**SQL PROJECT 1**

1. Get all employee details from “EmployeeDetail” table.

SELECT \* FROM EmployeeDetail;

2. Get only “FirstName” column from “EmployeeDetail” table.

SELECT FirstName FROM EmployeeDetail;

3. Get “FirstName” in upper case as “First Name”.

SELECT UPPER(FirstName) AS "First Name" FROM EmployeeDetail;

4. Get “FirstName” in upper case as “First Name”.

SELECT CONCAT(UPPER(FirstName), ' ') AS "First Name" FROM EmployeeDetail;

5. Combine FirstName and LastName and display it as “Name” (with whitespace).

SELECT CONCAT(FirstName, ' ', LastName) AS "Name" FROM EmployeeDetail;

6. Select employee detail whose name is “Vikas”.

SELECT \* FROM EmployeeDetail WHERE FirstName = 'Vikas';

7. Get all employee details whose “FirstName” starts with the letter ‘a’.

SELECT \* FROM EmployeeDetail WHERE FirstName LIKE 'a%';

8. Get all employee details whose “FirstName” starts with the letter ‘a’.

SELECT \* FROM EmployeeDetail WHERE FirstName LIKE 'a%';

9. Get all employee details whose “FirstName” ends with ‘h’.

SELECT \* FROM EmployeeDetail WHERE FirstName LIKE '%h';

10. Get all employee details whose “FirstName” starts with any single character between ‘a-p’.

SELECT \* FROM EmployeeDetail WHERE FirstName LIKE '[a-p]%';

11. Get all employee details whose “FirstName” does not start with any single character between ‘a-p’.

SELECT \* FROM EmployeeDetail WHERE FirstName NOT LIKE '[a-p]%';

12. Get all employee details whose “Gender” ends with ‘le’ and contains 4 letters.

SELECT \* FROM EmployeeDetail WHERE Gender LIKE '\_le' AND LENGTH(Gender) = 4;

13. Get all employee details whose “FirstName” starts with ‘A’ and contains 5 letters.

SELECT \* FROM EmployeeDetail WHERE FirstName LIKE 'A\_\_\_\_';

14. Get all employee details whose “FirstName” contains ‘%’.

SELECT \* FROM EmployeeDetail WHERE FirstName LIKE '%[%]%';

15. Get all unique “Department” from EmployeeDetail table.

SELECT DISTINCT Department FROM EmployeeDetail;

16. Get the highest “Salary” from EmployeeDetail table.

SELECT MAX(Salary) AS HighestSalary FROM EmployeeDetail;

17. Get the lowest “Salary” from EmployeeDetail table.

SELECT MIN(Salary) AS LowestSalary FROM EmployeeDetail;

18. Show “JoiningDate” in “dd mmm yyyy” format.

SELECT DATE\_FORMAT(JoiningDate, '%d %b %Y') AS FormattedJoiningDate FROM EmployeeDetail;

19. Show “JoiningDate” in “yyyy/mm/dd” format.

SELECT DATE\_FORMAT(JoiningDate, '%Y/%m/%d') AS FormattedJoiningDate FROM EmployeeDetail;

20. Show only the time part of the “JoiningDate”.

SELECT DATE\_FORMAT(JoiningDate, '%H:%i:%s') AS TimePart FROM EmployeeDetail;

21. Get only the Year part of “JoiningDate”.

SELECT YEAR(JoiningDate) AS JoiningYear FROM EmployeeDetail;

22. Get only the Month part of “JoiningDate”.

SELECT MONTH(JoiningDate) AS JoiningMonth FROM EmployeeDetail;

23. Get the system date.

SELECT CURDATE() AS SystemDate;

24. Get UTC date.

SELECT UTC\_DATE() AS UTCDate;

25. Get the first name, current date, joining date, and the difference between the current date and joining date in months.

SELECT

FirstName,

CURDATE() AS CurrentDate,

JoiningDate,

TIMESTAMPDIFF(MONTH, JoiningDate, CURDATE()) AS MonthsSinceJoining

FROM EmployeeDetail;

26. Get the first name, current date, joining date, and the difference between the current date and joining date in days.

SELECT

FirstName,

CURDATE() AS CurrentDate,

JoiningDate,

DATEDIFF(CURDATE(), JoiningDate) AS DaysSinceJoining

FROM EmployeeDetail;

27. Get all employee details whose joining year is 2013.

SELECT \* FROM EmployeeDetail WHERE YEAR(JoiningDate) = 2013;

28. Get all employee details whose joining month is Jan(1).

SELECT \* FROM EmployeeDetail WHERE MONTH(JoiningDate) = 1;

29. Get how many employees exist in “EmployeeDetail” table.

SELECT COUNT(\*) AS EmployeeCount FROM EmployeeDetail;

31. Select only one/top 1 record from “EmployeeDetail” table.

SELECT \* FROM EmployeeDetail LIMIT 1;

32. Select all employee details with the first name “Vikas”, “Ashish”, and “Nikhil”.

SELECT \* FROM EmployeeDetail WHERE FirstName IN ('Vikas', 'Ashish', 'Nikhil');

33. Select all employee details with the first name not in “Vikas”, “Ashish”, and “Nikhil”.

SELECT \* FROM EmployeeDetail WHERE FirstName NOT IN ('Vikas', 'Ashish', 'Nikhil');

34. Select the first name from “EmployeeDetail” table after removing white spaces from the right side.

SELECT RTRIM(FirstName) AS FirstNameWithoutRightSpaces FROM EmployeeDetail;

35. Select the first name from “EmployeeDetail” table after removing white spaces from the left side.

SELECT LTRIM(FirstName) AS FirstNameWithoutLeftSpaces FROM EmployeeDetail;

36. Display the first name and Gender as M/F (if male then M, if Female then F).

SELECT FirstName,

CASE WHEN Gender = 'Male' THEN 'M'

WHEN Gender = 'Female' THEN 'F'

END AS GenderShort

FROM EmployeeDetail;

37. Select the first name from “EmployeeDetail” table prefixed with “Hello”.

SELECT CONCAT('Hello ', FirstName) AS GreetedName FROM EmployeeDetail;

38. Get employee details from “EmployeeDetail” table whose Salary is greater than 600000.

SELECT \* FROM EmployeeDetail WHERE Salary > 600000;

39. Get employee details from “EmployeeDetail” table whose Salary is less than 700000.

SELECT \* FROM EmployeeDetail WHERE Salary < 700000;

40. Get employee details from “EmployeeDetail” table whose Salary is between 500000 and 600000.

SELECT \* FROM EmployeeDetail WHERE Salary BETWEEN 500000 AND 600000;

41. Give records of the “ProjectDetail” table.

SELECT \* FROM ProjectDetail;

42. Write the query to get the department and department-wise total (sum) salary from the “EmployeeDetail” table.

SELECT Department, SUM(Salary) AS TotalSalary FROM EmployeeDetail GROUP BY Department;

43. Write the query to get the department and department-wise total (sum) salary, display it in ascending order according to salary.

SELECT Department, SUM(Salary) AS TotalSalary FROM EmployeeDetail GROUP BY Department ORDER BY TotalSalary ASC;

44. Write the query to get the department and department-wise total (sum) salary, display it in descending order according to salary.

SELECT Department, SUM(Salary) AS TotalSalary FROM EmployeeDetail GROUP BY Department ORDER BY TotalSalary DESC;

45. Write the query to get the department, the total number of departments, total (sum) salary with respect to the department from the “EmployeeDetail” table.

SELECT Department, COUNT(DISTINCT Department) AS TotalDepartments, SUM(Salary) AS TotalSalary FROM EmployeeDetail GROUP BY Department;

46. Get department-wise average salary from the “EmployeeDetail” table ordered by salary ascending.

SELECT Department, AVG(Salary) AS AverageSalary FROM EmployeeDetail GROUP BY Department ORDER BY AverageSalary ASC;

47. Get department-wise maximum salary from the “EmployeeDetail” table ordered by salary ascending.

SELECT Department, MAX(Salary) AS MaxSalary FROM EmployeeDetail GROUP BY Department ORDER BY MaxSalary ASC;

48. Get department-wise minimum salary from the “EmployeeDetail” table ordered by salary ascending.

SELECT Department, MIN(Salary) AS MinSalary FROM EmployeeDetail GROUP BY Department ORDER BY MinSalary ASC;

49. Get department-wise minimum salary from the “EmployeeDetail” table ordered by salary ascending.

SELECT Department, MIN(Salary) AS MinSalary FROM EmployeeDetail GROUP BY Department ORDER BY MinSalary ASC;

50. Join both tables, “EmployeeDetail” and “ProjectDetail,” based on some common parameter.

SELECT \* FROM EmployeeDetail INNER JOIN ProjectDetail ON EmployeeDetail.EmployeeID = ProjectDetail.EmployeeDetailID;

51. Get employee name, project name ordered by the first name from “EmployeeDetail” and “ProjectDetail” for those employees who have assigned projects already.

SELECT EmployeeDetail.FirstName, ProjectDetail.ProjectName FROM EmployeeDetail INNER JOIN ProjectDetail ON EmployeeDetail.EmployeeID = ProjectDetail.EmployeeDetailID;

52. Get employee name, project name ordered by the first name from “EmployeeDetail” and “ProjectDetail” for all employees even if they have not assigned a project.

SELECT EmployeeDetail.FirstName, ProjectDetail.ProjectName FROM EmployeeDetail LEFT JOIN ProjectDetail ON EmployeeDetail.EmployeeID = ProjectDetail.EmployeeDetailID;

53. Get employee name, project name ordered by the first name from “EmployeeDetail” and “ProjectDetail” for all employees. If a project is not assigned, then display “-No Project Assigned.”

SELECT EmployeeDetail.FirstName, COALESCE(ProjectDetail.ProjectName, '-No Project Assigned') AS ProjectName

FROM EmployeeDetail

LEFT JOIN ProjectDetail ON EmployeeDetail.EmployeeID = ProjectDetail.EmployeeDetailID;

54. Get all project names even if they have not matched any employee ID, in the left table, ordered by the first name from “EmployeeDetail” and “ProjectDetail.”

SELECT EmployeeDetail.FirstName, COALESCE(ProjectDetail.ProjectName, '-No Project Assigned') AS ProjectName

FROM EmployeeDetail

RIGHT JOIN ProjectDetail ON EmployeeDetail.EmployeeID = ProjectDetail.EmployeeDetailID;

55. Get complete records (employee name, project name) from both tables ([EmployeeDetail], [ProjectDetail]). If no match is found in any table, then show NULL.

SELECT EmployeeDetail.FirstName, ProjectDetail.ProjectName

FROM EmployeeDetail

FULL OUTER JOIN ProjectDetail ON EmployeeDetail.EmployeeID = ProjectDetail.EmployeeDetailID;

56. Get complete records (employee name, project name) from both tables ([EmployeeDetail], [ProjectDetail]). If no match is found in any table, then show NULL.

SELECT EmployeeDetail.FirstName, ProjectDetail.ProjectName

FROM EmployeeDetail

FULL OUTER JOIN ProjectDetail ON EmployeeDetail.EmployeeID = ProjectDetail.EmployeeDetailID;

57. Get complete records (employee name, project name) from both tables ([EmployeeDetail], [ProjectDetail]). If no match is found in any table, then show NULL.

SELECT EmployeeDetail.FirstName, ProjectDetail.ProjectName

FROM EmployeeDetail

FULL OUTER JOIN ProjectDetail ON EmployeeDetail.EmployeeID = ProjectDetail.EmployeeDetailID;

58. Write down the query to fetch EmployeeName & Project who has assigned more than one project.

SELECT EmployeeDetail.FirstName, ProjectDetail.ProjectName

FROM EmployeeDetail

JOIN ProjectDetail ON EmployeeDetail.EmployeeID = ProjectDetail.EmployeeDetailID

GROUP BY EmployeeDetail.FirstName, ProjectDetail.ProjectName

HAVING COUNT(\*) > 1;

59. Write down the query to fetch ProjectName on which more than one employee is working along with EmployeeName.

SELECT ProjectDetail.ProjectName, EmployeeDetail.FirstName

FROM ProjectDetail

JOIN EmployeeDetail ON EmployeeDetail.EmployeeID = ProjectDetail.EmployeeDetailID

GROUP BY ProjectDetail.ProjectName, EmployeeDetail.FirstName

HAVING COUNT(\*) > 1;

60. Apply Cross Join in Both the tables.

SELECT \* FROM EmployeeDetail CROSS JOIN ProjectDetail;