Ex.No. 1

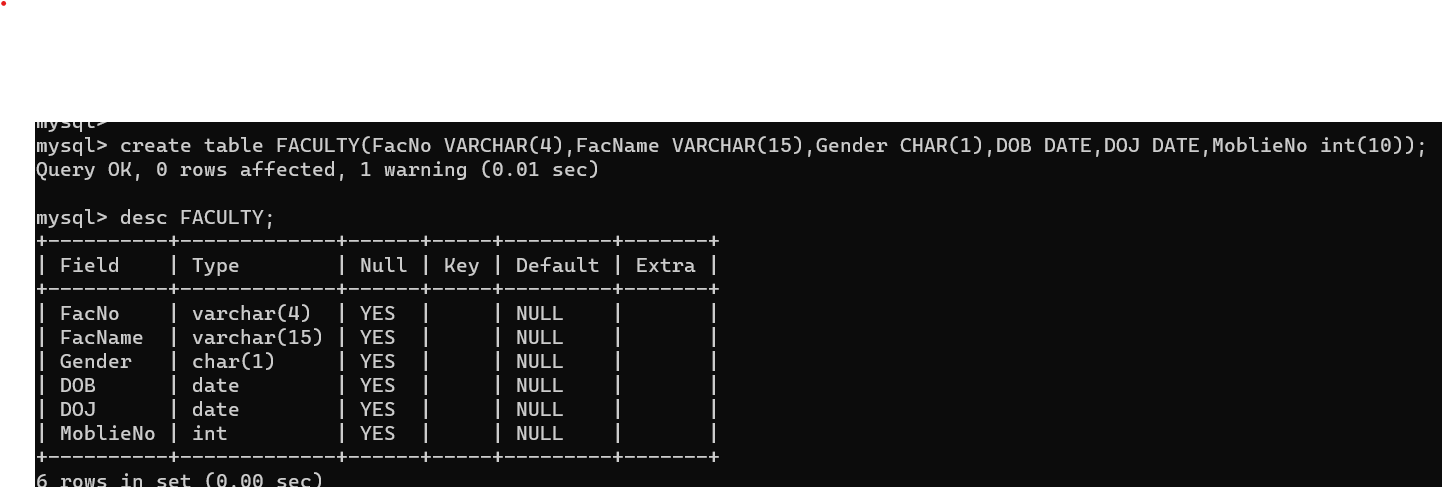
Date:

**DDL Commands – CREATE, ALTER, DROP**

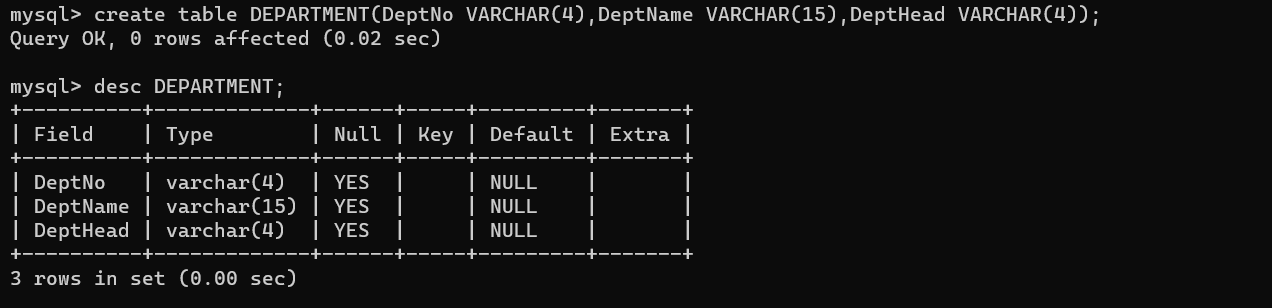
1) Create a table name STUDENT with following structure.



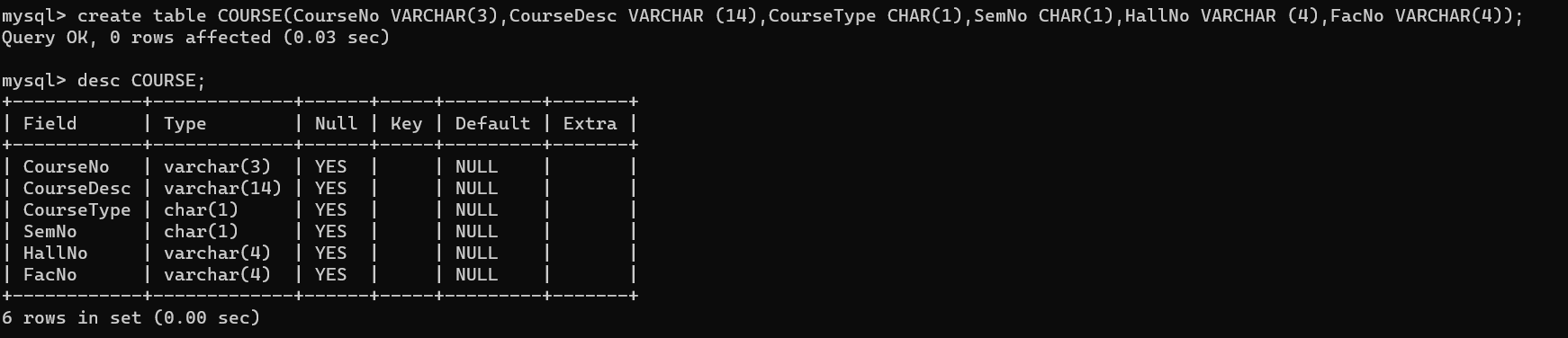
2) Create a table name FACULTY with following structure.



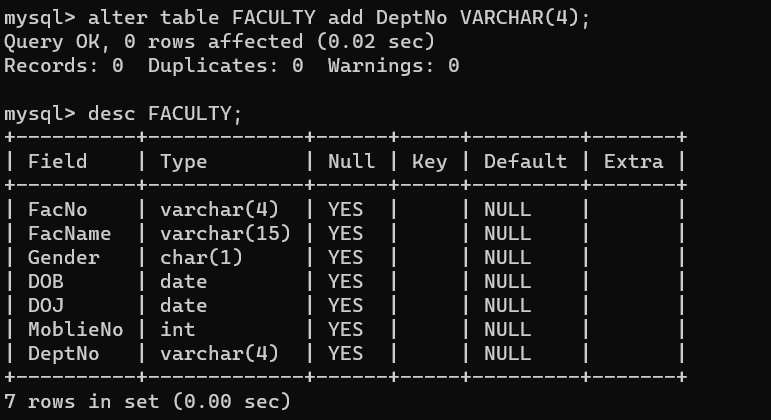
3) Create a table name DEPARTMENT with following structure.



4) Create a table name COURSE with following structure.



1. Modify the table FACULTY by adding a column name DeptNo of datatype VARCHAR(4)

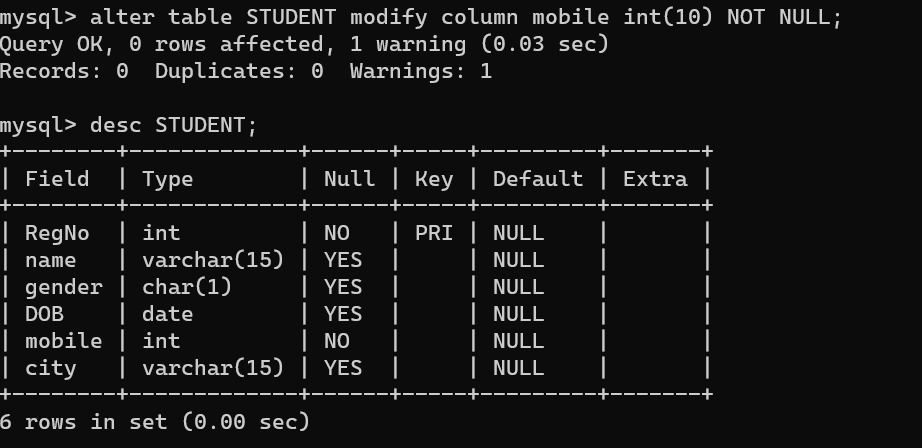
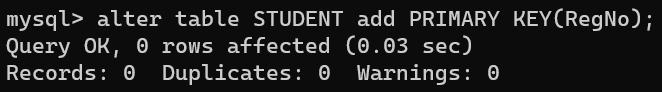


Ex.No. 2

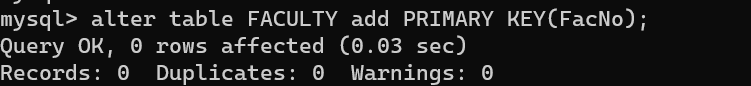
Date:

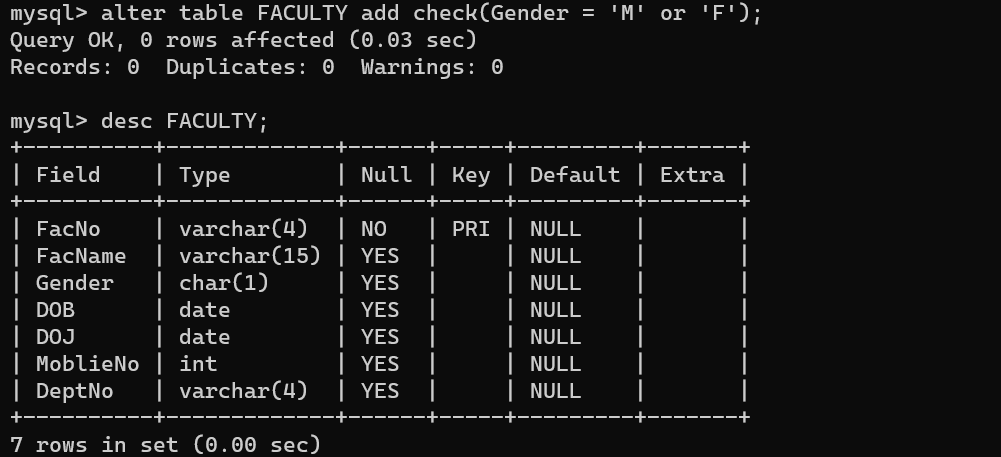
**DDL Commands with Constraints – PRIMARY, FOREIGN KEY, UNIQUE, CHECK**

1) Alter the table STUDENT with following structure.

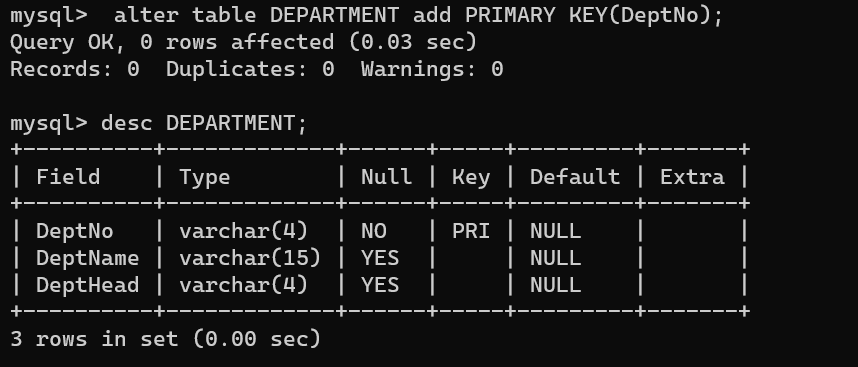


2)Alter the table name FACULTY with following structure. The DeptNo in this table refers the DeptNo in the DEPARTMENT table.

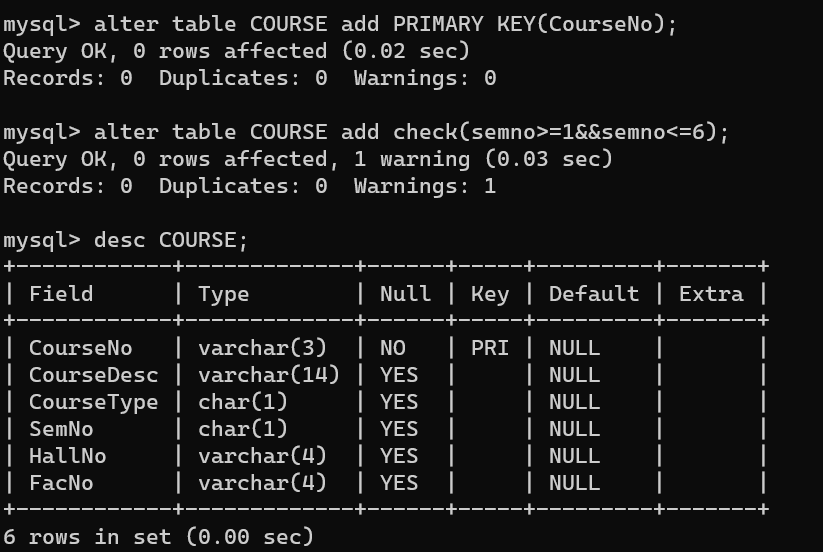




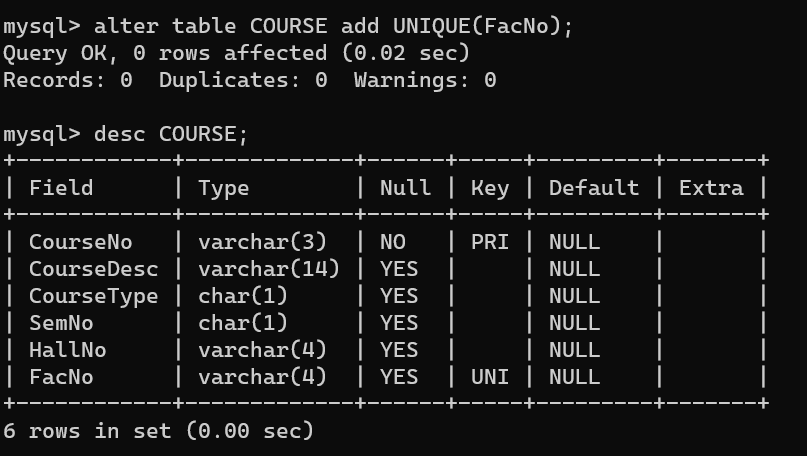
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3) | After the FACULTY table is successfully created, test if you can add a constraint | | | | | | |
| FOREIGN KEY to the DeptNo of this table | | | | | | | |
| 4) | Alter the table name DEPARTMENT with following structure. | | | | | | |
|  |  |  |  |  | |  |  |
|  |  |  |  |  | |  |  |
| # |  | Name |  |  |  |  |  |
|  |  | DeptNo |  | PRIMARY | |  |  |
| 1 |  |  | KEY | |  |  |
|  |  |  |  |  |



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 5) | Alter the table name COURSE with following structure. | | | | | | |  |
|  |  |  |  | |  | | |  |  |
|  |  |  | Column |  | Constraint | | |  |  |
|  | # |  | Name |  |  |  |  |  |  |
|  |  |  | CourseNo |  | PRIMARY | | |  |  |
|  | 1 |  |  | KEY | | |  |  |
|  |  |  |  |  |  |
|  | 2 |  | SemNo |  | 1 to 6 | | |  |  |



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 6) | Alter the table name COURSE with following structure. | | | | | | |
|  |  |  | |  | | |
|  |  | Column |  | Constraint | | |
| # |  | Name |  |  |  |  |
|  |  | FacNo |  | UNIQUE | | |
| 1 |  |  | KEY | | |



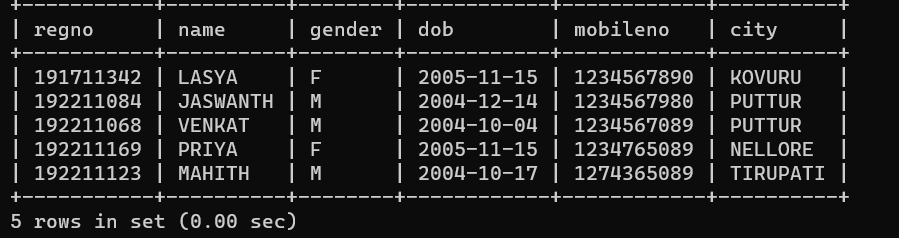
Ex.No :3

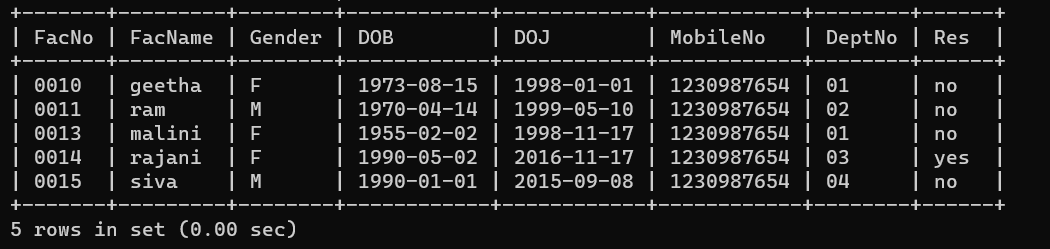
**DML Commands – INSERT, SELECT**

**Aim:**

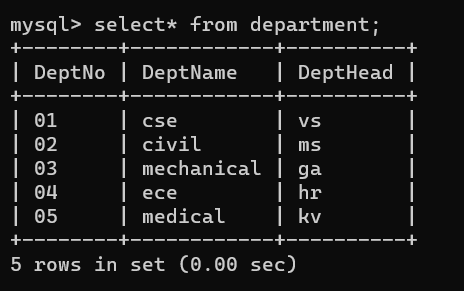
To perform Data Manipulation Language (DML) Commands such as INSERT, SELECT

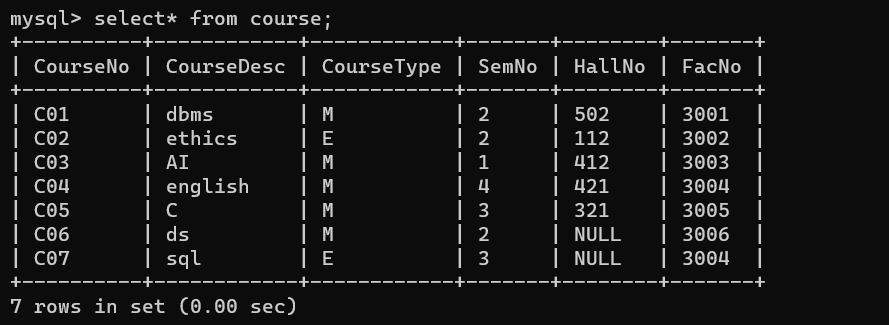
1.Populate all the five tables with your own data.



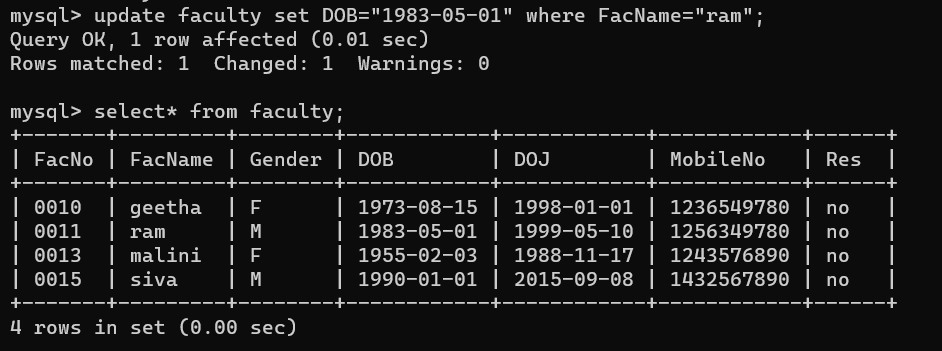


);

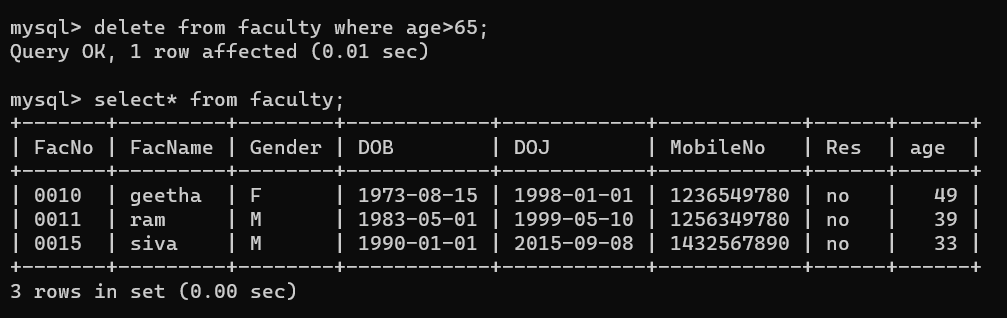




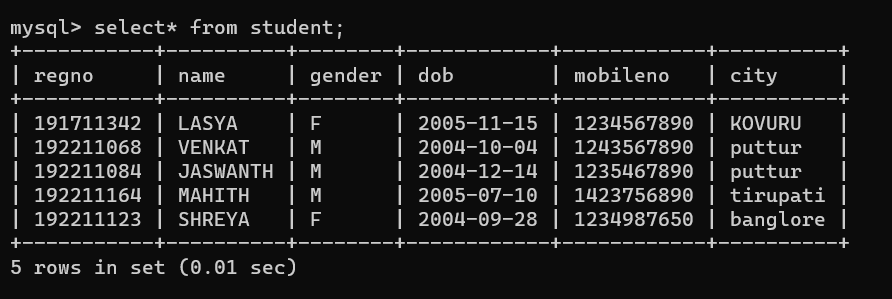
2.Modify the date of birth for the faculty whose name is 'RAM' with a value ‘1983-05-01’.

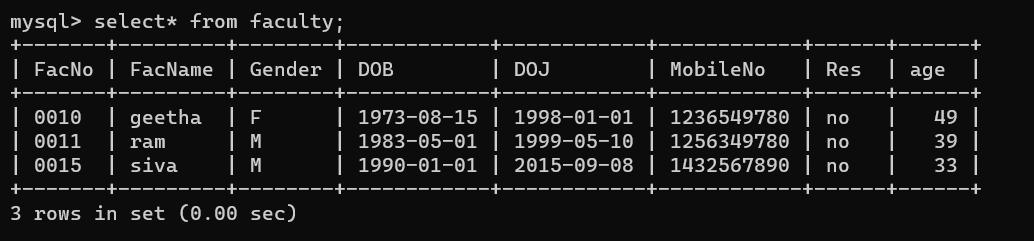


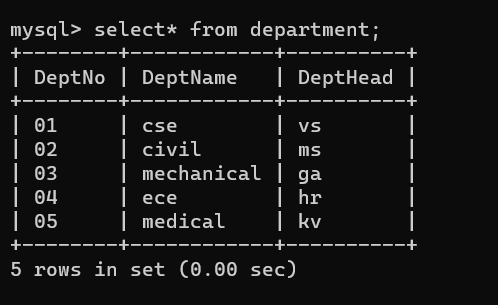
3.Remove all faculty who are having over 65 years.

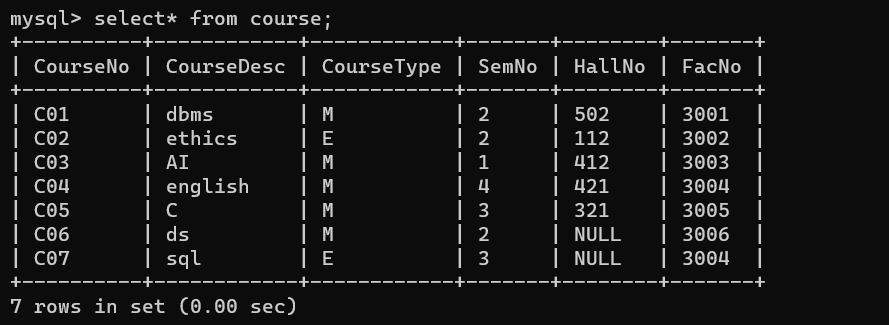


4.View all the records from the five tables. Exercise





****



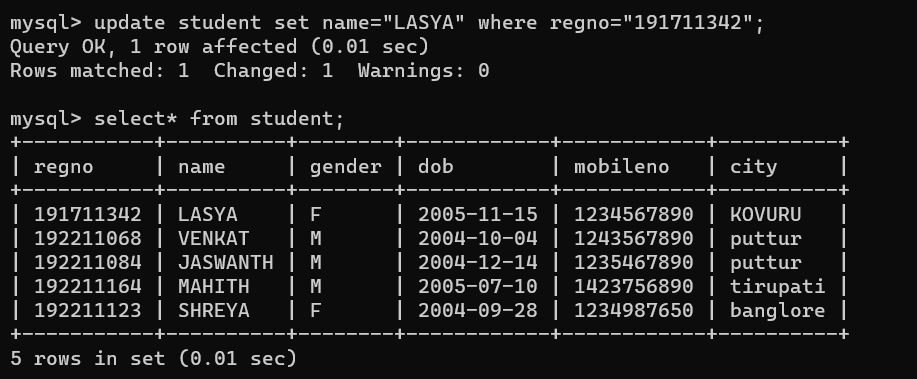
**RESULT**:

Data Manipulation Language (DML) Commands such as INSERT, SELECT are performed in the five tables.

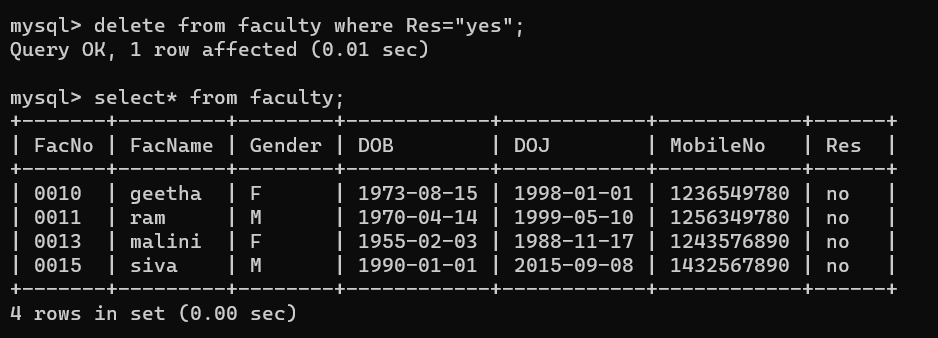
Ex no:4

**DML Commands with Constraints –UPDATE, DELETE**

1.Update the value of student name whose register number is ‘191711342’



2.Delete the record in the table FACULTY, who resigned her job.



**RESULT**:

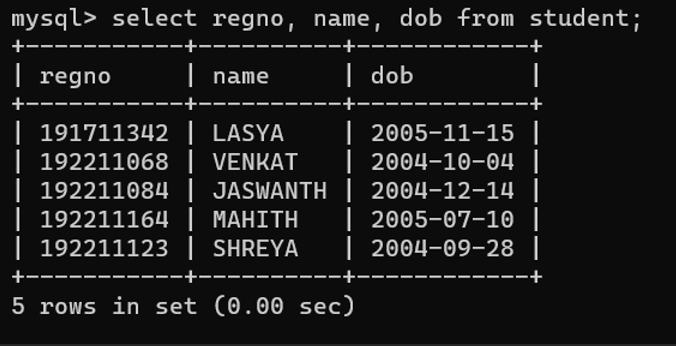
Data Manipulation Language (DML) Commands such as INSERT, SELECT, UPDATE, DELETE are performed in the five tables.

EX NO:5

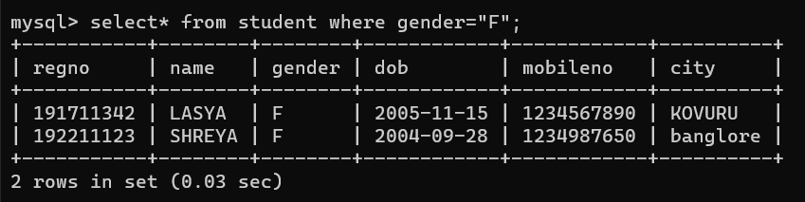
**SELECT with various clause – WHERE, pattern matching**

**WHERE:**

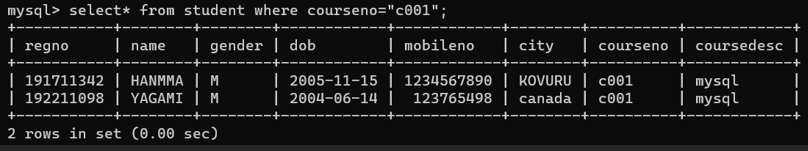
The student counsellor wanted to display the registration number, student name and date of birth for all the students.



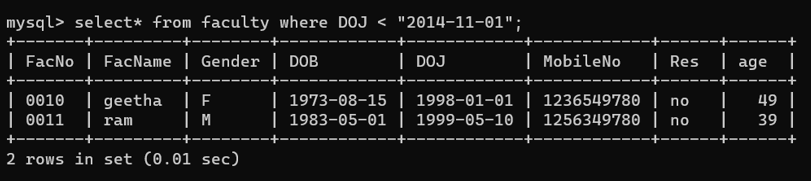
2.The controller of examinations wanted to list all the female students



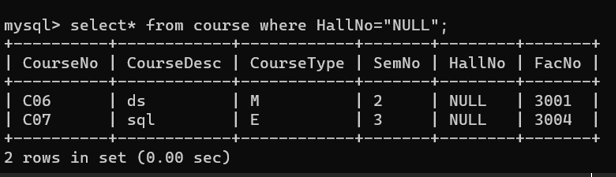
3.Who are the boy students registered for course with the course number “C001“



4.Display all faculty details joined before “November 2014”

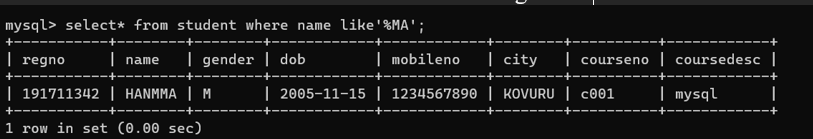


5.Display all the courses not allotted to halls

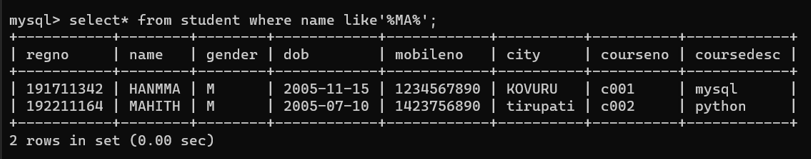


**LIKE:**

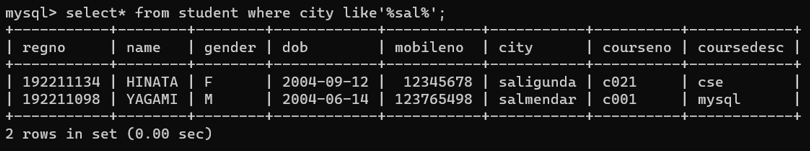
1. List the students whose name ends with the substring “ma”



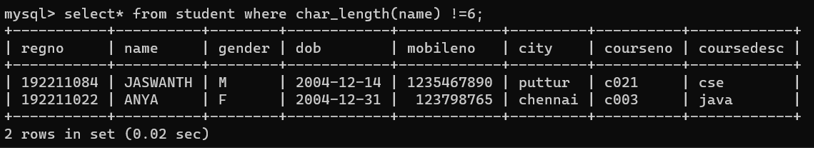
7.Display all students whose name contains the substring “ma”



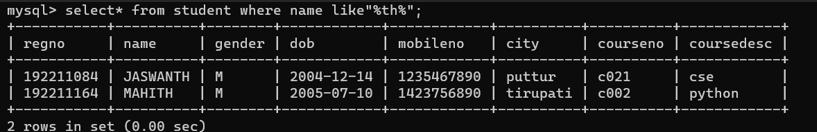
8.Find all the students who are located in cities having “Sal” as substring



9.Display the students whose names do not contain six letters.



10.Find all the students whose names contains “th”



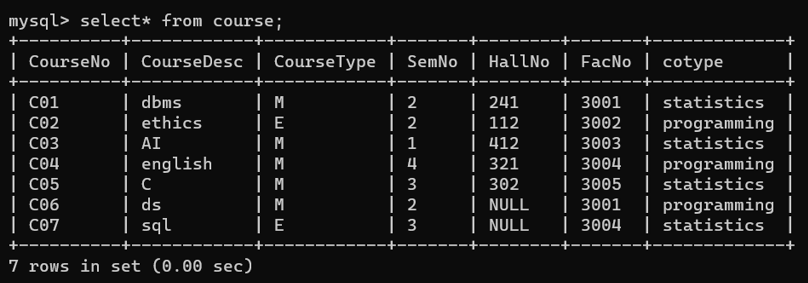
**RESULT**:

The records from the tables are displayed using SELECT commands with WHERE Clause and Pattern matching.

Ex. No: 6

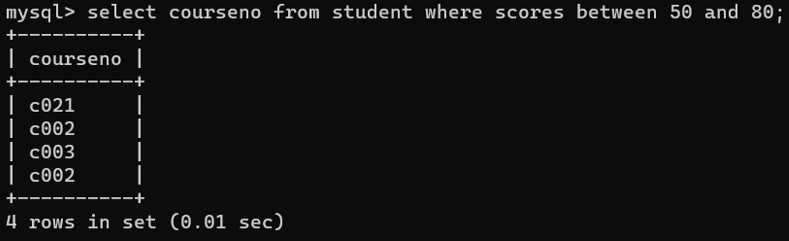
**SELECT with various clause – BETWEEN, IN, Aggregate function**

1. List the type of the courses “Statistics” and “Programming”

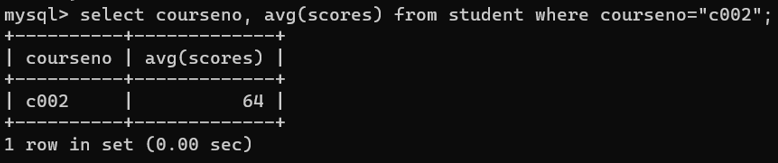


2.The instructor wants to know the CourseNos whose scores are in the range 50

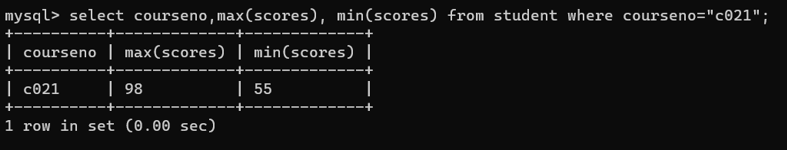
to 80

**AGGREGATE**

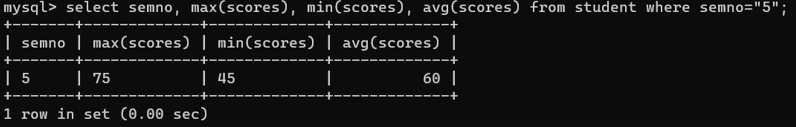
1. Find the average mark of “C002”.



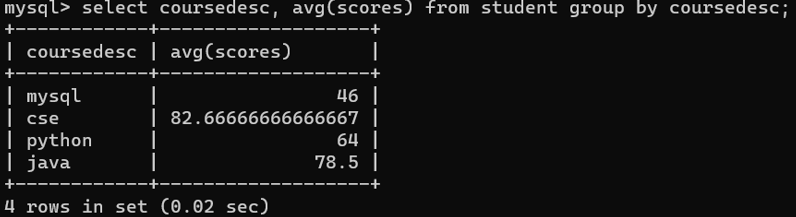
2.List the maximum, minimum mark for “C021”



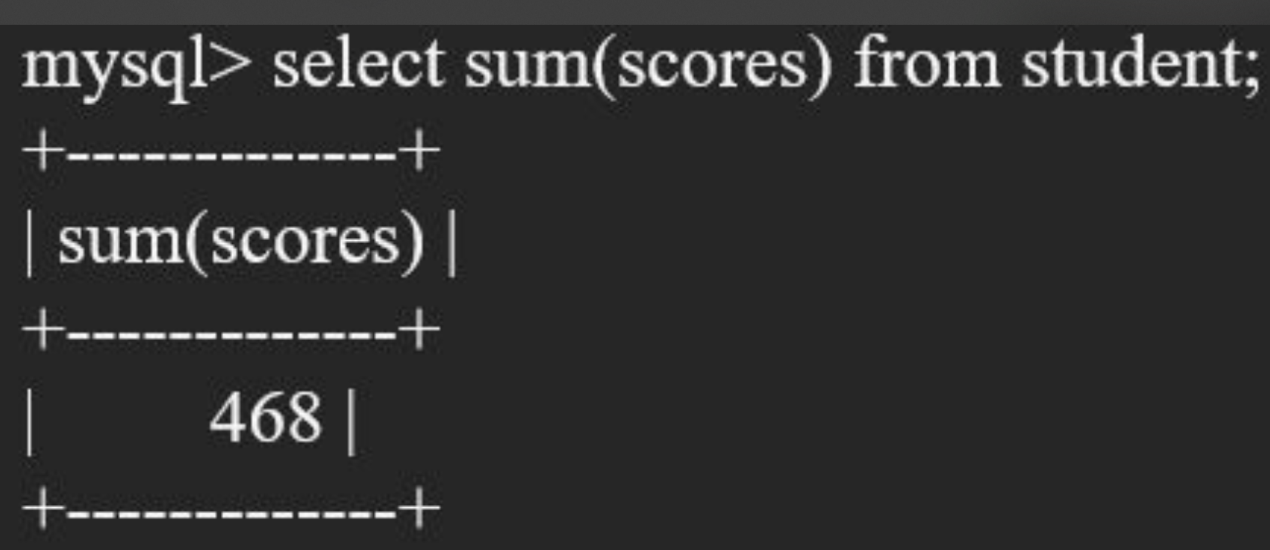
3.List the maximum, minimum, average mark for each subject in 5th semester



4.List the name of the courses and average mark of each courses.

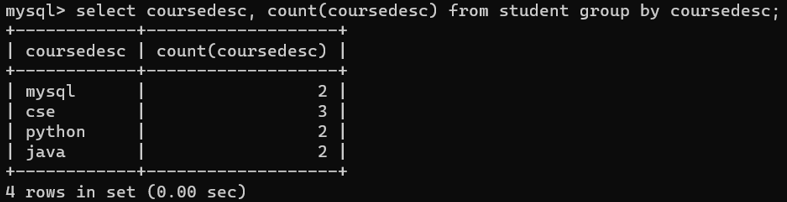


5.Calculate the sum of all the scores.

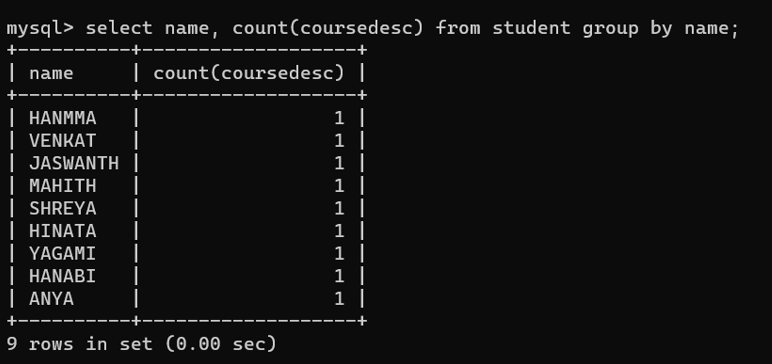


6.How many students are registered for each course? Display the

course description and the number of students registered in each course.



7.How many courses did each student register for? Use Assessment table.



**RESULT**:

The records from the tables are displayed using SELECT commands with WHERE Clause and Pattern matching.

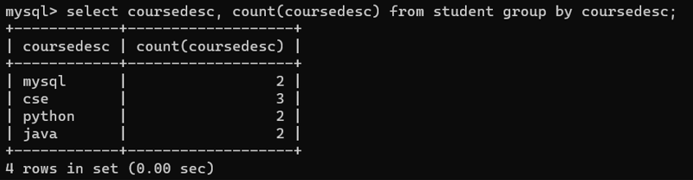
Ex. No.: 7

Date:

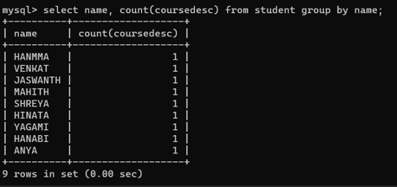
**SELECT with various clause – GROUP BY, HAVING, ORDER BY**

**GROUP BY - HAVING**

1. How many students are registered for each course? Display the course description and the number of students registered in each course.

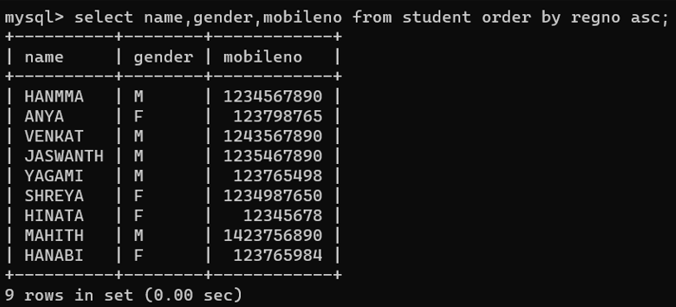


2.How many courses did each student register for? Use Assessment table.

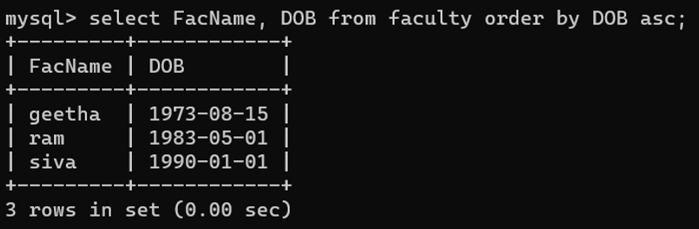


**ORDER BY**

1. Retrieve Name, Gender, MobileNo of all the students in ascending order of RegNo



2.List the faculty members in the order of older faculty first.



**RESULT**:

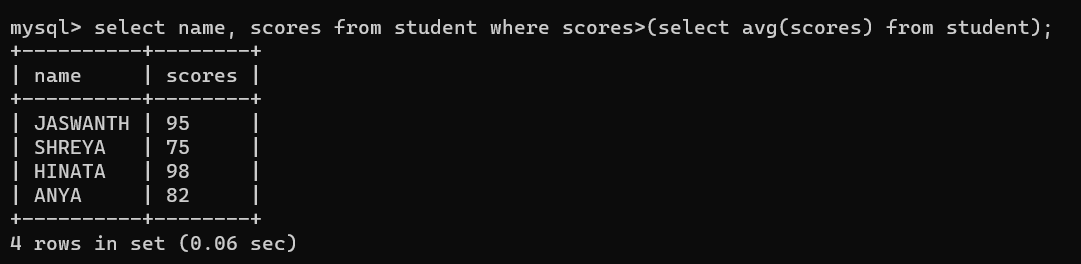
The records from the tables are displayed using SELECT commands with GROUP BY, HAVING and ORDER BY.

**Ex. No.: 8**

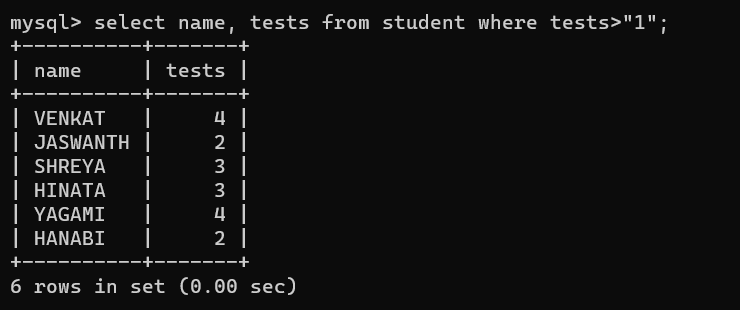
**QUERY WITH SUBQUERY & CORRELATED QUERY**

.

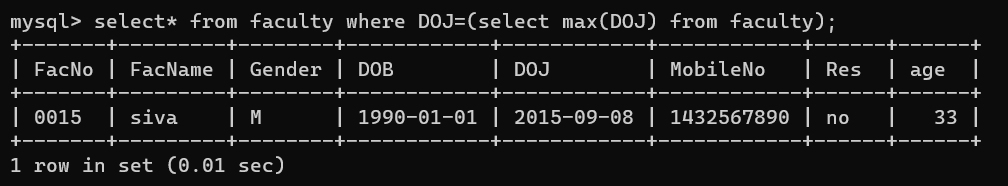
1.Which of the student’s score is greater than the average score?



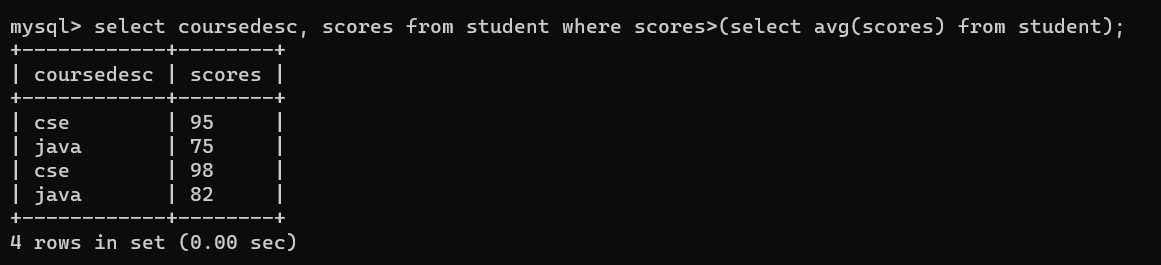
2.Which of the students’ have written more than one assessment test?



3.Which faculty has joined recently and when?



4.List the course and score of assessments that have the value more than the average score each Course

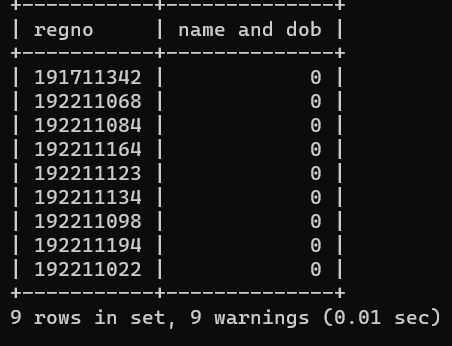


RESULT: The records from the tables are displayed using Sub-Query and Correlated Sub-Query.

Ex.No:9

**DATABASE VIEW, INDEX**

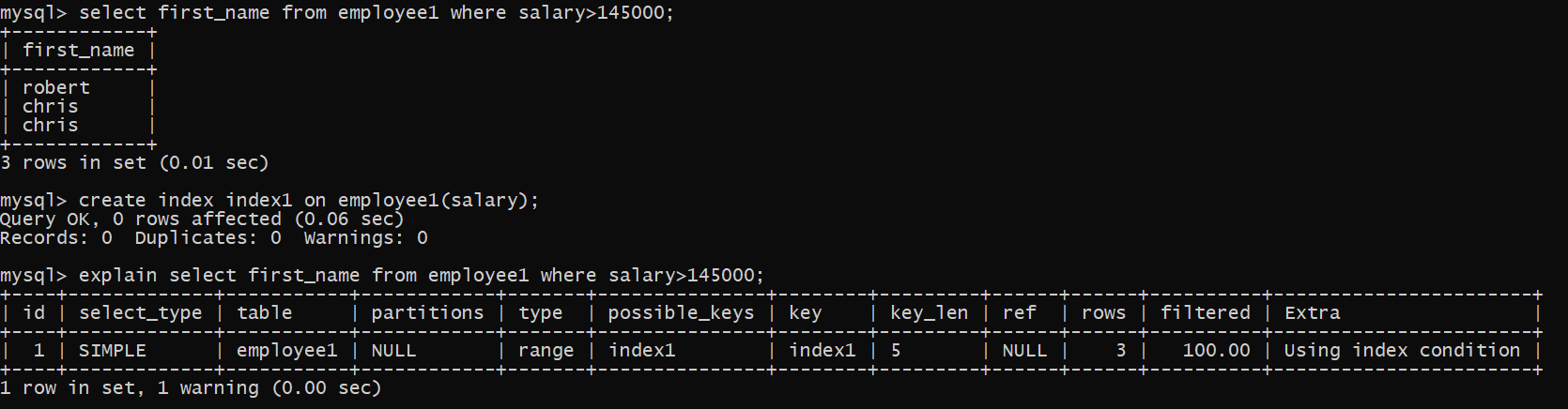
1)Create a view with name ‘std\_view’ using STUDENT table which holds the value of register number, name and DOB of student.

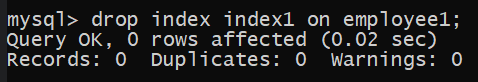


mysql> create index regno on student(regno);

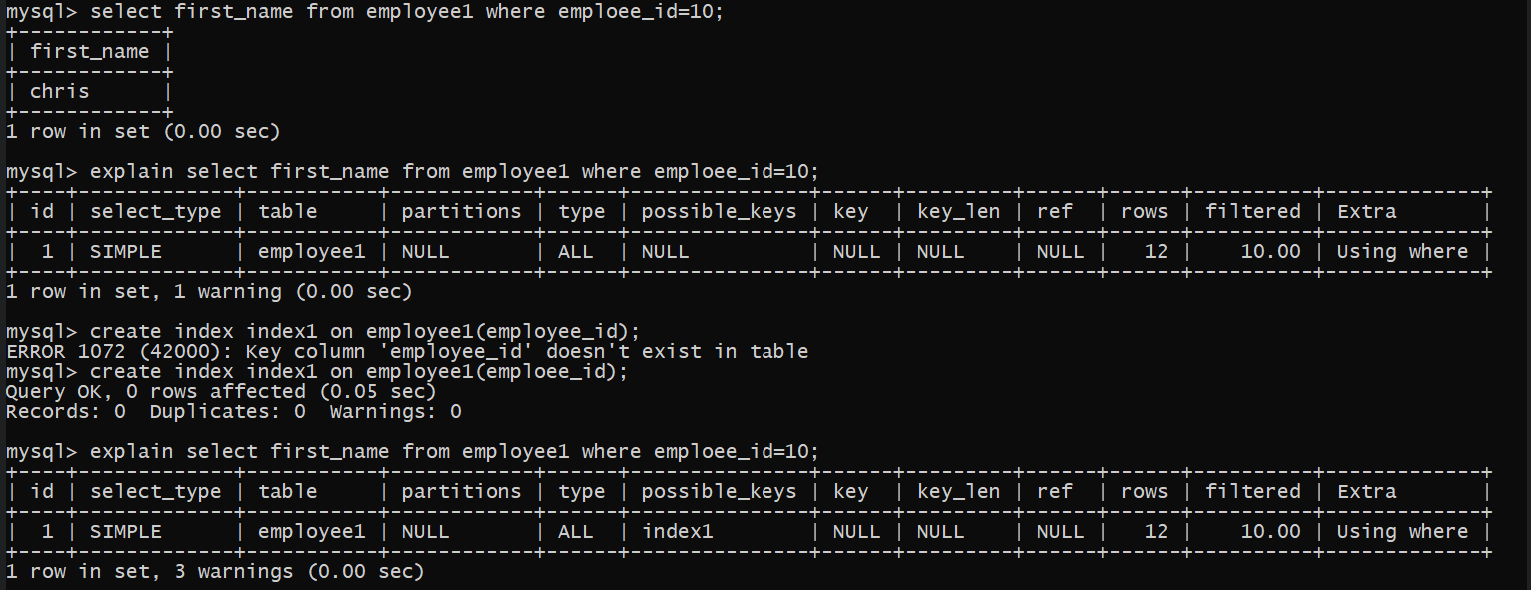
Query OK, 0 rows affected (0.15 sec)

2)Create index1 for ‘salary’ attribute from employee1 relation and list the first name of the employees whose salary is above 145000 and explain the working principle of index and drop index



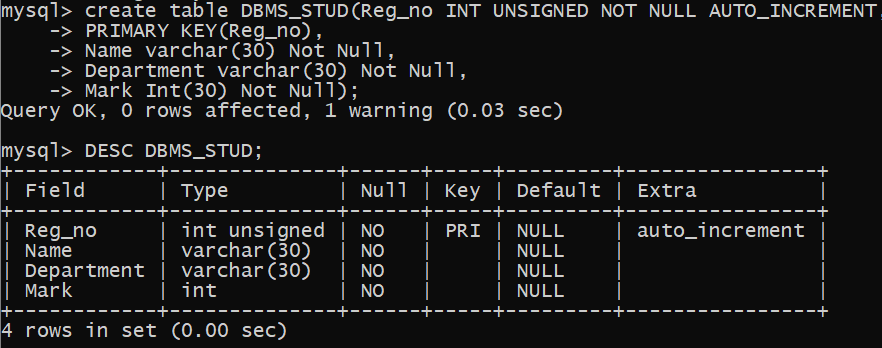


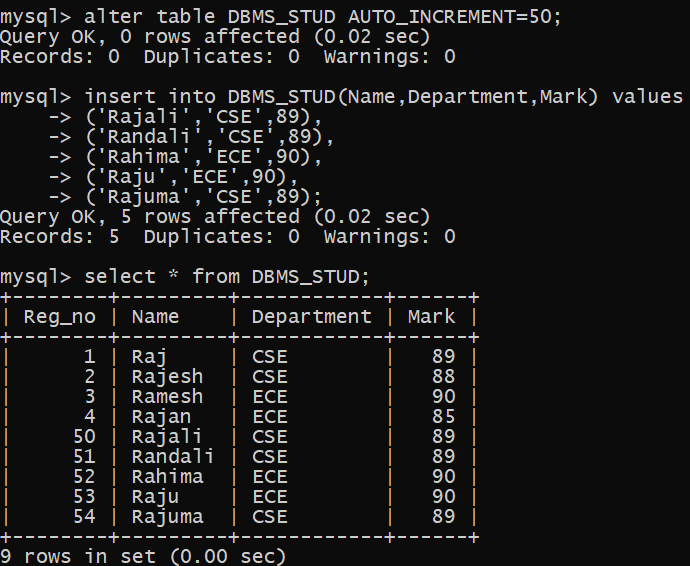
3)Create index1 for ‘employee\_id’ attribute and display the first\_name of the employee whose employee\_id is 10 and explain the working principle of index1



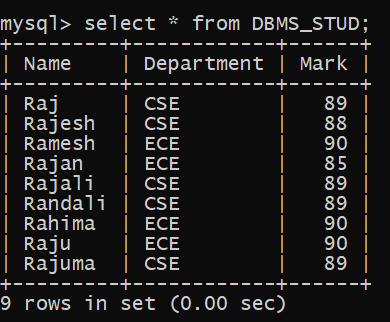
4) Populate register number using increment in DBMS\_STUD table

Manually Populate Register Number





5) Drop the auto increment



**RESULT**

The records from the tables are displayed using JOIN using EquiJoin, InnerJoin and OuterJoin.