

**1. Write the SQL Expressions for the following queries using suitable SQL operators .**

- A. `SELECT course_id, title, credits FROM course WHERE dept_name IN ('Physics', 'Music', 'Finance', 'Biology');`
- B. `SELECT * FROM instructor WHERE name LIKE 'K%' AND salary > 65000;`
- C. `SELECT name, dept_name, salary + (salary * 0.05) + (salary * 0.20) AS gross_salary, (salary + (salary * 0.05) + (salary * 0.20)) - (salary * 0.30) AS net_salary FROM instructor;`
- D. `SELECT * FROM instructor WHERE salary BETWEEN 60000 AND 80000;`
- E. `SELECT * FROM instructor WHERE name LIKE '_r%';`
- F. `SELECT name FROM instructor WHERE dept_name = 'Comp.Sci.' ORDER BY salary DESC;`
- G. `UPDATE instructor SET salary = salary * 1.15;`
- H. `UPDATE instructor SET salary = salary * 1.03 WHERE dept_name = 'Comp.Sci.' AND salary < 70000;`
- I. `SELECT name, salary * 12 AS annual_salary FROM instructor;`
- J. `UPDATE course SET title = 'Game Theory' WHERE title = 'Game Design';`
- K. `DELETE FROM instructor WHERE dept_name = 'History';`
- L. `DELETE FROM course WHERE course_id LIKE 'BIO%';`

**2. Write the SQL Expressions for the following queries using suitable SQL aggregate function.**

- A. `SELECT AVG(salary) FROM instructor WHERE dept_name = 'Physics';`
- B. `SELECT dept_name, AVG(salary) AS avg_salary FROM instructor GROUP BY dept_name;`

- C. `SELECT id, name, dept_name FROM instructor WHERE salary = (SELECT MAX(salary) FROM instructor);`
- D. `SELECT COUNT(*) FROM instructor WHERE dept_name = 'Comp.Sci.';`
- E. `SELECT SUM(credits) FROM course WHERE dept_name = 'Comp.Sci.';`
- F. `SELECT dept_name, COUNT(*) AS no_of_instructors, SUM(salary) AS total_salary FROM instructor WHERE dept_name IN ('Physics', 'Comp.Sci.') GROUP BY dept_name;`
- G. `SELECT dept_name, SUM(credits) FROM course WHERE dept_name IN ('Comp.Sci.', 'Biology') GROUP BY dept_name;`
- H. `SELECT building, SUM(budget) AS total_budget FROM department GROUP BY building;`
- I. `SELECT dept_name, COUNT(*) AS no_of_instructors FROM instructor GROUP BY dept_name;`
- J. `SELECT dept_name, COUNT(*) AS no_of_instructors FROM instructor GROUP BY dept_name ORDER BY no_of_instructors DESC;`
- K. `SELECT semester, COUNT(*) AS no_of_courses FROM course GROUP BY semester;`
- L. `SELECT dept_name FROM instructor GROUP BY dept_name HAVING COUNT(*) < 2;`
- M. `SELECT dept_name, COUNT(*) AS no_of_instructors FROM instructor WHERE dept_name <> 'Finance' GROUP BY dept_name HAVING COUNT(*) >= 2 ORDER BY no_of_instructors DESC;`
- N. `SELECT dept_name FROM instructor GROUP BY dept_name HAVING SUM(salary) > 50000;`
- O. `SELECT building, SUM(budget) AS total_budget FROM department WHERE building = 'Watson' GROUP BY building;`
- P. `SELECT MAX(salary) FROM instructor WHERE dept_name = 'Comp.Sci.';`

**3. Write the SQL Expressions for the following queries using suitable SQL scalar function.**

- A. `SELECT INITCAP('yourname') AS Name;`
- B. `SELECT SUBSTR('yourname', 2, 5) AS substring;`
- C. `SELECT LENGTH('Your University Full Name') AS length_of_university_name;`

- D. `SELECT INITCAP(name) FROM instructor;`
- E. `SELECT dept_name, SUBSTR(dept_name, 1, 3) AS dept_code FROM department;`
- F. `SELECT name, MONTH(join_date) AS joining_month FROM instructor;`
- G. `SELECT name, DATE_FORMAT(join_date, '%d/%m/%y') AS joining_date FROM instructor;`
- H. `SELECT name, TIMESTAMPDIFF(MONTH, join_date, CURDATE()) AS experience_in_months  
FROM instructor;`
- I. `SELECT name, CONCAT(TIMESTAMPDIFF(YEAR, join_date, CURDATE()), ' Years ',  
MOD(TIMESTAMPDIFF(MONTH, join_date, CURDATE()), 12), ' Months') AS experience FROM  
instructor;`
- J. `SELECT name, DAY(join_date) AS joining_day FROM instructor;`
- K. `SELECT CURDATE() + INTERVAL 15 DAY AS date_after_15_days;`
- L. `SELECT TRUNCATE(94204.27348, 2) AS truncated_value;`
- M. `SELECT 5 + 89 AS result;`
- N. `SELECT SQRT(6464312) AS square_root;`
- O. `SELECT LOWER('HELLO ITER') AS lower_case;`