Here is the neatly formatted version of all the SQL queries with proper spacing and structure:

Question 71

Query:

Write a SQL query to list employee names and department names for employees with a salary greater than 2000 using `INNER JOIN`.

SQL Query:

```sql
USE test;
SELECT ename, dname
FROM emp e
INNER JOIN dept d
ON e.deptno = d.deptno
WHERE sal > 2000;
```

Question 72

Query:

Write a SQL query to retrieve all employees and their department locations, including those without departments, using `LEFT JOIN`.

SQL Query:

```sql
```

```
USE test;

SELECT ename, location

FROM emp e

LEFT JOIN dept d

ON e.deptno = d.deptno;
```

Question 73

Query:

Write a SQL query to list all department numbers, department names, and their employee counts, including departments with no employees, using `RIGHT JOIN`.

SQL Query:

```sql
USE test;

SELECT d.deptno, d.dname, COUNT(empno) AS emp_count

FROM emp e

RIGHT JOIN dept d

ON e.deptno = d.deptno

GROUP BY d.deptno;
```

Question 74

Query:

Write a SQL query to simulate a `FULL OUTER JOIN` to list all employees and departments, including unmatched rows.

SQL Query:

```sql
USE test;
SELECT e.empno, e.ename, e.deptno, d.dname
FROM emp e
LEFT JOIN dept d
ON e.deptno = d.deptno
UNION
SELECT e.empno, e.ename, d.deptno, d.dname
FROM emp e
RIGHT JOIN dept d
ON e.deptno = d.deptno;
```

Question 75

Query:

Write a SQL query to find employees who are managers of other employees using a self-join.

SQL Query:

```sql
USE test;
SELECT DISTINCT e1.ename AS manager
FROM emp e1, emp e2
```

WHERE e1.empno = e2.mgr;

```

---

### **Question 76**

**Query:**

Write a SQL query to generate all possible employee-department combinations using `CROSS JOIN`.

**SQL Query:**

```sql

USE test;

SELECT ename, dname

FROM emp

CROSS JOIN dept;

```

Question 77

Query:

Write a SQL query to list departments with employees earning more than 2500 using `EXISTS`.

SQL Query:

```sql

USE test;

SELECT DISTINCT d.deptno, d.dname

```
FROM dept d, emp e

WHERE d.deptno = e.deptno

AND e.sal > 2500;
```

Question 78

Query:

Write a SQL query to find departments with the number of employees earning less than 1000 using `NOT EXISTS`.

SQL Query:

```sql
USE test;

SELECT d.dname, d.deptno

FROM dept d

WHERE NOT EXISTS (

   SELECT 1

   FROM emp e

   WHERE e.deptno = d.deptno

   AND e.sal < 1000

);
```

Question 79

Query:

Write a SQL query to find managers and the number of employees they manage in departments located in 'New York', using the primary key and foreign key constraints.

SQL Query:

```sql
USE test;
SELECT e1.ename AS manager_name, COUNT(e2.empno) AS emp_count
FROM emp e1
JOIN emp e2 ON e1.empno = e2.mgr
JOIN dept d ON e1.deptno = d.deptno
WHERE d.location = 'New York'
GROUP BY e1.ename;
```

Question 80

Query:

Write a SQL query to list all employees and departments, including those without matches, using a simulated `FULL JOIN`.

SQL Query:

```sql
USE test;
SELECT e.empno, e.ename, d.deptno, d.dname, d.location
FROM emp e
LEFT JOIN dept d ON e.deptno = d.deptno
UNION
```

```
SELECT e.empno, e.ename, d.deptno, d.dname, d.location

FROM emp e

RIGHT JOIN dept d ON e.deptno = d.deptno;
```

Question 81

Query:

Write a SQL query to list employee names and department names where the department is in 'Chicago' using `INNER JOIN`.

SQL Query:

```sql
USE test;

SELECT e.ename, d.dname

FROM emp e

INNER JOIN dept d

ON e.deptno = d.deptno

WHERE d.location = 'Chicago';
```

Question 82

Query:

Retrieve department-wise total salary and number of employees using `GROUP BY` and `JOIN`.

SQL Query:

```sql
USE test;
SELECT d.dname, COUNT(e.empno) AS num_employees, SUM(e.sal) AS total_salary
FROM dept d
LEFT JOIN emp e
ON d.deptno = e.deptno
GROUP BY d.dname;
```

Question 83

Query:

Write a SQL query to list departments with no assigned employees using `RIGHT JOIN`.

SQL Query:

```sql
USE test;
SELECT d.deptno, d.dname
FROM emp e
RIGHT JOIN dept d
ON e.deptno = d.deptno
WHERE e.empno IS NULL;
```

Question 84

Query:

Write a SQL query to combine employee and department data with duplicates using `UNION ALL`.

SQL Query:

```sql
USE test;
SELECT ename, dname
FROM emp
LEFT JOIN dept ON emp.deptno = dept.deptno
UNION ALL
SELECT ename, dname
FROM emp
RIGHT JOIN dept ON emp.deptno = dept.deptno;
```

Question 85

Query:

Write a SQL query to list employees and their managers' names using a `LEFT JOIN` for employees without managers.

SQL Query:

```sql
USE test;
SELECT e.ename AS employee, m.ename AS manager
FROM emp e
```

LEFT JOIN emp m

ON e.mgr = m.empno;

```

---

### **Question 86**

**Query:**

Write a SQL query to retrieve average salaries per department using `INNER JOIN` and `GROUP BY`.

**SQL Query:**

```sql

USE test;

SELECT d.deptno,

Here is the neatly formatted version of **Questions 87 to 90** with proper spacing and structure:

---

### **Question 87**

**Query:**

Write a SQL query to find departments with more than 3 employees using `INNER JOIN` and `HAVING`.

**SQL Query:**

```sql
```

```
USE test;

SELECT d.deptno, d.dname, COUNT(e.empno) AS emp_count

FROM dept d

JOIN emp e

ON d.deptno = e.deptno

GROUP BY d.deptno, d.dname

HAVING COUNT(e.empno) > 3;
```

Question 88

Query:

Write a SQL query to list employees and departments where the employee’s hire date is after 1980 using `INNER JOIN`.

SQL Query:

```sql
USE test;

SELECT e.ename, d.dname, e.hiredate

FROM emp e

JOIN dept d

ON e.deptno = d.deptno

WHERE e.hiredate > '1980-01-01';
```

Question 89

Query:

Find departments without employees using `LEFT JOIN` and `NULL` check.

SQL Query:

```sql
USE test;
SELECT d.dname AS Department, d.location
FROM dept d
LEFT JOIN emp e
ON d.deptno = e.deptno
WHERE e.empno IS NULL;
```

Question 90

Query:

Write a SQL query to list employee names and department names using an implicit join, ordered by employee name.

SQL Query:

```sql
USE test;
SELECT e.ename, d.dname
FROM emp e, dept d
WHERE e.deptno = d.deptno
ORDER BY e.ename;
```

```

---
```