:10:24 |
|              34224 | 2021-11-22 09:30:24 |
+--------------------+---------------------+
8 rows in set (0.00 sec)
```

11) Select command which displays the concat command that concatenation of product_id and customer_name.

SELECT CONCAT(PRODUCT_ID, " – " , CUSTOMER_NAME) FROM SALE_TABLE;

```
mysql> SELECT CONCAT(PRODUCT_ID," - ",CUSTOMER_NAME) FROM SALE_TABLE;
+---------------------------------------+
| CONCAT(PRODUCT_ID," - ",CUSTOMER_NAME) |
+---------------------------------------+
| 112 - DHARAMRAJ CHERALATHAN           |
| 121 - RAJESH PANDIT                   |
| 131 - SHARIYAR KHAN                   |
| 132 - ZAHIR KHAN                      |
| 141 - MAHATHIR HASSAN                 |
| 156 - SHARANJIT SINGH                 |
| 159 - HARMAN SINGH                    |
| 198 - KISAV SHERRY                    |
+---------------------------------------+
8 rows in set (0.00 sec)
```

12) Select the particular string character from the string and displays the position of that letter as we displays the letter 'E' from customer_name.

SELECT INSTR(CUSTOMER_NAME, 'E') FROM CUSTOMER_TABLE;

```
mysql> SELECT INSTR(CUSTOMER_NAME,'E') FROM CUSTOMER_TABLE;
+--------------------------+
| INSTR(CUSTOMER_NAME,'E') |
+--------------------------+
|                       13 |
|                        4 |
|                        0 |
|                        0 |
|                        0 |
|                        0 |
|                        0 |
|                        9 |
+--------------------------+
8 rows in set (0.00 sec)
```

13) Select command which displays the length of the char string product_name.

SELECT LENGTH(PRODUCT_NAME) FROM PURCHASE_TABLE;

```
mysql> SELECT LENGTH(PRODUCT_NAME) FROM PURCHASE_TABLE;
+----------------------+
| LENGTH(PRODUCT_NAME) |
+----------------------+
|                   12 |
|                    9 |
|                   10 |
|                   12 |
|                    8 |
|                    6 |
|                   16 |
|                    4 |
+----------------------+
8 rows in set (0.00 sec)
```

14) Select command which displays the left part of string from a position in owner's name in the inventories table.

SELECT LPAD(OWNER,10, 'IS OWNER') FROM INVENTORIES;

```
mysql> SELECT LPAD(OWNER,10,'IS OWNER') FROM INVENTORIES;
+---------------------------+
| LPAD(OWNER,10,'IS OWNER') |
+---------------------------+
| RADHE SHYA                |
| SAJJAN SIN                |
| RANJEET SI                |
+---------------------------+
3 rows in set (0.01 sec)
```

15) Select command which replaces a letter in the Batchname in the purchase_table.

SELECT REPLACE(BATCHNAME, ‘A’ , ‘C’) FROM PURCHASE_TABLE;

```
mysql> SELECT REPLACE(BATCHNAME,'A','C') FROM PURCHASE_TABLE;
+----------------------------+
| REPLACE(BATCHNAME,'A','C') |
+----------------------------+
| C113                       |
| B221                       |
| B221                       |
| B221                       |
| B121                       |
| B112                       |
| B121                       |
| B142                       |
+----------------------------+
8 rows in set (0.00 sec)
```

16) Select command which displays the lower case letters in the product_name in the table.

SELECT LOWER(PRODUCT_NAME) AS LOWER FROM PURCHASE_TABLE;

```
mysql> SELECT LOWER(PRODUCT_NAME) AS LOWER FROM PURCHASE_TABLE;
+------------------+
| LOWER            |
+------------------+
| transformers     |
| isolators        |
| insulators       |
| panel boards     |
| shackles         |
| meters           |
| circuit breakers |
| mccb             |
+------------------+
8 rows in set (0.00 sec)
```

17) Select Command which displays the ceil value of the result of division of selling_rate by some value.

SELECT CEIL(SELLING_RATE/4.79) FROM PURCHASE_TABLE;

```
mysql> SELECT CEIL(SELLING_RATE/4.79) FROM PURCHASE_TABLE;
+-------------------------+
| CEIL(SELLING_RATE/4.79) |
+-------------------------+
|                    3758 |
|                    8351 |
|                    8978 |
|                    4802 |
|                    4176 |
|                    5815 |
|                    5815 |
|                     136 |
+-------------------------+
8 rows in set (0.00 sec)
```

18) Select command which displays the cos value of the product_id.

SELECT COS(PRODUCT_ID) FROM PURCHASE_TABLE;

```
mysql> SELECT COS(PRODUCT_ID) FROM PURCHASE_TABLE;
+----------------------+
| COS(PRODUCT_ID)      |
+----------------------+
|   0.4559691044442761 |
| -0.04866360920015389 |
|   0.5842088171092893 |
|   0.9985900724399912 |
|  -0.9317223617435201 |
|  0.47165229356133864 |
| -0.34249477911590703 |
|  -0.9968285949694307 |
+----------------------+
8 rows in set (0.00 sec)
```

19) Select command that displays the modulus value if we divide the product_id by some value.

SELECT MOD(PRODUCT_ID,5) FROM PURCHASE_TABLE;

```
mysql> SELECT MOD(PRODUCT_ID,5) FROM PURCHASE_TABLE;
+--------------------+
| MOD(PRODUCT_ID,5) |
+--------------------+
|                  2 |
|                  1 |
|                  1 |
|                  2 |
|                  1 |
|                  1 |
|                  4 |
|                  3 |
+--------------------+
8 rows in set (0.00 sec)
```

20) Select command that displays the sqrt value of the purchase_rate from the purchase_table.

SELECT SQRT(PURCHASE_RATE) FROM PURCHASE_TABLE;

```
mysql> SELECT SQRT(PURCHASE_RATE) FROM PURCHASE_TABLE;
+----------------------+
| SQRT(PURCHASE_RATE) |
+----------------------+
|  118.32159566199232 |
|  130.38404810405297 |
|  104.88088481701516 |
|  120.41594578792295 |
|  104.88088481701516 |
|  120.41594578792295 |
|  136.01470508735443 |
|  15.811388300841896 |
+----------------------+
8 rows in set (0.00 sec)
```

21) Select command which displays the truncated value upto a certain specific value.

SELECT TRUNCATE(PRODUCT_ID/7.76688,4),PRODUCT_ID FROM CUSTOMER_TABLE;

```
mysql> SELECT TRUNCATE(PRODUCT_ID/7.76688,4),PRODUCT_ID FROM CUSTOMER_TABLE;
+--------------------------------+------------+
| TRUNCATE(PRODUCT_ID/7.76688,4) | PRODUCT_ID |
+--------------------------------+------------+
|                         14.4202 |        112 |
|                         15.5789 |        121 |
|                         16.8664 |        131 |
|                         16.9952 |        132 |
|                         18.1540 |        141 |
|                         20.0852 |        156 |
|                         20.4715 |        159 |
|                         25.4928 |        198 |
+--------------------------------+------------+
8 rows in set (0.00 sec)
```

22) Select command displays the greatest selling_rate, purchase_rate from the purchase_table.

SELECT GREATEST(SELLING_RATE,PURCHASE_RATE) FROM PURCHASE_TABLE;

```
mysql> SELECT GREATEST(SELLING_RATE,PURCHASE_RATE) FROM PURCHASE_TABLE;
+--------------------------------------+
| GREATEST(SELLING_RATE,PURCHASE_RATE) |
+--------------------------------------+
|                                18000 |
|                                40000 |
|                                43000 |
|                                23000 |
|                                20000 |
|                                27850 |
|                                27850 |
|                                  650 |
+--------------------------------------+
8 rows in set (0.00 sec)
```

APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB-9 IN PROJECT

1) We have to select customer_name from the customer_table where seller_address has “U” in it’s spellings by selecting the product_id as a common thing inside both of the tables.

SELECT CUSTOMER_NAME FROM CUSTOMER_TABLE WHERE PRODUCT_ID IN (SELECT PRODUCT_ID FROM SELLER_TABLE WHERE SELLER_ADDRESS LIKE “%U%”);

```
mysql> SELECT CUSTOMER_NAME FROM CUSTOMER_TABLE WHERE PRODUCT_ID IN (SELECT PRODUCT_ID FROM SELLER_TABLE WHERE SELLER_ADDRESS LIKE "%U%");
+----------------------+
| CUSTOMER_NAME        |
+----------------------+
| DHARAMRAJ CHERALATHAN |
| RAJESH PANDIT        |
| ZAHIR KHAN           |
| MAHATHIR HASSAN      |
| HARMAN SINGH         |
| KISAV SHERRY         |
+----------------------+
6 rows in set (0.00 sec)
```

2) We have to select seller_name from the seller_table where customer_name is “Rajesh Pandit” by selecting the product_id as a common thing inside both of the tables.

SELECT SELLER_NAME FROM SELLER_TABLE WHERE PRODUCT_ID IN (SELECT PRODUCT_ID FROM SALE_TABLE WHERE CUSTOMER_NAME = “RAJESH PANDIT”);

```
mysql> SELECT SELLER_NAME FROM SELLER_TABLE WHERE PRODUCT_ID IN (SELECT PRODUCT_ID FROM SALE_TABLE WHERE CUSTOMER_NAME="RAJESH PANDIT");
+-------------------+
| SELLER_NAME       |
+-------------------+
| RASAM ENTERPRISES |
+-------------------+
1 row in set (0.00 sec)
```

3) We have to select quantity from the sale_item_table where selling_rate is greater than 34588 by selecting the product_name as a common thing inside both of the tables.

SELECT QUANTITY FROM SALE_ITEM_TABLE WHERE PRODUCT_NAME IN (SELECT PRODUCT_NAME IN (SELECT PRODUCT_NAME FROM PURCHASE_TABLE WHERE SELLING_RATE >34588);

```
mysql> SELECT QUANTITY FROM SALE_ITEM_TABLE WHERE PRODUCT_NAME IN (SELECT PRODUCT_NAME FROM PURCHASE_TABLE WHERE SELLING_RATE>34588);
+----------+
| QUANTITY |
+----------+
| 125      |
| 160      |
+----------+
2 rows in set (0.00 sec)
```

4) We have to select seller_conatct,seller_address from the seller_table where amount_of_item_buyed is greater than 23567 by selecting the product_id as a common thing inside both of the tables.

SELECT SELLER_CONTACT,SELLER_ADDRESS FROM SELLER_TABLE WHERE PRODUCT_ID IN (SELECT PRODUCT_ID FROM SALE_TABLE WHERE AMOUNT_OF_ITEM_BUYED>23567);

```
mysql> SELECT SELLER_CONTACT,SELLER_ADDRESS FROM SELLER_TABLE WHERE PRODUCT_ID IN (SELECT PRODUCT_ID FROM SALE_TABLE WHERE AMOUNT_OF_ITEM_BUYED>23567);
+---------------+------------------------------+
| SELLER_CONTACT | SELLER_ADDRESS               |
+---------------+------------------------------+
| 0134-5676784   | 112,SHALINI INDUSTRIES,MUMBAI  |
| 0154-5633784   | 12,BRAHAMAAND POINT,BHOPAL     |
| 0254-5623724   | 189,FOCAL INDUSTRIES,LUDHIANA  |
| 0184-5143724   | 234,RADHE INDUSTRIES,NEW DELHI |
| 0175-5233774   | 1123,FOCAL POINT,PATIALA       |
| 0172-4233174   | 23,RESHAM INDUSTRIES,KOLKATA   |
+---------------+------------------------------+
6 rows in set (0.00 sec)
```

5) Select customer_contact from customer_table where aisle=4 or floor=2 by selecting the product_id as it is common to both of the tables.

SELECT CUSTOMER_CONTACT FROM CUSTOMER_TABLE WHERE PRODUCT_ID IN (SELECT PRODUCT_ID FROM WAREHOUSE WHERE AISLE= ‘4’ OR FLOOR = ‘2’);

```
mysql> SELECT CUSTOMER_CONTACT FROM CUSTOMER_TABLE WHERE PRODUCT_ID IN (SELECT PRODUCT_ID FROM WAREHOUSE WHERE AISLE='4' OR FLOOR='2');
+------------------+
| CUSTOMER_CONTACT |
+------------------+
| 67896-44445      |
| 71248-45678      |
| 68148-56789      |
| 87987-32466      |
| 86821-75178      |
+------------------+
5 rows in set (0.00 sec)
```

6) Select the purchase_rate from purchase_table where customer_contact has 78 in it’s contact no. by selecting product_id as it is common to both the tables.

SELECT PURCHASE_RATE FROM PURCHASE_TABLE WHERE PRODUCT_ID IN (SELECT PRODUCT_ID FROM CUSTOMER_TABLE WHERE CUSTOMER_CONTACT LIKE “%78%”);

```
mysql> SELECT PURCHASE_RATE FROM PURCHASE_TABLE WHERE PRODUCT_ID IN (SELECT PRODUCT_ID FROM CUSTOMER_TABLE WHERE CUSTOMER_CONTACT LIKE "%78%");
+---------------+
| PURCHASE_RATE |
+---------------+
|         14000 |
|         17000 |
|         11000 |
|         14500 |
|         11000 |
|           250 |
+---------------+
6 rows in set (0.00 sec)
```

7) Select the travel_agency_name from travel_agency where customer_name has ‘R’ in it’s customer_name by selecting product_id as it is common to both the tables.

SELECT TRAVEL_AGENCY_NAME FROM TRAVEL_AGENCY WHERE PRODUCT_ID IN (SELECT PRODUCT_ID FROM CUSTOMER_TABLE WHERE CUSTOMER_NAME LIKE “%R%”);

```
mysql> SELECT TRAVEL_AGENCY_NAME FROM TRAVEL_AGENCY WHERE PRODUCT_ID IN (SELECT PRODUCT_ID FROM CUSTOMER_TABLE WHERE CUSTOMER_NAME LIKE "%R%");
+--------------------+
| TRAVEL_AGENCY_NAME |
+--------------------+
| SS TRAVELS         |
| JS TRAVELS         |
| GS TRAVELS         |
| MARSHALL TRAVELS   |
| SANDHU TRAVELS     |
| RAJDEV TRAVELS     |
| SHARMA TRAVELS     |
| GREWAL TRAVELS     |
+--------------------+
8 rows in set (0.00 sec)
```

8) Select the cost_of_travel from travel_agency where amount_of_item_buyed >35000 by selecting product_id as it is common to both the tables.

SELECT COST_OF_TRAVEL FROM TRAVEL_AGENCY WHERE PRODUCT_ID IN (SELECT PRODUCT_ID FROM SALE_TABLE WHERE AMOUNT_OF_ITEM_BUYED>35000);

```
mysql> SELECT COST_OF_TRAVEL FROM TRAVEL_AGENCY WHERE PRODUCT_ID IN (SELECT PRODUCT_ID FROM SALE_TABLE WHERE AMOUNT_OF_ITEM_BUYED>35000);
+----------------+
| COST_OF_TRAVEL |
+----------------+
| 8000           |
| 9000           |
| 5600           |
+----------------+
3 rows in set (0.00 sec)
```

APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB-10 IN PROJECT

1) Select Distinct Aisle from warehouse join inventories join travel_agency as we join all the tables so specific info is greatly available with us.

SELECT DISTINCT(AISLE) FROM WAREHOUSE JOIN INVENTORIES JOIN TRAVEL_AGENCY;

```
mysql> SELECT DISTINCT(AISLE) FROM WAREHOUSE JOIN INVENTORIES JOIN TRAVEL_AGENCY;
+-------+
| AISLE |
+-------+
| 1     |
| 3     |
| 4     |
| 13    |
| 2     |
+-------+
5 rows in set (0.00 sec)
```

2) We want to select distinct branch from the join or the cartesian product of inventories and exports_and_imports.

SELECT DISTINCT(BRANCH) FROM INVENTORIES JOIN EXPORTS_AND_IMPORTS;

```
mysql> SELECT DISTINCT(BRANCH) FROM INVENTORIES JOIN EXPORTS_AND_IMPORTS ;
+-----------+
| BRANCH    |
+-----------+
| VANCOUVER |
| NEW DELHI |
| LONDON    |
| NEW YORK  |
| BRISBANE  |
+-----------+
5 rows in set (0.00 sec)
```

3) If we want to select distinct owner country from the join or the cartesian product of warehouse and exports_and_imports.

SELECT DISTINCT OWNER , COUNTRY FROM INVENTORIES JOIN EXPORTS_AND_IMPORTS JOIN WAREHOUSE;

```
mysql> SELECT DISTINCT OWNER,COUNTRY FROM INVENTORIES JOIN EXPORTS_AND_IMPORTS JOIN WAREHOUSE;
+--------------------+-----------+
| OWNER              | COUNTRY   |
+--------------------+-----------+
| RADHE SHYAM TIWARI | CANADA    |
| SAJJAN SINGH       | CANADA    |
| RANJEET SINGH      | CANADA    |
| RADHE SHYAM TIWARI | INDIA     |
| SAJJAN SINGH       | INDIA     |
| RANJEET SINGH      | INDIA     |
| RADHE SHYAM TIWARI | UK        |
| SAJJAN SINGH       | UK        |
| RANJEET SINGH      | UK        |
| RADHE SHYAM TIWARI | USA       |
| SAJJAN SINGH       | USA       |
| RANJEET SINGH      | USA       |
| RADHE SHYAM TIWARI | AUSTRALIA |
| SAJJAN SINGH       | AUSTRALIA |
| RANJEET SINGH      | AUSTRALIA |
+--------------------+-----------+
15 rows in set (0.00 sec)
```

4) If we want to select distinct amount_of_item_buyed from the join or the cartesian product of warehouse , sale_table, travel_agency.

SELECT DISTINCT AMOUNT_OF_BUYED FROM WAREHOUSE JOIN SALE_TABLE JOIN TRAVEL_AGENCY;

```
mysql> SELECT DISTINCT AMOUNT_OF_ITEM_BUYED FROM WAREHOUSE JOIN SALE_TABLE JOIN TRAVEL_AGENCY;
+----------------------+
| AMOUNT_OF_ITEM_BUYED |
+----------------------+
|                19500 |
|                42000 |
|                50000 |
|                40000 |
|                34600 |
|                31600 |
|                30500 |
|                 1200 |
+----------------------+
8 rows in set (0.00 sec)
```

5) Select distinct seller_name from the join or the cartesian product of sale_table,inventories,customer_table and the seller_table.

SELECT DISTINCT(SELLER_NAME) FROM SALE_TABLE JOIN INVENTORIES JOIN CUSTOMER_TABLE JOIN SELLER_TABLE;

```
mysql> SELECT DISTINCT(SELLER_NAME) FROM SALE_TABLE JOIN INVENTORIES JOIN CUSTOMER_TABLE JOIN SELLER_TABLE;
+-----------------------+
| SELLER_NAME           |
+-----------------------+
| SAGAR ENTERPRISES     |
| RASAM ENTERPRISES     |
| DILAWAR ENTERPRISES   |
| MAHAKSHAY ENTERPRISES |
| RADHE ENTERPRISES     |
| KHALSA ENTERPRISES    |
| DHARMA ENTERPRISES    |
| CHANDAR ENTERPRISES   |
+-----------------------+
8 rows in set (0.00 sec)
```

6) Select the distinct product_name from the join of exports_and_imports , purchase_table , seller_table, customer_table.

SELECT DISTINCT PRODUCT_NAME FROM EXPORTS_AND_IMPORTS JOIN PURCHASE_TABLE JOIN SELLER_TABLE JOIN CUSTOMER_TABLE;

```
mysql> SELECT DISTINCT(PRODUCT_NAME) FROM EXPORTS_AND_IMPORTS JOIN PURCHASE_TABLE JOIN SELLER_TABLE JOIN CUSTOMER_TABLE;
+------------------+
| PRODUCT_NAME     |
+------------------+
| TRANSFORMERS     |
| ISOLATORS        |
| INSULATORS       |
| PANEL BOARDS     |
| SHACKLES         |
| METERS           |
| CIRCUIT BREAKERS |
| MCCB             |
+------------------+
8 rows in set (0.00 sec)
```

7) Select the distinct batchname from the cartesian product or the join of the purchase_table, seller_table , customer_table where customer_name has ‘A’ in it’s letter in the spellings.

SELECT DISTINCT BATCHNAME FROM CUSTOMER_TABLE JOIN PURCHASE_TABLE JOIN SELLER_TABLE WHERE CUSTOMER_NAME LIKE “%A%”;

```
mysql> SELECT DISTINCT(BATCHNAME) FROM CUSTOMER_TABLE JOIN PURCHASE_TABLE JOIN SELLER_TABLE WHERE CUSTOMER_NAME LIKE "%A%";
+-----------+
| BATCHNAME |
+-----------+
| A113      |
| B221      |
| B121      |
| B112      |
| B142      |
+-----------+
5 rows in set (0.00 sec)
```

8) Select distinct branch from sale_table join sale_item_table and exports_and_imports where date is having month as April.

SELECT DISTINCT BRANCH FROM SALE_TABLE JOIN SALE_ITEM_TABLE JOIN EXPORTS_AND_IMPORTS WHERE DATE LIKE "_____04%";

```
mysql> SELECT DISTINCT(BRANCH) FROM SALE_TABLE JOIN SALE_ITEM_TABLE JOIN EXPORTS_AND_IMPORTS WHERE DATE LIKE "_____04%";
+-----------+
| BRANCH    |
+-----------+
| VANCOUVER |
| NEW DELHI |
| LONDON    |
| NEW YORK  |
| BRISBANE  |
+-----------+
5 rows in set (0.00 sec)
```

APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB-11 IN PROJECT

1) If we want to create index on batchname in purchase_table we use.

CREATE INDEX I1 ON PURCHASE_TABLE(BATCHNAME);

SHOW INDEX FROM PURCHASE_TABLE;

```
mysql> CREATE INDEX I1 ON PURCHASE_TABLE(BATCHNAME);
Query OK, 0 rows affected (0.15 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> SHOW INDEX FROM PURCHASE_TABLE;
+---------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------
+------------+
| Table         | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible
| Expression |
+---------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------
+------------+
| purchase_table |          0 | PRIMARY  |            1 | PRODUCT_ID  | A         |           8 |     NULL |   NULL |      | BTREE      |         |               | YES
| NULL       |
| purchase_table |          1 | I1       |            1 | BATCHNAME   | A         |           5 |     NULL |   NULL | YES  | BTREE      |         |               | YES
| NULL       |
+---------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------
+------------+
2 rows in set (0.02 sec)
```

2) If we want to create index on seller_address in seller_table

CREATE INDEX I2 ON SELLER_TABLE(SELLER_ADDRESS);

SHOW INDEX FROM SELLER_TABLE;

```
mysql> CREATE INDEX I2 ON SELLER_TABLE(SELLER_ADDRESS);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SHOW INDEX FROM SELLER_TABLE;
+--------------+------------+-----------+--------------+----------------+-----------+-------------+----------+--------+------+------------+---------+---------------+-------
--+------------+
| Table        | Non_unique | Key_name  | Seq_in_index | Column_name    | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visibl
e | Expression |
+--------------+------------+-----------+--------------+----------------+-----------+-------------+----------+--------+------+------------+---------+---------------+-------
--+------------+
| seller_table |          1 | FK_SELLER |            1 | PRODUCT_ID     | A         |           8 |     NULL |   NULL | YES  | BTREE      |         |               | YES
  | NULL       |
| seller_table |          1 | I2        |            1 | SELLER_ADDRESS | A         |           8 |     NULL |   NULL | YES  | BTREE      |         |               | YES
  | NULL       |
+--------------+------------+-----------+--------------+----------------+-----------+-------------+----------+--------+------+------------+---------+---------------+-------
--+------------+
2 rows in set (0.01 sec)
```

3) If we want to form a composite index in the table customer_table.

CREATE INDEX I3 ON CUSTOMER_TABLE(CUSTOMER_NAME,CUSTOMER_CONTACT);

SHOW INDEX FROM CUSTOMER_TABLE;

```
mysql> CREATE INDEX I3 ON CUSTOMER_TABLE(CUSTOMER_NAME,CUSTOMER_CONTACT);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SHOW INDEX FROM CUSTOMER_TABLE;
+----------------+------------+------------+--------------+------------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---
-------+------------+
| Table          | Non_unique | Key_name   | Seq_in_index | Column_name      | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | V
isible | Expression |
+----------------+------------+------------+--------------+------------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---
-------+------------+
| customer_table |          0 | PRODUCT_ID |            1 | PRODUCT_ID       | A         |           8 |     NULL |   NULL | YES  | BTREE      |         |               | Y
ES     | NULL       |
| customer_table |          1 | I3         |            1 | CUSTOMER_NAME    | A         |           8 |     NULL |   NULL |      | BTREE      |         |               | Y
ES     | NULL       |
| customer_table |          1 | I3         |            2 | CUSTOMER_CONTACT | A         |           8 |     NULL |   NULL |      | BTREE      |         |               | Y
ES     | NULL       |
+----------------+------------+------------+--------------+------------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---
-------+------------+
3 rows in set (0.01 sec)
```

4) If we want to form unique index on sale_table on product_id.

CREATE UNIQUE INDEX I4 ON SALE_TABLE(PRODUCT_ID);

SHOW INDEX FROM SALE_TABLE;

```
mysql> CREATE UNIQUE INDEX I4 ON SALE_TABLE(PRODUCT_ID);
Query OK, 0 rows affected, 1 warning (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 1

mysql> SHOW INDEX FROM SALE_TABLE;
+------------+------------+------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+-
-----------+
| Table      | Non_unique | Key_name   | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible |
Expression |
+------------+------------+------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+-
-----------+
| sale_table |          0 | PRODUCT_ID |            1 | PRODUCT_ID  | A         |           8 |     NULL |   NULL | YES  | BTREE      |         |               | YES     |
NULL       |
| sale_table |          0 | I4         |            1 | PRODUCT_ID  | A         |           8 |     NULL |   NULL | YES  | BTREE      |         |               | YES     |
NULL       |
+------------+------------+------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+-
-----------+
2 rows in set (0.01 sec)
```

5) Now we will use the drop command to drop the table.

DROP INDEX I1 ON PURCHASE_TABLE;

SHOW INDEX FROM PURCHASE_TABLE;

```
mysql> DROP INDEX I1 ON PURCHASE_TABLE;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SHOW INDEX FROM PURCHASE_TABLE;
+---------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------
+------------+
| Table         | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible
| Expression |
+---------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------
+------------+
| purchase_table |          0 | PRIMARY  |            1 | PRODUCT_ID  | A         |           8 |     NULL |   NULL |      | BTREE      |         |               | YES
| NULL       |
+---------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------
+------------+
1 row in set (0.01 sec)
```

6) Create a view that only shows us the product_name , quantity from the sale_item_table with some condition in it on amount_of_cost>23455;

CREATE VIEW V1 AS SELECT PRODUCT_NAME,QUANTITY FROM SALE_ITEM_TABLE WHERE AMOUNT_OF_COST>23455;

SELECT * FROM V1;

```
mysql> CREATE VIEW V1 AS SELECT PRODUCT_NAME,QUANTITY FROM SALE_ITEM_TABLE WHERE AMOUNT_OF_COST>23455;
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT * FROM V1;
+------------------+----------+
| PRODUCT_NAME     | QUANTITY |
+------------------+----------+
| ISOLATORS        | 125      |
| INSULATORS       | 160      |
| PANEL BOARDS     | 300      |
| SHACKLES         | 65       |
| METERS           | 165      |
| CIRCUIT BREAKERS | 200      |
+------------------+----------+
6 rows in set (0.01 sec)
```

7) Create a view on exports_and_imports table that selects all of the things inside the table.

CREATE VIEW V2 AS SELECT * FROM EXPORTS_AND_IMPORTS;

SELECT * FROM V2;

```
mysql> CREATE VIEW V2 AS SELECT * FROM EXPORTS_AND_IMPORTS;
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT * FROM V2;
+-----+-----------+-----------+
| SNO | BRANCH    | COUNTRY   |
+-----+-----------+-----------+
|   1 | VANCOUVER | CANADA    |
|   2 | NEW DELHI | INDIA     |
|   3 | LONDON    | UK        |
|   4 | NEW YORK  | USA       |
|   5 | BRISBANE  | AUSTRALIA |
+-----+-----------+-----------+
5 rows in set (0.00 sec)
```

8) Insert command will be used to insert the values in the view v2.

CREATE VIEW V3 AS SELECT * FROM WAREHOUSE;

SELECT * FROM V3;

INSERT INTO V3 VALUES(177,"8","5");

SELECT * FROM V3;

```
mysql> SELECT * FROM V3;
+------------+-------+-------+
| PRODUCT_ID | AISLE | FLOOR |
+------------+-------+-------+
|        112 | 1     | 2     |
|        121 | 3     | 1     |
|        131 | 4     | 5     |
|        132 | 13    | 2     |
|        141 | 3     | 4     |
|        156 | 4     | 2     |
|        159 | 2     | 6     |
|        198 | 4     | 2     |
+------------+-------+-------+
8 rows in set (0.00 sec)

mysql> INSERT INTO V3 VALUES(177,"8","5");
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM V3;
+------------+-------+-------+
| PRODUCT_ID | AISLE | FLOOR |
+------------+-------+-------+
|        112 | 1     | 2     |
|        121 | 3     | 1     |
|        131 | 4     | 5     |
|        132 | 13    | 2     |
|        141 | 3     | 4     |
|        156 | 4     | 2     |
|        159 | 2     | 6     |
|        177 | 8     | 5     |
|        198 | 4     | 2     |
+------------+-------+-------+
9 rows in set (0.00 sec)
```

9) Update command will be used to change the aisle of the product_id = 198.

UPDATE V3 SET AISLE="7" WHERE PRODUCT_ID = 198;

SELECT * FROM V3;

```
mysql> SELECT * FROM V3;
+------------+-------+-------+
| PRODUCT_ID | AISLE | FLOOR |
+------------+-------+-------+
|        112 | 1     | 2     |
|        121 | 3     | 1     |
|        131 | 4     | 5     |
|        132 | 13    | 2     |
|        141 | 3     | 4     |
|        156 | 4     | 2     |
|        159 | 2     | 6     |
|        177 | 8     | 5     |
|        198 | 4     | 2     |
+------------+-------+-------+
9 rows in set (0.00 sec)

mysql> UPDATE V3 SET AISLE="7" WHERE PRODUCT_ID = 198;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM V3;
+------------+-------+-------+
| PRODUCT_ID | AISLE | FLOOR |
+------------+-------+-------+
|        112 | 1     | 2     |
|        121 | 3     | 1     |
|        131 | 4     | 5     |
|        132 | 13    | 2     |
|        141 | 3     | 4     |
|        156 | 4     | 2     |
|        159 | 2     | 6     |
|        177 | 8     | 5     |
|        198 | 7     | 2     |
+------------+-------+-------+
9 rows in set (0.00 sec)
```

10) Drop Command will be used to delete the view v2 from the database.

DROP VIEW V2;

SELECT * FROM V2;

```
mysql> DROP VIEW V2;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT * FROM V2;
ERROR 1146 (42S02): Table 'inventory.v2' doesn't exist
```

REFERENCES

- https://www.camcode.com/asset-tags/what-is-an-inventory-management-system/
- https://www.unleashedsoftware.com/inventory-management-guide/inventory-management-systems
- https://en.wikipedia.org/wiki/Inventory_management_software
- https://www.tradegecko.com/inventory-management
- https://www.selecthub.com/inventory-management/types-of-inventory-management-systems/
- https://www.assetinfinity.com/blog/inventory-management-system-objectives

*** THANK YOU ***