

This query retrieves all rows in the EMPLOYEES table, even if there is no match in the DEPARTMENTS table. It also retrieves all rows in the DEPARTMENTS table, even if there is no match in the EMPLOYEES table.

Find the Solution for the following:

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;
```

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, 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

```
SELECT e.last-name, d.department-name, d.location-id, l.city
FROM 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
```

4. Display the employee last name and department name for all employees who have an 'a' 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 e.last-name
```

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, e.department_name FROM
employees e JOIN departments d ON e.department-id = d.department-id
JOIN location 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

```
SELECT e.last-name AS "Employee", e.employee-id AS "Emp#",
m.last-name AS "Manager", m.employee-id AS "Mgr#" FROM employee e
JOIN employee m ON e.manager-id = m.employee-id;
```



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 AS "Employee", e.employee-id AS "Emp#",
NVL(m.last-name, 'No Manager') AS "Manager", NVL(CHAR(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

```
SELECT e.last-name AS "Employee", e.department-id AS "Dept#", d.department-name AS "Dept Name" FROM employee e JOIN
departments d ON e.department-id = d.department-id WHERE
e.department-id = (SELECT department-id from employees WHERE
last-name = 'Taylor');
```

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

```
SELECT e.last-name, e.job-id, d.department-
DESC job-grades; name, e.salary, j.grade-level
FROM employee e JOIN departments d ON
e.department-id = d.department-id JOIN job-grade-
```

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');
```

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
```

✓

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	