**INSTITUTE OF ENGINEERING & TECHNOLOGY**

**MOHANLAL SUKHADIA UNIVERSITY, UDAIPUR**

**(RAJ.)**

***DEPARTMENT OF COMPUTER SCIENCE ENGINEERING***

***B. Tech - IV SEMESTER***

****

**Session 2022-23**

**Database Management System Lab**

**LABORATORY MANUAL**

**BT4CS08-CP02**

**Prepared by:**

**Shivam Chouhan**

**INDEX**

|  |  |  |
| --- | --- | --- |
| S.No. | Topic | Page No. |
| 1. | Queries with use of Create, Update, Drop, Truncate, Insert, Delete, Alter and other statements | 3 |
| 2. | To perform SQL Queries by using Between, Not, In, Null, Like | 8 |
| 3. | To perform SQL Queries by using Distinct, Where, Order by, Group by, Having | 11 |
| 4. | To create table shown with given constraints and perform given queries. | 17 |
| 5. | Create a database for employee management system by implementing various constraints. | 22 |
| 6. | To create the view, execute and verify the various operations as view | 25 |
| 7. | To execute and verify the SQL commands for various join operation | 28 |
| 8. | To execute and verify the SQL queries for given tables. | 37 |
| 9. | To implement triggers in DBMS. | 40 |

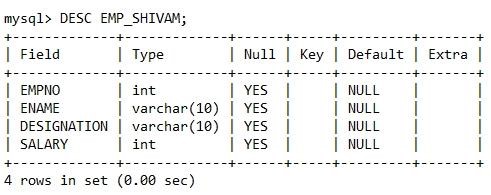
**DBMS PRACTICAL 1**

**Write queries with use of Create, update, drop, truncate, insert, delete, alter, and other statements.**

1. Write a query to create a table employee with empno, empname, designation and salary.



1. Write a query to display all details of employee.



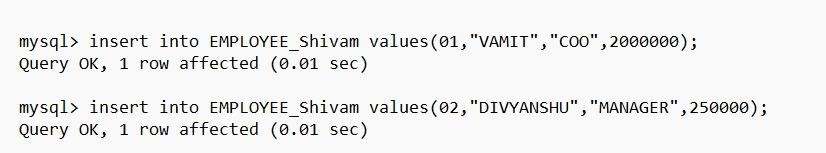
1. Write a query to drop database.



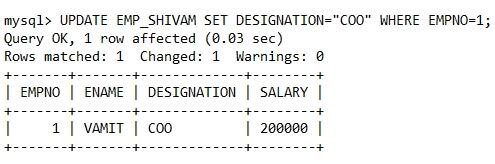
1. Write a query to truncate table.



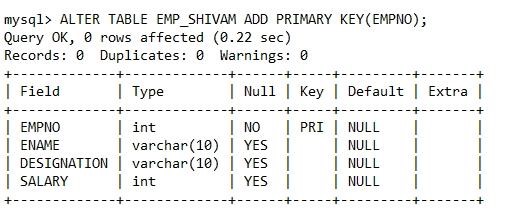
1. Write a query to insert data into employee table.



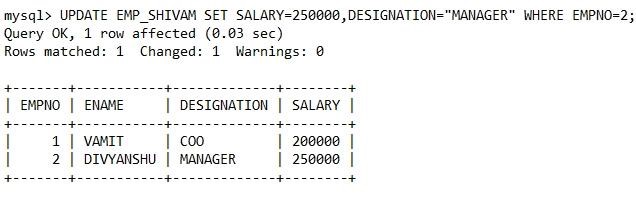
1. Write a query to update designation of VAMIT to “COO”.



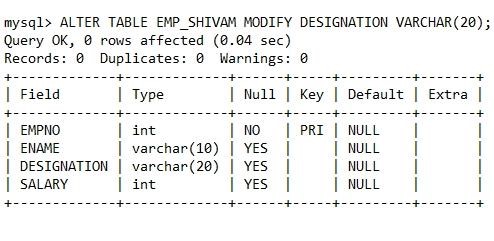
1. Write a query to update multiple record from employee



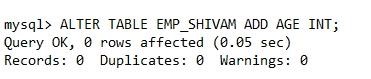
1. Alter table to add primary key.



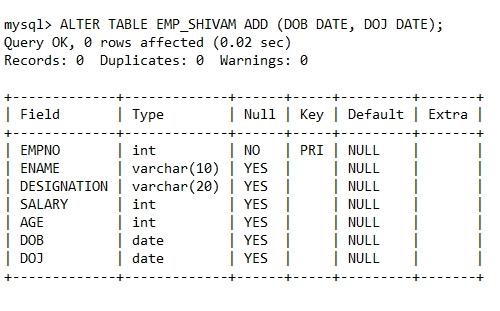
1. Write a query to alter column designation varchar(10) to designation varchar(20).



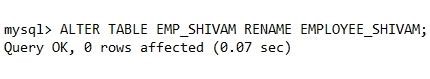
1. Write a query to add a new column to employee.



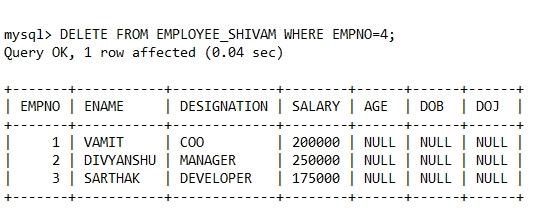
1. Write a query to add multiple columns in employee.



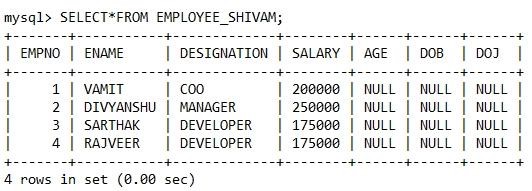
1. Write a query to rename table.



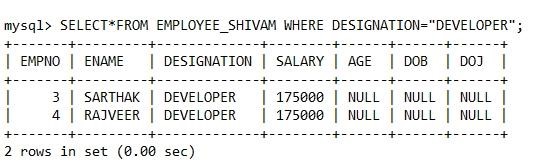
1. Write a query to delete employee with empno=4.



1. List all employee details.



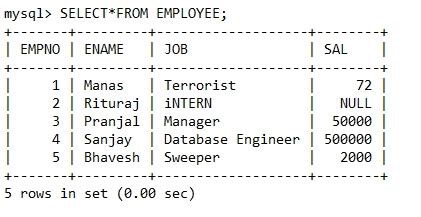
1. List all employee details who are developer.



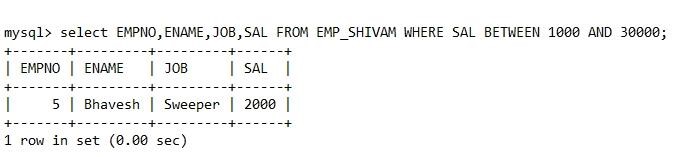
**DBMS PRACTICAL 2**

**Perform SQL Queries by using BETWEEN / NOT / IN / NULL / LIKE**.

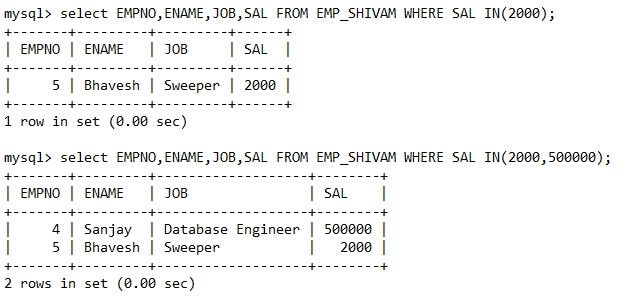
# Table: employee

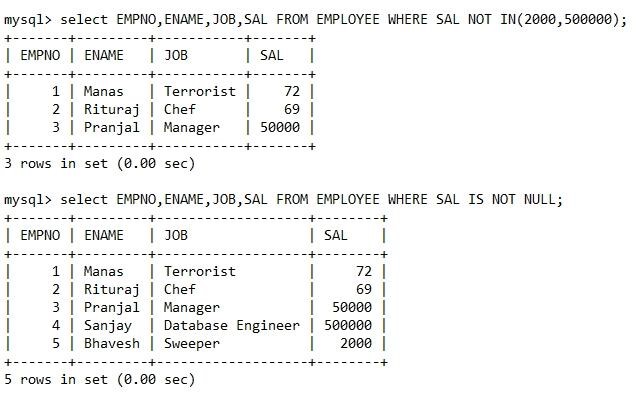


# A. Between

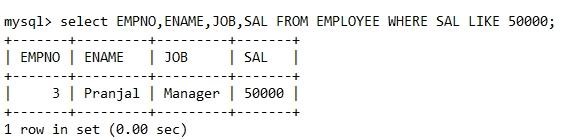


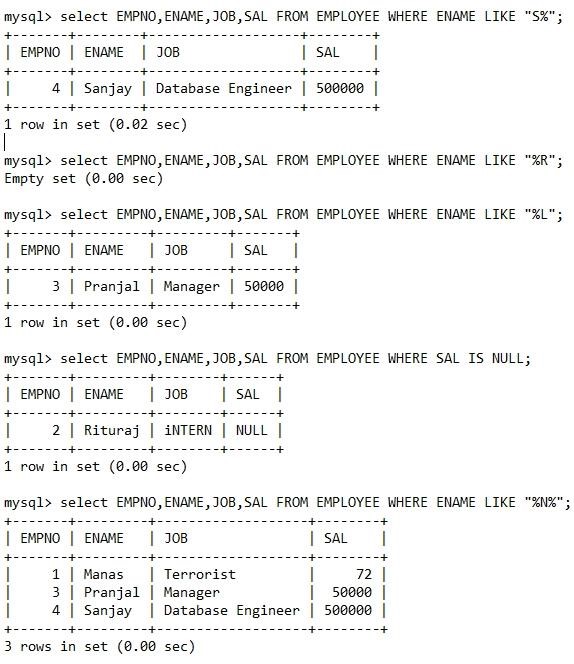
# B. In

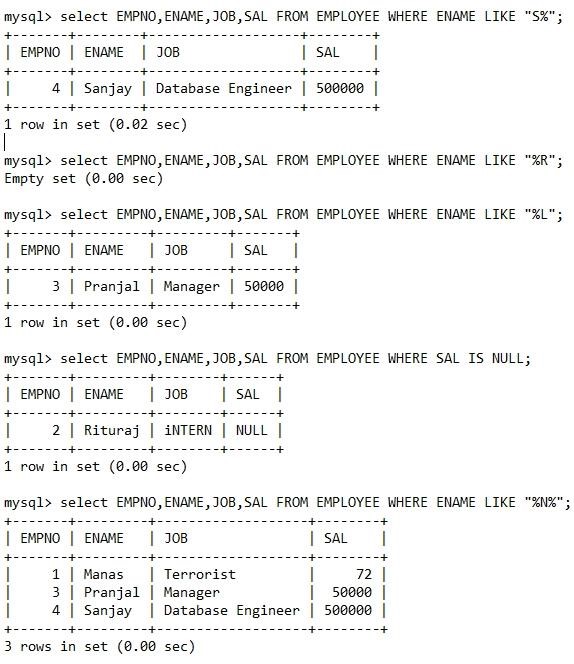


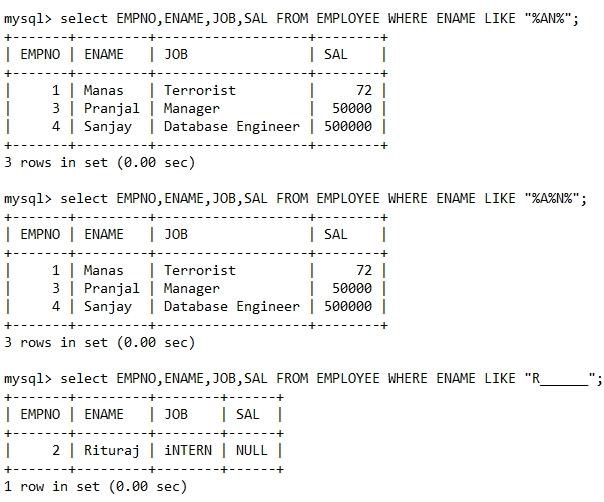


# Like



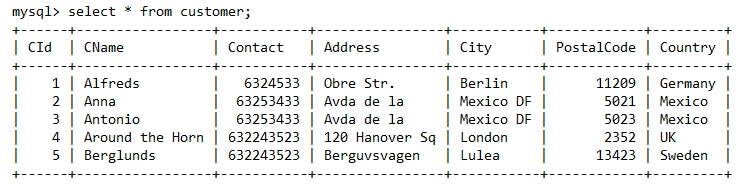






**DBMS PRACTICAL 3**

**Table : Customer**

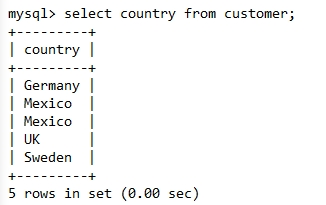
****

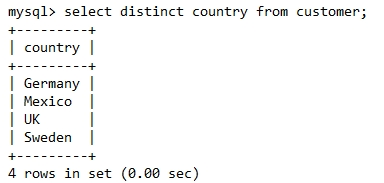
1. **Distinct Statement**

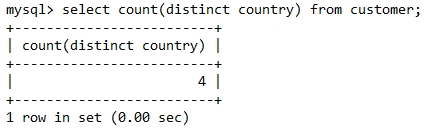
Syntax:

SELECT DISTINCT column1, column2, ...

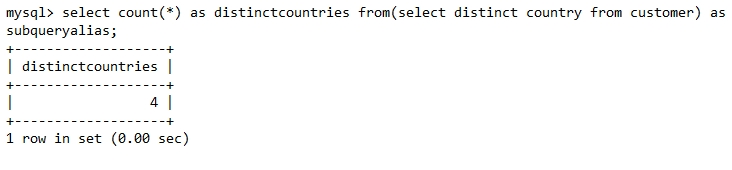
FROM table\_name;

**Example 1**

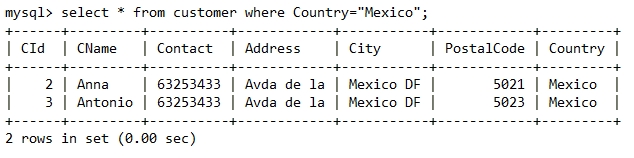
**Example 2**

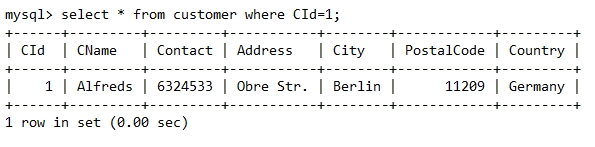
**Example 3**

**Example 4**

****

1. **Where Clause**

**Example 1**

**Example 2**

1. **Order by Clause**

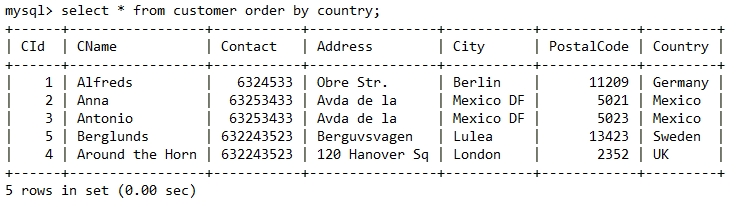
Syntax:

SELECT column1, column2, ...

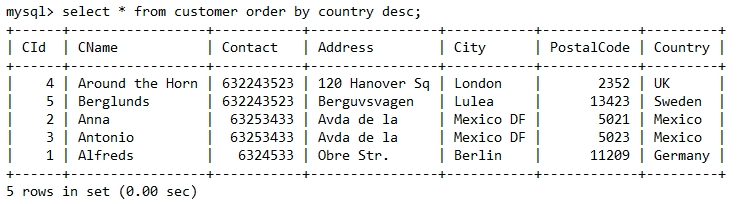
FROM table\_name

ORDER BY column1, column2, ... ASC/DESC;

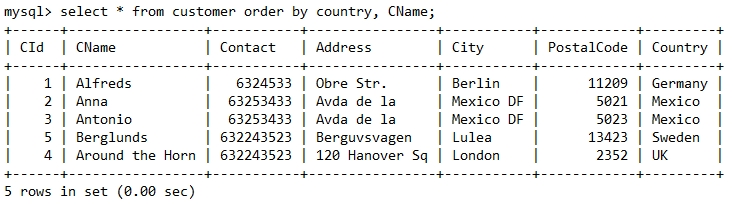
**Example 1**

****

**Example 2: Order by descending Order**

****

**Example 3: Using several columns**

**(A)**

**(B)**

1. **Group by Clause**

Syntax:

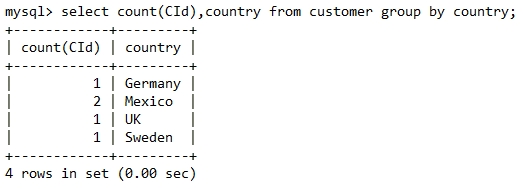
SELECT column\_name(s)

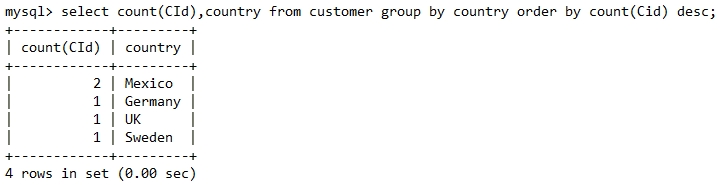
FROM table\_name

WHERE condition

GROUP BY column\_name(s)

ORDER BY column\_name(s);

**Example 1:**

**Example 2: Using Group by and Order by together**

1. **Having Clause**

Syntax:

SELECT column\_name(s)

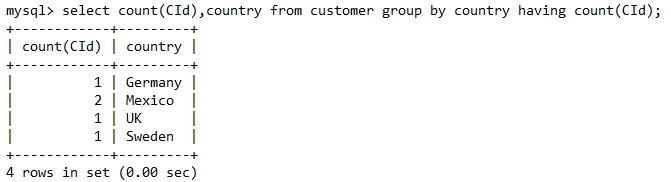
FROM table\_name

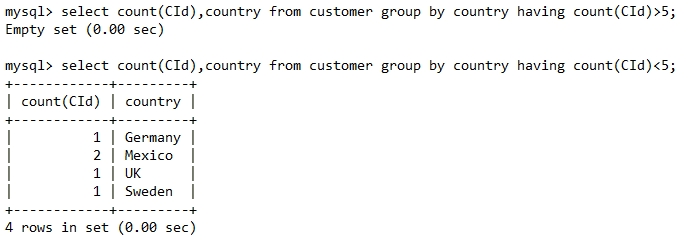
WHERE condition

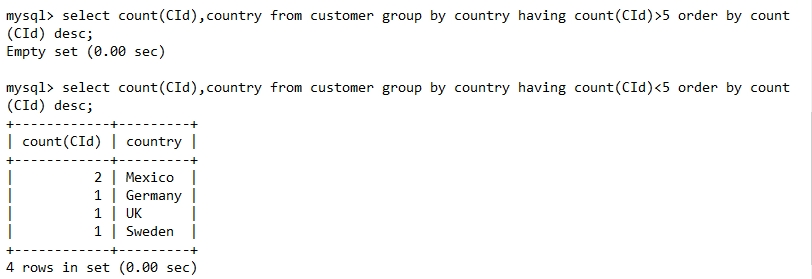
GROUP BY column\_name(s)

HAVING condition

ORDER BY column\_name(s);

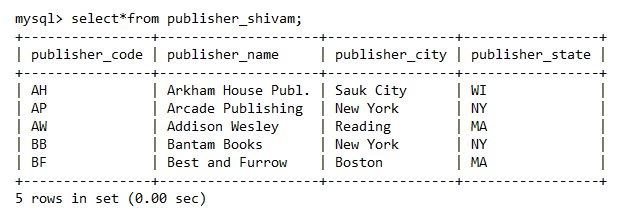
**Example 1:**

**Example 2:**

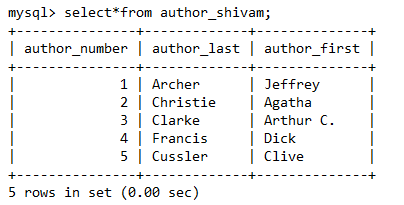
**Example 3:**

**DBMS PRACTICAL 4**

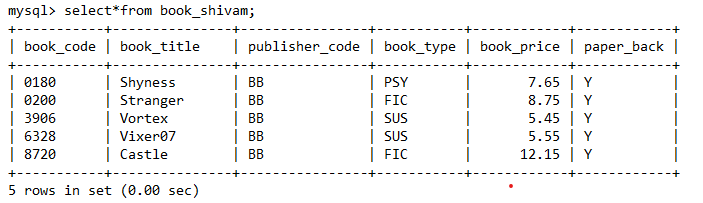
**Table : publisher\_shivam**

****

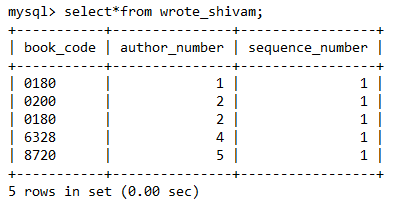
**Table :** **author\_shivam**

****

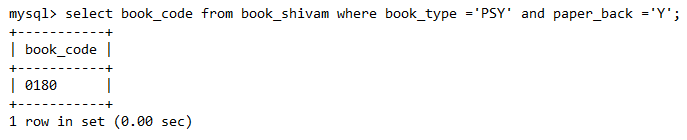
**Table :** **book\_shivam**

****

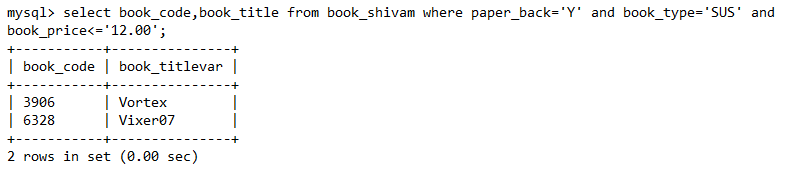
**Table :** **wrote\_shivam**

****

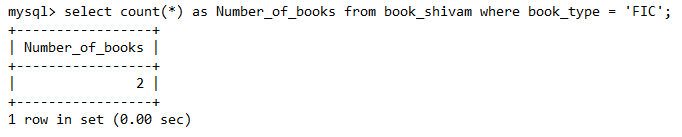
**Query 1**: List the book code of each book where book type is “PSY ‟ AND the paperback.

****

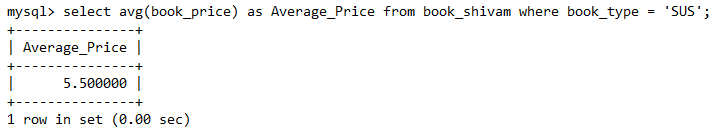
**Query 2**: List the code and title of each paperback book whose type is ART and whose price is less than RS. 12.00.

****

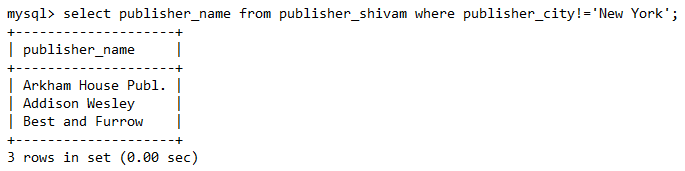
**Query 3:** How many books are of type “FIC‟.



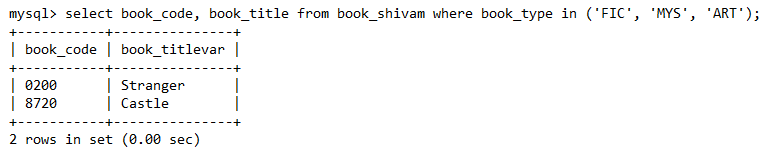
**Query 4**: Find the average price for books of type “SUS”.

****

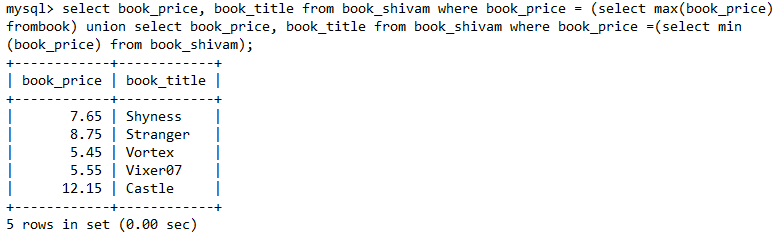
**Query 5**: List the name of every publisher not located in New Year state.



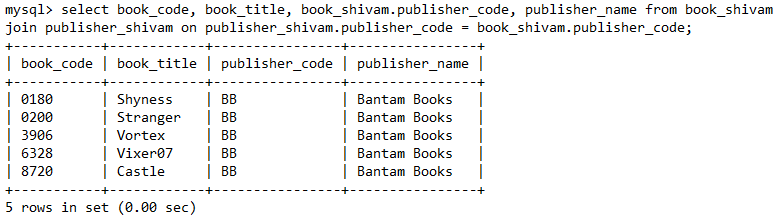
**Query 6**: List the book code and title of every book whose type is FIC, MYS, or ART. Use the IN operator in your command.



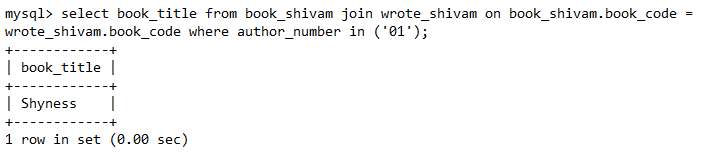
**Query 7**: Determine the name of most expensive, and most cheap book.



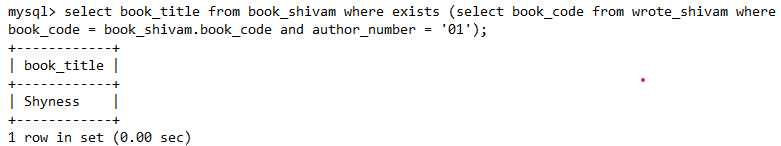
**Query 8:** For every book, list the book code, book title, publisher code, and publisher name.

****

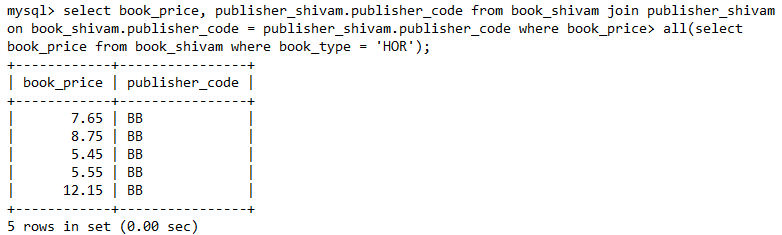
**Query 9**: Find the book title for every book written by author no. 01. Use the IN operator in your formation.



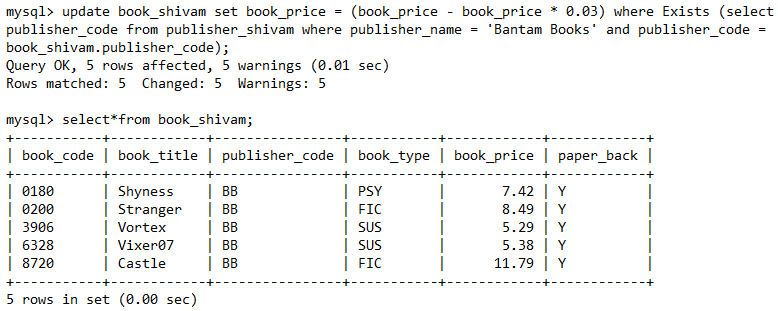
**Query 10**: Repeat 9, but now use exists operator in your formulation.



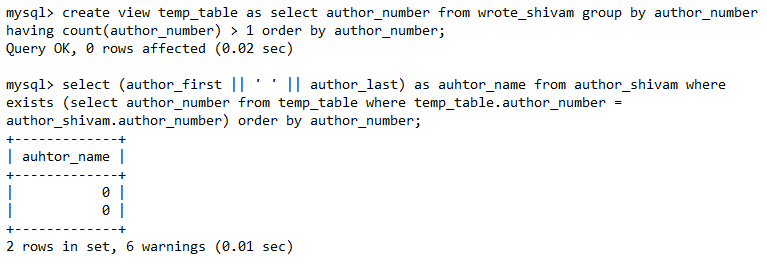
**Query 11**: Find book title and pub code for every book whose price is greater than book price of every book of type HOR (Use ALL).



**Query 12**: Bantam Books has decreased the price of its books by 3%. Update the price in Books Table.



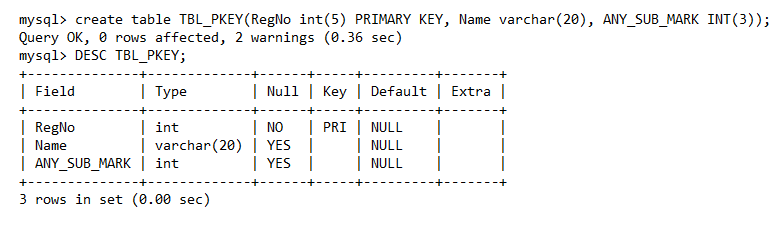
**Query 13**: Using EXIST or NOT EXIST, write a query to select name of author who have written more than one book.

****

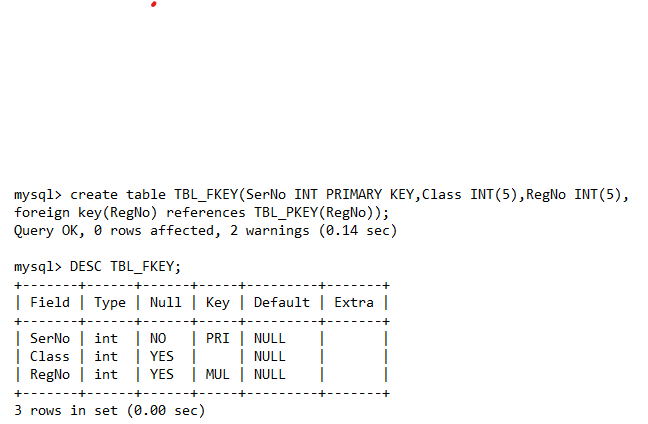
**DBMS PRACTICAL 5**

**Create a database for employee management system by implementing various constraints.**

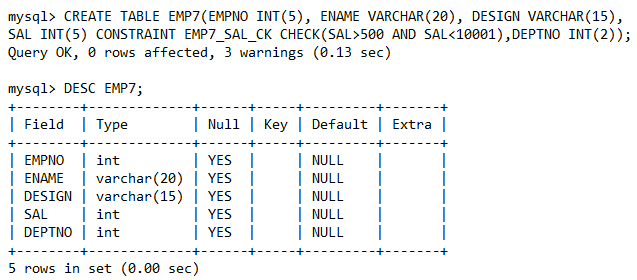
1. **Primary Key**

****

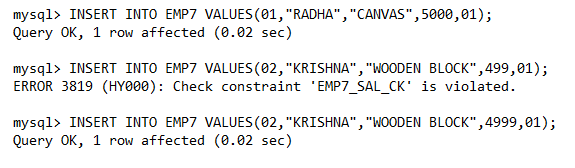
1. **Foreign Key**

****

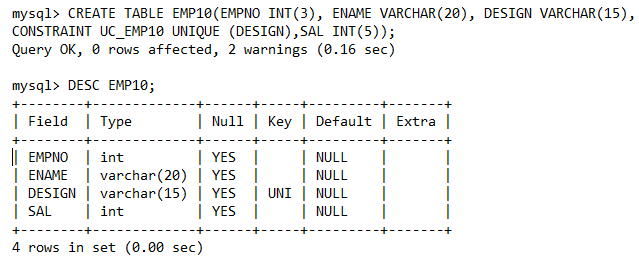
1. **Check**

****

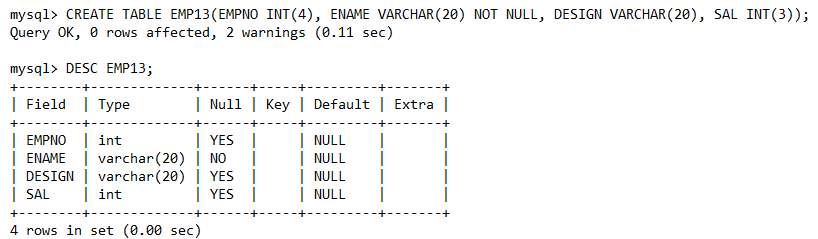
Example



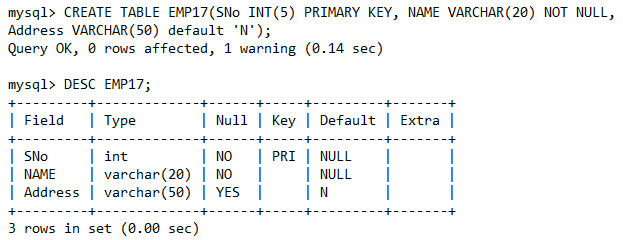
1. **Unique**

****

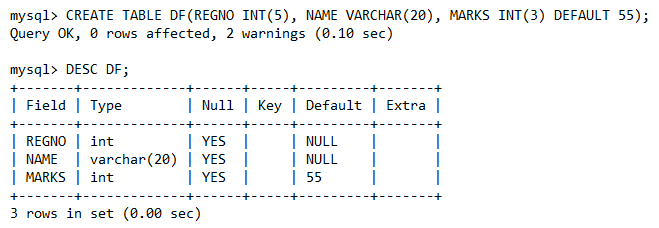
1. **Not Null**

****

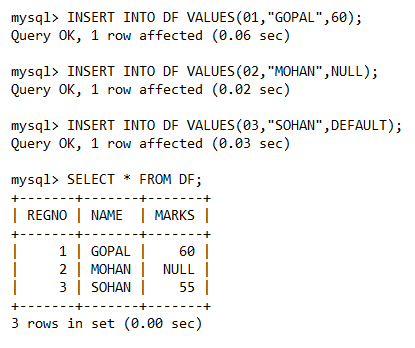
1. **Null**

****

1. **Default**

****

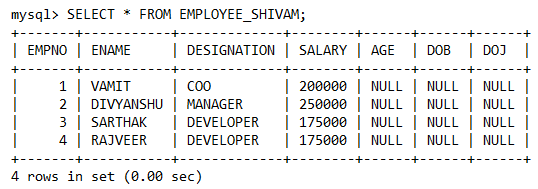
Example



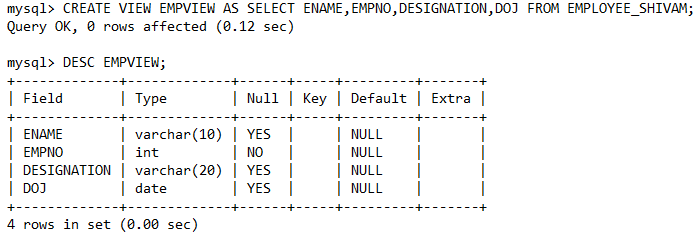
**DBMS PRACTICAL 6**

**To create the view, execute and verify the various operations as view.**

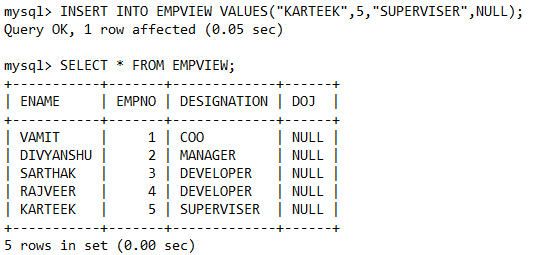
Existing table:EMPLOYEE\_SHIVA

****

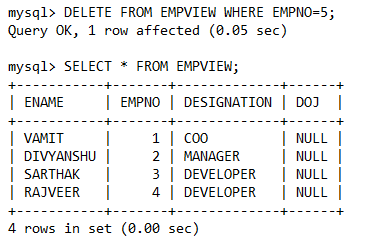
1. **Creation of view**



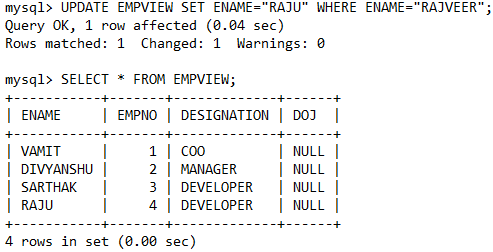
1. **Insertion of values in view**

****

1. **Deletion from view**

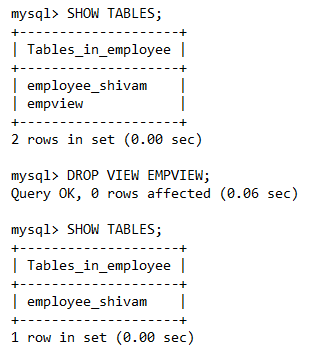
****

1. **Update statement**

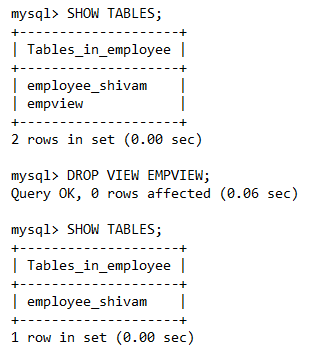
****

1. **Drop a view**

Tables in database before using drop command:

****

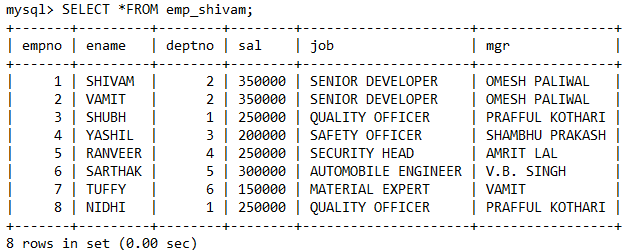
Tables in database after using drop command:



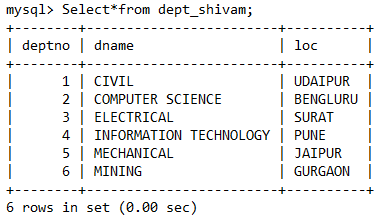
**DBMS PRACTICAL 7**

**To execute and verify the SQL commands for various join operation.**

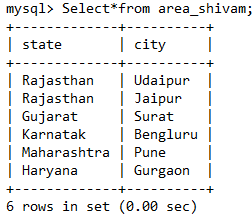
**Table: emp\_shivam**

****

**Table: dept\_shivam**

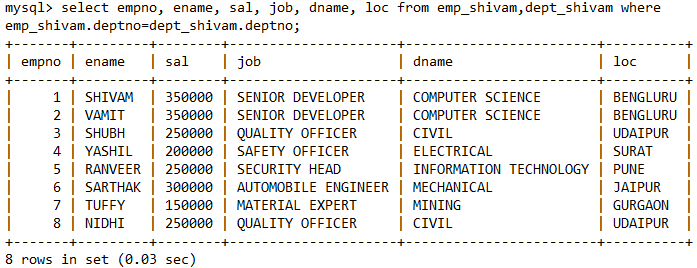
****

**Table: area\_shivam**

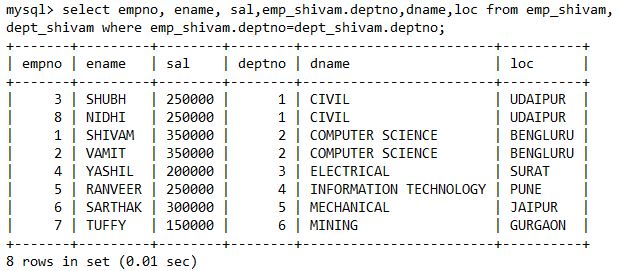
****

1. **Equi Join**

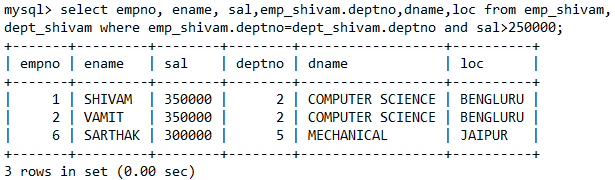
**Query 1:**

****

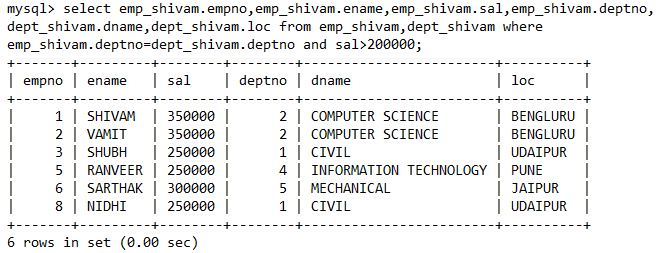
**Query 2:**



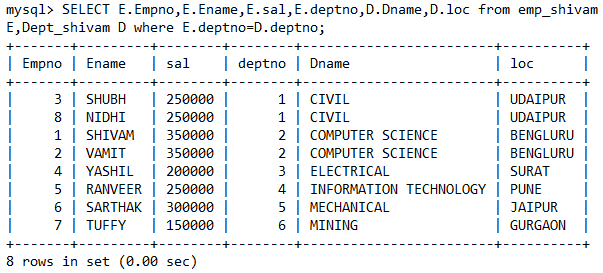
**Query 3:**



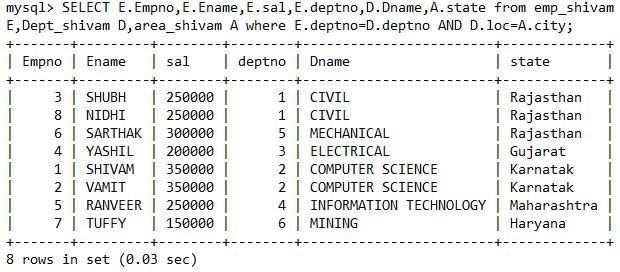
**Query 4:**



**Query 5:**

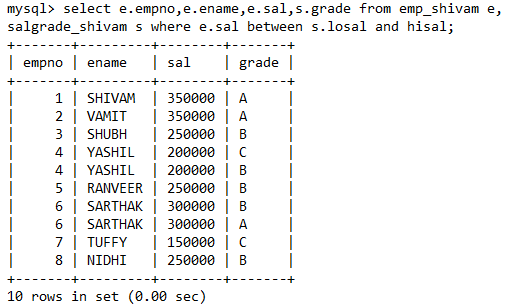


**Query 6:**

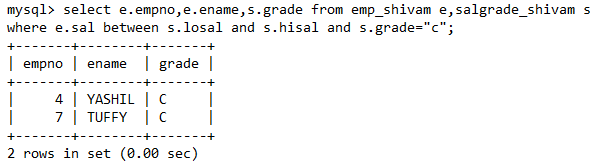


1. **Non Equi Join**

**Query 1:**

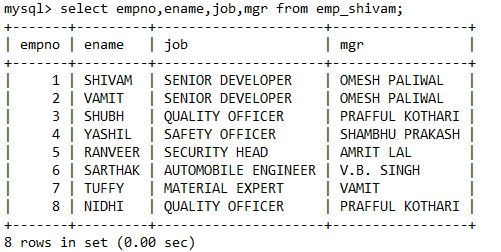
****

**Query 2:**

****

**Self Join**

**Query 1:**

****

**Query 2:**

****

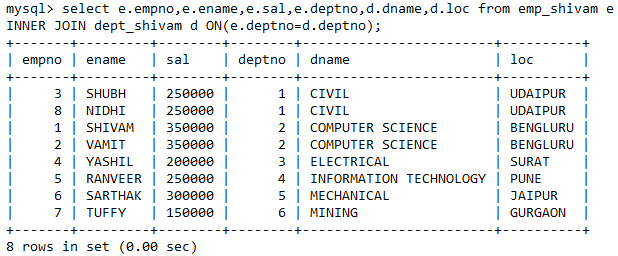
1. **Cartesian Product**

**Query:**

****

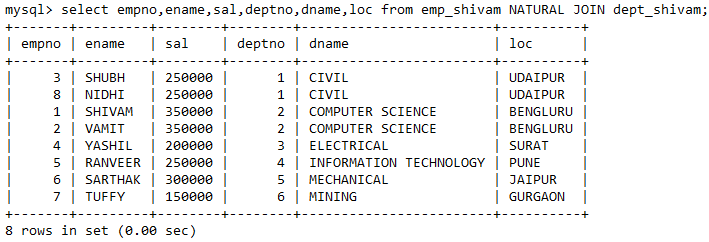
1. **Inner Join**

**Query :**

****

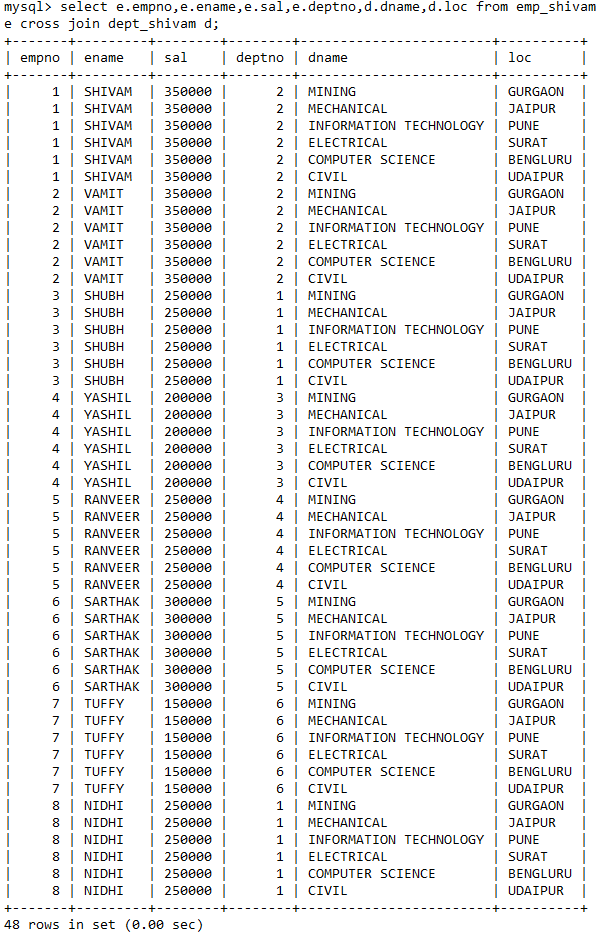
1. **Natural Join**

**Query :**

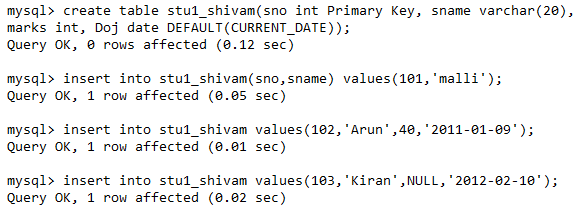
****

1. **Cross Join**

**Query :**

****

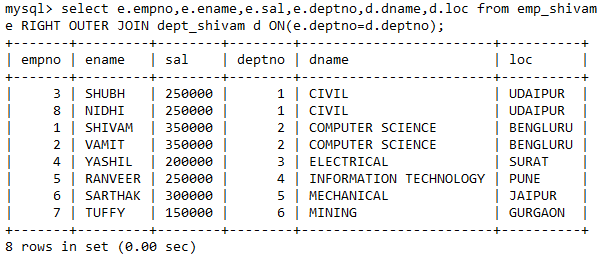
**Default:**

****

1. **Outer Join**

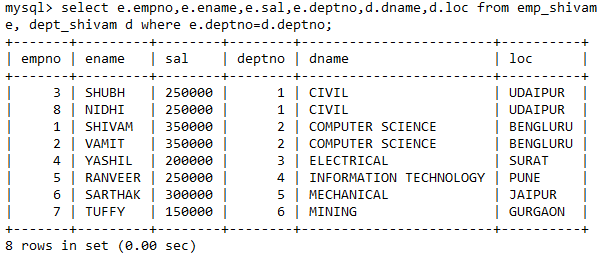
**7.1 Right Outer Join**

**Query :**

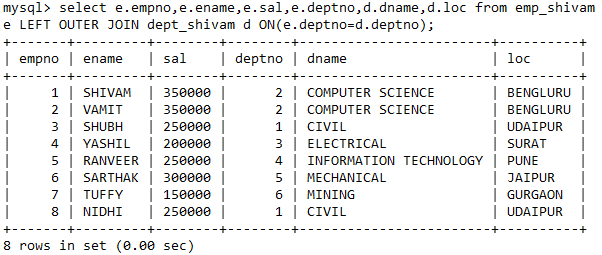
****

**7.2 Left Outer Join**

**Query 1 :**

****

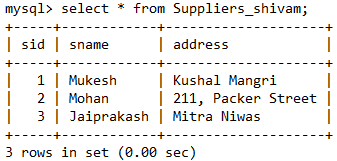
**Query 2 :**

****

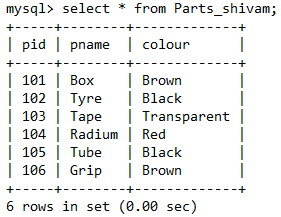
**DBMS PRACTICAL 8**

**To execute and verify the SQL queries for following tables.**

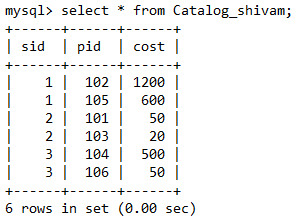
**Table: Suppliers\_shivam**

****

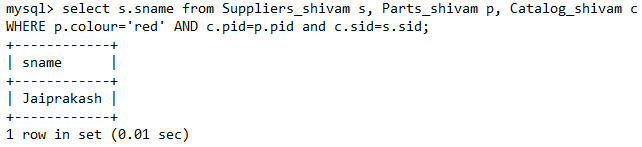
**Table: Parts\_shivam**

****

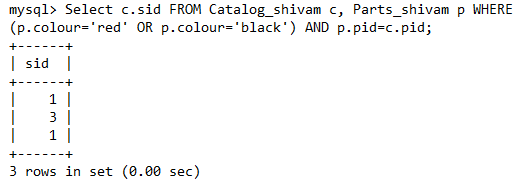
**Table: Catalog\_shivam**

****

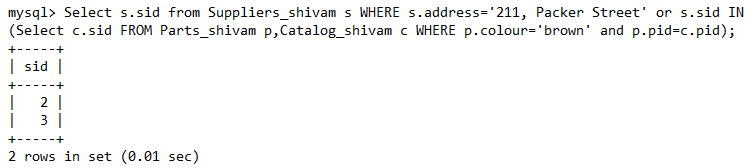
1. Find names of suppliers who supply some red part.



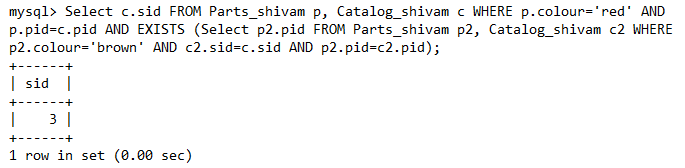
1. Find the sids of suppliers who supply some red or black part.



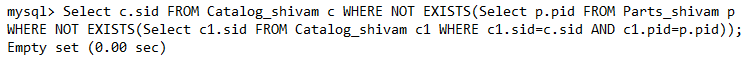
1. Find the sids of suppliers who supply some red part or are at 221, Packer Street.



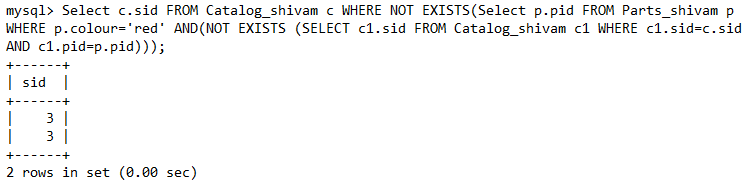
1. Find the sids of suppliers who supply some red part and some brown part.



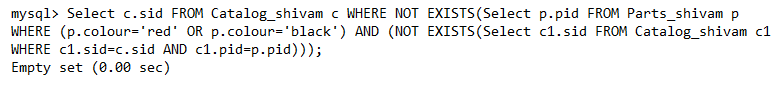
1. Find the sids of suppliers who supply every part.



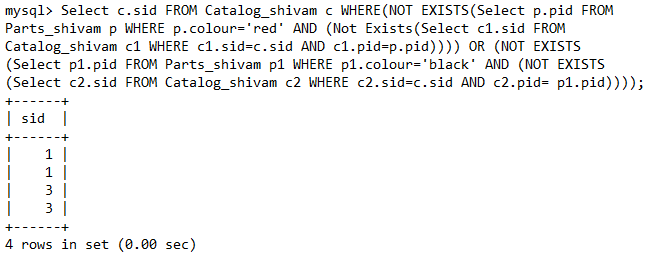
1. Find the sids of suppliers who supply red part.



1. Find the sids of suppliers who supply red part or black part.



1. Find the sids of suppliers who supply every red part or every black part.



1. Find pairs of sids such that the suppliers with the first sid charges more for some part than supplier with the second sid.



**DBMS PRACTICAL 9**

**Write queries to implement triggers in dbms.**

