Kabeer Ahmed
SE-19028

Database Management System
(DBMS – 204)

Experiment # 04
Displaying Data from Multiple Tables

Student Name: Kabeer Ahmed

Roll Number: SE-19028

Maximum Marks	Performance = 05	Viva = 05	Total = 10
Marks Obtained			
Remarks (if any)			

Experiment evaluated by

Instructor Name: Engr. Adiba Jafar

Signature and Date:

Kabeer Ahmed
SE-19028

OUTCOMES

After completing this lesson, you should be able to do the following:

1. Write SELECT statements to access data from more than one table using equality and nonequality joins
2. View data that generally does not meet a join condition by using outer joins
3. Join a table to itself by using a self join

Obtaining Data from Multiple Tables(EMPLOYEES, DEPARTMENTS)

Sometimes you need to use data from more than one table. In the example, the report displays data from two separate tables.

- Employee IDs exist in the EMP table.
- Department IDs exist in both the EMP and DEPT tables.
- Location IDs exist in the DEPT table.

To produce the report, you need to link the EMPLOYEES and DEPARTMENTS tables and access data from both of them.

Cartesian Products

1. A Cartesian product is formed when:
   –A join condition is omitted
   –A join condition is invalid
   –All rows in the first table are joined to all rows in the second table
2. To avoid a Cartesian product, always include a valid join condition in a WHERE clause.

Generating a Cartesian Product

EMP (14 rows)
DEPT (4 rows)
Cartesian product: 14x4=56 rows

```
SELECT ename, dname deptno FROM emp, dept;
```

Types of Joins

Oracle Proprietary SQL: 1999
Compliant Joins: Joins (8i and prior):

1. Equijoin
2. Cross joins
3. Nonequijoin
4. Natural joins
5. Outer join

Using clause

6. Self join
7. Full or two sided outer joins
8. Arbitrary join conditions for outer joins

Joining Tables Using Oracle Syntax

Use a join to query data from more than one table.

```
SELECT table1.column, table2.column FROM table1, table2 WHERE
table1.column1 = table2.column2;
```

- Write the join condition in the WHERE clause.

Kabeer Ahmed
SE-19028

- Prefix the column name with the table name when the same column name appears in more than one table.

What Is an Equijoin?

EMP, DEPT Foreign key Primary key Equijoins

To determine an employee's department name, you compare the value in the DEPTNO column in the EMP table with the DEPTNO values in the DEPT table. The relationship between the EMPLOYEES and DEPARTMENTS tables is an *equijoin* , that is, values in the DEPTNO column on both tables must be equal. Frequently, this type of join involves primary and foreign key complements.

Note: Equijoins are also called *simple joins* or *inner joins*

.

Retrieving Records with Equijoins

```
SELECT e.empno, e.ename,e.deptno, d.deptno,d.loc
FROM   emp e, dept d
WHERE  e.deptno = d.deptno;
```

Additional Search Conditions Using the AND Operator

```
SELECT ename, e.deptno,d.dname
FROM   emp e, dept d
WHERE  e.deptno = d.deptno AND    ename = 'SMITH';
```

Qualifying Ambiguous Column Names

• Use table prefixes to qualify column names that are in multiple tables.

• Improve performance by using table prefixes.

• Distinguish columns that have identical names but reside in different tables by using column aliases.

Using Table Aliases

- Simplify queries by using table aliases
- Improve performance by using table prefixes

```
SELECT e.empno, e.ename, e.deptno,d.deptno, d.loc
FROM   emp e, dept d
WHERE  e.deptno = d.deptno;
```

Joining More than Two Tables

1. EMP
2. DEPT
3. SALGRADE

To join n tables together, you need a minimum of n -1 join conditions. For example, to join three tables, a minimum of two joins is required.

```
SELECT e.ename, d.dname, s.grade,s.losal
FROM   emp e, dept d, salgrade s
WHERE  e.deptno = d.deptno  AND    e.sal BETWEEN s.losal AND s.hisal;
```

Nonequijoins

Salary in the EMP table must be between lowest salary and highest salary in the SALGRADES table.

Kabeer Ahmed
SE-19028

Retrieving Records with Nonequijoins

```
SELECT e.ename, e.sal, s.grade
FROM   emp e, gra s
WHERE  e.sal BETWEEN s.losal AND s.hisal ;
```

Outer Joins Syntax

- You use an outer join to also see rows that do not meet the join condition.
- The outer join operator is the plus sign (+).

```
SELECT table1.column, table2.column
FROM table1, table2
WHERE table1.column(+) = table2.column;
SELECT table1.column, table2.column
FROM table1, table2
 WHERE table1.column = table2.column(+);
```

Using Outer Joins

```
SELECT e.ename, e.deptno, d.dname
FROM emp e, dept d
WHERE e.deptno(+) = d.deptno;
```

Self Joins

EMPLOYEES (WORKER) EMPLOYEES (MANAGER)
MANAGER_ID in the WORKER table is equal to
EMPLOYEE_ID in the MANAGER
table.

Joining a Table to Itself

```
SELECT worker.ename || ' works for '
|| manager.ename
FROM   emp worker, emp manager
WHERE  worker.mgr = manager.empno;
```

Use a join to query data from more than one table.

Creating Cross Joins

- **The CROSS JOIN clause produces the cross-product of two tables.**
- **This is the same as a Cartesian product between the two tables.**

```
SELECT table1.column, table2.column
FROM table1
[CROSS JOIN table2 ] | [NATURAL JOIN table2 ] | [JOIN table2
USING ( column_name )] | [JOIN table2 ON( table1.column_name
= table2.column_name )] | [LEFT|RIGHT|FULL OUTER JOIN
table2 ON ( table1.column_name = table2.column_name )];
SELECT ename, dname FROM   emp
CROSS JOIN dept;
```

Creating Cross Joins
The example in the slide gives the same results as the following:

```
SELECT ename, dname FROM emp, dept;
```

Creating Natural Joins

• The NATURAL JOIN clause is based on all columns in the two tables that have the same name.
• It selects rows from the two tables that have equal values in all matched columns.
• If the columns having the same names have different data types, then an error is returned.

Kabeer Ahmed
SE-19028

Note: The join can happen only on columns having the same names and data types in both the tables. If the columns have the same name, but different data types, then the NATURAL JOIN syntax causes an error.

Retrieving Records with Natural Joins

```
SELECT deptno, dname,loc,empno,ename
FROM   dept
NATURAL JOIN emp;

SELECT  deptno, dname,loc  FROM    dept  NATURAL JOIN emp
WHERE   deptno IN (20, 50);
```

Creating Joins with the USING Clause

- If several columns have the same names but the data types do not match, the NATURAL JOIN clause can be modified with the USING clause to specify the columns that should be used for an equijoin.

Note: Use the USING clause to match only one column when more than one column matches.

- Do not use a table name or alias in the referenced columns.
- The NATURAL JOIN and USING clauses are mutually exclusive.

For example, this statement is valid:

```
SELECT e.sal, d.dname,deptno
FROM   emp e  JOIN dept d USING (deptno) WHERE  deptno = 20;
```

This statement is invalid because the DEPTNO is qualified in the where clause:

```
SELECT e.sal, d.dname,deptno   FROM emp e JOIN dept d USING (deptno)
WHERE d.deptno = 20;
```

ORA-25154: column part of USING clause cannot have qualifier
The same restriction applies to NATURAL
joins also. Therefore columns that have the same name in both
tables have to be used without any qualifiers.

Retrieving Records with the USING Clause

```
SELECT e.empno, e.ename, d.loc
FROM  emp e JOIN dept d
USING (deptno);

SELECT empno, ename,
emp.deptno, loc FROM   emp, dept
WHERE  emp.deptno = dept.deptno;
```

Creating Joins with the ON Clause

- The join condition for the natural join is basically an equijoin of all columns with the same name.
- To specify arbitrary conditions or specify columns to join, the ON clause is used.
- Separates the join condition from other search.
- The ON clause makes code easy to understand.

```
SELECT e.empno, e.ename, e.deptno,d.deptno, d.loc FROM   emp e JOIN dept d
ON     (e.deptno = d.deptno);
```

Kabeer Ahmed
SE-19028

INNER versus OUTER Joins

- In SQL: 1999, the join of two tables returning only matched rows is an inner join.
- A join between two tables that returns the results of the inner join as well as unmatched rows left (or right) tables is a left (or right) outer join.
- A join between two tables that returns the results of an inner join as well as the results of a left and right join is a full outer join.

Joins: Comparing SQL: 1999 to Oracle Syntax

LEFT OUTER JOIN

SELECT e.ename, e.deptno, d.dname FROM emp e
LEFT OUTER JOIN dept d
ON (e.deptno = d.deptno);

Example of LEFT OUTER JOIN

This query retrieves all rows in the EMPLOYEES table, which is the left table even if there is no match in the DEPARTMENTS table.

This query was completed in earlier releases as follows:

SELECT e.ename, e.deptno, d.dname FROM emp e, dept d WHERE d.deptno(+) = e.deptno;

RIGHT OUTER JOIN

SELECT e.ename, e.deptno, d.dname
FROM emp e
RIGHT OUTER JOIN dept d
ON (e.deptno = d.deptno);

Example of RIGHT OUTER JOIN

This query retrieves all rows in the DEPARTMENTS table, which is the right table even if there is no match in the EMPLOYEES table.

This query was completed in earlier releases as follows:

SELECT e.ename, e.deptno, d.dname
FROM emp e, dept d
WHERE d.deptno = e.deptno (+);

FULL OUTER JOIN

SELECT e.ename, e.deptno, d.dname
FROM emp e
FULL OUTER JOIN dept d
ON (e.deptno = d.deptno);

Additional Conditions

SELECT e.empno, e.ename, e.deptno,d.deptno, d.loc
FROM emp e JOIN dept d ON (e.deptno = d.deptno) AND e.mgr = 7566;

Kabeer Ahmed
SE-19028

LAB # 04

SOLUTIONS

1. **Write a query to display the name, department number, and department name for all employees.**

```
SQL> select e.ename, e.deptno, d.dname
  2  FROM emp e, dept d
  3  WHERE e.deptno = d.deptno;

ENAME                     DEPTNO DNAME
-------------------- ---------- --------------------
SMITH                         20 RESEARCH
SMITH                         20 RESEARCH
ALLEN                         30 SALES
ALLEN                         30 SALES
WARD                          30 SALES
WARD                          30 SALES
JONES                         20 RESEARCH
JONES                         20 RESEARCH
MARTIN                        30 SALES
MARTIN                        30 SALES
BLAKE                         30 SALES

ENAME                     DEPTNO DNAME
-------------------- ---------- --------------------
BLAKE                         30 SALES
CLARK                         10 ACCOUNTING
CLARK                         10 ACCOUNTING
SCOTT                         20 RESEARCH
SCOTT                         20 RESEARCH
KING                          10 ACCOUNTING
KING                          10 ACCOUNTING
TURNER                        30 SALES
TURNER                        30 SALES
ADAMS                         20 RESEARCH
ADAMS                         20 RESEARCH

ENAME                     DEPTNO DNAME
-------------------- ---------- --------------------
JAMES                         30 SALES
JAMES                         30 SALES
FORD                          20 RESEARCH
FORD                          20 RESEARCH
MILLER                        10 ACCOUNTING
MILLER                        10 ACCOUNTING

28 rows selected.
```

Kabeer Ahmed
SE-19028

2. **Create a unique listing of all jobs that are in department 30. Include the location of department in the output.**

```
SQL> SELECT DISTINCT job, loc
  2  FROM emp, dept
  3  WHERE emp.deptno = dept.deptno
  4  AND emp.deptno = 30;

JOB                  LOC
-------------------- --------------------
SALESMAN             CHICAGO
MANAGER              CHICAGO
CLERK                CHICAGO
```

3. **Write a query to display the employee name, department name AND location of all employees who earn a commission.**

```
32 rows selected.

SQL> SELECT e.ename, d.dname, d.loc
  2  FROM emp e, dept d
  3  WHERE e.deptno = d.deptno
  4  AND e.comm IS NOT NULL;

ENAME                DNAME                LOC
-------------------- -------------------- --------------------
ALLEN                SALES                CHICAGO
WARD                 SALES                CHICAGO
MARTIN               SALES                CHICAGO
TURNER               SALES                CHICAGO
ALLEN                SALES                CHICAGO
WARD                 SALES                CHICAGO
MARTIN               SALES                CHICAGO
TURNER               SALES                CHICAGO

8 rows selected.
```

Kabeer Ahmed
SE-19028

4. **Display the employee name and department name for all employees who have an *a* (lowercase) in their last names. Place your SQL statement in a text file named lab4_4.sql.**

```
SQL> SELECT ename, dname
  2  FROM emp, dept
  3  WHERE emp.deptno = dept.deptno
  4  AND ename LIKE '%a%';

no rows selected

SQL> save e://lab4_4.sql;
Created file e://lab4_4.sql
SQL> get e://lab4_4.sql;
  1  SELECT ename, dname
  2  FROM emp, dept
  3  WHERE emp.deptno = dept.deptno
  4* AND ename LIKE '%a%'
SQL> @e://lab4_4.sql;

no rows selected
```

5. **Write a query to display the name, job, department number, and department name for all employees who work in Chicago.**

```
SQL> SELECT e.ename, e.job, e.deptno, d.dname
  2  FROM emp e JOIN dept d
  3  ON (e.deptno = d.deptno)
  4  WHERE LOWER(d.loc) = 'chicago';

ENAME                JOB                       DEPTNO DNAME
-------------------- ------------------ ---------- --------------
ALLEN                SALESMAN                    30 SALES
ALLEN                SALESMAN                    30 SALES
WARD                 SALESMAN                    30 SALES
WARD                 SALESMAN                    30 SALES
MARTIN               SALESMAN                    30 SALES
MARTIN               SALESMAN                    30 SALES
BLAKE                MANAGER                     30 SALES
BLAKE                MANAGER                     30 SALES
TURNER               SALESMAN                    30 SALES
TURNER               SALESMAN                    30 SALES
JAMES                CLERK                       30 SALES

ENAME                JOB                       DEPTNO DNAME
-------------------- ------------------ ---------- --------------
JAMES                CLERK                       30 SALES

12 rows selected.
```

Kabeer Ahmed
SE-19028

6. **Display the employee name and employee number who has manager. Label the columns Employee , Emp# , Manager , and Mgr# , respectively**

```
SQL> SELECT ename "Employee", empno "Emp#"
  2  FROM emp
  3  WHERE mgr IS NOT NULL;

Employee                       Emp#
------------------- ----------
SMITH                          7369
ALLEN                          7499
WARD                           7521
JONES                          7566
MARTIN                         7654
BLAKE                          7698
CLARK                          7782
SCOTT                          7788
TURNER                         7844
ADAMS                          7876
JAMES                          7900

Employee                       Emp#
------------------- ----------
FORD                           7902
MILLER                         7902

13 rows selected.
```

Kabeer Ahmed
SE-19028

7. **Modify question6 to display all employees including King, who has no manager. Order the results by the employee number**

```
SQL> edit e://lab4_6.sql;

SQL> get e://lab4_6.sql;
  1  SELECT ename "Employee", empno "Emp#"
  2  FROM emp
  3* ORDER BY empno
SQL> save e://lab4_7.sql;
Created file e://lab4_7.sql
SQL> get e://lab4_7.sql;
  1  SELECT ename "Employee", empno "Emp#"
  2  FROM emp
  3* ORDER BY empno
SQL> @e://lab4_7.sql;

Employee                       Emp#
------------------- ----------
SMITH                          7369
ALLEN                          7499
WARD                           7521
JONES                          7566
MARTIN                         7654
BLAKE                          7698
CLARK                          7782
SCOTT                          7788
KING                           7839
TURNER                         7844
ADAMS                          7876

Employee                       Emp#
------------------- ----------
JAMES                          7900
FORD                           7902
MILLER                         7902

14 rows selected.
```

Kabeer Ahmed
SE-19028

8. **Create a query that displays employee names, department numbers, and all the employees who work in the same department as a given employee. Give each column an appropriate label.**

```
SQL>  SELECT e.deptno department, e.ename employee, c.ename colleague
  2   FROM emp e JOIN emp c
  3   ON (e.deptno = c.deptno)
  4   WHERE e.empno <> c.empno
  5   ORDER BY e.deptno, e.ename, c.ename;

DEPARTMENT EMPLOYEE             COLLEAGUE
---------- -------------------- --------------------
        10 CLARK                KING
        10 CLARK                MILLER
        10 KING                 CLARK
        10 KING                 MILLER
        10 MILLER               CLARK
        10 MILLER               KING
        20 ADAMS                FORD
        20 ADAMS                JONES
        20 ADAMS                SCOTT
        20 ADAMS                SMITH
        20 FORD                 ADAMS

DEPARTMENT EMPLOYEE             COLLEAGUE
---------- -------------------- --------------------
        20 FORD                 JONES
        20 FORD                 SCOTT
        20 FORD                 SMITH
        20 JONES                ADAMS
        20 JONES                FORD
        20 JONES                SCOTT
        20 JONES                SMITH
        20 SCOTT                ADAMS
        20 SCOTT                FORD
        20 SCOTT                JONES
        20 SCOTT                SMITH

DEPARTMENT EMPLOYEE             COLLEAGUE
---------- -------------------- --------------------
        20 SMITH                ADAMS
        20 SMITH                FORD
        20 SMITH                JONES
        20 SMITH                SCOTT
        30 ALLEN                BLAKE
        30 ALLEN                JAMES
        30 ALLEN                MARTIN
        30 ALLEN                TURNER
```

Kabeer Ahmed
SE-19028

9. **Show the structure of the SALGRADE table. Create a query that displays the name, job, department name, salary, and grade for all employees.**

```
SQL> DESC salgrade
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 GRA                                       NOT NULL VARCHAR2(20)
 LOWEST_SAL                                         NUMBER
 HIGHEST_SAL                                        NUMBER
```

```
SQL> SELECT e.ename, e.job, d.dname, s.gra
  2  FROM emp e, dept d, salgrade s
  3  WHERE e.deptno = d.deptno
  4  AND e.sal BETWEEN s.lowest_sal AND s.highest_sal;

ENAME                JOB                  DNAME
-------------------- -------------------- --------------------
GRA
--------------------
CLARK                MANAGER              ACCOUNTING
A

MILLER               CLERK                ACCOUNTING
A

KING                 PRESIDENT            ACCOUNTING
B


ENAME                JOB                  DNAME
-------------------- -------------------- --------------------
GRA
--------------------
JONES                MANAGER              RESEARCH
A

ADAMS                CLERK                RESEARCH
A

SCOTT                ANALYST              RESEARCH
B
```

Kabeer Ahmed
SE-19028

10. **Create a query to display the name and hire date of any employee hired after employee Davies.**

```
SQL> SELECT e.ename, e.hiredate
  2  FROM emp e, emp davies
  3  WHERE davies.ename = 'Davies'
  4  AND davies.hiredate < e.hiredate;

no rows selected
```

11. **Display the names and hire dates for all employees who were hired before their managers, along with their manager's ID and hire dates. Label the columns Employee , Emp Hired , Manager , and Mgr Hired , respectively.**

```
EMPLOYEE             EMP_HIRED MANAGER              MGR_HIRED
-------------------- --------- -------------------- ---------
ALLEN                20-FEB-81 BLAKE                01-MAY-81
WARD                 22-FEB-81 BLAKE                01-MAY-81
JONES                02-APR-81 KING                 17-NOV-81
BLAKE                01-MAY-81 KING                 17-NOV-81
CLARK                09-JUN-81 KING                 17-NOV-81
SMITH                17-DEC-80 FORD                 03-DEC-81
SMITH                17-DEC-80 MILLER               23-JAN-82

7 rows selected.
```