Intermediate SQL

Course Summary

Description

This course covers ANSI/ISO standard SQL, with examples in SQL Server, Oracle, DB2 LUW and z/OS, and MySQL, with emphasis on SQL Server and Oracle. Labs can be done in any of those database environments.

Objectives

After taking this course, students will be able to:

- Sample a very large database to have accurate knowledge of the structure and contents
- Use several tools to monitor and possibly improve the performance of complex queries on large databases, including using the query execution plan, optimizer hints, and other tools
- Know when and how to use temporary tables, query-scope tables, Common Table Expressions, and subqueries of various types to solve complex problems
- Handle all logical and performance issues with joins and subqueries on large tables
- Handle recursive relationships with recursive With, the Oracle Connect By, and/or the SQL Server HierarchyID data type
- Handle character data with built-in functions, Soundex, full-text searches, and Regular Expressions
- Use Case logic to control how sorting, grouping and other operations work
- Summarize data, including handling missing values, creating pivot reports using Pivot or Case logic, using Rollup and Cube, and using basic analytic (OLAP) functions

Topics

- Introduction
- Techniques for solving complex problems
- Using multiple tables
- Text handling issues
- Case logic
- SQL Summarization
- Stored Procedures

Audience

This course is designed for application developers.

Prerequisites

Before taking this course, students should take an SQL Basics course, or have equivalent work experience, and had several weeks of experience using SQL.

Duration

Two days

Intermediate SQL

Course Outline

I. Introduction

- A. The ISO Standard
- B. Learning about the database structure
- C. Sampling a large database
- D. Performance considerations
  - 1. Minimize locking
  - 2. Maximize index use
  - 3. Statistics and Access Paths
  - 4. Viewing a query execution plan
  - 5. Getting performance help
  - 6. Optimizer hints

II. Techniques for solving complex SQL problems

- A. Creating temporary and query-scope tables and indexes
- B. Review of inserting, updating, and deleting
- C. Common Table Expressions (CTEs)

III. Using multiple tables

- A. Intersection: inner joins
- B. Join performance considerations
- C. Inner joins vs. Outer joins
- D. Left, Right, and Full outer joins
- E. Recursive joins and complex relationships
  - 1. More complex relationships
  - 2. The "bill of materials" problem
  - 3. Recursive With
  - 4. Oracle's Connect By
  - 5. SQL Server HierarchyID data type
- F. Difference: Not Exists and Not IN
- G. Set operators: Union, Union All

IV. Text handling issues

- A. Text handling functions
- B. Unicode
- C. Regular Expressions
- D. Creating a full-text index, using the Contains function

V. Case logic

- A. Review of "simple" When clauses
- B. Review of "searched" When clauses
- C. Case in other contexts
  - 1. Within functions and expressions
  - 2. In a From clause
  - 3. In an Order By clause
  - 4. In an Update
  - 5. With nested Selects
  - 6. In Group By

VI. SQL Summarization

- A. Review Group By
- B. Filtering groups with Having
- C. Additional summarizing examples
  - 1. Using Pivot clause
  - 2. Using Case logic
- D. Analytic (OLAP) functions
- E. Other considerations: logical and performance issues

VII. Stored Procedures

- A. Overview of stored procedures
- B. Basic stored procedures

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically