

SQL ASSIGNMENTS

We have two table EmployeeDetails and Employee Salary Table

EXECUTE the CREATE Query and Insert the Given records in the particular table in your SQL Editor.

1. **EmployeeDetails** Table

```
CREATE TABLE EmployeeDetails (
    EmpId INT PRIMARY KEY,
    FullName VARCHAR(100),
    ManagerId INT,
    DateOfJoining DATE,
    City VARCHAR(100)
);
```

2. **EmployeeSalary** Table

```
CREATE TABLE EmployeeSalary (
    EmpId INT,
    Project VARCHAR(50),
    Salary DECIMAL(10, 2),
    Variable DECIMAL(10, 2),
    FOREIGN KEY (EmpId) REFERENCES
EmployeeDetails(EmpId)
);
```

Explanation:

- **EmployeeDetails Table:**
 - EmpId: Employee ID, which is the primary key.
 - FullName: Employee's full name.
 - ManagerId: The ID of the manager overseeing the employee.



- DateOfJoining: The date the employee joined the company.
- City: The city where the employee is based.
- **EmployeeSalary Table:**
 - EmpId: Employee ID, which is a foreign key referencing EmpId in the EmployeeDetails table.
 - Project: The project associated with the employee.
 - Salary: The base salary of the employee.
 - Variable: The variable component of the employee's salary (e.g., bonus).



Table 1: EmployeeDetails

```
INSERT INTO EmployeeDetails (EmpId, FullName, ManagerId,
DateOfJoining, City) VALUES
(101, 'Alice Johnson', 321, '2022-05-15', 'New York'),
(102, 'Bob Smith', 876, '2020-03-12', 'Los Angeles'),
(103, 'Charlie Brown', 986, '2021-08-23', 'Chicago'),
(104, 'David Williams', 876, '2019-11-05', 'Houston'),
(105, 'Eve Davis', 321, '2023-01-07', 'Phoenix'),
(106, 'Frank Miller', 986, '2018-12-19', 'Philadelphia'),
(107, 'Grace Wilson', 876, '2022-03-28', 'San Antonio'),
(108, 'Hank Moore', 321, '2021-09-14', 'San Diego'),
(109, 'Ivy Taylor', 986, '2020-02-11', 'Dallas'),
(110, 'Jack Anderson', 876, '2022-11-30', 'San Jose'),
(111, 'Karen Thomas', 321, '2021-07-16', 'Austin'),
(112, 'Liam Jackson', 986, '2023-04-21', 'Fort Worth'),
(113, 'Mia White', 876, '2019-06-03', 'Columbus'),
(114, 'Noah Harris', 321, '2020-12-10', 'Charlotte'),
(115, 'Olivia Martin', 986, '2021-10-25', 'San Francisco'),
(116, 'Paul Garcia', 876, '2023-07-18', 'Indianapolis'),
(117, 'Quinn Martinez', 321, '2022-09-07', 'Seattle'),
(118, 'Rachel Rodriguez', 986, '2020-01-15', 'Denver'),
(119, 'Steve Clark', 876, '2021-03-19', 'Washington'),
(120, 'Tina Lewis', 321, '2019-08-31', 'Boston');
```

Table 2: EmployeeSalary

```
INSERT INTO EmployeeSalary (EmpId, Project, Salary, Variable)
VALUES
(101, 'P1', 7500, 500),
(102, 'P2', 9200, 700),
(103, 'P1', 6700, 600),
(104, 'P3', 8300, 900),
(105, 'P2', 7800, 800),
(106, 'P3', 9100, 1000),
(107, 'P1', 6200, 400),
(108, 'P2', 8800, 750),
(109, 'P3', 9500, 1100),
(110, 'P1', 7200, 650),
(111, 'P2', 8700, 850),
(112, 'P3', 9300, 1200),
(113, 'P1', 7900, 550),
(114, 'P2', 6800, 450),
(115, 'P3', 8400, 900),
(116, 'P1', 7600, 500),
(117, 'P2', 8900, 1000),
(118, 'P3', 9200, 1100),
(119, 'P1', 8100, 600),
(120, 'P2', 8300, 750);
```

Inserting of the 20 records into each table, filling them with various employee details and salary data. Make sure the IDs match across the two tables where relevant. Adjust the ManagerId, DateOfJoining, and City values as needed to match your requirements.

Solve the Below Questions

Basics and Intermediate Questions

PART-1

Q1)SQL Query to fetch records that are present in one table but not in another table.

Q2)SQL query to fetch all the employees who are not working on any project.

Q3)SQL query to fetch all the Employees from EmployeeDetails who joined in the Year 2020.

Q4)Fetch all employees from EmployeeDetails who have a salary record in EmployeeSalary.

Q5)Write an SQL query to fetch a project-wise count of employees.

Q6)Fetch employee names and salaries even if the salary value is not present for the employee.



Q7)Write an SQL query to fetch all the Employees who are also managers.

Q8)Write an SQL query to fetch duplicate records from EmployeeDetails.

Q9)Write an SQL query to fetch only odd rows from the table.

Q10)Write a query to find the 3rd highest salary from a table without top or limit keyword.

PART- 2

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

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

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

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

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

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

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

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

Ques.10. 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.

PART - 3

Ques.1 Write an SQL query to fetch all the Emplds which are present in either of the tables – ‘EmployeeDetails’ and ‘EmployeeSalary’.

Ques.2 Write an SQL query to fetch common records between two tables.

Ques.3. Write an SQL query to fetch records that are present in one table but not in another table.

Ques.4. Write an SQL query to fetch the Emplds that are present in both the tables – ‘EmployeeDetails’ and ‘EmployeeSalary’.

Ques.5. Write an SQL query to fetch the Emplds that are present in EmployeeDetails but not in EmployeeSalary.

Ques.6. Write an SQL query to fetch the employee’s full names and replace the space



Ques.7. Write an SQL query to fetch the position of a given character(s) in a field.

Ques.8. Write an SQL query to display both the EmpId and ManagerId together.

Ques.9. Write a query to fetch only the first name(string before space) from the FullName column of the EmployeeDetails table.

Ques.10. Write an SQL query to uppercase the name of the employee and lowercase the city values.

PART - 4

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

Ques.2. Write an SQL query to update the employee names by removing leading and trailing spaces.

Ques.3. Fetch all the employees who are not working on any project.

Ques.4. Write an SQL query to fetch employee names having a salary greater than or equal to 5000 and less than or equal to 10000.

Ques.5. Write an SQL query to find the current date-time.

Ques.6. Write an SQL query to fetch all the Employee details from the EmployeeDetails table who joined in the Year 2020.

Ques.7. Write an SQL query to fetch all employee records from the EmployeeDetails table who have a salary record in the EmployeeSalary table.

Ques.8. Write an SQL query to fetch the project-wise count of employees sorted by project’s count in descending order.



Ques.9. 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.

Ques.10. Write an SQL query to join 3 tables.

Hints

PART 1

- 1. SQL Query to fetch records that are present in one table but not in another table:**
 - Use a LEFT JOIN between the two tables and filter for NULL values in the right table.

- 2. SQL query to fetch all the employees who are not working on any project:**
 - Use a LEFT JOIN between EmployeeDetails and EmployeeSalary and look for NULL in the Project column.

- 3. SQL query to fetch all the Employees from EmployeeDetails who joined in the Year 2020:**
 - Use the YEAR() function on the DateOfJoining column.

- 4. Fetch all employees from EmployeeDetails who have a salary record in EmployeeSalary:**
 - Perform an INNER JOIN between EmployeeDetails and EmployeeSalary on EmpId.

- 5. Write an SQL query to fetch a project-wise count of employees:**
 - Use the GROUP BY clause on the Project column and apply COUNT().

- 6. Fetch employee names and salaries even if the salary value is not present for the employee:**
 - Use a LEFT JOIN between EmployeeDetails and EmployeeSalary.



7. Write an SQL query to fetch all the Employees who are also managers:

- Check where the EmpId of an employee is present as a ManagerId for other employees.

8. Write an SQL query to fetch duplicate records from EmployeeDetails:

- Use the GROUP BY clause on all columns and apply HAVING COUNT(*) > 1.

9. Write an SQL query to fetch only odd rows from the table:

- Utilize the ROW_NUMBER() window function and filter based on odd values.

10. Write a query to find the 3rd highest salary from a table without using TOP or LIMIT:

- Use the DISTINCT and ORDER BY with a subquery or OFFSET and FETCH clause if available in your SQL dialect.

PART 2

1. **Fetch the EmpId and FullName of all employees working under Manager with id '986':**
 - Use a WHERE clause to filter ManagerId.
2. **Fetch the different projects available from the EmployeeSalary table:**
 - Use DISTINCT with the SELECT statement.
3. **Fetch the count of employees working in project 'P1':**
 - Use COUNT() and GROUP BY with a WHERE clause for the specific project.
4. **Find the maximum, minimum, and average salary of the employees:**
 - Use MAX(), MIN(), and AVG() aggregate functions.
5. **Find the employee id whose salary lies in the range of 9000 and 15000:**
 - Use the BETWEEN clause with the WHERE condition.
6. **Fetch those employees who live in Toronto and work under the manager with ManagerId – 321:**
 - Combine AND conditions for City and ManagerId.
7. **Fetch all the employees who either live in California or work under a manager with ManagerId – 321:**
 - Combine OR conditions for City and ManagerId.



8. **Fetch all those employees who work on Projects other than P1:**
 - Use NOT with the WHERE clause.

9. **Display the total salary of each employee adding the Salary with Variable value:**
 - Use the + operator to sum the Salary and Variable columns.

10. **Fetch the employees whose name begins with any two characters, followed by "hn", and ends with any sequence of characters:**
 - Use the LIKE operator with wildcards (%) and (_).

PART 3

1. **Fetch all the EmpIds which are present in either of the tables – ‘EmployeeDetails’ and ‘EmployeeSalary’:**
 - Use the UNION operator.
2. **Fetch common records between two tables:**
 - Use an INNER JOIN on EmpId.
3. **Fetch records that are present in one table but not in another table:**
 - Use a LEFT JOIN and filter for NULL in the right table.
4. **Fetch the EmpIds that are present in both the tables – ‘EmployeeDetails’ and ‘EmployeeSalary’:**
 - Use an INNER JOIN on EmpId.
5. **Fetch the EmpIds that are present in EmployeeDetails but not in EmployeeSalary:**
 - Use a LEFT JOIN and filter for NULL in the EmployeeSalary table.
6. **Fetch the employee’s full names and replace the space:**
 - Use the REPLACE() function to substitute spaces with another character.
7. **Fetch the position of a given character(s) in a field:**
 - Use the INSTR() or POSITION() function depending on your SQL dialect.



8. **Display both the EmpId and ManagerId together:**
 - Use the CONCAT() function or || operator.

9. **Fetch only the first name (string before space) from the FullName column:**
 - Use the SUBSTRING() or LEFT() function with CHARINDEX() or LOCATE().

10. **Uppercase the name of the employee and lowercase the city values:**
 - Use the UPPER() and LOWER() functions.

PART 4

- 1. Find the count of the total occurrences of a particular character – ‘n’ in the FullName field:**
 - Use the LENGTH() function combined with REPLACE() to count occurrences.

- 2. Update the employee names by removing leading and trailing spaces:**
 - Use the TRIM() function.

- 3. Fetch all the employees who are not working on any project:**
 - Use a LEFT JOIN with a NULL check in the Project column.

- 4. Fetch employee names having a salary greater than or equal to 5000 and less than or equal to 10000:**
 - Use the BETWEEN clause in the WHERE condition.

- 5. Find the current date-time:**
 - Use the NOW() or CURRENT_TIMESTAMP function.

- 6. Fetch all Employee details from the EmployeeDetails table who joined in the Year 2020:**
 - Use the YEAR() function on DateOfJoining.

- 7. Fetch all employee records from the EmployeeDetails table who have a salary record in the EmployeeSalary table:**
 - Use an INNER JOIN on EmpId.



8. **Fetch the project-wise count of employees sorted by project's count in descending order:**
 - Use GROUP BY with ORDER BY COUNT(*) DESC.

9. **Fetch employee names and salary records. Display the employee details even if the salary record is not present:**
 - Use a LEFT JOIN between EmployeeDetails and EmployeeSalary.

10. **Write an SQL query to join 3 tables:**
 - Use multiple JOIN statements to combine three tables on relevant keys.