SQL Programming

Duration: *(Face-to-Face & Remote-Live)*, or 21 Hours *(On-Demand)*

Price: £1,345 *(Face-to-Face & Remote-Live)*, or £1,145 *(On-Demand)*

Discounts: For multiple course purchases, please contact us for applicable discounts.

Delivery Options: Perform training at your own pace via our on-demand training option or attend regularly scheduled live courses via remote-live attendance.

Students Will Learn

- Designing normalized table structures for relational databases
- Creating databases and tables
- Using primary and foreign keys
- Writing SQL queries
- Using inner and outer joins
- Using set operators (`UNION`, `INTERSECT`, `EXCEPT`)
- Using DML for `SELECT`, `INSERT`, `UPDATE`, `DELETE`
- Using subqueries
- Using triggers and stored procedures
- Using aggregate functions to return totals and subtotals
- Using table expressions

Course Description

This SQL programming course teaches students relational database fundamentals and SQL programming skills. Topics covered include relational database architecture, database design techniques, and simple and complex query skills. This class is intended for analysts, developers, designers, administrators, and managers new to the SQL programming language. Upon completion, participants will understand SQL functions, join techniques, database objects and constraints, and will be able to write useful SELECT, INSERT, UPDATE and DELETE statements. Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

Students who require coverage of Microsoft-specific features including T-SQL data types, stored procedure enhancements, intrinsic functions, T-SQL cursors and table-valued functions should attend the Microsoft Transact-SQL Programming course rather than this course.

Course Prerequisites

General computer knowledge.

Course Overview

Relational Database Fundamentals

- Overview of Relational Database Concepts
- Relational Databases and Relational Database Management Systems
- Data Normalization
- DDL Syntax

Creating a Database

- Database Development Methodology Overview
- Building a Logical Data Model
    - Identifying Entities and Attributes
    - Isolating Keys
    - Relationships Between Entities
    - Creating Entity-Relationship Diagrams
- Transforming to Physical Design
    - Migrating Entities to Tables
    - Selecting Primary Keys
    - Defining Columns
    - Enforcing Relationships with Foreign Keys
- Constructing the Database Using DDL
    - Creating Tables, Indexes, Constraints and Views
    - Dropping Tables, Indexes, Constraints and Views
    - Modifying Tables, Indexes, Constraints and Views

Advanced Query Techniques

- Inner Joins
- Outer Joins (Left, Right, Full)
- Performing Self-Joins
- Subqueries
    - Simple
    - Correlated
- Using the `EXISTS` Operator
- Tips for Developing Complex SQL Queries
- Using Aggregate Functions
    - `AVG`
    - `COUNT`
    - `SUM`

Writing Basic SQL Queries

- Displaying Table Structures
- Retrieving Column Data From a Table or View
- Selecting Unique Values
- Filtering Rows Using the `WHERE` Clause
- Sorting Results Using `ORDER BY`
- Joining Multiple Tables
- Using Column and Table Aliases

Manipulating Query Results

- Using Row Functions
    - Character
    - Numeric
    - Date and Time
    - Data Conversion (`CAST` and `CONVERT`)
- Using the `CASE` Function
- Handling Null Values

Manipulating Table Data Using SQL's Data Manipulation Language (DML)

- Inserting Data into Tables
- Updating Existing Data
- Deleting Records
- Truncating Tables
- Implementing Data Integrity with Transactions
    - Beginning Explicit Transactions
    - Committing Transactions
    - Rolling Back Transactions

- MIN
  - MAX
- Performing Set Operations

  - UNION
  - INTERSECT
  - EXCEPT/MINUS
- Aggregating Results Using GROUP BY
- Restricting Groups with the HAVING Clause
- Creating Temporary Tables

User-Defined Functions

- Definition and Benefits of Use
- CREATE FUNCTION

  - Syntax
  - RETURN Clause and the RETURNS Statement
  - Scalar vs. Table Functions
- Comparison with Stored Procedures
- Returning Scalar Values and Tables
- ALTER and DROP FUNCTION

Stored Procedures

- Definition and Benefits of Use
- CREATE PROCEDURE

  - Syntax
  - Variables and Parameters
- Control of Program Flow
- ALTER and DROP PROCEDURE
- Implementation Differences

Triggers

- Definition and Benefits of Use
- Alternatives (e.g., Constraints)
- CREATE TRIGGER

  - Syntax
  - Trigger Types
- "Inserted" (or "NEW") and "Deleted" (or "OLD") Tables
- Event Handling and Trigger Execution
- ALTER and DROP TRIGGER

Working with Table Expressions

- Overview of Table Expressions
- Working with Views
- Using Derived Tables
- Common Table Expressions
- Table-Valued Functions