ASSIGNMENT - 6

```
1. CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    employee_name VARCHAR(100),
    salary DECIMAL(10, 2),
    department_number INT,
    hire_date DATE
);
CREATE TABLE departments (
    department_number INT PRIMARY KEY,
    department_name VARCHAR(50),
    location VARCHAR(50)
);

INSERT INTO employees (employee_id, employee_name, salary,
department_number, hire_date) VALUES
(1, 'John Smith', 55000.00, 1, '2015-06-01'),
(2, 'Jane Doe', 60000.00, 2, '2016-03-10'),
(3, 'Mark Taylor', 45000.00, 1, '2017-07-23'),
(4, 'Emily Davis', 75000.00, 3, '2018-02-14'),
(5, 'Michael Brown', 52000.00, 1, '2016-11-18'),
(6, 'Linda White', 85000.00, 3, '2019-05-25'),
(7, 'Robert Green', 49000.00, 1, '2020-01-07'),
(8, 'Jennifer Clark', 62000.00, 2, '2015-10-15'),
(9, 'Jessica Miller', 58000.00, 1, '2019-08-09'),
(10, 'James Wilson', 72000.00, 3, '2014-12-22');

INSERT INTO departments (department_number, department_name, location)
VALUES
(1, 'HR', 'New York'),
(2, 'Finance', 'Chicago'),
(3, 'Engineering', 'San Francisco');
```

```
mysql> select * from employees;
+-------------+---------------+----------+-----------------+------------+
| employee_id | employee_name | salary   | department_number | hire_date  |
+-------------+---------------+----------+-----------------+------------+
|           1 | John Smith    | 55000.00 |               1 | 2015-06-01 |
|           2 | Jane Doe      | 60000.00 |               2 | 2016-03-10 |
|           3 | Mark Taylor   | 45000.00 |               1 | 2017-07-23 |
|           4 | Emily Davis   | 75000.00 |               3 | 2018-02-14 |
|           5 | Michael Brown | 52000.00 |               1 | 2016-11-18 |
|           6 | Linda White   | 85000.00 |               3 | 2019-05-25 |
|           7 | Robert Green  | 49000.00 |               1 | 2020-01-07 |
|           8 | Jennifer Clark| 62000.00 |               2 | 2015-10-15 |
|           9 | Jessica Miller| 58000.00 |               1 | 2019-08-09 |
|          10 | James Wilson  | 72000.00 |               3 | 2014-12-22 |
+-------------+---------------+----------+-----------------+------------+
10 rows in set (0.00 sec)
```

```
mysql> select * from departments;
+-------------------+-----------------+---------------+
| department_number | department_name | location      |
+-------------------+-----------------+---------------+
|                 1 | HR              | New York      |
|                 2 | Finance         | Chicago       |
|                 3 | Engineering     | San Francisco |
+-------------------+-----------------+---------------+
3 rows in set (0.00 sec)
```

1. Find the employee with the highest salary

```
mysql> select employee_name , salary from employees order by salary limit 1;
+---------------+----------+
| employee_name | salary   |
+---------------+----------+
| Mark Taylor   | 45000.00 |
+---------------+----------+
1 row in set (0.00 sec)
```

2. Find the names of employees who earn more than the average salary

```
mysql> select employee_name , salary from employees where salary > (select avg(salary) from employees);
+----------------+----------+
| employee_name  | salary   |
+----------------+----------+
| Emily Davis    | 75000.00 |
| Linda White    | 85000.00 |
| Jennifer Clark | 62000.00 |
| James Wilson   | 72000.00 |
+----------------+----------+
4 rows in set (0.04 sec)
```

3. Retrieve the name of the department where the employee 'Jane Doe' works

```
mysql> SELECT department_name
    -> FROM departments d
    -> JOIN employees e ON d.department_number = e.department_number
    -> WHERE e.employee_name = 'Jane Doe';
+-----------------+
| department_name |
+-----------------+
| Finance         |
+-----------------+
1 row in set (0.04 sec)
```

4. Find count of employees hired after the average hire date of all employees

```
mysql> select employee_name from employees where hire_date > (select avg(hire_date) from employees);
+----------------+
| employee_name  |
+----------------+
| Mark Taylor    |
| Emily Davis    |
| Linda White    |
| Robert Green   |
| Jessica Miller |
+----------------+
5 rows in set (0.00 sec)
```

5. Find the name of the department that has the lowest number of employees

```
mysql> select department_name from departments join employees on employees.department_number = departments.department_number group by
 departments.department_name order by count(employees.employee_id) ASC limit 1;
+-----------------+
| department_name |
+-----------------+
| Finance         |
+-----------------+
1 row in set (0.04 sec)
```

6. List the employees whose salary is above the average salary of their department

```
mysql> select employee_name, salary from employees where salary > (select avg(salary) from employees where employees.department_numbe
r = employees.department_number);
+----------------+----------+
| employee_name  | salary   |
+----------------+----------+
| Emily Davis    | 75000.00 |
| Linda White    | 85000.00 |
| Jennifer Clark | 62000.00 |
| James Wilson   | 72000.00 |
+----------------+----------+
4 rows in set (0.00 sec)
```

7. Find employees who work in a department located in 'Chicago'

```
mysql> select employee_name from employees join departments on employees.department_number = departments.department_number where depa
rtments.location = 'Chicago';
+----------------+
| employee_name  |
+----------------+
| Jane Doe       |
| Jennifer Clark |
+----------------+
2 rows in set (0.00 sec)
```

8. Retrieve employees who have the same salary as the employee with ID 2

```
mysql> SELECT employee_name
    -> FROM employees
    -> WHERE salary = (SELECT salary FROM employees WHERE employee_id = 2);
+---------------+
| employee_name |
+---------------+
| Jane Doe      |
+---------------+
1 row in set (0.01 sec)
```

9. Find departments that have employees earning more than $60,000

```
mysql> select department_name from departments join employees on departments.department_number = employees.department_number where em
ployees.salary > 60000;
+-----------------+
| department_name |
+-----------------+
| Engineering     |
| Engineering     |
| Finance         |
| Engineering     |
+-----------------+
4 rows in set (0.00 sec)
```

10. List all employees whose salary is higher than any employee in department 1

```
mysql> select employee_name from employees join departments on employees.department_number = departments.department_number where sala
ry  > (select max(salary) from employees where department_number = '1');
+----------------+
| employee_name  |
+----------------+
| Jane Doe       |
| Emily Davis    |
| Linda White    |
| Jennifer Clark |
| James Wilson   |
+----------------+
5 rows in set (0.00 sec)
```

11. List all employees along with their department name

```
mysql> select employee_name,department_name from employees join departments on employees.department_number = departments.department_n
umber;
+----------------+-----------------+
| employee_name  | department_name |
+----------------+-----------------+
| John Smith     | HR              |
| Jane Doe       | Finance         |
| Mark Taylor    | HR              |
| Emily Davis    | Engineering     |
| Michael Brown  | HR              |
| Linda White    | Engineering     |
| Robert Green   | HR              |
| Jennifer Clark | Finance         |
| Jessica Miller | HR              |
| James Wilson   | Engineering     |
+----------------+-----------------+
10 rows in set (0.00 sec)
```

12. Find the number of employees in each department

```
mysql> SELECT d.department_name, COUNT(e.employee_id) AS employee_count
    -> FROM Departments d
    -> LEFT JOIN Employees e ON d.department_number = e.department_number
    -> GROUP BY d.department_name;
+-----------------+----------------+
| department_name | employee_count |
+-----------------+----------------+
| HR              |              5 |
| Finance         |              2 |
| Engineering     |              3 |
+-----------------+----------------+
3 rows in set (0.00 sec)
```

13. List employees who work in 'Engineering' department

```
mysql> SELECT e.employee_name
    -> FROM Employees e
    -> JOIN Departments d ON e.department_number = d.department_number
    -> WHERE d.department_name = 'Engineering';
+---------------+
| employee_name |
+---------------+
| Emily Davis   |
| Linda White   |
| James Wilson  |
+---------------+
3 rows in set (0.04 sec)
```

14. Find the department name and location for each employee

```
mysql> SELECT e.employee_name, d.department_name, d.location
    -> FROM Employees e
    -> JOIN Departments d ON e.department_number = d.department_number;
+----------------+-----------------+---------------+
| employee_name  | department_name | location      |
+----------------+-----------------+---------------+
| John Smith     | HR              | New York      |
| Jane Doe       | Finance         | Chicago       |
| Mark Taylor    | HR              | New York      |
| Emily Davis    | Engineering     | San Francisco |
| Michael Brown  | HR              | New York      |
| Linda White    | Engineering     | San Francisco |
| Robert Green   | HR              | New York      |
| Jennifer Clark | Finance         | Chicago       |
| Jessica Miller | HR              | New York      |
| James Wilson   | Engineering     | San Francisco |
+----------------+-----------------+---------------+
10 rows in set (0.00 sec)
```

15. Retrieve employees who work in departments located in 'New York' or 'Chicago'

```
mysql> SELECT e.employee_name
    -> FROM Employees e
    -> JOIN Departments d ON e.department_number = d.department_number
    -> WHERE d.location IN ('New York', 'Chicago');
+----------------+
| employee_name  |
+----------------+
| John Smith     |
| Jane Doe       |
| Mark Taylor    |
| Michael Brown  |
| Robert Green   |
| Jennifer Clark |
| Jessica Miller |
+----------------+
7 rows in set (0.04 sec)
```

16. Find departments with more than 3 employees

```
mysql> SELECT d.department_name
    -> FROM Departments d
    -> JOIN Employees e ON d.department_number = e.department_number
    -> GROUP BY d.department_name
    -> HAVING COUNT(e.employee_id) > 3;
+-----------------+
| department_name |
+-----------------+
| HR              |
+-----------------+
1 row in set (0.00 sec)
```

17. List employees who were hired before the average hire date of their department

```
mysql> SELECT e.employee_name
    -> FROM Employees e
    -> WHERE e.hire_date < (
    ->   SELECT AVG(hire_date)
    ->   FROM Employees
    ->   WHERE department_number = e.department_number
    -> );
+----------------+
| employee_name  |
+----------------+
| John Smith     |
| Mark Taylor    |
| Michael Brown  |
| Jennifer Clark |
| James Wilson   |
+----------------+
5 rows in set (0.00 sec)
```

18. Find the name and salary of the employee who works in the 'HR' department and has the highest salary

```
mysql>
mysql> SELECT e.employee_name, e.salary
    -> FROM Employees e
    -> JOIN Departments d ON e.department_number = d.department_number
    -> WHERE d.department_name = 'HR'
    -> ORDER BY e.salary DESC
    -> LIMIT 1;
+---------------+----------+
| employee_name  | salary   |
+---------------+----------+
| Jessica Miller | 58000.00 |
+---------------+----------+
1 row in set (0.00 sec)
```

19. Retrieve the department name and the total salary expense for each department20. Find all employees who work in departments with a salary expense greater than $200,000.

```
mysql> SELECT e.employee_name
    -> FROM Employees e
    -> WHERE e.department_number IN (
    ->     SELECT d.department_number
    ->     FROM Employees e
    ->     JOIN Departments d ON e.department_number = d.department_number
    ->     GROUP BY d.department_number
    ->     HAVING SUM(e.salary) > 200000
    -> );
+---------------+
| employee_name  |
+---------------+
| John Smith     |
| Mark Taylor    |
| Emily Davis    |
| Michael Brown  |
| Linda White    |
| Robert Green   |
| Jessica Miller |
| James Wilson   |
+---------------+
8 rows in set (0.04 sec)
```