

# LABORATION 2

Designa en databas för en bokhandel

## Inledning

Vi vill bygga en databas som kan användas till en bokhandel med flera butiker.

Du får i uppgift att designa den i SQL-server med nycklar, relationer, integritetsvillkor och ett ER-diagram. Nedan finns några minimikrav på tabeller och vyer vi vill ha med, men därutöver vill vi att du lägger till ytterligare ett antal tabeller som du tror att vi får användning för. Vi vill även att du populerar databasen med lämpliga testdata för demonstrationssyfte.

## Tabell: "Författare"

I tabellen författare vill vi ha en "Identitetskolumn" (identity) kallad ID som PK. Därutöver vill vi ha kolumnerna: Förnamn, Efternamn och Födelsedatum i passande datatyper.

## Tabell: "Böcker"

I tabellen böcker vill vi ha ISBN13 som primärnyckel med lämpliga constraints. Utöver det vill vi ha kolumnerna: Titel, Språk, Pris, och Utgivningsdatum av passande datatyper. Sist vill vi ha en FK-kolumn "FörfattareID" som pekar mot tabellen "Författare". Utöver dessa kolumner får du gärna lägga till egna med information som du tycker kan vara bra att lagra om varje bok.

## Tabell: "Butiker"

Utöver ett identity-ID så behöver tabellen kolumner för att lagra butiksnamn samt adressuppgifter.

## Tabell: "LagerSaldo"

I denna tabell vill vi ha 3 kolumner: ButikID som kopplas mot Butiker, ISBN som kopplas mot böcker, samt Antal som säger hur många exemplar det finns av en given bok i en viss butik. Som PK vill vi ha en kompositnyckel på kolumnerna ButikID och ISBN.

## Övriga tabeller

De 4 tabellerna som är specificerade ovan är ett minimum att implementera. Utöver det ska du dock lägga till *ytterligare minst 2 tabeller (minst 4 för VG)* med information som kan vara lämplig att lagra för bokhandelns syfte, och skapa nycklar och relationer mellan dessa.

Exempel på tabeller skulle kunna vara "Kunder", "Ordrar" och "Förlag". Vad behöver man spara för uppgifter i de olika tabellerna? Vad för andra tabeller kan vi behöva? Hur kommer de vara relaterade till varandra och våra 4 tidigare tabeller? Eventuellt kan vi behöva uppdatera våra 4 tidigare tabeller med kolumner för att relatera till de nya.

## Demodata

För demonstration vill vi ha minst 3 butiker, 4 författare, 10 boktitlar med tillhörande saldo. I tabeller du själv skapar lägger du till lämpliga testdata så att vi ska kunna demonstrera upplägget.

## Vy: "v\_TitlarPerFörfattare"

Vi vill även ha en vy som sammanställer data från tabellerna. Vyn ska innehålla följande fyra kolumner (med en rad per författare):

"**Namn**" – Hela namnet på författaren. "**Ålder**" – Hur gammal författaren är.

"**Titlar**" – Hur många olika titlar vi har i "Böcker" av den angivna författaren.

"**Lagervärde**" – Totala värdet (pris) för författarens böcker i samtliga butiker.

Exempeldata: `select top 1 * from v_TitlarPerFörfattare`

Namn	Ålder	Titlar	Lagervärde
Emma Askling	43 år	3 st	4182 kr

### SP: "usp\_FlyttaBok" (svårare)

Vi vill att det ska finnas en stored procedure i databasen som på ett *integritetssäkert* sätt flyttar exemplar av böcker från en butik till en annan. Den ska ta en parameter med ID på butik som man flyttar från, en parameter med ID på butik man flyttar till, ett ISBN, samt en valbar (optional) fjärde parametersom tar antal exemplar man vill flytta (default = 1). Vilka hänsyn behöver vi ta för dataintegritet?

TIPS: se på transaktioner och hur de fungerar

### Flera författare på samma bok (svårare)

Uppdatera relationen författare/böcker från en – många till en många –många-relation, så att vi kan lagra böcker med flera författare i systemet.

TIPS: läs på om *junction tables* som används för att lösa upp många – många-relationer.

### En extra vy (svårare)

Skapa ytterligare en vy som sammanställer för bokhandeln relevant information från minst två av de tabeller som du lagt till under rubriken "Övriga tabeller". Nu bör vyn innehålla någon form att aggregering/gruppering. Man ska även skriva med en kommentar (i koden eller vid inlämning) som motiverar hur bokhandeln kan ha nytta av den sammanställda vyn.

### Övrigt

Tänk på att välja lämpliga datatyper och sätta integritetsvillkor där du anser det lämpligt. Databasen ska innehålla ett ER-diagram som visar relationerna mellan alla tabeller på ett tydligt sätt.

### Redovisning

Skriv queries som löser problemen ovan och spara i en .sql fil som du döper till ditt för- och efternamn. Exempel: "KalleKarlsson.sql".

Komplettera din SQL-fil med de nödvändiga testerna, så som:

- `SELECT * FROM <alla tabeller>`
- `SELECT * FROM <alla vyer>`
- `EXEC <usp_namn> [parametrar]`

Ladda upp filen för inlämning på Omniway

Kan lösas individuellt eller **i par**. Om ni jobbat i par så skriv det som en kommentar när ni lämnar in på Omniway. Ange vem parkamraten är.

## Betygskriterier

Denna uppgiften är inte betygsgrundande i denna kursen.

## Tips

Det kan vara besvärligt att hitta på testdata. Ett alternativ är att låta en dator generera testdata, som sedan kan klistras in i databasen. En väldigt bra sajt för att generera testdata är Mockaroo, <https://www.mockaroo.com/>. Där kan man få ut testdata i form av ett SQL-skript, som kan köras i SSMS.

Ett annat tips är att modellera databasen i ett verktyg för E/R-diagram, till exempel [Lucidchart](#). Då kan du låta verktyget generera det nödvändiga skriptet för att skapa tabeller, primärnycklar och främmande nycklar.