SQL Questions MySQL Test (20 Questions) – Difficult Level 1. Basic Query with LIMIT Retrieve the first 5 employees by highest salary. Hint: Use LIMIT .

* SELECT \* from Employees order by salary DESC LIMIT 5;

A screenshot of a computer

Description automatically generated

2. OFFSET with Sorting Retrieve the 5 employees with the lowest salary, skipping the first 10 records. Hint: Combine ORDER BY with LIMIT and OFFSET .

->SELECT \* from Employees order by salary DESC LIMIT 5 OFFSET 10

A screenshot of a computer

Description automatically generated

3. Aggregate with HAVING Display each department's total salary, but only show departments where the total salary exceeds $30,000. Hint: Use GROUP BY and HAVING .

-> SELECT department\_id,sum(salary) as s from Employees GROUP BY department\_id having s> 15000

A screenshot of a computer

Description automatically generated

4. Conditional Logic ( CASE Statement) For each employee, display their salary and a note if it is above or below $7000. Hint: Use the CASE statement.

-> select employee\_id,first\_name,last\_name,salary ,

CASE

when salary>7000 then 'Above $7000'

ELSE 'Below $7000'

END as Note

from Employees;

A screenshot of a computer

Description automatically generated

5. Date Functions List the projects that started in the last 6 months. Hint: Use CURDATE() and DATE\_SUB() .

-> select project\_id, project\_name , start\_date from Projects

where start\_date>= date\_sub(CURDATE(), INTERVAL 6 MONTH);

6. RIGHT JOIN with Null Handling Display all projects, including those with no employees assigned. Hint: Use RIGHT JOIN .

7. Update Query with a Condition Increase the salary of all employees in the 'Finance' department by 12%. Hint: Use UPDATE .

8. String Functions Retrieve the first and last names of employees whose last names start with 'S'. Hint: Use LIKE .

-> SELECT first\_name, last\_name

FROM employees

WHERE last\_name LIKE 'S%';

A screenshot of a computer

Description automatically generated

9. Aggregation with COUNT Count the number of employees assigned to each department. Hint: Use COUNT and GROUP BY .

-> SELECT d.department\_name, COUNT(e.employee\_id) AS employee\_count

FROM departments d

LEFT JOIN employees e ON d.department\_id = e.department\_id

GROUP BY d.department\_name;

A screenshot of a computer program

Description automatically generated

10. JOIN and Aggregation Find the total number of hours worked by employees on each project. Hint: Use JOIN and SUM .

-> SELECT p.project\_name, w.employee\_id , SUM(w.hours\_worked)AS total\_hours

FROM projects p

JOIN EmployeeProjects w ON p.project\_id = w.project\_id

GROUP BY w.project\_id

A screenshot of a computer

Description automatically generated

11. Subquery with Aggregation Select the employees who earn more than the average salary. Hint: Use a subquery to find the average salary.

-> SELECT first\_name, last\_name, salary

FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);

A screenshot of a computer

Description automatically generated

12. Date Formatting Display each project's start date in the format 'Month Day, Year' (e.g., January 01, 2024). Hint: Use DATE\_FORMAT() .

13. Multiple Joins List all employees, their department names, and the projects they have worked on. Hint: Use multiple JOIN clauses.

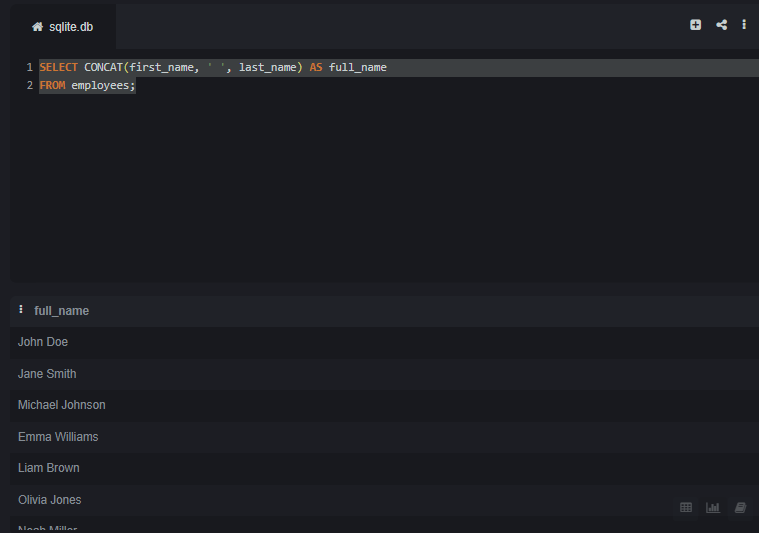
14. Using RIGHT JOIN and CASE List all projects, showing 'Assigned' if any employees are assigned and 'Unassigned' otherwise. Hint: Use RIGHT JOIN with CASE .

15. Aggregation with Conditions Find the average salary of employees in each department, but only show departments where more than 5 employees work. Hint: Use GROUP BY with HAVING .

16. String Manipulation Concatenate employees' first and last names into a single full name, separated by a space. Hint: Use CONCAT() .

-> SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name

FROM employees;



17. Update Query with Multiple Conditions Set all employees with salaries above $8000 and belonging to the 'IT' department to a new salary of $8500. Hint: Use UPDATE with WHERE .

-> UPDATE employees e

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

SET e.salary = 8500

WHERE e.salary > 8000 AND d.department\_name = 'IT';

18. Nested Queries Find the employee with the highest salary who works in the 'HR' department. Hint: Use a subquery with MAX() .

-> SELECT \*

FROM employees

WHERE salary = (

SELECT MAX(salary)

FROM employees e

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

WHERE d.department\_name = 'HR'

);

A screenshot of a computer

Description automatically generated

19. Date Calculation Find employees who have been assigned to projects that will end within the next 2 months. Hint: Use CURDATE() and DATE\_ADD() .

->

20. Join and Aggregation with SUM and HAVING 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. Hint: Use JOIN , SUM , and HAVING .