1. **MySQL Aggregate Function Exercises: Write a query to list the number of jobs available in the employees table**

Write a query to list the number of jobs available in the employees table.

SELECT COUNT(DISTINCT job\_id) FROM employees;

1. **MySQL Aggregate Function Exercises: Write a query to get the total salaries payable to employees**

Write a query to get the total salaries payable to employees.

SELECT SUM(salary) FROM employees;

1. **MySQL Aggregate Function Exercises: Write a query to get the minimum salary from employees table**

Write a query to get the minimum salary from employees table.

SELECT MIN(salary) FROM employees;

1. **MySQL Aggregate Function Exercises: Write a query to get the maximum salary of an employee working as a Programmer**

Write a query to get the maximum salary of an employee working as a Programmer.

SELECT MAX(salary) FROM employees WHERE job\_id = 'IT\_PROG';

1. **MySQL Aggregate Function Exercises: Write a query to get the average salary and number of employees working the department 90**

Write a query to get the average salary and number of employees working the department 90.

SELECT AVG(salary),count(\*) FROM employees WHERE department\_id = 90;

1. **MySQL Aggregate Function Exercises: Write a query to get the highest, lowest, sum, and average salary of all employees**

Write a query to get the highest, lowest, sum, and average salary of all employees.

SELECT ROUND(MAX(salary),0) 'Maximum', ROUND(MIN(salary),0) 'Minimum', ROUND(SUM(salary),0) 'Sum', ROUND(AVG(salary),0) 'Average' FROM employees;

1. **MySQL Aggregate Function Exercises: Write a query to get the number of employees with the same job**

Write a query to get the number of employees with the same job.

SELECT job\_id, COUNT(\*) FROM employees GROUP BY job\_id;

1. **MySQL Aggregate Function Exercises: Write a query to get the difference between the highest and lowest salaries**

Write a query to get the difference between the highest and lowest salaries.

SELECT MAX(salary) - MIN(salary) DIFFERENCE FROM employees;

1. **MySQL Aggregate Function Exercises: Write a query to find the manager ID and the salary of the lowest-paid employee for that manager**

Write a query to find the manager ID and the salary of the lowest-paid employee for that manager.

SELECT manager\_id, MIN(salary)

FROM employees

WHERE manager\_id IS

NOT NULL GROUP BY manager\_id

ORDER BY MIN(salary) DESC;

1. **MySQL Aggregate Function Exercises: Write a query to get the department ID and the total salary payable in each department**

Write a query to get the department ID and the total salary payable in each department.

SELECT department\_id, SUM(salary) FROM employees GROUP BY department\_id;

1. **MySQL Aggregate Function Exercises: Write a query to get the average salary for each job ID excluding programmer**

Write a query to get the average salary for each job ID excluding programmer.

SELECT job\_id, AVG(salary) FROM employees WHERE job\_id <> 'IT\_PROG' GROUP BY job\_id;

1. **MySQL Aggregate Function Exercises: Write a query to get the total salary, maximum, minimum, average salary of employees, for department ID 90 only**

Write a query to get the total salary, maximum, minimum, average salary of employees (job ID wise), for department ID 90 only.

SELECT job\_id, SUM(salary), AVG(salary), MAX(salary), MIN(salary) FROM employees WHERE department\_id = '90' GROUP BY job\_id;

1. **MySQL Aggregate Function Exercises: Write a query to get the job ID and maximum salary greater than or equal to $4000**

Write a query to get the job ID and maximum salary of the employees where maximum salary is greater than or equal to $4000.

SELECT job\_id, MAX(salary) FROM employees GROUP BY job\_id HAVING MAX(salary) >=4000;

1. **MySQL Aggregate Function Exercises: Write a query to get the average salary for all departments employing more than 10 employees**

Write a query to get the average salary for all departments employing more than 10 employees.

SELECT job\_id, AVG(salary), COUNT(\*) FROM employees GROUP BY department\_id HAVING COUNT(\*) > 10;