Ex. No: 4 12.02.2025	JOIN OPERATION

AIM:

To execute independent sub-query, correlated sub-query and correlated sub-query using EXIST operator.

Employee table:

SQL> create table empl68(id number(4) primary key, name varchar(18) , age number(2) , area varchar(15) , workhours number(2));

Table created.

SQL> insert into empl68 values(68 , 'jayashangav' , 18 , 'perundurai' , 19);

1 row created.

SQL> insert into empl68 values(73 , 'jegan' , 19 , 'salem' , 20);

1 row created.

SQL> insert into empl68 values(74 , 'sanjay' , 20 , 'erode' , 21);

1 row created.

SQL> insert into empl68 values(80 , 'karthik' , 21 , 'vijayapuri' , 22);

1 row created.

SQL> insert into empl68 values(82 , 'kavin' , 22 , 'thingalur' , 23);

1 row created.

SQL> select * from empl68;

```
        ID   NAME        AGE    AREA  WORKHOURS
---------- -------- ---------- ----- ------------
        68 jayashangav      18 perundurai   19
        73 jegan            19 salem        20
        74 sanjay           20 erode        21
        80 karthik          21 vijayapuri   22
        82 kavin            22 thingalur    23
```

Projects table:

SQL> create table pro68(proid number(3) primary key , id number(4) , proname varchar(5) , prostdate date);

Table created.

SQL> insert into pro68 values (1, 68, 'pro1', to_date('28-02-2024', 'dd-mm-yyyy'));

1 row created.

SQL> insert into pro68 values (2, 73, 'pro2', to_date('01-03-2024', 'dd-mm-yyyy'));

1 row created.

```
SQL> insert into pro68 values (3, 74, 'pro3', to_date('04-03-2024', 'dd-mm-yyyy'));

1 row created.

SQL> insert into pro68 values (4, 76, 'pro4', to_date('10-03-2024', 'dd-mm-yyyy'));

1 row created.

SQL> insert into pro68 values (5, 80, 'pro5', to_date('21-03-2024', 'dd-mm-yyyy'));

1 row created.

SQL> insert into pro68 values (6, 82, 'pro6', to_date('24-03-2024', 'dd-mm-yyyy'));

1 row created.

SQL> insert into pro68 values (7, 84, 'pro7', to_date('28-03-2024', 'dd-mm-yyyy'));

1 row created.
SQL> insert into pro68 values (8, 93, 'pro8', to_date('30-03-2024', 'dd-mm-yyyy'));

1 row created.

SQL> select * from pro68;
```

PROID	ID	PRONAME	PROSTDATE
1	68	pro1	28-FEB-2024
2	73	pro2	01-MAR-2024
3	74	pro3	04-MAR-2024
4	76	pro4	10-MAR-2024
5	80	pro5	21-MAR-2024
6	82	pro6	24-MAR-2024
7	84	pro7	28-MAR-2024
8	93	pro8	30-MAR-2024

8 rows selected.

INNER JOIN:

SQL> select empl68.id, empl68.name, empl68.area, pro68.proid, pro68.proname from empl68 inner join pro68 on empl68.id = pro68.id;

ID	NAME	AREA	PROID	PRONAME
68	jayashangav	perundurai	1	pro1
73	jegan	salem	2	pro2
74	sanjay	erode	3	pro3
80	karthik	vijayapuri	5	pro5
82	kavin	thingalur	6	pro6

FULL JOIN:

SQL> select pro68.proid, empl68.name, pro68.proname, empl68.area from empl68 full join pro68 on empl68.id = pro68.id;

```
  PROID      NAME          PRONAME    AREA
----------  -----------    -------    ---------
    1  jayashangav    pro1       perundurai
    2  jegan          pro2       salem
    3  sanjay         pro3       erode
    5  karthik        pro5       vijayapuri
    6  kavin          pro6       thingalur
    7                 pro7
    8                 pro8
```

LEFT OUTER JOIN:

SQL> select pro68.proid, empl68.name, pro68.proname, empl68.workhours from empl68 left join pro68 on empl68.id = pro68.id;

```
 PROID      NAME          PRONAME    WORKHOURS
----------  -----------    -------    ----------
      1  jayashangav    pro1       19
      2  jegan          pro2       20
      3  sanjay         pro3       21
      5  karthik        pro5       22
      6  kavin          pro6       23
```

RIGHT OUTER JOIN:

SQL> select pro68.proid, empl68.name, pro68.proname, empl68.workhours from empl68 right join pro68 on empl68.id = pro68.id;

```
  PROID      NAME          PRONAME    WORKHOURS
----------  -----------   -------    ----------
         1  jayashangav   pro1       19
         2  jegan         pro2       20
         3  sanjay        pro3       21
         5  karthik       pro5       22
         6  kavin         pro6       23
         7                pro7
         8                pro8
```

NATURAL JOIN:

SQL> select id, name, proname from empl68 natural join pro68;

```
    ID       NAME          PRONAME
----------  -----------   -------
        68  jayashangav   pro1
        73  jegan         pro2
        74  sanjay        pro3
        80  karthik       pro5
        82  kavin         pro6
```

CROSS JOIN:

SQL> select empl68.id, empl68.name, pro68.proname from empl68 cross join pro68;

ID	NAME	PRONAME
68	jayashangav	pro1
68	jayashangav	pro2
68	jayashangav	pro3
68	jayashangav	pro4
68	jayashangav	pro5
68	jayashangav	pro6
68	jayashangav	pro7
68	jayashangav	pro8
73	jegan	pro1
73	jegan	pro2
73	jegan	pro3
73	jegan	pro4
73	jegan	pro5
73	jegan	pro6
73	jegan	pro7
73	jegan	pro8
74	sanjay	pro1
74	sanjay	pro2
74	sanjay	pro3
74	sanjay	pro4
74	sanjay	pro5
74	sanjay	pro6
74	sanjay	pro7
74	sanjay	pro8

```
        ID       NAME         PRONAME
----------  -----------   -------
        80  karthik       pro1
        80  karthik       pro2
        80  karthik       pro3
        80  karthik       pro4
        80  karthik       pro5
        80  karthik       pro6
        80  karthik       pro7
        80  karthik       pro8
        82  kavin         pro1
        82  kavin         pro2
        82  kavin         pro3
        82  kavin         pro4
        82  kavin         pro5
        82  kavin         pro6
        82  kavin         pro7
        82  kavin         pro8

40 rows selected.
```

BETA JOIN:

SQL> select empl68.id, empl68.name, empl68.age from empl68

join pro68 on empl68.workhours > pro68.proid;

ID	NAME	AGE
68	jayashangav	18
68	jayashangav	18
68	jayashangav	18
68	jayashangav	18
68	jayashangav	18
73	jegan	19
73	jegan	19
73	jegan	19
73	jegan	19
73	jegan	19
73	jegan	19
74	sanjay	20
74	sanjay	20
74	sanjay	20
74	sanjay	20
74	sanjay	20
74	sanjay	20
74	sanjay	20
80	karthik	21
80	karthik	21
80	karthik	21
80	karthik	21

80	karthik	21
80	karthik	21
80	karthik	21
80	karthik	21
80	karthik	21
82	kavin	22
82	kavin	22
82	kavin	22
82	kavin	22
82	kavin	22
82	kavin	22
82	kavin	22
82	kavin	22
82	kavin	22
82	kavin	22
82	kavin	22

CONTENTS	MARKS ALLOTED	MARKS OBTAINED
Aim, Algorithm, SQL, PL/SQL	30	
Execution and Result	20	
Viva	10	
Total	60	

RESULT:

Thus, inner join, left join, right join, natural join, full join, cross join using two tables were executed successfully.