LABASSIGNMENT-6

Name:- Bibek Chand Sah

Roll:- 22054029

Section:- CSE-05

1. Create and populate the following table ‘EMP06’. Make Emp_no the primary key and F_name not null.

Emp_no	F_name	L_name	Salary	Dept_no
101	Jai		90000	1
102	Viru		80000	1
103	Gabbar	Singh	70000	2
104	Basanti		60000	3
105	Ram	Lal	50000	3
106	Radha	Thakur	30000	3

➔ CREATE TABLE EMP06 (Emp_no INT PRIMARY KEY, F_name VARCHAR(50) NOT NULL, L_name VARCHAR(50), Salary DECIMAL(10, 2), Dept_no INT);

INSERT INTO EMP06 VALUES (&Emp_no, ‘&F_name’, ‘&L_name’, &Salary, &Dept_no);

2. Create and populate the following table ‘PROJECT’. Make P_no the primary key and put a default value constraint on P_Loc with value=‘Mumbai’.

P_no	P_name	P_Loc
1	XYZ	Pune
2	ABC	Pune
3	IJK	

➔ CREATE TABLE PROJECT (P_no INT PRIMARY KEY, P_name VARCHAR(50), P_Loc VARCHAR(50) DEFAULT 'Mumbai');

INSERT INTO PROJECT VALUES (&P_no, ‘&P_name’, ‘&P_Loc’);

3. Create and populate the following EMP_PROJ table. Make(Emp_no, P_no) the primary key.

Emp_no	P_no
101	1
102	1
103	2
104	2
101	2
105	2

➔ CREATE TABLE EMP_PROJ (Emp_no INT, P_no INT, PRIMARY KEY (Emp_no, P_no));

INSERT INTO EMP_PROJ VALUES (&Emp_no, &P_no);

4. Display the employee's first names with the project name's they are working on.

➔ SELECT e.first_name, p.project_name FROM Employee e JOIN Project_Assignment pa ON e.employee_id = pa.employee_id JOIN Project p ON pa.project_id = p.project_id;

5. In which city Gabbar Singh works.

➔ SELECT e.city FROM Employee e WHERE e.first_name = 'Gabbar' AND e.last_name = 'Singh';

6. Find the employee names who are not yet assigned to any project (using minus).

➔ SELECT first_name, last_name FROM Employee MINUS SELECT e.first_name, e.last_name FROM Employee e JOIN Project_Assignment pa ON e.employee_id = pa.employee_id;

7. Find the employee names who are not yet assigned to any project (using outer join).

➔ SELECT e.first_name, e.last_name FROM Employee e LEFT JOIN Project_Assignment pa ON e.employee_id = pa.employee_id WHERE pa.employee_id IS NULL;

8. Find the project names where no employees are working (using outer join).

➔ SELECT p.project_name FROM Project p LEFT JOIN Project_Assignment pa ON p.project_id = pa.project_id WHERE pa.project_id IS NULL;

9. Find all the employee names who are working in project number 1 and project 'ABC' (using union).

➔ (SELECT e.first_name, e.last_name FROM Employee e JOIN Project_Assignment pa ON e.employee_id = pa.employee_id JOIN Project p ON pa.project_id = p.project_id WHERE p.project_id = 1) UNION (SELECT e.first_name, e.last_name FROM Employee e JOIN Project_Assignment pa ON e.employee_id = pa.employee_id JOIN Project p ON pa.project_id = p.project_id WHERE p.project_name = 'ABC');

10. Find all the employee names who are working in both project number 1 and project number 2 (using intersect).

➔ (SELECT e.first_name, e.last_name FROM Employee e JOIN Project_Assignment pa ON e.employee_id = pa.employee_id WHERE pa.project_id = 1) INTERSECT (SELECT e.first_name, e.last_name FROM Employee e JOIN Project_Assignment pa ON e.employee_id = pa.employee_id WHERE pa.project_id = 2);

11. Find the number of employees working in each project.

➔ SELECT p.project_name, COUNT(pa.employee_id) AS num_employees FROM Project p JOIN Project_Assignment pa ON p.project_id = pa.project_id GROUP BY p.project_name;

12. Find the average salary of each department.

➔ SELECT d.dept_name, AVG(e.salary) AS avg_salary FROM Department d JOIN Employee e ON d.dept_id = e.dept_id GROUP BY d.dept_name;

13. Find the department number with the number of employees working in each department where the average salary is greater than 60000 and number of employees greater than 1.

➔ SELECT d.dept_id, COUNT(e.employee_id) AS num_employees FROM Department d JOIN Employee e ON d.dept_id = e.dept_id GROUP BY d.dept_id HAVING AVG(e.salary) > 60000 AND COUNT(e.employee_id) > 1;

14. Find all the employees who earn more than Basanti.

➔ SELECT e.first_name, e.last_name FROM Employee e WHERE e.salary > (SELECT salary FROM Employee WHERE first_name = 'Basanti');

15. Find all the employees who earn more than the average salary of all employees.

➔ SELECT e.first_name, e.last_name FROM Employee e WHERE e.salary > (SELECT AVG(salary) FROM Employee);

16. Find the employee who earns the highest salary.

➔ SELECT first_name, last_name FROM Employee WHERE salary = (SELECT MAX(salary) FROM Employee);

17. Find the employee who earns the highest salary in dept_no 3.

➔ SELECT first_name, last_name FROM Employee WHERE salary = (SELECT MAX(salary) FROM Employee WHERE dept_id = 3);

18. Find the employee earning the second highest salary.

➔ SELECT first_name, last_name FROM Employee WHERE salary = (SELECT MAX(salary) FROM Employee WHERE salary < (SELECT MAX(salary) FROM Employee));

19. Find the dept_no having the highest average salary.

➔ SELECT dept_id FROM Employee GROUP BY dept_id HAVING AVG(salary) = (SELECT MAX(avg_salary) FROM (SELECT AVG(salary) AS avg_salary FROM Employee GROUP BY dept_id));

20. Find the employee with the third highest salary among all the employees

➔ SELECT first_name, last_name FROM Employee WHERE salary = (SELECT DISTINCT salary FROM Employee ORDER BY salary DESC LIMIT 1 OFFSET 2);