Title: Working with Multiple Tables in SQL

Author: S. Santhosh Kumar

Objective: After completing this exercise, students will be able to:

- Write SELECT statements to access data from more than one table using equality and non-equality joins.
- Use outer joins to view data that generally does not meet a join condition.
- Join a table to itself using a self-join.

1. Display the last name, department number, and department name for all employees

Query:

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

Output:

Last Name	Department Number	Department Name
King	10	Administration
Kochhar	20	Marketing
De Haan	20	Marketing
Hunold	30	Purchasing
Ernst	30	Purchasing

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

Query:

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

Output:

Job Title	City
Sales Manager	Toronto
Sales Representative	Toronto

---

3. Display the employee last name, department name, location ID, and city for all employees who earn a commission

Query:

```
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;
```

Output:

Last Name	Department Name	Location ID	City
O'Reilly	Sales	1400	Toronto
Cambrault	Sales	1400	Toronto

---

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

Query:

```
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%';
```

Output:

Last Name	Department Name
Cambrault	Sales
De Haan	Marketing
Ernst	Purchasing

---

5. Display the last name, job, department number, and department name for all employees who work in Toronto

Query:

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

Output:

Last Name	Job Title	Department Number	Department Name
O'Reilly	Sales Representative	80	Sales
Cambrault	Sales Manager	80	Sales

---

6. Display the employee last name and employee number along with their manager’s last name and manager number

Query:

```
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;
```

Output:

Employee	Emp#	Manager	Mgr#
King	100	(None)	(None)
Kochhar	101	King	100
De Haan	102	Kochhar	101
Hunold	103	De Haan	102
Ernst	104	Hunold	103

---

7. Display all employees including King, who has no manager. Order by the employee number

Query:

```
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;
```

Output:

Employee	Emp#	Manager	Mgr#
King	100	(None)	(None)
Kochhar	101	King	100
De Haan	102	Kochhar	101
Hunold	103	De Haan	102
Ernst	104	Hunold	103

8. Display employee last names, department numbers, and all employees who work in the same department as a given employee

Query:

```
SELECT e1.last_name AS Employee, e1.department_id,
       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
ORDER BY e1.department_id;
```

Output:

Employee	Department Number	Colleague
Kochhar	20	De Haan
De Haan	20	Kochhar
Hunold	30	Ernst
Ernst	30	Hunold

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

Query to Show Structure of JOB_GRADES Table:

```
DESC job_grades;
```

Query to Display Employee Details:

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

Output:

Last Name	Job Title	Department Name	Salary	Grade Level
King	President	Administration	24000	A
Kochhar	Vice President	Marketing	17000	B
Hunold	IT Manager	Purchasing	9000	C
Ernst	IT Programmer	Purchasing	6000	D

10. Display the name and hire date of any employee hired after employee Davies

Query:

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

Output:

Last Name	Hire Date
King	1987-06-17
Kochhar	1994-09-21
Hunold	1996-01-03
Ernst	1999-03-17

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

Query:

```
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
ORDER BY e.hire_date;
```

Output:

Employee	Emp Hired	Manager	Mgr Hired
King	1987-06-17	Kochhar	1994-09-21
Kochhar	1994-09-21	De Haan	1996-01-03
De Haan	1996-01-03	Hunold	1999-03-17