4. Find the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number

```
1  SELECT
2      ROUND(MAX(salary)) AS Maximum,
3      ROUND(MIN(salary)) AS Minimum,
4      ROUND(SUM(salary)) AS Sum,
5      ROUND(AVG(salary)) AS Average
6  FROM employees;
7
```

Results   Explain   Describe   Saved SQL   History

| MAXIMUM | MINIMUM | SUM | AVERAGE |
|---|---|---|---|
| 60000 | 4000 | 224450 | 14963 |

5. Modify the above query to display the minimum, maximum, sum, and average salary for each job type.

```
1  SELECT
2      job_id,
3      ROUND(MIN(salary)) AS Minimum,
4      ROUND(MAX(salary)) AS Maximum,
5      ROUND(SUM(salary)) AS Sum,
6      ROUND(AVG(salary)) AS Average
7  FROM employees
8  GROUP BY job_id;
9
```

Results   Explain   Describe   Saved SQL   History

| JOB_ID | MINIMUM | MAXIMUM | SUM | AVERAGE |
|---|---|---|---|---|
| HR_REP | 6000 | 9000 | 15000 | 7500 |
| AD_VP | 17000 | 17000 | 17000 | 17000 |
| ST_CLERK | 10000 | 10000 | 10000 | 10000 |
| AD_ASST | 4000 | 4250 | 12450 | 4150 |
| SA_MAN | 12000 | 12000 | 24000 | 12000 |
| IT_PROG | 7500 | 60000 | 67500 | 33750 |
| SA_REP | 7500 | 55000 | 78500 | 19625 |

6.Write a query to display the number of people with the same job. Generalize the query so that the user in the HR department is prompted for a job title.

```
1  SELECT
2      job_id,
3      COUNT(*) AS Number_of_People
4  FROM employees
5  WHERE UPPER(job_id) = UPPER(:job_title)
6  GROUP BY job_id;
7
```

Results   Explain   Describe   Saved SQL   History

| JOB_ID | NUMBER_OF_PEOPLE |
|---|---|
| IT_PROG | 2 |

7. Determine the number of managers without listing them. Label the column Number of Managers. Hint: Use the MANAGER_ID column to determine the number of managers.

```
1    SELECT
2        COUNT(DISTINCT manager_id) AS Number_of_Managers
3    FROM employees
4    WHERE manager_id IS NOT NULL;
5
```

Results    Explain    Describe    Saved SQL    History

| NUMBER_OF_MANAGERS |
|---|
| 6 |

8. Find the difference between the highest and lowest salaries. Label the column DIFFERENCE.

```
1    SELECT
2        (MAX(salary) - MIN(salary)) AS Difference
3    FROM employees;
4
```

Results    Explain    Describe    Saved SQL    History

| DIFFERENCE |
|---|
| 56000 |

9. Create a report to display the manager number and the salary of the lowest-paid employee for that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is $6,000 or less. Sort the output in descending order of salary.

```
1    SELECT
2        manager_id,
3        MIN(salary) AS Lowest_Salary
4    FROM employees
5    WHERE manager_id IS NOT NULL
6    GROUP BY manager_id
7    HAVING MIN(salary) > 6000
8    ORDER BY Lowest_Salary DESC;
9
```

Results    Explain    Describe    Saved SQL    History

| MANAGER_ID | LOWEST_SALARY |
|---|---|
| 101 | 17000 |
| 105 | 12000 |
| 1000 | 10000 |
| 112 | 8000 |

10. Create a query to display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate column headings.

```
1   SELECT
2       COUNT(*) AS Total_Employees,
3       SUM(CASE WHEN EXTRACT(YEAR FROM hire_date) = 1995 THEN 1 ELSE 0 END) AS Hired_1995,
4       SUM(CASE WHEN EXTRACT(YEAR FROM hire_date) = 1996 THEN 1 ELSE 0 END) AS Hired_1996,
5       SUM(CASE WHEN EXTRACT(YEAR FROM hire_date) = 1997 THEN 1 ELSE 0 END) AS Hired_1997,
6       SUM(CASE WHEN EXTRACT(YEAR FROM hire_date) = 1998 THEN 1 ELSE 0 END) AS Hired_1998
7   FROM employees;
8
```

Results   Explain   Describe   Saved SQL   History

| TOTAL_EMPLOYEES | HIRED_1995 | HIRED_1996 | HIRED_1997 | HIRED_1998 |
|---|---|---|---|---|
| 15 | 0 | 0 | 0 | 0 |

11. Create a matrix query to display the job, the salary for that job based on department number, and the total salary for that job, for departments 20, 50, 80, and 90, giving each column an appropriate heading.

```
1   SELECT
2       job_id,
3       SUM(CASE WHEN department_id = 20 THEN salary ELSE 0 END) AS Dept20,
4       SUM(CASE WHEN department_id = 50 THEN salary ELSE 0 END) AS Dept50,
5       SUM(CASE WHEN department_id = 80 THEN salary ELSE 0 END) AS Dept80,
6       SUM(CASE WHEN department_id = 90 THEN salary ELSE 0 END) AS Dept90,
7       SUM(salary) AS Total_Salary
8   FROM employees
9   WHERE department_id IN (20, 50, 80, 90)
10  GROUP BY job_id;
11
```

Results   Explain   Describe   Saved SQL   History

| JOB_ID | DEPT20 | DEPT50 | DEPT80 | DEPT90 | TOTAL_SALARY |
|---|---|---|---|---|---|
| HR_REP | 15000 | 0 | 0 | 0 | 15000 |
| ST_CLERK | 0 | 10000 | 0 | 0 | 10000 |
| SA_MAN | 0 | 0 | 24000 | 0 | 24000 |
| IT_PROG | 7500 | 0 | 0 | 0 | 7500 |
| SA_REP | 0 | 0 | 23500 | 0 | 23500 |

12.Write a query to display each department's name, location, number of employees, and the average salary for all the employees in that department. Label the column name-Location, Number of people, and salary respectively. Round the average salary to two decimal places. i need sql code for these questions

```
1   SELECT
2       d.department_name || ' - ' || l.city AS "Department-Location",
3       COUNT(e.employee_id) AS "Number of People",
4       ROUND(AVG(e.salary), 2) AS "Average Salary"
5   FROM employees e
6   JOIN departments d ON e.department_id = d.department_id
7   JOIN locations l ON d.location_id = l.location_id
8   GROUP BY d.department_name, l.city;
9
```

Results   Explain   Describe   Saved SQL   History

| Department-Location | Number of People | Average Salary |
|---|---|---|
| Sales - Toronto | 5 | 9500 |