Ex. No.: 8

Date:

**Joins – EquiJoin, InnerJoin, OuterJoin**

**AIM:**

To perform JOIN using EquiJoin, InnerJoin, OuterJoin on the given relation.

**DESCRIPTION**:

**JOIN**

A MySQL join is a method of linking data from one or more table based on values of the common column between tables.

MySQL supports the following types of joins:

1. [Cross join](http://www.mysqltutorial.org/www.mysqltutorial.org/mysql-cross-join/)
2. [Inner join](http://www.mysqltutorial.org/www.mysqltutorial.org/mysql-inner-join.aspx)
3. [Left join](http://www.mysqltutorial.org/www.mysqltutorial.org/mysql-left-join.aspx)
4. [Right join](http://www.mysqltutorial.org/www.mysqltutorial.org/mysql-right-join/)

**CROSS JOIN**

The CROSS JOIN makes a Cartesian product of rows from multiple tables. Suppose, you join t1 and t2 tables using the CROSS JOIN, the result set will include the combinations of rows from the t1 table with the rows in the t2 table.

**INNER JOIN**

To join two tables, the INNER JOIN compares each row in the first table with each row in the second table to find pairs of rows that satisfy the join-predicate. Whenever the join-predicate is satisfied by matching non-NULL values, column values for each matched pair of rows of the two tables are included in the result set.

**LEFT JOIN**

Unlike an INNER JOIN, a LEFT JOIN returns all rows in the left table including rows that satisfy join-predicate and rows do not. For the rows that do not match the join-predicate, NULLs appear in the columns of the right table in the result set.

**RIGHT JOIN**

A RIGHT JOIN is similar to the LEFT JOIN except that the treatment of tables is reversed. With a RIGHT JOIN, every row from the right table ( t2) will appear in the result set. For the rows in the right table that do not have the matching rows in the left table ( t1), NULLs appear for columns in the left table ( t1).

**SYNTAX**:

**CROSS JOIN:**

SELECT

t1.id, t2.id

FROM

t1

CROSS JOIN t2;

**INNER JOIN:**

SELECT

t1.id, t2.id

FROM

t1

INNER JOIN

t2 ON t1.pattern = t2.pattern;

**LEFT JOIN:**

SELECT

t1.id, t2.id

FROM

t1

LEFT JOIN

t2 ON t1.pattern = t2.pattern

ORDER BY t1.id;

**RIGHT JOIN**:

SELECT

t1.id, t2.id

FROM

t1

RIGHT JOIN

t2 on t1.pattern = t2.pattern

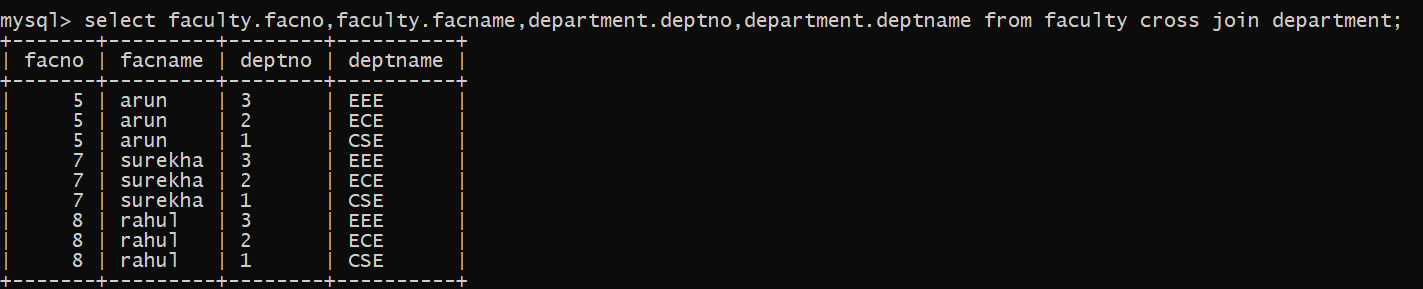
ORDER BY t2.id;

**Questions**:

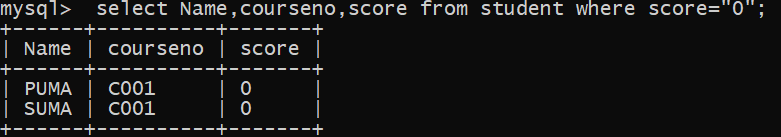
1. List the departments where the faculty members are working.
2. Find the student who has no score in any of the courses. List student name and course number.
3. The office clerk needs the names of the courses taken by the faculty belonging to ‘ECE department’ whose name is ‘Kamal’

**OUTPUTS:**

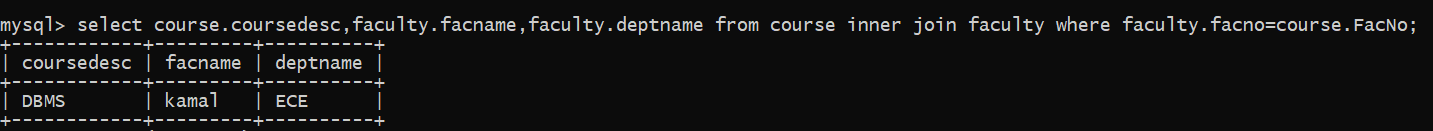
1)



2)



3)



**RESULT**: The records from the tables are displayed using JOIN using EquiJoinInnerJoin, OuterJoin.