|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** | | | | |
| **Program Name:** B. Tech | | | | **Assignment Type: Lab** | | | **Academic Year:**2025-2026 | | |
| **Course Coordinator Name** | | | | Venkataramana Veeramsetty | | | | | |
| **Instructor(s) Name** | | | | |  | | --- | | Dr. V. Venkataramana (Co-ordinator) | | Dr. T. Sampath Kumar | | Dr. Pramoda Patro | | Dr. Brij Kishor Tiwari | | Dr.J.Ravichander | | Dr. Mohammand Ali Shaik | | Dr. Anirodh Kumar | | Mr. S.Naresh Kumar | | Dr. RAJESH VELPULA | | Mr. Kundhan Kumar | | Ms. Ch.Rajitha | | Mr. M Prakash | | Mr. B.Raju | | Intern 1 (Dharma teja) | | Intern 2 (Sai Prasad) | | Intern 3 (Sowmya) | | NS\_2 ( Mounika) | | | | | | |
| **Course Code** | | | 24CS002PC215 | **Course Title** | | AI Assisted Coding | | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | | Week9 - Tuesday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicable to**  **Batches** | |  | | | |
| **AssignmentNumber:16.2**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***Expected Time***  ***to complete*** |  |
|  | 1 | * 1. Display all records from the employee’s table.   Query:SELECT \* FROM employees;     1. 2. Display only employee names and their departments.   Query:SELECT first\_name, last\_name, department FROM employees;     1. 3. Show unique department names.   Query:SELECT DISTINCT department FROM employees;     1. 4. Find employees with salary greater than 50000.   Query:SELECT \* FROM employees WHERE salary > 50000;     1. 5. Find employees from the IT department.   Query:SELECT \* FROM employees WHERE department = 'IT';     1. 6. Display employees hired after 2020.   Query:SELECT \* FROM employees WHERE hire\_date > '2020-01-01';     1. 7. Show employees in ascending order of salary.   Query:SELECT \* FROM employees ORDER BY salary ASC;     1. 8. Show top 3 highest-paid employees.   Query:SELECT \* FROM employees ORDER BY salary DESC LIMIT 3;     1. 9. Count total employees in the company.   Query:SELECT COUNT(\*) AS total\_employees FROM employees;     1. 10. Find the average salary of employees.   Query:SELECT AVG(salary) AS average\_salary FROM employees;     1. 11. Find the highest and lowest salary.   Query:SELECT MAX(salary) AS highest\_salary FROM employees;    SELECT MIN(salary) AS lowest\_salary FROM employees;     1. 12. Find total salary expenditure per department.   Query:SELECT department, SUM(salary) AS total\_salary\_expenditure  FROM employees  GROUP BY department;     1. 13. Display departments having more than one employee.   Query:SELECT department  FROM employees  GROUP BY department  HAVING COUNT(\*) > 1;     1. 14. Show average salary by department.   Query:SELECT department, AVG(salary) AS average\_salary  FROM employees  GROUP BY department;     1. 15. Count employees hired each year.   Query:SELECT YEAR(hire\_date) AS hire\_year, COUNT(\*) AS employees\_hired  FROM employees  GROUP BY YEAR(hire\_date)  ORDER BY hire\_year;     1. 16. List employees with their department locations.   Query:SELECT e.emp\_id, e.first\_name, e.last\_name, e.department, d.location  FROM employees e  JOIN department d ON e.department = d.dept\_name;     1. 17. Find employees working in Bangalore.   Query:SELECT e.\*  FROM employees e  JOIN department d ON e.department = d.dept\_name  WHERE d.location = 'Bangalore';     1. 18. Display all employees even if they don’t belong to a department.   Query:SELECT e.\*  FROM employees e  LEFT JOIN department d ON e.department = d.dept\_name;     1. 19. Find departments with no employees.   Query:SELECT d.dept\_name  FROM department d  LEFT JOIN employees e ON d.dept\_name = e.department  WHERE e.emp\_id IS NULL;     1. 20. Count employees in each department.   Query:SELECT department, COUNT(\*) AS employee\_count  FROM employees  GROUP BY department;     1. 21. Find employees earning above average salary.   Query:SELECT \*  FROM employees  WHERE salary > (SELECT AVG(salary) FROM employees);     1. 22. Find the department with the highest average salary.   Query:SELECT department, AVG(salary) AS avg\_salary  FROM employees  GROUP BY department  ORDER BY avg\_salary DESC  LIMIT 1;     1. 23. Find employees hired most recently.   Query:SELECT \*  FROM employees  ORDER BY hire\_date DESC  LIMIT 1;     1. 24. Find employees earning the second highest salary.   Query:SELECT \*  FROM employees  WHERE salary = (      SELECT MAX(salary)      FROM employees      WHERE salary < (SELECT MAX(salary) FROM employees)  );     1. 25. Find all employees in the same department as 'Amit Sharma'.   Query:SELECT \*  FROM employees  WHERE department = (SELECT department FROM employees WHERE first\_name = 'Amit' AND last\_name = 'Sharma');     1. 26. Increase salary by 10% for IT employees.   Query:UPDATE employees  SET salary = salary \* 1.10  WHERE department = 'IT';   1. 27. Change department of employee 'Ravi' to Marketing.   Query:UPDATE employees  SET department = 'Marketing'  WHERE first\_name = 'Ravi';   1. 28. Delete employees with salary below 40000.   Query:DELETE FROM employees  WHERE salary < 40000;   1. 29. Add a new column 'email' to employees.   Query:ALTER TABLE employees  ADD email VARCHAR(100);   1. 30. Update email IDs for all employees.   Query:UPDATE employees  SET email = CONCAT(LOWER(first\_name), '.', LOWER(last\_name), '@example.com');   1. 31. Find top 2 departments by average salary.   Query:SELECT department, AVG(salary) AS avg\_salary  FROM employees  GROUP BY department  ORDER BY avg\_salary DESC  LIMIT 2;     1. 32. Find how many employees work in each city.   Query:SELECT d.location, COUNT(e.emp\_id) AS employee\_count  FROM employees e  JOIN departments d ON e.department = d.dept\_name  GROUP BY d.location;     1. 33. Show employee count and total salary together.   Query:SELECT COUNT(\*) AS employee\_count, SUM(salary) AS total\_salary  FROM employees;     1. 34. Display employees with names starting with 'A'.   Query:SELECT \*  FROM employees  WHERE first\_name LIKE 'A%';     1. 35. Display employees whose last name ends with 'a'.   Query:SELECT \*  FROM employees  WHERE last\_name LIKE '%a';     1. 36. Find employees hired in 2020.   Query:SELECT \*  FROM employees  WHERE YEAR(hire\_date) = 2020;     1. 37. Show number of days since each employee was hired.   Query:SELECT first\_name, last\_name, DATEDIFF(CURDATE(), hire\_date) AS days\_since\_hire  FROM employees;     1. 38. Display employee names in uppercase.   Query:SELECT UPPER(first\_name) AS first\_name, UPPER(last\_name) AS last\_name  FROM employees;     1. 39. Concatenate first and last names.   Query:SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name  FROM employees;     1. 40. Find employees whose salary is between 45000 and 60000.   Query:  SELECT \*  FROM employees  WHERE salary BETWEEN 45000 AND 60000;     1. 41. Create a view for high salary employees (>55000).   Query:CREATE VIEW high\_salary\_employees AS  SELECT \* FROM employees  WHERE salary > 55000;   1. 42. Display all records from the view.   Query:SELECT \* FROM high\_salary\_employees;     1. 43. Add NOT NULL constraint to department name.   Query:ALTER TABLE employees  MODIFY department VARCHAR(50) NOT NULL;   1. 44. Drop the view.   Query:DROP VIEW high\_salary\_employees;   1. 45. Rename the employees table to staff.   Query:ALTER TABLE employees RENAME TO staff;   1. 46. Create a backup copy of the employees table.   Query:CREATE TABLE employees\_backup AS SELECT \* FROM employees;   1. 47. Delete all data but keep the structure.   Query:DELETE FROM employees;   1. 48. Drop the employees backup table.   Query:DROP TABLE employees\_backup;   1. 49. Create an index on employee last name.   Query:CREATE INDEX idx\_last\_name ON employees(last\_name);   1. 50. Drop the index.   Query:  DROP INDEX idx\_last\_name ON employees; | | | | | | Week9 - Monday |  |

Employee Table



Department Table

