# Keegan Vaz SE CMPN B 242 192120

# ST. FRANCIS INSTITUTE OF TECHNOLOGY

MT. POINSUR, BORIVALI (W), MUMBAI

# LAB MANUAL

### **EXPERIMENT NO. 6**

Aim: - Perform Join operations and Complex queries.

Theory:-

1. Different types of Joins.

# 1. SQL JOINS:

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

Here are the different types of the JOINs in SQL:

1. **EQUI JOIN:** Most joins are "equi-joins" where the data from a column in one table exactly matches data in the column of another table. (P.K & F.K).

**SELECT \* FROM** 

Employee e, department d WHERE

e.DepartmentID = d.DepartmentID;

2. **NATURAL JOIN:** Natural join is basically an equijoin followed by removal of the superfluous attributes.

SELECT \* FROM employee

NATURAL JOIN department;

3. (INNER) JOIN: Returns records that have matching values in both

tables SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

INNER JOIN table2

ON table 1.matching column = table 2.matching column;

4. **LEFT (OUTER) JOIN**: Returns all records from the left table, and the matched records from the right table.

SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

LEFT JOIN table2

ON table 1.matching column = table 2.matching column;

5. **RIGHT (OUTER) JOIN**: Returns all records from the right table, and the matched records from the left table.

SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

RIGHT JOIN table2

ON table 1.matching column = table 2.matching column;

6. **FULL (OUTER) JOIN**: Returns all records when there is a match in either left or right table.

SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

FULL JOIN table2

ON table 1.matching column = table 2.matching column;

7. **SELF JOIN:** A self-join is joining a table to itself

SELECT a.column name, b.column name...

FROM table1 a, table1 b

WHERE a.common filed = b.common field;

2. Set Comparison & set membership operator :

## **SET COMPARISON OPERATORS:**

A sub query is a Select-From-Where expression that is nested within another query. Sub query is to perform tests for

- 1. Set membership
  - o IN tests for presence of set membership
    - As Value
      - Select distinct customer-name from borrower where address in ('Mumbai', 'Pune');
    - As Query
      - Select distinct customer-name from borrower where customer-name in (select customer-name from depositor);
  - o NOT IN tests for absence of set membership
    - As Value
      - Select distinct customer-name from borrower where customer-name not in ('Smith', 'Jones')
    - As Query
      - Select distinct customer-name from borrower where customer-name not in (select customer-name from depositor)
- 2. Make Set comparisons
  - o Some or any Clause: greater than at least one
    - o Select b\_name from Branch Where assets >some (Select assets from branch where b city='Brooklyn');

# o all Clause: greater than all

 Select b\_name from Branch Where assets >all (Select assets from branch where b city='Brooklyn');

### Lab Exercise:-

1. For the following relational Schema, Solve the queries:

Table:-Emp	oloyee				
Empld	EmpName	Department	ContactNo	Emailld	EmpHeadld
101	Isha	E-101	1234567890	abc@gmail.com	105
102	Priya	E-104	1234567890	abc@gmail.com	103
103	Neha	E-101	1234567890	abc@gmail.com	101
104	Rahul	E-102	1234567890	abc@gmail.com	105
105	abhishek	E-101	1234567890	abc@gmail.com	102

Table:-EmpDept						
Deptld	DeptName	DeptHead				
E-101	HR	105				
E-102	Development	101				
E-103	House Keeping					
E-104	sales	104				
E-105	purchase	104				

Table:-EmpSalary						
Empld Salary IsPermaner						
101	2000	Yes				
102	10000	Yes				
103	5000	No				
104	1900	Yes				
105	2300	yes				

### **TABLE EMPLOYEE:**

```
SQL> CREATE TABLE Employee (EmpId number(10) primary key, EmpName varchar2(20),
Department varchar2(20), ContactNo number(10), Emailid varchar2(20), EmpHeadId
 number(10));
Table created.
SQL> DESC Employee;
 Name
                                           Null?
                                                     Type
                                            NOT NULL NUMBER(10)
 EMPID
 EMPNAME
                                                     VARCHAR2(20)
 DEPARTMENT
                                                     VARCHAR2(20)
 CONTACTNO
                                                     NUMBER(10)
 EMAILID
                                                     VARCHAR2(20)
 EMPHEADID
                                                     NUMBER(10)
SQL> INSERT INTO Employee VALUES (101, 'Isha', 'E-
101', 1234567890, 'abc@gmail.com', 105);
```

```
1 row created.
SQL> INSERT INTO Employee VALUES (102, 'Priya', 'E-
104', 1234567890, 'abc@gmail.com', 103);
1 row created.
SQL> INSERT INTO Employee VALUES (103, 'Neha', 'E-
101', 1234567890, 'abc@gmail.com', 101);
1 row created.
SQL> INSERT INTO Employee VALUES (104, 'Rahul', 'E-
102', 1234567890, 'abc@gmail.com', 105);
1 row created.
SQL> INSERT INTO Employee VALUES (105, 'Abhishek', 'E-
101', 1234567890, 'abc@gmail.com', 102);
1 row created.
SQL> SELECT * FROM Employee;
   EMPID EMPNAME
                   DEPARTMENT CONTACTNO
EMAILID EMPHEADID
 101 Isha E-101 1234567890
                      105
abc@gmail.com
     102 Priya E-104 1234567890 ail.com 103
abc@gmail.com
     103 Neha E-101 1234567890 ail.com 101
abc@gmail.com
                   DEPARTMENT CONTACTNO
  EMPID EMPNAME
                EMPHEADID
104 Rahul E-102
abc@gmail.com 105
                                  1234567890
105 Abhishek E-101
abc@gmail.com 102
                                          1234567890
```

### **TABLE EMPSALARY:**

```
SQL> CREATE TABLE EmpSalary (EmpId number(10) primary key, Salary number(15), I
sPermanent varchar2(20));
Table created.
SQL> DESC EmpSalary;
                                        Null? Type
 Name
 EMPID
                                          NOT NULL NUMBER(10)
 SALARY
                                                  NUMBER(15)
 ISPERMANENT
                                                  VARCHAR2(20)
SQL> INSERT INTO EmpSalary VALUES (101, 2000, 'Yes');
1 row created.
SQL> INSERT INTO EmpSalary VALUES (102, 10000, 'Yes');
1 row created.
SQL> INSERT INTO EmpSalary VALUES (103, 5000, 'No');
1 row created.
SQL> INSERT INTO EmpSalary VALUES (104, 1900, 'Yes');
1 row created.
SQL> INSERT INTO EmpSalary VALUES (105, 2300, 'Yes');
1 row created.
SQL> SELECT * FROM EmpSalary;
    EMPID SALARY ISPERMANENT
      101
               2000 Yes
            10000 Yes
      102
               5000 No
      103
      104
               1900 Yes
      105 2300 Yes
```

# **TABLE EMPDEPT:**

```
SQL> CREATE TABLE EmpDept (DeptId varchar2(20) primary key, DeptName varchar2(20), DeptHead number(10) NULL);

Table created.
```

```
SQL> DESC EmpDept;
 Name
                                          Null? Type
                                          NOT NULL VARCHAR2(20)
 DEPTID
 DEPTNAME
                                                  VARCHAR2(20)
 DEPTHEAD
                                                   NUMBER(10)
SQL> INSERT INTO EmpDept VALUES ('E-101', 'HR', 105);
1 row created.
SQL> INSERT INTO EmpDept VALUES ('E-102', 'Development', 101);
1 row created.
SQL> INSERT INTO EmpDept VALUES ('E-103', 'House Keeping', NULL);
1 row created.
SQL> INSERT INTO EmpDept VALUES ('E-104', 'Sales', 104);
1 row created.
SQL> INSERT INTO EmpDept VALUES ('E-105', 'Purchase', 104);
1 row created.
SQL> SELECT * FROM EmpDept;
DEPTID
                    DEPTNAME DEPTHEAD
E-101
                    HR
                                                105
                    Development
E-102
                                                101
E-103
                    House Keeping
E-104
                    Sales
                                               104
E-105
                   Purchase
                                               104
```

#### **Oueries:**

- 1. For the above given Relational Schema, Perform the following Join Operations:
- i. Use Natural Join, to join employee and EmpDept tables.

```
SQL> SELECT * FROM Employee NATURAL JOIN EmpDept;

EMPID EMPNAME DEPARTMENT CONTACTNO
```

EMAILID		EMPHEADID	DEPTID	DEPTNAME
DEPTHEAD				
	Isha		E-101	1234567890
abc@gmail.co	OM	105	E-101	HR
103				
101 ]		105	E-101	1234567890
abc@gmail.co	OIII	105	E-102	Development
EMPID E	EMPNAME 		DEPARTMENT	CONTACTNO
EMAILID		EMPHEADID	DEPTID	DEPTNAME
DEPTHEAD				
404			F 404	4224567000
101 labc@gmail.co		105	E-101 E-103	1234567890 House Keeping
65				
101 ]	Isha		E-101	1234567890
abc@gmail.co	om	105	E-104	Sales
EMPTD F	FMPNAMF		DEPARTMENT	CONTACTNO
EMAILID		EMPHEADID	DEPTID	DEPTNAME
DEPTHEAD				
104				
104				
101 1			E-101	1234567890
abc@gmail.co	OM	105	E-105	Purchase
102 F	Priya		E-104	1234567890
EMPID E	EMPNAME		DEPARTMENT	CONTACTNO
EMAILID			DEDTTD	DEDTNAME
			<u></u>	DEPTNAME
DEPTHEAD				
abc@gmail.co	om	103	E-101	HR
105				
102	Priya		E-104	1234567890
102	<del>-                                    </del>			

abc@gmail.c	COM	103	E-102	Development
EMPID	EMPNAME		DEPARTMENT	CONTACTNO
			DEPTID	
DEPTHEAD				
102	Priya		E-104	1234567890
abc@gmail.c	COM	103	E-103	House Keeping
	Priya		E-104	1234567890
abc@gmail.c	COM	103	E-104	Sales
			DEPARTMENT	CONTACTNO
			DEPTID	
DEPTHEAD				
	Priya		E-104	1234567890
abc@gmail.d 104	COM	103	E-105	Purchase
	Neha		E-101	1234567890
abc@gmail.o	COM	101	E-101	HR
			DEPARTMENT	CONTACTNO
			DEPTID	DEPTNAME
DEPTHEAD				
105				
	Neha			1234567890
abc@gmail.c	COM	101	E-102	Development
103	Neha		E-101	1234567890
EMPID	EMPNAME		DEPARTMENT	CONTACTNO
EMAILID		EMPHEADID	DEPTID	DEPTNAME

DEPTHEAD			
abc@gmail.com	101	E-103	House Keeping
103 Neha		E-101	1234567890
abc@gmail.com	101	E-104	Sales
104			
EMPID EMPNAME		DEPARTMENT	CONTACTNO
EMAILID	EMPHEADID	DEPTID	DEPTNAME
DEPTHEAD			
103 Neha		E-101	1234567890
abc@gmail.com	101		Purchase
104			
104 Rahul		E-102	1234567890
abc@gmail.com	105		HR
105			
200			
ΕΜΡΤΌ ΕΜΡΝΔΜΕ		DEPARTMENT	CONTACTNO
EMAILID	EMPHEADTD	DEPTID	DEPTNAME
DEPTHEAD			
104 Rahul		E-102	1234567890
abc@gmail.com	105	E-102	Development
101	103	102	Development
101			
104 Rahul		E-102	1224567900
	105		1234567890
abc@gmail.com	105	E-103	House Keeping
CMDID CMDNAME		DEDARTMENT	CONTACTNO
		DEPARTMENT	
		DEDITO	
EMAILID			
DEPTHEAD			
404 B - L 1		F 103	1224567800
104 Rahul		E-102	1234567890
abc@gmail.com	105	E-104	Sales
104			

104 Rahul		E-102	1234567890
		DEPARTMENT	CONTACTNO
EMAILID	EMPHEADID	DEPTID	
DEPTHEAD			
abc@gmail.com 104	105	E-105	Purchase
105 Abhishek			1234567890
abc@gmail.com 105	102	E-101	HR
EMPID EMPNAME		DEPARTMENT	CONTACTNO
EMAILID			
DEPTHEAD			
105 Abhishek	102		1234567890
abc@gmail.com 101	102	E-102	Development
105 Abhishek		E-101	1234567890
abc@gmail.com	102	E-103	House Keeping
EMPID EMPNAME		DEPARTMENT	CONTACTNO
EMAILID			
DEPTHEAD			
105 Abhishek		E-101	1234567890
abc@gmail.com 104	102	E-104	Sales
105 Abhishek		E-101	1234567890
abc@gmail.com	102	E-105	Purchase
EMPID EMPNAME		DEPARTMENT	CONTACTNO
EMAILID DEPTHEAD		DEPTID	DEPTNAME

ii. Use Right outer Join, to join employee and EmpDept tables.

CODE:				
	Employee RIGH	HT OUTER JOIN	EmpDept	ON Employee.Department = E
<pre>mpDept.DeptId;</pre>				
EMPID EMPNAME		DEPARTMENT		CONTACTNO
EMAILID	EMPHEADID	DEPTID		DEPTNAME
DEPTHEAD				
101 Isha		E-101		1234567890
abc@gmail.com 105	105			HR
102 Priya abc@gmail.com 104	103	E-104 E-104		1234567890 Sales
EMPID EMPNAME		DEPARTMENT		
EMAILID	EMPHEADID	DEPTID		
DEPTHEAD				
103 Neha		E-101		1234567890
abc@gmail.com 105	101	E-101		HR
104 Rahul		E-102		1234567890
abc@gmail.com	105	E-102		Development
EMPID EMPNAME		DEPARTMENT		CONTACTNO
EMAILID	EMPHEADID	DEPTID		DEPTNAME
DEPTHEAD				
101				
105 Abhishek abc@gmail.com	102	E-101 E-101		1234567890 HR

105		
EMPID EMPNAME	DEPARTMENT	CONTACTNO
EMAILID		DEPTNAME
DEPTHEAD		
	E-103	House Keeping
104	E-105	Purchase
104		
7 rows selected.		

iii. Use Self Join, to display the names of Employee head along with each employee details.

			actNo, a.Emailid, b.Emp
as head from Employ	ee a,Employ	ee b where a.EmpHeadI	d=b.EmpId;
EMPID EMPNAME		DEPARTMENT	CONTACTNO
EMAILID	HEAD		
103 Neha		 E-101	1234567890
abc@gmail.com	Isha		
105 Abhishek		E-101	1234567890
abc@gmail.com	Priya		
102 Priya		E-104	1234567890
abc@gmail.com	Neha		
EMPID EMPNAME		DEPARTMENT	CONTACTNO
EMAILID			
101 Isha		E-101	1234567890
abc@gmail.com	Abhishek		

104 Rahul	E-102	1234567890
abc@gmail.com	Abhishek	

iv. Use Inner Join, to join employee and EmpDept tables for department id doesn't match.

SQL> SELECT * FROM En	nployee INNE	ER JOIN EmpDept ON Em	ployee.Department != EmpDep
<pre>t.DeptId;</pre>			
EMPTP EMPNAME		DEDARTMENT	CONTACTNO
		DEPARTMENT	
EMAILID			
DEPTHEAD			
101 Taba		F 101	1224567000
101 Isha abc@gmail.com	105	E-101 E-102	1234567890 Development
101	100	L-102	peveropilienc
101 Isha		E-101	1234567890
abc@gmail.com	105	E-103	House Keeping
EMPID EMPNAME		DEPARTMENT	CONTACTNO
EMAILID		DEPTID	
DEPTHEAD			
101 Isha		E-101	1234567890
abc@gmail.com	105	E-104	Sales
104			
101 Isha		F-101	1234567890
abc@gmail.com		E-105	Purchase
EMPID EMPNAME		DEPARTMENT	CONTACTNO
EMAILID	EMPHEADID	DEPTID	DEPTNAME
DEPTHEAD			
104			
102 Priya		E-104	1234567890

	403	F 404	UB
abc@gmail.com 105	103	E-101	HR
100			
102 Priya		E-104	1234567890
EMPID EMPNAME		DEPARTMENT	CONTACTNO
EMAILID		DEPTID	DEPTNAME
DEPTHEAD			
abc@gmail.com	103	E-102	Development
101			
102 Priya		E-104	1234567890
abc@gmail.com	103	E-103	House Keeping
FMPTD FMPNAMF		DEPARTMENT	CONTACTNO
EMAILID	EMPHEADID	DEPTID	DEPTNAME
DEPTHEAD			
102 Priya		E-104	1234567890
abc@gmail.com	103	E-105	Purchase
104			
103 Neha		E-101	1234567890
abc@gmail.com		E-102	Development
101			2 2 3 2 3 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
EMPID EMPNAME		DEPARTMENT	CONTACTNO
EMAILID			DEPTNAME
DEPTHEAD			
103 Neha		E-101	1234567890
abc@gmail.com	101		House Keeping
103 Neha		E-101	1234567890
abc@gmail.com	101	E-104	Sales
		DEDARTMENT	CONTACTNO
EMPID EMPNAME		DEPARTMENT	CONTACTNO
EMAILID	EMPHEADID	DEPTTD	DEPTNAME
ENTELLO	- ZHI-HEADID	DEI TID	DET HVAIL

DEPTHEAD			
104			
103 Neha abc@gmail.com	101	E-101 F-105	1234567890 Purchase
104			
104 Rahul		E-102	1234567890
EMPID EMPNAME		DEPARTMENT	CONTACTNO
EMAILID		DEPTID	
DEPTHEAD			
abc@gmail.com 105	105	E-101	HR
104 Rahul abc@gmail.com	105		1234567890 House Keeping
EMDID EMDNAME			CONTACTNO
EMPID EMPNAME		DEPARTMENT	CONTACTNO
EMAILID	EMPHEADID		DEPTNAME
EMAILID	EMPHEADID	DEPTID	DEPTNAME
EMAILID  DEPTHEAD  104 Rahul	EMPHEADID	DEPTID  E-102	DEPTNAME
EMAILID  DEPTHEAD	EMPHEADID	DEPTID	DEPTNAME
EMAILID  DEPTHEAD  104 Rahul  abc@gmail.com  104  104 Rahul	EMPHEADID	DEPTID  E-102	DEPTNAME  1234567890 Sales  1234567890
EMAILID  DEPTHEAD  104 Rahul  abc@gmail.com	EMPHEADID	DEPTID E-102 E-104	DEPTNAME 1234567890 Sales
EMAILID  DEPTHEAD  104 Rahul  abc@gmail.com  104  104 Rahul  abc@gmail.com  104  EMPID EMPNAME	EMPHEADID 105	E-102 E-102 E-105  DEPARTMENT	DEPTNAME  1234567890 Sales  1234567890 Purchase  CONTACTNO
EMAILID  DEPTHEAD  104 Rahul  abc@gmail.com  104  104 Rahul  abc@gmail.com  104  EMPID EMPNAME  EMAILID	EMPHEADID  105  105  EMPHEADID	E-102 E-105  DEPARTMENT DEPTID	DEPTNAME  1234567890 Sales  1234567890 Purchase  CONTACTNO DEPTNAME
EMAILID  DEPTHEAD  104 Rahul  abc@gmail.com  104  104 Rahul  abc@gmail.com  104  EMPID EMPNAME  EMAILID	EMPHEADID  105  105  EMPHEADID	E-102 E-102 E-105 DEPARTMENT	DEPTNAME  1234567890 Sales  1234567890 Purchase  CONTACTNO DEPTNAME
EMAILID  DEPTHEAD  104 Rahul  abc@gmail.com  104  104 Rahul  abc@gmail.com  104  EMPID EMPNAME   EMAILID  DEPTHEAD	EMPHEADID  105  105  EMPHEADID	DEPTID  E-102 E-105  DEPARTMENT  DEPTID  DEPTID	DEPTNAME  1234567890 Sales  1234567890 Purchase  CONTACTNO DEPTNAME

105	Abhishek		E-101	1234567890
abc@gmail.c		102	E-103	House Keeping
C <b>B</b>				
EMDID	EMDNIAME		DEDARTMENT	CONTACTNO
EMPID	EMPNAME		DEPARTMENT	CONTACTNO
EMAILID		EMPHEADID	DEPTID	DEPTNAME
DEPTHEAD				
105	Abhishek		E-101	1234567890
abc@gmail.c	com	102	E-104	Sales
104				
105	Abhishek		E-101	1234567890
EMPTD	EMPNAME		DEPARTMENT	CONTACTNO
LIII 10	LITT NAME			CONTACTNO
				DEDTUME
EMAILID		EMPHEADID	DEPTID	DEPTNAME
DEPTHEAD				
abc@gmail.c	com	102	E-105	Purchase
104				
104				
20 rows sel	lected.			

v. Use Equi Join, to join employee and EmpDept tables.

SQL> SELEC	Γ * FROM	Employee, Em	pDept WHERE	<pre>Employee.Department =</pre>	EmpDept.DeptId
;					
EMPID	EMPNAME		DEPARTMENT	CONTACTNO	
EMAILID		EMPHEADID	DEPTID	DEPTNAME	
DEPTHEAD					
101	Isha		E-101	1234567890	
abc@gmail.o	COM	105	E-101	HR	
103	Neha		E-101	1234567890	
abc@gmail.o	COM	101	E-101	HR	

EMPID	EMPNAME		DEPARTMENT	CONTACTNO
EMAILID	E	MPHEADID	DEPTID	DEPTNAME
DEPTHEAD				
105 abc@gmail.c		102		1234567890 HR
104 abc@gmail.c	Rahul	105		1234567890 Development
	EMPNAME		DEPARTMENT	
	E	MPHEADID	DEPTID	
DEPTHEAD				
101				
	Priya	100		1234567890
abc@gmail.c	com	103	E-104	Sales

# 2. For the above given Relational Schema, Perform the following Nested Sub Query Operations:

i. Display the department details of a company which is assigned to the employee with employee id above 103.

# **CODE:**

SQL> SELECT * FRO	M EmpDept WHERE [	DeptId IN	(SELECT D	epartment	FROM	Employee	WHE
RE EmpId>103 );							
DEPTID	DEPTNAME		DEPTHEAD	)			
E-101	HR		105	;			
E-102	Development		101				

ii. Display the details of Employee who is working under 'Priya'.

```
SQL> SELECT * FROM Employee WHERE EmpHeadId IN (SELECT EmpId FROM Employee WHER E EmpName = 'Priya');
```

iii. Display the details of Employee who is the department head of HR. iv. Display the detail of employee who is working in 'development' department and they are permanent.

### CODE:

```
SQL> SELECT * FROM Employee WHERE EmpId IN (SELECT DeptHead FROM EmpDept WHERE

DeptName = 'HR');

EMPID EMPNAME DEPARTMENT CONTACTNO

EMAILID EMPHEADID

105 Abhishek E-101 1234567890

abc@gmail.com 102
```

iv. Display the detail of employee who is working in 'development' department and they are permanent.

#### CODE:

```
SQL> SELECT * FROM Employee WHERE EmpId IN (SELECT EmpId FROM EmpSalary WHERE I sPermanent = 'Yes') AND Department IN (SELECT DeptId FROM EmpDept WHERE DeptNam e = 'Development');

EMPID EMPNAME DEPARTMENT CONTACTNO

EMAILID EMPHEADID

104 Rahul E-102 1234567890

abc@gmail.com 105
```

v. Display the salary of the employee who is currently working in the HR Department and is a permanent employee.

```
SQL> SELECT Salary FROM EmpSalary WHERE IsPermanent = 'Yes' AND EmpId IN (SELEC T EmpId FROM Employee WHERE Department IN (SELECT DeptId FROM EmpDept WHERE DeptName = 'HR'));

SALARY
```

```
2000
2300
SQL>
```

vi. Display the maximum salary of an employee with its details from each department.

# **CODE:**

SQL> SELECT * FROM En	nployee INNER JO	IN EmpS	alary ON	Employee.Empl	Id = EmpSalary.
EmpId WHERE Employee.	EmpId IN (SELEC	T EmpId	FROM Emp	Salary WHERE	Salary IN (SEL
ECT MAX(Salary) FROM	Employee NATURA	L JOIN	EmpSalary	GROUP BY Dep	partment));
EMPID EMPNAME	DEPA 	RTMENT		CONTACTNO	
EMAILID	EMPHEADID	EMPID	SALAR	Y ISPERMANEN	Г
102 Priya	E-10	4		1234567890	
abc@gmail.com	103	102	1000	0 Yes	
104 Rahul	E-10	2		1234567890	
abc@gmail.com	105	104	190	0 Yes	
103 Neha	E-10	1		1234567890	
abc@gmail.com	101	103	500	0 No	

# 3. For the above given Relational Schema, Perform the following Set Operator Operations:

i. Display the department which is not yet being assigned to any employee up till now.

## **CODE:**

```
SQL> SELECT DeptId FROM EmpDept MINUS SELECT Department FROM Employee;

DEPTID
------
E-103
E-105
```

ii. Find Id of employee for salary less than 5000 and greater than 2300.

# **CODE:**

SQL> SELECT EmpId FROM EmpSalary WHERE Salary<5000 AND Salary>2300;

iii. Find Id of employee for salary less than 2000.

### **CODE:**

```
SQL> SELECT EmpId From EmpSalary WHERE SALARY<2000;

EMPID
-----
104
```

iv. Find Employee Names starting from A, P and N.

# **CODE:**

```
SQL> SELECT EmpName FROM Employee WHERE EmpName LIKE 'A%' OR EmpName LIKE 'P%'

OR EmpName LIKE 'N%';

EMPNAME

-----

Priya
Neha
Abhishek
```

v. Find employee details other than employee having salary less than 2000.

```
SQL> SELECT * FROM Employee WHERE EmpId IN (SELECT EmpId FROM Employee MINUS SE
LECT EmpId FROM EmpSalary WHERE Salary<2000);
    EMPID EMPNAME DEPARTMENT CONTACTNO
               EMPHEADID
EMAILID
    101 Isha
mail.com
                         E-101
                                          1234567890
abc@gmail.com
                     105
     102 Priya
ail.com 10
                          E-104
                                          1234567890
abc@gmail.com
                     103
     103 Neha
                          E-101
                                           1234567890
abc@gmail.com
                      101
```

EMPID EMPNAME		DEPARTMENT	CONTACTNO
EMAILID	EMPHEADID		
 105 Abhishek		E-101	1234567890
abc@gmail.com	102		

# **Conclusion:**

- 1. The various Join operations have been learnt and practically implemented on complex queries for the employee table.
- 2. It can be concluded that the join clause is used to combine rows from two or more tables, based on a related column.
- 3. The set comparison operators are used to perform nested queries in SQL.
- 4. Thus, the queries based on SQL join operators have been performed and the obtained results have been recorded.