**Question 1 :**

SELECT e.\* FROM employees e JOIN ( SELECT dept\_id, salary FROM ( SELECT dept\_id, salary, DENSE\_RANK() OVER (PARTITION BY dept\_id ORDER BY salary DESC) AS rnk FROM employees) ranked WHERE rnk <= 5) top5 ON e.dept\_id = top5.dept\_id AND e.salary = top5.salary;

**Question 2 :**

DELIMITER //

CREATE FUNCTION getWorkingDays(joining\_date DATE)

RETURNS INT

DETERMINISTIC

BEGIN

DECLARE total\_days INT;

SET total\_days = DATEDIFF(CURDATE(), joining\_date);

RETURN total\_days;

END //

DELIMITER ;

**Question 3 :**

DELIMITER //

CREATE PROCEDURE updateSalary(IN emp\_id INT)

BEGIN

DECLARE work\_days INT;

DECLARE current\_salary DECIMAL(10,2);

SELECT getWorkingDays(joining\_date), salary

INTO work\_days, current\_salary

FROM employees

WHERE emp\_no = emp\_id;

IF work\_days > 1000 THEN

SET current\_salary = current\_salary \* 1.15; -- 15% hike

ELSEIF work\_days > 500 THEN

SET current\_salary = current\_salary \* 1.10; -- 10% hike

ELSE

SET current\_salary = current\_salary \* 1.05; -- 5% hike

END IF;

UPDATE employees

SET salary = current\_salary

WHERE emp\_no = emp\_id;

END //

DELIMITER ;