JOINS

SQL_QUERY:

```
create table DEPT (deptid varchar(5),dname varchar(20),est_year integer);
create table EMPLOYEE(regno integer,name varchar(20),dname varchar(20),city
varchar(10));
select * from DEPT
select * from EMPLOYEE

insert into DEPT values ('A','IT',2001);
insert into DEPT values ('B','ENG',2002);
insert into DEPT values ('C','MECH',2005);
insert into DEPT values ('D','CSE',2003);
insert into DEPT values ('E','CIVIL',2007);
insert into DEPT values ('F','CHEM',2009);
insert into DEPT values ('G','EEE',2011);


insert into EMPLOYEE values (1,'KESHAV','IT','CHN');
insert into EMPLOYEE values (2,'MAHENDRA','ENG','NCR');
insert into EMPLOYEE values (3,'RAJAT','IT','KER');
insert into EMPLOYEE values (4,'KIRAN','IT','CHN');
insert into EMPLOYEE values (5,'SUDESHI','IT','BAN');
insert into EMPLOYEE values (6,'RAJEEV','ENG','CHN');
insert into EMPLOYEE values (5,'RAM','CSE','KER');
insert into EMPLOYEE values (6,'AKASH','CIVIL','BAN');
insert into EMPLOYEE values (7,'NEHA','MATHS','KER');
insert into EMPLOYEE values (8,'VISHNU','MATHS','CHN');


select e.regno ,e.name,e.city,d.deptid ,d.dname,d.est_year  from DEPT d inner join
EMPLOYEE e on d.dname=e.dname;


select e.regno ,e.name,e.city,d.deptid ,d.dname,d.est_year  from EMPLOYEE e inner join
DEPT d on d.dname=e.dname;


select e.regno ,e.name,e.city,d.deptid ,d.dname,d.est_year  from DEPT d left join
EMPLOYEE e on d.dname=e.dname;

select e.regno ,e.name,e.city,d.deptid ,d.dname,d.est_year  from DEPT d right join
EMPLOYEE e on d.dname=e.dname;


select *  from DEPT d cross join EMPLOYEE e ;
select *  from EMPLOYEE e cross join DEPT d  ;


select a.regno ,a.name , a.city ,a.dname from EMPLOYEE a , EMPLOYEE e where
a.dname=e.dname;
```

INNER JOIN:

The INNER JOIN keyword selects all rows from both the tables as long as the condition satisfies. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e. value of the common field will be same.

SYNTAX:

```
select attribute-list from table1-name  inner join table2-name  on condition;
```

SQL_QUERY:

```
select e.regno ,e.name,e.city,d.deptid ,d.dname,d.est_year  from DEPT d inner join
EMPLOYEE e on d.dname=e.dname;
```

```
select e.regno ,e.name,e.city,d.deptid ,d.dname,d.est_year  from EMPLOYEE e inner join
DEPT d on d.dname=e.dname;
```

LEFT JOIN:

This join returns all the rows of the table on the left side of the join and matching rows for the table on the right side of join. The rows for which there is no matching row on right side, the result-set will contain *null*. LEFT JOIN is also known as LEFT OUTER JOIN

SYNTAX:

```
select attribute-list from left-table left join right-table on condition;
```

SQL_QUERY:

```
select e.regno ,e.name,e.city,d.deptid ,d.dname,d.est_year  from DEPT d left join
EMPLOYEE e on d.dname=e.dname;
```

RIGHT JOIN:

RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of join. The rows for which there is no matching row on left side, the result-set will contain *null*. RIGHT JOIN is also known as RIGHT OUTER JOIN.

SYNTAX:

```
select attribute-list from left-table right join right-table on condition;
```

SQL_QUERY:

```
select e.regno ,e.name,e.city,d.deptid ,d.dname,d.est_year  from DEPT d right join
EMPLOYEE e on d.dname=e.dname;
```

CROSS JOIN:

The SQL CROSS JOIN produces a result set which is the number of rows in the first table multiplied by the number of rows in the second table if no WHERE clause is used along with CROSS JOIN. This kind of result is called as Cartesian Product.

SYNTAX:

```
select *  from table1 cross join table2 ;
```

SQL_QUERY:

```
select *  from DEPT d cross join EMPLOYEE e ;
select *  from EMPLOYEE e cross join DEPT d  ;
```

SELF JOIN:

A Self-join is a join in which a table is joined with itself (which is also called Unary relationships), To join a table itself means that each row of the table is combined with itself and with every other row of the table.

SYNTAX:

```
select attribute-list from table-name, table-name where condition;
```

SQL_QUERY:

```
select a.regno ,a.name , a.city ,a.dname from EMPLOYEE a , EMPLOYEE e where
a.dname=e.dname;
```