PART-1

1. **SQL Query to fetch records that are present in one table but not in another table.**

SYNTAX:

SELECT ED.\*,ES.\* FROM EMPLOYEEDETAILS AS ED LEFT JOIN EMPLOYEESALARY AS ES ON ED.EMPID=ES.EMPID WHERE ES.EMPID IS NULL;

SELECT \* FROM EMPLOYEEDETAILS WHERE NOT EXISTS (SELECT \* FROM EMPLOYEESALARY WHERE [dbo].[EMPLOYEESALARY].EMPID=[dbo].[EMPLOYEEDETAILS].EMPID);

1. **SQL query to fetch all the employees who are not working on any project.**

SYNTAX:

SELECT ED.\*,ES.\* FROM EMPLOYEEDETAILS AS ED LEFT JOIN EMPLOYEESALARY AS ES ON ED.EMPID=ES.EMPID WHERE ES.EMPID IS NULL;

SELECT \* FROM EMPLOYEEDETAILS WHERE NOT EXISTS (SELECT \* FROM EMPLOYEESALARY WHERE [dbo].[EMPLOYEESALARY].EMPID=[dbo].[EMPLOYEEDETAILS].EMPID);

1. **SQL query to fetch all the Employees from EmployeeDetails who joined in the Year 2020.**

SYNTAX:

SELECT \* FROM EMPLOYEEDETAILS WHERE YEAR(DATEOFJOINING)=2020;

1. **Fetch all employees from EmployeeDetails who have a salary record in EmployeeSalary.**

SYNTAX:

SELECT ED.\*,ES.\* FROM EMPLOYEEDETAILS AS ED JOIN EMPLOYEESALARY AS ES ON ED.EMPID=ES.EMPID;

1. **Write an SQL query to fetch a project-wise count of employees.**

SYNTAX:

SELECT COUNT(EMPID) AS EMP\_COUNT,PROJECT FROM EMPLOYEESALARY GROUP BY PROJECT;

1. **Fetch employee names and salaries even if the salary value is not present for the employee.**

SYNTAX:

SELECT ED.\*,ES.\* FROM EMPLOYEEDETAILS AS ED LEFT JOIN EMPLOYEESALARY AS ES ON ED.EMPID=ES.EMPID;

1. **Write an SQL query to fetch all the Employees who are also managers.**

SYNTAX:

SELECT E1.\* FROM EMPLOYEEDETAILS E1, EMPLOYEEDETAILS E2 WHERE E1.EMPID=E2.MANAGERID;

1. **Write an SQL query to fetch duplicate records from EmployeeDetails.**

SYNTAX:

SELECT MANAGERID,COUNT(\*) FROM EMPLOYEEDETAILS GROUP BY MANAGERID HAVING COUNT(\*)>1;

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

SYNTAX:

SELECT \* FROM (SELECT \*,ROW\_NUMBER() OVER( ORDER BY EMPID ASC) AS ROW\_NUM FROM EMPLOYEEDETAILS) AS A WHERE ROW\_NUM %2<>0;

1. **Write a query to find the 3rd highest salary from a table without top or limit keyword.**

SYNTAX:

SELECT \* FROM (SELECT \*, DENSE\_RANK() OVER(ORDER BY SALARY DESC) AS SAL FROM EMPLOYEESALARY) AS ES WHERE SAL=3;

Part-2

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

Syntax:

SELECT EMPID,FULLNAME FROM EMPLOYEEDETAILS WHERE MANAGERID=986;

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

SYNTAX:

SELECT DISTINCT(PROJECT) FROM EMPLOYEESALARY;

1. **Write an SQL query to fetch the count of employees working in project P1.**

SYNTAX:

SELECT COUNT(EMPID) AS COUNT\_EMP FROM EMPLOYEESALARY WHERE PROJECT='P1';

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

SYNTAX:

SELECT MAX(SALARY) AS MAX\_SALARY,MIN(SALARY) AS MIN\_SALARY, AVG(SALARY) AS AVG\_SALARY FROM EMPLOYEESALARY;

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

SYNTAX:

SELECT EMPID FROM EMPLOYEESALARY WHERE SALARY BETWEEN 9000 AND 15000;

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

SYNTAX:

SELECT EMPID, FULLNAME FROM EMPLOYEEDETAILS WHERE MANAGERID=321 AND CITY='TORONTO';

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

SYNTAX:

SELECT \* FROM EMPLOYEEDETAILS WHERE CITY='California' or MANAGERID=321;

1. **Write an SQL query to fetch all those employees who work on Projects other than P1.**

SYNTAX:

SELECT \* FROM EMPLOYEESALARY WHERE PROJECT!='P1';

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

SYNTAX:

SELECT EMPID,(SALARY+VARIABLE)AS TOTAL\_SALARY FROM EMPLOYEESALARY;

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

SYNTAX:

SELECT \* FROM EMPLOYEEDETAILS WHERE FULLNAME LIKE '\_\_hn%';

Part – 3

1. **Write an SQL query to fetch all the EmpIds which are present in either of the tables – "EmployeeDetails" and "EmployeeSalary".**

Syntax:

select ed.empid from EMPLOYEEDETAILS ed full outer join EMPLOYEESALARY es on ed.EMPID=es.EMPID;

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

Syntax:

select ed.empid from EMPLOYEEDETAILS ed inner join EMPLOYEESALARY es on ed.EMPID=es.EMPID;

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

Syntax:

select ed.empid from EMPLOYEEDETAILS ed left join EMPLOYEESALARY es on ed.EMPID=es.EMPID;

1. **Write an SQL query to fetch the EmpIds that are present in both the tables – "EmployeeDetails" and "EmployeeSalary".**

Syntax:

select ed.empid from EMPLOYEEDETAILS ed inner join EMPLOYEESALARY es on ed.EMPID=es.EMPID;

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

Syntax:

select ed.empid from EMPLOYEEDETAILS ed left join EMPLOYEESALARY es on ed.EMPID=es.EMPID;

1. **Write an SQL query to fetch the employee's full names and replace the space**

Syntax:

select replace(FULLNAME,' ','') as FullName from EMPLOYEEDETAILS;

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

Syntax:

select FullName,charindex('a', FullName) as Char\_Index\_A from EMPLOYEEDETAILS;

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

Syntax:

select concat(EMPID,MANAGERID) as Emp\_manager\_id from EMPLOYEEDETAILS;

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

Syntax:

select FULLNAME, LEFT(FULLNAME, CHARINDEX(' ',FULLNAME)-1) as FIRST\_NAME from EMPLOYEEDETAILS;

1. **Write an SQL query to uppercase the name of the employee and lowercase the city values.**

Syntax:

select UPPER(FullName) as FULLNAME, lower(city) as CITY from EMPLOYEEDETAILS;

PART-4

1. **Write an SQL query to find the count of the total occurrences of a particular character – 'n' in the FullName field**

Syntax:

select FullName,LEN(replace(FULLNAME,' ',''))-LEN(replace(replace(FULLNAME,' ',''),'n','')) as Count\_Char\_n from EMPLOYEEDETAILS;

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

Syntax:

select replace(FULLNAME,' ','') as FullName from EMPLOYEEDETAILS;

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

Syntax:

select EMPID from EMPLOYEESALARY where PROJECT is null;

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

Syntax:

select ed.fullname from EMPLOYEEDETAILS ed join EMPLOYEESALARY es on ed.EMPID=es.EMPID where es.SALARY>=5000 and es.SALARY<=10000;

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

Syntax:

select GETDATE();

1. **Write an SQL query to fetch all the Employee details from the EmployeeDetails table who joined in the Year 2020.**

Syntax:

select \* from EMPLOYEEDETAILS where YEAR(dateofjoining)=2020;

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

Syntax:

select ed.\*, es.salary from EMPLOYEEDETAILS ed join EMPLOYEESALARY es on ed.EMPID=es.EMPID where es.SALARY is not null;

1. **Write an SQL query to fetch the project-wise count of employees sorted by project's count in descending order.**

Syntax:

select project, count(empid) as count\_empid from EMPLOYEESALARY where project is not null and EMPID is not null group by project;

1. **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.**

Syntax:

select ed.\*, es.\* from EMPLOYEEDETAILS ed full outer join EMPLOYEESALARY es on ed.empid=es.empid;

1. **Write an SQL query to join 3 tables.**

Syntax:

SELECT \* FROM table1 INNER JOIN table2 ON table1.id = table2.id INNER JOIN table3 ON table2.id = table3.id;