

Module 5

What to learn

Inner join

Outer join

Left Outer Join

Right Outer join

Full Outer Join

Self-Join

Practice Exercise

Practice 1

Practice Exercise: Using the UNION Clause

You are provided with two tables: **Products** and **Services**. The **Products** table contains the following columns: **ProductID**, **ProductName**, **Category**, and **Price**. The **Services** table contains the following columns: **ServiceID**, **ServiceName**, **Category**, and **Cost**. Write an SQL query using the **UNION** clause to list all the distinct names of products and services along with their categories. Ensure that column names in the result set match and are intuitive, such as **Name** and **Category**.

Practice 2

Practice Exercise: Applying the EXCEPT Clause

You are given two tables: **EnrolledStudents** (columns: **StudentID**, **Name**) and **GraduatedStudents** (columns: **StudentID**, **Name**). Write an SQL query using the **EXCEPT** clause to identify students who are enrolled but have not graduated. Display their **StudentID** and **Name** in the result.

Practice 3

Practice Exercise: Utilizing the INTERSECT Clause

You are provided with two tables: **CorporateEmails** (columns: **EmailID**, **EmailAddress**, **Status**) and **PersonalEmails** (columns: **EmailID**, **EmailAddress**, **Status**). Write an SQL query using the **INTERSECT** clause to find **EmailAddress** values that exist in both tables and whose **Status** column is marked as 'Active'. Display the **EmailAddress** and **Status** in the result.

Practice 4

Practice Exercise: Derived Tables for Aggregation

You have a **SalesData** table with the following columns: **SalesID**, **Region**, **ProductCategory**, and **TotalSales**. Write an SQL query that uses a derived table to calculate the total sales

for each **Region**. The derived table should group the results by **Region**, and the main query should fetch the regional totals.

Practice 5

Practice Exercise: Building a Common Table Expression (CTE)

You have a table called **EmployeeProjects** with the following columns: **EmployeeID**, **ProjectID**, **HoursWorked**. Write an SQL query using a CTE to calculate the total hours worked by each employee across all projects. The main query should then display the **EmployeeID** and their **TotalHours**.

Assignment Exercise

Assignment 1

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT	MANAGER_ID
1	John	Abraham	1000000	01-JAN-13 12.00.00 AM	Banking	NULL
2	Michael	Clarke	800000	01-JAN-13 12.00.00 AM	Insurance	1
3	Roy	Thomas	700000	01-FEB-13 12.00.00 AM	Banking	1
4	Tom	Jose	600000	01-FEB-13 12.00.00 AM	Insurance	2
5	Jerry	Pinto	650000	01-FEB-13 12.00.00 AM	Insurance	3
6	Philip	Mathew	750000	01-JAN-13 12.00.00 AM	Services	3
7	TestName1	123	650000	01-JAN-13 12.00.00 AM	Services	2
8	TestName2	Lname%	600000	01-FEB-13 12.00.00 AM	Insurance	2

EMPLOYEE_REF_ID	INCENTIVE_DATE	INCENTIVE_AMOUNT
1	01-FEB-13	5000
2	01-FEB-13	3000
3	01-FEB-13	4000
1	01-JAN-13	4500
2	01-JAN-13	3500

Get difference between JOINING_DATE and INCENTIVE_DATE from employee and incentives table

Assignment 2

Select first_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000

Assignment 3

Select first_name, incentive amount from employee and incentives table for all employees even if they didn't get incentives.

Assignment 4

Select EmployeeName, ManagerName from the employee table.

Assignment 5

Select first_name, incentive amount from employee and incentives table for all employees even if they didn't get incentives and set incentive amount as 0 for those employees who didn't get incentives.

Assignment 6

Assignment: Employee and Department Portal for Resource Management

Background: A company wants to manage employees and departments through a database. The database tracks employee details, their respective departments, and their project assignments. You must use SQL concepts including UNION, EXCEPT, INTERSECT, Derived Tables, and CTEs to query data for various business scenarios.

Tables Provided:

Employees: EmployeeID (PK), FirstName, LastName, DepartmentID.

Departments: DepartmentID (PK), DepartmentName, ManagerID.

Projects: ProjectID (PK), ProjectName, DepartmentID.

ProjectAssignments: AssignmentID (PK), EmployeeID, ProjectID, HoursWorked.

Business Requirements:

Create a query using UNION to list all unique Department Names from both the Employees and Departments tables.

Identify employees who are not assigned to any projects using EXCEPT queries.

Find employees who are assigned to the same projects using INTERSECT queries.

Write a query using a Derived Table to calculate the total hours worked on projects by department. The main query should fetch the department names along with their total hours.

Write a query using a CTE to generate a list showing each employee's name and the total hours they have contributed to all projects combined.

Output:

You are expected to create and execute queries for the above requirements, ensuring accuracy and optimal performance. Provide screenshots of the result sets to confirm

your answers.

Online Reference

No online Reference

Introduction to Relational Databases

Introduction to Select Statement

Filtering Results with WHERE Statements

Utilizing Joins

Executing Sub queries and Unions

Aggregating Data

Advanced Data Aggregations

Built in Functions

Query Optimization

Modifying Data

Advanced Data Modification

Stored Procedure

Transaction

Error handling

Designing Tables

triggers