EXERCISE NUMBER:10

1. 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

```
mysql> select round(max(salary),0)"Maximum", round(min(salary),0)"Minimum", sum(salary)"sum", round(avg(salary),0)"average" from empl;
+---------+---------+--------+---------+
| Maximum | Minimum | sum    | average |
+---------+---------+--------+---------+
|  130000 |   12000 | 362000 |   90500 |
+---------+---------+--------+---------+
1 row in set (0.00 sec)
```

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

```
mysql> select round(max(salary),0)"Maximum", round(min(salary),0)"Minimum", sum(salary)"sum", round(avg(salary),0)"average" from empl GROUP BY jobty
pe;
+---------+---------+--------+---------+
| Maximum | Minimum | sum    | average |
+---------+---------+--------+---------+
|  120000 |  100000 | 220000 |  110000 |
|  130000 |  130000 | 130000 |  130000 |
|   12000 |   12000 |  12000 |   12000 |
+---------+---------+--------+---------+
3 rows in set (0.00 sec)
```

3. 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

```
mysql> select job_id, count(*) AS total_employee from empl GROUP BY job_id ORDER BY  total_employee DESC;
+--------+----------------+
| job_id | total_employee |
+--------+----------------+
|   1001 |              2 |
|   1002 |              1 |
|   1003 |              1 |
+--------+----------------+
3 rows in set (0.00 sec)
```

4. 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.

```
mysql> SELECT manager_id, count(*) AS number_of_managers FROM empl GROUP BY manager_id;
+------------+--------------------+
| manager_id | number_of_managers |
+------------+--------------------+
|         10 |                  3 |
|         20 |                  1 |
+------------+--------------------+
2 rows in set (0.00 sec)
```

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

```
mysql> SELECT MAX(salary)-MIN(salary) DIFFERENCE FROM empl;
+------------+
| DIFFERENCE |
+------------+
|     118000 |
+------------+
1 row in set (0.00 sec)
```

6.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.

```
mysql> SELECT manager_id, MIN(salary) AS min_salary FROM empl WHERE manager_id IS NOT NULL GROUP BY manager_id HAVING MIN(salary) > 6000 ORDER BY mi
n_salary DESC;
+------------+------------+
| manager_id | min_salary |
+------------+------------+
|         10 |     100000 |
|         20 |      12000 |
+------------+------------+
2 rows in set (0.01 sec)
```

7. 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.

```
mysql> SELECT
    -> COUNT(*) AS total_employee,
    ->     COUNT(IF(entry_year = 1995, 1, NULL)) AS hired_in_1995,
    ->     COUNT(IF(entry_year = 1996, 1, NULL)) AS hired_in_1996,
    ->     COUNT(IF(entry_year = 1997, 1, NULL)) AS hired_in_1997,
    ->     COUNT(IF(entry_year = 1998, 1, NULL)) AS hired_in_1998
    -> FROM
    -> empl;
+----------------+---------------+---------------+---------------+---------------+
| total_employee | hired_in_1995 | hired_in_1996 | hired_in_1997 | hired_in_1998 |
+----------------+---------------+---------------+---------------+---------------+
|              4 |             1 |             1 |             1 |             1 |
+----------------+---------------+---------------+---------------+---------------+
1 row in set (0.00 sec)
```

8. 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.

```
mysql> SELECT
    ->     jobtype AS Job,
    ->     SUM(CASE WHEN department_id = 20 THEN salary ELSE 0 END) AS salary_dept_20,
    ->     SUM(CASE WHEN department_id = 50 THEN salary ELSE 0 END) AS salary_dept_50,
    ->     SUM(CASE WHEN department_id = 80 THEN salary ELSE 0 END) AS salary_dept_80,
    ->     SUM(CASE WHEN department_id = 90 THEN salary ELSE 0 END) AS salary_dept_90,
    ->     SUM(salary) AS total_salary
    -> FROM
    -> empl
    -> WHERE department_id IN(20, 50,80,90)
    -> GROUP BY jobtype;
+-------------+----------------+----------------+----------------+----------------+--------------+
| Job         | salary_dept_20 | salary_dept_50 | salary_dept_80 | salary_dept_90 | total_salary |
+-------------+----------------+----------------+----------------+----------------+--------------+
| dev         |         100000 |         120000 |              0 |              0 |       220000 |
| staff       |              0 |              0 |         130000 |              0 |       130000 |
| data analyst |              0 |              0 |              0 |          12000 |        12000 |
+-------------+----------------+----------------+----------------+----------------+--------------+
3 rows in set (0.00 sec)
```

9. 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.

```
mysql> SELECT
    ->     department_name AS `Department`,
    ->     location AS `Location`,
    ->     COUNT(employee_id) AS `Number of people`,
    ->     ROUND(AVG(salary), 2) AS `Salary`
    -> FROM
    -> empl
    -> GROUP BY
    -> department_name, location;
+------------+---------------+------------------+-----------+
| Department | Location      | Number of people | Salary    |
+------------+---------------+------------------+-----------+
| tech       | New York      |                2 | 110000.00 |
| Marketing  | San Francisco |                1 | 130000.00 |
| Finance    | Chicago       |                1 |  12000.00 |
+------------+---------------+------------------+-----------+
3 rows in set (0.00 sec)
```