

SQL Server Transact-SQL®-programmering - 4 dagar

kurser 532

- Du får lära dig att**
- Utveckla databasapplikationer för SQL Server-databaser med Transact-SQL
 - Skapa modulär kod med hjälp av lagrade procedurer
 - Utveckla återanvändbar kod med skalära och tabellvärderade funktioner
 - Hantera körtidsfel i Transact-SQL för att skapa stabil mjukvara
 - Granska dataförändringar med hjälp av AFTER-triggrar
 - Hantera samstämmighet med hjälp av transaktioner och låsalternativ

Sammanfattning Transact-SQL är en väsentlig del av SQL Server. Transact-SQL-funktioner ger användare möjlighet att skapa skalbara, distribuerade applikationer som motsvarar kraven från moderna organisationer. Kursen ger erfarenhet av att skapa lagrade procedurer och triggrar och att utveckla T-SQL som kan utnyttja SQL Server maximalt.

Vem bör delta Databasutvecklare, administratörer och analytiker som vill utöka sina kunskaper om SQL Server 2005 eller 2008. Erfarenhet av SQL Server i nivå med kurs 137, "SQL Server 2008: omfattande introduktion", förutsätts.

Praktiska övningar En rad praktiska övningar ger erfarenhet av att utveckla Transact-SQL, bland övningarna kan nämnas:

- Lägga till en kolumn i en databastabell med hjälp av ALTER TABLE
- Skicka data in i en lagrad procedur med hjälp av parametrar
- Modifiera data i en transaktion
- Hindra fel med TRY...CATCH
- Anropa en användardefinierad funktion i en SQL-sats
- Skriva triggrar för att genomföra avancerad validering
- Härleda förändringar i metadata med hjälp av DDL-triggrar

SQL Server Transact-SQL[®]-programmering - 4 dagar

kurser 532

Arkitekturen i SQL Server

- SQL Server: översikt över versioner
- Introduktion till verktygen
- SQL Server Management Studio

Hantera tabeller med DDL

Skapa scheman

- Hantera scheman
- Använda schemareferenser eller standardschemat
- Gömma scheman med synonymer

Skapa tabeller

- Välja lämpliga SQL Server-datatype
- Skapa tabeller med CREATE TABLE

Lägga till restriktioner

- Stärka exklusiviteten med PRIMARY KEY och UNIQUE-restriktioner
- Validera relationer med FOREIGN KEY

Hämta data med lagrade procedurer i Transact-SQL

Processer för batch och lagring

- Minimera nätverkstrafik med batchar och procedurer
- Kompilering och exekvering av lagrade procedurer
- Använda skalära och aggregerade funktioner

Välja data

- Utveckla lagrade procedurer som extraherar data från flera olika servrar
- Använda dynamiska frågor med hjälp av OPENROWSET och OPENQUERY
- Använda fjärrprocedurer
- Kombinera resultat från flera databaser
- Använda RETURN-värden från lagrade procedurer

Deklarera variabler och parametrar

- Skapa och använda lokala variabler
- Skicka indata- och utdataparametrar
- Fråga efter globala variabler

Anropa inbyggda skalära funktioner

- Konvertera data med hjälp av CAST och CONVERT
- Ordna data med rankingfunktioner

Underhålla data

Modifiera data

- Infoga, uppdatera och ta bort data
- Säkerställa datakonsistens med transaktioner och distribuerade transaktioner

- Hantera samtidighet med isoleringsnivåer
- Grundläggande om SQL Server-låsning
- Undvika blockeringsproblem med "read-committed snapshot isolation"
- Hantera låsningar med hjälp av ledtrådar

Programmera procedural statements

- Implementera villkor med IF...ELSE
- Loopa med WHILE och GOTO
- Skapa kodblock med BEGIN...END
- Avlusning med PRINT
- Returnera data med RETURN
- Avlusa T-SQL i Visual Studio

Hantera fel

- Uppmärksamma klienten på problem med RAISERROR
- Hindra fel med TRY...CATCH
- Hantera öppna transaktioner när ett undantag sker

Producera resultat-set på serversidan

- Skapa och använda temporära tabeller
- Bearbeta rader på servern med cursors
- Utnyttja tabellvariabler

Utveckla vyer, funktioner och triggrar

Lagra förfrågningar på servern

- Dölja komplexitet med vyer
- Avhjälpa affärsproblem med hjälp av multistatement tabellvärdefunktioner

Skapa användardefinierade funktioner

- Beräkna värden med skalära funktioner
- Bearbeta flera olika rader returnerade från en tabellvärdesfunktion
- Dra nytta av schemabindning

Formulera triggrar

- INSTEAD OF kontra AFTER-triggrar
- Upptäcka radförändringar med hjälp av infogade och borttagna tabeller
- Spåra förändringar i metadata med hjälp av DDL-triggrar
- Granska användaråtkomst med hjälp av en LOGON-trigger
- Spåra förändringar i data med hjälp av satsen OUTPUT