

Datamodellering med ER

Læringsmål

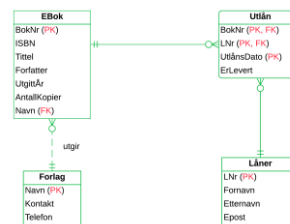
- ❑ Forstå formålet med datamodellering og hvordan det brukes i databasedesign.
- ❑ Kunne bruke ER (Entity Relationship) og UML (Unified Modeling Language) for å lage datamodeller.
- ❑ Kunne bruke entiteter med attributter og identifikatorer.
- ❑ Kunne bruke en-til-en, en-til-mange og mange-til-mange forhold.
- ❑ Kunne bruke svake entiteter og identifiserende forhold.
- ❑ Kunne bruke flerverdiattributter, sammensatte attributter, subtyper, aggregering og komposisjon i UML.
- ❑ Kjenne til noen vanlige modelleringsmønstre.

Datamodellering – hva og hvorfor?

- ❑ Det er ikke opplagt hvilke tabeller en database bør inneholde og hvilke kolonner disse tabellene bør ha.
- ❑ Vi må prøve oss litt fram for å finne en hensiktsmessig struktur.
- ❑ Det har vist seg nyttig å bruke «visuelle modeller» i denne prosessen.
- ❑ **ER** (Entity Relationship) er et slikt visuelt modellspråk.
- ❑ Et **ER-diagram** viser strukturen til en database.
- ❑ Fungerer som en «arkitekttegning» for en ny database.

Introduksjon til ER-diagrammer

- ❑ ER-diagrammer er en visuell beskrivelse av strukturen til en database.
 - Hver **entitet** (boksene) svarer til en tabell.
 - **Attributter** (i nederste del av boksene) svarer til kolonner.
 - **Identifikatorer** er understreket og svarer til primærnøkler.
 - **Forhold** (linjene) svarer til fremmednøkler.



- ER-diagrammer brukes når vi **planlegger** hva databasen skal inneholde.
- Gir god oversikt.
- **Lesbare** også for de som ikke er IT-eksperter.

Kravanalyse og faktainnsamling

- ❑ **Kravanalyse** (requirements analysis) systematiserer hva et slikt system skal gjøre.
- ❑ Starter med **faktainnsamling**: studere rapporter, skjemaer, gjøre intervjuer og spørreundersøkelser.
- ❑ Eksempel på krav til **biblioteksystemet** (se forrige side):
 - Systemet skal lagre opplysninger om lånerne, bøker, utlån og forlag.
 - Om hver e-bok skal man lagre ISBN, tittel, forfatter og utgivelsesår.
 - En e-bok blir publisert av ett bestemt forlag, men et forlag kan publisere mange e-bøker (mange titler).

Fra skisse til modell til database

- ❑ Vi bygger ut datamodeller med flere detaljer etter hvert som vi beveger oss fra **krav** til tidlige **skisser** til stadig mer detaljerte **datamodeller** til ferdig **database**.
- ❑ Eksempel på en tidlig skisse for biblioteksystemet:



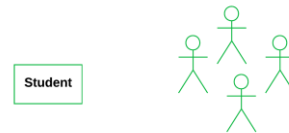
Entiteter og attributter

- En **entitet** er et objekt eller en «ting» vi ønsker å lagre informasjon om.
- En entitet har et sett med interessante egenskaper som kalles for **attributter**.
- Når vi «beveger» oss mot ferdig database vil attributtene bli tildelt **datatyper**. I første omgang sløyfer vi dette.

Student	Innlegg
StudentNr	Nr INTEGER
Fornavn	Brukernavn VARCHAR(50)
Etternavn	Dato DATE
Adresse	Melding VARCHAR(140)
Telefon	

Type og forekomst

- Tabellen Student inneholder mange rader.
- Hver rad beskriver én person.
- En **entitetsforekomst** svarer til en rad i en tabell.
 - Eksempel: (StudentNr=1, Fornavn='Hans', Etternavn='Hansen', Adresse='Hansegata 3')
- Entitetstypen** representerer mengden av forekomster – og svarer dermed til hele tabellen.



Forhold



- Kunde og Ordre er **entiteter**.
- Det er et **forhold** mellom Kunde og Ordre: En ordre tilhører en kunde.

Kardinalitet

1:N
(én-til-mange)



1:1
(én-til-én)

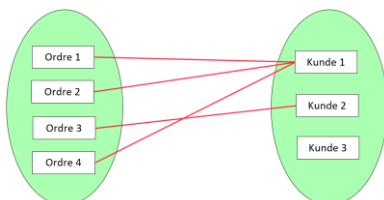


M:N
(mange-til-mange)



Type og forekomst

- Vi skiller også mellom type og forekomst av **forhold**.
 - Forekomst: Eierforholdet mellom kunde 2 og ordre 3.
 - Type: Samlingen av alle eierforhold (alle linjene).



Minimums- og maksimumskardinalitet

- En avdeling kan ha minimum 0 og maksimum mange ansatte, mens en ansatt jobber i nøyaktig én avdeling (minimum 1 og maksimum 1).
 - Symbolet nærmest entitetene angir **maksimumskardinalitet**.
 - Kråkefot** betyr «mange», mens | betyr 1.
 - Symbolet innenfor angir **minimumskardinalitet**.
 - Sirkel betyr 0 og | betyr 1.

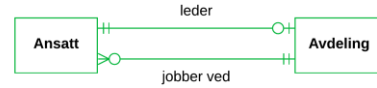


Minimums- og maksimumskardinalitet

- ❑ **Forenklet notasjon** (brukes i noen dialekter):
 - En sirkel alene betyr maksimum=minimum=0.
 - En | alene betyr maksimum=minimum=1.
- ❑ Generelt er det mulig å bruke andre «tall» enn 0, 1 og «mange».
 - Eksempel: 3..7
 - Forskjellen på 0, 1 og «mange» avgjør tabellstruktur.
- ❑ Vi bruker både minimum og maksimum:
 - Sirkel og kråkefot betyr «0 eller mange».
 - Vertikal strek og kråkefot betyr «1 eller mange».
 - Sirkel og vertikal strek betyr «0 eller 1».
 - To vertikale streker betyr «nøyaktig 1».

Roller og navn på forhold

- ❑ **Navn** på forhold gjør modellen mer lesbar.
 - Avgjørende ved flere forhold mellom de samme to entitetene.



- ❑ Kan også bruke **roller**.
 - En lærer kan spille rollen som sensor.



Ikke-identifiserende forhold



- ❑ Vi skal nå legge på litt flere detaljer. Entitetene blir utvidet med attributter, og vi starter jobben med å sette på primærnøkler (angitt med PK).
- ❑ Vi kommer dessuten til å skille mellom heltrukne og stipledede linjer når vi tegner forhold. Forholdet mellom Kunde og Ordre er et eksempel på et ikke-identifiserende forhold.

Svake entiteter / identifiserende forhold



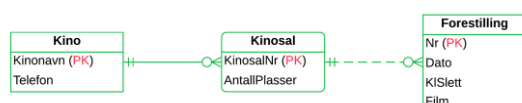
- ❑ En **svak entitet** arver deler av sin identifikator fra en annen og kan ikke eksistere uten denne.
 - En Kinosal kan ikke eksistere uten en tilhørende Kino.
 - Identifikatoren for Kinosal er (delvis) avledet fra identifikatoren for Kino.
 - MySQL Workbench: Velg **identifying relationship**.

Løpenumre eller svake entiteter

- ❑ Utstrakt bruk av svake entiteter kan føre til primærnøkler sammensatt av mange kolonner.



- ❑ Kan alltid erstattes av **surrogatnøkler** (løpenumre).



Oppløsning av mange-til-mange forhold

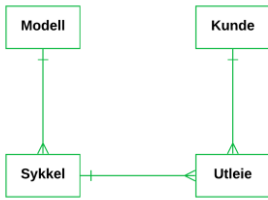


Er det interessante egenskaper ved forholdet?



Ulike abstraksjonsnivåer

- ❑ Av og til er det ikke ønskelig å vise alle detaljene i en datamodell.
- ❑ Oversiktsbilde uten attributter.
- ❑ Tar ikke stilling til svake entiteter / identifiserende forhold.



Mer detaljerte datamodeller

- ❑ Hvis vi tar med primærnøkler (PK), fremmednøkler (FK) og datatyper i diagrammet har vi en presis beskrivelse av databasen. Dette kaller vi for en **logisk datamodell**.
- ❑ Kap. 8 forklarer veien fra begrepsmessig til logisk modell.

