

DBMS LAB ASSIGNMENT 8 MEHER SHRISHTI NIGAM 20BR1193

CHALLENGING LAB EXERCISE

1. Create the suitable tables and populate the data as per the table given. [6]

```
CREATE TABLE EMP_20BR1193 (EMPLOYEEID INT PRIMARY KEY, FIRSTNAME VARCHAR2(10), LASTNAME  
VARCHAR2(10), SALARY INT, JOINDATE DATE, DEPARTMENT VARCHAR(10), GENDER VARCHAR2(10));
```

INSERT ALL

```
    INTO EMP_20BR1193 VALUES (1, 'Sriram', 'Kumar', 60000, '01-Dec-2013', 'IT', 'Male' )
```

```
    INTO EMP_20BR1193 VALUES (2, 'Shanthi', 'Devi', 50000, '10-Oct-2014', 'HR', 'Female')
```

```
    INTO EMP_20BR1193 VALUES (3, 'Sham', 'Sundar', 100000, '20-Dec-2014', 'IT', 'Male')
```

```
    INTO EMP_20BR1193 VALUES (4, 'Ram', 'Kishan', 43000, '20-Dec-2014', 'HR', 'Male')
```

```
    INTO EMP_20BR1193 VALUES (5, 'Rahul', 'Dravid', 83000, '20-Dec-2014', 'Payroll', 'Male')
```

```
    INTO EMP_20BR1193 VALUES (6, 'Mahesh', 'Kumar', 35000, '20-Dec-2014', 'HR', 'Male')
```

```
SELECT * FROM DUAL;
```

```
CREATE TABLE PROJECT_20BR1193 (PROJECTID INT PRIMARY KEY, EMPLOYEEID INT, PROJECTNAME  
VARCHAR2(10), CONSTRAINT FK_G FOREIGN KEY (EMPLOYEEID) REFERENCES  
EMP_20BR1193(EMPLOYEEID));
```

INSERT ALL

```
    INTO PROJECT_20BR1193 VALUES (1, 1, 'Project1')
```

```
    INTO PROJECT_20BR1193 VALUES (2, 1, 'Project2')
```

```
    INTO PROJECT_20BR1193 VALUES (3, 1, 'Project3')
```

```
    INTO PROJECT_20BR1193 VALUES (4, 2, 'Project4')
```

```
    INTO PROJECT_20BR1193 VALUES (5, 3, 'Project5')
```

```
    INTO PROJECT_20BR1193 VALUES (6, 3, 'Project6')
```

```
    INTO PROJECT_20BR1193 VALUES (7, 3, 'Project7')
```

```
    INTO PROJECT_20BR1193 VALUES (8, 4, 'Project8')
```

```
    INTO PROJECT_20BR1193 VALUES (9, 5, 'Project9')
```

```
SELECT * FROM DUAL;
```

```
SQL> CREATE TABLE EMP_20BRS1193 (EMPLOYEEID INT PRIMARY KEY, FIRSTNAME VARCHAR2(10), LASTNAME VARCHAR2(10), SALARY INT, JOINDATE DATE, DEPARTMENT VARCHAR(10), GENDER VARCHAR2(10));

Table created.

SQL> INSERT ALL
  2 INTO EMP_20BRS1193 VALUES (1, 'Sriram', 'Kumar', 60000, '01-Dec-2013', 'IT', 'Male' )
  3 INTO EMP_20BRS1193 VALUES (2, 'Shanthi', 'Devi', 50000, '10-Oct-2014', 'HR', 'Female')
  4 INTO EMP_20BRS1193 VALUES (3, 'Sham', 'Sundar', 100000, '20-Dec-2014', 'IT', 'Male')
  5 INTO EMP_20BRS1193 VALUES (4, 'Ram', 'Kishan', 43000, '20-Dec-2014', 'HR', 'Male')
  6 INTO EMP_20BRS1193 VALUES (5, 'Rahul', 'Dravid', 83000, '20-Dec-2014', 'Payroll', 'Male')
  7 INTO EMP_20BRS1193 VALUES (6, 'Mahesh', 'Kumar', 35000, '20-Dec-2014', 'HR', 'Male')
  8 SELECT * FROM DUAL;

6 rows created.

SQL> CREATE TABLE PROJECT_20BRS1193 (PROJECTID INT PRIMARY KEY, EMPLOYEEID INT, PROJECTNAME VARCHAR2(10), CONSTRAINT FK_G FOREIGN KEY (EMPLOYEEID ) REFERENCES EMP_20BRS1193(EMPLOYEEID));

Table created.

SQL> INSERT ALL
  2 INTO PROJECT_20BRS1193 VALUES (1, 1, 'Project1')
  3 INTO PROJECT_20BRS1193 VALUES (2, 1, 'Project2')
  4 INTO PROJECT_20BRS1193 VALUES (3, 1, 'Project3')
  5 INTO PROJECT_20BRS1193 VALUES (4, 2, 'Project4')
  6 INTO PROJECT_20BRS1193 VALUES (5, 3, 'Project5')
  7 INTO PROJECT_20BRS1193 VALUES (6, 3, 'Project6')
  8 INTO PROJECT_20BRS1193 VALUES (7, 3, 'Project7')
  9 INTO PROJECT_20BRS1193 VALUES (8, 4, 'Project8')
 10 INTO PROJECT_20BRS1193 VALUES (9, 5, 'Project9')
 11 SELECT * FROM DUAL;

9 rows created.

SQL> _
```

2. Get employee name, project name order by firstname from "Employee" and "Project" for those employees who have been assigned to project already. [3]

SELECT FIRSTNAME, LASTNAME, PROJECTNAME AS PROJ_NAME FROM PROJECT_20BRS1193 LEFT JOIN EMP_20BRS1193 ON EMP_20BRS1193.EMPLOYEEID = PROJECT_20BRS1193.EMPLOYEEID ORDER BY FIRSTNAME;

```
SQL> SELECT FIRSTNAME, LASTNAME, PROJECTNAME AS PROJ_NAME FROM PROJECT_20BRS1193 LEFT JOIN EMP_20BRS1193 ON
EMP_20BRS1193.EMPLOYEEID = PROJECT_20BRS1193.EMPLOYEEID ORDER BY FIRSTNAME;

FIRSTNAME  LASTNAME  PROJ_NAME
-----
Rahul      Dravid    Project9
Ram        Kishan    Project8
Sham       Sundar    Project5
Sham       Sundar    Project7
Sham       Sundar    Project6
Shanthi    Devi      Project4
Sriram     Kumar     Project1
Sriram     Kumar     Project2
Sriram     Kumar     Project3

9 rows selected.
```

3. Get employee name, project name order by firstname from "Employee" and "Project" for all employees even though they have not been assigned to a project. [3]

SELECT FIRSTNAME, LASTNAME, PROJECTNAME AS PROJ_NAME FROM PROJECT_20BRS1193 RIGHT JOIN EMP_20BRS1193 ON EMP_20BRS1193.EMPLOYEEID = PROJECT_20BRS1193.EMPLOYEEID ORDER BY FIRSTNAME;

```
SQL> SELECT FIRSTNAME, LASTNAME, PROJECTNAME AS PROJ_NAME FROM PROJECT_20BRS1193 RIGHT JOIN EMP_20BRS1193 ON
EMP_20BRS1193.EMPLOYEEID = PROJECT_20BRS1193.EMPLOYEEID ORDER BY FIRSTNAME;

FIRSTNAME  LASTNAME  PROJ_NAME
-----
Mahesh     Kumar
Rahul      Dravid    Project9
Ram        Kishan    Project8
Sham       Sundar    Project5
Sham       Sundar    Project7
Sham       Sundar    Project6
Shanthi    Devi      Project4
Sriram     Kumar     Project1
Sriram     Kumar     Project2
Sriram     Kumar     Project3

10 rows selected.
```

4. Get employee name, project name order by firstname from "Employee" and "Project" for all employee. If project is not assigned to a particular employee display the project name with the text "No Project Assigned" [3]

```
SELECT FIRSTNAME, LASTNAME, NVL(PROJECTNAME, 'No Project Assigned') AS PROJ_NAME FROM  
PROJECT_20BRS1193 RIGHT JOIN EMP_20BRS1193 ON EMP_20BRS1193.EMPLOYEEID =  
PROJECT_20BRS1193.EMPLOYEEID ORDER BY FIRSTNAME;
```

```
SQL> SELECT FIRSTNAME, LASTNAME, NVL(PROJECTNAME, 'No Project Assigned') AS PROJ_NAME FROM PROJECT_20BRS1193  
RIGHT JOIN EMP_20BRS1193 ON EMP_20BRS1193.EMPLOYEEID = PROJECT_20BRS1193.EMPLOYEEID ORDER BY FIRSTNAME;
```

FIRSTNAME	LASTNAME	PROJ_NAME
Mahesh	Kumar	No Project Assigned
Rahul	Dravid	Project9
Ram	Kishan	Project8
Sham	Sundar	Project5
Sham	Sundar	Project7
Sham	Sundar	Project6
Shanthi	Devi	Project4
Sriram	Kumar	Project1
Sriram	Kumar	Project2
Sriram	Kumar	Project3

10 rows selected.

5. How many employees get salary which is greater than the average salary of the employees? [2]

```
SELECT COUNT(*) FROM EMP_20BRS1193 WHERE SALARY > (SELECT AVG(SALARY) FROM EMP_20BRS1193);
```

```
SQL> SELECT COUNT(*) FROM EMP_20BRS1193 WHERE SALARY > (SELECT AVG(SALARY) FROM EMP_20BRS1193);
```

COUNT(*)
2

6. List the employee names of employees who work in multiple projects. [2]

```
SELECT DISTINCT FIRSTNAME, LASTNAME FROM EMP_20BRS1193 E INNER JOIN PROJECT_20BRS1193 P ON  
E.EMPLOYEEID = P.EMPLOYEEID WHERE E.EMPLOYEEID IN (SELECT EMPLOYEEID FROM  
PROJECT_20BRS1193 GROUP BY EMPLOYEEID HAVING COUNT(EMPLOYEEID) > 1);
```

```
SQL> SELECT DISTINCT FIRSTNAME, LASTNAME FROM EMP_20BRS1193 E INNER JOIN PROJECT_20BRS1193 P ON E.EMPLOYEEID  
= P.EMPLOYEEID WHERE E.EMPLOYEEID IN (SELECT EMPLOYEEID FROM PROJECT_20BRS1193 GROUP BY EMPLOYEEID HAVING C  
OUNT(EMPLOYEEID) > 1);
```

FIRSTNAME	LASTNAME
Sriram	Kumar
Sham	Sundar

7. Display the employee name of the employee who draws highest salary in each department. [3]

```
SELECT FIRSTNAME, SALARY, DEPARTMENT FROM EMP_20BRS1193 WHERE SALARY = ANY(SELECT  
MAX(SALARY) FROM EMP_20BRS1193 GROUP BY DEPARTMENT) ORDER BY FIRSTNAME;
```

```
SQL> SELECT FIRSTNAME, SALARY, DEPARTMENT FROM EMP_20BRS1193 WHERE SALARY = ANY(SELECT MAX(SALARY) FROM EMP_  
20BRS1193 GROUP BY DEPARTMENT) ORDER BY FIRSTNAME;
```

FIRSTNAME	SALARY	DEPARTMENT
Rahul	83000	Payroll
Sham	100000	IT
Shanthi	50000	HR

8. Display the senior-most employee in each department based on joining date. [3]

SELECT * FROM EMP_20BRS1193 E WHERE JOINDATE = ANY(SELECT MIN(JOINDATE) FROM EMP_20BRS1193 WHERE E.DEPARTMENT = DEPARTMENT);

```
SQL> SELECT * FROM EMP_20BRS1193 E WHERE JOINDATE = ANY(SELECT MIN(JOINDATE) FROM EMP_20BRS1193 WHERE E.DEPARTMENT = DEPARTMENT);
```

EMPLOYEEID	FIRSTNAME	LASTNAME	SALARY	JOINDATE	DEPARTMENT	GENDER
1	Sriram	Kumar	60000	01-DEC-13	IT	Male
2	Shanthi	Devi	50000	10-OCT-14	HR	Female
5	Rahul	Dravid	83000	20-DEC-14	Payroll	Male