DEPARTMENT OF COMPUTER SYSTEMS ENGINEERING
MEHRAN UNIVERSITY OF ENGINEERING & TECHNOLOGY, JAMSHORO
Database Management Systems (4th Semester) 18CS
Lab Experiment 7

Roll No: **Date of Conduct:**

Submission Date: **Grade Obtained:**

Problem Recognition (0.3)	Completeness & accuracy (0.4)	Timeliness (0.3)	Score (1.0)

Objective: To retrive data from multiple tables (Joins).

Tools: MySQL, Oracle

Introduction:

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

Let us look at a selection from the "Orders" table:

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

Then, look at a selection from the "Customers" table:

CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

Notice that the "CustomerID" column in the "Orders" table refers to the "CustomerID" in the "Customers" table. The relationship between the two tables above is the "CustomerID" column.

Then, we can create the following SQL statement (that contains an INNER JOIN), that selects records that have matching values in both tables:

Example

```
SELECT Orders.OrderID,Customers.CustomerName,Orders.OrderDate
FROM Orders
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;
```

and it will produce something like this:

OrderID	CustomerName	OrderDate
10308	Ana Trujillo Emparedados y helados	9/18/1996
10365	Antonio Moreno Taquería	11/27/1996
10383	Around the Horn	12/16/1996
10355	Around the Horn	11/15/1996
10278	Berglunds snabbköp	8/12/1996

Types of SQL Joins

Here are the different types of the JOINs in SQL:

i. **SQL Inner Join:** The INNER JOIN keyword selects records that have matching values in both tables.

Syntax:

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

ii. **SQL Left Join:** The LEFT JOIN keyword returns all records from the left table (table1), and the matched records from the right table (table2). The result is NULL from the right side if there is no match.

Syntax:

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

iii. **SQL Right join:** The RIGHT JOIN keyword returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match.

Syntax:

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

Note: In some databases RIGHT JOIN is called RIGHT OUTER JOIN.

iv. **SQL Full join:** The FULL OUTER JOIN keyword returns all records when there is a match in left (table1) or right (table2) table records.

Note: FULL OUTER JOIN can potentially return very large result-sets!

Syntax:

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```

v. **SQL Self join:** A self-JOIN is a regular join, but the table is joined with itself.

Syntax:

```
SELECT column_name(s)
FROM table1 T1, table1 T2
WHERE condition;
```

T1 and T2 are different table aliases for the same table.

Lab Task

1. List the name of the employees with the name of their immediate higher authority.

Task:

```
2     SELECT A.ENAME "EMPLOYEE", B.ENAME "REPORTS TO"
3           FROM EMP A, EMP B WHERE A.MGR=B.ID;
4
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data

(Read Only)

EMPLOYEE	REPORTS TO
MERKEL	BUSH
PUTIN	BUSH
TOOSK	PUTIN
CARNEGIE	PUTIN
THATCHER	TOOSK
FORD	CARNEGIE

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

Task:

```
1     SELECT DISTINCT e.job, d.loc
2     FROM emp e, dept d
3     WHERE e.DEPTNO = d.dept_no
4     AND e.DEPTNO = 30
```

1 Result | 2 Profiler | 3 Messages

(Read Only)

job	loc
SALESMAN	CHICAGO
CLERK	CHICAGO

3. Display the employee name and department name for all employees who have an A in their name.

Task:

```
1     SELECT e.`ENAME`, d.`dname`
2       FROM emp e, dept d
3         WHERE e.`DEPTNO` = d.dept_no
4     AND e.`ENAME` LIKE '%A%'
```

1 Result | 2 Profiler | 3 Messages | 4 Table D

(Read Only)

ENAME	dname
THATCHER	RESEARCH
CARNEGIE	RESEARCH
BAROSSO	SALES
WALTON	SALES
CHIRACK	SALES
GATES	SALES

4. Display the employee name and department name for all employees who work in DALLAS.

Task:

```
1      SELECT e.ename, e.job, e.deptno, d.dname
2      FROM emp e, dept d
3      WHERE e.deptno = d.dept_no
4      AND d.loc = 'DALLAS'
5
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data

(Read Only)

ename	job	deptno	dname
THATCHER	CLERK	20	RESEARCH
PUTIN	MANAGER	20	RESEARCH
CARNEGIE	ANALYST	20	RESEARCH
FORD	CLERK	20	RESEARCH
TOOSK	ANALYST	20	RESEARCH
JACK	TEACHER	20	RESEARCH

5. Display the employee name employee number along with their manager name and manager's number for all employees including KING, who has no manager. Label the columns employee, Emp#, Manager, and Mgr#, respectively.

Task:

```
1      SELECT w.ename "Employee", w.id "EMP#",
2           m.ename "Manager", m.id "Mgr#"
3        FROM emp w
4       LEFT OUTER JOIN emp m
5      ON (w.mgr = m.id);
```

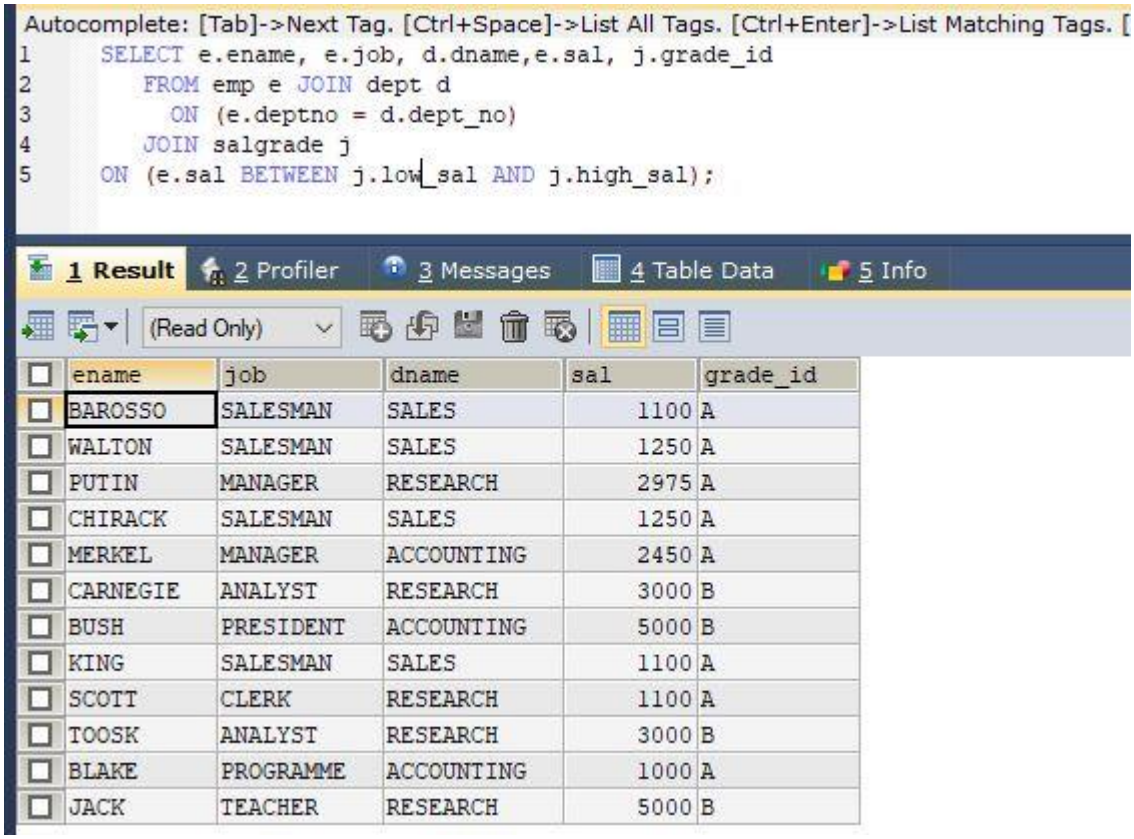
1 Result | 2 Profiler | 3 Messages | 4 Table

(Read Only)

Employee	EMP#	Manager	Mgr#
THATCHER	7369	TOOSK	7902
BAROSSO	7499	*(NULL)*	*(NULL)*
WALTON	7521	*(NULL)*	*(NULL)*
PUTIN	7566	BUSH	7839
CHIRACK	7654	*(NULL)*	*(NULL)*
MERKEL	7782	BUSH	7839
CARNEGIE	7788	PUTIN	7566
BUSH	7839	*(NULL)*	*(NULL)*
GATES	7844	*(NULL)*	*(NULL)*
FORD	7876	CARNEGIE	7788
BUFFETT	7900	*(NULL)*	*(NULL)*
TOOSK	7902	PUTIN	7566
BLAKE	7903	*(NULL)*	*(NULL)*
JACK	7904	*(NULL)*	*(NULL)*

6. Create a query that will display the name, job, department name, salary, grade for all employees.

Task:



```
Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List All Tags. [Ctrl+Enter]->List Matching Tags. [
1    SELECT e.ename, e.job, d.dname,e.sal, j.grade_id
2       FROM emp e JOIN dept d
3         ON (e.deptno = d.dept_no)
4       JOIN salgrade j
5    ON (e.sal BETWEEN j.low_sal AND j.high_sal);
```

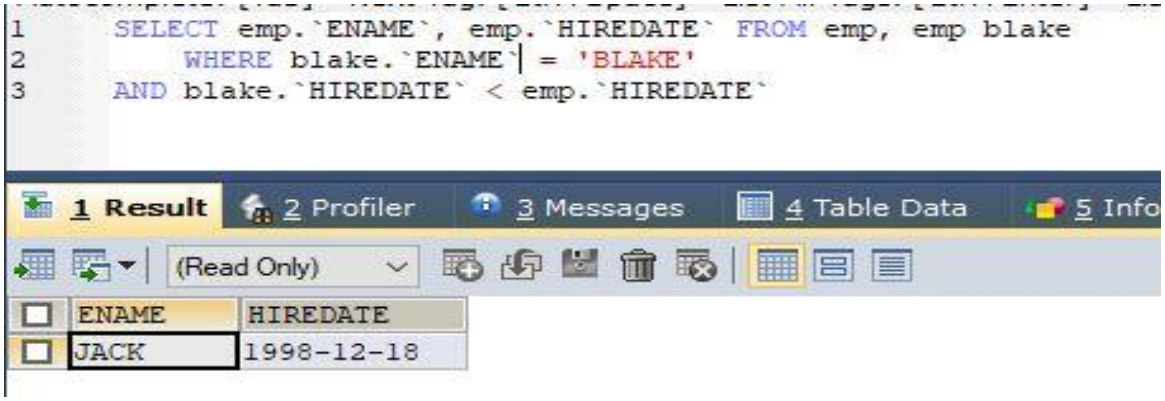
1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info

(Read Only)

ename	job	dname	sal	grade_id
BAROSSO	SALESMAN	SALES	1100	A
WALTON	SALESMAN	SALES	1250	A
PUTIN	MANAGER	RESEARCH	2975	A
CHIRACK	SALESMAN	SALES	1250	A
MERKEL	MANAGER	ACCOUNTING	2450	A
CARNEGIE	ANALYST	RESEARCH	3000	B
BUSH	PRESIDENT	ACCOUNTING	5000	B
KING	SALESMAN	SALES	1100	A
SCOTT	CLERK	RESEARCH	1100	A
TOOSK	ANALYST	RESEARCH	3000	B
BLAKE	PROGRAMME	ACCOUNTING	1000	A
JACK	TEACHER	RESEARCH	5000	B

7. Create a query to display the name and hire date of any employee hired after employee BLAKE.

Task:



```
1    SELECT emp.`ENAME`, emp.`HIREDATE` FROM emp, emp blake
2        WHERE blake.`ENAME` = 'BLAKE'
3    AND blake.`HIREDATE` < emp.`HIREDATE`
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info

(Read Only)

ENAME	HIREDATE
JACK	1998-12-18

8. Display all employees' names and hire dates along with their manager's name and hire date for all employees who were hired before their managers. Label the columns Employee, Emp Hiredate, Manager, and Mgr. Hiredate, respectively.

Task:

```
1   SELECT e.ename "Employee", e.hiredate "Emp Hiredate", m.ename "Manager", m.hiredate "Mgr Hiredate"
2           FROM emp e, emp m WHERE e.mgr = m.ID
3   AND e.hiredate < m.hiredate
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info

(Read Only)

Employee	Emp Hiredate	Manager	Mgr Hiredate
THATCHER	1980-12-17	TOOSK	1981-12-03
PUTIN	1981-04-02	BUSH	1981-11-17
MERKEL	1981-06-09	BUSH	1981-11-17