

8.WORKING WITH MULTIPLE TABLES

1. Write a query to display the last name, department number, and department name for all Employees.

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e
JOIN departments d
ON e.department_id = d.department_id;
```

| LAST_NAME | DEPARTMENT_ID | DEPARTMENT_NAME |
|-----------|---------------|-----------------|
| King | 10 | Administration |
| Davies | 80 | Sales |
| Smith | 80 | Sales |

2. Create a unique listing of all jobs that are in department 80. Include the location of the department in the output.

```
SELECT DISTINCT e.job_id, l.city
FROM employees e
JOIN departments d
ON e.department_id = d.department_id
JOIN locations l
ON d.location_id = l.location_id
WHERE e.department_id = 80;
```

| JOB_ID | CITY |
|--------|----------|
| SA_REP | New York |
| SA_MAN | New York |

3. Write a query to display the employee last name, department name, location ID, and city of all employees who earn a commission

```
SELECT e.last_name, d.department_name, d.location_id, l.city
FROM employees e
JOIN departments d
```

```

ON e.department_id = d.department_id
JOIN locations l
ON d.location_id = l.location_id
WHERE e.commission_pct IS NOT NULL;

```

| LAST_NAME | DEPARTMENT_NAME | LOCATION_ID | CITY |
|-----------|-----------------|-------------|----------|
| Davies | Sales | 2 | New York |
| Smith | Sales | 2 | New York |

2. Display the employee last name and department name for all employees who have an a(lowercase) in their last names. P

```

SELECT e.last_name, d.department_name
FROM employees e
JOIN departments d
ON e.department_id = d.department_id
WHERE LOWER(e.last_name) LIKE '%a%';

```

| LAST_NAME | DEPARTMENT_NAME |
|-----------|-----------------|
| Davies | Sales |

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

```

SELECT e.last_name, e.job_id, e.department_id, d.department_name
FROM employees e
JOIN departments d
ON e.department_id = d.department_id
JOIN locations l
ON d.location_id = l.location_id
WHERE l.city = 'Toronto';

```

| LAST_NAME | JOB_ID | DEPARTMENT_ID | DEPARTMENT_NAME |
|-----------|---------|---------------|-----------------|
| King | AD_PRES | 10 | Administration |

6. Display the employee last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#.

```

Respectively SELECT e.last_name AS Employee, e.employee_id AS Emp#,      m.last_name
AS Manager, m.employee_id AS Mgr#
FROM employees e
LEFT JOIN employees m
ON e.manager_id = m.employee_id;

```

| EMPLOYEE | EMP# | MANAGER | MGR# |
|----------|------|---------|------|
| Johnson | 4 | King | 1 |
| Davies | 2 | King | 1 |
| Williams | 5 | Davies | 2 |
| Smith | 3 | Davies | 2 |
| King | 1 | - | - |

7. Modify lab4_6.sql to display all employees including King, who has no manager. Order the results by the employee number.

```

SELECT e.last_name, e.employee_id, m.last_name AS Manager
FROM employees e
LEFT JOIN employees m
ON e.manager_id = m.employee_id
ORDER BY e.employee_id;

```

| LAST_NAME | EMPLOYEE_ID | MANAGER |
|-----------|-------------|---------|
| King | 1 | - |
| Davies | 2 | King |
| Smith | 3 | Davies |
| Johnson | 4 | King |
| Williams | 5 | Davies |

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

```

SELECT e.last_name AS Employee, e.department_id AS Dept#
FROM employees e
JOIN employees emp

```

ON e.department_id = emp.department_id
 WHERE emp.employee_id = 2; -- Replace with a specific employee ID

| EMPLOYEE | DEPT# |
|----------|-------|
| Davies | 80 |
| Smith | 80 |

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

DESCRIBE job_grades;

```
SELECT e.last_name, e.job_id, d.department_name, e.salary, jg.grade_level
FROM employees e
JOIN departments d
ON e.department_id = d.department_id
JOIN job_grades jg
ON e.salary BETWEEN jg.lowest_sal AND jg.highest_sal;
```

Object Type **TABLE** Object **JOB_GRADES**

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | M |
|-------------------|--------------------|-----------|--------|-----------|-------|-------------|---|
| <u>JOB_GRADES</u> | <u>GRADE_LEVEL</u> | VARCHAR2 | 5 | - | - | 1 | |
| | <u>LOWEST_SAL</u> | NUMBER | 22 | - | - | - | |
| | <u>HIGHEST_SAL</u> | NUMBER | 22 | - | - | - | |

no data found

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

```
SELECT e.last_name, e.hire_date
FROM employees e
WHERE e.hire_date > (SELECT hire_date FROM employees WHERE last_name = 'Davies');
```

| LAST_NAME | HIRE_DATE |
|-----------|------------|
| Smith | 04/23/2006 |
| Williams | 12/01/2007 |

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

```
SELECT e.last_name AS Employee, e.hire_date AS "Emp Hired",  
m.last_name AS Manager, m.hire_date AS "Mgr Hired"  
FROM employees e  
JOIN employees m  
ON e.manager_id = m.employee_id  
WHERE e.hire_date < m.hire_date;
```

no data found