JOINS: (29-10-2024)
======

- In RDBMS data can be stored in multiple tables.From those multiple tables if we want to retrieve the required data / information then we use a technique is known as "JOINS".
- Joins are used to retrieve the data / information from multiple tables at a time.
- Sqlserver supports the following types of joins are,

i) Equi / Inner join

ii) Outer joins

> left outer join
> right outer join
> full outer join

iii) Non equi join

iv) Cross join / Cartisean join

v) Self join

i) Equi / Inner join:
===============

- when we retrieve the required data / information multiple tables based on an " = " operator is known as equi join / inner join.
- when we use equi join we should maintain a common column(optional) in both tables but
datatypes of columns must be match.
- relationship between table is just optional for joining tables in database.
- equi join is always retrieving matching rows from the tables.

syntax:
======

```
select * from <table name1> <join key> <table name2> on <join condition>;
```

syntax for join condition in the above query:
=====================================

```
<table name1>.<common column name> = <table name2>.<common column name>
                        (or)
<table alias name1>.<common column name>=<table alias nam2>.<common column name>;
```

30-10-2024:
==========

DEMO_TABLES:
=============

EX:

```
CREATE TABLE COURSE(CID INT,CNAME VARCHAR(20),CFEE DECIMAL(6,2))
INSERT INTO COURSE VALUES(1,'.NET',5500),(2,'SQLSERVER',2500),(3,'PYTHON',5200)
```

```
CREATE TABLE STUDENT(STID INT,SNAME VARCHAR(10),CID INT)
INSERT INTO STUDENT
VALUES(1021,'SMITH',1),(1022,'ALLEN',1),(1023,'JONES',2),(1024,'ADAMS',NULL)
```

EX:

waq to fetch student and their corresponding course details from multiple tables?

```
SELECT * FROM STUDENT JOIN COURSE ON STUDENT.CID=COURSE.CID;
                              (OR)
SELECT * FROM STUDENT S JOIN COURSE C ON S.CID=C.CID;
                              (OR)
SELECT * FROM STUDENT S INNER JOIN COURSE C ON S.CID=C.CID
```

Rule for JOINS:
=============

- a row in the first table is comparing with all rows of the second table.

EX:

waq to fetch student,course details from multiple table who are joined in SQLSERVER course?

```
SELECT STID,SNAME,CNAME,CFEE FROM STUDENT S INNER JOIN COURSE C
ON S.CID=C.CID WHERE CNAME='SQLSERVER';
                    (OR)
SELECT STID,SNAME,CNAME,CFEE FROM STUDENT S INNER JOIN COURSE C
ON S.CID=C.CID AND CNAME='SQLSERVER';
```

EX:

waq to display employees from EMP,DEPT tables who are working in the location is "CHICAGO"?

```
SELECT ENAME,LOC FROM EMP E INNER JOIN DEPT D
ON E.DEPTNO=D.DEPTNO AND LOC='CHICAGO';
```

EX:

waq to display sum of salaries of department name wise by using equi join?

```
SELECT  DNAME,SUM(SAL) FROM EMP E INNER JOIN DEPT D ON E.DEPTNO=D.DEPTNO
GROUP BY DNAME;
```

EX:

waq to display deptno,sum of salaries of department name wise by using equi join?

```
SELECT  D.DEPTNO,DNAME,SUM(SAL) FROM EMP E INNER JOIN DEPT D ON
E.DEPTNO=D.DEPTNO
GROUP BY D.DEPTNO,DNAME;
```

EX:

waq to display sum of salaries of department names by using equi join if sum of salary of the department is less than 10000?

```
SELECT DNAME,SUM(SAL) FROM EMP E INNER JOIN DEPT D
ON E.DEPTNO=D.DEPTNO GROUP BY DNAME HAVING SUM(SAL)<10000;
```

02-11-2024:
==========

OUTER JOINS:
============

- In the above equi join we are retrieving matching rows(data) from tables only.if we want to retrieve matching and also unmatching rows from the multiple tables then we use a technique is called as "OUTER JOINS".
- Outer joins are again three types those are,

i) LEFT OUTER JOIN:
==================

- retrieving matching rows from both tables and unmatching rows from the left side table only.

EX:

```
SELECT * FROM STUDENT S LEFT OUTER JOIN COURSE C ON S.CID=C.CID;
SELECT * FROM COURSE C LEFT OUTER JOIN STUDENT S ON C.CID=S.CID;
```

ii) RIGHT OUTER JOIN:
==================

- retrieving matching rows from both tables and unmatching rows from the right side table only.

EX:

```
SELECT * FROM STUDENT S RIGHT OUTER JOIN COURSE C ON S.CID=C.CID;
SELECT * FROM COURSE C RIGHT OUTER JOIN STUDENT S ON C.CID=S.CID;
```

iii) FULL OUTER JOIN:
===================

- retrieving matching rows and unmatching rows from both tables at a time.

EX:

```
SELECT * FROM STUDENT S FULL OUTER JOIN COURSE C ON S.CID=C.CID;
```

NON-EQUI JOIN:
==============

- retrieving data from multiple table based on any condition except an " = " operator condition.

- in this join we will use < , > , <= , >= , != (or) < >, BETWEEN,AND,OR,......etc.

DEMO_TABLES:
==============

```
CREATE TABLE TEST11(SNO INT,ENAME VARCHAR(20));
INSERT INTO TEST11 VALUES(1,'SMITH'),(2,'ALLEN');

CREATE TABLE TEST12(SNO INT,SALARY MONEY);
INSERT INTO TEST12 VALUES(1,25000),(3,45000);
```

EX:

```
SELECT * FROM TEST11 T1 JOIN TEST12 T2 ON T1.SNO>T2.SNO;
SELECT * FROM TEST11 T1 JOIN TEST12 T2 ON T1.SNO>=T2.SNO;
SELECT * FROM TEST11 T1 JOIN TEST12 T2 ON T1.SNO<T2.SNO;
SELECT * FROM TEST11 T1 JOIN TEST12 T2 ON T1.SNO<=T2.SNO;
SELECT * FROM TEST11 T1 JOIN TEST12 T2 ON T1.SNO!=T2.SNO;
SELECT * FROM TEST11 T1 JOIN TEST12 T2 ON T1.SNO<>T2.SNO;
```

EX:

waq to display employees whose salary is between low salary and high salary from EMP,SALGRADE tables?

```
SELECT ENAME,SAL,LOSAL,HISAL FROM EMP  JOIN SALGRADE
ON SAL BETWEEN LOSAL AND HISAL;
		(OR)
SELECT ENAME,SAL,LOSAL,HISAL FROM EMP JOIN SALGRADE
ON (SAL>=LOSAL) AND (SAL<=HISAL);
```

04-11-2024:
==========

CROSS JOIN:
===========

- joining two or more than two tables without any condition.
- in cross join mechanism each row in a table is comparing with each row of another table.

for example a table is having(m) no.of rows and another table is having "n" no.of rows then

the result is (mxn) rows.

Ex:

```
SELECT * FROM STUDENT CROSS JOIN COURSE;
```

DEMO_TABLES:
==============

EX:

```
CREATE TABLE ITEMS1(SNO INT,INAME VARCHAR(10),PRICE MONEY)
INSERT INTO ITEMS1 VALUES(1,'PIZZA',180),(2,'BURGER',85)

CREATE TABLE ITEMS2(SNO INT,INAME VARCHAR(10),PRICE MONEY)
INSERT INTO ITEMS2 VALUES(101,'PEPSI',25),(102,'COCACOLA',20)
```

EX:

```
SELECT I1.INAME,I1.PRICE,I2.INAME,I2.PRICE,I1.PRICE+I2.PRICE AS TOTAL_AMOUNT
FROM ITEMS1 I1 CROSS JOIN ITEMS2 I2;
```

SELF JOIN:

==========

- joining a table by itself is called as self join.

  (or)
- comparing a table data by itself is called as self join.
- self join can be implemented on a single table only.
- when we use self join we must create alias names on a table otherwise we cannot implement self join.
- we can create any no.of alias names on a single table but each alias name should be different name.
- when we create alias name on a table internally system is preparing a virtual table on each alias name and storing under buffer memory.
- self join can be implemented at two cases:

  Case-1: comparing a single column values by itself with in the table .

  Case-2: comparing two different columns values to each other with in the table.

Case-1: comparing a single column values by itself with in the table:

==

Ex:

waq to display employees details who are working in the same location where the employee "SMITH" is also working?

DEMO_TABLE:

=============

EX:

```
CREATE TABLE TEST(ENAME VARCHAR(10),LOC VARCHAR(10));
INSERT INTO TEST
VALUES('SMITH','HYD'),('ALLEN','MUMBAI'),('JONES','HYD'),('ADAMS','CHENNAI');
```

SOL:

```
SELECT T2.ENAME,T2.LOC FROM TEST T1 JOIN TEST T2  ON T1.LOC=T2.LOC AND
T1.ENAME='SMITH';
```

Ex:
waq to display employees whose salary is same as the salary of the employee "SCOTT"?

```
SELECT E1.ENAME,E1.SAL FROM EMP E1 JOIN EMP E2
ON E1.SAL=E2.SAL AND E2.ENAME='SCOTT';
```

05-11-2024:
==========
Case-2: comparing two different columns values to each other with in the table:
==
Ex:
waq to display managers and their employees from emp table?

```
SELECT M.ENAME AS MANAGERS,E.ENAME AS EMPLOYEES
FROM EMP E JOIN EMP M ON M.EMPNO=E.MGR;
```

Ex:
waq to display employees who are working under "BLAKE" manager?

```
SELECT M.ENAME AS MANAGERS,E.ENAME AS EMPLOYEES FROM
EMP E JOIN EMP M ON M.EMPNO=E.MGR  AND / WHERE  M.ENAME='BLAKE';
```

Ex:
waq to display BLAKE manager?

```
SELECT M.ENAME AS MANAGERS,E.ENAME AS EMPLOYEES
FROM EMP E JOIN EMP M ON M.EMPNO=E.MGR AND  E.ENAME='BLAKE';
```

Ex:
waq to display employees who are joined before their manager?

```
SELECT E.ENAME AS EMPLOYEES,E.HIREDATE AS E_DOJ,M.ENAME AS MANAGERS,
M.HIREDATE AS M_DOJ FROM EMP E JOIN EMP M ON M.EMPNO=E.MGR
AND E.HIREDATE<M.HIREDATE;
```

Ex:
waq to display employees whose salary is more than their manager salary?

```
SELECT E.ENAME AS EMPLOYEES,E.SAL AS E_SALARY,M.ENAME AS MANAGERS,
M.SAL AS M_SALARY FROM EMP E JOIN EMP M ON M.EMPNO=E.MGR
AND E.SAL>M.SAL;
```

How to join more than two tables:
=========================
syntax:
======

```
SELECT * FROM <TN1> <join key> <TN2> ON <JOIN CONDITION1>
<join key> <TN3> ON <JOIN CONDITION2>
```

```
<join key> <TN4> ON <JOIN CONDITION3>
.......................................................................
.......................................................................
<join key> <TN n> ON <JOIN CONDITION n-1>;
```

DEMO_TABLE:
============
EX:

```
CREATE TABLE REGISTER(REGNO INT,REGDATE DATE,CID INT);
INSERT INTO REGISTER
VALUES(1001,'2024-10-03',1),(1002,'2024-10-28',2),(1003,'2024-11-05',NULL);

SELECT * FROM COURSE;
SELECT * FROM STUDENT;
SELECT * FROM REGISTER;
```

EX:

```
SELECT * FROM STUDENT S INNER JOIN COURSE C ON S.CID=C.CID
INNER JOIN REGISTER R ON C.CID=R.CID;
```

===
======