

# Designa en databas för en bokhandel

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, integritetsvilkor 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 "Identetskolumn" (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 för att vi ska kunna demonstrera upplägget.

### Vy: "TitlarPerFörfattare"

Vi vill även ha en vy som sammanställer data från tabellerna. Vyn ska innehålla följande 4 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.

Exempel data: "select top 1 \* from TitlarPerFörfattare"

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

## **SP: "FlyttaBok" (VG)**

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 parameter som tar antal exemplar man vill flytta (default = 1). Vilka hänsyn behöver vi ta för dataintegritet?

## **Flera författare på samma bok (VG)**

För VG, uppdatera relationen författare/böcker från one-many till en many-many relation, så att vi kan lagra böcker med flera författare i systemet.

## **En extra vy (VG)**

Skapa ytterligare en vy som sammanställer för bokhandeln relevant information från minst två av de tabeller som du lagt till under rubriken "Övriga tabeller". För att nå upp till VG-nivå 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 integritetsvilkor där du anser det lämpligt. Databasen ska innehålla ett ER-diagram som visar relationerna mellan alla tabeller på ett tydligt sätt.

## **Python**

Skriv pythonkod som ansluter till databasen med hjälp av SQLAlchemy. Från programmet ska användaren kunna göra fritextsökningar mot boktitlar och få upp en lista med böcker som matchar. Man ska även kunna se hur många exemplar som finns av böckerna i varje butik. Tänk på att endast ge programmet de rättigheter som behövs, samt skydda från SQL-injection.

## Redovisning

Uppgiften kan göras självständig eller i par. Om ni jobbar i par så ska ni skriva en kommentar på ithsdistans vem ni jobbat ihop med.

Ta en backup av den färdiga databasen och döp den till ditt för- och efternamn. Exempel: "FredrikJohansson.bak". Ladda upp filen för inlämning på ITHSdistans.

## Betygskriterier

För godkänt krävs:

- Databasen ska innehålla minst 6 (4+2) tabeller relevanta för bokhandeln.
- Tabeller ska vara normaliserade enligt 3NF.
- Alla relationer ska ha PK och FK nycklar, samt relationsvilkor som förhindrar FK's att peka på nycklar som inte existerar.
- Kolumner ska använda lämpliga datatyper för den typ av information som ska lagras.
- Databasen ska innehålla ett ER-diagram som visar relationerna mellan alla tabeller på ett tydligt sätt.
- Databasen ska innehålla lämpliga testdata för demonstration.
- Databasen ska innehålla vyn "TitlarPerFörfattare" enligt specifikation.
- Redovisa pythonkod, med funktionalitet enligt beskrivning ovan.

För väl godkänt krävs även:

- Databasen ska innehålla minst 8 (4+4) tabeller relevanta för bokhandeln. (Det ska vara minst 8 entiteter. Junktion tables räknas ej.)
- Databasen är gjord så böcker kan ha flera författare.
- Databasen ska innehålla SP "FlyttaBok" enligt specifikation.
- Databasen har ytterligare en vy som sammanställer för bokhandeln relevant information från minst två av de tabeller som du lagt till under rubriken "Övriga tabeller". Motivera hur bokhandeln kan ha nytta av den sammanställda vyn.