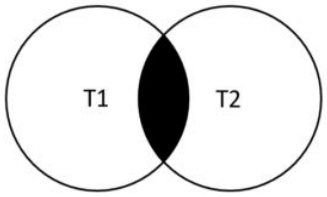
SQL

1. Inner Join, Outer Join, Cross Join

INNER JOIN :



Consider tables t1 and t2 then inner join of 2 tables selects or fetches records that have a matching value in both the tables.

Syntax:

Select t1.colName, t2.colName ..... From t1 **INNER JOIN** t2
on t1.colCommon = t2.colCommon;

Example:

Employee: {Emp_id, Emp_name, Dept_id} (Table1)
Department: {Dept_id,Dept_name} (Table2)

Emp_id	Emp_Name	Dept_id
2	ABC	1
1	XYZ	2
3	PQR	1

Dept_id	Dept_Name
1	Admin
2	IT

Select Emp_id, Dept_Name from Employee **INNER JOIN** Department on Employee.Dept_id = Department.Dept_id;

Emp_id	Dept_Name
2	Admin
1	IT
3	Admin

OUTER JOIN : (Left Outer Join & Right Outer Join)

Consider tables t1 and t2 then outer join of 2 tables selects or fetches all records that have value either matched or not from both the tables.

Syntax:

Select t1.colName, t2.colName ..... From t1 **OUTER JOIN** t2
on t1.colCommon = t2.colCommon;

Example:

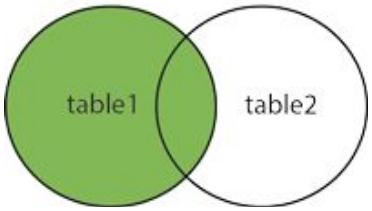
Employee: {Emp_id, Emp_name, Dept_id} (Table1)
Department: {Dept_id,Dept_name} (Table2)

Emp_id	**Emp_Name**	**Dept_id**
21	ABC	1
19	XYZ	2
34	PQR	3

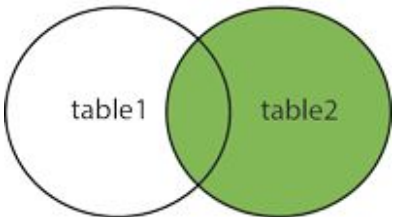
Dept_id	**Dept_Name**
1	Admin
2	IT
4	Transport

Select Emp_Name, Dept_Name from Employee **OUTER JOIN** Department on Employee.Dept_id = Department.Dept_id;

Emp_Name	**Dept_Name**
ABC	Admin
XYZ	IT
PQR	**NULL**
NULL	Transport



Left Outer Join



Right Outer Join

Select Emp_id, Dept_id, Dept_Name from Employee **LEFT JOIN** Department on Employee.Dept_id = Department.Dept_id;

Emp_id	**Dept_id**	**Dept_Name**
21	1	Admin
19	2	IT
34	3	**NULL**

Select Emp_id, Dept_id, Dept_Name from Employee **RIGHT JOIN** Department on Employee.Dept_id = Department.Dept_id;

Emp_id	**Dept_id**	**Dept_Name**
21	1	Admin
19	2	IT
NULL	4	Transport

CROSS JOIN :

Consider tables t1 and t2 then cross join of 2 tables produces a set of records which is the number of rows in the first table multiplied by the number of rows in the second table.

Syntax:

Select t1.colName, t2.colName ..... From t1 **CROSS JOIN** t2;

Example:

Student: {Student_id, Student_name} (Table1)
Course: {Course_id, Course_name} (Table2)

Student_id	**Student_name**
27	ABC
92	XYZ

Course_id	**Course_name**
A3066	Data Structures
A1054	Web Technologies

Select Student_name, Course_name from Student **CROSS JOIN** Course;

Student_name	Course_name
ABC	Data Structures
ABC	Web Technologies
XYZ	Data Structures
XYZ	Web Technologies

2. Rank and Dense Rank functions

Rank() :

Rank() is a function which assigns unique value for each row in the obtained result set.

Syntax:

Select rank() over ([partition by partition_expression, ...order by sort_expression [asc | desc], ...) as ColName, ColTable..... from tableName;

Example: Student: {id, course, semester}

id	course	semester
92	OS	1
92	DAA	2
87	DAA	2
54	WT	4

EX1 -

Select rank() over (order by id desc) as rank, id, course from Student;

rank	id	course
1	92	OS
1	92	DAA
3	87	DAA
4	54	WT

EX2 -

Select rank() over (partition by course order by semester) as rank, id, course, semester from Student;

rank	id	course	semester
1	92	DAA	2
1	87	DAA	2
1	92	OS	1
1	54	WT	4

DENSE _ RANK() :

Dense_rank() is similar to rank() function but it doesn't skip any rank if there is similar data of previous records.

Syntax:

Select dense_rank() over ([partition by partition_expression, ...order by sort_expression [asc | desc], ...) as ColName, ColTable..... from tableName;

Consider the same example of rank() function

Select rank() over (order by id desc) as rank, id, course from Student;

rank	id	course
1	92	OS
1	92	DAA
2	87	DAA
3	54	WT

3. Write a query to get employees and their manager

Given Employee (Table) - {Emp_ID, Emp_Name, Mgr_ID}

Emp_ID	Emp_Name	Mgr_ID
1	John	4
2	Smith	3
3	Kumar	4
4	Satya	5
5	Rahul	1

Query:

Select e.Emp_Name as Employee, m.Emp_Name as Manager from Employee e **JOIN** Employee m on e.Mgr_ID = m.Emp_ID;

Employee	Manager
John	Satya
Smith	Kumar
Kumar	Satya
Satya	Rahul
Rahul	John