

SQL Programming Language: A Comprehensive Hands-On Introduction - 3 Days

Course 925 Overview

- You Will Learn How To**
- Write SQL code based on ANSI/ISO standards to build and maintain database structures
 - Update database content with SQL and transaction handling
 - Retrieve data from single or multiple tables
 - Process data with row and aggregate functions
 - Manipulate data with correlated and noncorrelated subqueries
 - Apply views to break down problems and enhance security
- Course Benefits** SQL forms the cornerstone of all relational database operations. The ability to write the SQL language is essential for those who develop database applications. This course provides a solid foundation of the SQL programming language that enables you to build, query and manipulate databases. Working in Oracle or SQL Server databases throughout this course, you compare the ANSI/ISO standard with the SQL implementations of these two common database products.
- Who Should Attend** This course is valuable for anyone who needs to learn SQL programming. An understanding of relational database and basic programming concepts is helpful.
- Hands-On Training** In this course, you gain hands-on experience programming with SQL in Oracle databases and Microsoft SQL Server environments. Exercises include:
- Creating and modifying tables, constraints and indexes
 - Modifying table contents
 - Retrieving data from tables
 - Joining multiple tables
 - Applying row and aggregate functions
 - Embedding subqueries within statements

SQL Programming Language: A Comprehensive Hands-On Introduction - 3 Days

Course 925 Outline

SQL Overview

- Outlining the importance of SQL as the cornerstone of all database activity
- Applying the ANSI/ISO standards
- Describing the fundamental building blocks: tables, columns, primary keys and foreign keys
- Defining terminology

Building the Database Schema

Creating tables and columns

- Comparing data types across platforms
- Building tables with CREATE TABLE
- Modifying table structure with ALTER TABLE
- Adding columns to an existing table
- Increasing column size
- Removing tables with DROP TABLE

Protecting data integrity with constraints

- Defining constraint types
- Guaranteeing uniqueness with primary key constraints
- Enforcing referential integrity with foreign key constraints
- Imposing business rules with check constraints
- Enabling and disabling constraints
- Removing constraints with ALTER TABLE

Improving performance with indexes

- Expediting data retrieval with indexes
- Recommending guidelines for index creation

Manipulating Data

Modifying table contents

- Adding table rows with INSERT
- Changing row content with UPDATE
- Removing rows with DELETE

Applying transactions

- Atomic Consistent Isolated Durable (ACID) rules
- Controlling transactions with COMMIT and ROLLBACK

Writing Single Table Queries

- Retrieving data with SELECT
- Including columns and expressions in query results
- Restricting rows with the WHERE filter

- Sorting the result with ORDER BY
- Handling NULL values in expressions
- Avoiding NULL value pitfalls in filter conditions

Querying Multiple Tables

Applying the ANSI/ISO standard join syntax

- Matching related rows with INNER JOIN
- Including nonmatched rows with OUTER JOIN
- Creating a Cartesian product with CROSS JOIN
- Joining a table to itself

Combining results with set operators

- Stacking results with UNION
- Identifying matching rows with INTERSECT
- Utilizing EXCEPT to find nonmatching rows

Employing Functions in Data Retrieval

Processing data with row functions

- Solving mathematical problems with functions
- Manipulating text strings
- Converting date/time presentation
- Conditional formatting with the CASE expression
- Utilizing the CASE expression to simulate IF tests
- Dealing with NULL values

Performing analysis with aggregate functions

- Summarizing data using SUM, AVG and COUNT
- Finding the highest and lowest values with MAX and MIN
- Defining the summary level with GROUP BY
- Applying filter conditions with HAVING

Constructing Nested Queries

Applying subqueries in filter conditions

- Correlated vs. noncorrelated subqueries
- Embedding subqueries in several levels
- Testing the existence of rows
- Single row vs. multirow subqueries

Including subqueries in expressions

- Placing subqueries in the column list
- Creating complex expressions containing subqueries

- Handling subqueries that return no rows

Developing In-Line and Stored Views

Breaking down complex problems

- Selecting data from a query result set
- Subqueries in the FROM clause

Creating views in a database

- Building reusable code
- Refining user access privileges