

**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

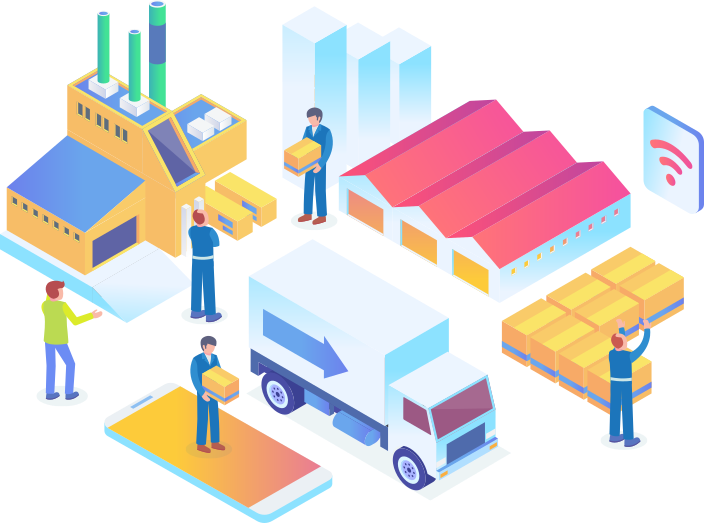
*19124018*

*IT(G-1)*

AYAN GUPTA

*19124019*

*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**  |  |  |  | | --- | --- | --- | | S.NO | TITLE OF TOPIC | PAGE NO. | | 1. | **DESCRIPTION OF PROJECT AND PROBLEM TO SOLVE** | **8-9** | | 2. | **DESCRIPTIVE FUNCTIONALITIES AND SCHEMA DESIGN** | **10-13** | | 3. | **APPLICATIONS OF THE SQL QUERIES AND COMMANDS DESCRIBING ABOUT ALL THE TABLES** | **14-23** | | 4. | **APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB -1** | **22-26** | | 5. | **APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB -2** | **22-26** | | 6. | **APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB -3** | **27-30** | | 7. | **APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB -4** | **31-35** | | 8. | **APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB -5** | **36-39** | | 9. | **APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB -6** | **40-44** | | 10. | **APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB -7** | **45-48** | | 11. | **APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB -8** | **49-61** | | 12. | **APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB -9** | **62-64** | | 13. | **APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB -10** | **65-68** | | 14. | **APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB -11 AND REFERENCES** | **69-76** | |
|  |
| ***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](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/Ontology_(computer_science)).

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***

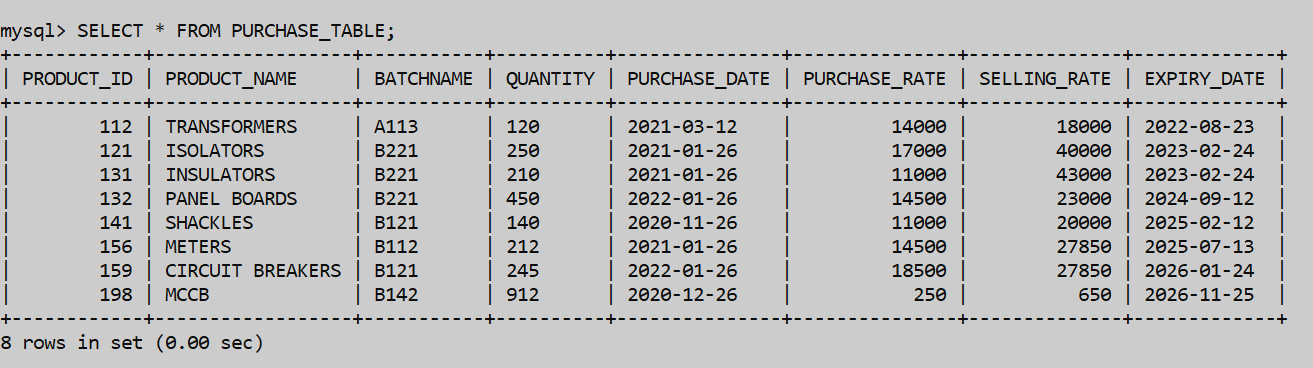
# **APPLICATIONS OF THE SQL QUERIES AND COMMANDS DESCRIBING ABOUT ALL THE TABLES USED.**

# 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**

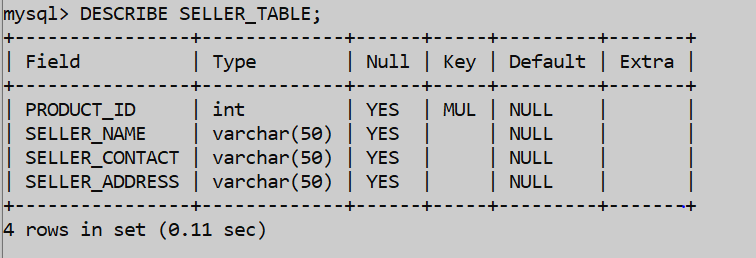
# DESCRIBE PURCHASE\_TABLE;

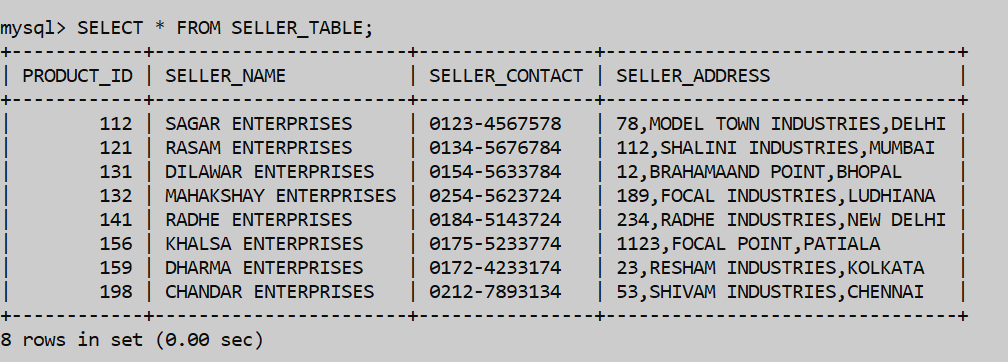
# 



* TABLE NAME – **SELLER\_TABLE**

DESCRIBE SELLER\_TABLE;

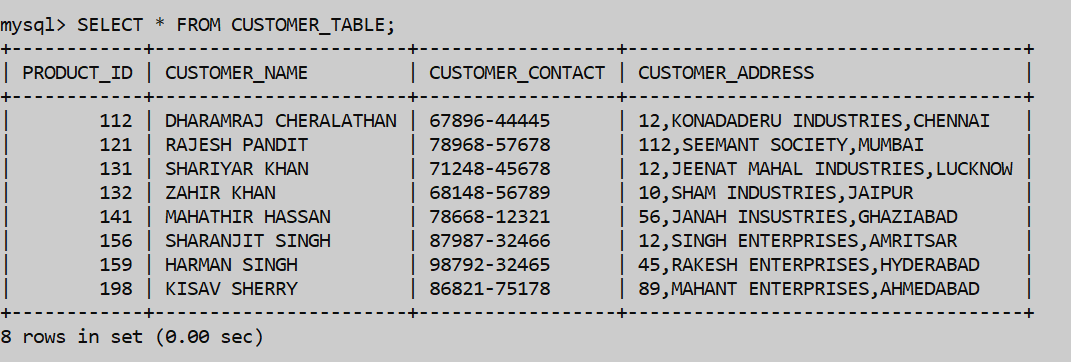




* TABLE NAME – **CUSTOMER\_TABLE**

DESCRIBE CUSTOMER\_TABLE;

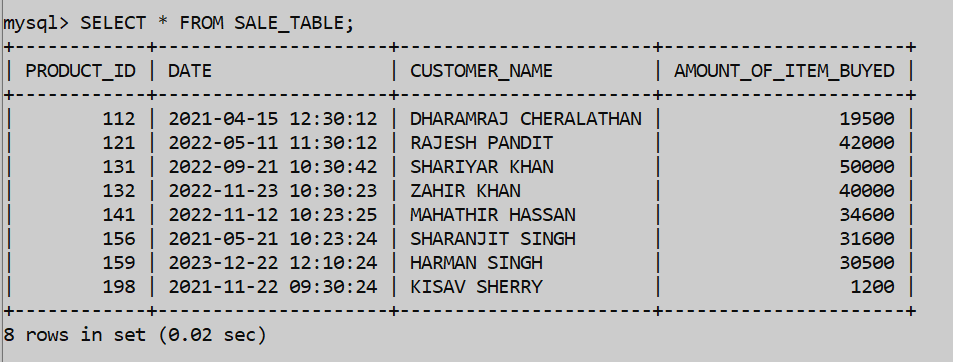




* TABLE NAME – **SALE\_TABLE**

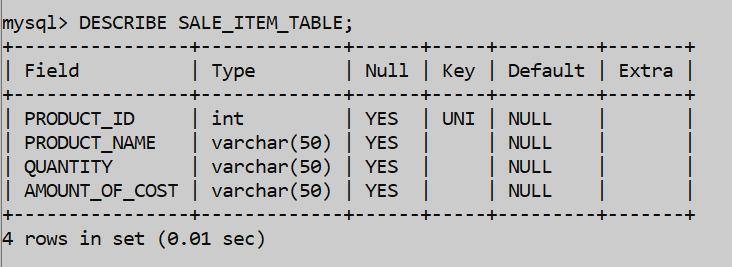
DESCRIBE SALE\_TABLE;

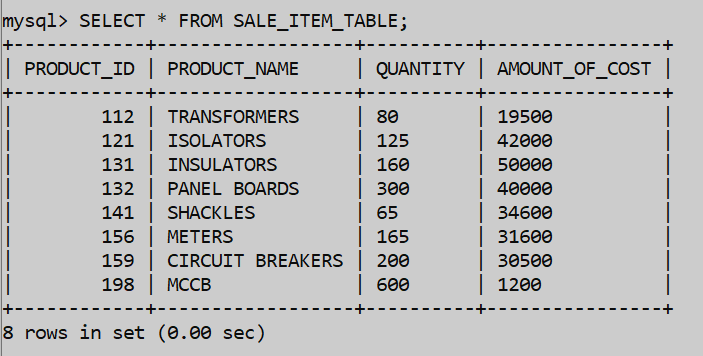




* TABLE NAME – **SALE\_ITEM\_TABLE**

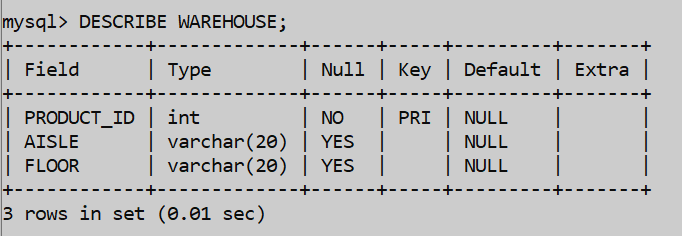
DESCRIBE SALE\_ITEM\_TABLE;

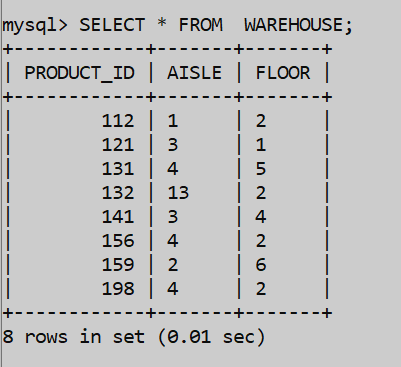




* TABLE NAME – **WAREHOUSE**

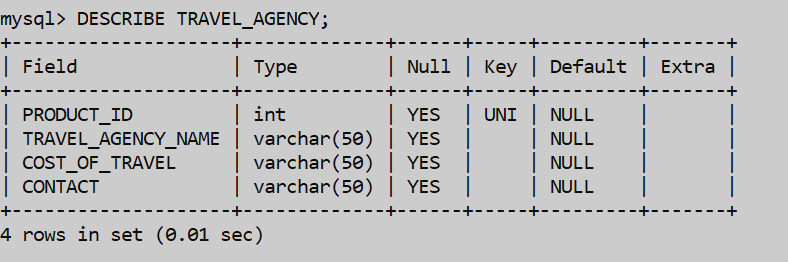
DESCRIBE WAREHOUSE;

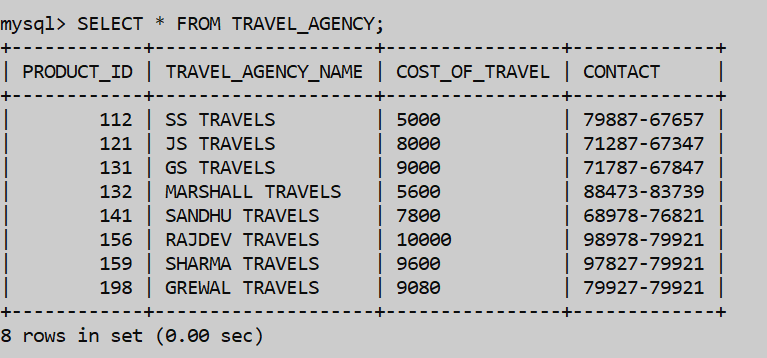




* TABLE NAME – **TRAVEL\_AGENCY**

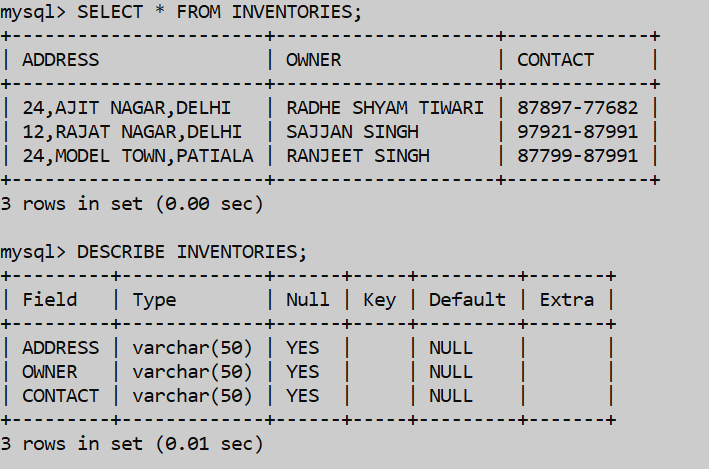
DESCRIBE TRAVEL\_AGENCY;





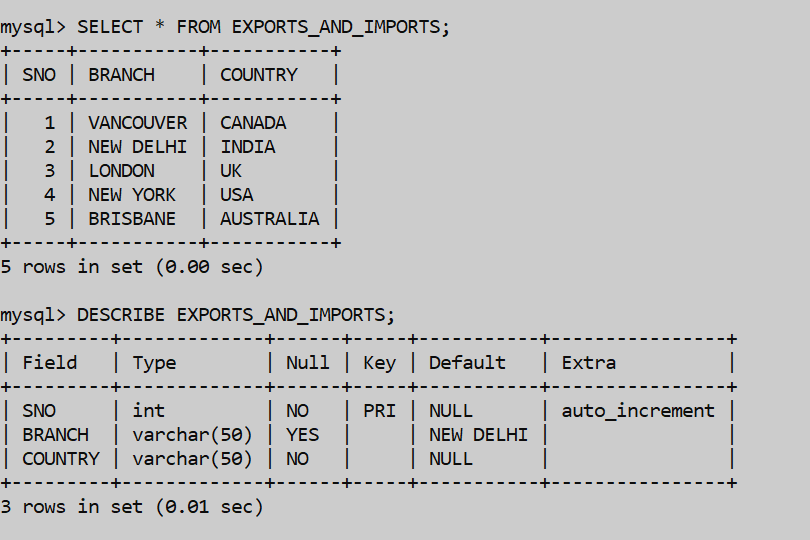
* TABLE NAME – **INVENTORIES**

DESCRIBE INVENTORIES;



* TABLE NAME – **EXPORTS\_AND\_IMPORTS**

DESCRIBE EXPORTS\_AND\_IMPORTS;



# **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.

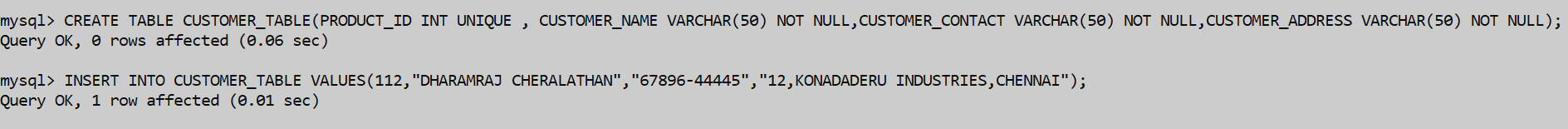
USE INVENTORY;



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

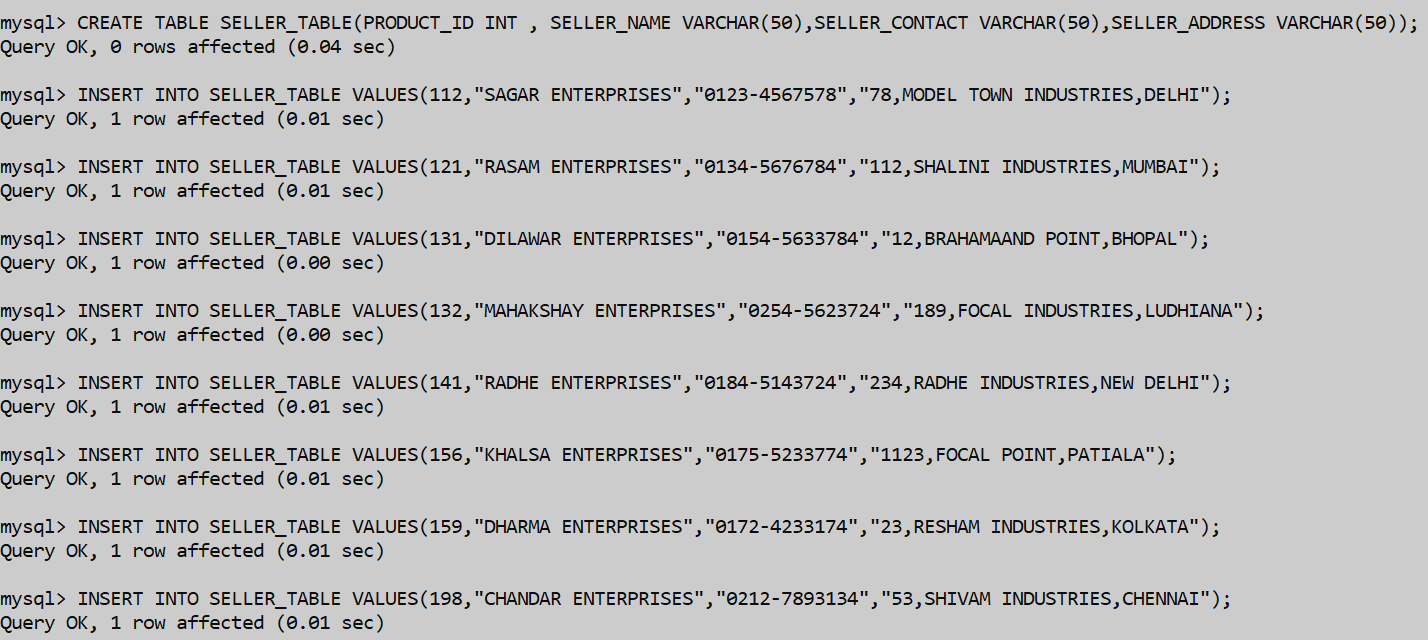
CREATE TABLE CUSTOMER\_TABLE(PRODUCT\_ID INT UNIQUE,CUSTOMER\_NAME(50) NOT NULL,CUSTOMER\_CONTACT VARCHAR(50) NOT NULL,CUSTOMER\_ADDRESS VARCHAR(50) NOT NULL;

INSERT COMMAND WITH INSERTING VALUES.



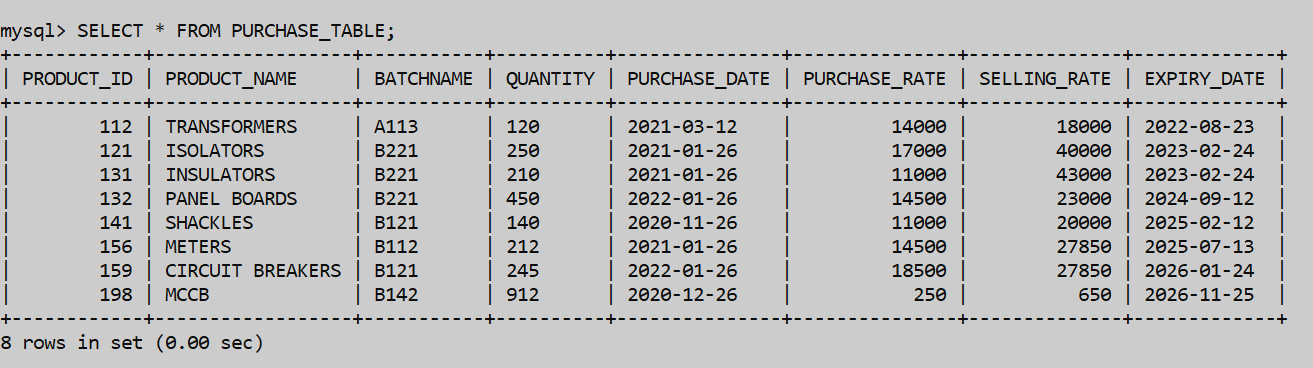
1. Now Here we Will use the query to insert the values inside the created table we have created inside our database.

INSERT COMMAND WITH INSERTING VALUES.



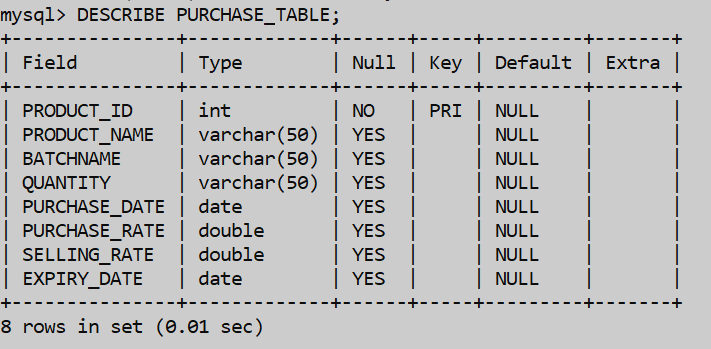
1. Now here we will use the select command or the select query to select something from the table or select something specific from the table.

SELECT \* FROM PURCHASE\_TABLE;



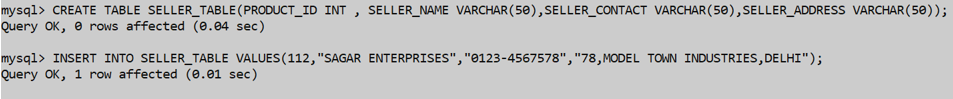
1. The command of the describe table will be used to describe whole of the table that what are the attributes we have used and what are the datatypes we have used inside our attributes alongside with the keys we have used and the extras we will use inside the table or the database.

DESCRIBE PURCHASE\_TABLE;



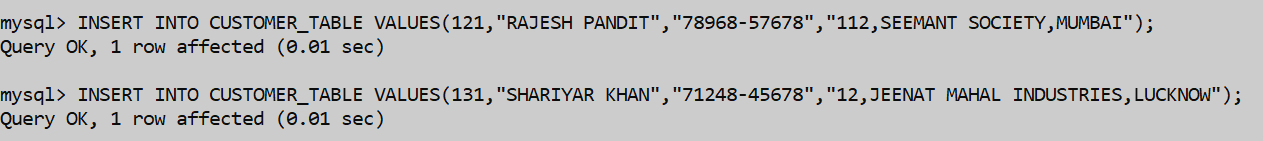
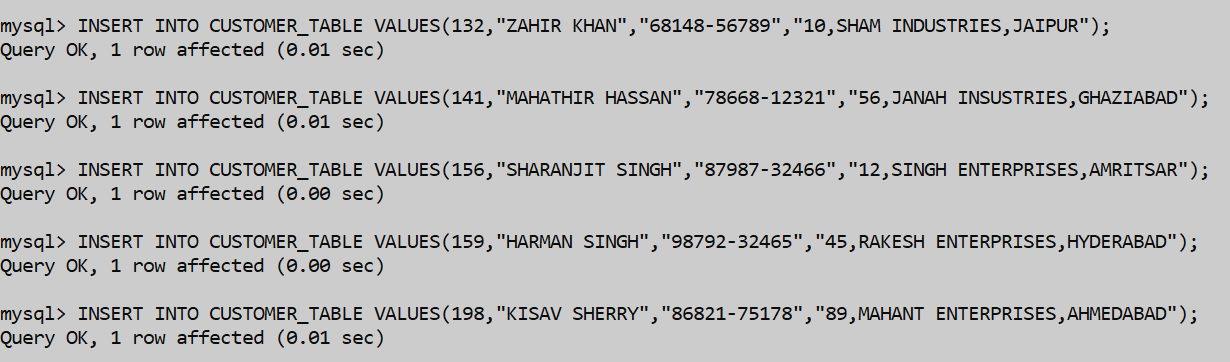
1. Here Again the create command will be used similarly to create another table.

CREATE AND THE INSERT COMMAND.



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

INSERT COMMAND TO INSERT THE VALUES.



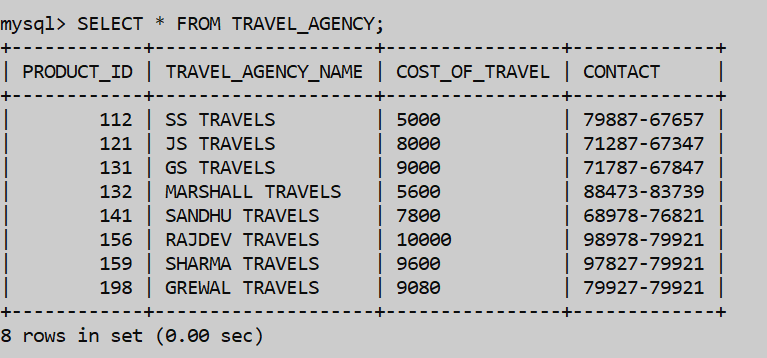
1. Here Again the describe command will be used similarly to describe another table.

DESCRIBE SALE\_TABLE;



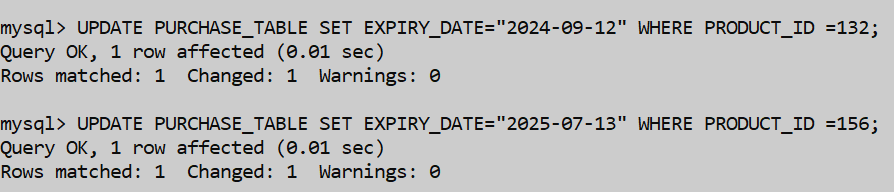
1. Here Again the select command will be used similarly to select another table values from the table or the entire table will be selected.

SELECT \* FROM TRAVEL\_AGENCY;



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

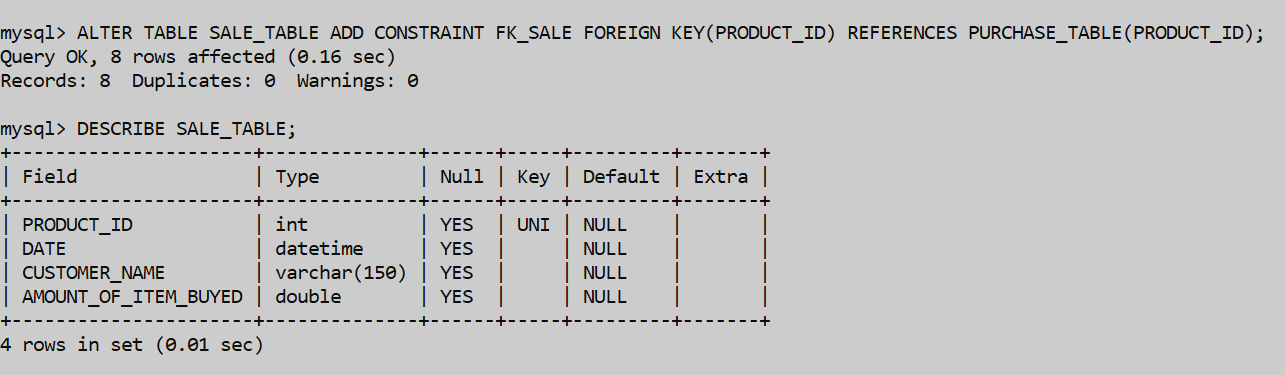
UPDATE PURCHASE\_TABLE SET EXPIRY\_DATE = “2024-09-12” WHERE PRODUCT\_ID= 132;

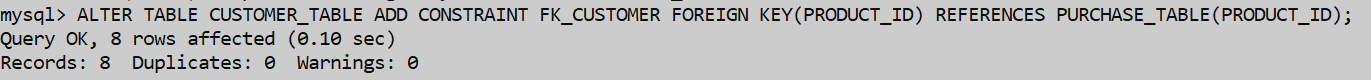


# **APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB-3 IN PROJECT**

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);





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

DROP TABLE CUSTOMER\_TABLE;



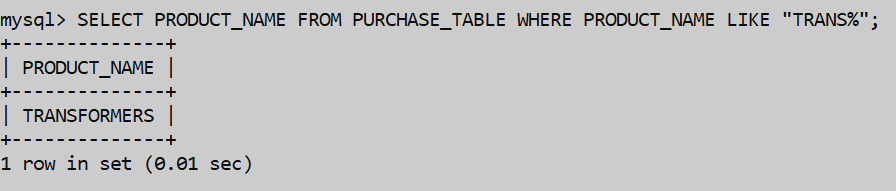
1. Delete Command Will used to delete a row from the table .

DELETE FROM SELLER\_TABLE WHERE SELLER\_NAME = “ZAHIR KHAN”;



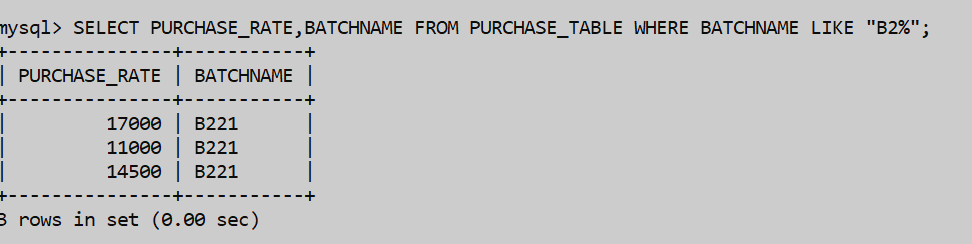
1. Here We will be using the like and where command in select query which select product\_name whose name starts from ‘TRANS’.

SELECT PRODUCT\_NAME FROM PURCHASE\_TABLE WHERE PRODUCT\_NAME LIKE “TRANS%”;



1. 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%”;



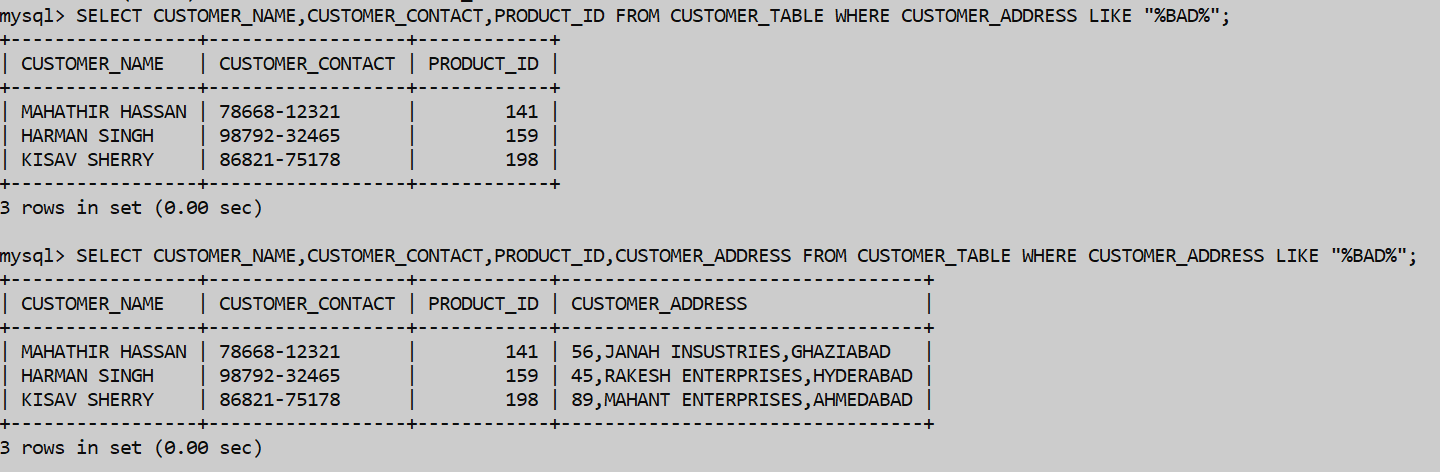
1. If we want to select seller\_name, seller\_contact from the table whose address is situated in “DELHI”.

SELECT SELLER\_NAME, SELLER\_CONTACT FROM seller\_TABLE WHERE SELLER\_ADDRESS LIKE “%DELHI%”;



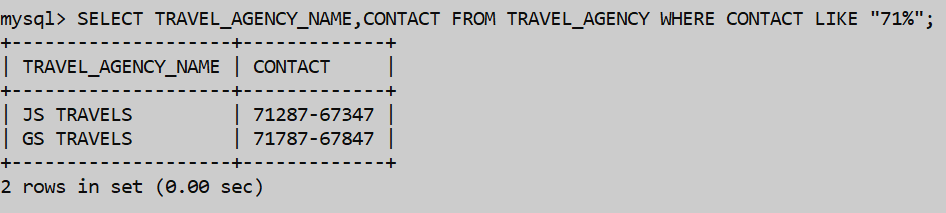
1. Select The specific Information from the table whose city names consists of “BAD” consisting of various cities.

SELECT CUSTOMER\_NAME,CUSTOMER\_CONTACT,PRODUCT\_ID FROM CUSTOMER\_TABLE WHERE CUSTOMER\_ADDRESS LIKE “%BAD%”;



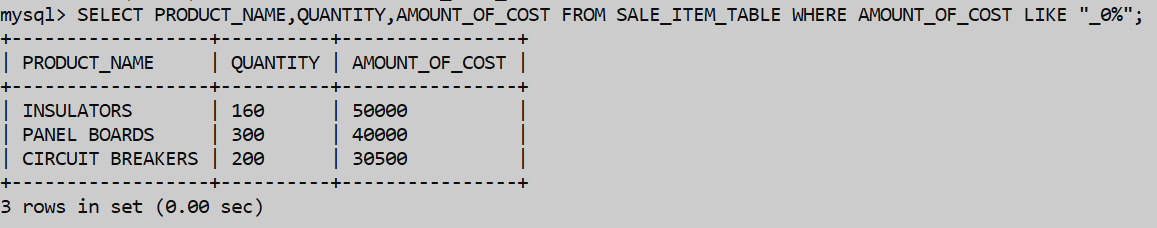
1. If We want to select specific information from the table which has the info to select some contact info from table which has first two digits having “71” in them.

SELECT TRAVEL\_AGENCY\_NAME, CONTACT FROM TRAVEL\_AGENCY WHERE CONTACT LIKE “71%”;



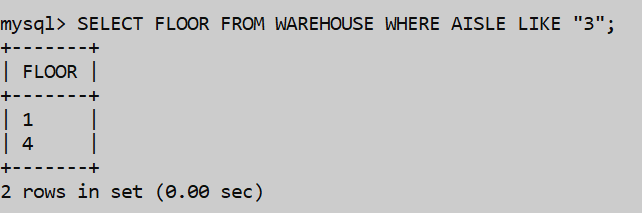
1. Select some info from the table from the table which have the cost ending from “0” in the table.

SELECT PRODUCT\_NAME,QUANTITY,AMOUNT\_OF\_COST FROM SALE\_ITEM\_TABLE WHERE AMOUNT\_OF\_COST LIKE “\_0%”;



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

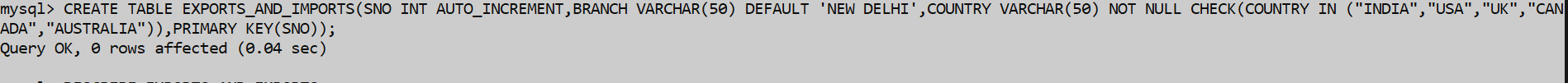
SELECT FLOOR FROM WAREHOUSE WHERE AISLE LIKE “3”;



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

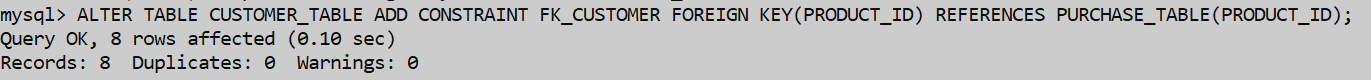
1. A command to implement auto\_increment , check ,primary key and various other constraint s inside the table.

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));



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

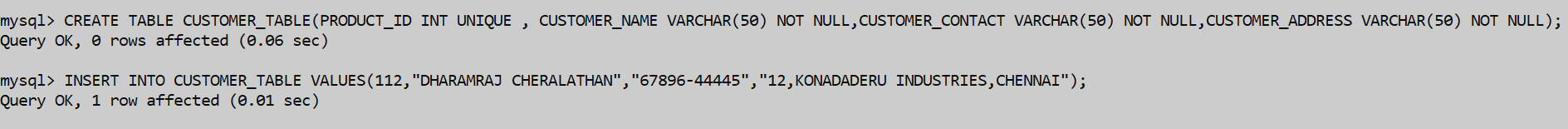
ALTER TABLE CUSTOMER\_TABLE ADD CONSTRAINT FK\_CUSTOMER FOREIGN KEY(PRODUCT\_ID) REFRENCES PURCHASE\_TABLE(PRODUCT\_ID);



1. 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.



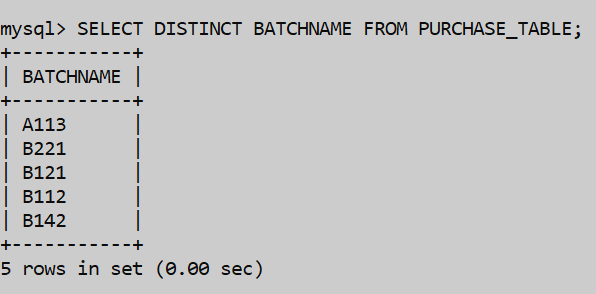
1. A describe command that will describe the table which will tell about the different datatypes present inside the table or the extras which are present inside the table.

ALTER TABLE WAREHOUSE ADD PRIMARY KEY(PRODUCT\_ID);



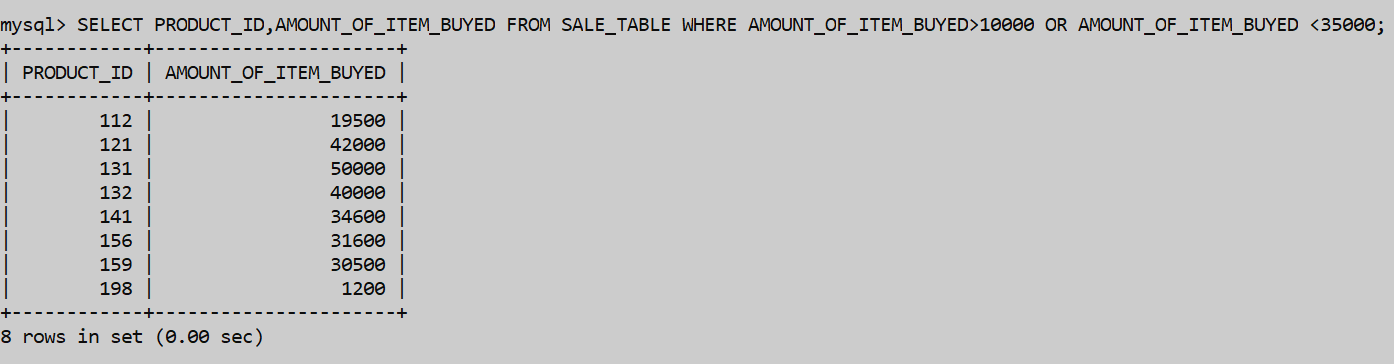
1. A command to implement distinct batch name from the table .

SELECT DISTINCT BATCHNAME FROM PURCHASE\_TABLE;



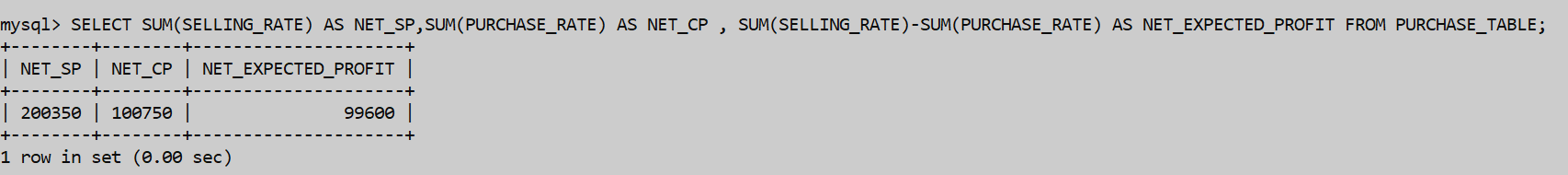
1. A command to implement the Or in the sql commands which give us the conditional statements.

SELECT PRODUCT\_ID, AMOUNT\_OF\_ITEM\_BUYED FROM SALE\_TABLE WHERE AMOUNT\_OF\_ITEM\_BUYED>10000 OR AMOUNT\_OF\_ITEM\_BUYED<35000 ;



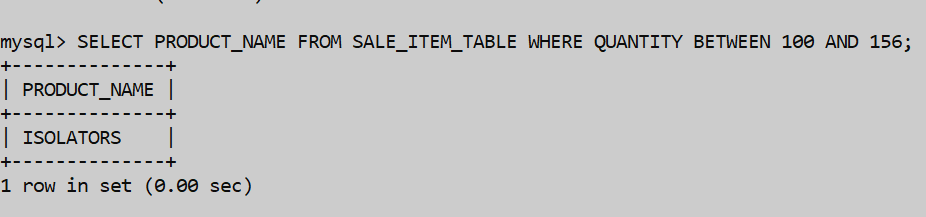
1. A command to implement the alias or the as command query inside our command and also finding the sum of selling\_rate.

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;



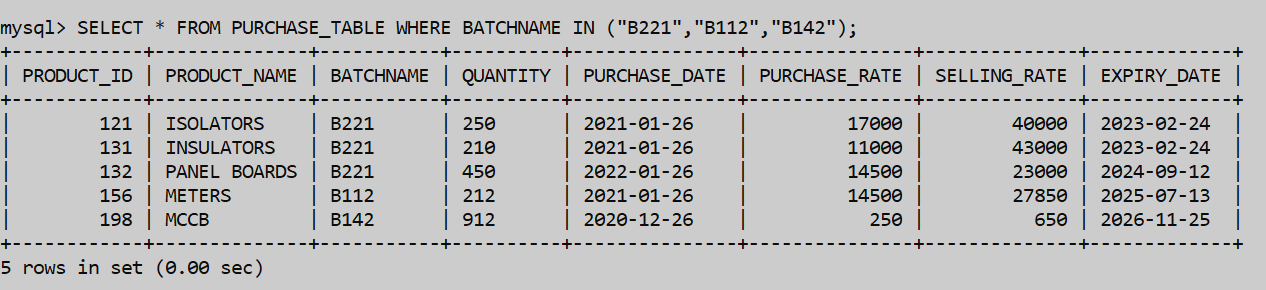
1. Select command with and constraint or condition which imposes that to select purchase\_rate and selling\_rate in a range and also the use of between which specifies the range of given values .

SELECT PRODUCT\_NAME FROM SALE\_ITEM\_TABLE WHERE QUANTITY BETWEEN 100 ANS 156;



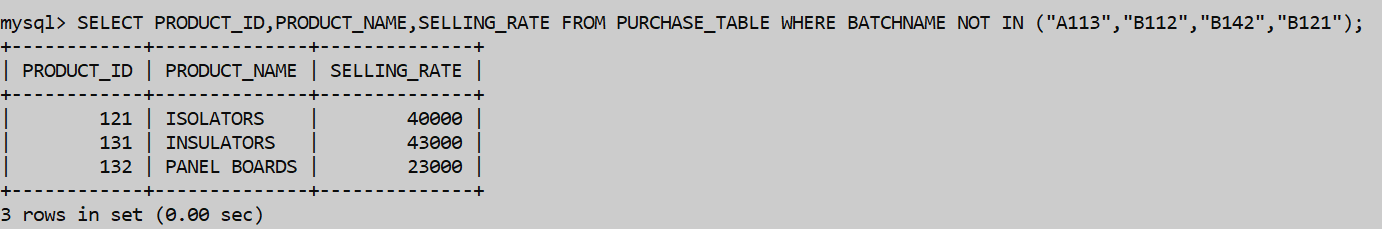
1. Select command with the in clause to select batchname from the given set of values given in the command.

SELECT \* FROM PURCHASE\_TABLE WHERE BATCHNAME IN (“B221” , “B112” , “B142”);



1. 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”);



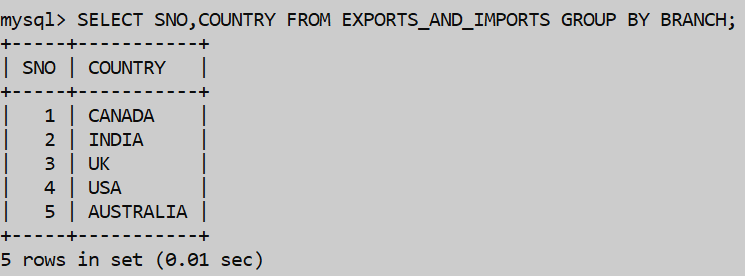
1. Select command to implement the order by clause by use of descending order of values by ordering the selling\_rate in descending order.

SELECT \* FROM PURCHASE\_TABLE ORDER BY SELLING\_RATE DESC;



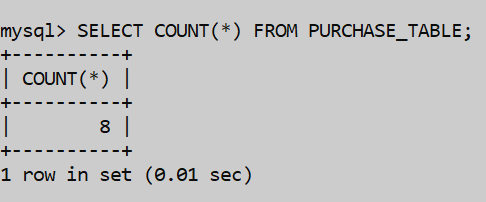
1. Select command to implement the group by clause to display all of the values by grouping the data according to a specific attribute as we do here by selecting a set of values from table by grouping according to branch.

SELECT SNO,COUNTRY FROM EXPORTS\_AND\_IMPORTS GROUP BY BRANCH;



1. Select command which makes the use of count clause as here we count the no. of rows in the purchase\_table.

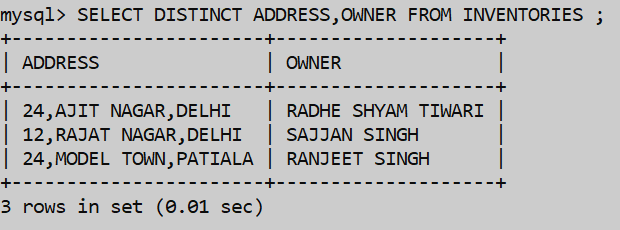
SELECT COUNT(\*) FROM PURCHASE\_TABLE;



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

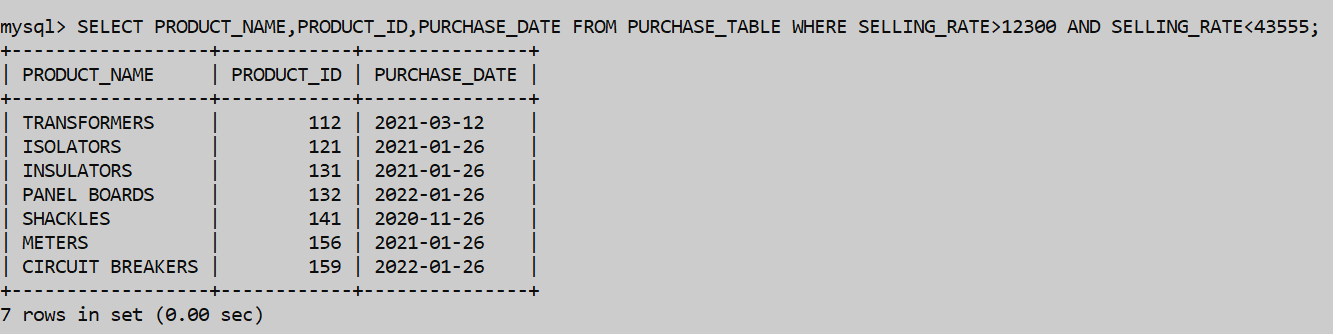
1. Select command with the distinct constraint which selects distinct address, owner of inventories table.

SELECT DISTINCT ADDRESS,OWNER FROM INVENTORIES;



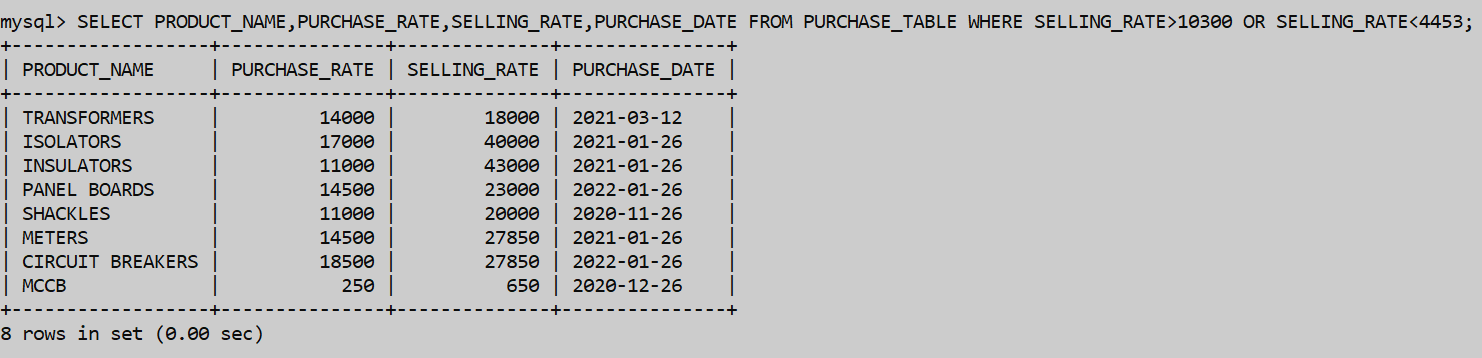
1. Select Command with the where and and clause which gives the specific range of values to be included of selling\_rate.

SELECT PRODUCT\_NAME,PRODUCT\_ID,PURCHASE\_DATE FROM PURCHASE\_TABLE WHERE SELLING\_RATE>12300 AND SELLING\_RATE<43555;



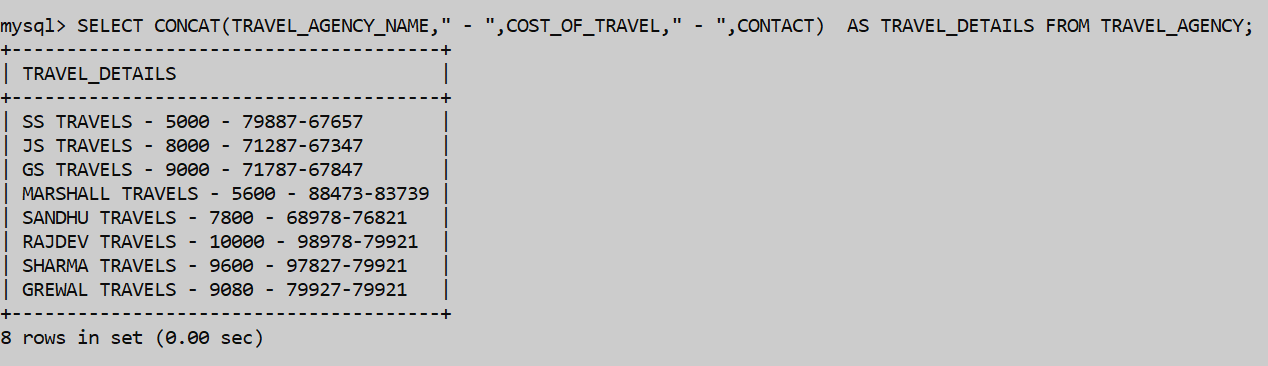
1. Select Command with the where and or clause which gives the specific range of values to be included of selling\_rate.

SELECT PRODUCT\_NAME,PURCHASE\_RATE,SELLING\_RATE,PURCHASE\_DATE FROM PURCHASE\_TABLE WHERE SELLING\_RATE>10300 OR SELLING\_RATE<4453;



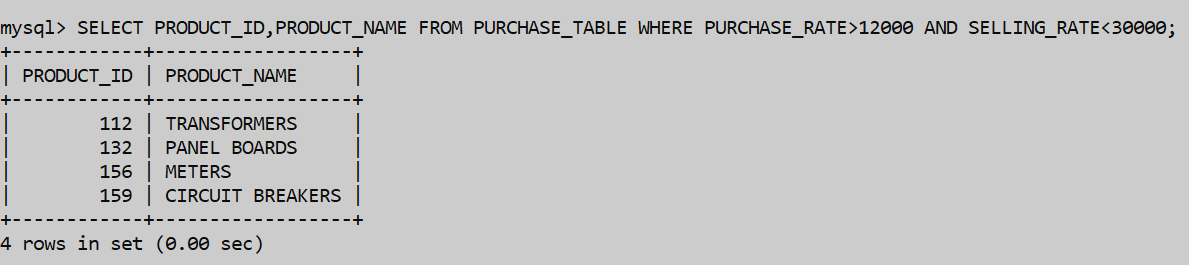
1. Concat command in select query uses the concatenation of various values of attribuyes of travel\_agency table.

SELECT CONCAT(TRAVEL\_AGENCY\_NAME, “ – “ , CONTACT) AS TRAVEL\_DETAILS FROM TRAVEL\_AGENCY;



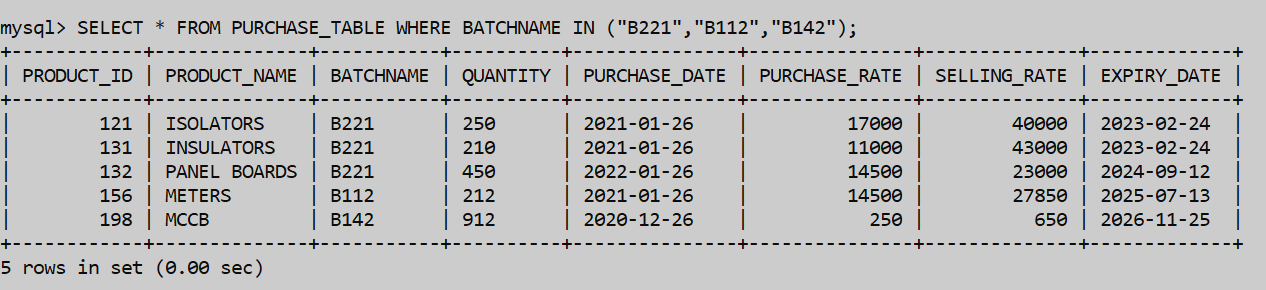
1. Select command which uses as command in the use of between clause by the use of taking the range of values of selling\_rate.

SELECT PRODUCT\_ID,PRODUCT\_NAME FROM PURCHASE\_TABLE WHERE PURCHASE\_RATE >12000 AND SELLING\_RATE<30000;



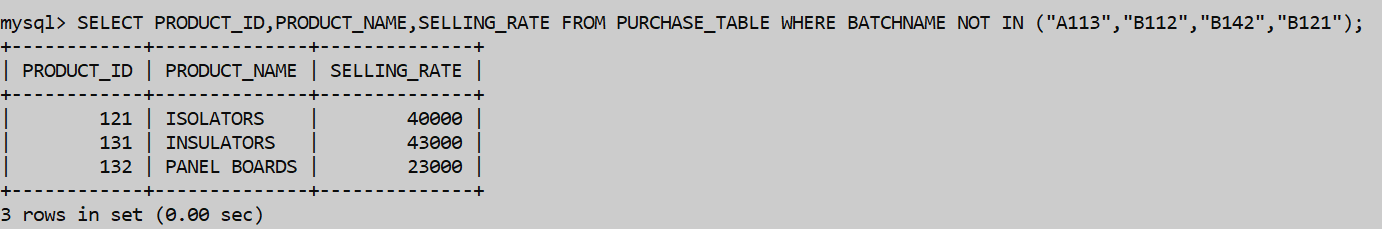
1. Select command which makes use of in clause to take a set of values in batchname.

SELECT \* FROM PURCHASE\_TABLE WHERE BATCHNAME IN (“B221”, “B112” , “B142”);



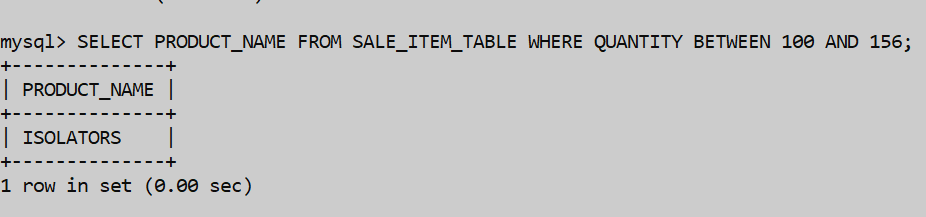
1. Select command which makes use of not in clause to not take a set of values in batchname.

SELECT PRODUCT\_ID,PRODUCT\_NAME,SELLING\_RATE FROM PURCHASE\_TABLE WHERE BATCHNAME NOT IN (“A113” , “B112” , “B142”, “B121”);



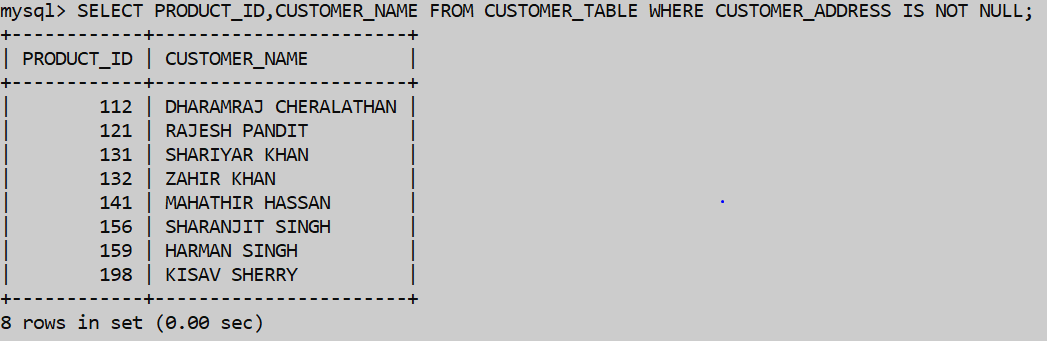
1. Select command which makes the use of between and and clause which makes use of taking the set of values of quantity in a specific range of values and selects some info.

SELECT PRODUCT\_NAME FROM SALE\_ITEM\_TABLE WHERE QUANTITY BETWEEN 100 AND 156;



1. 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;



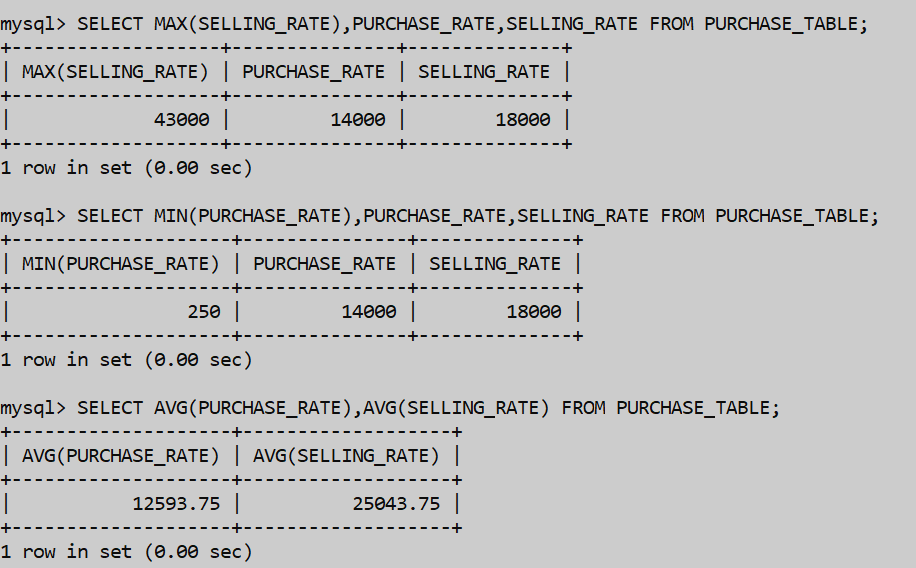
# **APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB-6 IN PROJECT**

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;

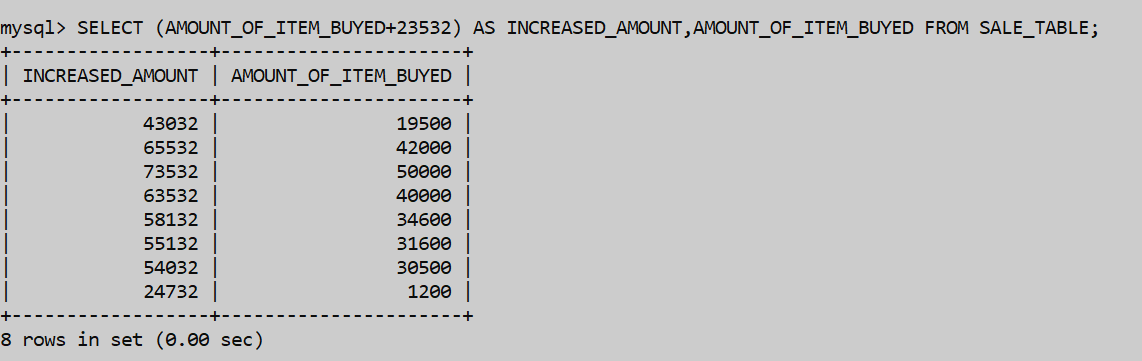
SELECT MIN(SELLING\_RATE), PURCHASE\_RATE,SELLING\_RATE FROM PURCHASE\_TABLE;

SELECT AVG(SELLING\_RATE),AVG(SELLING\_RATE) FROM PURCHASE\_TABLE;



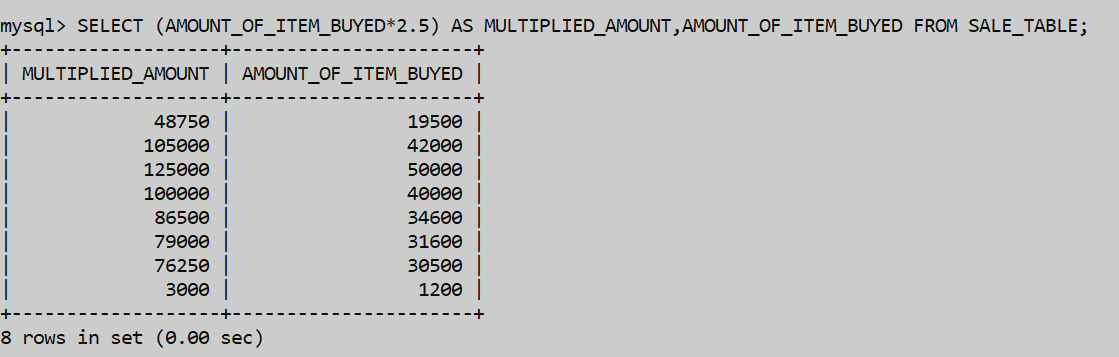
1. Select command to implement the add some value to the amount\_of\_item\_buyed.

SELECT (AMOUNT\_OF\_ITEM\_BUYED+23532) AS INCREASED\_AMOUNT, AMOUNT\_OF\_ITEM\_BUYED FROM SALE\_TABLE;



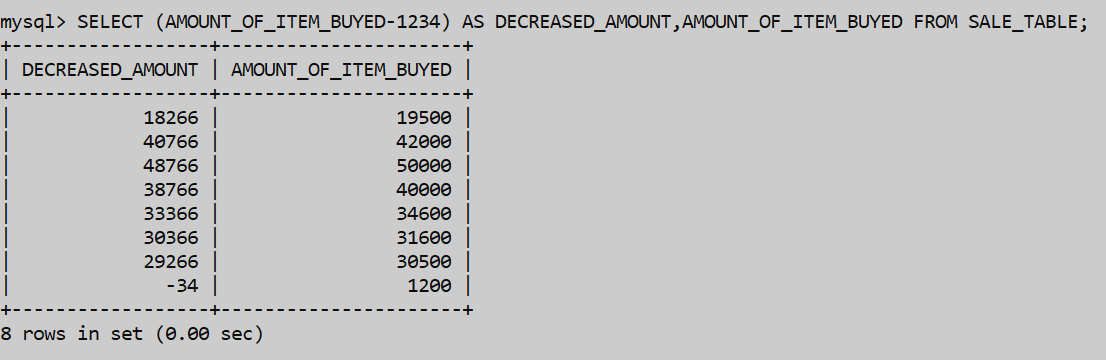
1. Select Command to multiply some value to the amount\_of\_item\_buyed in the command.

SELECT(AMOUNT\_OF\_ITEM\_BUYED\*2.5)AS MULTIPLIED\_AMOUNT,AMOUNT\_OF\_ITEM\_BUYED FROM SALE\_TABLE;



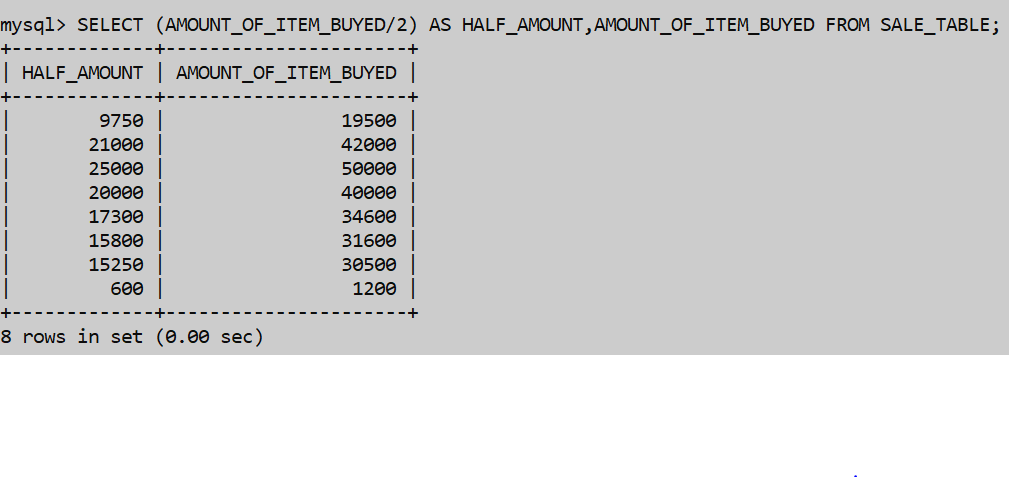
1. Select Command to subtract some value from the amount\_of\_item\_buyed in the command.

SELECT (AMOUNT\_OF\_ITEM\_BUYED-1234) AS DECREASED\_AMOUNT,AMOUNT\_OF\_ITEM\_BUYED FROM SALE\_TABLE;



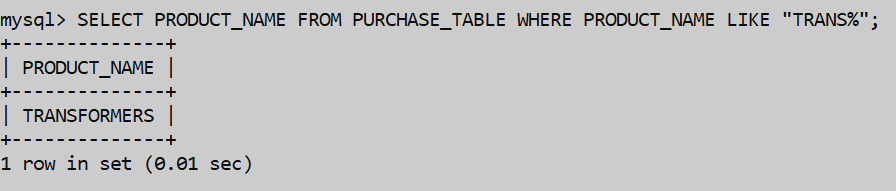
1. Select Command to divide some value to the amount\_of\_item\_buyed in the command.

SELECT (AMOUNT\_OF\_ITEM\_BUYED/2) AS HALF\_AMOUNT,AMOUNT\_OF\_ITEM\_BUYED FROM SALE\_TABLE;



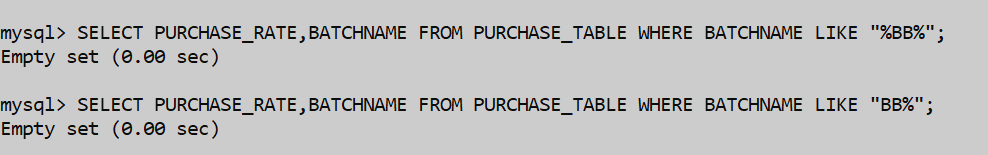
1. Select command which uses the like command clause which displays the product name starting from “TRANS”.

SELECT PRODUCT\_NAME FROM PURCHASE\_TABLE WHERE PRODUCT\_NAME LIKE “TRANS%”;



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

SELECT PURCHASE\_RATE,BATCHNAME FROM PURCHASE\_TABLE WHERE BATCHNAME LIKE “%BB%”;



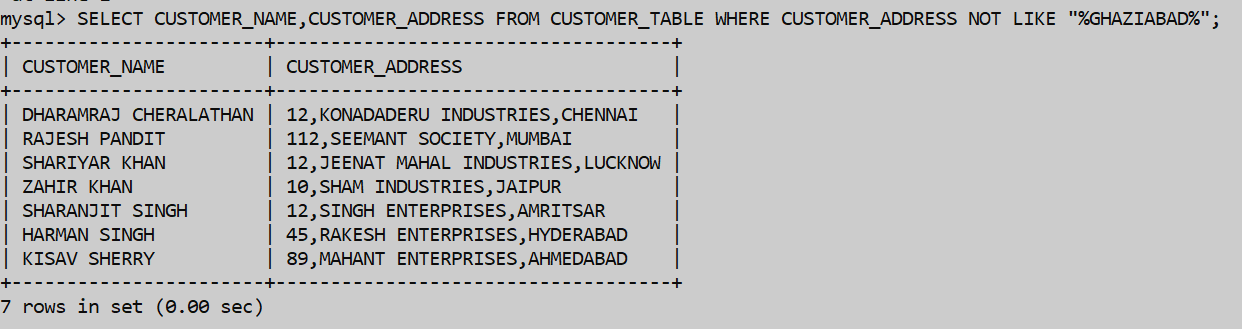
1. Select command which displays all the info from purchase\_table which had date not having march month.

SELECT \* FROM PURCHASE\_TABLE WHERE PURCHASE\_DATE NOT LIKE “\_\_\_\_\_03%”;



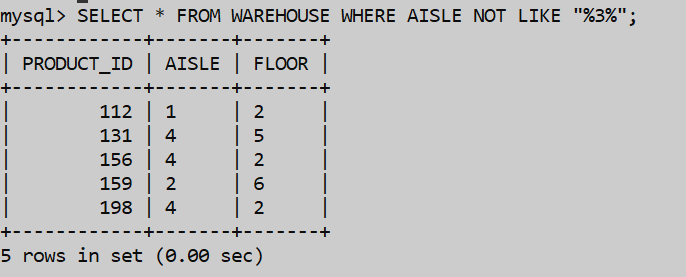
1. Select some info from the customer\_table which has address not having the address city as “GHAZIABAD”.

SELECT CUSTOMER\_NAME,CUSTOMER\_ADDRESS FROM CUSTOMER\_TABLE WHERE CUSTOMER\_ADDRESS NOT LIKE “%GHAZIABAD%”;



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

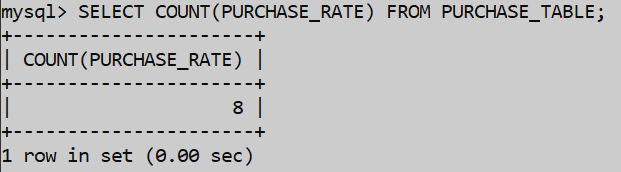
SELECT \* FROM WAREHOUSE WHERE AISLE NOT LIKE “%3%”;



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

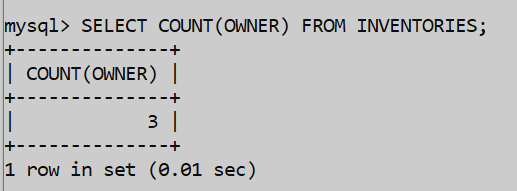
1. Select the total purchase\_rate of different types of the products in the purchase\_table.

SELECT COUNT(PURCHASE\_RATE) FROM PURCHASE\_TABLE;



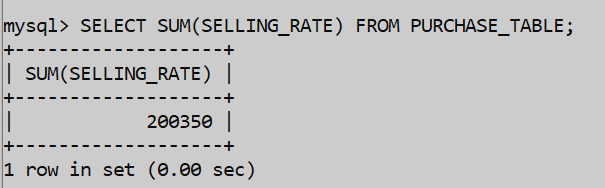
1. Select the total no. of owners of different types of the products in the inventories.

SELECT COUNT(OWNER) FROM INVENTORIES;



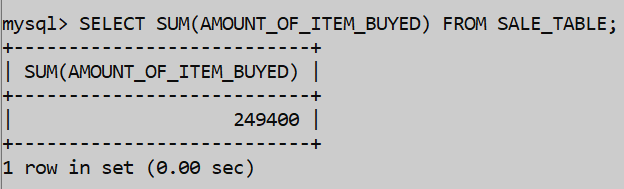
1. Select the total or the sum of the selling\_rate in the purchase\_table.

SELECT SUM(SELLING\_RATE) FROM PURCHASE\_TABLE;



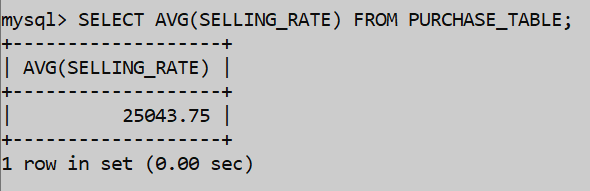
1. Select the total or the sum of the amount\_of\_item\_buyed in the sale\_table.

SELECT SUM(AMOUNT\_OF\_ITEM\_BUYED) FROM SALE\_TABLE;



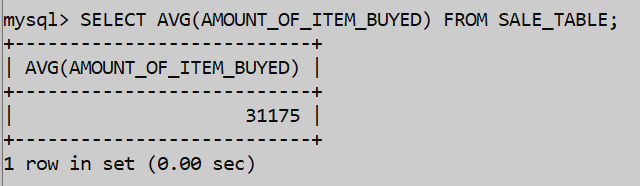
1. Select Command which finds the avg(selling\_rate) from the purchase\_table.

SELECT AVG(SELLING\_RATE) FROM PURCHASE\_TABLE;



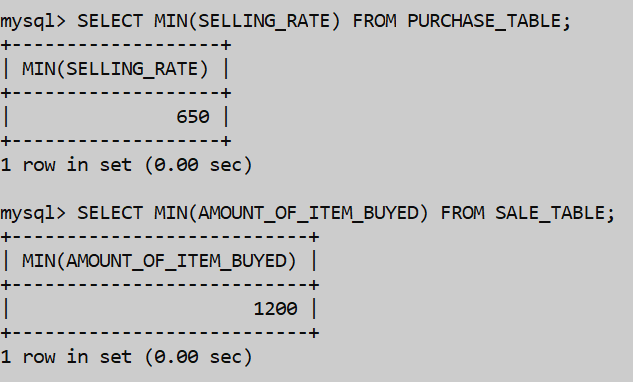
1. Select the avg of the amount\_of\_item\_buyed in the sale\_table.

SELECT AVG(AMOUNT\_OF\_ITEM\_BUYED) FROM SALE\_TABLE;



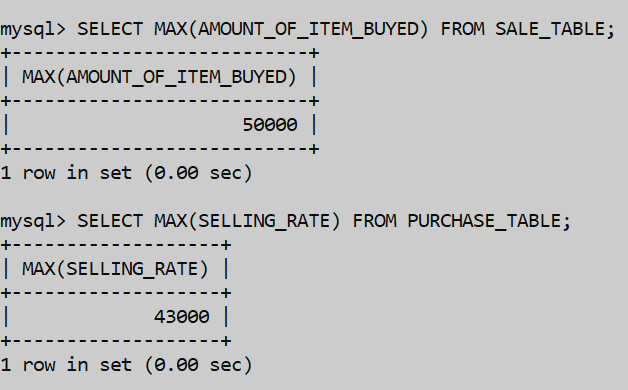
1. Select command with the min clause which gives the min value of amount\_of\_item\_buyed and selling\_rate.

SELECT MIN(SELLING\_RATE) FROM PURCHASE\_TABLE;



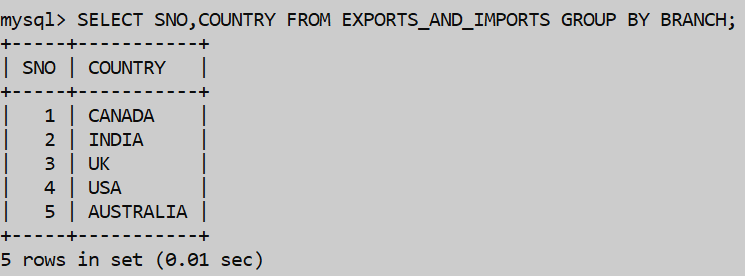
1. Select command with the max clause which gives the min value of amount\_of\_item\_buyed and selling\_rate.

SELECT MAX(AMOUNT\_OF\_ITEM\_BUYED) FROM SALE\_TABLE;



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

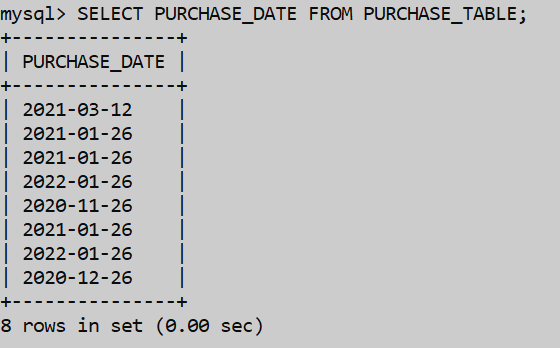
SELECT SNO, COUNTRY FROM EXPORTS\_AND\_IMPORTS GROUP BY BRANCH;



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

1. Select command to show the purchase\_date from purchase\_table.

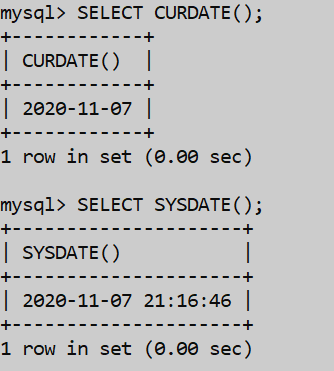
SELECT PURCHASE\_DATE FROM PURCHASE\_TABLE;



1. Select command to show the current date and the system date of the system.

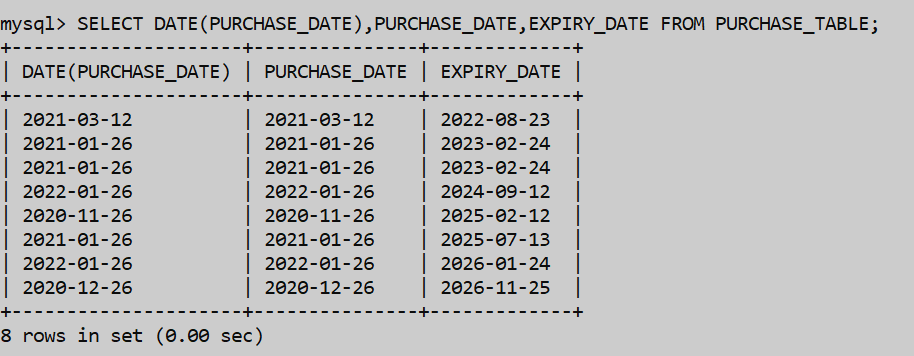
SELECT CURDATE();

SELECT SYSDATE();



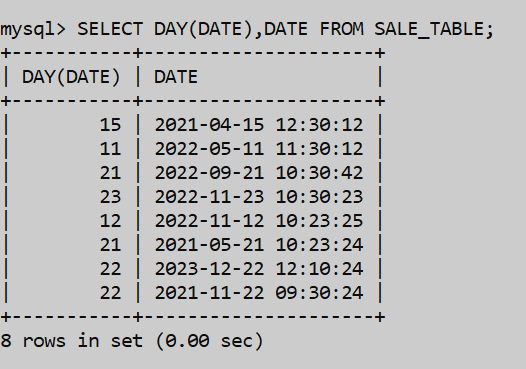
1. Select command with the capability of displaying the date of the purchase\_date.

SELECT DATE(PURCHASE\_DATE),PURCHASE\_DATE, EXPIRY\_DATE FROM PURCHASE\_TABLE;



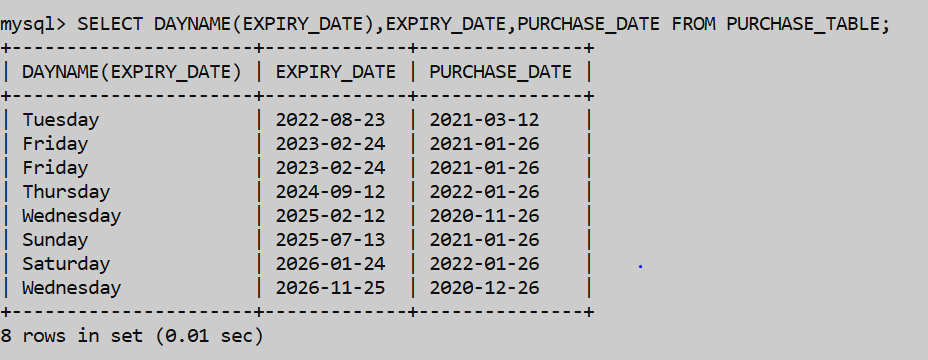
1. Select command to display the day of the date from the table.

SELECT DAY(DATE), DATE FROM SALE\_TABLE;



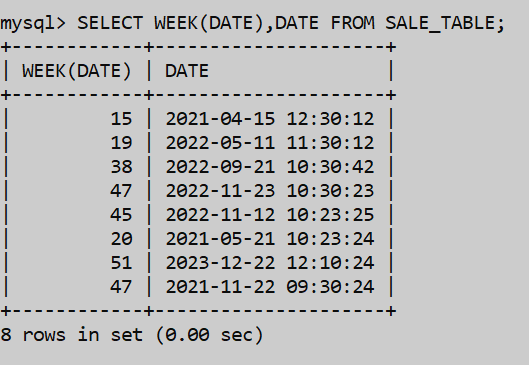
1. Select command to display the dayname of the expiry\_date of the stock in purchase\_table.

SELECT DAYNAME(EXPIRY\_DATE),EXPIRY\_DATE,PURCHASE\_DATE FROM PURCHASE\_TABLE;



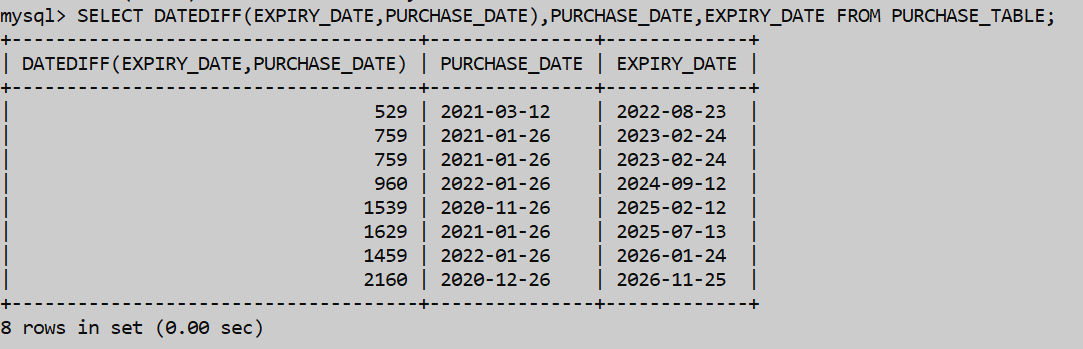
1. Select command to display the week of the day.

SELECT WEEK(DATE) , DATE FROM SALE\_TABLE;



1. 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;



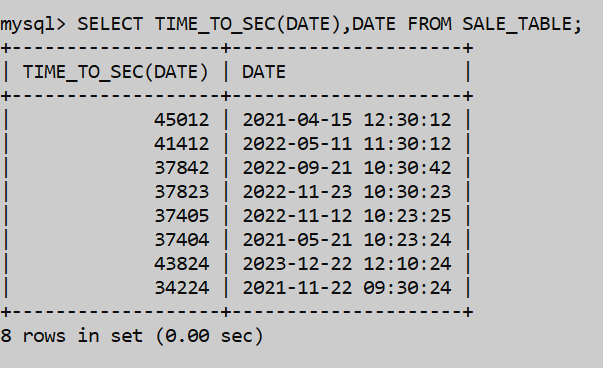
1. Select command which takes the minutes from the date.

SELECT EXTRACT(MINUTE FROM DATE) FROM SALE\_TABLE;



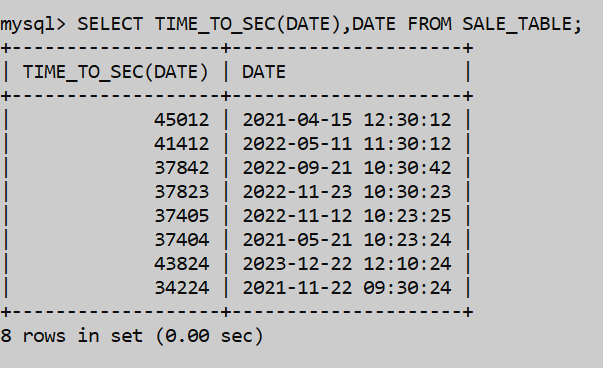
1. Select command which displays the time to seconds that are displayed in the date from sale\_table.

SELECT TIME\_TO\_SEC(DATE) , DATE FROM SALE\_TABLE;



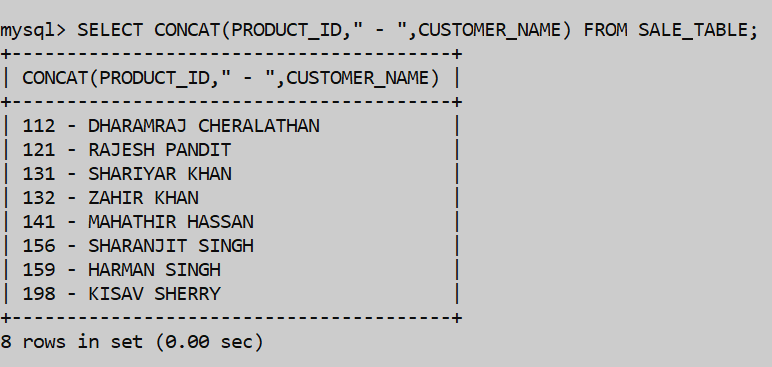
1. Select command that displays the no. of days displayed in the date from sale\_table.

SELECT TIME\_TO\_SEC(DATE),DATE FROM SALE\_TABLE;



1. Select command which displays the concat command that concatenation of product\_id and customer\_name.

SELECT CONCAT(PRODUCT\_ID, “ – “ , CUSTOMER\_NAME) FROM SALE\_TABLE;



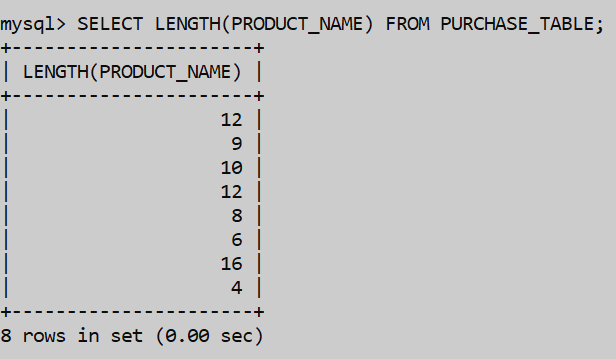
1. Select the particular string character from the string and displays the position of that letter as we displays the letter ‘E’ from customer\_name.

SELECT INSTR(CUSTOMER\_NAME, ‘E’) FROM CUSTOMER\_TABLE;



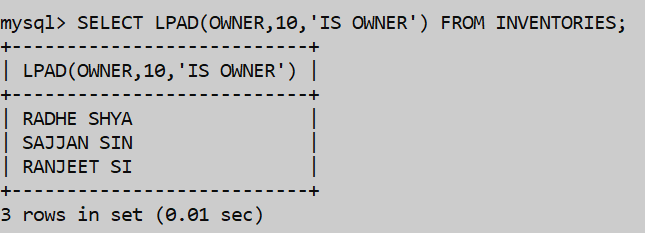
1. Select command which displays the length of the char string product\_name.

SELECT LENGTH(PRODUCT\_NAME) FROM PURCHASE\_TABLE;



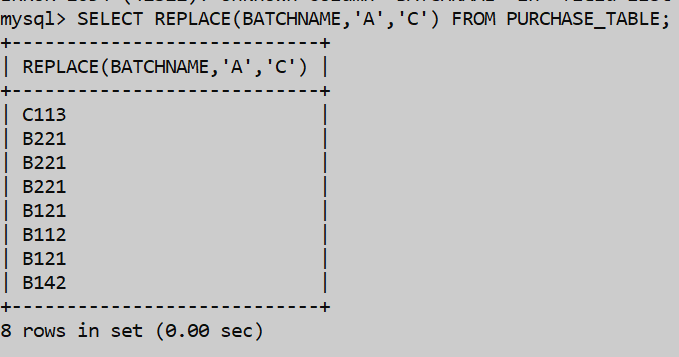
1. Select command which displays the left part of string from a position in owner’s name in the inventories table.

SELECT LPAD(OWNER,10, ‘IS OWNER’) FROM INVENTORIES;



1. Select command which replaces a letter in the Batchname in the purchase\_table.

SELECT REPLACE(BATCHNAME, ‘A’ , ‘C’) FROM PURCHASE\_TABLE;



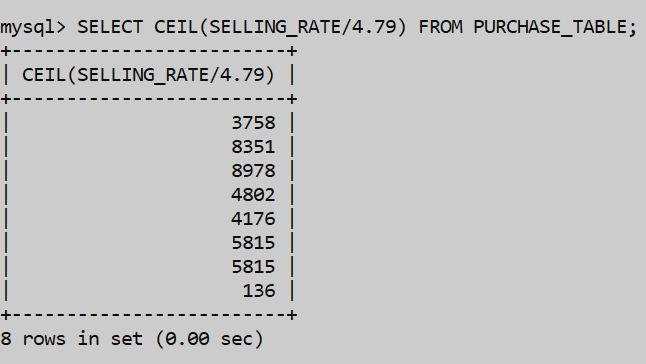
1. Select command which displays the lower case letters in the product\_name in the table.

SELECT LOWER(PRODUCT\_NAME) AS LOWER FROM PURCHASE\_TABLE;



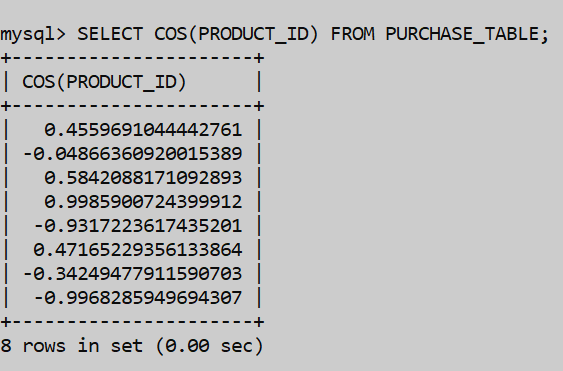
1. Select Command which displays the ceil value of the result of division of selling\_rate by some value.

SELECT CEIL(SELLING\_RATE/4.79) FROM PURCHASE\_TABLE;



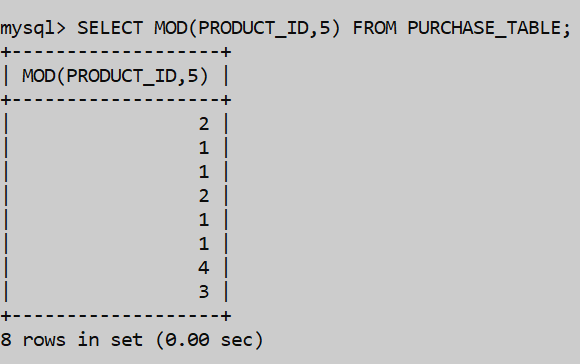
1. Select command which displays the cos value of the product\_id.

SELECT COS(PRODUCT\_ID) FROM PURCHASE\_TABLE;



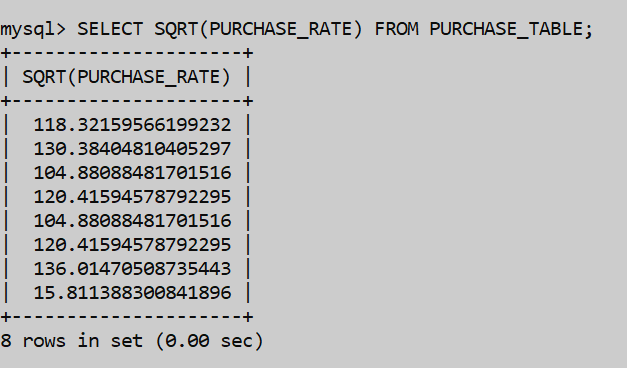
1. Select command that displays the modulus value if we divide the product\_id by some value.

SELECT MOD(PRODUCT\_ID,5) FROM PURCHASE\_TABLE;



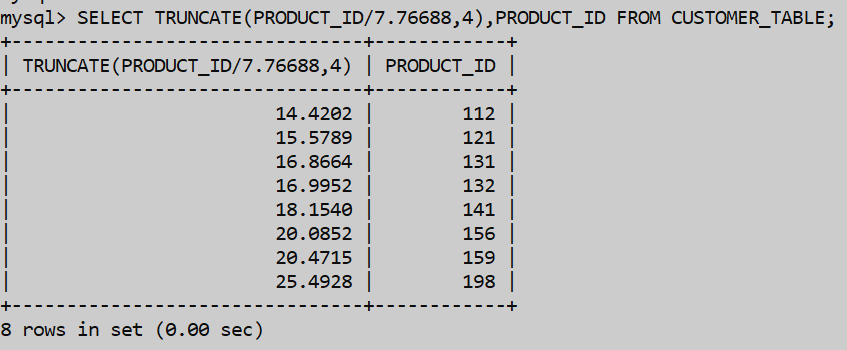
1. Select command that displays the sqrt value of the purchase\_rate from the purchase\_table.

SELECT SQRT(PURCHASE\_RATE) FROM PURCHASE\_TABLE;



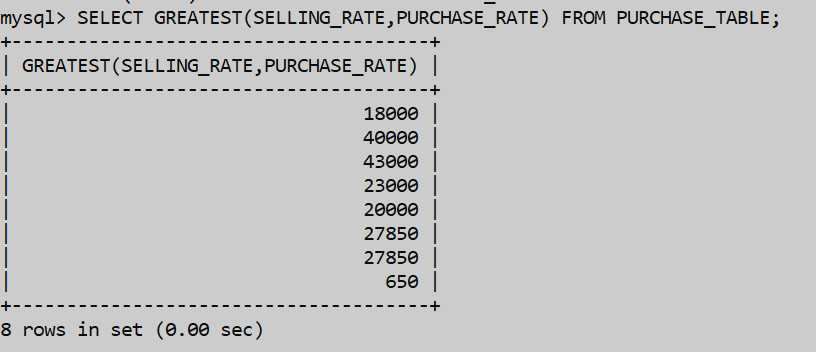
1. Select command which displays the truncated value upto a certain specific value.

SELECT TRUNCATE(PRODUCT\_ID/7.76688,4),PRODUCT\_ID FROM CUSTOMER\_TABLE;



1. Select command displays the greatest selling\_rate, purchase\_rate from the purchase\_table.

SELECT GREATEST(SELLING\_RATE,PURCHASE\_RATE) FROM PURCHASE\_TABLE;



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

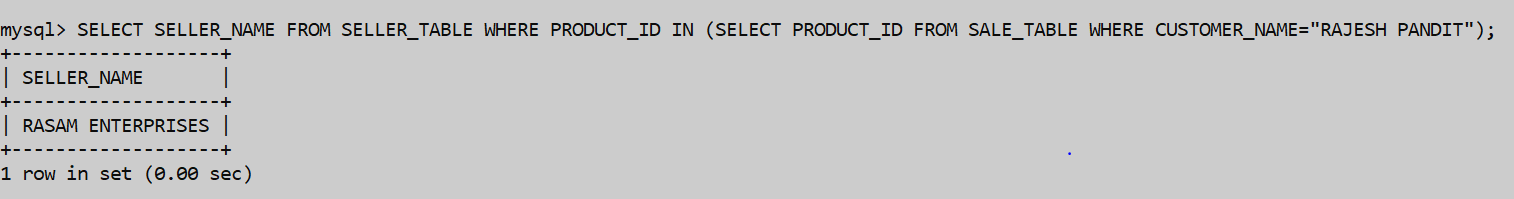
1. We have to select customer\_name from the customer\_table where seller\_address has “U” in it’s spellings by selecting the product\_id as a common thing inside both of the tables.

SELECT CUSTOMER\_NAME FROM CUSTOMER\_TABLE WHERE PRODUCT\_ID IN (SELECT PRODUCT\_ID FROM SELLER\_TABLE WHERE SELLER\_ADDRESS LIKE “%U%”);



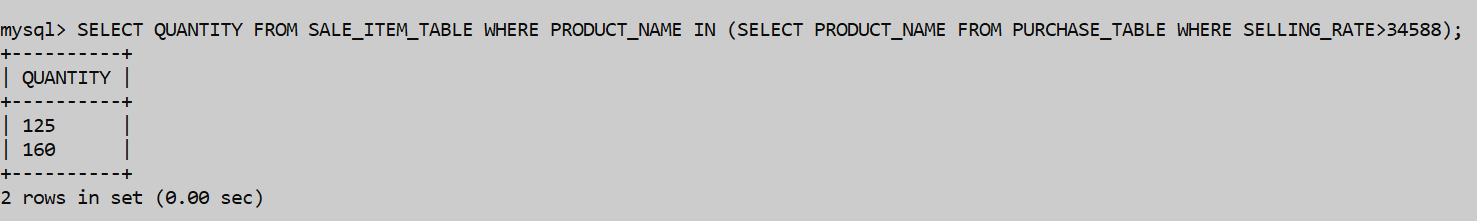
1. We have to select seller\_name from the seller\_table where customer\_name is “Rajesh Pandit” by selecting the product\_id as a common thing inside both of the tables.

SELECT SELLER\_NAME FROM SELLER\_TABLE WHERE PRODUCT\_ID IN (SELECT PRODUCT\_ID FROM SALE\_TABLE WHERE CUSTOMER\_NAME = “RAJESH PANDIT”);



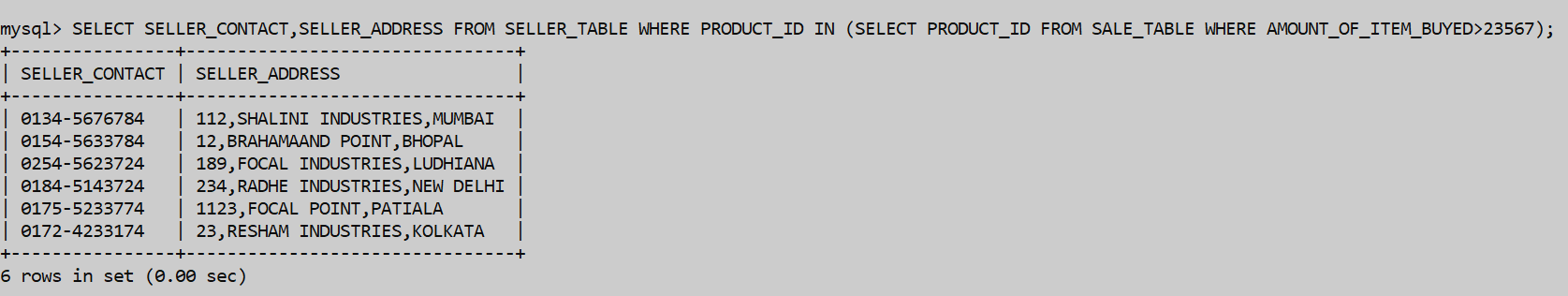
1. We have to select quantity from the sale\_item\_table where selling\_rate is greater than 34588 by selecting the product\_name as a common thing inside both of the tables.

SELECT QUANTITY FROM SALE\_ITEM\_TABLE WHERE PRODUCT\_NAME IN (SELECT PRODUCT\_NAME IN (SELECT PRODUCT\_NAME FROM PURCHASE\_TABLE WHERE SELLING\_RATE >34588);



1. 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);



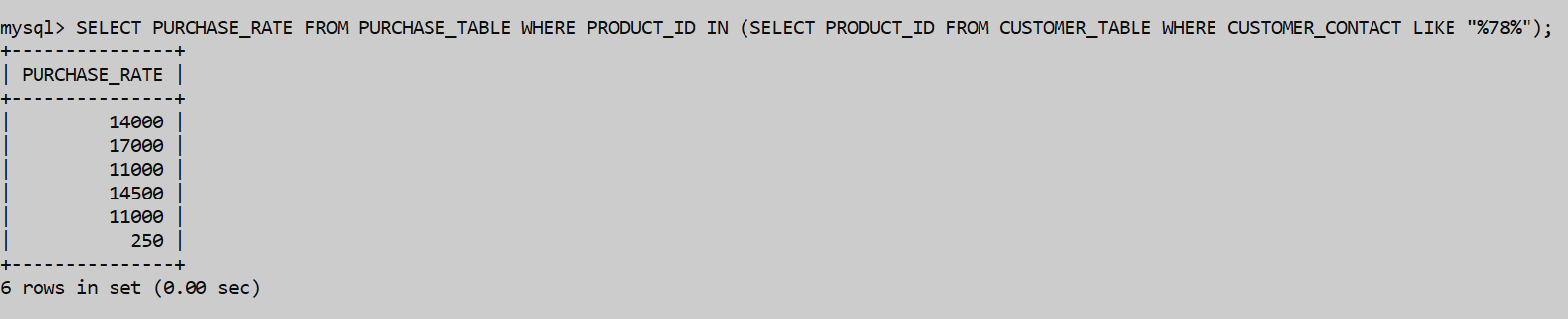
1. Select customer\_contact from customer\_table where aisle=4 or floor=2 by selecting the product\_id as it is common to both of the tables.

SELECT CUSTOMER\_CONTACT FROM CUSTOMER\_TABLE WHERE PRODUCT\_ID IN (SELECT PRODUCT\_ID FROM WAREHOUSE WHERE AISLE= ‘4’ OR FLOOR = ‘2’);



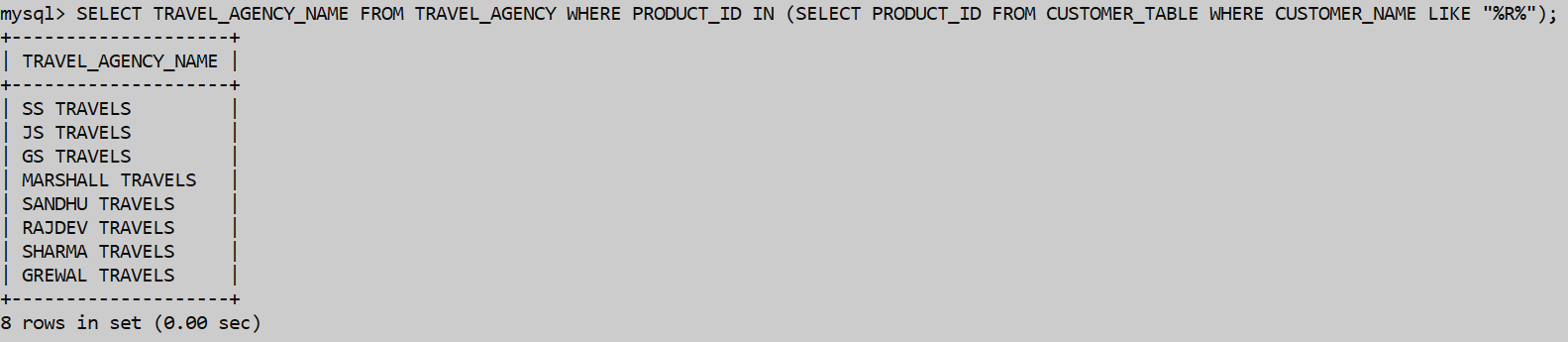
1. Select the purchase\_rate from purchase\_table where customer\_contact has 78 in it’s contact no. by selecting product\_id as it is common to both the tables.

SELECT PURCHASE\_RATE FROM PURCHASE\_TABLE WHERE PRODUCT\_ID IN (SELECT PRODUCT\_ID FROM CUSTOMER\_TABLE WHERE CUSTOMER\_CONTACT LIKE “%78%”);



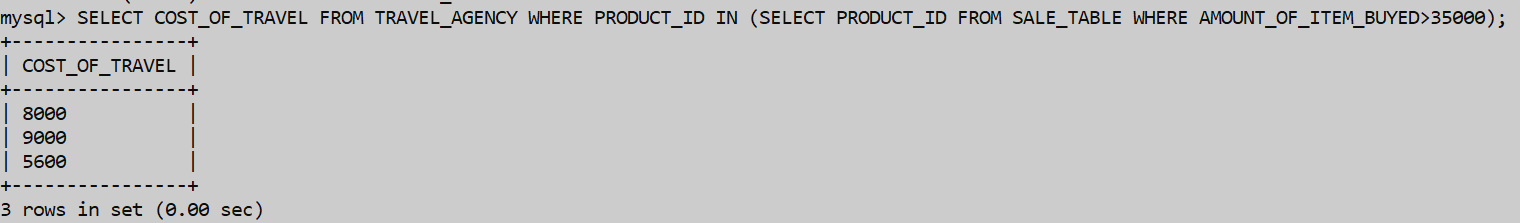
1. Select the travel\_agency\_name from travel\_agency where customer\_name has ‘R’ in it’s customer\_name by selecting product\_id as it is common to both the tables.

SELECT TRAVEL\_AGENCY\_NAME FROM TRAVEL\_AGENCY WHERE PRODUCT\_ID IN (SELECT PRODUCT\_ID FROM CUSTOMER\_TABLE WHERE CUSTOMER\_NAME LIKE “%R%”);



1. 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.

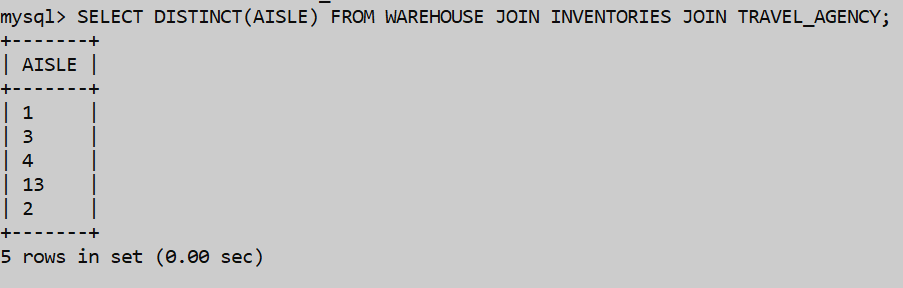
SELECT COST\_OF\_TRAVEL FROM TRAVEL\_AGENCY WHERE PRODUCT\_ID IN (SELECT PRODUCT\_ID FROM SALE\_TABLE WHERE AMOUNT\_OF\_ITEM\_BUYED>35000);



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

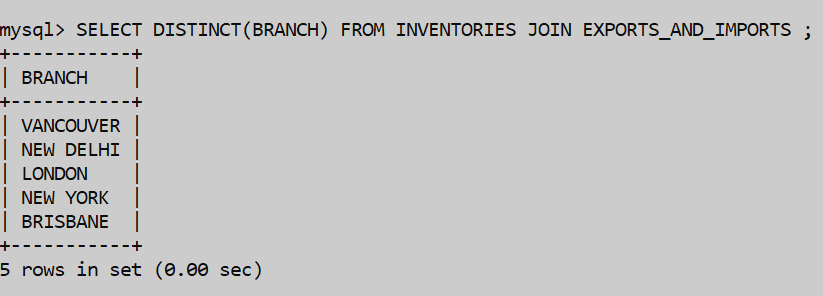
1. Select Distinct Aisle from warehouse join inventories join travel\_agency as we join all the tables so specific info is greatly available with us.

SELECT DISTINCT(AISLE) FROM WAREHOUSE JOIN INVENTORIES JOIN TRAVEL\_AGENCY;



1. 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;



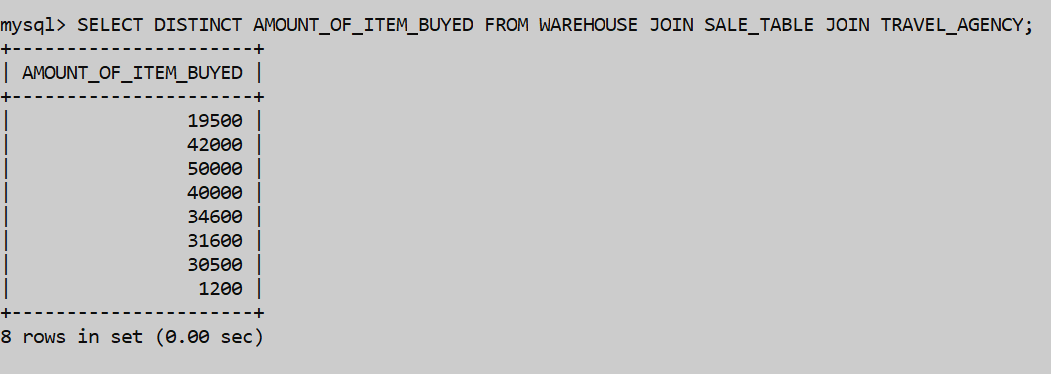
1. If we want to select distinct owner country from the join or the cartesian product of warehouse and exports\_and\_imports.

SELECT DISTINCT OWNER , COUNTRY FROM INVENTORIES JOIN EXPORTS\_AND\_IMPORTS JOIN WAREHOUSE;



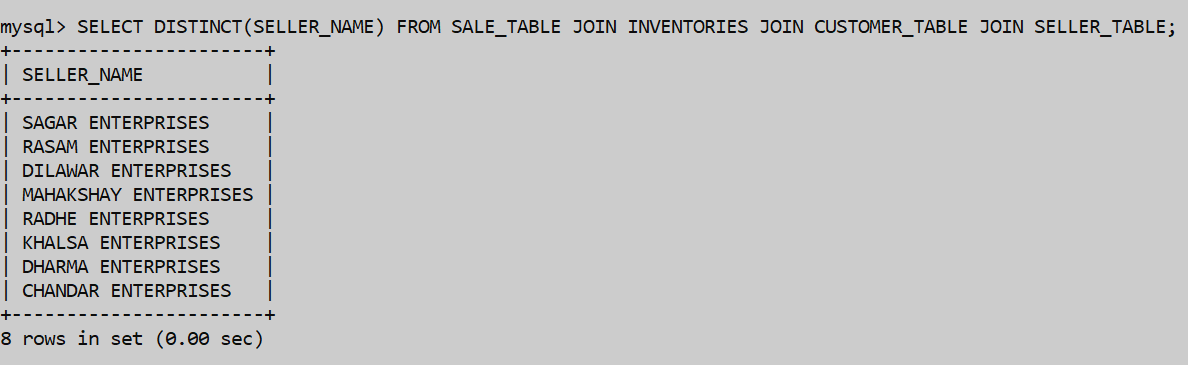
1. If we want to select distinct amount\_of\_item\_buyed from the join or the cartesian product of warehouse , sale\_table, travel\_agency.

SELECT DISTINCT AMOUNT\_OF\_BUYED FROM WAREHOUSE JOIN SALE\_TABLE JOIN TRAVEL\_AGENCY;



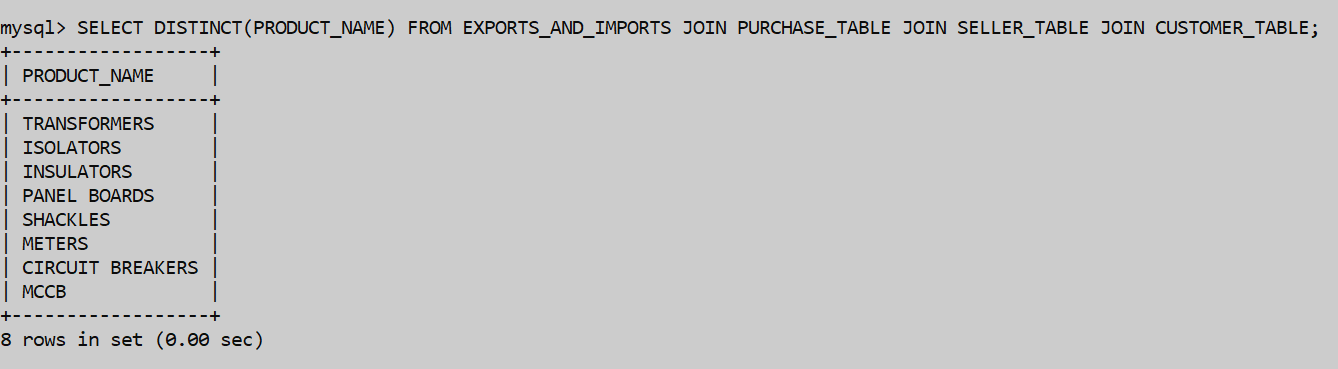
1. Select distinct seller\_name from the join or the cartesian product of sale\_table,inventories,customer\_table and the seller\_table.

SELECT DISTINCT(SELLER\_NAME) FROM SALE\_TABLE JOIN INVENTORIES JOIN CUSTOMER\_TABLE JOIN SELLER\_TABLE;



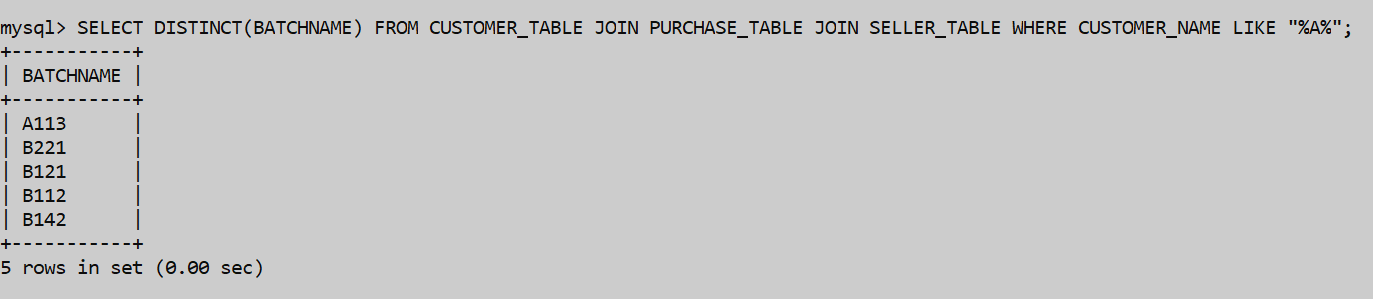
1. Select the distinct product\_name from the join of exports\_and\_imports , purchase\_table , seller\_table, customer\_table.

SELECT DISTINCT PRODUCT\_NAME FROM EXPORTS\_AND\_IMPORTS JOIN PURCHASE\_TABLE JOIN SELLER\_TABLE JOIN CUSTOMER\_TABLE;



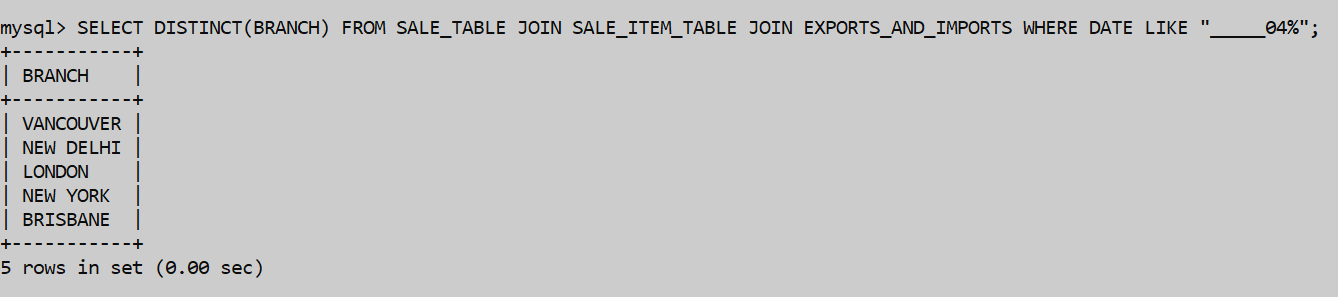
1. Select the distinct batchname from the cartesian product or the join of the purchase\_table, seller\_table , customer\_table where customer\_name has ‘A’ in it’s letter in the spellings.

SELECT DISTINCT BATCHNAME FROM CUSTOMER\_TABLE JOIN PURCHASE\_TABLE JOIN SELLER\_TABLE WHERE CUSTOMER\_NAME LIKE “%A%”;



1. 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%”;

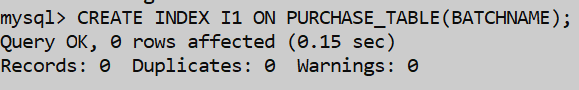


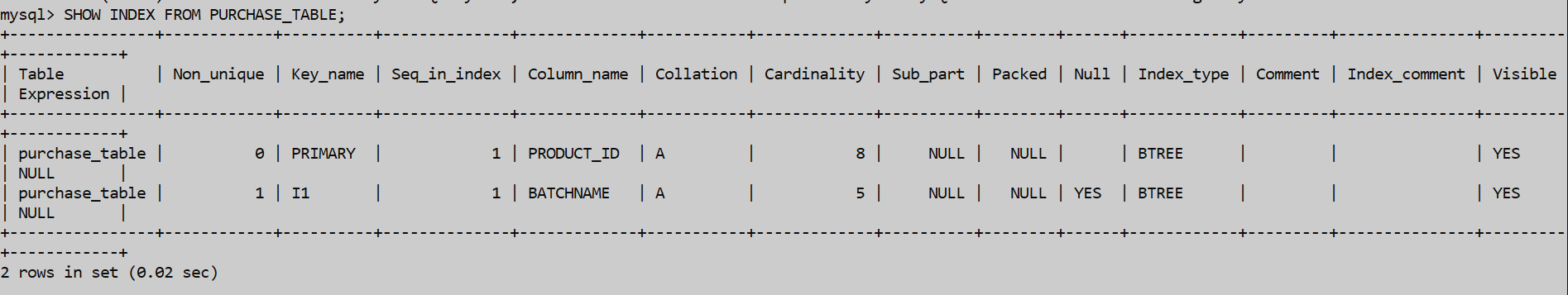
# **APPLICATIONS OF THE SQL QUERIES AND COMMANDS IMPLEMENTED IN LAB-11 IN PROJECT**

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;

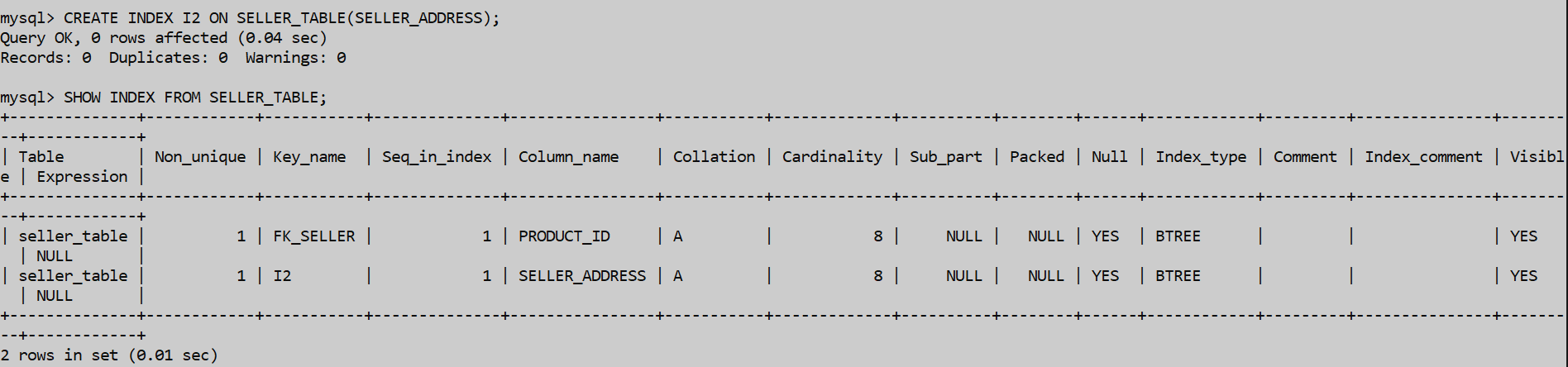




1. If we want to create index on seller\_address in seller\_table

CREATE INDEX I2 ON SELLER\_TABLE(SELLER\_ADDRESS);

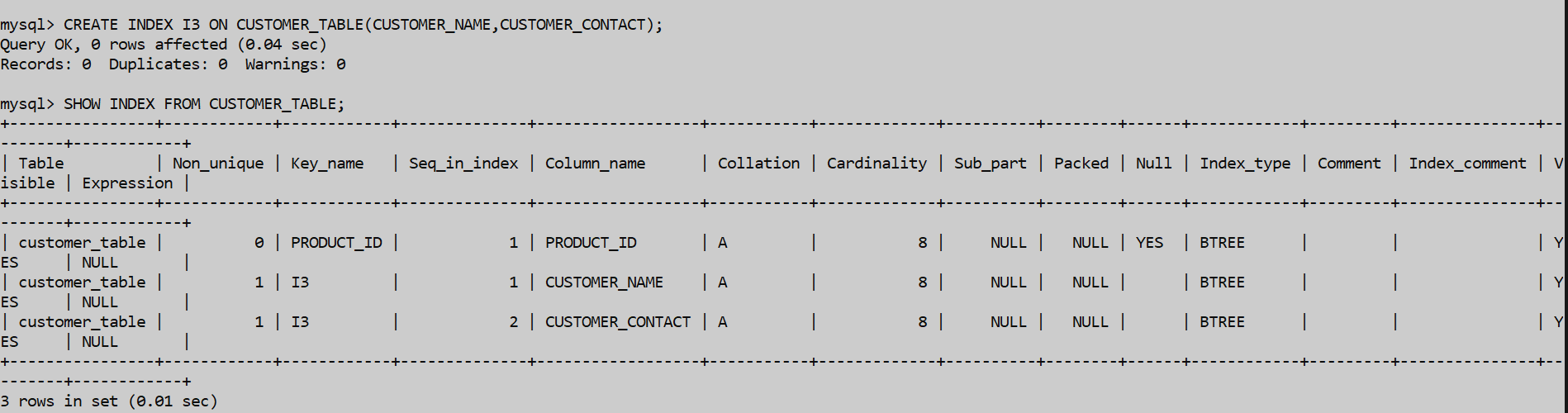
SHOW INDEX FROM SELLER\_TABLE;



1. If we want to form a composite index in the table customer\_table.

CREATE INDEX I3 ON CUSTOMER\_TABLE(CUSTOMER\_NAME,CUSTOMER\_CONTACT);

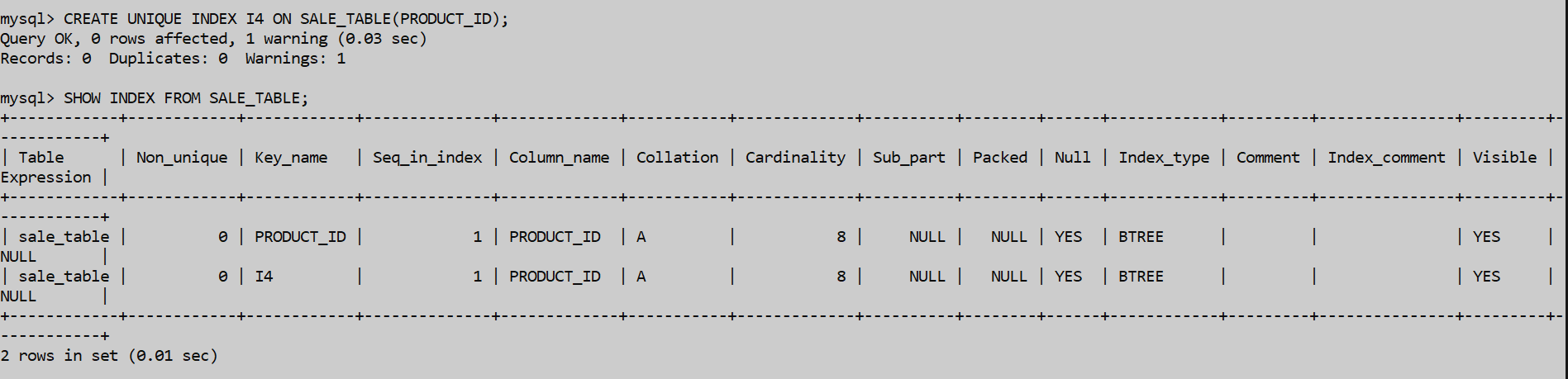
SHOW INDEX FROM CUSTOMER\_TABLE;



1. If we want to form unique index on sale\_table on product\_id.

CREATE UNIQUE INDEX I4 ON SALE\_TABLE(PRODUCT\_ID);

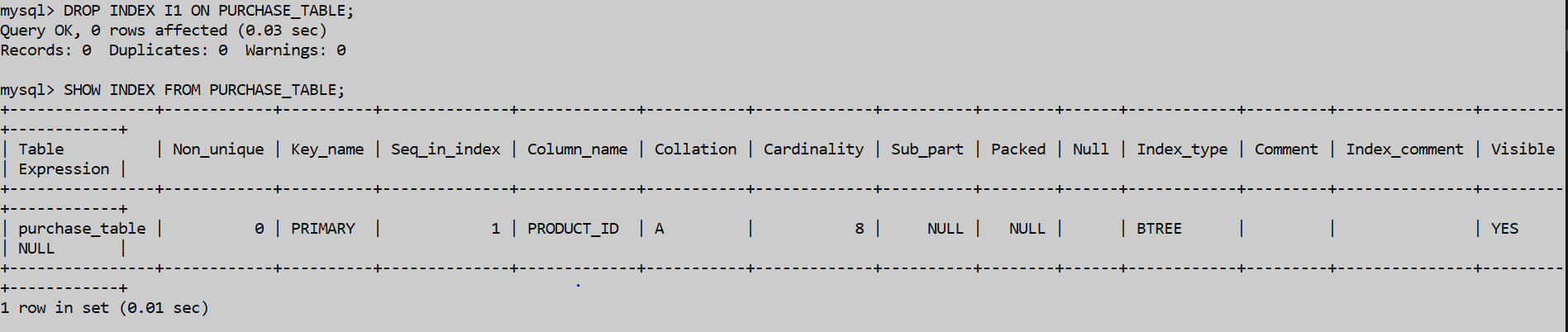
SHOW INDEX FROM SALE\_TABLE;



1. Now we will use the drop command to drop the table.

DROP INDEX I1 ON PURCHASE\_TABLE;

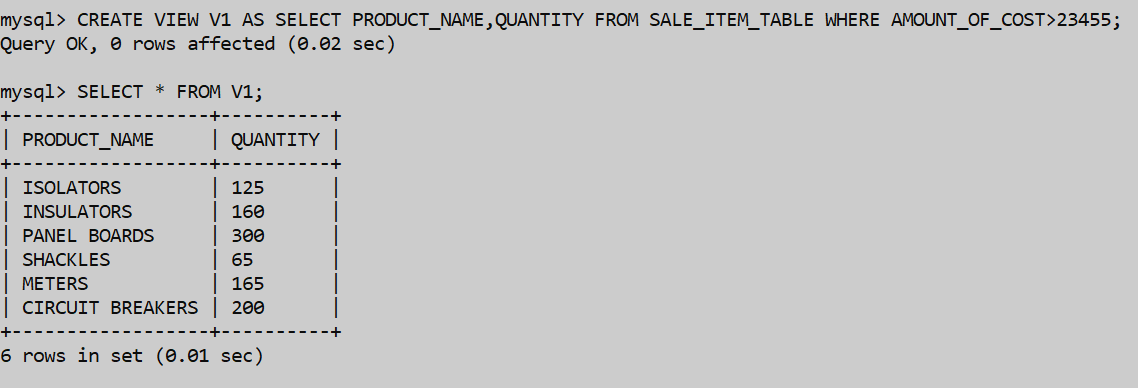
SHOW INDEX FROM PURCHASE\_TABLE;



1. 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;

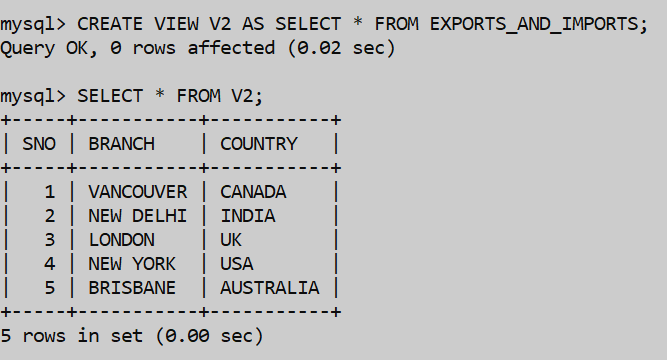
SELECT \* FROM V1;



1. 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;



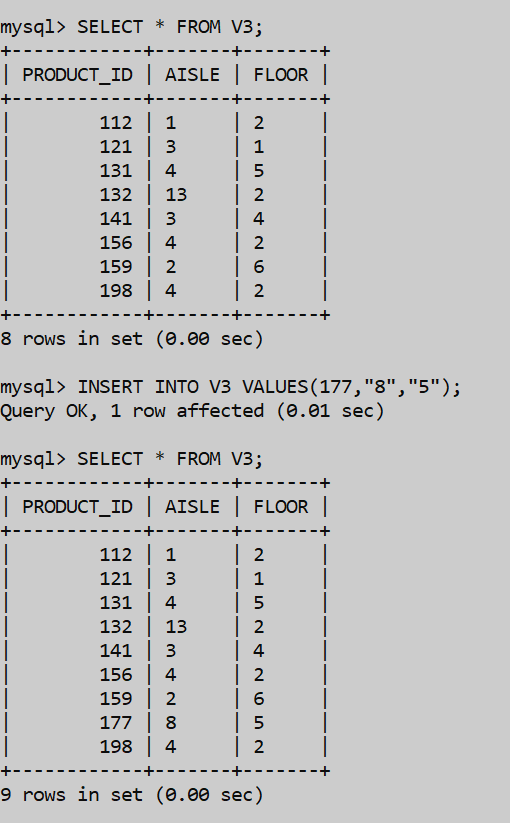
1. Insert command will be used to insert the values in the view v2.

CREATE VIEW V3 AS SELECT \* FROM WAREHOUSE;

SELECT \* FROM V3;

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

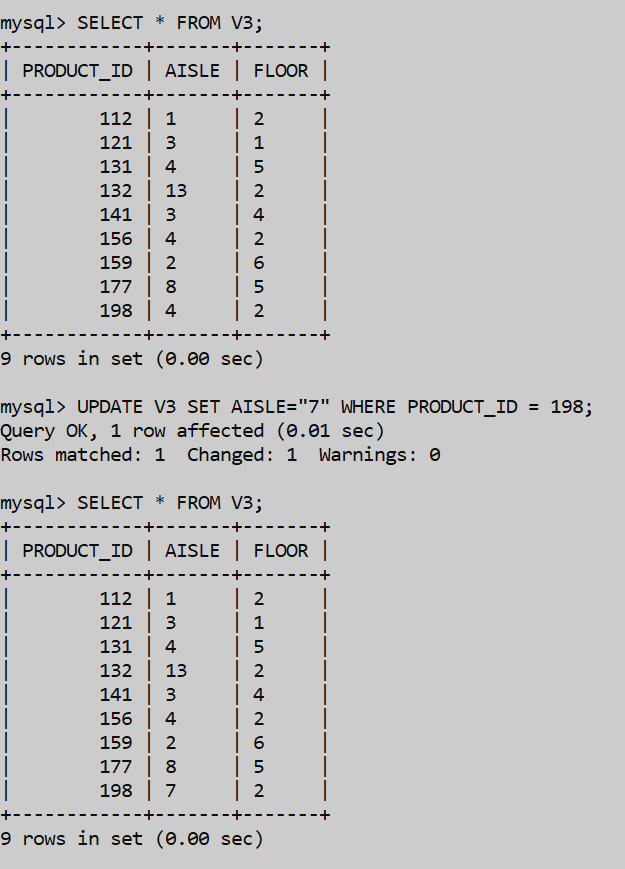
SELECT \* FROM V3;



1. Update command will be used to change the aisle of the product\_id = 198.

UPDATE V3 SET AISLE="7" WHERE PRODUCT\_ID = 198;

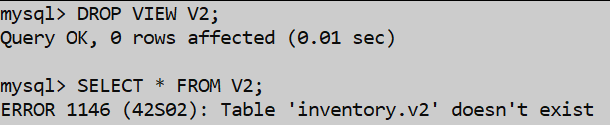
SELECT \* FROM V3;



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

DROP VIEW V2;

SELECT \* FROM V2;



**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 \*\*\***