Data Analyst Training Programme

JOINs, Subqueries & CTEs

Connecting Data Across Tables

The Multi-Table Reality

- Business data lives in separate tables
- Relationships connect related information
- No single table tells the complete story

Three Powerful Techniques

- **JOINs:** combine tables horizontally
- **Subqueries:** nested queries for complex logic
- **CTEs:** readable, reusable query components

Business Impact

- Customer orders with product details
- Sales analysis across categories
- Complex analytical reporting
- Data warehouse-style analytics

Understanding JOIN Types

INNER JOIN

- Returns only matching records from both tables
- Most common JOIN type
- Excludes orphaned records

LEFT JOIN (LEFT OUTER JOIN)

- Returns all records from left table
- Includes non-matching records with NULL values
- Essential for finding gaps in data

RIGHT JOIN (RIGHT OUTER JOIN)

- Returns all records from right table
- Less commonly used than LEFT JOIN

Key Concepts

- Foreign key relationships
- ON clause defines join conditions
- NULL handling in outer joins

Advanced JOIN Techniques

Multiple JOINs
- Chain tables together for complex analysis
- Order matters for performance
- Each JOIN adds another relationship layer

JOIN Conditions
- Equality joins (most common)
- Range joins for date periods
- Complex conditions with AND/OR

Self JOINs
- Join table to itself
- Useful for hierarchical data
- Employee-manager relationships

Performance Considerations
- Index usage in JOIN conditions
- Join order optimisation
- Filtering early vs late

Subqueries: Queries Within Queries

Subquery Types
- Scalar subqueries (return single value)
- List subqueries (return multiple values)
- Table subqueries (return multiple rows/columns)

Subquery Locations
- SELECT clause: calculated columns
- WHERE clause: filtering conditions
- FROM clause: virtual tables

Correlated vs Non-Correlated
- Non-correlated: independent of outer query
- Correlated: references outer query columns
- Performance implications

Common Use Cases
- Finding above-average values
- Existence checking
- Complex filtering logic

Common Table Expressions (CTEs)

CTE Fundamentals
- WITH clause creates named result sets
- Available for duration of single query
- Improves readability and maintenance

CTE Advantages
- Break complex queries into logical steps
- Reuse subqueries multiple times
- Easier debugging and testing
- Self-documenting code

Multiple CTEs
- Chain multiple WITH clauses
- Build complex analysis step by step
- Reference earlier CTEs in later ones

Recursive CTEs
- Handle hierarchical data
- Advanced technique for organisational charts
- Tree-like data structures

Assignment

Complete these multi-table analysis challenges using the Northwind database:

Basic JOINs

- List all products with their category names and supplier information
- Show all orders with customer names, employee names, and shipper details
- Find customers who have never placed an order

Advanced JOIN Analysis

- Calculate total revenue by employee (who processed the orders)
- Show product sales performance including products that were never ordered
- Create a detailed order analysis showing customer, product, category, and supplier details for orders from 1996

Subquery Challenges

- Find products priced above their category average
- Identify customers who have spent more than the overall customer average
- List categories that have more products than the average category size

CTE Complex Analysis

- Build a customer ranking system showing total spent, order count, and their rank within their country
- Create a monthly sales analysis with growth rates and moving averages
- Develop a product performance report showing sales rank within categories and overall business contribution
- Design a supplier performance analysis combining product count, average prices, and total revenue generated

Until Next Week Sunday...

See you next week on Sunday, **[student name]**.

You now have the power to connect any tables in any database and answer complex business questions that span multiple data sources. These skills turn you from someone who queries single tables into a true data analyst who can solve real business problems.

Thank you, [student name].

Any Questions?