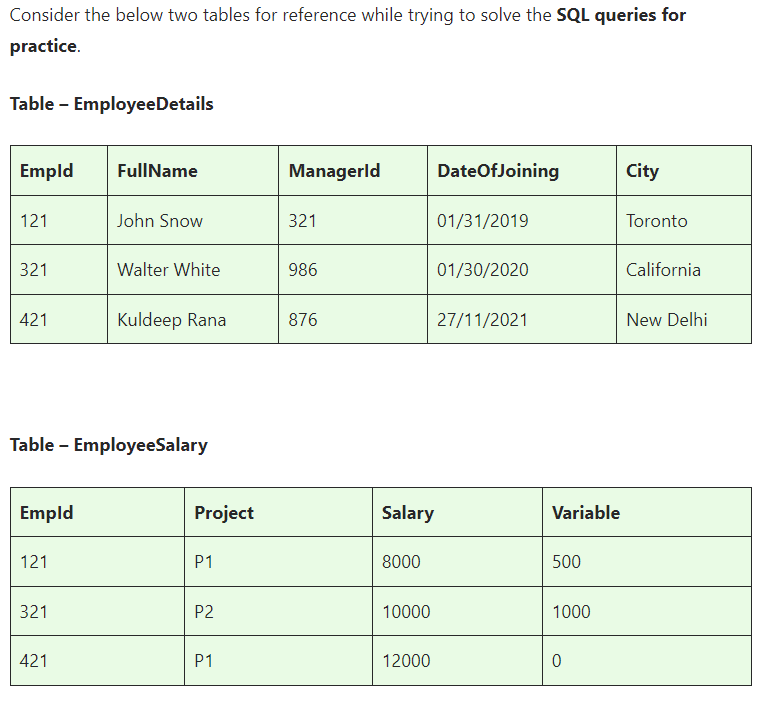
Logo

Description automatically generated

**SQL ASSIGNMENTS**



**Basics and Intermediate ASSINGMENT**

**ASSINGMENT nos - 1**

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

**SELECT \***

**FROM EmployeeDetails**

**WHERE EmpId NOT IN (SELECT EmpId FROM EmployeeSalary);**

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

**SELECT EmpId, FullName**

**FROM EmployeeDetails**

**WHERE EmpId NOT IN (SELECT EmpId FROM EmployeeSalary);**

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

**SELECT \***

**FROM EmployeeDetails**

**WHERE YEAR(DateOfJoining) = 2020;**

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

**SELECT \***

**FROM EmployeeDetails**

**WHERE EmpId IN (SELECT EmpId FROM EmployeeSalary);**

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

**SELECT Project, COUNT(EmpId) AS EmployeeCount**

**FROM EmployeeSalary**

**GROUP BY Project;**

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

**SELECT ED.FullName, ES.Salary**

**FROM EmployeeDetails ED**

**LEFT JOIN EmployeeSalary ES ON ED.EmpId = ES.EmpId;**

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

**SELECT EmpId, FullName**

**FROM EmployeeDetails**

**WHERE EmpId IN (SELECT ManagerId FROM EmployeeDetails);**

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

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

**FROM EmployeeDetails**

**GROUP BY EmpId, FullName, ManagerId, DateOfJoining, City**

**HAVING COUNT(\*) > 1;**

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

**WITH RowNumbered AS (**

**SELECT \*, ROW\_NUMBER() OVER (ORDER BY EmpId) AS RowNum**

**FROM EmployeeDetails**

**)**

**SELECT \***

**FROM RowNumbered**

**WHERE RowNum % 2 = 1;**

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

**SELECT MIN(Salary) AS ThirdHighestSalary**

**FROM (SELECT DISTINCT Salary**

**FROM EmployeeSalary**

**ORDER BY Salary DESC**

**OFFSET 2 ROWS FETCH NEXT 1 ROWS ONLY) AS SubQuery;**

**ASSINGMENT nos - 2**

**Ques.1. Write an SQL query to fetch the EmpId and FullName of all the employees working under the 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(\*) AS EmployeeCount**

**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) AS MaxSalary,**

**MIN(Salary) AS MinSalary,**

**AVG(Salary) AS AvgSalary**

**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**

**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 the manager with ManagerId – 321.**

**SELECT \***

**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 \***

**FROM EmployeeDetails**

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

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

**SELECT \***

**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 ends with any sequence of characters.**

**SELECT \***

**FROM EmployeeDetails**

**WHERE FullName LIKE '\_\_hn%';**

**ASSINGMENT nos - 3**

**Ques.1 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.2 Write an SQL query to fetch common records between two tables.**

**SELECT ED.EmpId, ED.FullName, ES.Project, ES.Salary**

**FROM EmployeeDetails ED**

**INNER JOIN EmployeeSalary ES ON ED.EmpId = ES.EmpId;**

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

**SELECT EmpId, FullName**

**FROM EmployeeDetails**

**WHERE EmpId NOT IN (SELECT EmpId FROM EmployeeSalary);**

**Ques.4. 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.5. 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.6. Write an SQL query to fetch the employee’s full names and replace the space**

**SELECT REPLACE(FullName, ' ', '') AS FullNameWithoutSpace**

**FROM EmployeeDetails;**

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

**SELECT CHARINDEX('n', FullName) AS PositionOfN**

**FROM EmployeeDetails;**

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

**SELECT EmpId, ManagerId**

**FROM EmployeeDetails;**

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

**SELECT LEFT(FullName, CHARINDEX(' ', FullName) - 1) AS FirstName**

**FROM EmployeeDetails;**

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

**SELECT UPPER(FullName) AS UpperCaseName, LOWER(City) AS LowerCaseCity**

**FROM EmployeeDetails;**

**ASSINGMENT nos - 4**

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

**SELECT FullName, LEN(FullName) - LEN(REPLACE(FullName, 'n', '')) AS CountOfN**

**FROM EmployeeDetails;**

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

**UPDATE EmployeeDetails**

**SET FullName = LTRIM(RTRIM(FullName));**

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

**SELECT \***

**FROM EmployeeDetails**

**WHERE EmpId NOT IN (SELECT EmpId FROM EmployeeSalary);**

**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.**

**SELECT ED.FullName**

**FROM EmployeeDetails ED**

**JOIN EmployeeSalary ES ON ED.EmpId = ES.EmpId**

**WHERE ES.Salary BETWEEN 5000 AND 10000;**

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

**SELECT GETDATE() AS CurrentDateTime;**

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

**SELECT \***

**FROM EmployeeDetails**

**WHERE YEAR(DateOfJoining) = 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.**

**SELECT \***

**FROM EmployeeDetails**

**WHERE EmpId IN (SELECT EmpId FROM EmployeeSalary);**

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

**SELECT Project, COUNT(\*) AS EmployeeCount**

**FROM EmployeeSalary**

**GROUP BY Project**

**ORDER BY EmployeeCount DESC;**

**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.**

**SELECT ED.EmpId, ED.FullName, ES.Salary**

**FROM EmployeeDetails ED**

**LEFT JOIN EmployeeSalary ES ON ED.EmpId = ES.EmpId;**

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

**SELECT ED.EmpId, ED.FullName, ES.Salary, EP.ProjectName**

**FROM EmployeeDetails ED**

**LEFT JOIN EmployeeSalary ES ON ED.EmpId = ES.EmpId**

**LEFT JOIN EmployeeProjects EP ON ED.EmpId = EP.EmpId;**

**Advanced ASSINGMENT**



**Q1)Write a query to fetch the EmpFname from the EmployeeInfo table in the upper case and use the ALIAS name as EmpName.**

**SELECT UPPER(EmpFname) AS EmpName FROM EmployeeInfo;**

**Q2)Write a query to fetch the number of employees working in the department ‘HR’.**

**SELECT COUNT(\*) AS HR\_Employees FROM EmployeeInfo WHERE Department = 'HR';**

**Q3)Write a query to get the current date.**

**SELECT CURRENT\_DATE();**

**Q4)Write a query to retrieve the first four characters of EmpLname from the EmployeeInfo table.**

**SELECT SUBSTRING(EmpLname, 1, 4) AS ShortLname FROM EmployeeInfo;**

**Q5)Write a query to fetch only the place name(string before brackets) from the Address column of EmployeeInfo table.**

**SELECT SUBSTRING\_INDEX(Address, '(', 1) AS Place FROM EmployeeInfo;**

**Q6)Write a query to create a new table that consists of data and structure copied from the other table.**

**CREATE TABLE NewEmployeeInfo AS SELECT \* FROM EmployeeInfo;**

**Q7)Write q query to find all the employees whose salary is between 50000 to 100000.**

**SELECT \* FROM EmployeePosition WHERE Salary BETWEEN 50000 AND 100000;**

**Q8)Write a query to find the names of employees that begin with ‘S’**

**SELECT EmpFname FROM EmployeeInfo WHERE EmpFname LIKE 'S%';**

**Q9)Write a query to fetch top N records.**

**SELECT \* FROM EmployeeInfo LIMIT 3;**

**Q10)Write a query to retrieve the EmpFname and EmpLname in a single column as “FullName”. The first name and the last name must be separated with space.**

**SELECT CONCAT(EmpFname, ' ', EmpLname) AS FullName FROM EmployeeInfo;**

### ****Q11. Write a query find number of employees whose DOB is between 02/05/1970 to 31/12/1975 and are grouped according to gender****

SELECT Gender, COUNT(\*) AS EmployeeCount

FROM EmployeeInfo

WHERE DOB BETWEEN '1970-05-02' AND '1975-12-31'

GROUP BY Gender;

### ****Q12. Write a query to fetch all the records from the EmployeeInfo table ordered by EmpLname in descending order and Department in the ascending order.****

SELECT \* FROM EmployeeInfo

ORDER BY EmpLname DESC, Department ASC;

### ****Q13. Write a query to fetch details of employees whose EmpLname ends with an alphabet ‘A’ and contains five alphabets.****

SELECT \* FROM EmployeeInfo

WHERE EmpLname LIKE '\_\_\_\_A';

### ****Q14. Write a query to fetch details of all employees excluding the employees with first names, “Sanjay” and “Sonia” from the EmployeeInfo table.****

SELECT \* FROM EmployeeInfo

WHERE EmpFname NOT IN ('Sanjay', 'Sonia');

### ****Q15. Write a query to fetch details of employees with the address as “DELHI(DEL)”.****

SELECT \* FROM EmployeeInfo WHERE Address = 'Delhi(DEL)';

### ****Q16. Write a query to fetch all employees who also hold the managerial position.****

**SELECT e.\***

**FROM EmployeeInfo e**

**JOIN EmployeePosition p ON e.EmpID = p.EmpID**

**WHERE p.EmpPosition = 'Manager';**

### ****Q17.**** Write a query to fetch the department-wise count of employees sorted by department’s count in ascending order.

**SELECT Department, COUNT(\*) AS EmployeeCount**

**FROM EmployeeInfo**

**GROUP BY Department**

**ORDER BY EmployeeCount ASC;**

### ****Q18. Write a query to calculate the even and odd records from a table.****

**SELECT \* FROM EmployeeInfo WHERE EmpID % 2 = 0;**

**SELECT \* FROM EmployeeInfo WHERE EmpID % 2 != 0;**

### ****Q19.**** Write a SQL query to retrieve employee details from EmployeeInfo table who have a date of joining in the EmployeePosition table.

**SELECT e.\***

**FROM EmployeeInfo e**

**JOIN EmployeePosition p ON e.EmpID = p.EmpID;**

### ****Q20. Write a query to retrieve two minimum and maximum salaries from the EmployeePosition table.****

**SELECT DISTINCT Salary**

**FROM EmployeePosition**

**ORDER BY Salary ASC**

**LIMIT 2;**

**SELECT DISTINCT Salary**

**FROM EmployeePosition**

**ORDER BY Salary DESC**

**LIMIT 2; -- Maximum salaries**

### ****Q21.**** Write a query to find the Nth highest salary from the table without using TOP/limit keyword.

SELECT DISTINCT Salary

FROM EmployeePosition e1

WHERE 2 = (SELECT COUNT(DISTINCT Salary)

FROM EmployeePosition e2

WHERE e2.Salary > e1.Salary);

### ****Q22. Write a query to retrieve duplicate records from a table.****

**SELECT EmpID, COUNT(\*) AS Count**

**FROM EmployeePosition**

**GROUP BY EmpID**

**HAVING Count > 1;**

### ****Q23. Write a query to retrieve the list of employees working in the same department.****

**SELECT e1.EmpFname AS Employee1, e2.EmpFname AS Employee2, e1.Department**

**FROM EmployeeInfo e1**

**JOIN EmployeeInfo e2**

**ON e1.Department = e2.Department AND e1.EmpID != e2.EmpID;**

### ****Q24. Write a query to retrieve the last 3 records from the EmployeeInfo table.****

**SELECT \* FROM EmployeeInfo ORDER BY EmpID DESC LIMIT 3;**

### ****Q25. Write a query to find the third-highest salary from the EmpPosition table.****

**SELECT DISTINCT Salary**

**FROM EmployeePosition e1**

**WHERE 2 = (SELECT COUNT(DISTINCT Salary)**

**FROM EmployeePosition e2**

**WHERE e2.Salary > e1.Salary);**

### ****Q26. Write a query to display the first and the last record from the EmployeeInfo table.****

**SELECT \* FROM EmployeeInfo ORDER BY EmpID ASC LIMIT 1;**

**SELECT \* FROM EmployeeInfo ORDER BY EmpID DESC LIMIT 1;**

### ****Q27. Write a query to add email validation to your database****

**ALTER TABLE EmployeeInfo**

**ADD Email VARCHAR(255) CHECK (Email LIKE '%\_@\_%.\_%');**

### ****Q28. Write a query to retrieve Departments who have less than 2 employees working in it.****

**SELECT Department**

**FROM EmployeeInfo**

**GROUP BY Department**

**HAVING COUNT(\*) < 2;**

### ****Q29. Write a query to retrieve EmpPostion along with total salaries paid for each of them****

**SELECT EmpPosition, SUM(Salary) AS TotalSalary**

**FROM EmployeePosition**

**GROUP BY EmpPosition;**

### ****Q30. Write a query to fetch 50% records from the EmployeeInfo table.****

**SELECT \* FROM EmployeeInfo LIMIT (SELECT COUNT(\*) / 2 FROM EmployeeInfo);**

Logo

Description automatically generated