



DHA SUFFA UNIVERSITY

Department of Computer Science

CS-2003 Database Systems
Spring 2023

LAB 05 Joins

OBJECTIVE(S)

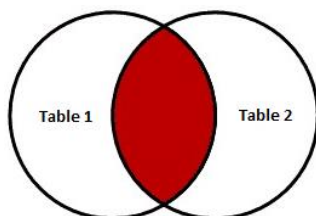
- Learn about joins.
- Learn about union.

JOINS

A JOIN clause is used to combine rows from two or more tables based on a common field between them allowing us to retrieve data from multiple tables. There are 4 different kinds of joins as described below.

INNER JOIN

This join returns records that have matching values in both tables. The records in Table 1 that do not have matching values in Table 2 will not show.



- **SELECT** col1_name, col2_name, col3_name
FROM tb1_name
(INNER) JOIN tb2_name
ON tb1_name.col_name = tb2_name.col_name;
- **SELECT** t1.col_name(s), t2.col_name(s)
FROM tb1_name as t1
(INNER) JOIN tb2_name as t2
ON t1.col_name = t2.col_name;

TASK

- Create the following tables and set appropriate foreign keys.
Student(sid, name, cid, marks, did)
Dept(did, name)
Course(cid, cname)
- Insert/update relevant data into each table.
- Display a list of all the students along with their departments.
- Write a query to get the following output:

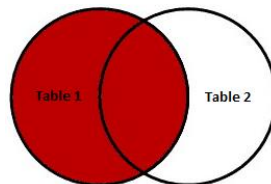
name	name	marks
isaac	db	90
william	cc	92
brad	cc	88

OUTER JOIN

This join is further divided into three categories:

LEFT JOIN

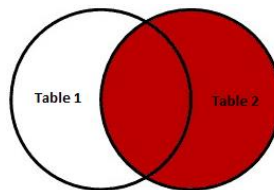
This join returns all the records from the left table and only the matched records from the right table or NULL if there is no match in the right table.



- **SELECT** col1_name, col2_name, col3_name
FROM tb1_name
LEFT JOIN tb2_name
ON tb1_name.col_name = tb2_name.col_name;
- **SELECT** t1.col_name(s), t2.col_name(s)
FROM tb1_name as t1
LEFT JOIN tb2_name as t2
ON t1.col_name = t2.col_name;

RIGHT JOIN

This join does the reverse of the aforementioned LEFT JOIN. It returns all the records from the right table and only the matched records from the left table or NULL if there is no match in the left table.



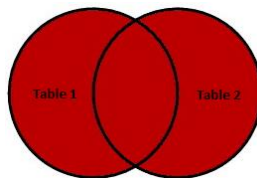
- **SELECT** col1_name, col2_name, col3_name
FROM tb1_name
RIGHT JOIN tb2_name
ON tb1_name.col_name = tb2_name.col_name;
- **SELECT** t1.col_name(s), t2.col_name(s)
FROM tb1_name as t1
RIGHT JOIN tb2_name as t2
ON tb1_name.col_name = tb2_name.col_name;

TASK

- Display a list of all the courses along with the total number of students enrolled in each course.
- Display only those courses that have more than five students enrolled in them.
- Display a list of all the students along with the total number of courses they are enrolled in.
- Display a list of only those student who are enrolled in two or more courses.

FULL JOIN

This join combines the results of both LEFT and RIGHT joins and returns all the records wherever there is a match in either the left or the right table. Some DB management software (such as MySQL) do not support FULL JOIN. In such cases, the UNION clause is used to combine the two joins.



- **SELECT** col1_name, col2_name, col3_name
 FROM tb1_name
 FULL OUTER JOIN tb2_name
 ON tb1_name.col_name = tb2_name.col_name;
- **SELECT** t1.col_name(s), t2.col_name(s)
 FROM tb1_name as t1
 FULL OUTER JOIN tb2_name as t2
 ON tb1.col_name = tb2.col_name;

UNION

This union keyword is used to combine the results of two or more select statements. By default, UNION returns only distinct records. In order to use the UNION operator, we must ensure that:

- Each **SELECT** statement in **UNION** must have the same number of columns.
- The columns in each **SELECT** statement must be in the same order.
- The columns must have similar datatypes.

The syntax for using UNION is:

- **SELECT** col1_name, col2_name, col3_name **FROM** tb1_name
 WHERE condition(s)
 UNION
 SELECT col1_name, col2_name, col3_name **FROM** tb2_name
 WHERE condition(s);

In order to allow duplicate values, we use **UNION ALL**.

- **SELECT** col1_name, col2_name, col3_name **FROM** tb1_name
 WHERE condition(s)
 UNION ALL
 SELECT col1_name, col2_name, col3_name **FROM** tb2_name
 WHERE condition(s);

As mentioned previously, we can use UNION in place of FULL OUTER JOIN where required.

- **SELECT** t1.col_name(s), t2.col_name(s)
 FROM tb1_name as t1
 LEFT JOIN tb2_name as t2
 ON t1.col_name = t2.col_name
 UNION **SELECT** t1.col_name(s), t2.col_name(s)
 FROM tb1_name as t1
 RIGHT JOIN tb2_name as t2
 ON t1.col_name = t2.col_name;

TASK

- Display a list of all the courses along with the names of all the students enrolled in them (irrespective of whether any of the values is NULL).
- Display the number of students that are not assigned to any department along with the departments that have no students assigned to them.

LAB ASSIGNMENT

Use the tables created in Lab 04 to complete this assignment.

1. Count the number of employees in each department. Also, display the maximum salary of each department.
2. Display the id, full name, designation, and salary of the lowest paid employee.
3. Display the names of all the designations along with the employees assigned to that designation (irrespective of whether the designation has been assigned to someone or not).
4. List all the countries with more than two departments. Also, give the total number of departments in that country.
5. List all the cities to which the employees belong to or where the departments exist.
6. List all the departments along with their employees' names (irrespective of whether any of the values is NULL).

SUBMISSION GUIDELINES

- Take a screenshot of each task. Ensure that all screenshots have a white background and black text. You can alter the background and text colors through the properties of the MySQL command line client.
- Place all the screenshots in a single word file labeled with Roll No and Lab No. e.g. **'cs181xxx_Lab06'**
- Convert the file into PDF.
- Submit the PDF file at [LMS](#)
- **-100%** policies for plagiarism.