Table – EmployeeDetails

EmpId	FullName	ManagerId	DateOfJoining	City
121	John Snow	321	01/31/2014	Toronto
321	Walter White	986	01/30/2015	California
421	Kuldeep Rana	876	27/11/2016	New D

Table – EmployeeSalary

EmpId	Project	Salary	Variable
121	**P1**	**8000**	**500**
321	**P2**	**10000**	**1000**
421	**P1**	**12000**	**0**

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 fetch 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 EmployeeDetails INTERSECT SELECT * FROM ManagerSalary;
```

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

```
SELECT * FROM EmployeeDetails 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;
```

MySQL – Since MySQL doesn't have MINUS operator so we can use LEFT join-

```
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.

Using sub query-

```
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.

Using sub query-

```
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;
```