SQL Joins in MySQL Workbench

1. Introduction to Joins

Definition:

A join is used in SQL to combine rows from two or more tables based on a related column between them. This allows retrieving data from multiple tables by defining relationships using common columns.

Explanation:

Joins help bring together data from multiple tables to create meaningful results. Each type of join has a specific way of combining tables, and these include **inner joins**, **outer joins**, **cross joins**, and **self joins**.

2. Types of Joins in MySQL

1. Natural Join:

A **natural join** is a type of join that automatically joins tables based on columns with the same name and compatible data types in both tables. It eliminates the need for explicitly specifying the join condition.

2. Inner Join (Equi Join)

An **inner join** returns only the rows where there is a match in both tables based on the specified condition. Rows with no match in either table are excluded.

3. Left Outer Join (Left Join)

A **left outer join** returns all rows from the left table (in this case, emp), and the matching rows from the right table (in this case, dept). If there is no match, NULL values are returned for the right table's columns.

4. Right Outer Join (Right Join)

A **right outer join** returns all rows from the right table (dept), and the matching rows from the left table (emp). If there is no match, NULL values are returned for the left table's columns.

5. Cross Join (Cartesian Join)

A **cross join** returns the Cartesian product of both tables, meaning that every row in the left table is combined with every row in the right table. There is no condition for the relationship between tables.

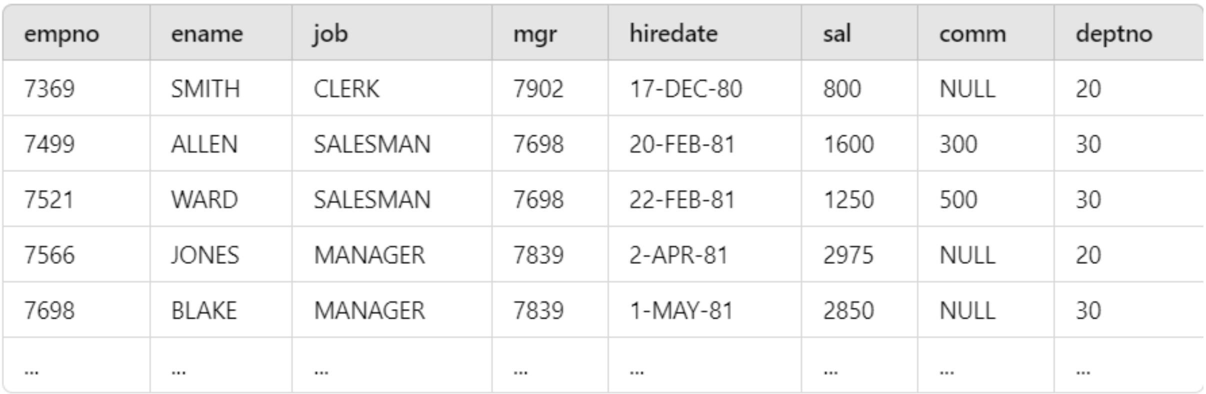
6. Self Join

A **self join** is when a table is joined with itself. It is used to find relationships among rows within the same table, such as identifying an employee’s manager.

---

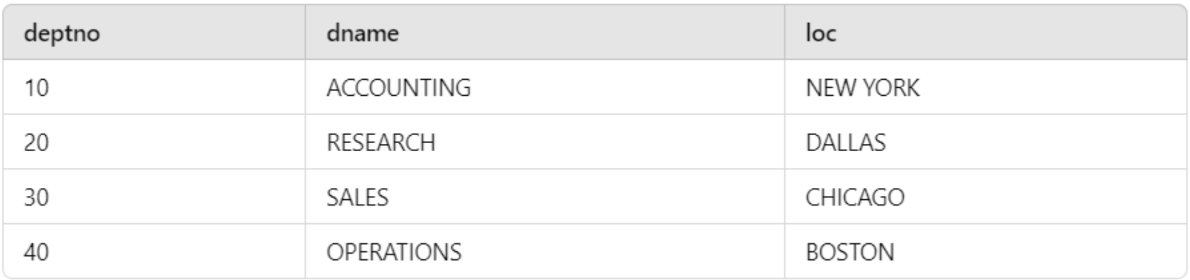
3. EMP and DEPT Table Structures

emp Table



empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	17-DEC-80	800	NULL	20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	2-APR-81	2975	NULL	20
7698	BLAKE	MANAGER	7839	1-MAY-81	2850	NULL	30
...	...	...	...	...	...	...	...

dept Table



deptno	dname	loc
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

---

4. Detailed Explanation of Joins Using EMP and DEPT Tables

4.1 Natural Join

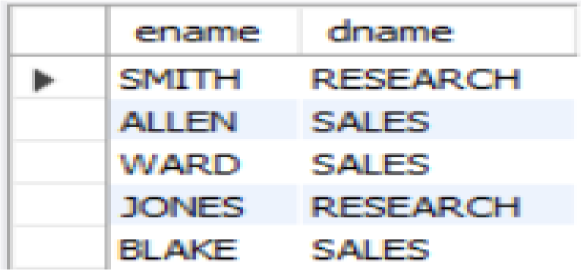
Definition:

A **natural join** automatically combines rows from two tables based on columns with the same name and compatible data types in both tables. Unlike other types of joins, there’s no need to explicitly specify the join condition, as the database assumes the join is performed on all matching column names.

Example:

```
SELECT emp.ename, dept.dname
FROM emp
NATURAL JOIN dept;
```

Result:



	ename	dname
▶	SMITH	RESEARCH
	ALLEN	SALES
	WARD	SALES
	JONES	RESEARCH
	BLAKE	SALES

Explanation:

In this query, MySQL automatically joins the emp and dept tables based on the column deptno, since it exists in both tables. There is no need to explicitly specify the join condition (ON emp.deptno = dept.deptno). However, you must ensure that the column names match exactly between the tables.

Be cautious with **natural join** as it can join on unintended columns if they have the same name, even if those columns aren’t logically related. It’s generally better to use **inner join** for more control over the join condition.

4.2 Inner Join (Equi Join)

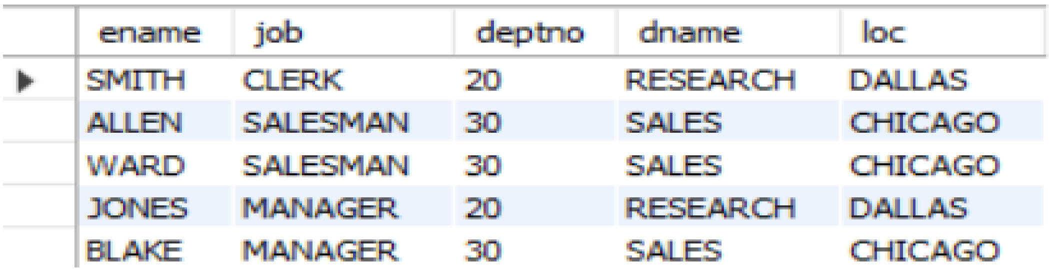
Definition:

An inner join returns rows where there is a match between the emp and dept tables based on the common column deptno. If there is no match, the rows from both tables will not be included in the result.

Example:

SELECT emp.ename, emp.job, dept.dname, dept.loc
FROM emp
INNER JOIN dept ON emp.deptno = dept.deptno;

Result:



ename	job	deptno	dname	loc
SMITH	CLERK	20	RESEARCH	DALLAS
ALLEN	SALESMAN	30	SALES	CHICAGO
WARD	SALESMAN	30	SALES	CHICAGO
JONES	MANAGER	20	RESEARCH	DALLAS
BLAKE	MANAGER	30	SALES	CHICAGO

Explanation:
Only employees with matching deptno in both the emp and dept tables are returned. Unmatched employees are excluded.

4.3 Left Outer Join (Left Join)

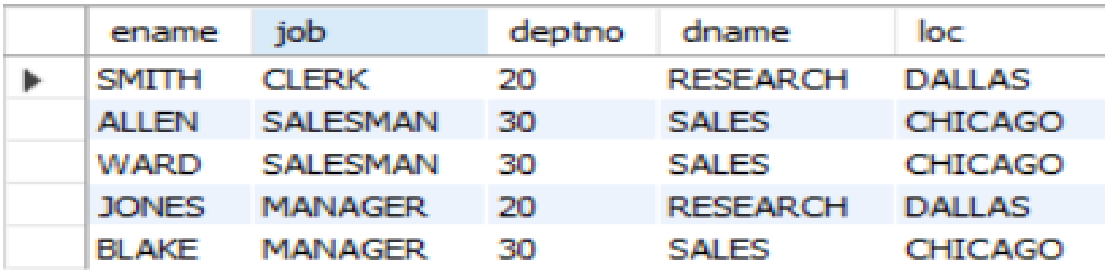
Definition:

A left outer join returns all rows from the left table (emp), and the matching rows from the right table (dept). If there is no match, NULL values are returned for the right table's columns.

Example:

SELECT emp.ename, emp.job, dept.deptno,dept.dname, dept.loc
FROM emp
LEFT JOIN dept ON emp.deptno = dept.deptno;

Result:



ename	job	deptno	dname	loc
SMITH	CLERK	20	RESEARCH	DALLAS
ALLEN	SALESMAN	30	SALES	CHICAGO
WARD	SALESMAN	30	SALES	CHICAGO
JONES	MANAGER	20	RESEARCH	DALLAS
BLAKE	MANAGER	30	SALES	CHICAGO

Explanation:
All employees are returned, even those who do not have a matching department. For unmatched employees, the dname and loc are set to NULL.

4.4 Right Outer Join (Right Join)

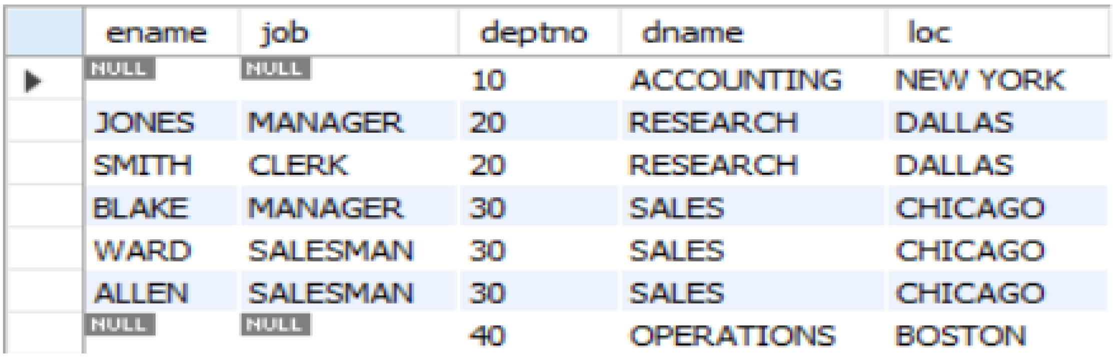
Definition:

A right outer join returns all rows from the right table (dept), and the matching rows from the left table (emp). If there is no match, NULL values are returned for the left table's columns.

Example:

```
SELECT emp.ename, emp.job, dept.deptno, dept.dname, dept.loc
FROM emp
RIGHT JOIN dept ON emp.deptno = dept.deptno;
```

Result:



	ename	job	deptno	dname	loc
▶	NULL	NULL	10	ACCOUNTING	NEW YORK
	JONES	MANAGER	20	RESEARCH	DALLAS
	SMITH	CLERK	20	RESEARCH	DALLAS
	BLAKE	MANAGER	30	SALES	CHICAGO
	WARD	SALESMAN	30	SALES	CHICAGO
	ALLEN	SALESMAN	30	SALES	CHICAGO
	NULL	NULL	40	OPERATIONS	BOSTON

Explanation:
All departments are returned, even if there are no employees assigned to them. For unmatched departments, the employee columns (ename, job) are set to NULL.

4.5 Cross Join (Cartesian Join)

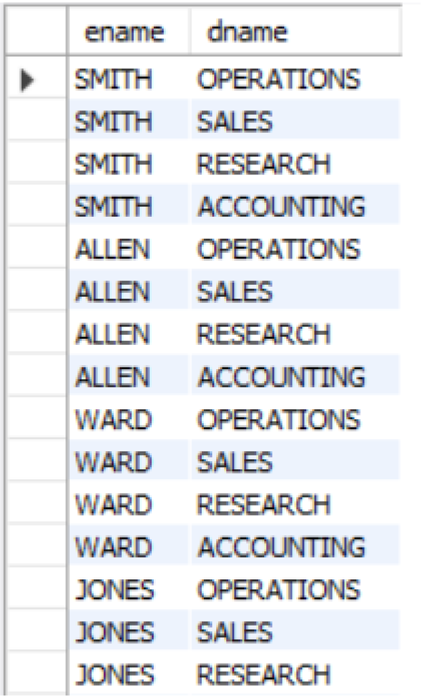
Definition:

A cross join returns the Cartesian product of both tables. Every row from the emp table is combined with every row from the dept table, regardless of matching deptno.

Example:

```
SELECT emp.ename, dept.dname
FROM emp
CROSS JOIN dept;
```

Result:



ename	dname
SMITH	OPERATIONS
SMITH	SALES
SMITH	RESEARCH
SMITH	ACCOUNTING
ALLEN	OPERATIONS
ALLEN	SALES
ALLEN	RESEARCH
ALLEN	ACCOUNTING
WARD	OPERATIONS
WARD	SALES
WARD	RESEARCH
WARD	ACCOUNTING
JONES	OPERATIONS
JONES	SALES
JONES	RESEARCH

...

and so on...

Explanation:
Every employee is paired with every department, resulting in a large number of combinations. If there are 5 employees and 4 departments, the result will have 20 rows (5 x 4 = 20).

4.6 Self Join

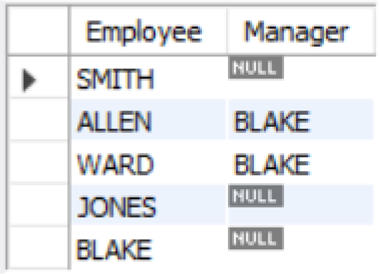
Definition:

A self join is when a table is joined with itself. It is useful for comparing rows within the same table, such as finding an employee’s manager.

Example:

```
SELECT e1.ename AS Employee, e2.ename AS Manager
FROM emp e1
LEFT JOIN emp e2 ON e1.mgr = e2.empno;
```

Result:



Employee	Manager
SMITH	NULL
ALLEN	BLAKE
WARD	BLAKE
JONES	NULL
BLAKE	NULL

Explanation:
This query finds the manager for each employee by joining the `emp` table with itself. If an employee has no manager, the result will be NULL.