Experiment No 16: Implement SQL queries for Inner and Outer Join

I. Practical Significance:

Join is used to combine the data spread across tables. A join is performed by the 'where' clause which combines the specified rows of tables. This practical allows students to join two or more tables.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To efficiently join data from multiple tables

III. COURSE LEVEL LEARNIN OUTCOMES (COS):

CO3 - Manage database using SQL.

IV. LABORATORY LEARNING OUTCOME:

Execute the queries based on Inner & Outer join

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.
b. Follow installation steps.
c. Follow ethical practices.

VI. Relevant Theoretical Background

Joins :

Serial No	Join Type	Explanation	Syntax	Example	Output
1	Equi-Join	A join based on equalities using the = operator. It retrieves rows with matching	SELECT columns FROM table1, table2 WHERE table1.column1 = table2.column2;	SELECT emp. empno, emp.ename, emp.deptno, dept.deptno, dept.loc FROM emp, dept	Rows with matching deptno values from both emp and dept tables.

		values in both tables.		WHERE emp.deptno = dept.deptno;	
2	Non Equi-Join	A join using relational operators other than = (e.g., <, >, <=, >=, !=).	SELECT columns FROM table1, table2 WHERE table1.column1 < table2.column2;	SELECT e. ename, e. salary, s. grade FROM emp e, salgrade s WHERE e.salary BETWEEN s.losal AND s.hisal;	Rows from emp and salgrade where emp.salary falls between salgrade.losal and salgrade.hisal.
3	Self Join	A join where a table is joined with itself. It compares rows within the same table.	SELECT a.column1, b.column2 FROM table a, table b WHERE a.column = b.column;	SELECT worker.ename "employee", manager.ename "manager" FROM emp worker, emp manager WHERE worker.mgr = manager.empno;	Rows where each worker's manager is also listed in the emp table.
4	Left Outer Join	Returns all records from the left table and matched records from the right table. Returns NULL for non-matching rows from the right table.	SELECT columns FROM table1 LEFT OUTER JOIN table2 ON table1.column = table2.column;	SELECT EMP.emp_id, EMP.name, DEPT.dept_name FROM EMP LEFT OUTER JOIN DEPT ON EMP.dept_id = DEPT.dept_id;	All records from EMP table and matching records from DEPT. Non-matching DEPT rows are NULL.
5	Right Outer	Returns all records from	SELECT columns FROM	SELECT EMP.emp_id,	All records from DEPT

	Join	the right table and matched records from the left table. Returns NULL for non-matching rows from the left table.	table1 RIGHT OUTER JOIN table2 ON table1.column = table2.column;	EMP.name, DEPT.dept_name FROM EMP RIGHT OUTER JOIN DEPT ON EMP.dept_id = DEPT.dept_id;	table and matching records from EMP. Non-matching EMP rows are NULL.

Note:

Guidelines for Equi-Join:

1. Rows in one table can be joined to rows in another table according to the common values existing in corresponding columns, that are usually primary and foreign key columns.
2. When writing a select statement that joins tables, precede the column name with the table name. (e.g. dept.deptno)
3. If the same column name appears in more than one table, the column name must be prefixed to the table name. (e.g. dept.deptno, emp.deptno)
4. If the column names are unique, then we need not prefix it with the table name.

Table aliases

Table aliases are used to make multiple tables queries shorter and more readable. As a result, we give an alias name or short name to the table in the 'from' clause. The alias can be used instead of the table name throughout the query.

OR

Following are the types of joins:

- Equi - join
- Non equi-join
- Self-Join
- Outer Join

Equi -join

A join, which is based on equalities, is called equi-join. In equi-join comparison operator equal to (=) is used to perform a join. It retrieves rows from tables having a common column. It is also called simple join.

Syntax: select table1.column, table1.column,table2.column,....,
from table1, table2
where table1.column1 = table2.column2;

Example:

```
Select emp.empno,emp. ename,emp.deptno,
    dept.deptno,dept.loc
from emp, dept
where  emp.deptno = dept. deptno;
```

Guidelines:

1. Rows in one table can be joined to rows in another table according to the common values existing in corresponding columns, that are usually primary and foreign key columns.
2. When writing a select statement that joins tables, precede the column name with the table name. (e.g. dept.deptno)
3. If the same column name appears in more than one table, the column name must be prefixed to the table name. (e.g. dept.deptno, emp.deptno)
4. If the column names are unique, then we need not prefix it with the table name.

```
select e. empno, e.ename, e.deptno,
d.deptno, d.loc
from emp e, dept d
where e.deptno = d.deptno;
```

Note: - The above example is same as example of equi join but uses table aliases where 'e' refers to emp table and 'd' refers to dept table.

Non equi-join

A join that specifies the relationship between columns belonging to different tables by making use of the relational operators (<, >, <=, >=, ! =) other than '=' operator is called as non equi-join.

To use non equi-join create the following table.

Table - Salgrade

Column-name	Datatype
Grade	Number (4)
Losal	Number (8)
Hisal	Number (8)

[**Note** - Students will insert the following records in a given table.]

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

To relate emp and salgrade tables using non-equi join.

Example:

```
select e.ename, e.salary, s.grade
from emp e, salgrade s
where e.salary
between s.losal and s.hisal;
```

OR

```
select e.ename, e.salary, s.grade
from emp e, salgrade s
   where e.salary >= s.losal and
         e.salary <= s.hisal;
```

Self-join

Joining a table to itself is known as self-join. i.e. it joins one row in a table to another. It can compare each row of the table to itself and with other rows of the same table.

Example: To find the name of each employee's manager you need to join EMP table to itself.

```
select worker.ename "employee", manager.ename "manager"
        from emp worker, emp manager
where worker.mgr = manager.empno;
```

Note:

The above example joins the emp table to itself. To stimulate two tables in the FROM clause, there are two aliases, namely WORKER and MANAGER, for the same table, EMP.

Outer Join

An outer join returns all the rows returned by simple join or equi join as well as those rows from one table that *do not match any row from the other table*.

Consider following tables for outer join:

EMP Table

emp_id	ename	dept_id
1	Riya	1
2	Pranav	2
3	Mansi	NULL

DEPT Table

dept_id	dept_name
1	HR
2	IT
3	Finance

1. LEFT OUTER JOIN

Description: The LEFT OUTER JOIN returns all records from the left table (EMP), and the matched records from the right table (DEPT). The *result is NULL* from the right side, *if there is no match.*

Syntax:

```
SELECT columns
FROM EMP
LEFT OUTER JOIN DEPT
ON EMP.dept_id = DEPT.dept_id.
```

Example:

```
SELECT EMP.emp_id, EMP.name, DEPT.dept_name
FROM EMP
LEFT OUTER JOIN DEPT
ON EMP.dept_id = DEPT.dept_id;
```

Output:

emp_id	name	dept_name
1	Riya	HR
2	Pranav	IT
3	Mansi	NULL

2. RIGHT OUTER JOIN

Description: The RIGHT OUTER JOIN returns all records from the right table (DEPT), and the matched records from the left table (EMP). The result is NULL from the left side, when there is no match.

Syntax:

```
SELECT columns
FROM EMP
RIGHT OUTER JOIN DEPT
ON EMP.dept_id = DEPT.dept_id;
```

Example:

```
SELECT EMP.emp_id, EMP.name, DEPT.dept_name
FROM EMP
RIGHT OUTER JOIN DEPT
ON EMP.dept_id = DEPT.dept_id;
```

Output:

emp_id	name	dept_name
1	Riya	HR
2	Pranav	IT
NULL	NULL	Finance

3. FULL OUTER JOIN

Description: The FULL OUTER JOIN returns all records when there is a match in either left (EMP) or right (DEPT) table records. It *returns NULL for records* that *do not have a match in the other table.*

Syntax:

```
SELECT columns
```

```
FROM EMP
FULL OUTER JOIN DEPT
ON EMP.dept_id = DEPT.dept_id;
```

Example:

```
SELECT EMP.emp_id, EMP.name, DEPT.dept_name
FROM EMP
FULL OUTER JOIN DEPT
ON EMP.dept_id = DEPT.dept_id;
```

Output:

emp_id	name	dept_name
1	Riya	HR
2	Pranav	IT
3	Mansi	NULL
NULL	NULL	Finance

VII. Required Resources/apparatus/equipment with specifications

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite,Oracle Live SQL etc.	All

VIII. Procedure

1. Create tables for given application
2. Assign Primary key for created table
3. Join two or more tables

* Practical related questions

1] →

A join is used to combine rows from two or more tables based on a related columns between based on a related column between them.

- Types of joins
① Inner join
② outer join

1] Inner join

Returns only the rows where there is a match in both tables.

Syntax:

SELECT column_name list from table 1 inner join table 2 ON table 1. column_name = table 2. column_name;

Ex:

SELECT product_id, Product_name, category_name From product inner join Categories on Product. Category_id = Categories. categorityId;

2] outer join

outer join is based on both match and unmatch data. outer

Outer join is further divided into following parts:

A] Left outer join.

B] Right outer join.

C] Full outer join.

A] Left outer Join:
The Left outer Join returns all rows from the Left table even if there are No match in the in the right table null Value are show at the place of right table value.

Syntax:
SELECT Columns From table 1 Left outer Join table 2 ON table1. Column = table2. Column;

ex:
SELECT ~~Emp~~. * from student 1 left outer Join student_data 2 on student 1 . Roll_no = student_data 2 Roll_no;

B] Right outer Join:
Returns all records from the right table and matched records from the left table Returns Null for non-matching rows for from the left table.

Syntax:
SELECT Columns from table 1 Right outer Join table 2 ON table1. column = table 2. Column

ex:
SELECT * from student 1 Right outer Join student data 2 ON student 1 . Roll_no = student data 2. Roll_no;

C] Full outer Join
The Full outer Join return a Result with the matching data of of both the table and remaining rows of both left table then the Right table Null Value are Show the place of table value.

Syntax:
SELECT * from table 1 Full outer Join s table 2
ON table 1. Roll_no = table 2. Roll_no;

Ex:
SELECT * from Student_data 1 Full outer Join Student_data 2 ON Student_data 1. Roll_no = Student_data 2. Roll_no;

2] ⟶
An outer Join is a type of SQL Join that returns all records from one or both of the tables involved, even if there is not a matching record in the other table. This contrasts with an inner Join, which only returns records where there is a match between both tables.

There are three main types of outer Joins.
1] Left outer Join
2] Right outer Join
3] Full outer Join

1] Left outer Join :
Returns all records from the left table and the matched records from the left table and the

matched records from the right table. if there is no match, the result is NULL on the side of the right table.

Syntax:
SELECT Columns from table1 left outer join table2 ON table1. column = table2. column;

ex:
SELECT * from student1 left outer join student 2 on student1. Roll_no = student2. Roll_no;

2] Right outer join:
Returns all records from the right table and the matched records from the left table. if there is no match, the result is NULL on the side of the left table.

Syntax:
SELECT columns from table1 Right outer join table2 on table1. Column = table2. Column

ex:
SELECT * from student 1 Right outer join Student 2 ON Student 1. Roll_no = Student2. Roll_no;

3] Full outer join:
Returns all records when there is a match in either left or right table records if there is no match, the result will have Nulls on the side without a match.

The Full outer join return a result with the matching data of both the table and remaining rows of both left table then the right table Null value are show the place of table value.

Syntax:
SELECT *from table1 full outer join table2
ON table1.Roll_no = table2.Roll_no;

Ex:
SELECT *from Student data 1 full outer join Student_data2 ON student_data1.Roll_no = Student_data2.Roll_no;

* Exercise

1] →

```
SELECT employee_number, name, department_number.
department_location from Employee e JOIN
Department d ON e.department_number = d.
department_number where e.name = 'Nikhil';
```

2] →

```
SELECT e.employee_number, e.name, e.department_
number from employee e Join Department d
ON e.department_number = d.department_number
where d.department_name = 'sales';
```

3] →

```
SELECT e.employee_number, e.department_number
from employee e Join Department d ON
e.department_number = d.department_number
where d.department_name != 'Sales';
```

4] →

```
SELECT e.name As employee_name, e.salary
from Employee e Join Employee m on
e.manager_id = m.employee_id where
m.name = 'Sumit Patil';
```

IX. Result(s)

In this practical we studied implement SQL queries for inner and outer Join

X. Practical related questions (Provide space for answers)

Note: Below are a few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO

1. Define Join. List types of Joins.
2. Describe outer join.

(Space for answer)

XI. Exercise

1. Display employee Nikhil's employee number, name, department number, and department location.