

LABORATION 2

Skapa och modifiera innehåll i en relationsdatabas

Övningen syftar till att lära sig hur data i en databas kan hanteras (jmf SQL DML) för att tillgodose verksamhetens informationsbehov.

Utgå från den databas ni skapade i laboration 1. Ni ska nu skapa queries genom att använda verktyget SQL Server Management Studio. För mer information om hur queries kan konstrueras se t.ex. "SQL Lathund VT-2010", "SQL Server exempel" eller kap. 6-7 i kurslitteraturen.

Varje query som ni skapar i arbetsuppgiften nedan **ska vid inlämningen vara redovisade på följande sätt** (observera att ni ska skriva era queries i **Backus Naur Form** – se s.138 i Connolly & Begg):

I. *[Fråga i klartext, d.v.s. på "vanlig svenska" (pseudokod)]*

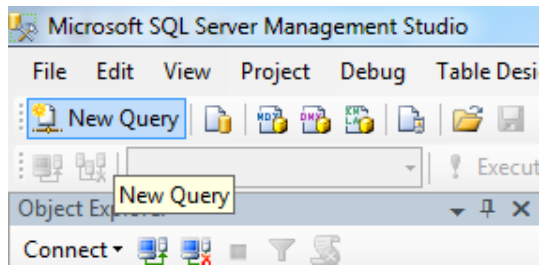
II. *[SQL kod för queryn]*

III. *[Här klistrar ni in en skärmdump över det resultat ni fått efter att ha exekverat queryn. Om ni får ett resultat i stil med <1 row affected> tar ni istället skärmdump på tabellens data.]*

Arbetsuppgift

Skapa diverse frågor:

- 1 uppdateringsexempel (UPDATE).
- 1 borttagningsexempel (DELETE).
- 1 insättningsexempel (INSERT).
- 1 informationsutdragsexempel (SELECT).
- 3 avancerade informationsutdragsexempel som hämtar information ifrån flera olika tabeller (SELECT). Förslagsvis kan ni använda er av joins för detta. Läs mer om hur joins fungerar här: http://www.w3schools.com/sql/sql_join.asp
- 3 queries där ni använder er av aggregat eller andra funktioner, t.ex. SUM, COUNT, AVG eller MAX (se Connolly & Begg s. 149-151).
- 1 query som är nestad (subquery).
- 1 vy (VIEW).
- 1 stored procedure (som t.ex. uppdaterar kundtabellen med en ny kund (kunddata som in-parameter)) och skapa en trigger (t.ex. en som utlöses av samma stored procedure och skriver ut att en ny kund lagts till i tabellen). Notera att det måste gå att ange ny data som ska läggas in med er stored procedure varje gång den exekveras, detta ska alltså inte hårdkodas i proceduren.



Figur 1. I SQL Editor toolbar, välj New Query

Lycka till!