**JOINS**

**Joins is taking data from more than one tables.**

**Three types of Joins:**

1. **INNER**
2. **OUTER**
3. **CROSS**

**Inner-Join**

**Retrieving common records from the 2 tables on equality condition –**

**To see names and department names from emp and dept table**

**ANSI 89 Method:**

Select ename, dname

From emp, dept

Where emp.deptno = dept.deptno;

**ANSI 92 Method:**

select ename, dname

from emp inner join dept

on emp.deptno = dept.deptno;

In the Select Statement if the common column has to be displayed then it has to be prefixed by table name in both ANSI and NON-ANSI methods.

select ename, dname, **emp.deptno**

from emp join dept

on emp.deptno = dept.deptno;

Best practice will be using table alias:

select e.ename, d.dname, e.deptno

from emp **e** join dept **d**

on e.deptno = d.deptno;

Filtering records along with join:

select e.ename, d.dname, e.sal

from emp e join dept d

on e.deptno = d.deptno

where e.sal > 2000

Can display all the columns of a table by prefixing **table name.\*** or **table alias.\***

select e.\*

from emp e join dept d

on e.deptno = d.deptno

**Demo of multiple columns matching in two tables**.

Drop Table L1;

Drop Table L2;

go

create table L1**(a int, b int, c int,** d int, e int, f int);

create table L2(**a int, b int, c int**, p int, q int, r int);

Insert Into L1 Values(**1,1,1**, 101,102,103);

Insert Into L1 Values(1,1,2, 104,105,106);

Insert Into L1 Values(**1,1,3**,107,108,109);

Insert Into L1 Values(1,2,1,110,111,112);

Insert Into L1 Values**(1,2,2**,113,114,115);

Insert Into L1 Values(1,2,3,116,117,118);

Insert Into L2 Values(**1,1,1**, 201,202,203);

Insert Into L2 Values(**1,1,3**,204,205,206);

Insert Into L2 Values(1,2,9,207,208,209);

Insert Into L2 Values(**1,2,2**,210,211,212);

Insert Into L2 Values(1,7,3,213,214,215);

Select \* from L1;

Select \* from L2;

Select L1.a, L1.b, L1.c, L1.d,L2.p, L2.q

from L1 Join L2

on L1.a = L2.a; -- 30 Rows

Select L1.a, L1.b, L1.c, L1.d,L2.p, L2.q

from L1 Join L2

on L1.a = L2.a and L1.b = L2.b; -- 12 Rows

Select L1.a, L1.b, L1.c, L1.d,L2.p, L2.q

from L1 Join L2

on L1.a = L2.a and L1.b = L2.b and L1.c = L2.c -- Only 3 Rows

**Example 1 🡪 Retrieving common records from the 3 tables on equality condition –**

create table e

(empno Integer,

ename varchar(10),

deptno Integer);

create table d

(deptno Integer,

dname varchar(10),

pcode varchar(5));

create table p

(pcode varchar(5),

pname varchar(10));

--Insert records in e

insert into e

values(1,'A',10);

insert into e

values(2,'B',20);

insert into e

values(3,'C',30);

insert into e

values(4,'D',20);

-- Insert records in d

insert into d

values(10,'Prod','P1');

insert into d

values(20,'Maint','P2');

insert into d

values(30,'Insp','P1');

--Insert records in p

insert into p

values('P1','Nuts');

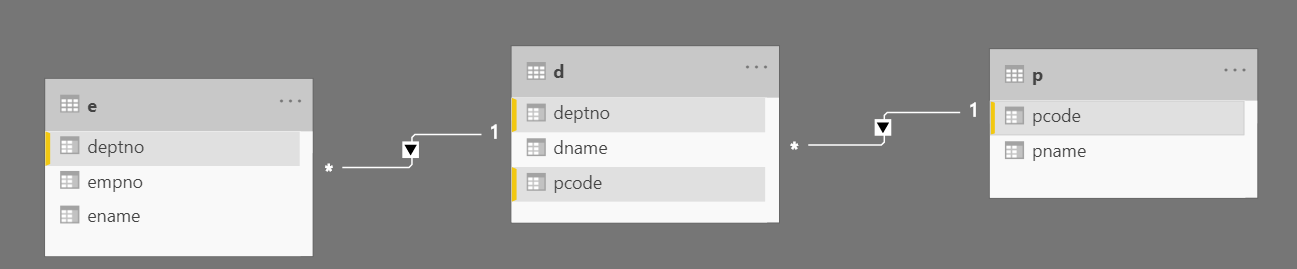
insert into p

values('P2','Bolts');

insert into p

values('P3','Gears');

**Between e and d deptno is common and between d and p, pcode is common**

****

**To see names, department names and product names from the 3 tables on matching values –**

**ANSI 89 Method:**

select ename, dname, pname

from e, d, p

where e.deptno = d.deptno AND d.pcode = p.pcode;

**ANSI 92 Method:**

select ename, dname, pname

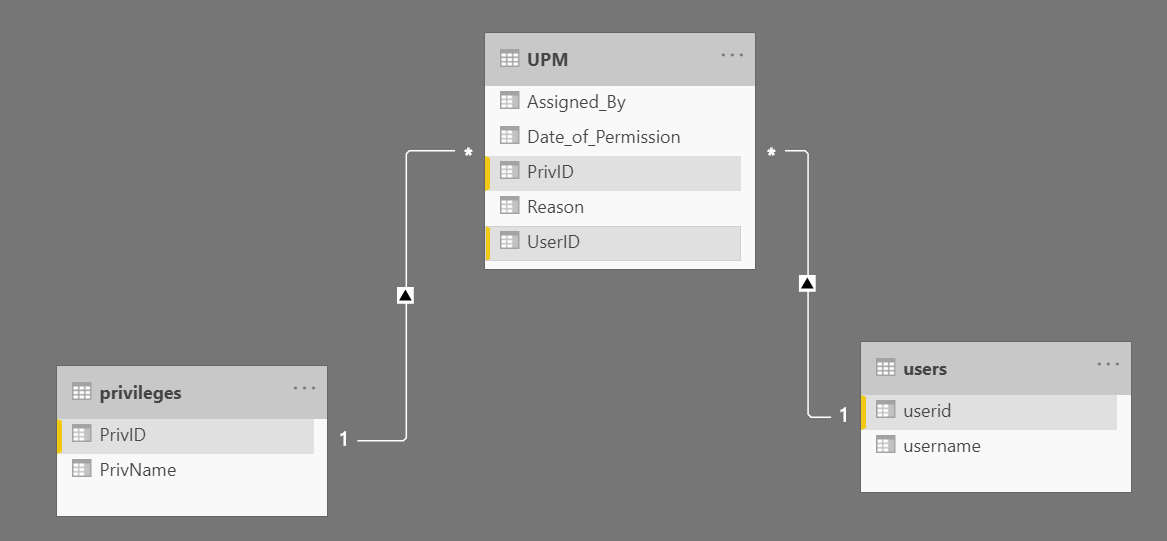
from e join d

on e.deptno = d.deptno

join p

on d.pcode = p.pcode;

**Example 2 of 3 Tables Joins 🡪** **Resolving Many to Many relationship using the Junction Table.**



* An user will have one or multiple permissions.
* At the same time a single permission can be given to multiple users!!

create table Users

(UserID varchar(3) Primary Key,

Username varchar(30)

)

create table Privileges

(PrivID varchar(3) Primary Key,

PrivName varchar(30)

)

create table UPM

(UserID varchar(3) References users,

PrivID varchar(3) References privileges,

Date\_of\_Permission Date,

Assigned\_By varchar(30),

Reason varchar(500)

)

Insert into Users Values('U1','Smith')

Insert into Users Values('U2','Martin')

Insert into Users Values('U3','Allen')

Insert into Privileges Values('P1','Select')

Insert into Privileges Values('P2','Insert')

Insert into Privileges Values('P3','Update')

Insert into Privileges Values('P4','Delete')

Select \* from Users

Select \* from Privileges

Insert into UPM Values('U1','P1','12-Dec-2018','Roger','Project Usage')

Insert into UPM Values('U1','P2','27-Jan-2019','Roger','Project Usage')

Insert into UPM Values('U1','P4','31-Mar-2019','King','POC Usage')

Insert into UPM Values('U2','P3','21-Dec-2018','Roger','Project Usage')

Insert into UPM Values('U3','P2','2-Jan-2019','James','Client Demo Usage')

Insert into UPM Values('U3','P4','6-Jan-2019','James','Client Demo Usage')

select \* from UPM

Select \* from Users

Select \* from Privileges

select \* from UPM

---------------------------------------------------------------

-- Display User Name and corresponding Privilege Name

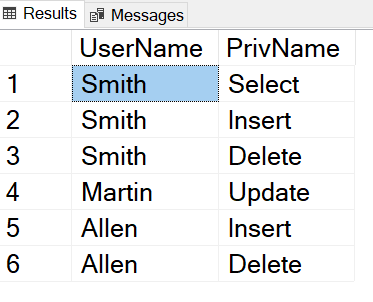
Select U.UserName, P.PrivName

From Users U Join UPM UP

On U.UserID = UP.UserID

Join Privileges P

On UP.PrivID = P.PrivID



**Left Outer Join**

**To take matching records from both the tables and all remaining records from the left table.**

create table emp1

(empno varchar(5),

ename varchar(15),

deptno Integer);

create table dept1

(deptno Integer,

dname varchar(40));

insert into emp1

values('E1','Smith',10);

insert into emp1

values('E2','Roger',20);

insert into emp1

values('E3','Martin',10);

insert into emp1

values('E4','Kim',30);

insert into emp1

values('E5','Glen',null);

insert into emp1

values('E6','Richards',80);

insert into dept1

values(10,'Accounts');

insert into dept1

values(20,'Production');

insert into dept1

values(30,'Marketing');

insert into dept1

values(40,'Inspection');

insert into dept1

values(50,'Research and Development');

In the tables emp1 and dept1, both the tables have some records non-matching.

To see names of the employee and their departments and also the names of the employees who do not have any department

select ename,dname

from emp1 **LEFT** outer join dept1

on emp1.deptno = dept1.deptno

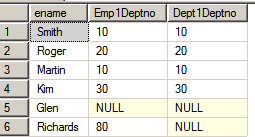
**Generally Left outer join is used to find unmatched records within two tables:**

To display ename and deptno from emp1 table which are not in dept1 table --

select ename, emp1.deptno as Emp1Deptno, dept1.deptno as Dept1Deptno

from emp1 LEFT outer join dept1

on emp1.deptno = dept1.deptno

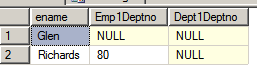


select ename, emp1.deptno as Emp1Deptno, dept1.deptno as Dept1Deptno

from emp1 LEFT outer join dept1

on emp1.deptno = dept1.deptno

**where dept1.deptno is null**



**Note 🡪 Where clause gets applied on the entire output of Left Outer Join result set.**

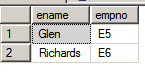
So finally the query will be:

select ename, emp1.empno

from emp1 LEFT outer join dept1

on emp1.deptno = dept1.deptno

where dept1.deptno is null



**Right Outer Join**

**To take matching records from both the tables and all remaining records from the right table.**

To see the names of the employees and their department names and also the department names which do not have any employee;

select ename, dname

from emp1 **RIGHT** outer join dept1

on emp1.deptno = dept1.deptno

**Full Outer Join**

**First Matching records from both tables, then remaining records from left table and then the remaining records from the right table are displayed.**

To see employee names and their departments, employees who do not have department as well as department names which are not having any employee

select ename, dname

from emp1 **FULL** outer join dept1

on emp1.deptno = dept1.deptno;

**Cross Join**

No join condition. So each row of first table gets combined with each record of the other.

Cartesian product takes place.

select ename, dname

from emp **CROSS** join dept;

Cross join is effective for financial applications such as calculations of interest rates for each month.

create table rates

(scheme char(2),

roi float);

create table period

(code char(5),

month float);

insert into rates

values('s1',5);

insert into rates

values('s2',5.5);

insert into rates

values('s3',6);

insert into rates

values('s4',6.5);

insert into rates

values('s5',7);

insert into rates

values('s6',7.5);

insert into rates

values('s7',8);

insert into period

values('c1',1);

insert into period

values('c2',2);

insert into period

values('c3',3);

insert into period

values('c4',4);

insert into period

values('c5',5);

insert into period

values('c6',6);

insert into period

values('c7',7);

insert into period

values('c8',8);

insert into period

values('c9',9);

insert into period

values('c10',10);

insert into period

values('c11',11);

insert into period

values('c12',12);

In the tables period and rates nothing is common. Still cross product can be achieved.

Now each roi is to be multiplied by each month for statistical report –

**Effective use of Cross join –**

Example

**ANSI 89 Method:**

select roi, month, roi\*month as "Interest"

from rates, period;

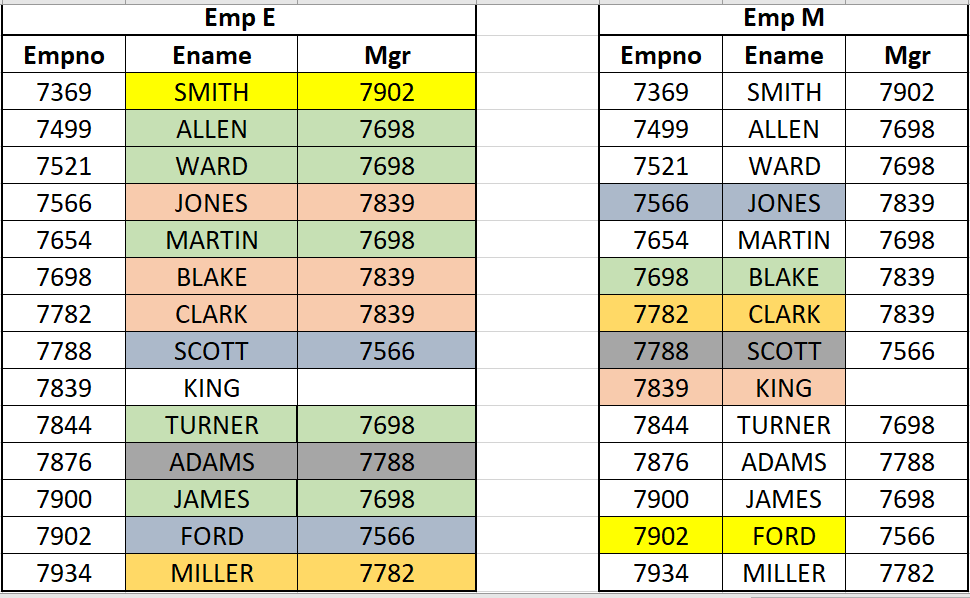
**ANSI 92 Method:**

select roi, month, roi\*month as "Interest"

from rates **CROSS** join period;

**Self- Join**

Example 1 🡪 To show hierarchical data



Select E.Ename as "Sub Oridnate", M.Ename as "Manager"

From Emp E Join Emp M

On E.Mgr= M.Empno

Example 2

-- To see if a record is duplicated

--Inserting a duplicate record of Ford (Only empno is different)

insert into emp

values(7903,'FORD','ANALYST','7566','03-DEC-81',3000,null,10);

select e.empno,e.ename,e.job

from emp e,emp e2

where e.empno <> e2.empno

and e.ename = e2.ename

and e.job = e2.job

and e.mgr = e2.mgr;