DA _ SQL TASK PRACTICE - 4

```
mysql> -- 1. Create employee table (MySQL compatible types)
mysql> CREATE TABLE employee (
    ->     employee_id INT,
    ->     first_name VARCHAR(100),
    ->     last_name VARCHAR(100),
    ->     department VARCHAR(100),
    ->     salary DECIMAL(10, 2),
    ->     hire_date DATE
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql> -- 2. Insert sample data
mysql> INSERT INTO employee VALUES (1, 'Alice', 'Brown', 'HR', 62000, '2020-01-15');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES (2, 'Bob', 'Smith', 'Engineering', 75000, '2019-03-20');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES (3, 'Charlie', 'Davis', 'HR', 58000, '2018-07-01');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES (4, 'David', 'Evans', 'Sales', 49000, '2021-11-30');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES (5, 'Eve', 'Foster', 'Engineering', 81000, '2017-05-14');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES (6, 'Frank', 'Green', 'Engineering', 66000, '2022-02-10');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO employee VALUES (7, 'Grace', 'Hill', 'Sales', 52000, '2020-10-07');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES (8, 'Hank', 'Ivory', 'HR', 55000, '2019-04-25');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES (9, 'Ivy', 'Johnson', 'Marketing', 47000, '2021-03-18');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES (10, 'Jack', 'Knight', 'Marketing', 51000, '2023-06-29');
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> -- 3. HAVING and GROUP BY Queries
mysql> SELECT department
    -> FROM employee
    -> GROUP BY department
    -> HAVING AVG(salary) > 60000;
+-------------+
| department  |
+-------------+
| Engineering |
+-------------+
1 row in set (0.00 sec)

mysql>
mysql> SELECT department
    -> FROM employee
    -> GROUP BY department
    -> HAVING COUNT(*) > 2 AND AVG(salary) > 55000;
```

```
+-------------+
| department  |
+-------------+
| HR          |
| Engineering |
+-------------+
2 rows in set (0.00 sec)

mysql>
mysql> SELECT department
    -> FROM employee
    -> GROUP BY department
    -> HAVING SUM(salary) > 50000;
+-------------+
| department  |
+-------------+
| HR          |
| Engineering |
| Sales       |
| Marketing   |
+-------------+
4 rows in set (0.00 sec)

mysql>
mysql> SELECT *
    -> FROM employee
    -> WHERE department IN (
    ->     SELECT department
    ->     FROM employee
    ->     GROUP BY department
    ->     HAVING MIN(salary) > 45000
```

```
   -> );
+-------------+------------+-----------+-------------+----------+------------+
| employee_id | first_name | last_name | department  | salary   | hire_date  |
+-------------+------------+-----------+-------------+----------+------------+
|           1 | Alice      | Brown     | HR          | 62000.00 | 2020-01-15 |
|           2 | Bob        | Smith     | Engineering | 75000.00 | 2019-03-20 |
|           3 | Charlie    | Davis     | HR          | 58000.00 | 2018-07-01 |
|           4 | David      | Evans     | Sales       | 49000.00 | 2021-11-30 |
|           5 | Eve        | Foster    | Engineering | 81000.00 | 2017-05-14 |
|           6 | Frank      | Green     | Engineering | 66000.00 | 2022-02-10 |
|           7 | Grace      | Hill      | Sales       | 52000.00 | 2020-10-07 |
|           8 | Hank       | Ivory     | HR          | 55000.00 | 2019-04-25 |
|           9 | Ivy        | Johnson   | Marketing   | 47000.00 | 2021-03-18 |
|          10 | Jack       | Knight    | Marketing   | 51000.00 | 2023-06-29 |
+-------------+------------+-----------+-------------+----------+------------+
10 rows in set (0.00 sec)

mysql>
mysql> SELECT department
   -> FROM employee
   -> GROUP BY department
   -> HAVING COUNT(*) >= 3 AND AVG(salary) < 65000;
+------------+
| department |
+------------+
| HR         |
+------------+
1 row in set (0.00 sec)

mysql>
mysql> -- 4. ORDER BY Queries
```

```
mysql> SELECT * FROM employee ORDER BY first_name ASC;
+-------------+------------+-----------+-------------+----------+------------+
| employee_id | first_name | last_name | department  | salary   | hire_date  |
+-------------+------------+-----------+-------------+----------+------------+
|           1 | Alice      | Brown     | HR          | 62000.00 | 2020-01-15 |
|           2 | Bob        | Smith     | Engineering | 75000.00 | 2019-03-20 |
|           3 | Charlie    | Davis     | HR          | 58000.00 | 2018-07-01 |
|           4 | David      | Evans     | Sales       | 49000.00 | 2021-11-30 |
|           5 | Eve        | Foster    | Engineering | 81000.00 | 2017-05-14 |
|           6 | Frank      | Green     | Engineering | 66000.00 | 2022-02-10 |
|           7 | Grace      | Hill      | Sales       | 52000.00 | 2020-10-07 |
|           8 | Hank       | Ivory     | HR          | 55000.00 | 2019-04-25 |
|           9 | Ivy        | Johnson   | Marketing   | 47000.00 | 2021-03-18 |
|          10 | Jack       | Knight    | Marketing   | 51000.00 | 2023-06-29 |
+-------------+------------+-----------+-------------+----------+------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM employee ORDER BY hire_date DESC;
+-------------+------------+-----------+-------------+----------+------------+
| employee_id | first_name | last_name | department  | salary   | hire_date  |
+-------------+------------+-----------+-------------+----------+------------+
|          10 | Jack       | Knight    | Marketing   | 51000.00 | 2023-06-29 |
|           6 | Frank      | Green     | Engineering | 66000.00 | 2022-02-10 |
|           4 | David      | Evans     | Sales       | 49000.00 | 2021-11-30 |
|           9 | Ivy        | Johnson   | Marketing   | 47000.00 | 2021-03-18 |
|           7 | Grace      | Hill      | Sales       | 52000.00 | 2020-10-07 |
|           1 | Alice      | Brown     | HR          | 62000.00 | 2020-01-15 |
|           8 | Hank       | Ivory     | HR          | 55000.00 | 2019-04-25 |
|           2 | Bob        | Smith     | Engineering | 75000.00 | 2019-03-20 |
|           3 | Charlie    | Davis     | HR          | 58000.00 | 2018-07-01 |
|           5 | Eve        | Foster    | Engineering | 81000.00 | 2017-05-14 |
```

```
+-------------+------------+-----------+-------------+----------+------------+
10 rows in set (0.00 sec)

mysql> SELECT first_name, salary FROM employee ORDER BY salary ASC;
+------------+----------+
| first_name | salary   |
+------------+----------+
| Ivy        | 47000.00 |
| David      | 49000.00 |
| Jack       | 51000.00 |
| Grace      | 52000.00 |
| Hank       | 55000.00 |
| Charlie    | 58000.00 |
| Alice      | 62000.00 |
| Frank      | 66000.00 |
| Bob        | 75000.00 |
| Eve        | 81000.00 |
+------------+----------+
10 rows in set (0.00 sec)

mysql>
mysql> -- 5. Aggregate Functions by Department
mysql> SELECT department, SUM(salary) AS total_salary FROM employee GROUP BY department;
+-------------+--------------+
| department  | total_salary |
+-------------+--------------+
| HR          |    175000.00 |
| Engineering |    222000.00 |
| Sales       |    101000.00 |
| Marketing   |     98000.00 |
+-------------+--------------+
```

4 rows in set (0.00 sec)

```
mysql> SELECT department, AVG(salary) AS avg_salary FROM employee GROUP BY department;
+-------------+--------------+
| department  | avg_salary   |
+-------------+--------------+
| HR          | 58333.333333 |
| Engineering | 74000.000000 |
| Sales       | 50500.000000 |
| Marketing   | 49000.000000 |
+-------------+--------------+
4 rows in set (0.00 sec)

mysql> SELECT department, COUNT(*) AS total_employees FROM employee GROUP BY department;
+-------------+-----------------+
| department  | total_employees |
+-------------+-----------------+
| HR          |               3 |
| Engineering |               3 |
| Sales       |               2 |
| Marketing   |               2 |
+-------------+-----------------+
4 rows in set (0.00 sec)

mysql> SELECT department, MAX(salary) AS highest_salary FROM employee GROUP BY department;
+-------------+----------------+
| department  | highest_salary |
+-------------+----------------+
| HR          |       62000.00 |
| Engineering |       81000.00 |
| Sales       |       52000.00 |
```

```
| Marketing   |      51000.00 |
+-------------+----------------+
4 rows in set (0.00 sec)

mysql>
mysql> -- 6. Total salary grouped by hire year (MySQL-compatible)
mysql> SELECT YEAR(hire_date) AS hire_year, SUM(salary) AS total_salary
    -> FROM employee
    -> GROUP BY hire_year;
+-----------+--------------+
| hire_year | total_salary |
+-----------+--------------+
|      2020 |    114000.00 |
|      2019 |    130000.00 |
|      2018 |     58000.00 |
|      2021 |     96000.00 |
|      2017 |     81000.00 |
|      2022 |     66000.00 |
|      2023 |     51000.00 |
+-----------+--------------+
7 rows in set (0.00 sec)
```