71.

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

+-------+------------+

| ename | dname |

+-------+------------+

*/

USE test;

select ename,dname from emp e inner join dept d where e.deptno=d.deptno and sal>2000;

72.

/* Write a SQL query to retrieve all employees and their department locations,

including those without departments, using LEFT JOIN.

+--------+----------+

| ename | location |

+--------+----------+

*/

USE test;

select ename,location from emp e left join dept d on e.deptno=d.deptno;

73. /* Write a SQL query to list all department numbers, department names and their

employee counts, including departments with no employees, using RIGHT JOIN.

+--------+------------+-----------+

| deptno | dname | emp_count |

+--------+------------+-----------+

*/

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 deptno;

74

/* Write a SQL query to simulate a FULL OUTER JOIN to list all employees and

departments, including unmatched rows.

+-------+--------+--------+------------+

| empno | ename | deptno | dname |

+-------+--------+--------+------------+

*/

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;

75.

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

+---------+

| manager |

+---------+

*/

USE test;

select distinct e1.ename as manager from emp e1 ,emp e2 where e1.empno=e2.mgr;

76.

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

+--------+------------+

| ename | dname |

+--------+------------+

*/

USE test;

select ename,dname from emp cross join dept ;

77.

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

+--------+------------+

| deptno | dname |

+--------+------------+

*/

USE test;

select distinct d.deptno,d.dname from dept d ,emp e where d.deptno=e.deptno and e.sal>2500;

78.

/* Write a SQL query to find departments with number of employees earning less

than 1000 using NOT EXISTS.

+------------+--------+

| dname | deptno |

+------------+--------+

*/

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);

79.

/* 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.

+--------------+-----------+

| manager_name | emp_count |

+--------------+-----------+

*/

USE test;

select e1.ename manager_name,count(e2.empno) 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;

80.

/* Write a SQL query to list all employees and departments, including those

without matches, using a simulated FULL JOIN.

+-------+--------+--------+------------+----------+

| empno | ename | deptno | dname | location |

+-------+--------+--------+------------+----------+

*/

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;

81.

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

```
+-------+-------+
| ename | dname |
+-------+-------+
*/
USE test;

SELECT e.ename, d.dname
FROM emp e
INNER JOIN dept d ON e.deptno = d.deptno
WHERE d.location = 'Chicago';
```

82.

/*

Retrieve Department-wise Total Salary and Number of Employees (Using GROUP BY and JOIN)

```
+------------+---------------+--------------+
| dname      | num_employees | total_salary |
+------------+---------------+--------------+
*/
USE test;
```

select d.dname,count(e.empno) num_employees,sum(e.sal) total_salary from dept d left join emp e on d.deptno=e.deptno group by d.dname;

83.

/* Write a SQL query to list departments with no assigned employees using

RIGHT JOIN.

+--------+---------+

| deptno | dname |

+--------+---------+

*/

USE test;

select d.deptno,d.dname from emp e right join dept d on e.deptno=d.deptno where e.empno is null;

84.

/* Write a SQL query to combine employee and department data with duplicates

using UNION ALL.

+--------+------------+

| ename | dname |

+--------+------------+

*/

```
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;
```

85.

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

+----------+---------+
| employee | manager |
+----------+---------+

*/
USE test;
SELECT e.ename AS employee, m.ename AS manager
FROM emp e
LEFT JOIN emp m ON e.mgr = m.empno;
```

86.

```
/* Write a SQL query to retrieve average salaries per department using INNER
```

JOIN and GROUP BY.

+--------+------------+-------------+

| deptno | dname | avg_salary |

+--------+------------+-------------+

*/

USE test;

SELECT d.deptno, d.dname, AVG(e.sal) AS avg_salary

FROM dept d

JOIN emp e ON d.deptno = e.deptno

GROUP BY d.deptno, d.dname;

87.

/* Write a SQL query to find departments with more than 3 employees using

INNER JOIN and HAVING.

+--------+----------+-----------+

| deptno | dname | emp_count |

+--------+----------+-----------+

*/

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;
```

88.

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

+--------+------------+------------+
| ename  | dname      | hiredate   |
+--------+------------+------------+
*/
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';
```

89.

```
/*
Find Departments Without Employees (Using LEFT JOIN and NULL Check)

+------------+----------+
| Department | Location |
```

```
+------------+----------+
```

```
*/
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;
```

90.

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

+--------+------------+
| ename  | dname      |
+--------+------------+

*/
USE test;

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