**My Sql Database Questions and Answers**

1> Retrieve the first 5 employees by highest salary.

A. SELECT name, salary FROM employees ORDER BY salary DESC LIMIT 5;

2> Retrieve the 5 employees with the lowest salary, skipping the first 10 records.

A. SELECT name, salary FROM employees ORDER BY salary ASC LIMIT 5 OFFSET 10;

3> Display each department's total salary, but only show departments where the total salary exceeds $30,000.

A. SELECT department, SUM(salary) AS total\_salary FROM employees GROUP BY department HAVING total\_salary > 30000;

4> For each employee, display their salary and a note if it is above or below $7000.

A. SELECT name,

salary,

CASE

WHEN salary > 7000 THEN 'Above $7000'

ELSE 'Below $7000'

END AS salary\_note

FROM employees;

5> List the projects that started in the last 6 months.

A. SELECT \*

FROM projects

WHERE start\_date >= DATE\_SUB(CURDATE(), INTERVAL 6 MONTH);

6> Display all projects, including those with no employees assigned.

SELECT p.project\_id, p.project\_name, e.employee\_id, e.employee\_name

FROM projects p

LEFT JOIN employees e ON p.project\_id = e.project\_id;

7> Increase the salary of all employees in the 'Finance' department by 12%.

A. UPDATE employees SET salary = salary \* 1.12 WHERE department = 'Finance';

8> Retrieve the first and last names of employees whose last names start with 'S'.

A. SELECT first\_name, last\_name FROM employees WHERE last\_name LIKE 'S%';

9> Count the number of employees assigned to each department.

A. SELECT department, COUNT(\*) AS employee\_count

FROM employees

GROUP BY department;

10> Find the total number of hours worked by employees on each project.

A. SELECT project\_id, SUM(hours\_worked) AS total\_hours FROM project\_hours GROUP BY project\_id;

11> Select the employees who earn more than the average salary.

A. SELECT \* FROM employees WHERE salary > (SELECT AVG(salary) FROM employees);

12> Display each project's start date in the format 'Month Day, Year' (e.g., January 01, 2024)

A. SELECT project\_id, project\_name, DATE\_FORMAT(start\_date, '%M %d, %Y') AS formatted\_start\_date FROM projects;

13> List all employees, their department names, and the projects they have worked on.

A. SELECT e.first\_name,

e.last\_name,

d.department\_name,

p.project\_name

FROM employees e

JOIN departments d ON e.department\_id = d.department\_id

LEFT JOIN employee\_projects ep ON e.employee\_id = ep.employee\_id

LEFT JOIN projects p ON ep.project\_id = p.project\_id;

14> List all projects, showing 'Assigned' if any employees are assigned and 'Unassigned' otherwise.

A. SELECT p.project\_id, p.project\_name,

CASE

WHEN ep.employee\_id IS NOT NULL THEN 'Assigned'

ELSE 'Unassigned'

END AS assignment\_status

FROM projects p

LEFT JOIN employee\_projects ep ON p.project\_id = ep.project\_id

GROUP BY p.project\_id, p.project\_name;

15> Find the average salary of employees in each department, but only show departments where more than 5 employees work

A. SELECT department\_id, AVG(salary) AS average\_salary FROM employees GROUP BY department\_id HAVING COUNT(\*) > 5;

16> Concatenate employees' first and last names into a single full name, separated by a space.

A. SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name FROM employees;

17> Set all employees with salaries above $8000 and belonging to the 'IT' department to a new salary of $8500.

A. UPDATE employees SET salary = 8500 WHERE salary > 8000 AND department = 'IT';

18> Find the employee with the highest salary who works in the 'HR' department.

A. SELECT \* FROM employees WHERE department = 'HR' ORDER BY salary DESC LIMIT 1;

19> Find employees who have been assigned to projects that will end within the next 2 months

A. SELECT DISTINCT e.employee\_id,

e.first\_name,

e.last\_name

FROM employees e

JOIN employee\_projects ep ON e.employee\_id = ep.employee\_id

JOIN projects p ON ep.project\_id = p.project\_id

WHERE p.end\_date BETWEEN CURDATE() AND DATE\_ADD(CURDATE(), INTERVAL 2 MONTH);

20> For each project, show the total salary billed based on the hours worked by employees, but only for projects where total billing exceeds $10,000

A. SELECT p.project\_id,

p.project\_name,

SUM(e.salary \* ep.hours\_worked) AS total\_billing

FROM projects p

JOIN employee\_projects ep ON p.project\_id = ep.project\_id

JOIN employees e ON ep.employee\_id = e.employee\_id

GROUP BY p.project\_id, p.project\_name

HAVING total\_billing > 10000;