MySQL Predefined Functions and Interview Queries with Full Examples

Topic	Definition, Full Query & Explanation
CURDATE()	■ Definition: Returns the current date. ■ Query: SELECT CURDATE(); ■ Explanation: Displays today’s date from the system.
NOW()	■ Definition: Returns current date and time. ■ Query: SELECT NOW(); ■ Explanation: Returns both date and time of execution.
DATE_ADD()	■ Definition: Adds a time interval to a date. ■ Query: SELECT DATE_ADD('2025-10-04', INTERVAL 10 DAY); ■ Explanation: Adds 10 days to the given date.
DATE_SUB()	■ Definition: Subtracts a time interval from a date. ■ Query: SELECT DATE_SUB('2025-10-04', INTERVAL 2 MONTH); ■ Explanation: Subtracts 2 months from the date.
YEAR()	■ Definition: Extracts year from a date. ■ Query: SELECT * FROM employees WHERE YEAR(join_date) = YEAR(CURDATE()); ■ Explanation: Retrieves employees who joined in the current year.
MONTH()	■ Definition: Extracts month from a date. ■ Query: SELECT * FROM employees WHERE MONTH(join_date) = 10; ■ Explanation: Retrieves employees who joined in October.
DATEDIFF()	■ Definition: Returns number of days between two dates. ■ Query: SELECT DATEDIFF('2025-10-10', '2025-10-04') AS days_difference; ■ Explanation: Shows the difference of 6 days between the two dates.
IFNULL()	■ Definition: Replaces NULL with an alternate value. ■ Query: SELECT name, IFNULL(commission, 0) AS commission_value FROM employees; ■ Explanation: Displays commission as 0 where it's NULL.
COALESCE()	■ Definition: Returns first non-null value in the list. ■ Query: SELECT COALESCE(middle_name, first_name, 'N/A') AS preferred_name FROM employees; ■ Explanation: Displays the first available (non-null) value.

CONCAT()	■ Definition: Combines multiple strings. ■ Query: SELECT CONCAT(first_name, ' ', last_name) AS full_name FROM employees; ■ Explanation: Combines first and last name of employees.
UPPER()	■ Definition: Converts text to uppercase. ■ Query: SELECT UPPER(name) AS upper_name FROM employees; ■ Explanation: Converts all employee names to uppercase.
LOWER()	■ Definition: Converts text to lowercase. ■ Query: SELECT LOWER(department_name) FROM departments; ■ Explanation: Converts department names to lowercase.
ROUND()	■ Definition: Rounds a number to a given decimal place. ■ Query: SELECT ROUND(salary, 0) AS rounded_salary FROM employees; ■ Explanation: Rounds salary to nearest whole number.
SUM()	■ Definition: Calculates total of a numeric column. ■ Query: SELECT SUM(salary) AS total_salary FROM employees; ■ Explanation: Returns total salary of all employees.
AVG()	■ Definition: Calculates average of a numeric column. ■ Query: SELECT AVG(salary) AS average_salary FROM employees; ■ Explanation: Shows average salary of employees.
COUNT()	■ Definition: Counts number of rows matching a condition. ■ Query: SELECT COUNT(*) AS total_employees FROM employees WHERE department_id = 2; ■ Explanation: Counts employees in department 2.
GROUP BY	■ Definition: Groups rows with the same values for aggregation. ■ Query: SELECT department_id, AVG(salary) AS avg_salary FROM employees GROUP BY department_id; ■ Explanation: Finds average salary in each department.
HAVING	■ Definition: Filters aggregated results. ■ Query: SELECT department_id, AVG(salary) AS avg_salary FROM employees GROUP BY department_id HAVIN ■ Explanation: Returns departments with avg salary greater than 50,000.
ORDER BY	■ Definition: Sorts query results. ■ Query: SELECT * FROM employees ORDER BY salary DESC; ■ Explanation: Displays employees sorted by salary in descending order.

LIMIT	■ Definition: Restricts number of rows returned. ■ Query: SELECT * FROM employees ORDER BY salary DESC LIMIT 3; ■ Explanation: Returns top 3 highest-paid employees.
Second Highest Salary	■ Question: Retrieve the second highest salary. ■ Query: SELECT salary FROM employees ORDER BY salary DESC LIMIT 1 OFFSET 1; ■ Explanation: Skips the highest and fetches the second highest salary.
Employees Joined This Year	■ Question: Get list of employees who joined in current year. ■ Query: SELECT * FROM employees WHERE YEAR(join_date) = YEAR(CURDATE()); ■ Explanation: Compares join year with current year.
Duplicate Records	■ Question: Find duplicate records. ■ Query: SELECT name, COUNT(*) FROM employees GROUP BY name HAVING COUNT(*) > 1; ■ Explanation: Shows employee names that appear more than once.
Employees Without Manager	■ Question: Display employees without a manager. ■ Query: SELECT * FROM employees WHERE manager_id IS NULL; ■ Explanation: Lists employees who don't have a manager assigned.
Total Salary by Department	■ Question: Find total salary paid to each department. ■ Query: SELECT department_id, SUM(salary) AS total_salary FROM employees GROUP BY department_id; ■ Explanation: Aggregates total salary by department.
Highest Average Salary Dept	■ Question: Department with highest average salary. ■ Query: SELECT department_id, AVG(salary) AS avg_salary FROM employees GROUP BY department_id ORDE ■ Explanation: Returns department having highest average salary.
Top 3 Salaries per Department	■ Question: Get top 3 highest salaries from each department. ■ Query: SELECT e1.* FROM employees e1 WHERE 3 > (SELECT COUNT(DISTINCT e2.salary) FROM employee ■ Explanation: Uses subquery to get top 3 salaries within each department.
Delete Duplicates	■ Question: Remove duplicate records. ■ Query: DELETE e1 FROM employees e1 INNER JOIN employees e2 WHERE e1.id > e2.id AND e1.name = e2.n ■ Explanation: Deletes duplicates while keeping one copy.
Employee and Department Names	■ Question: Display employee names with their department. ■ Query: SELECT e.name, d.department_name FROM employees e INNER JOIN departments d ON e.department_ ■ Explanation: Joins employees and departments tables to show names together.