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

MySQL – using MID

```
SELECT MID(FullName, 1, LOCATE(' ',FullName))  FROM EmployeeDetails;
```

SQL Server – using SUBSTRING

```
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();
```

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';
```

Or

```
SELECT * FROM EmployeeDetails WHERE YEAR(DateOfJoining) = '2020';
```

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?

In MySQL using LIMIT-

```
SELECT * FROM EmployeeSalary ORDER BY Salary DESC LIMIT N;
```

In SQL server using TOP command-

```
SELECT TOP N * FROM EmployeeSalary ORDER BY Salary DESC;
```

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

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.

```
SELECT Salary FROM EmployeeSalary Emp1 WHERE N-1 = ( SELECT COUNT( DISTINCT
( Emp2.Salary ) ) FROM EmployeeSalary Emp2 WHERE Emp2.Salary > Emp1.Salary);
```