

Relationsdatabaser: design, verktyg och metoder - 4 dagar

kurser 382

- Du får lära dig att**
- Konstruera, bygga och integrera en effektiv relationsdatabas
 - Använda ett CASE-verktyg för att fånga strukturen hos en befintlig databas
 - Utveckla en datamodell för en tillämpning
 - Skapa en effektiv och stabil databaskonstruktion genom normalisering
 - Bygga en relationsdatabas från en logisk databaskonstruktion
 - Söka data i en relationsdatabas med hjälp av enkla SQL-frågor
- Sammanfattning** Relationsdatabaserna driver både applikationer viktiga för företaget och webb-förberedda applikationer. Att skapa en databasdesign som på ett korrekt och komplett sätt motsvarar användarkraven är viktigt för att nå framgång. Kursen ger omfattande grundkunskaper om hur man utformar, designar och arbetar med relationsdatabaser, vilket gör att du sedan kan delta i utvecklingsprocessen och effektivt använda relationsdatabaser i din miljö.
- Vem bör delta** Alla som är engagerade i konstruktion, uppbyggnad och användning av relationsdatabaser, implementering av databastillämpningar eller ledning av projekt för utveckling av databaser.
- Praktiska övningar** I en rad praktiska övningar och seminarier i små grupper får du genom en kontinuerlig fallstudie färdigheter att analysera, konstruera, bygga och använda en relationsdatabas. Övningarna omfattar bl a:
- Använda ett CASE-verktyg för att analysera en befintlig databas
 - Utveckla datamodeller som representerar olika datamiljöer
 - Skapa en logisk datamodell som identifierar entiteter, attribut och relationer
 - Använda normaliseringsregler för dataattribut
 - Använda ett CASE-verktyg för att generera SQL
 - Bygga en databas som motsvarar den logiska databasuppbyggnaden
 - Konstruera enkla SQL-frågor för att få åtkomst till databasen

Relationsdatabaser: design, verktyg och metoder - 4 dagar

kurser 382

Introduktion

Översikt över DBMS-teknologi

- Viktiga begrepp och termer
- Hur data hämtas, organiseras och lagras
- Vikten av affärsregler
- Använda databaser
- Utvecklingsprocessen bakom databaser

DBMS och relaterade användarverktyg

- Frågespråk
- Verktyg för utveckling av frågor och applikationer
- CASE-verktyg för analys och konstruktion av databaser

Hur RDBMS fungerar

Relationsteknologins grunder

- Relationsdatabasens struktur
- Tabeller, attribut och relationer
- Primära och främmande nycklar
- Integritetsregler
- Datamanipulation: selektion, projektion, sammanslagning, union, snitt, skillnad

Komponenter i ett RDBMS

- En integrerad, aktiv datakatalog
- Frågeoptimeraren
- En maskin som hanterar datan
- Front-end-verktyg för ökad tillgänglighet

Konstruktion av relationsdatabaser

Steg-för-steg-metoder

- Utveckling av den logiska datamodellen
- Överföra datamodellen till relationsmodellen
- Specificering av integritetsreglerna
- Definition av data i datakatalogen

Relationsmodellering

- "Entities", attribut och identifierare
- Relationsbeskrivning: en-till-en, en-till-flera, flera-till-flera
- Obligatoriska och frivilliga relationer
- Riktlinjer för välformade E-R-diagram
- Lösa många-till-många-relationer för implementering
- Riktlinjer för ett välformulerat ER-diagram

Normalisera data för att bygga tabeller

- Varför/varför inte normalisera?
- Undvika anomalier vid uppdatering
- Identifiera funktionella beroenden
- Använda regler för normalisering

Arbeta med ett CASE-verktyg

- Databaskonstruktion och -dokumentation

- Generera den SQL som behövs för att bygga databasen
- "Reverse engineering" för att utnyttja konstruktionen hos en befintlig databas

Fysisk databaskonstruktion

- Gruppera och koppla tabeller mot diskfiler för bättre prestanda och dataunderhåll
- Fragmentera stora tabeller
- Planerad denormalisering kontra oavsiktlig denormalisering
- Indexering för prestanda och integritet

Bygga upp och avfråga

relationsdatabaser

Grunderna för SQL

- Ett dynamiskt språk i utveckling
- ANSI- och ISO-standarder

Använda relationsdatabasen

- Definiera databasen och dess objekt: tabeller, nycklar, vyer och index
- Deklarera integritetsbegränsningar
- Ändra strukturer och begränsningar

Få åtkomst till databasen med SQL

- Skapa frågor för att få fram exakt den önskade informationen
- Sammanfoga tabeller för att hämta relaterad data
- Uppdatera data samtidigt som databasintegriteten bibehålls

Skapa och använda vyer

- Definiera vyer för att förenkla och öka säkerheten
- Anpassa fönster för databasen
- Skapa frågor och uppdatera via vyer

Databasernas utveckling

Förstärka företagets regler för dataintegritet

- Definiera deklarativa begränsningar
- Serverprogrammering: i Java och procedurrella språk

Trender

- Modellbygge i analys och konstruktion
- Fokusering på affärsregler
- Skapa en intelligent server
- Använda lagrade procedurer och triggrar