1. List the employee numbers, names, department numbers and the department name.

2. Display the list of employees working in each department. Display the department information even if no employee belongs to that department.

3. List the employee names along with their manager’s name.

4. List all employees who joined the company before their managers.

5. Display the different designations in department no 20 and 30.

6. List the jobs common to department 20 and 30.

7. List the jobs unique to department 20.

8. List the employees belonging to the department of MILLER.

9. List the job with highest average salary.

10. List names of the employee who earns lowest salary in each job

11. List names of the employee who earns lowest salary in each department

12. List employee details who earn salary greater than the average salary for their department.

13. List employee details whose salary is greater than average salary of all the employees joined before 1st April 1981.

```sql
-- 1. List the employee numbers, names, department numbers, and the department name.
SELECT e.empno, e.ename, e.deptno, d.dname
FROM emp e
JOIN dept d ON e.deptno = d.deptno;

-- Using INNER JOIN keyword:
SELECT e.empno, e.ename, e.deptno, d.dname
FROM emp e
INNER JOIN dept d ON e.deptno = d.deptno;

-- Using table aliases for brevity:
SELECT e.empno, e.ename, e.deptno, d.dname
FROM emp e
JOIN dept d ON e.deptno = d.deptno;

-- Using the USING clause:
SELECT empno, ename, e.deptno, dname
FROM emp
JOIN dept USING (deptno);

-- Using the NATURAL JOIN keyword (which automatically joins columns with the same name):
SELECT empno, ename, dept.deptno, dname
FROM emp
NATURAL JOIN dept;

-- Using the WHERE clause instead of JOIN:
SELECT e.empno, e.ename, e.deptno, d.dname
FROM emp e, dept d
WHERE e.deptno = d.deptno;
```

-- 2. Display the list of employees working in each department. Display the department information even if no employee belongs to that department.

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

FROM dept d

LEFT JOIN emp e ON d.deptno = e.deptno

ORDER BY d.dname;
```

using the "(+)" syntax for the left join:

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

FROM dept d, emp e

WHERE d.deptno = e.deptno(+)

ORDER BY d.dname;
```

-- 3. List the employee names along with their manager’s name.

```
SELECT e.ename AS employee_name, m.ename AS manager_name

FROM emp e

LEFT JOIN emp m ON e.mgr = m.empno;
```

uses implicit INNER JOIN through the comma [,] notation. This means that only employees with a corresponding manager will be listed.

```
select m.ename Manager, e.ename Employee from emp m, emp e where
e.mgr=m.empno;
```

-- 4. List all employees who joined the company before their managers.

```
SELECT e.ename AS employee_name, e.hiredate AS employee_hire_date, m.ename AS manager_name, m.hiredate AS manager_hire_date

FROM emp e

JOIN emp m ON e.mgr = m.empno

WHERE e.hiredate < m.hiredate;
```

Using where clause

```
SELECT e.ename AS employee_name, e.hiredate AS employee_hire_date, m.ename AS manager_name, m.hiredate AS manager_hire_date

FROM emp e, emp m

where e.mgr = m.empno AND e.hiredate < m.hiredate;
```

```
-- 5. Display the different designations in department no 20 and 30.

SELECT DISTINCT job

FROM emp

WHERE deptno IN (20, 30);
```

```
-- 6. List the jobs common to department 20 and 30.

SELECT job

FROM emp

WHERE deptno = 20

INTERSECT

SELECT job

FROM emp

WHERE deptno = 30;
```

```
-- 7. List the jobs unique to department 20.
```

```sql
SELECT DISTINCT job
FROM emp
WHERE deptno = 20
AND job NOT IN (SELECT job FROM emp WHERE deptno != 20);

-- 8. List the employees belonging to the department of MILLER.
SELECT empno, ename
FROM emp
WHERE deptno = (SELECT deptno FROM emp WHERE ename = 'MILLER');

-- 9. List the job with the highest average salary.
SELECT job
FROM emp
GROUP BY job
HAVING AVG(sal) = (
   SELECT MAX(avg_sal)
   FROM (SELECT AVG(sal) AS avg_sal FROM emp GROUP BY job)
);



-- 10. List names of the employee who earns the lowest salary in each job.
SELECT job, ename
FROM emp
WHERE (job, sal) IN (SELECT job, MIN(sal) FROM emp GROUP BY job);

-- 11. List names of the employee who earns the lowest salary in each department.
SELECT deptno, ename
FROM emp
WHERE (deptno, sal) IN (SELECT deptno, MIN(sal) FROM emp GROUP BY deptno);
```

```
-- 12. List employee details who earn a salary greater than the average salary for their department.

SELECT *

FROM emp e

WHERE e.sal > (

   SELECT AVG(sal)

   FROM emp

   WHERE deptno = e.deptno

);
```

```
-- 13. List employee details whose salary is greater than average salary of all the employees joined before 1st April 1981.

SELECT *

FROM emp

WHERE sal > (SELECT AVG(sal) FROM emp WHERE hiredate < TO_DATE('1981-04-01', 'YYYY-MM-DD'));
```