

Data Definition Language (DDL) og Data Manipulation Language (DML)

Du har allerede arbejdet grundigt med database-design i Fase A, hvor du udviklede en domænemodel og normaliserede dine data. Dette har givet dig en solid grundforståelse for, hvordan man strukturerer data effektivt. Nu er tiden kommet til at omsætte dette design til en fungerende database i Fase B. I denne fase vil du arbejde med at implementere din databasedesign ved hjælp af SQL og C#-programmering. Gennem en række øvelser vil du få praktisk erfaring med SQL's DDL og DML samt integrere databasen med en applikation gennem repository-mønstret i C#.

Øvelse 1: Terminologi

Forventning: At sikre, at du har en klar forståelse af de grundlæggende begreber og typer inden for SQL, som er nødvendige for at arbejde med databaseimplementering.

Tidsramme: 30 minutter

Øvelse 1.1:

Brug Ordet rundt til at reflektere over begreberne "SQL", "SQL-statement", "Auto-increment", DDL, DML, "SQL Connection", "SQL Command", "ExecuteReader", "ExecuteNonQuery", "ExecuteScalar" og "ConnectionString".

SQL: Structured Query Language, bruges til at kommunikere med databaser.

SQL-statement: En kommando skrevet i SQL, der udfører en bestemt opgave som at oprette, ændre, hente eller slette data.

Auto-increment: En funktion der automatisk øger værdien af en numerisk kolonne, normalt brugt til primærnøgler.

DDL: Data Definition Language, bruges til at definere eller ændre database-strukturen (f.eks. CREATE, ALTER).

DML: Data Manipulation Language, bruges til at manipulere data i tabellerne (f.eks. INSERT, UPDATE, DELETE).

SQL Connection: Bruges i applikationer for at etablere en forbindelse til databasen.

SQL Command: Bruges til at eksekvere SQL-kommandoer fra en applikation.

ExecuteReader: Bruges til at køre SQL-kommandoer, som henter data (f.eks. SELECT).

ExecuteNonQuery: Bruges til at eksekvere kommandoer, der ikke returnerer data, men ændrer strukturen eller indholdet (f.eks. INSERT, UPDATE, DELETE).

ExecuteScalar: Bruges til at returnere en enkelt værdi fra en SQL-forespørgsel.

ConnectionString: En streng, der indeholder de nødvendige oplysninger for at kunne oprette forbindelse til en database (serveradresse, databasenavn, brugernavn og kodeord).

Øvelse 1.2:

Udfyld tabellen nedenfor med en kort beskrivelse af formål og type for hver SQL-sætning.

SQL Statement	Formål	Type
CREATE DATABASE	Opretter en ny database	DDL
CREATE TABLE	Opretter en ny tabel i databasen	DDL
DROP	Sletter en database eller en tabel	DDL
ALTER	Ændrer en eksisterende tabel	DDL
TRUNCATE	Fjerner alle rækker fra en tabel, men beholder strukturen	DDL
INSERT	Indsætter nye rækker i en tabel	DML
UPDATE	Opdaterer eksisterende rækker	DML
DELETE	Sletter rækker fra en tabel	DML

Sørg for, at alle får mulighed for at tale.

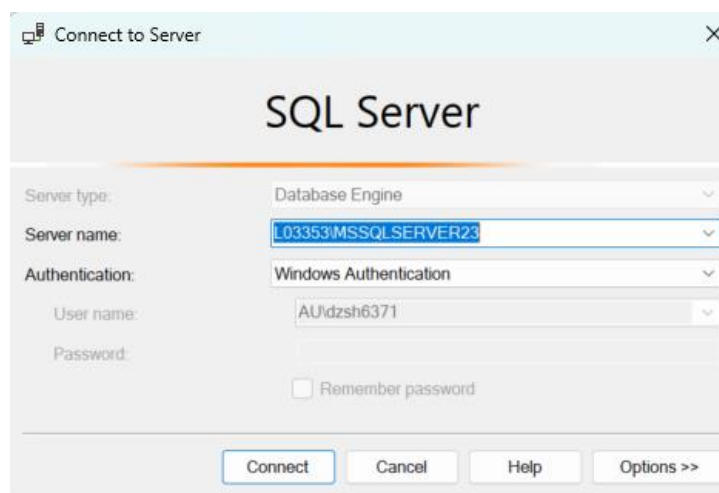
Øvelse 2: Klargøring af SQL Server Management Studio (SSMS)

Forventning: I denne øvelse vil du lære at konfigurere SQL Server Management Studio (SSMS) og klargøre den til brug i realisering af UML-databasemodellen. Der forventes, at SSMS er allerede installeret på din maskine.

Tidsramme: 15 minutter

Øvelse 2.1: Login i SSMS

Åbn SQL Server Management Studio, og sikr dig, at der er valgt "Database Engine" som "Server type". Bekræft også, at "Authentication" er sat til "Windows Authentication".



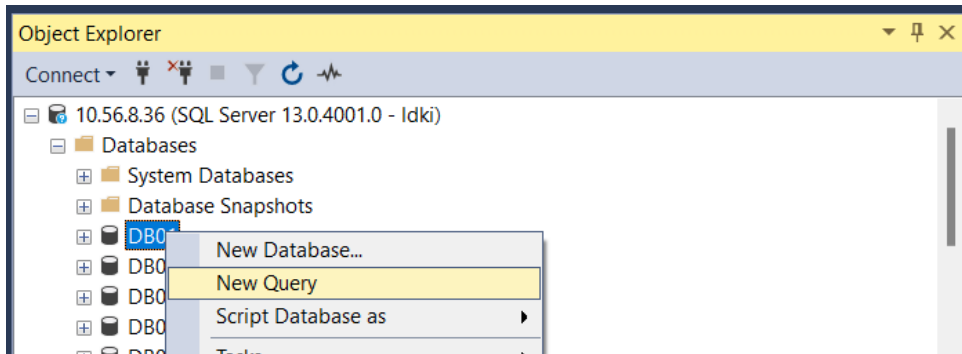
Tryk på "Connect"-knappen. Herefter burde du være i stand til at se serveren i din "Object Explorer".

Øvelse 2.2: Opret en database

Du kan oprette en ny database ved at højreklikke på Databases og vælge "New Database" og så navngive databasen.

Men i næste øvelse, vil du skrive SQL sætninger til at realisere UML-databasemodellen i SSMS ved at folde

”Databases” ud ved at tykke på det lille kryds, og højreklik på den database, du vil oprette og vælg ”New Query”, som på billedet nedenunder:



Nu er du klar til at indsætte SQL statements.

Øvelse 3: DDL – Implementering af Database for Webshop

Forventning: I denne øvelse skal du implementere den database, du designede og normaliserede i Fase A, ved hjælp af SQL's Data Definition Language (DDL). Benyt din viden fra forberedelse i DDL læringsobjektet til udformningen.

Tidsramme: 30 minutter

Udfør følgende:

- Opret en ny database til din webshop ved hjælp af CREATE DATABASE sætning.
- **Hvert teammedlem** opretter en eller flere tabeller i databasen, herunder primær- og fremmednøgler ved hjælp af CREATE TABLE. Husk AUTO INCREMENT constraint ved nødvendighed. Sørg for at oprette de tabeller, du har lavet relationsskemaer for.
- Brug ALTER TABLE, hvis du manglede at definere en attribut eller en nøgle.
- Diskutere jeres individuelt sætninger og saml dem i en query. Her burde I at have alle ”CREATE TABLE” sætninger samlet et sted.
- Benytte Query til at udføre jeres SQL ”CREATE TABLE”-sætninger. Åben jeres ”Query” og indsæt jeres SQL ”CREATE TABLE”-sætninger.
- **Bemærk:** Du skal genopfriske din Object Explorer (inde i SSMS), før du kan se jeres oprettede tabel. Og du skal genopfriske din Local Cache for at få IntelliSense i din Query: Edit → IntelliSense → Refresh Local Cache
- Kopiere de sætninger i en tekstfil (som backup).

Øvelse 4: DML – Manipulation af Data for Webshop

Forventning: Denne øvelse fokuserer på Data Manipulation Language (DML), hvor du skal arbejde med at tilføje, opdatere og slette data i din webshop-database.

Tidsramme: 30 minutter

Gennemfør følgende opgaver i SQL Server Management Studio (SSMS):

- Hvad er syntaksen af INSERT INTO-sætning?

```
INSERT INTO Participant (Email, FirstName, LastName, ImagePath, SemesterId)
VALUES ('john.doe@example.com', 'John', 'Doe', '/images/johndoe.png', 2);
```

SQL

- Brug INSERT INTO til at tilføje data i tabellerne til din webshop-database. **Hvert teammedlem** tilføjer data i denne tabel, man har oprettet. Du kan bruge generativ AI til at oprette dummy data.
- Tilføj de INSERT INTO-sætninger til den tekstfil, der indeholder CREATE sætninger. Nu har du databasens struktur og data samlet et sted som backup.
- Brug UPDATE til at opdatere produktpriser til nuværende priser + 3%.
- Brug DELETE til at slette kunder, der bor i en bestemt by eller forældede ordrer.

Øvelse 5: Implementering af Repository Pattern i C# (Face C)

Forventning: Nu hvor din database er operationaliseret og fyldt med data, er det tid til at integrere den med din applikation. I denne øvelse skal du implementere repository pattern i din webshop-applikation ved hjælp af C#.

Tidsramme: 120 minutter

- **Hvert teammedlem** implementer repository-klasse for denne klasse man har implementeret i modellaget, for at håndtere CRUD-operationer (Create, Read, Update, Delete) mod din webshop-database. Sørg for at de forudsætningerne for at arbejde med Fase C i din Visual Studio-applikation er på plads. Genlæs læringsobjektet, hvis du støder på nogle udfordringer.
- Saml og test dine repositories for at sikre, at de interagerer korrekt med databasen.

Øvelse 6: Vidensdeling kl. 14:30

Fremvis din kørende database og demonstrer, hvordan din C#-applikation interagerer med databasen gennem repository-mønstret.

Tidsramme: 30 minutter