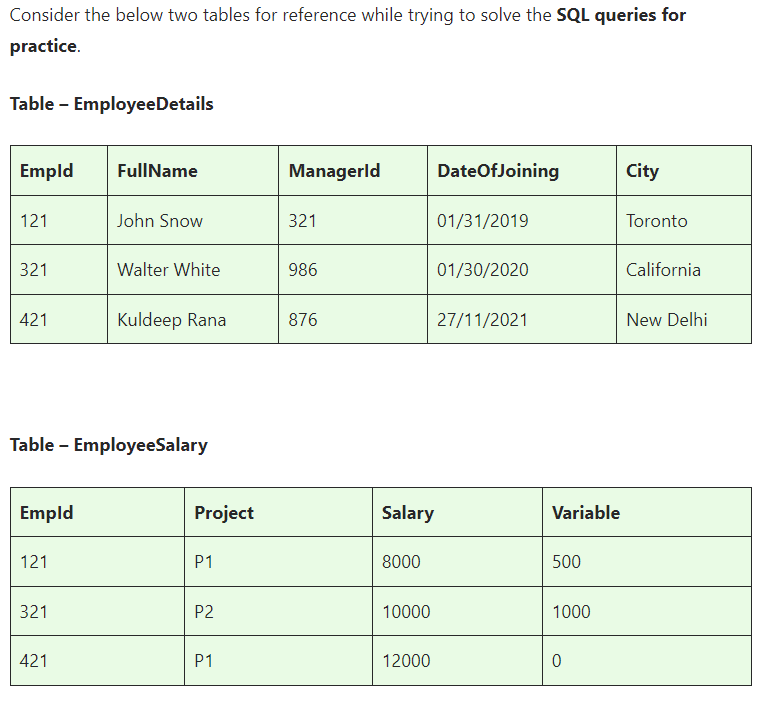
Logo

Description automatically generated

**SQL ASSIGNMENTS**



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table : EmployeeDetails | |  | |  | |  | |  |
| EmpId | | FullName | | ManagerID | | DateOfJoining | | City |
| 121 | | John Snow | | 321 | | 01/31/2019 | | Toronto |
| 321 | | Walter White | | 986 | | 01/30/2020 | | Califonia |
| 421 | | Kuldeep Rana | | 876 | | 27-11-2021 | | New Delhi |
| Table: EmployeeSalary |  | |  | |  | |
| EmpId | Project | | Salary | | Variable | |
| 121 | P1 | | 8000 | | 500 | |
| 321 | P2 | | 10000 | | 1000 | |
| 421 | P1 | | 12000 | | 0 | |

**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 EmployeeDetails.\***

**FROM EmployeeDetails**

**LEFT JOIN EmployeeSalary ON EmployeeDetails.EmpId = EmployeeSalary.EmpId**

**WHERE EmployeeSalary.EmpId IS NULL;**

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

**SELECT \***

**FROM EmployeeDetails**

**WHERE EmpId NOT IN (**

**SELECT EmpId**

**FROM EmployeeSalary**

**);**

**Or**

**SELECT EmployeeDetails.\***

**FROM EmployeeDetails**

**LEFT JOIN EmployeeSalary ON EmployeeDetails.EmpId = EmployeeSalary.EmpId**

**WHERE EmployeeSalary.EmpId IS NULL;**

**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 ED.\***

**FROM EmployeeDetails ED**

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

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

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

**FROM EmployeeSalary**

**GROUP BY Project;**

**or**

**SELECT ES.Project, COUNT(ED.EmpId) AS EmployeeCount**

**FROM EmployeeDetails ED**

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

**GROUP BY ES.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 e1.\***

**FROM EmployeeDetails e1**

**SELF JOIN EmployeeDetails e2 ON e1.EmpId = e2.ManagerID;**

**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.**

**SELECT \***

**FROM EmployeeDetails**

**WHERE MOD(EmpId, 2) <> 0;**

**or**

**SELECT \***

**FROM EmployeeSalary**

**WHERE MOD(EmpId, 2) <> 0;**

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

SELECT DISTINCT Salary

FROM EmployeeDetails Emp1

WHERE 3 = (

SELECT COUNT(DISTINCT Salary)

FROM EmployeeDetails Emp2

WHERE Emp2.Salary >= Emp1.Salary

);

or

SELECT MAX(Salary) AS ThirdHighestSalary ---3rd execution

FROM EmployeeDetails

WHERE Salary < (

SELECT MAX(Salary) ----2nd execution

FROM EmployeeDetails

WHERE Salary < (

SELECT MAX(Salary) ----1st execution

FROM EmployeeDetails

)

);

**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 ED.\***

**FROM EmployeeDetails ED**

**INNER JOIN EmployeeDetails Manager ON ED.ManagerID = Manager.EmpId**

**WHERE ED.City = 'Toronto' AND Manager.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 ED.\***

**FROM EmployeeDetails ED**

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

**WHERE ES.Project <> 'P1';**

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

**SELECT EmpId, SUM(Salary + Variable) AS TotalSalary**

**FROM EmployeeSalary**

**GROUP BY EmpId;**

**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 \***

**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 ED.\***

**FROM EmployeeDetails ED**

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

**WHERE ES.EmpId IS NULL;**

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

**SELECT ED.EmpId**

**FROM EmployeeDetails ED**

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

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

**SELECT ED.EmpId**

**FROM EmployeeDetails ED**

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

**WHERE ES.EmpId IS NULL;**

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

**SELECT REPLACE(FullName, ' ', '\_') AS NewFullName**

**FROM EmployeeDetails;**

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

**SELECT INSTR(FullName, 'a') AS Character**

**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 SUBSTRING\_INDEX(FullName, ' ', 1) AS FirstName**

**FROM EmployeeDetails;**

**Or**

**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**

**LENGTH(FullName) - LENGTH(REPLACE(FullName, 'n', '')) AS Count\_of\_n**

**FROM EmployeeDetails;**

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

**UPDATE EmployeeDetails**

**SET FullName = TRIM(FullName);**

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

**SELECT ED.\***

**FROM EmployeeDetails ED**

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

**WHERE ES.Project IS NULL;**

**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 >= 5000 AND ES.Salary <= 10000;**

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

**SELECT GETDATE() AS CurrentDateTime;**

**Or**

**SELECT SYSDATE AS CurrentDateTime FROM DUAL;**

**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 ED.\***

**FROM EmployeeDetails ED**

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

**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 COUNT(\*) 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.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.FullName, ES.Salary, P.Project**

**FROM EmployeeDetails ED**

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

**JOIN Projects P ON ES.Project = P.ProjectId;**

**Or**

**SELECT columns**

**FROM table1**

**JOIN table2 ON table1.column = table2.column**

**JOIN table3 ON table2.column = table3.column;**