upGrad ENTERPRISE

Advanced SQL Concepts (Joins and CTEs)

By

Narasimha Rao T

Microsoft.Net FSD Trainer

Professional Development Trainer

tnrao.trainer@gmail.com

Microsoft® SQL Server®

Day-18 (01-Aug-2025)

1. Review of All Join Types

Joins are used to combine rows from two or more tables based on a related column.

1.1 INNER JOIN

- Returns only the rows that have matching values in both tables.

```sql
SELECT *
FROM Orders o
INNER JOIN Customers c ON o.CustomerID = c.CustomerID;
```

1.2 LEFT JOIN (LEFT OUTER JOIN)

- Returns all rows from the left table, and matched rows from the right table. NULL if no match.

```
SELECT *
FROM Customers c
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID;
```

1.3 RIGHT JOIN (RIGHT OUTER JOIN)

- Returns all rows from the right table, and matched rows from the left table.

```
SELECT *
FROM Orders o
RIGHT JOIN Customers c ON o.CustomerID = c.CustomerID;
```

1.4 FULL OUTER JOIN

- Returns rows when there is a match in one of the tables. NULLs for missing matches.

```sql
SELECT *
FROM Customers c
FULL OUTER JOIN Orders o ON c.CustomerID = o.CustomerID;
```

1.5 CROSS JOIN

- Returns the Cartesian product: all combinations of rows from both tables.

```sql
SELECT *
FROM Products p
CROSS JOIN Categories c;
```

1.6 SELF JOIN

- A table is joined with itself.

```sql
SELECT a.EmployeeName, b.ManagerName
FROM Employees a
JOIN Employees b ON a.ManagerID = b.EmployeeID;
```

2. Choosing the Right Join

Scenario	Suggested Join
Matching records in both tables	`INNER JOIN`
Need all records from one table	`LEFT JOIN` or `RIGHT JOIN`
Need all records from both tables	`FULL OUTER JOIN`
Every combination of two tables	`CROSS JOIN`
Comparing rows within the same table	`SELF JOIN`

Tip: Always consider data relationships and cardinality.

3. What is Common Table Expressions (CTEs)?

3.1 Introduction to CTEs

A CTE is a temporary result set defined within the execution scope of a `SELECT`, `INSERT`, `UPDATE`, or `DELETE` statement.

3.2 Syntax of a CTE

```sql
WITH CTE_Name AS (
    SELECT column1, column2
    FROM Table
    WHERE condition
)
SELECT *
FROM CTE_Name;
```

4. Advantages of CTEs

- **Improved readability** of complex queries.
- **Modular query design** — break down logic into steps.
- **Reusability** within the same statement (especially useful in joins).
- Reduces redundancy compared to repeating subqueries.

5. CTE vs Subquery: Readability and Reuse

Feature	CTE	Subquery
Readability	High – easier to understand	Lower – nested and complex
Reusability	Can be reused in main query	Must be repeated
Recursive	Yes	No
Scope	One query	Limited to nesting location

6. Using CTEs with Joins

CTEs can simplify join operations by abstracting intermediate results.

Example:

```sql
WITH HighValueOrders AS (
    SELECT OrderID, CustomerID, TotalAmount
    FROM Orders
    WHERE TotalAmount > 1000
)
SELECT c.CustomerName, h.TotalAmount
FROM HighValueOrders h
JOIN Customers c ON h.CustomerID = c.CustomerID;
```

7. Real-Time Case Studies

Case 1: Employee Hierarchy

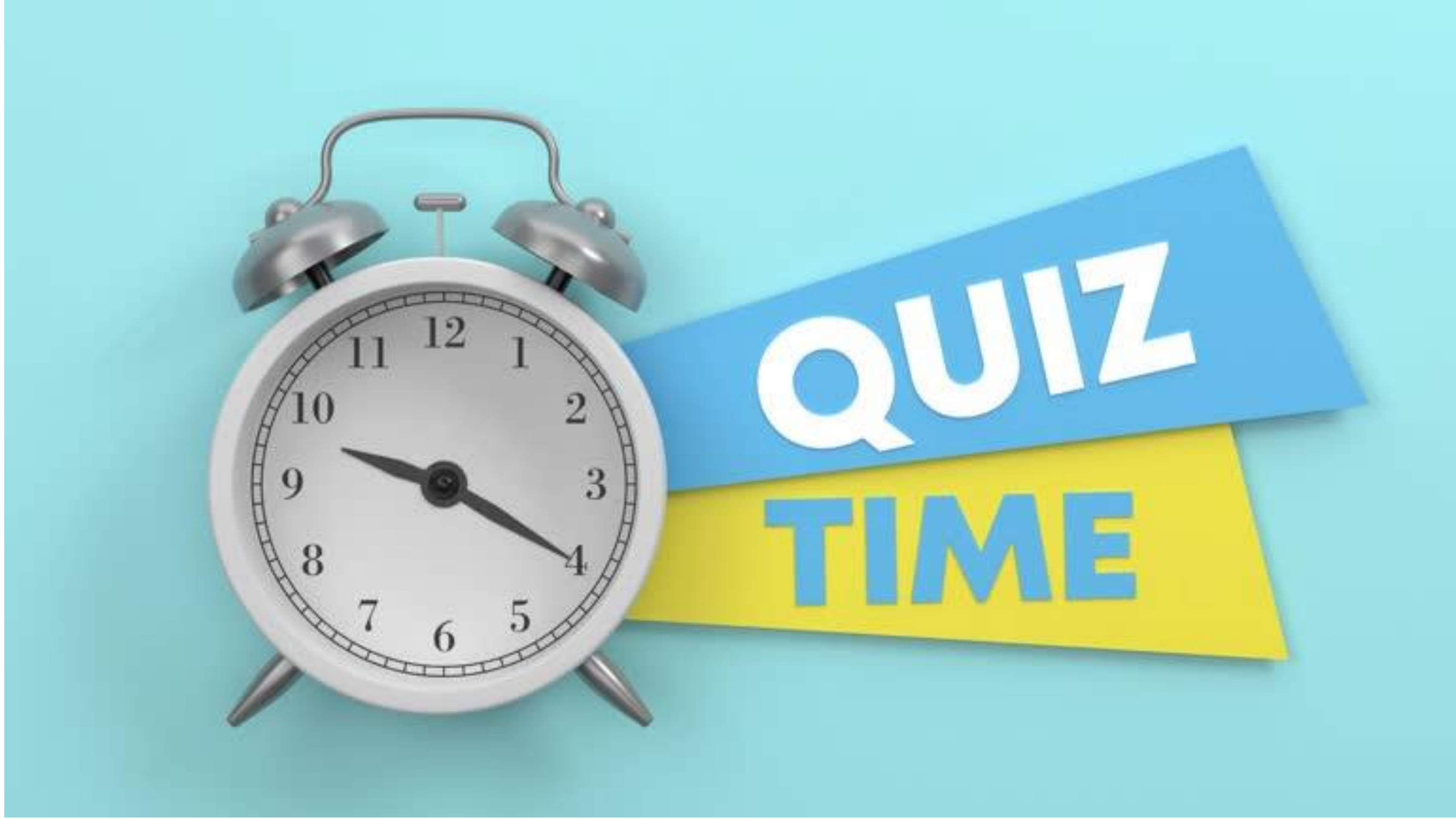
Using a recursive CTE to find all subordinates under a manager.

```
WITH EmployeeHierarchy AS (
    SELECT EmployeeID, ManagerID, Name
    FROM Employees
    WHERE ManagerID IS NULL
    UNION ALL
    SELECT e.EmployeeID, e.ManagerID, e.Name
    FROM Employees e
    JOIN EmployeeHierarchy eh ON e.ManagerID = eh.EmployeeID
)
SELECT * FROM EmployeeHierarchy;
```

Case 2: Sales Analysis

Find top 3 salespeople by region.

```sql
WITH RankedSales AS (
    SELECT SalespersonID, Region, TotalSales,
           RANK() OVER (PARTITION BY Region ORDER BY TotalSales DESC) AS rank
    FROM Sales
)
SELECT *
FROM RankedSales
WHERE rank <= 3;
```


QUIZ
TIME

MCQs

1. Which join returns all records from both tables with NULLs in unmatched rows?

   ○ A. INNER JOIN

   ○ B. LEFT JOIN

   ○ C. FULL OUTER JOIN

   ○ D. CROSS JOIN

2. Which keyword starts a Common Table Expression?
    - A. TEMP
    - B. DEFINE
    - C. WITH
    - D. AS

3. CTEs are best used when:

   - A. You want to store data permanently
   - B. You need to simplify complex subqueries
   - C. You want to enforce primary keys
   - D. You are deleting data

True/False

1. A CTE can be recursive.
2. A CROSS JOIN filters data using ON clause.
3. CTEs can be reused in multiple SQL queries.

upGrad ENTERPRISE

Q & A

Narasimha Rao T

tnrao.trainer@gmail.com