Exp:8 **WORKING WITH MULTIPLE TABLES**

Date:

---

1. **Write a query to display the last name, department number, and department name for all employees.**
   ```sql
   SELECT e.last_name, e.department_id, d.department_name
   FROM employees e
   JOIN departments d ON e.department_id = d.department_id;
   ```

2. **Create a unique listing of all jobs that are in department 80. Include the location of the department in the output.**
   ```sql
   SELECT DISTINCT e.job_id, d.location_id
   FROM employees e
   JOIN departments d ON e.department_id = d.department_id
   WHERE e.department_id = 80;
   ```

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

4. **Display the employee last name and department name for all employees who have a lowercase "a" in their last names.**
   ```sql
   SELECT e.last_name, d.department_name
   FROM employees e
   JOIN departments d ON e.department_id = d.department_id
   WHERE e.last_name LIKE '%a%';
   ```

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

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.**
```sql
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;
```

7. **Modify the query to display all employees including King, who has no manager. Order the results by the employee number.**
```sql
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
ORDER BY e.employee_id;
```

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.**
```sql
SELECT e1.last_name AS Employee, e1.department_id AS Dept#, e2.last_name AS Colleague
FROM employees e1
JOIN employees e2 ON e1.department_id = e2.department_id
WHERE e1.employee_id != e2.employee_id;
```

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.**

```sql
-- To show structure:
DESC JOB_GRADES;

-- Query to display employee info with job grades:
SELECT e.last_name AS Name, e.job_id AS Job, d.department_name, e.salary, j.grade_level
FROM employees e
JOIN departments d ON e.department_id = d.department_id
JOIN job_grades j ON e.salary BETWEEN j.lowest_sal AND j.highest_sal;
```

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

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

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.**

```sql
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;
```
