Sql commands for tests

**Ques.1. Write an SQL query to fetch the EmpId and FullName of all the employees working under Manager with id – ‘986’.**  
SELECT EmpId, FullName

FROM EmployeeDetails

WHERE ManagerId = 986;

**Ques.2. Write an SQL query to fetch the different projects available from the EmployeeSalary table.**

SELECT DISTINCT(Project)

FROM EmployeeSalary;

**Ques.3. Write an SQL query to fetch the count of employees working in project ‘P1’.**

SELECT COUNT(\*)

FROM EmployeeSalary

WHERE Project = 'P1';

**Ques.4. Write an SQL query to find the maximum, minimum, and average salary of the employees.**

SELECT Max(Salary),

Min(Salary),

AVG(Salary)

FROM EmployeeSalary;

**Ques.5. Write an SQL query to find the employee id whose salary lies in the range of 9000 and 15000.**

SELECT EmpId, Salary

FROM EmployeeSalary

WHERE Salary BETWEEN 9000 AND 15000;

**Ques.6. Write an SQL query to fetch those employees who live in Toronto and work under manager with ManagerId – 321.**

SELECT EmpId, City, ManagerId

FROM EmployeeDetails

WHERE City='Toronto' AND ManagerId='321';

**Ques.7. Write an SQL query to** f**etch all the employees who either live in California or work under a manager with ManagerId – 321.**

SELECT EmpId, City, ManagerId

FROM EmployeeDetails

WHERE City='California' OR ManagerId='321';

**Ques.8. Write an SQL query to fetch all those employees who work on Project other than P1.**

SELECT EmpId

FROM EmployeeSalary

WHERE NOT Project='P1';

or

SELECT EmpId

FROM EmployeeSalary

WHERE Project <> 'P1';

**Ques.9. Write an SQL query to display the total salary of each employee adding the Salary with Variable value.**

SELECT EmpId,

Salary+Variable as TotalSalary

FROM EmployeeSalary;

**Ques.10. Write an SQL query to fetch the employees whose name begins with any two characters, followed by a text “hn” and ending with any sequence of characters.**

SELECT FullName

FROM EmployeeDetails

WHERE FullName LIKE ‘\_\_hn%’;

**Ques.11. Write an SQL query to fetch all the EmpIds which are present in either of the tables – ‘EmployeeDetails’ and ‘EmployeeSalary’.**

SELECT EmpId FROM EmployeeDetails

UNION

SELECT EmpId FROM EmployeeSalary;

**Ques.12. Write an SQL query to fetch common records between two tables.**

SELECT \* FROM EmployeeSalary

INTERSECT

SELECT \* FROM ManagerSalary;

MySQL – Since MySQL doesn’t have INTERSECT operator so we can use the sub query-

SELECT \*

FROM EmployeeSalary

WHERE EmpId IN

(SELECT EmpId from ManagerSalary);

**Ques.13. Write an SQL query to fetch records that are present in one table but not in another table.**

SELECT \* FROM EmployeeSalary

MINUS

SELECT \* FROM ManagerSalary;

SELECT EmployeeSalary.\*

FROM EmployeeSalary

LEFT JOIN

ManagerSalary USING (EmpId)

WHERE ManagerSalary.EmpId IS NULL;

**Ques.14. Write an SQL query to fetch the EmpIds that are present in both the tables –   ‘EmployeeDetails’ and ‘EmployeeSalary.**

SELECT EmpId FROM

EmployeeDetails

where EmpId IN

(SELECT EmpId FROM EmployeeSalary);

**Ques.15. Write an SQL query to fetch the EmpIds that are present in EmployeeDetails but not in EmployeeSalary.**

SELECT EmpId FROM

EmployeeDetails

where EmpId Not IN

(SELECT EmpId FROM EmployeeSalary);

**Ques.16. Write an SQL query to fetch the employee full names and replace the space with ‘-’.**

SELECT REPLACE(FullName, ' ', '-')

FROM EmployeeDetails;

**Ques.17. Write an SQL query to fetch the position of a given character(s) in a field.**

SELECT INSTR(FullName, 'Snow')

FROM EmployeeDetails;

**Ques.18. Write an SQL query to display both the EmpId and ManagerId together.**

SELECT CONCAT(EmpId, ManagerId) as NewId

FROM EmployeeDetails;

**Ques.19. Write a query to fetch only the first name(string before space) from the FullName column of the EmployeeDetails table.**

SELECT MID(FullName, 1, LOCATE(' ',FullName))

FROM EmployeeDetails;

SELECT SUBSTRING(FullName, 1, CHARINDEX(' ',FullName))

FROM EmployeeDetails;

**Ques.20. Write an SQL query to upper case the name of the employee and lower case the city values.**

SELECT UPPER(FullName), LOWER(City)

FROM EmployeeDetails;

**Ques.21. Write an SQL query to find the count of the total occurrences of a particular character – ‘n’ in the FullName field.**

SELECT FullName,

LENGTH(FullName) - LENGTH(REPLACE(FullName, 'n', ''))

FROM EmployeeDetails;

**Ques.22. Write an SQL query to update the employee names by removing leading and trailing spaces.**

UPDATE EmployeeDetails

SET FullName = LTRIM(RTRIM(FullName));

**Ques.23. Fetch all the employees who are not working on any project.**

SELECT EmpId

FROM EmployeeSalary

WHERE Project IS NULL;

**Ques.24. Write an SQL query to fetch employee names having a salary greater than or equal to 5000 and less than or equal to 10000.**

SELECT FullName

FROM EmployeeDetails

WHERE EmpId IN

(SELECT EmpId FROM EmployeeSalary

WHERE Salary BETWEEN 5000 AND 10000);

**Ques.25. Write an SQL query to find the current date-time.**

. MySQL-

SELECT NOW();

SQL Server-

SELECT getdate();

Oracle-

SELECT SYSDATE FROM DUAL;

**Ques.26. Write an SQL query to fetch all the Employees details from EmployeeDetails table who joined in the Year 2020.**

SELECT \* FROM EmployeeDetails

WHERE DateOfJoining BETWEEN '2020/01/01'

AND '2020/12/31';

**Ques.27. Write an SQL query to fetch all employee records from EmployeeDetails table who have a salary record in EmployeeSalary table.**

SELECT \* FROM EmployeeDetails E

WHERE EXISTS

(SELECT \* FROM EmployeeSalary S

WHERE E.EmpId = S.EmpId);

**Ques.28. Write an SQL query to fetch project-wise count of employees sorted by project’s count in descending order.**

SELECT Project, count(EmpId) EmpProjectCount

FROM EmployeeSalary

GROUP BY Project

ORDER BY EmpProjectCount DESC;

**Ques.29. Write a query to fetch employee names and salary records. Display the employee details even if the salary record is not present for the employee.**

SELECT E.FullName, S.Salary

FROM EmployeeDetails E

LEFT JOIN

EmployeeSalary S

ON E.EmpId = S.EmpId;

**Ques.30. Write an SQL query to join 3 tables.**

SELECT column1, column2

FROM TableA

JOIN TableB ON TableA.Column3 = TableB.Column3

JOIN TableC ON TableA.Column4 = TableC.Column4;

**Ques. 31. Write an SQL query to fetch all the Employees who are also managers from the EmployeeDetails table.**

SELECT DISTINCT E.FullName

FROM EmployeeDetails E

INNER JOIN EmployeeDetails M

ON E.EmpID = M.ManagerID;

**Ques.32. Write an SQL query to fetch duplicate records from EmployeeDetails (without considering the primary key – EmpId).**

SELECT FullName, ManagerId, DateOfJoining, City, COUNT(\*)

FROM EmployeeDetails

GROUP BY FullName, ManagerId, DateOfJoining, City

HAVING COUNT(\*) > 1;

**Ques.33. Write an SQL query to remove duplicates from a table without using a temporary table.**

DELETE E1 FROM EmployeeDetails E1

INNER JOIN EmployeeDetails E2

WHERE E1.EmpId > E2.EmpId

AND E1.FullName = E2.FullName

AND E1.ManagerId = E2.ManagerId

AND E1.DateOfJoining = E2.DateOfJoining

AND E1.City = E2.City;

**Ques.34. Write an SQL query to fetch only odd rows from the table.**

SELECT \* FROM EmployeeDetails

WHERE MOD (EmpId, 2) <> 0;

**Ques.35. Write an SQL query to fetch only even rows from the table.**

SELECT \* FROM EmployeeDetails

WHERE MOD (EmpId, 2) = 0;

**Ques.36. Write an SQL query to create a new table with data and structure copied from another table.**

CREATE TABLE NewTable

SELECT \* FROM EmployeeSalary;

**Ques.37. Write an SQL query to create an empty table with the same structure as some other table.**

CREATE TABLE NewTable

SELECT \* FROM EmployeeSalary where 1=0;

**Ques.38. Write an SQL query to fetch top n records?**

SELECT \*

FROM EmployeeSalary

ORDER BY Salary DESC LIMIT N;

Or

SELECT TOP N \*

FROM EmployeeSalary

ORDER BY Salary DESC;

**Ques.39. Write an SQL query to find the nth highest salary from table.**

Ans, Using Top keyword (SQL Server)-

SELECT TOP 1 Salary

FROM (

SELECT DISTINCT TOP N Salary

FROM Employee

ORDER BY Salary DESC

)

ORDER BY Salary ASC;

Using limit clause(MySQL)-

SELECT Salary

FROM Employee

ORDER BY Salary DESC LIMIT N-1,1;

**Ques.40. Write SQL query to find the 3rd highest salary from a table without using the TOP/limit keyword.**

In order to find the 3rd highest salary

SELECT Salary

FROM EmployeeSalary Emp1

WHERE 2 = (

SELECT COUNT( DISTINCT ( Emp2.Salary ) )

FROM EmployeeSalary Emp2

WHERE Emp2.Salary > Emp1.Salary

)

For nth highest salary-

SELECT Salary

FROM EmployeeSalary Emp1

WHERE N-1 = (

SELECT COUNT( DISTINCT ( Emp2.Salary ) )

FROM EmployeeSalary Emp2

WHERE Emp2.Salary > Emp1.Salary

)