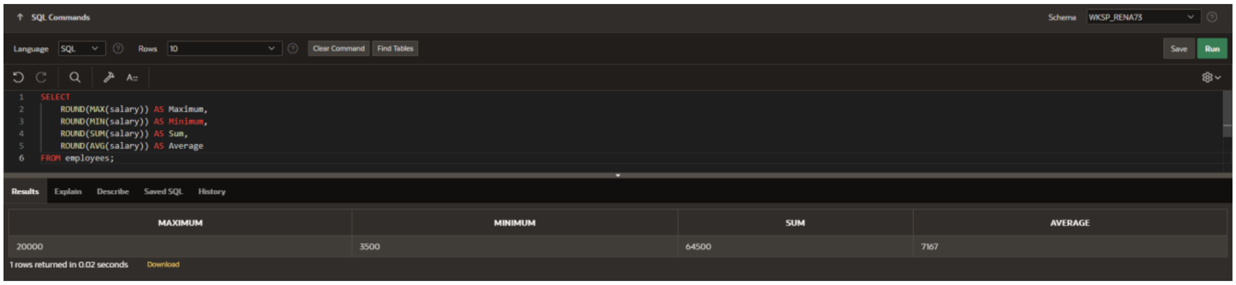
EXERCISE-8

Aggregating Data Using Group Functions

NAME	PRATHESHA J
ROLL NO	241001172
DEPARTMENT	IT

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

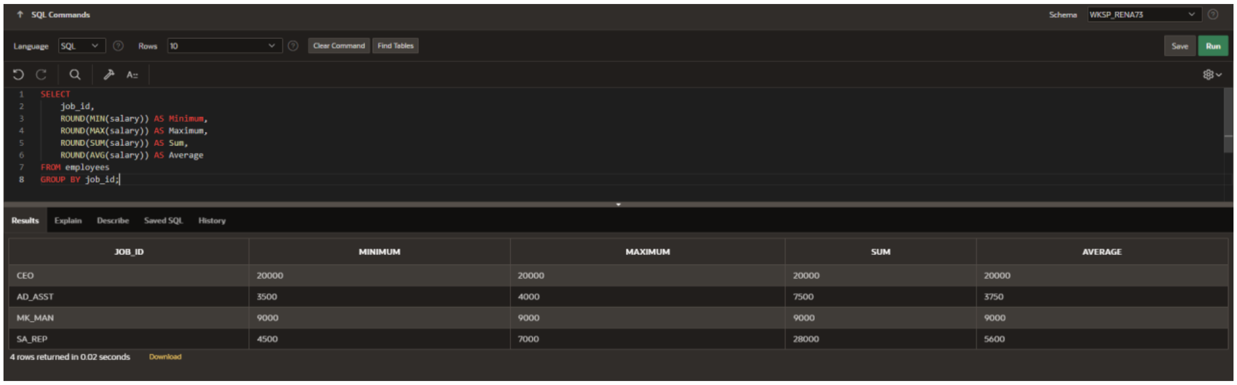


```sql
SELECT
    ROUND(MAX(salary)) AS Maximum,
    ROUND(MIN(salary)) AS Minimum,
    ROUND(SUM(salary)) AS Sum,
    ROUND(AVG(salary)) AS Average
FROM employees;
```

MAXIMUM	MINIMUM	SUM	AVERAGE
20000	3500	64500	7167

1 rows returned in 0.02 seconds

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

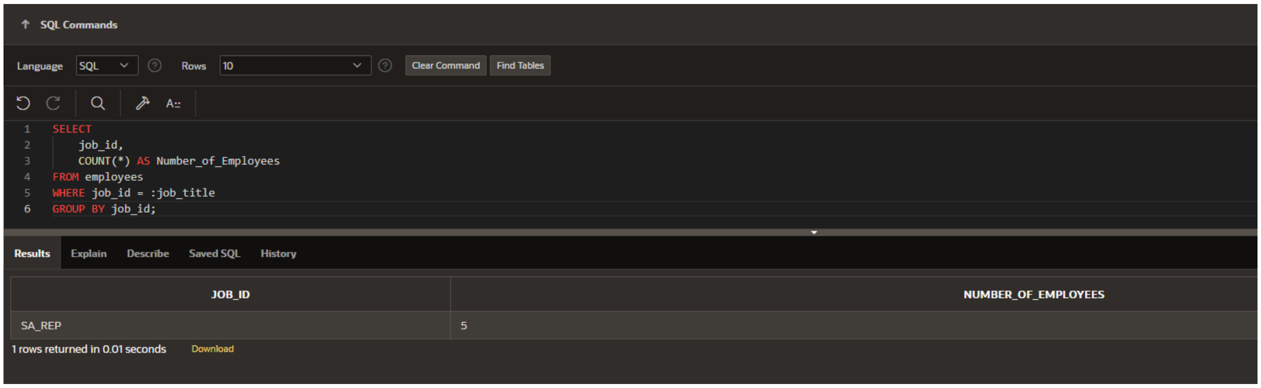


```sql
SELECT
    job_id,
    ROUND(MIN(salary)) AS Minimum,
    ROUND(MAX(salary)) AS Maximum,
    ROUND(SUM(salary)) AS Sum,
    ROUND(AVG(salary)) AS Average
FROM employees
GROUP BY job_id;
```

JOB_ID	MINIMUM	MAXIMUM	SUM	AVERAGE
CEO	20000	20000	20000	20000
AD_ASST	3500	4000	7500	3750
MK_MAN	9000	9000	9000	9000
SA_REP	4500	7000	28000	5600

4 rows returned in 0.02 seconds

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.

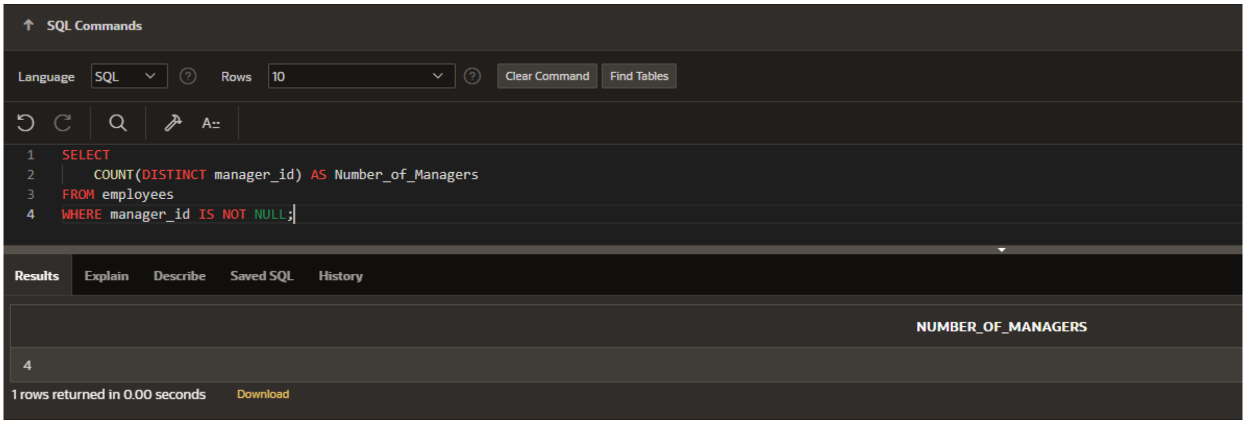


```sql
SELECT
    job_id,
    COUNT(*) AS Number_of_Employees
FROM employees
WHERE job_id = :job_title
GROUP BY job_id;
```

JOB_ID	NUMBER_OF_EMPLOYEES
SA_REP	5

1 rows returned in 0.01 seconds

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.

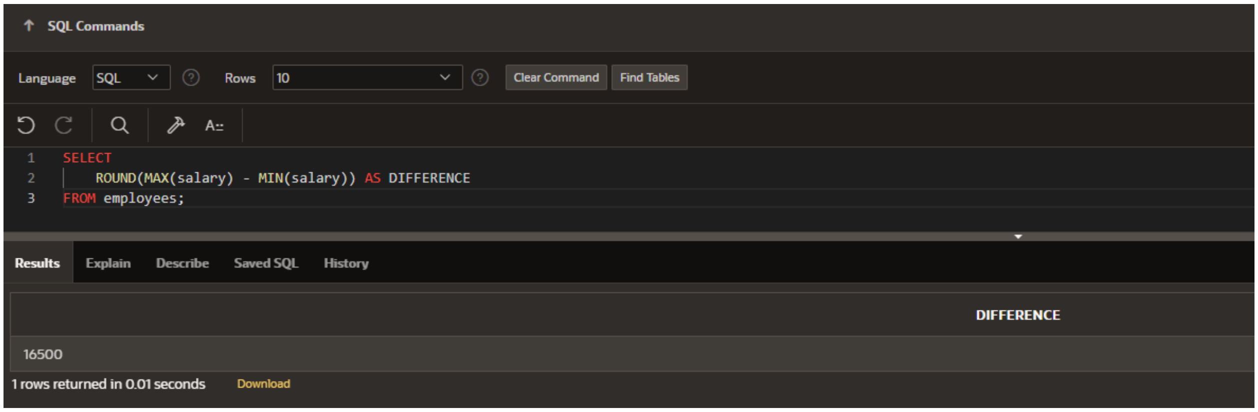


```sql
SELECT
    COUNT(DISTINCT manager_id) AS Number_of_Managers
FROM employees
WHERE manager_id IS NOT NULL;
```

NUMBER_OF_MANAGERS
4

1 rows returned in 0.00 seconds

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

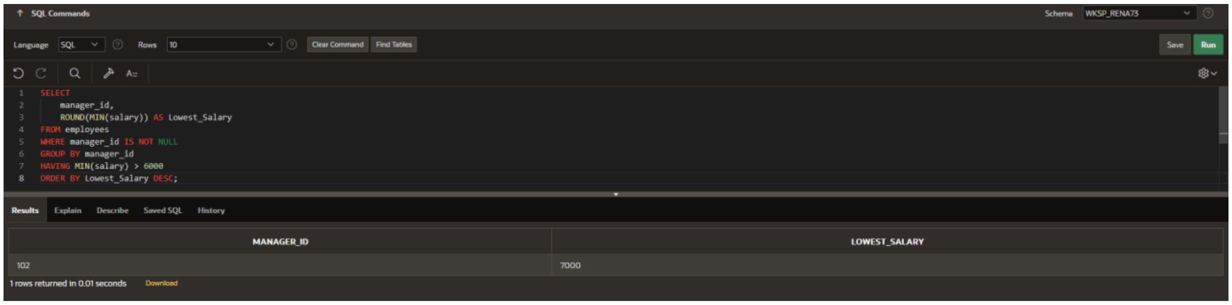


```sql
SELECT
    ROUND(MAX(salary) - MIN(salary)) AS DIFFERENCE
FROM employees;
```

DIFFERENCE
16500

1 rows returned in 0.01 seconds

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.

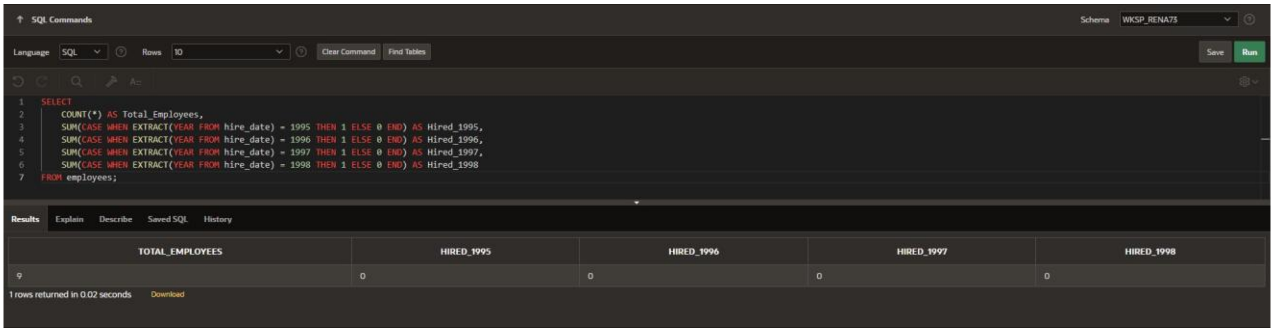


```sql
SELECT
    manager_id,
    ROUND(MIN(salary)) AS Lowest_Salary
FROM employees
WHERE manager_id IS NOT NULL
GROUP BY manager_id
HAVING MIN(salary) > 6000
ORDER BY Lowest_Salary DESC;
```

MANAGER_ID	LOWEST_SALARY
102	7000

1 rows returned in 0.01 seconds

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.

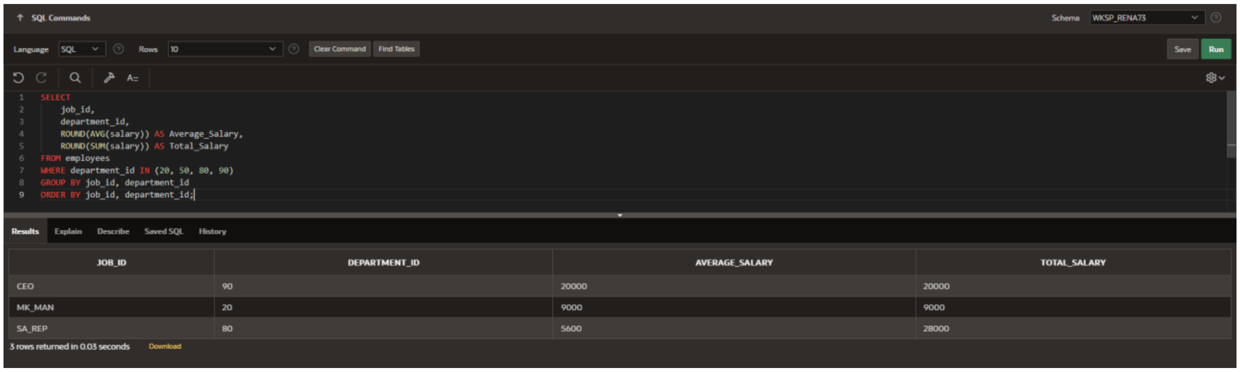


```
SELECT
    COUNT(*) AS Total_Employees,
    SUM(CASE WHEN EXTRACT(YEAR FROM hire_date) = 1995 THEN 1 ELSE 0 END) AS Hired_1995,
    SUM(CASE WHEN EXTRACT(YEAR FROM hire_date) = 1996 THEN 1 ELSE 0 END) AS Hired_1996,
    SUM(CASE WHEN EXTRACT(YEAR FROM hire_date) = 1997 THEN 1 ELSE 0 END) AS Hired_1997,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

TOTAL_EMPLOYEES	HIRED_1995	HIRED_1996	HIRED_1997	HIRED_1998
9	0	0	0	0

1 rows returned in 0.02 seconds

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.

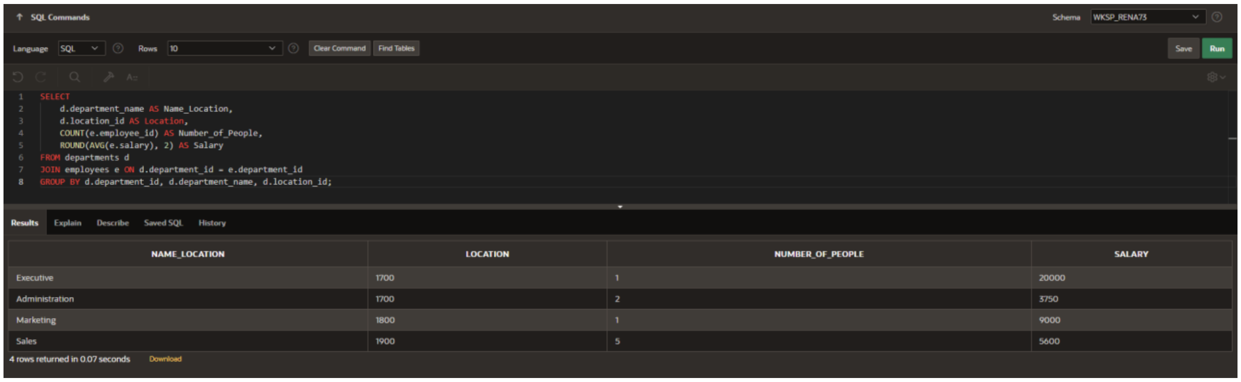


```
SELECT
    job_id,
    department_id,
    ROUND(AVG(salary)) AS Average_Salary,
    ROUND(SUM(salary)) AS Total_Salary
FROM employees
WHERE department_id IN (20, 50, 80, 90)
GROUP BY job_id, department_id
ORDER BY job_id, department_id;
```

JOB_ID	DEPARTMENT_ID	AVERAGE_SALARY	TOTAL_SALARY
CEO	90	20000	20000
MK_MAN	20	9000	9000
SA_REP	80	5600	28000

3 rows returned in 0.03 seconds

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.



```sql
SELECT
    d.department_name AS Name_Location,
    d.location_id AS Location,
    COUNT(e.employee_id) AS Number_of_People,
    ROUND(AVG(e.salary), 2) AS Salary
FROM departments d
JOIN employees e ON d.department_id = e.department_id
GROUP BY d.department_id, d.department_name, d.location_id;
```

NAME_LOCATION	LOCATION	NUMBER_OF_PEOPLE	SALARY
Executive	1700	1	20000
Administration	1700	2	3750
Marketing	1800	1	9000
Sales	1900	5	5600

4 rows returned in 0.07 seconds