|  |  |
| --- | --- |
| **A picture containing logo  Description automatically generated** | **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)** |
|  |  |  |  |

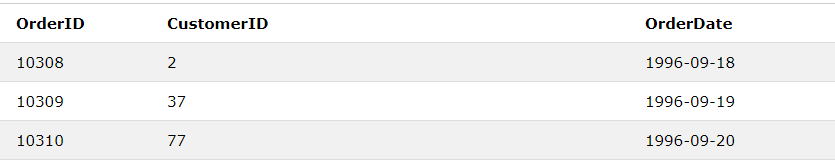
**Objective: To retrive data from multiple tables (Joins).**

**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:



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

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:



## Types of SQL Joins

Here are the different types of the JOINs in SQL:

1. **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 table2ON table1.column\_name *=* table2.column\_name*;*

1. **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 table1LEFT JOIN table2ON table1.column\_name *=* table2.column\_name;

1. **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 table2ON table1.column\_name = table2.column\_name;

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

1. **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 table2ON table1.column\_name *=* table2.column\_nameWHERE condition;

1. **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

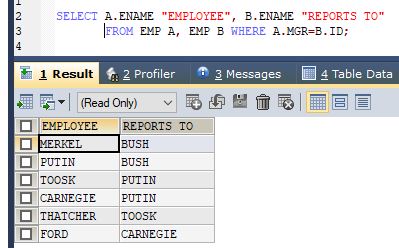
WHERE condition;

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.**

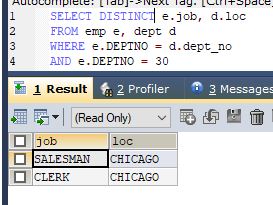
**Task:**

****

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

**department 30 in the output.**

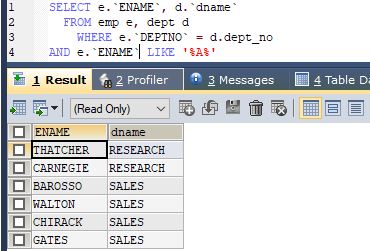
**Task:**

****

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

**their name.**

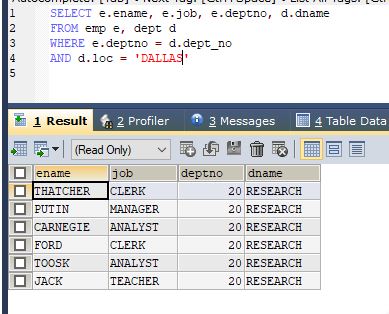
**Task:**

****

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

**DALLAS.**

**Task:**

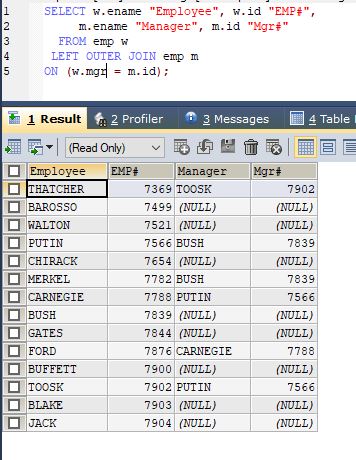
****

**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**

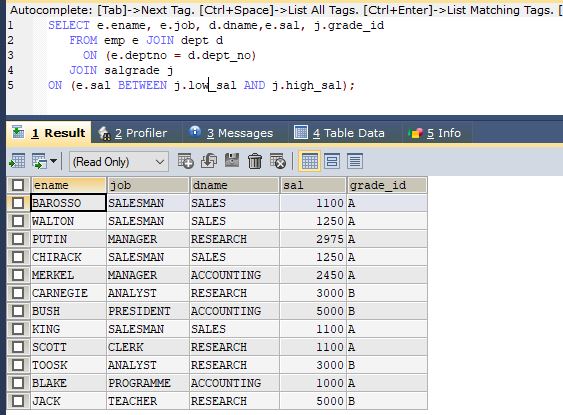
**columns employee, Emp#, Manager, and Mgr#, respectively.**

**Task:**

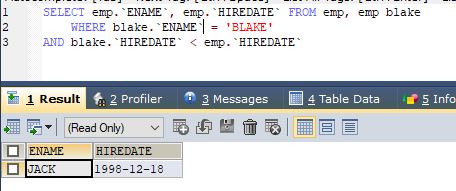
****

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

**employees.**

**Task:**

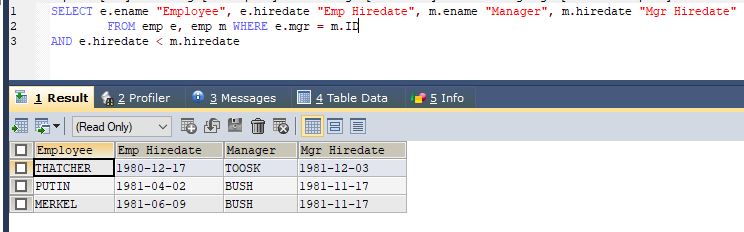
**7. Create a query to display the name and hire date of any employee hired after employee BLAKE.**

**Task:**

**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:**

****