

# Övning 1: När ska man dela upp data i separata tabeller?

## Scenario A: Användarprofil

Du har en användare med följande information:

- Användarnamn
- Lösenord
- Förnamn
- Efternamn
- Email
- Gatuadress
- Postnummer
- Stad
- Land
- Bio (fritext om användaren)
- Profilbild (sökväg till bildfil)
- Registreringsdatum
- Senast inloggad
- Prenumerationsstatus
- Prenumerationens startdatum
- Prenumerationens slutdatum
- Betalningsmetod
- Faktureringsinformation

### Uppgift:

1. Titta på all denna information. Borde allt vara i samma tabell?
2. Finns det logiska grupperingar av data?
3. Vilken data ändras ofta vs sällan?
4. Vilken data är obligatorisk vs frivillig?
5. Rita två alternativa datamodeller:
  - En där allt är i samma tabell
  - En där data är uppdelad i flera tabeller
6. Diskutera för- och nackdelar med båda approacherna

## Scenario B: Produktinformation

En produkt har:

- Produktnamn
- Artikelnummer
- Pris
- Lagersaldo
- Beskrivning på svenska
- Beskrivning på engelska
- Beskrivning på tyska

- Tekniska specifikationer
- Dimensioner
- Vikt
- Tillverkare
- Tillverkarens artikelnummer
- Kategori
- Underkategori

#### **Uppgift:**

1. Identifiera data som:
  - Ändras ofta (dynamisk data)
  - Ändras sällan (statisk data)
  - Är språkberoende
2. Föreslå en uppdelning i tabeller baserat på din analys
3. Motivera dina val

## Övning 2: Förstå Many-to-Many relationer

### Scenario: Kurssystem

Du har:

- Studenter (id, namn, email)
- Kurser (kod, namn, poäng)

En student kan läsa flera kurser. En kurs kan ha flera studenter.

#### **Uppgift:**

1. Försök först rita en datamodell UTAN en kopplingstabell
2. Förklara varför detta inte fungerar
3. Rita en ny modell MED en kopplingstabell
4. Vilken extra information skulle kunna vara relevant att spara i kopplingstabellen?  
(Tips: tänk på när en student påbörjade kursen, betyg, etc.)

### Scenario: Taggar

Du har ett system där:

- Artiklar kan ha flera taggar
- Samma tagg kan användas på flera artiklar
- Man vill kunna söka artiklar baserat på taggar
- Man vill kunna se vilka taggar som är populärast

#### **Uppgift:**

1. Rita en datamodell som möjliggör detta

2. Förklara vilken data som behöver sparas i kopplingstabellen
3. Förklara varför man inte kan lösa detta genom att ha en kolumn "taggar" i artikeltabellen med kommaseparerade värden

## Övning 3: Skilja på Data och Funktionalitet

### Scenario: Bokningssystem

Krav:

- Användare ska kunna boka tider
- Man ska kunna se lediga tider
- Man ska kunna avboka senast 24h innan
- Man ska få en påminnelse dagen innan
- Dubbelbokning ska inte vara möjligt
- Man ska kunna se sin bokningshistorik

#### Uppgift:

1. Identifiera vilka av dessa krav som handlar om:
  - Data som måste lagras
  - Funktionalitet som ska implementeras
2. Rita en datamodell som BARA innehåller nödvändiga tabeller och fält
3. Förklara hur din datamodell möjliggör varje funktionellt krav, utan att implementera själva funktionaliteten

### Scenario: Betygssystem

Krav:

- Lärare ska kunna registrera betyg
- Betyg ska godkännas av examinator
- Student ska kunna se sina betyg
- Betyg kan inte ändras efter att de godkänts
- Systemet ska räkna ut snittbetyg
- Man ska kunna se betygsutveckling över tid

#### Uppgift:

1. Separera kraven i:
  - Data som behöver lagras
  - Beräkningar/funktionalitet
2. Rita en datamodell som bara fokuserar på data
3. För varje funktionellt krav, förklara vilken data i din modell som behövs för att implementera kravet

## Tips för alla övningar:

- Börja med att identifiera vad som faktiskt är data vs funktionalitet
- Tänk på vilken data som behövs för rapporter och sökningar
- Fundera över hur ofta olika typer av data uppdateras
- Överväg prestanda vid olika modellalternativ
- Tänk på att normalisering handlar om att undvika redundans och uppdateringsanomalier