**MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY**

SANTOSH, TANGAIL-1902



DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Lab Report**

**Experiment no :** 05

**Experiment name :** Different Types of Joining Operation.

**Course Title :** Database Management Systems Lab

**Course Code :** ICT-2108

|  |  |
| --- | --- |
| Submitted by | Submitted to |
| Name : Kuldip Saha Mugdha  ID : IT22018  2nd Year, 1st Semester  Session: 2021-2022  Dept. of ICT, MBSTU | Dr. Md. Abir Hossain  Associate Professor ,  DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY  MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY |

**Date of performance : 25/3/2024**

**Date of submission : 01/04/2024**

**Experiment No:** 05

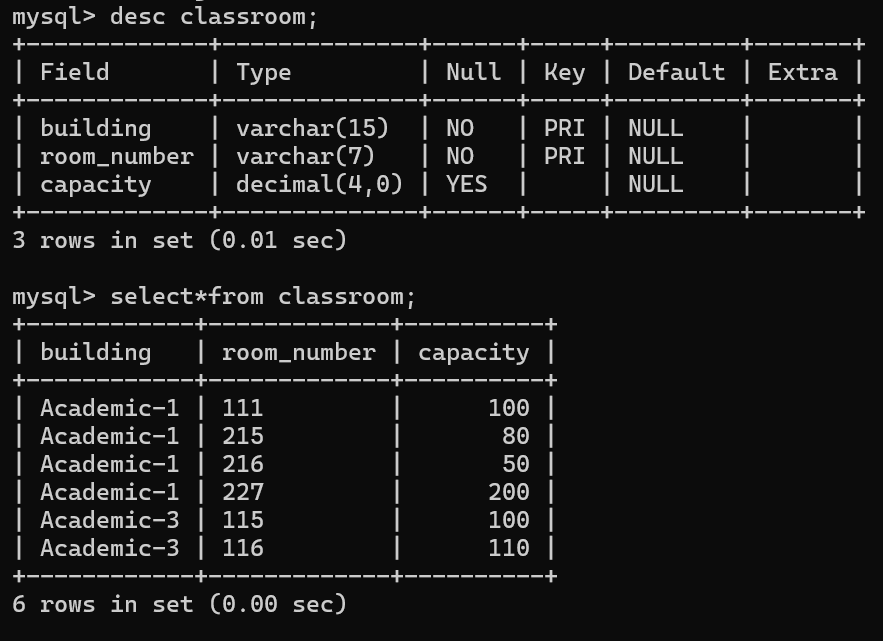
**Experiment Name:** Different Types of Joining Operation.

**Objectives:** The main objective of this experiment is to use different types of join in MySQL. We will learn how to join two tables using different types of joining. And also learn how to use joining with condition to find desire values.

**Required Instruments**:

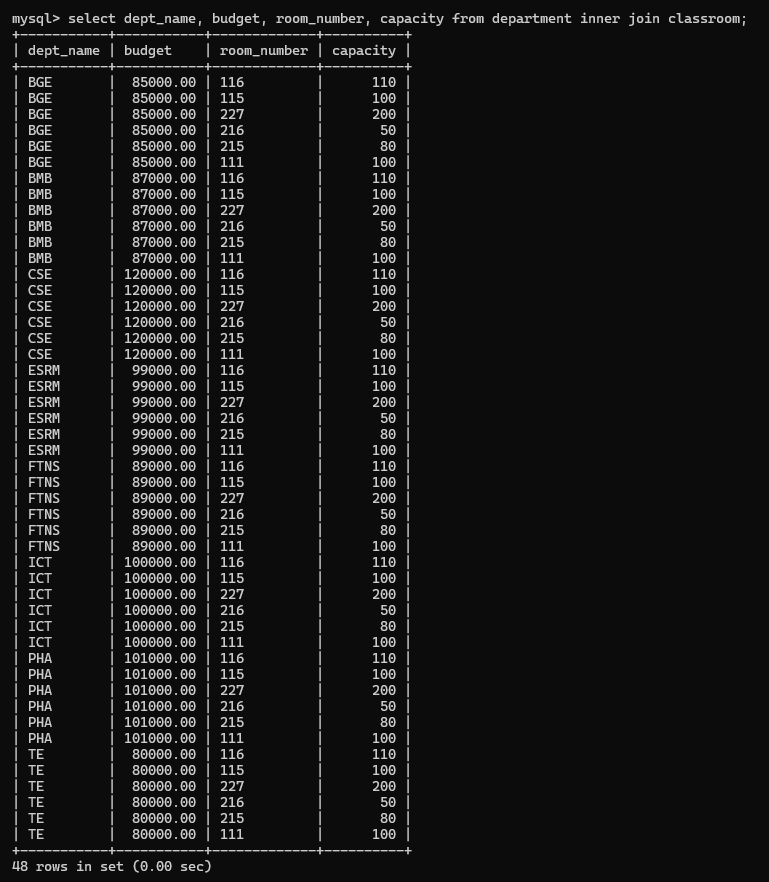
1. MySQL 8.3 Command line Client.
2. Computer (at least 4GB RAM).

For the purpose of joining, a classroom table is created and data has been inserted in that table.

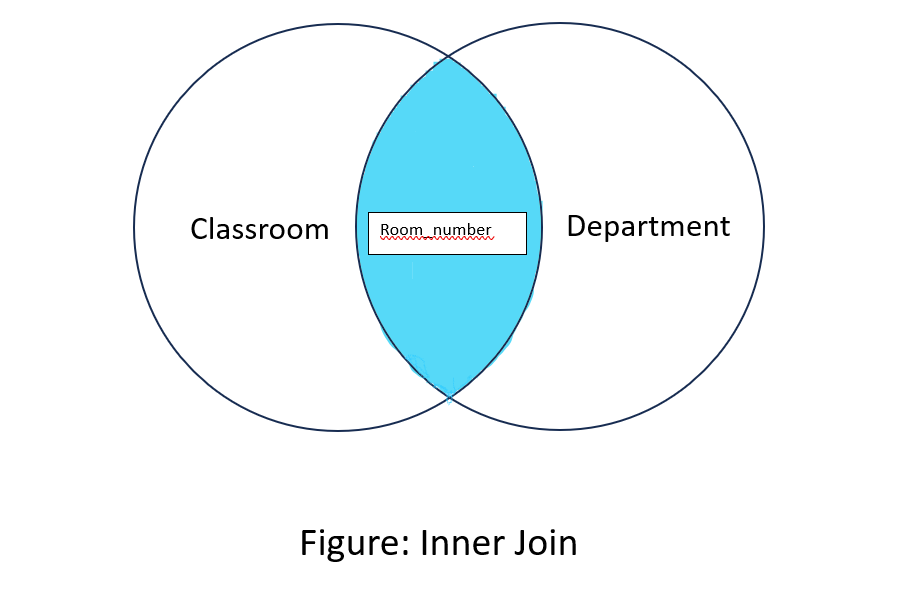
****

**Inner join:**

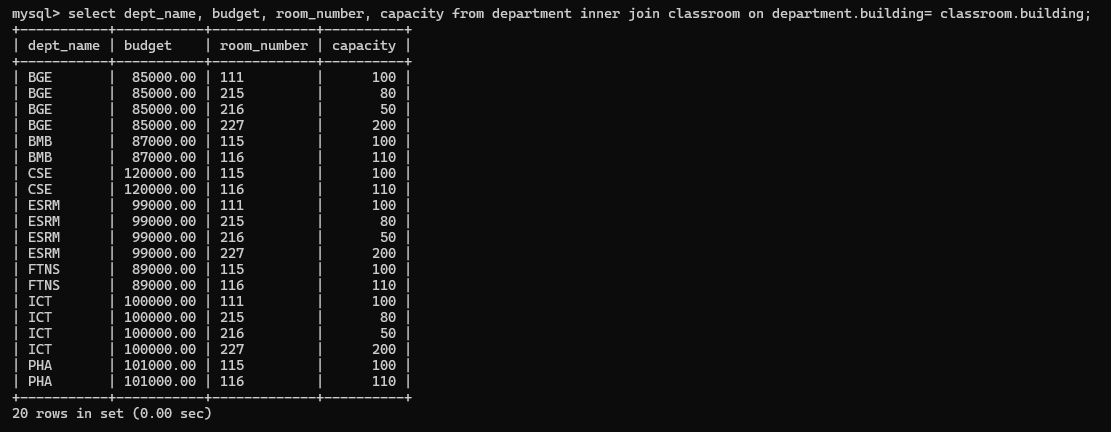
**SQL expression:** select dept\_name, budget, room\_number, capacity from department inner join classroom;



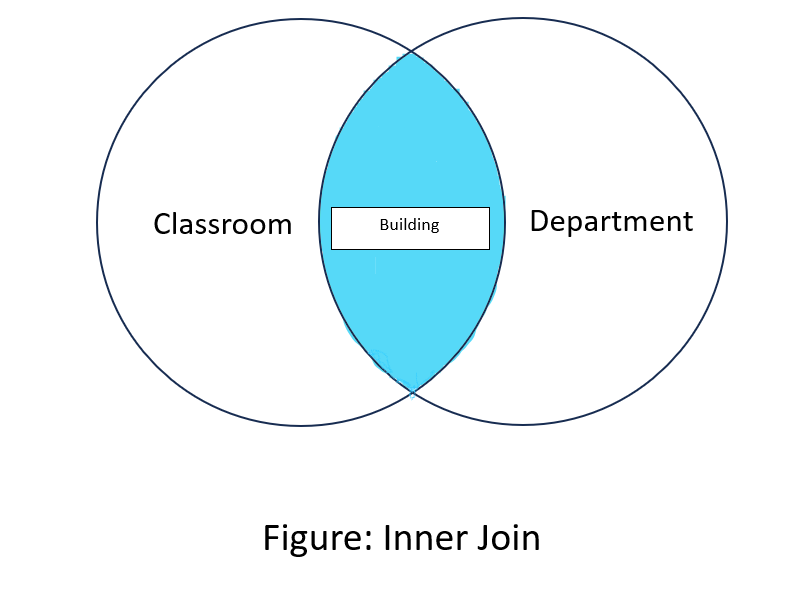
Here, this query selects the department name, budget, room\_number, and classroom capacity by joining the department and classroom tables based on the shared “room\_number” field.



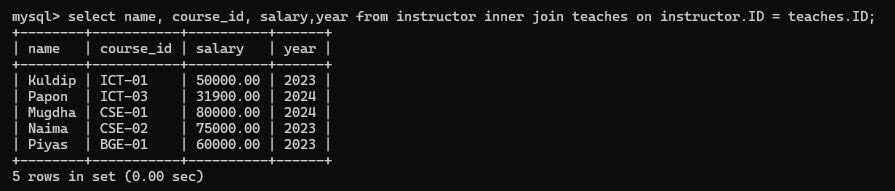
**SQL expression:** select dept\_name, budget, room\_number, capacity from department inner join classroom on department.building= classroom.building;



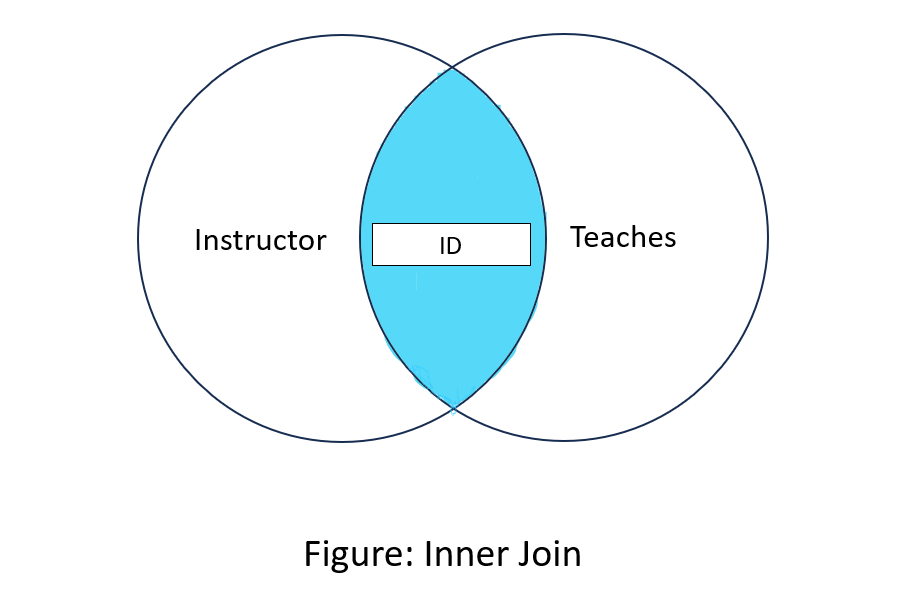
Here, this query selects the department name, budget, classroom’s room number, and classroom’s capacity by joining the two tables (department and classroom) based on shared “building” field. This will return data only where the department and classroom are located in the same building.



**SQL expression:** select name, course\_id, salary,year from instructor inner join teaches on instructor.ID = teaches.ID;

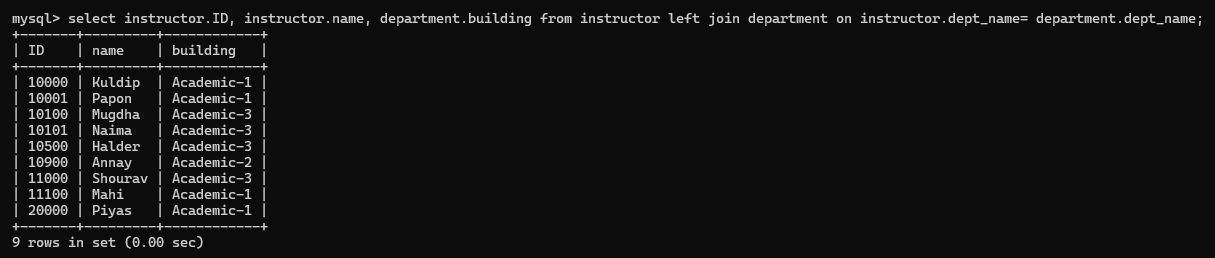


This query selects the instructor's name, course ID, salary, and the year they taught the course, joining the instructor and teaches tables using the common ID field.

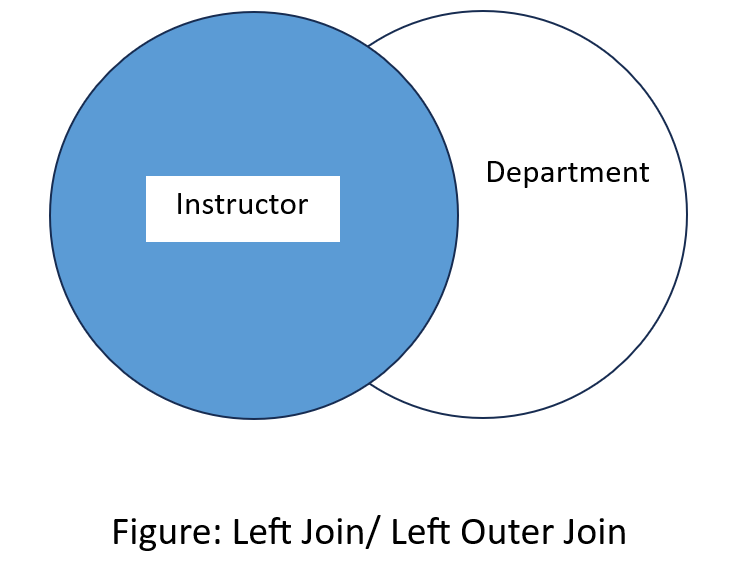


**Left join:**

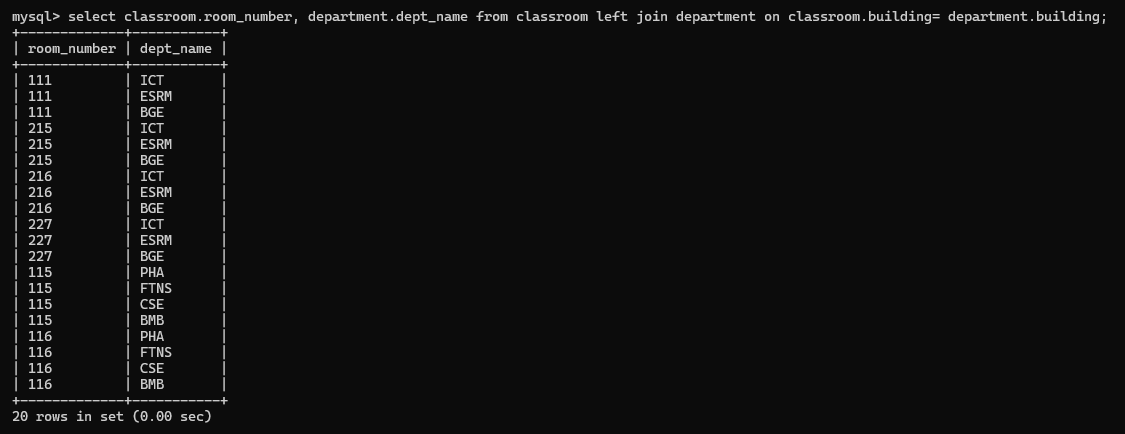
**SQL expression:** select instructor.ID, instructor.name, department.building from instructor left join department on instructor.dept\_name= department.dept\_name;



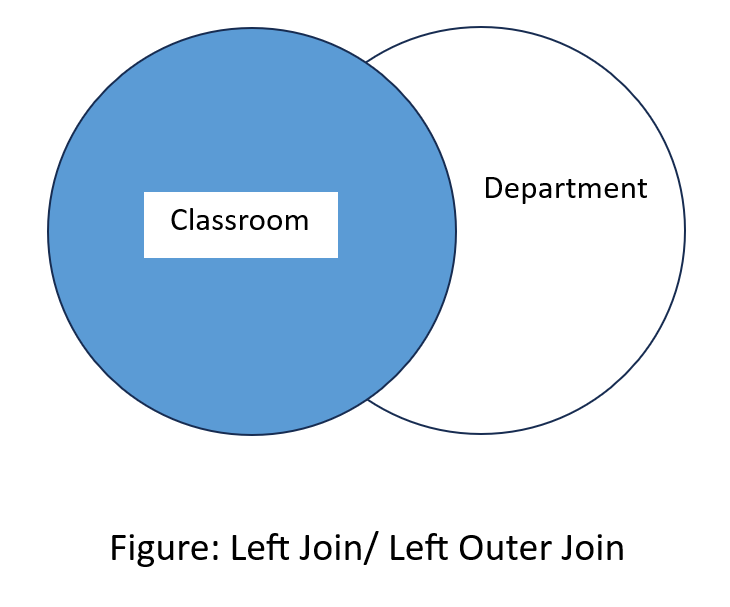
This query performs a left join between the instructor and department tables based on the “dept\_name” field. The left join ensures that all records from the instructor table are included, even if there’s no corresponding entry in the department table.



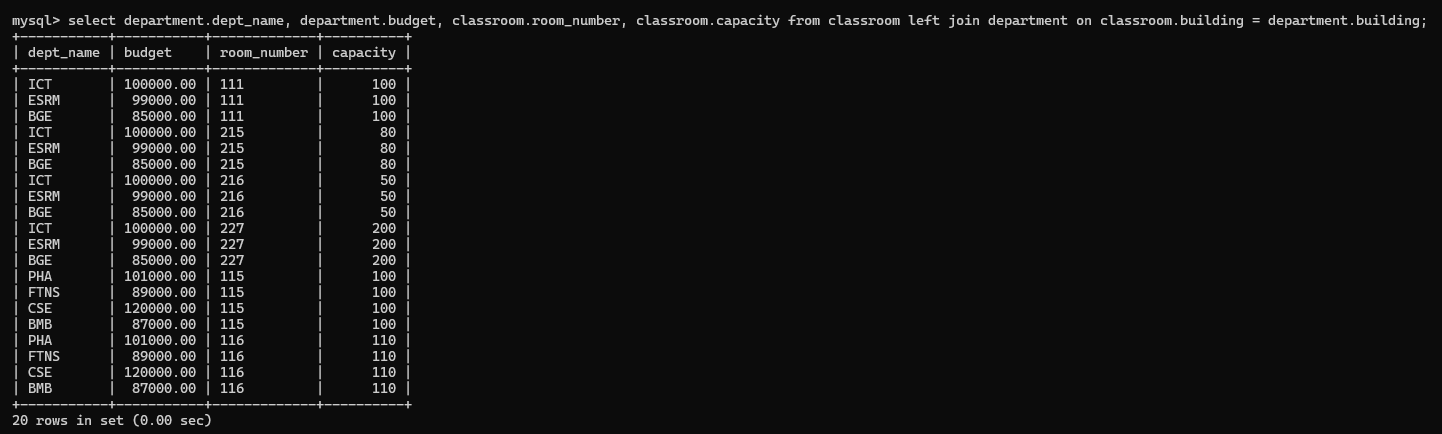
**SQL expression:** select classroom.room\_number, department.dept\_name from classroom left join department on classroom.building= department.building;



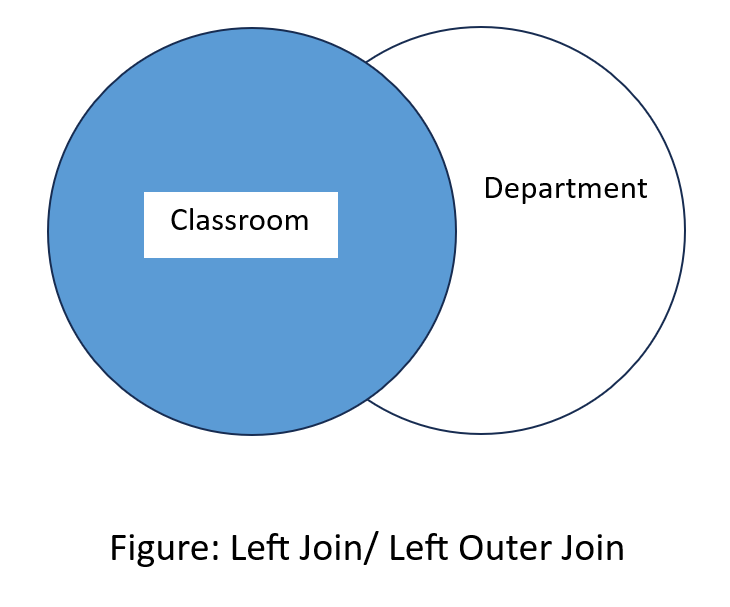
This query performs a left join between the Classroom and department tables based on the “building” field. The left join ensures that all records from the classroom table are included, even if there’s no corresponding entry in the department table.



**SQL expression:** select department.dept\_name, department.budget, classroom.room\_number, classroom.capacity from classroom left join department on classroom.building = department.building;

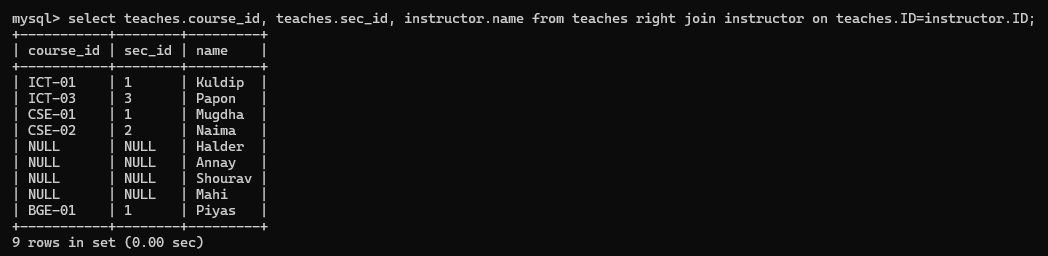


This query uses a LEFT JOIN to combine rows from the classroom and department tables based on the common building column. The LEFT JOIN ensures that all rows from the classroom table are included, even if there is no matching building in the department table, with NULL values being returned for the dept\_name and budget in such cases.

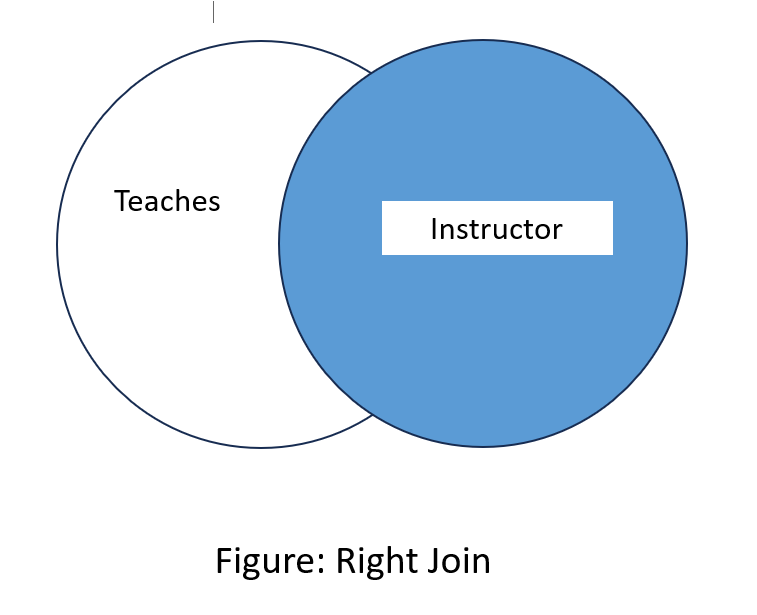


**Right join:**

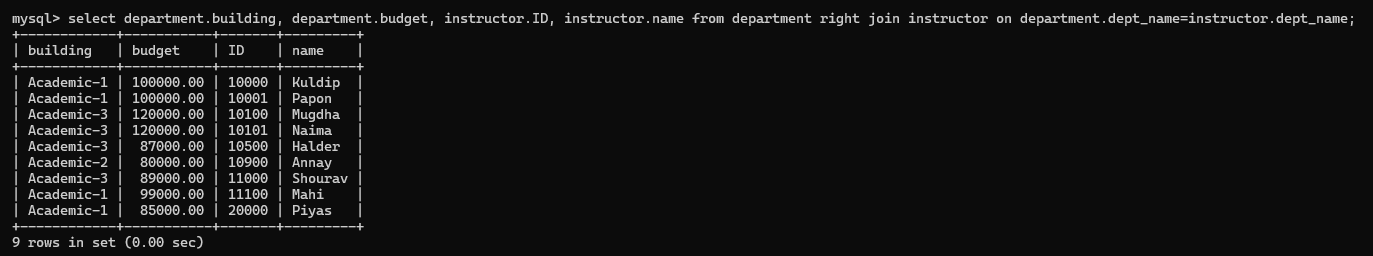
**SQL expression:** select teaches.course\_id, teaches.sec\_id, instructor.name from teaches right join instructor on teaches.ID=instructor.ID;



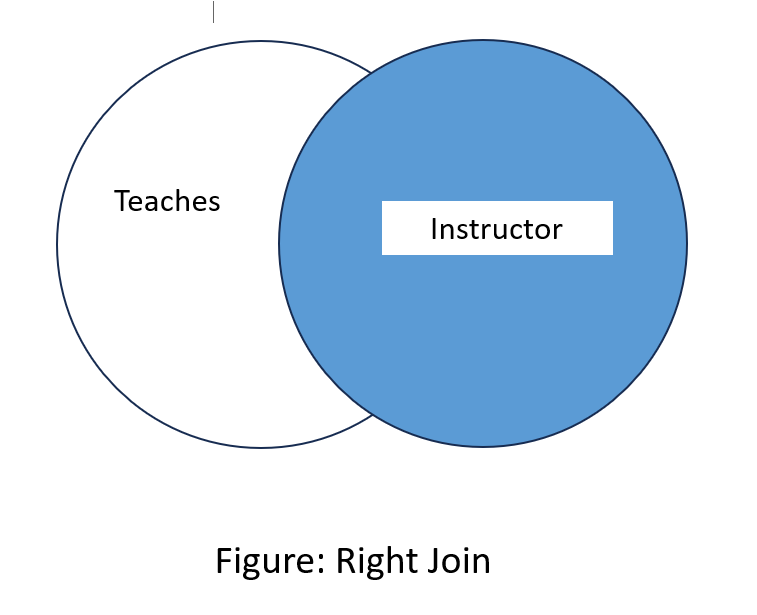
This query uses a Right Join, which ensures that all rows from the instructor table are returned, even if there is no matching ID in the teaches table. If no match is found, the course\_id and sec\_id from the teaches table will be NULL.



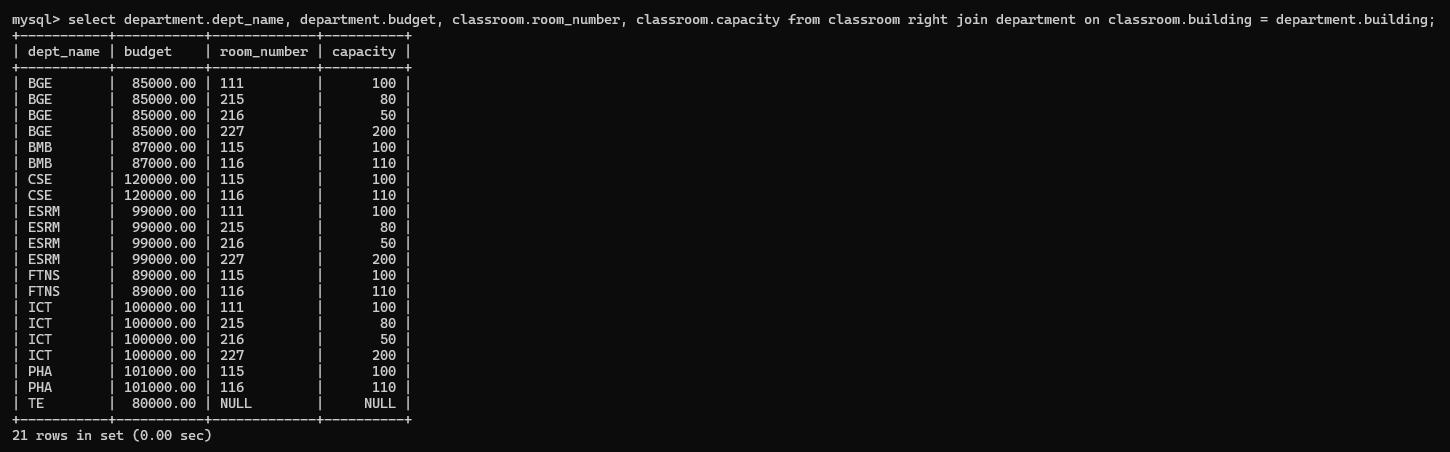
**SQL expression:** select department.building, department.budget, instructor.ID, instructor.name from department right join instructor on department.dept\_name=instructor.dept\_name;



The Right Join ensures that all rows from the instructor table are returned, even if there is no matching dept\_name in the department table. If no match is found, the building and budget from the department table will be NULL.



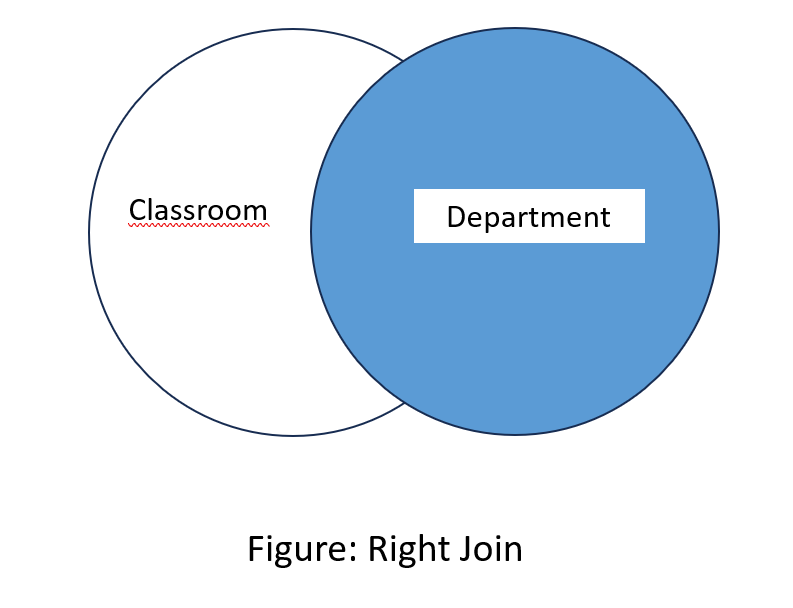
**SQL expression:** select department.dept\_name, department.budget, classroom.room\_number, classroom.capacity from classroom right join department on classroom.building = department.building;



This SQL query will retrieve the following columns:

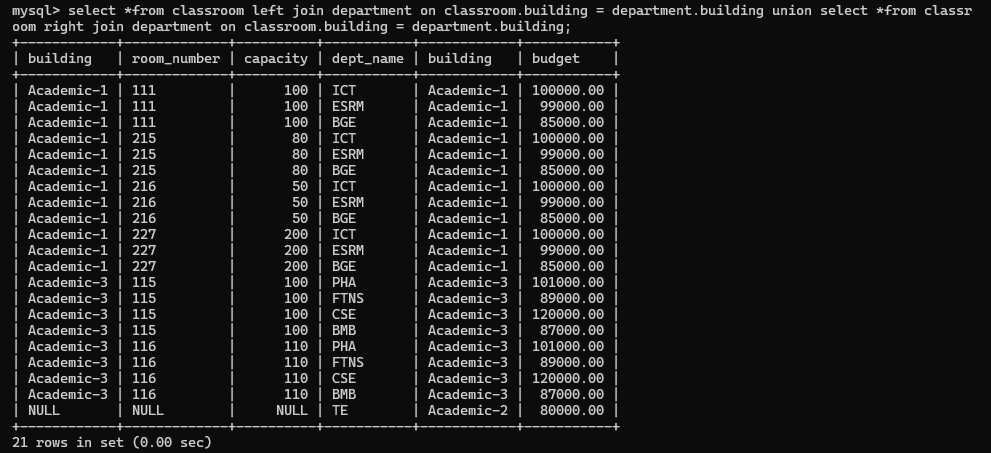
* dept\_name and budget from the department table.
* room\_number and capacity from the classroom table.

The RIGHT JOIN ensures that all rows from the department table are returned, even if there is no matching building in the classroom table. If no match is found in the classroom table, the room\_number and capacity will return NULL.

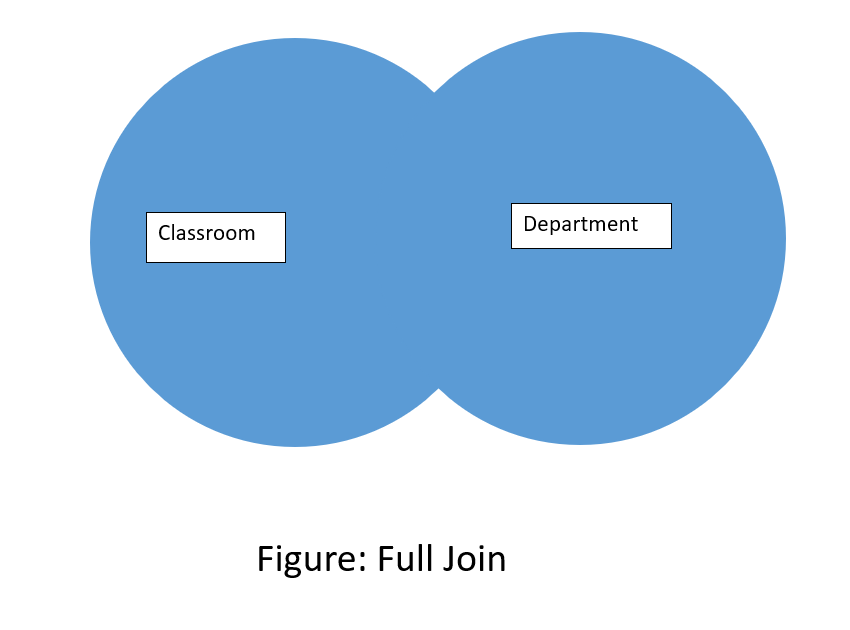


**Full join:**

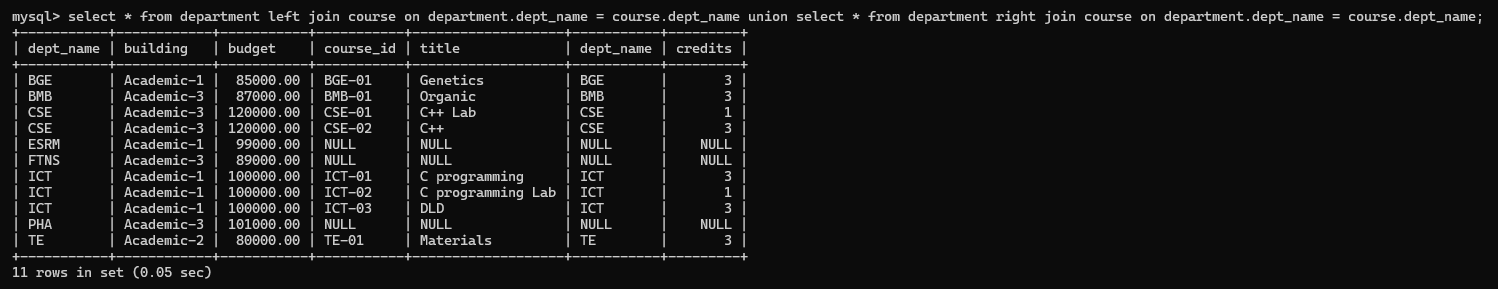
**SQL expression:** select\*from classroom left join department on classroom.building = department.building union select\*from classroom right join department on classroom.building = department.building;

****

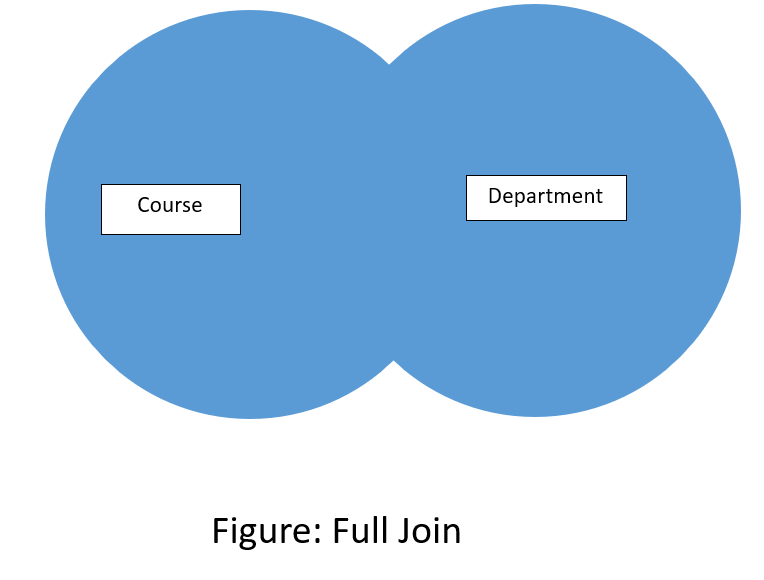
The above query is a full join that combines the results of two different joins—LEFT JOIN and RIGHT JOIN—using the UNION operator. This allows you to retrieve all unique rows from both joins meaning it will have all the data from both classroom and department.



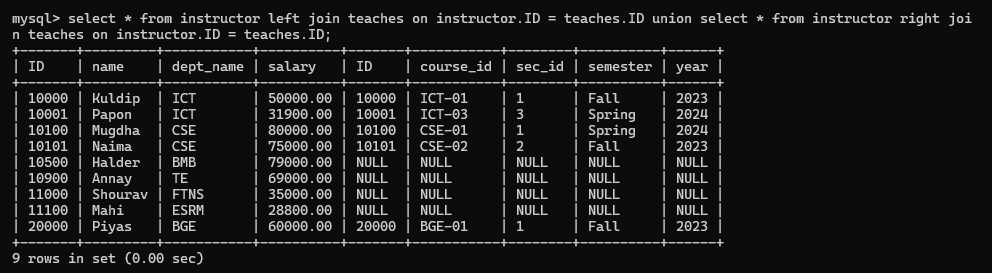
**SQL expression:** select \* from department left join course on department.dept\_name = course.dept\_name union select \* from department right join course on department.dept\_name = course.dept\_name;

****

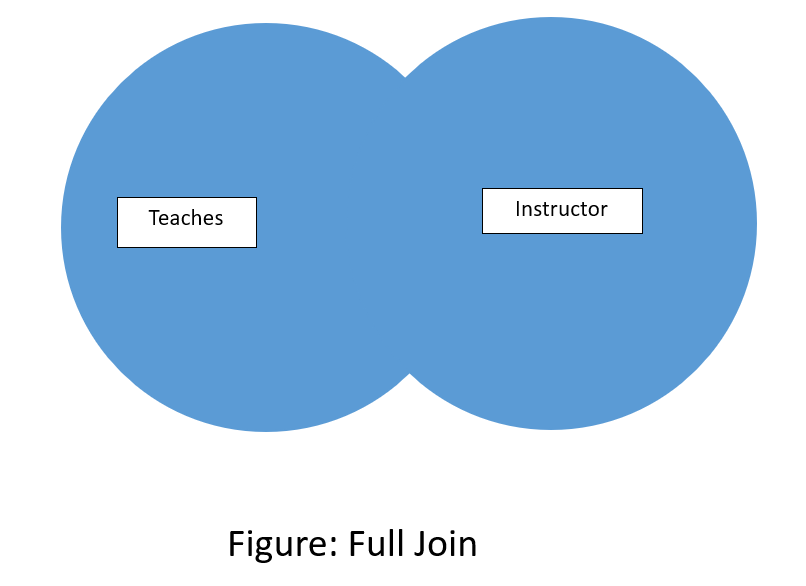
The above query is a full join that combines the results of two different joins—LEFT JOIN and RIGHT JOIN—using the UNION operator. This allows you to retrieve all unique rows from both joins meaning it will have all the data from both course and department.



**SQL expression:** select \* from instructor left join teaches on instructor.ID = teaches.ID union select \* from instructor right join teaches on instructor.ID = teaches.ID;

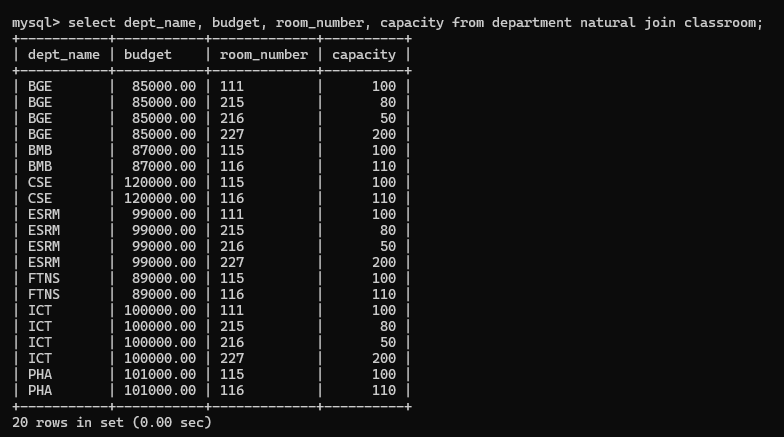


The above query is a full join that combines the results of two different joins—LEFT JOIN and RIGHT JOIN—using the UNION operator. This allows you to retrieve all unique rows from both joins meaning it will have all the data from both teaches and instructor.

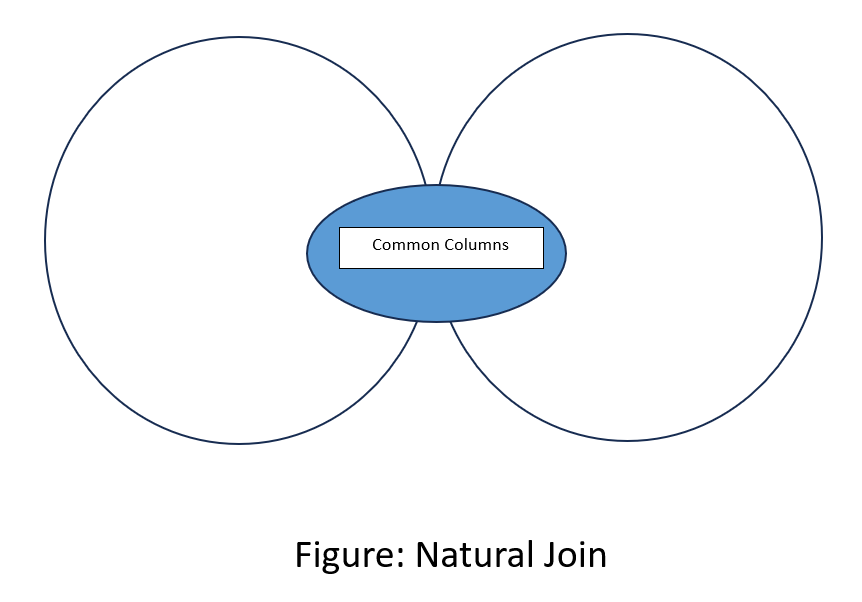


**Natural join:**

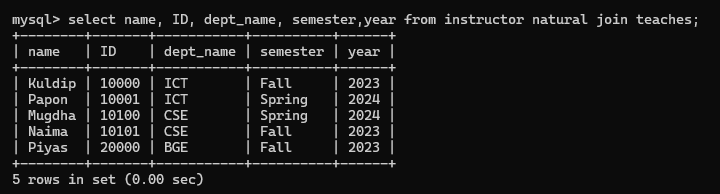
**SQL expression:** select dept\_name, budget, room\_number, capacity from department natural join classroom;

****

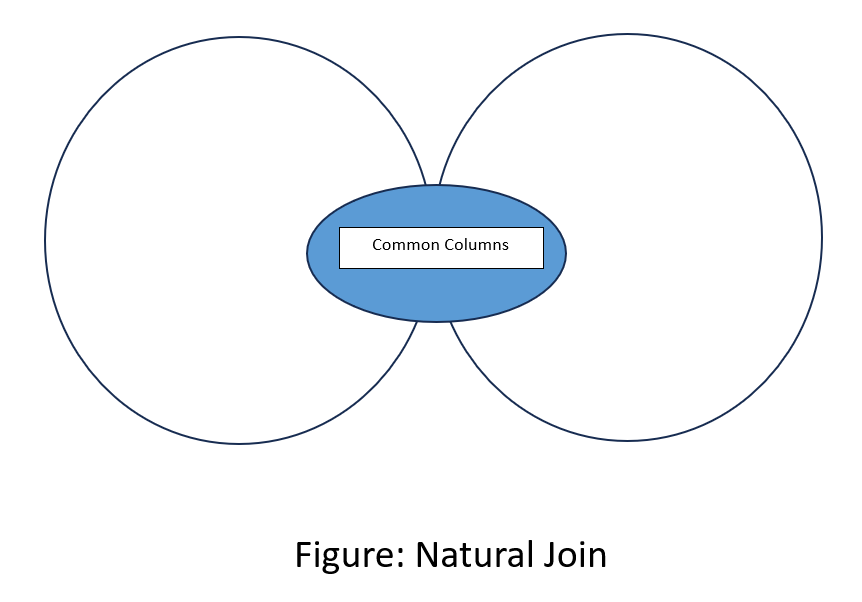
This SQL query is a natural join which will combine the department and classroom tables based on the common columns that has the same in both department and classroom.

****

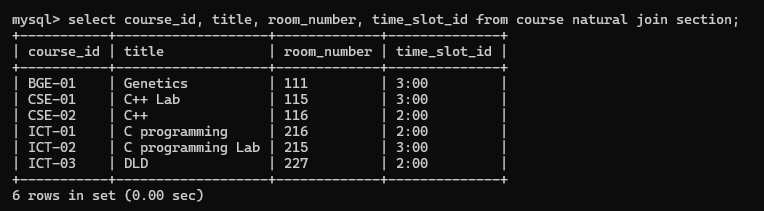
**SQL expression:** select name, ID, dept\_name, semester,year from instructor natural join teaches;



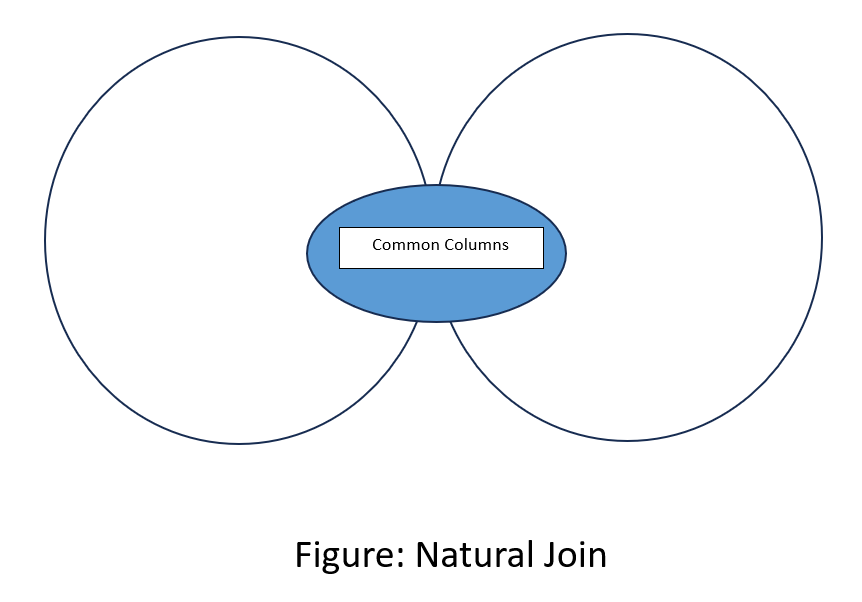
This SQL query is a natural join which will combine the department and classroom tables based on the common columns that has the same in both instructor and teaches.

****

**SQL expression:** select course\_id, title, room\_number, time\_slot\_id from course natural join section;



This SQL query is a natural join which will combine the department and classroom tables based on the common columns that has the same in both course and section.

****

**Discussion:** Using the above queries we can use various types of joining like inner join, left join, right join, natural join, full join to find desire data from databases.