INFO221v12
IR II

Truls Pedersen
Institutt for informasjons- og medievitenskap
Universitetet i Bergen

Oversikt

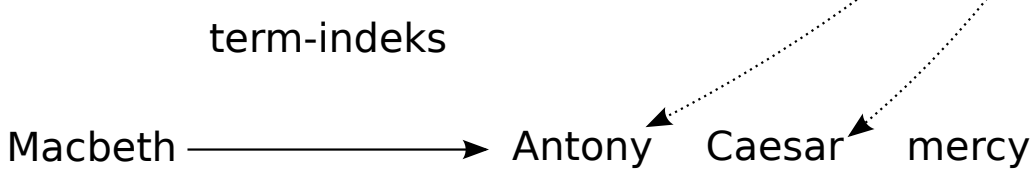
- Terminologi
- Datamodellering
- Modelleringssystemer
  - RM
  - ER og EER
  - Generelt
  - SSM
- Multimedia
- Dublin Core
- MPEG-7
- Annotering (merking)
- Søking med jokertegn
- Likhetsmål (ord)

Terminologi

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
Antony	1	1	0	0	0	1	
Brutus	1	1	0	1	0	0	
Caesar	1	1	0	1	1	1	
Calpurnia	0	1	0	0	0	0	
Cleopatra	1	0	0	0	0	0	
mercy	1	0	1	1	1	1	
worser	1	0	1	1	1	0	
...							

Terminologi

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
Antony	1	1	0	0	0	1	
Brutus	1	1	0	1	0	0	
Caesar	1	1	0	1	1	1	
Calpurnia	0	1	0	0	0	0	
Cleopatra	1	0	0	0	0	0	
mercy	1	0	1	1	1	1	
worser	1	0	1	1	1	0	
...							



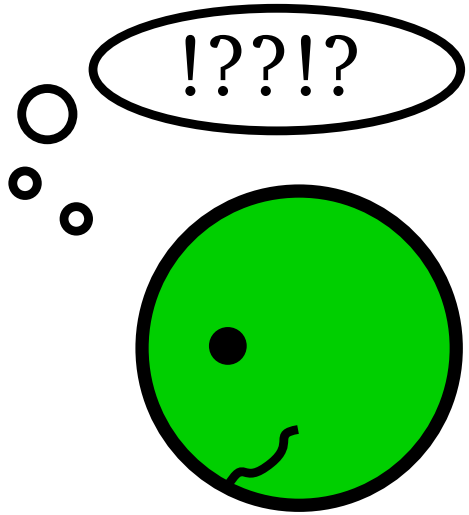
Terminologi

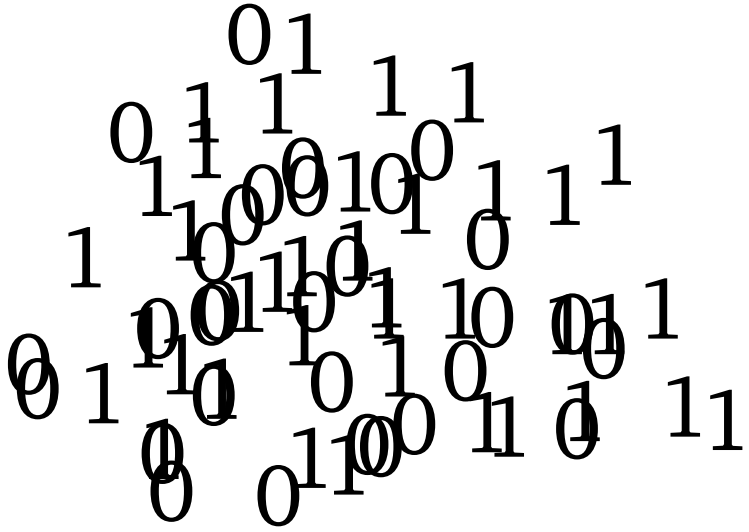
	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
Antony	1	1	0	0	0	1	
Brutus	1	1	0	1	0	0	
Caesar	1	1	0	1	1	1	
Calpurnia	0	1	0	0	0	0	
Cleopatra	1	0	0	0	0	0	
mercy	1	0	1	1	1	1	
worser	1	0	1	1	1	0	
...							

invertert term-indeks

Brutus → Antony and Cleopatra, Julius Caesar, Hamlet

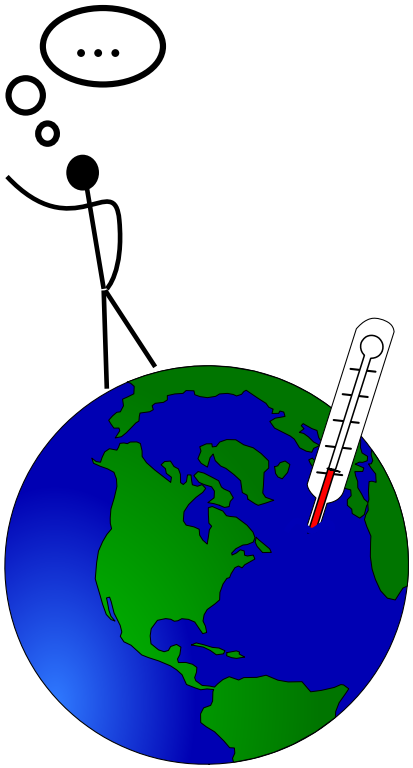
Datamodellering





Datamodellering

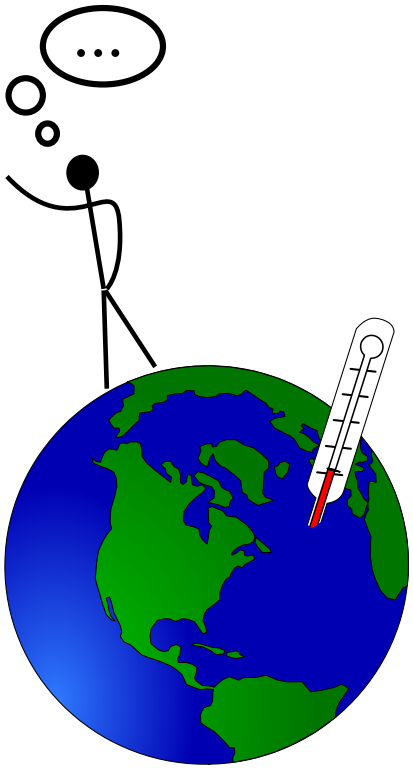
Fakta



Earth: Brian Pearl@clker.com
Therm.: OCAL@clker.com

Datamodellering

Fakta



Earth: Brian Pearl@clker.com
Therm.: OCAL@clker.com

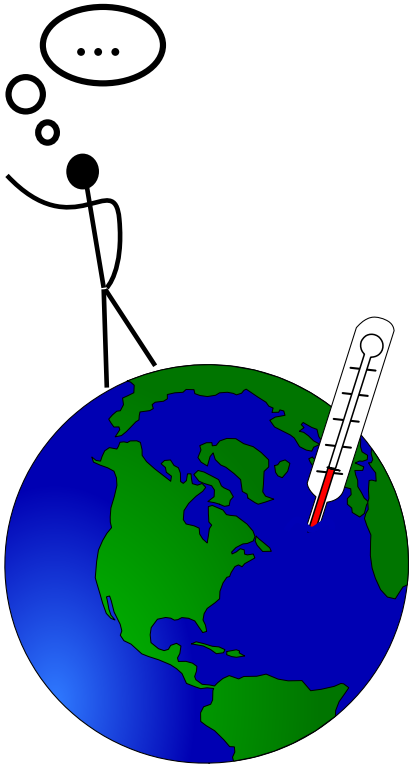
Data

Jeg synes at ...

271.15K

Datamodellering

Fakta



Earth: Brian Pearl@clker.com
Therm.: OCAL@clker.com

Data

Jeg synes at ...

271.15K

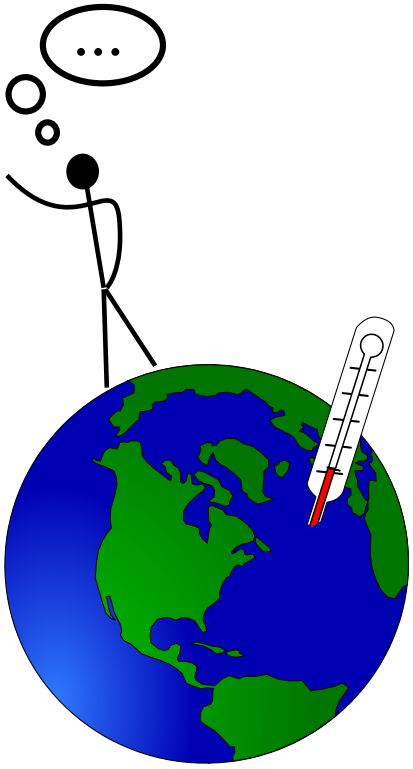
Strukturert data

```
<103, "Jeg synes at ...",
      auser,
      ["First!",...]>
```

```
<103103, 30.0, 30.0
         271.15,
         1298888664,
         ascientist>
```

Datamodellering

Fakta



Earth: Brian Pearl@clker.com
Therm.: OCAL@clker.com

Data

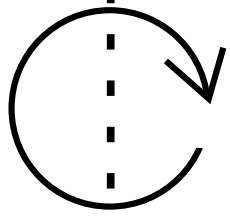
Jeg synes at ...

271.15K

Strukturert data

```
<103, "Jeg synes at ...",
      auser,
      ["First!",...]>
```

```
<103103, 30.0, 30.0
         271.15,
         1298888664,
         ascientist>
```



Datamodellering

Datamodellering

1. Identifiser og beskriv *informasjonsbehovet*
   Hvilken informasjon er viktig/ønskelig for systemet vi konstruerer?

Datamodellering

1. Identifiser og beskriv *informasjonsbehovet*
   Hvilken informasjon er viktig/ønskelig for systemet vi konstruerer?

2. Spesifiser *datagrunnlaget*
   Hvilke data må informasjonssystemet holde styr på?

Datamodellering

1. Identifiser og beskriv *informasjonsbehovet*
   Hvilken informasjon er viktig/ønskelig for systemet vi konstruerer?

2. Spesifiser *datagrunnlaget*
   Hvilke data må informasjonssystemet holde styr på?

3. Spesifiser *datastrukturen*
   Hvordan representerer vi disse dataene?

Datamodellering

1. Identifiser og beskriv *informasjonsbehovet*
   Hvilken informasjon er viktig/ønskelig for systemet vi konstruerer?

2. Spesifiser *datagrunnlaget*
   Hvilke data må informasjonssystemet holde styr på?

3. Spesifiser *datastrukturen*
   Hvordan representerer vi disse dataene?

Denne prosessen skal helst produsere en *datamodell* som vi kan bruke i utviklingen av informasjonssystemet. (*metadata*)

Designprosessen

1. Utvikle en (semantisk) modell sammen med (representanter fra) brukergruppen.

2. Oversette denne modellen til en strukturell beskrivelse. Beskrive eventuelle krav (funksjonalitet/begrensninger) utover det rent strukturelle.

3. Implementere modellen i DBMSet vi har valgt.

Modelleringssystemer

Siden modellen både skal fungere som kommunikasjons-medium og spesifikasjon (og de forskjellige mottakerne har ulik bakgrunn) finnes det mange typer systemer.

Vi skiller bl.a. mellom *grafiske* og *deklerative* datamodeller.

- Grafisk: ER, EER, ...
- Deklerative systemer (en liste av påstander):
  RM, AI/prolog, Dublin Core, DDL
- Tabular: skjema + innhold

Modelleringssystemer - RM

EMPLOYEE	<Id, Name, Address, PictureID, CV, ...>
CUSTOMER	<Id, Name, Address, Rating, ...>
PRODUCT	<Id, Name, Pictures, Video, Description, ...>
Contract	<EId, CId, PId, date, quantity, cost, ... >

Nordbotten

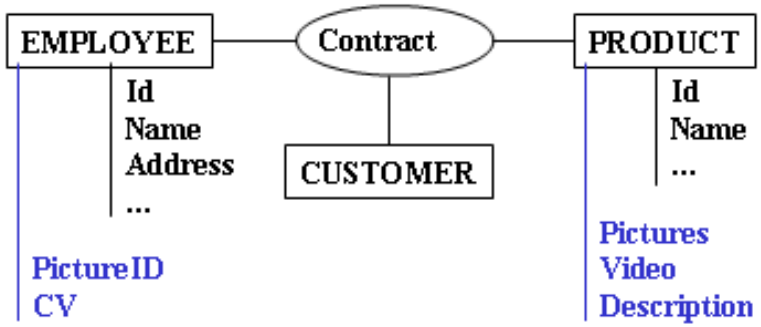
Kanskje den enkleste datamodelltypen.

Hver tabell representerer et predikat.

Større strukturer blir vanskelig å visualisere og kommunisere.

Veldig nær kobling til faktisk implementasjon (SQL, ol.).

Modelleringssystemer - ER



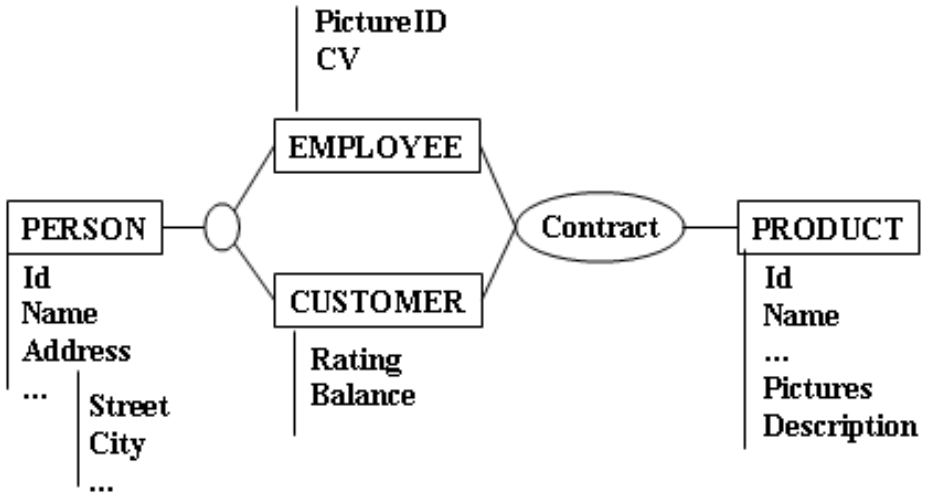
Nordbotten

Visulaliserer datastrukturen på en mer intuitiv måte.

Strukturen lar seg ikke representere eksplisitt.

Mange ønskelige semantiske begrensninger lar seg ikke uttrykke.

Modelleringssystemer - EER



Nordbotten

Utvidelse av ER.

Lar oss uttrykke hierarkiske egenskaper ved relasjoner mellom entiteter.

Modelleringssystemer - generelt

Det finnes veldig mange forslag til datamodelleringssystemer.

Disse klarer i ulik grad å være anvendelige i alle steg i modelleringsprosessen.

Systemet skal kunne beskrive:

1. *Struktur* - typer entiteter og relasjoner
2. *Begrensninger* - tillatte verdier, forhold, antall, ...
3. *Operasjoner* - integritet, beregning av verdier, ...

Det er blitt gjort mye forskning på hvordan vi kan gjøre rimelige begrensninger på en enkel og oversiktlig måte.

Modelleringssystemer - SSM

SSM - *structural semantic model* - ble først foreslått av Nordbotten.

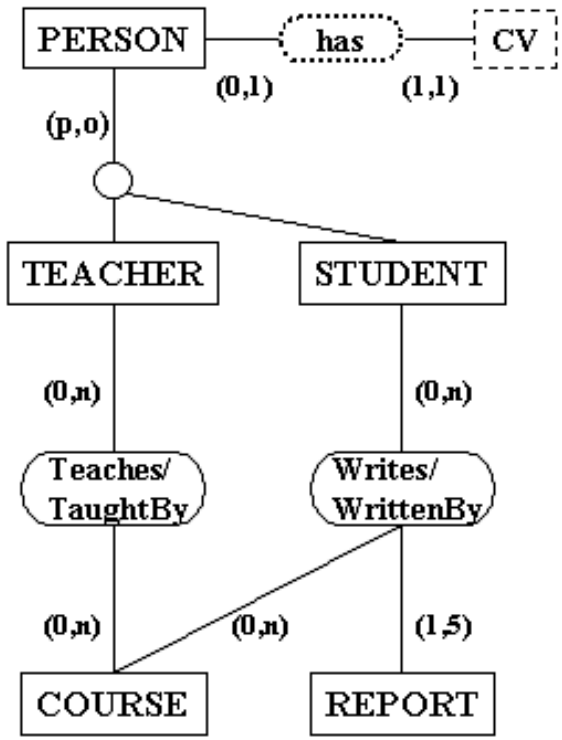
Utvider EERs uttrykkskraft, men forenkler den grafiske syntaksen.

Systemet har syntaks for å beskrive:

1. Tre type *entiteter*: grunnleggende, underklasse og svake
2. Fire typer *relasjoner mellom entiteter*: *n*-ære relasjoner, samt tre type hierarkiske forhold
3. Fire *attributtyper*: atomær, fler-verdi, sammensatt og utledet
4. *Domenespesifikasjon* for attributtene
5. *Kardinalitetsbegrensninger*
6. *Verdibegrensninger*

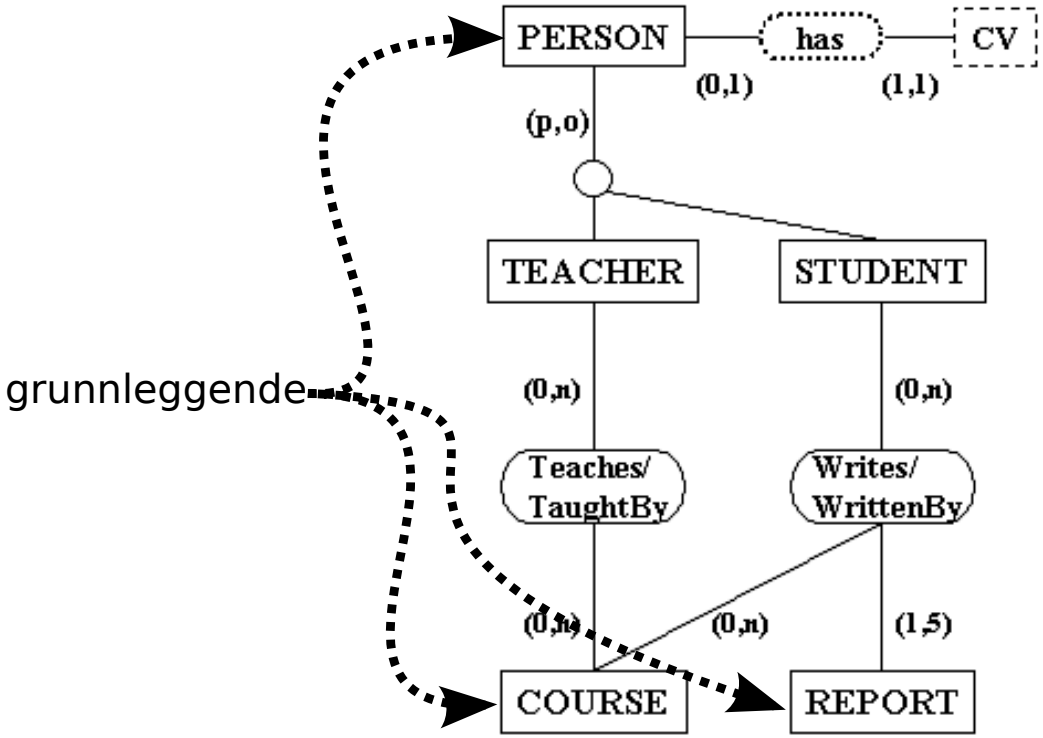
SSM - entitetstyper og relasjoner

- Entiteter: grunnleggende(og underklasser) og svake
- Relasjoner: tilhørighet, hierarkisk (med begrensninger)



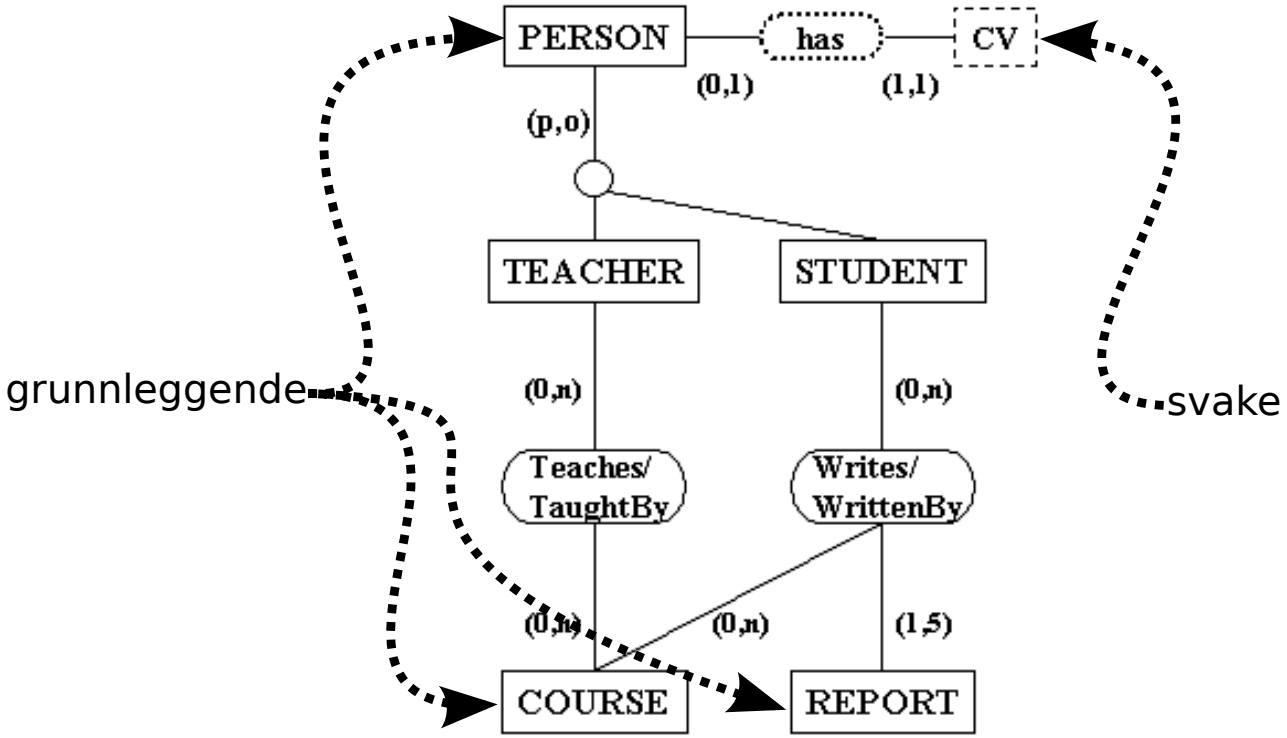
SSM - entitetstyper og relasjoner

- Entiteter: grunnleggende(og underklasser) og svake
- Relasjoner: tilhørighet, hierarkisk (med begrensninger)



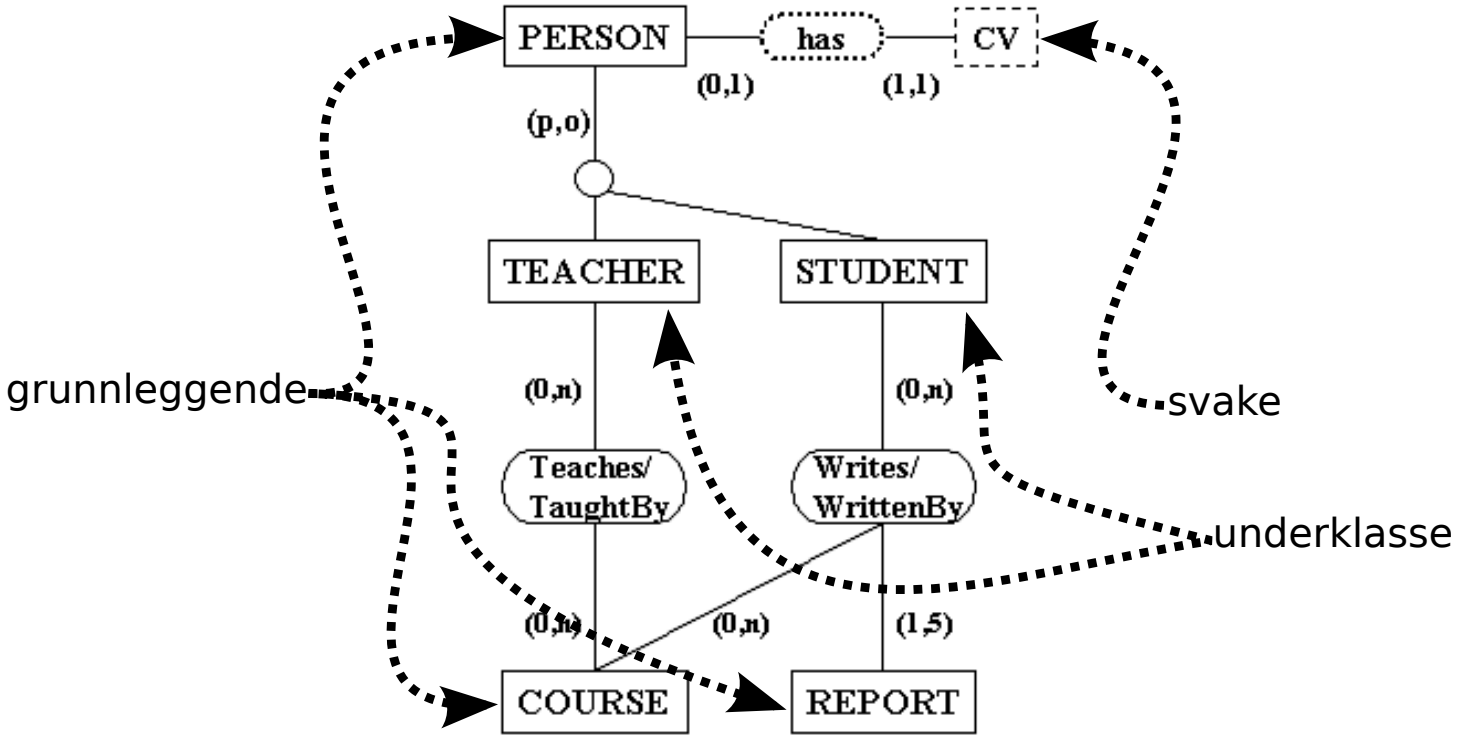
SSM - entitetstyper og relasjoner

- Entiteter: grunnleggende(og underklasser) og svake
- Relasjoner: tilhørighet, hierarkisk (med begrensninger)



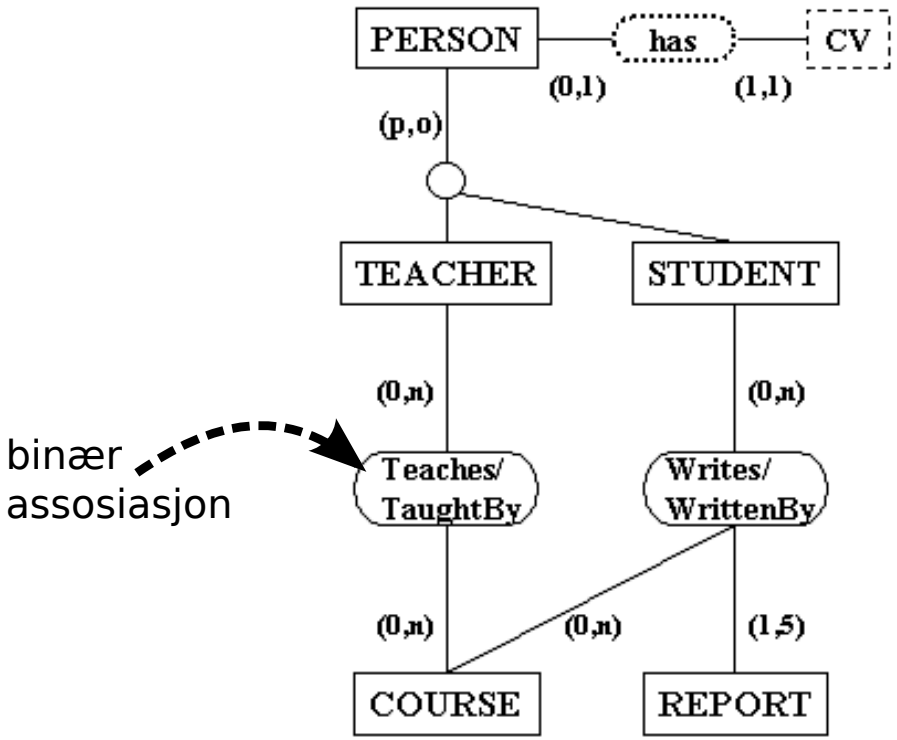
SSM - entitetstyper og relasjoner

- Entiteter: grunnleggende(og underklasser) og svake
- Relasjoner: tilhørighet, hierarkisk (med begrensninger)



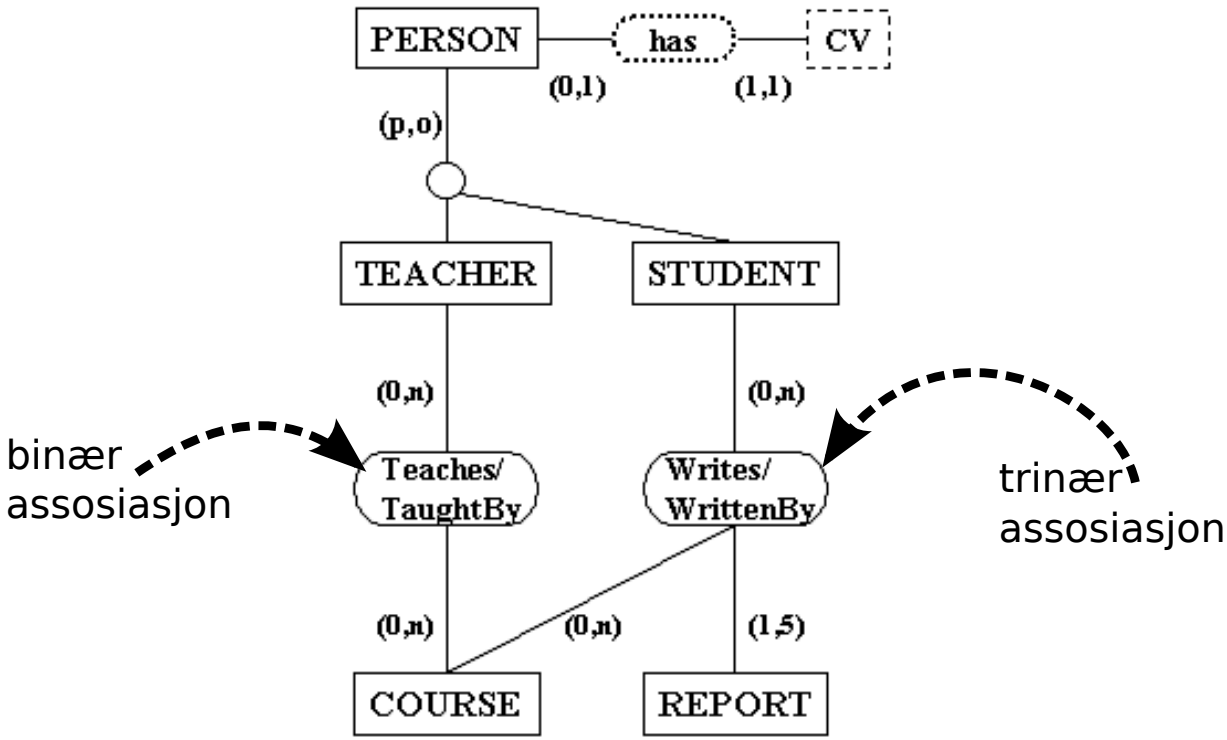
SSM - entitetstyper og relasjoner

- Entiteter: grunnleggende(og underklasser) og svake
- Relasjoner: tilhørighet, hierarkisk (med begrensninger)



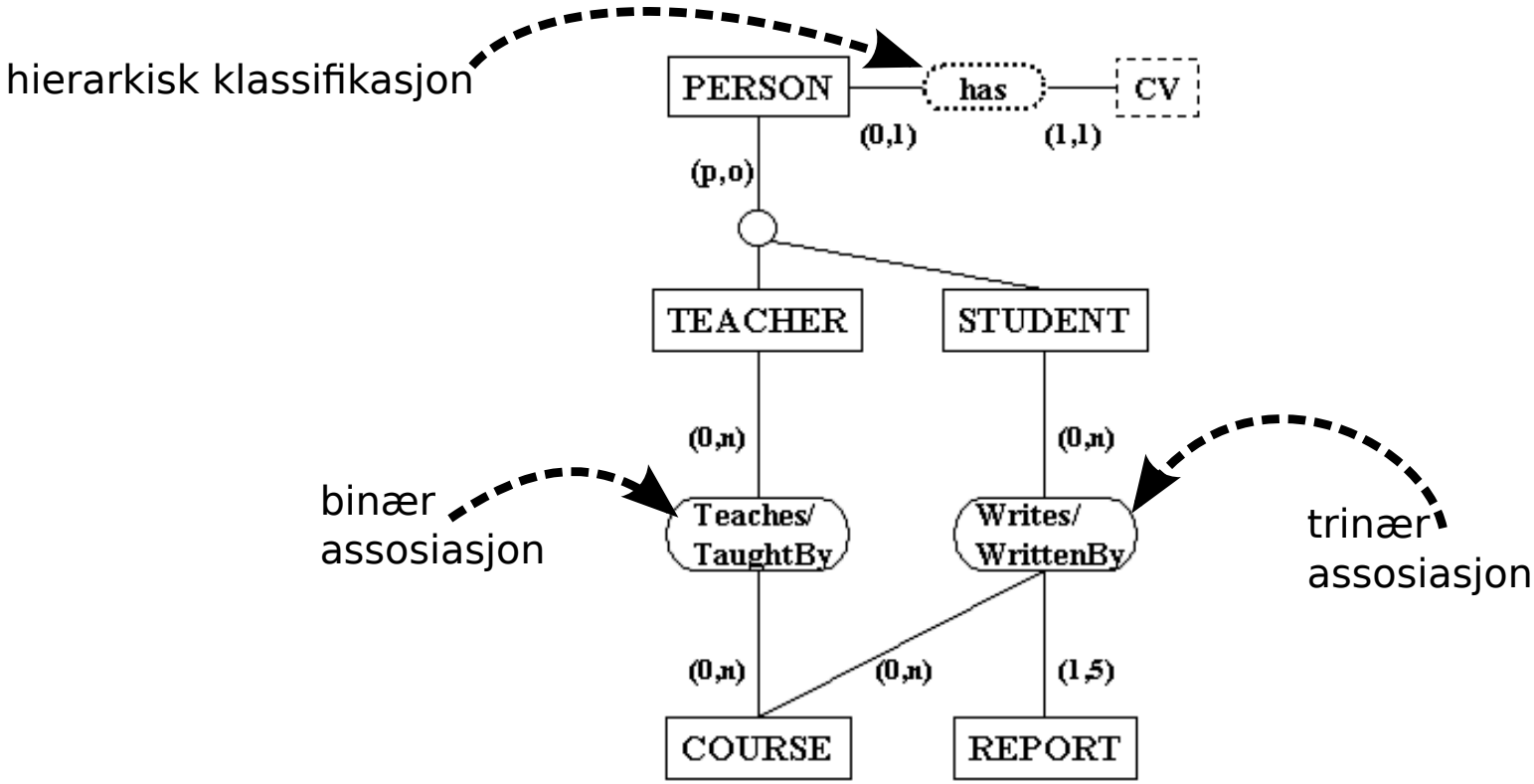
SSM - entitetstyper og relasjoner

- Entiteter: grunnleggende(og underklasser) og svake
- Relasjoner: tilhørighet, hierarkisk (med begrensninger)

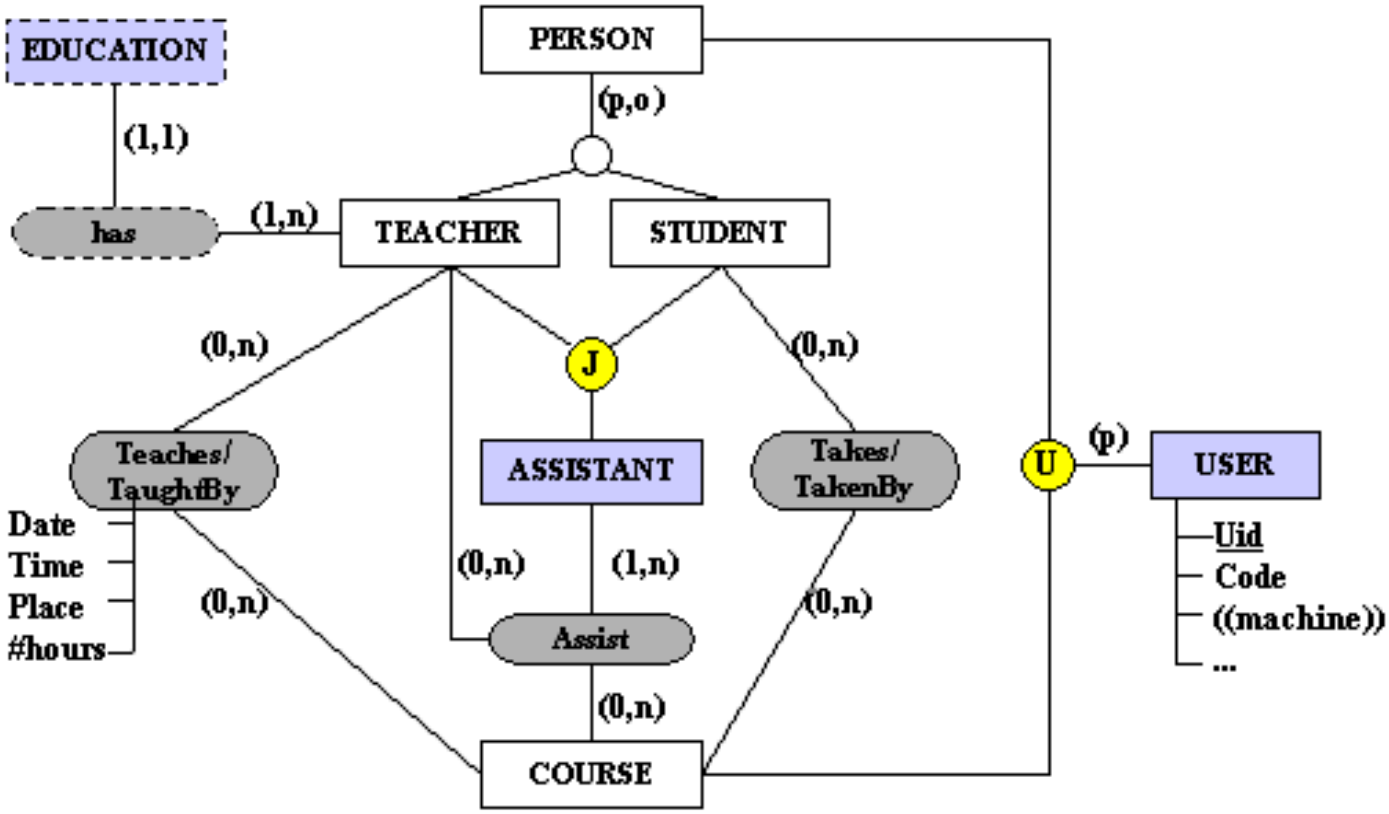


SSM - entitetstyper og relasjoner

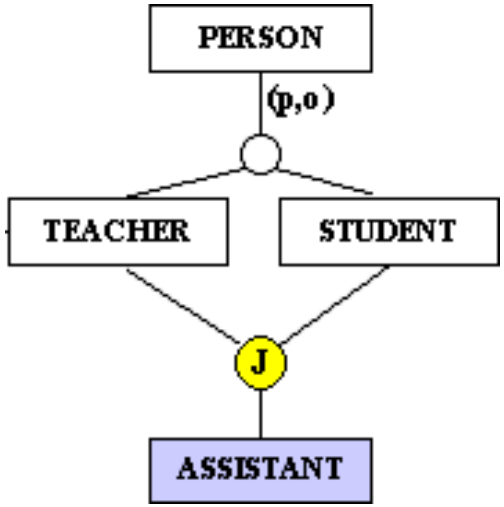
- Entiteter: grunnleggende(og underklasser) og svake
- Relasjoner: tilhørighet, hierarkisk (med begrensninger)



SSM - entitetstyper og relasjoner



SSM - entitetstyper og relasjoner

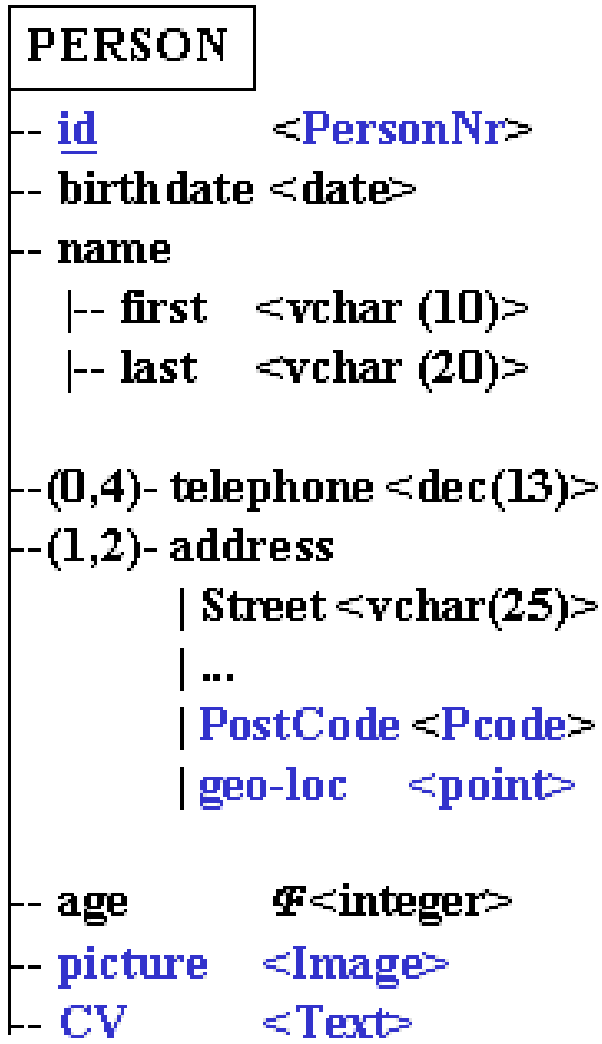


Delt underklasse (*shared subclass*), *Assistant*, utvider en del av student og foreleser med attributter spesifikk for assistenter.

Kategori (*category subclass*) representerer en samling av entitetstyper. *Person* er en partiell, overlappende kategori.

(partiell (deltagelse): person *må* ikke være foreleser/student; overlappende (medlemskap): en foreleser kan være en student.)

SSM - attributter



SSM - attributter

Typer Attributtens struktur:
Atomær (*id*), sammensatt (*navn*), flerverdi (*tlf.*), utledet (*kl.*), media (*bilde*)

Domenespesifikasjon Veldefinert domene for hver attributt:
integer, character, BLOB, ...

Verdibegrensninger Rimelige begrensninger:
antall siffer i et telefonnummer, ...

OBS! I SSM kan også *relasjoner* ha attributter!

Hvis en relasjon har mange attributter kan det være en idé å heller modellere denne som en entitet.

SSM - relasjoner

Vi kan ha assosiative relasjoner av vilkårlig aritet:

- `eksamen(student, fag)`,
- `oblig(student, foreleser, assistent)`,
- ⋮

Vi har sett annotasjonen

- *p*/*t* (partiell/total),
- *o*/*d* (overlappende/disjunkt).

Vi har også kardinalitetsbegrensninger, skrevet (min, max).

Multimediamodeller

Aspekter (metadata) vi bør ta med når vi modellerer medier:

Semantisk beskriver mening/innhold i dokumentet
betydning av en rapport, stemningen i et bilde...
(vanskelig/umulig å ekstrahere automatisk)

Kontekstuell beskriver dokumentets forhold til sine omgivelser
forfatter, tema...
(enklere, men ikke 100% sikker)

Strukturell beskriver dokumentets (interne) struktur
oppløsning/størrelse, format...
(lett å oppdrive automatisk)

Multimedia - Semantisk metadata

Semantisk metadata kan være svært vanskelig å oppdrive.

Bilder funnet på internett er ofte funnet innlemmet i en webside.

En *web crawler* vil forsøke å trekke ut informasjon om bildet fra teksten nær bildet.

Evt. kan indekstermer bli manuelt spesifisert av en ekspert i et standardisert vokabular.

Multimedia - Kontekstuell/strukturell metadata

Kontekstuell

- Forfatter/tegner, publiseringsdato, dokumentets plassering, ...
- Ofte nyttig i forhold til søking.
- Ofte enkelt å oppdrive automatisk.

Strukturell

- Dokument lengde/størrelse, språk, formatering, farge, ...
- Mindre nyttig i søking, men mye brukt i forbindelse med *presentasjonen*.
- Generelt enkel å finne.

Dublin Core

Standard for metadata for *dokumenter*:

Type	DC element
Semantisk	Tittel, Emne, Beskrivelse, Type[1], Dekning
Kontekstuell	Forfatter, Bidrager, Utgiver, Dato, Rettigheter, Kilde, Forhold
Strukturell	Type[2], Format, Språk, Id

Type[1] beskriver kategori/funksjon/sjanger: rapport, dikt, ...
Type[2] beskriver mediatype: bilde, video, tekst, ...

DC er beregnet for å beskrive *statiske* dokumenter manuelt ved bruk av *domene ontologier* (*kontrollerte vokabularer*).

Dublin Core

Dublin Core hevdes å være i stand til å tilstrekkelig beskrive alle typer multimedia objekter (dokumenter).

Men...

- Søkeren er kanskje ikke kjent med den formelle domene ontologien,
- DC beskriver *hele* dokumentet, ikke deler (scener, segmenter, ...)
- Er ment å genereres manuelt, svært tidkrevende. (Ofte praktisk umulig.)

MPEG-7

“Multimedia Content Description Interface”

MPEG designet denne standarden for å beskrive video filmer, men kan anvendes på forskjellige typer dynamiske medier.

Rammeverket lar oss beskrive deler av mediet, uten å stille formelle krav til beskrivelsene.

Det *anbefales* å begrense beskrivelsene til lav-nivås beskrivelser: farge, form, kamera, ...

Annotering - Tekstdokumenter

Manuell annotering er pålitelig, men

1. Svært tidkrevende, samt kjedelig $\Rightarrow$ kan føre til utelatelser og feil
2. Subjektive vurderinger kan gjøre at informasjon som *søker* mener er viktig blir utelatt
3. Ord som brukes som søkeord er kanskje ikke de samme som dokumentet ble annotert med (f.eks. synonymer)

Automatisk annotering er rask, men

1. Begrenset hva vi kan trekke ut
2. Mindre sikker tolkning/vurdering
3. Utfordringer med slike metoder: Heaps' og Zipfs lov (senere i kurset)

Annotering - Bildedokumenter

Bilder er *ustrukturerte*, de består ikke av enkle “byggeklosser” (ord).

DC kan fange mye metadata om bilder.

Spesielt for bilder beskriver boken flere nivå av beskrivelser:

Nivå	Karakteristikk	Eksempel
1	strukturell	farge, tekstur
2	objekter (typer)	trær, bygninger
3	objekter (spesifikke)	Det Hvite Hus, Jens Stoltenberg
4	hendelser	toppmøte, sermoni
5	stemning	formelt, diplomatisk

Søkestrukturer

Hashtabeller kan være svært raske, men har ulemper som gjør dem ugunstig for oss. Mye vedlikehold (ofte vanskelig å oppdatere). Kan bli svært plasskrevende hvis vi har store datamengder (kan minskres ved å ofre litt hastighet).

Søketrær er lett å vedlikeholde og krever ikke mye mer plass enn vi putter inn i dem. De er også forholdsvis raske. De enkleste (binærtrærene) er svært velegnet til å søke i mengder med ord.

Ofte brukte variasjoner er *balanserte binære søketrær* og *B-trær*.

Jokertegn (*wildcards*)

Hvis vi ønsker å søke etter ord som begynner på `nom` i et søketre, begynner vi på begynnelsen av ordet og leter oss nedover i treet. Når vi har funnet noen som samsvarer med siste tegn vil “alle” barnene til denne noen være ord som begynner med `nom`. Et slikt søk spesifiseres ofte som ``nom*''.

Jokertegn (*wildcards*)

Hvis vi ønsker å søke etter ord som begynner på `nom` i et søketre, begynner vi på begynnelsen av ordet og leter oss nedover i treet. Når vi har funnet noen som samsvarer med siste tegn vil "alle" barnene til denne noen være ord som begynner med `nom`. Et slikt søk spesifiseres ofte som ``nom*''.

Hvis dette søket ble gjort i et søketre der elementene var indeksert *baklengs*, har vi funnet alle ordene som slutter med `mon`!

Hvis vi tar snittet av resultatene fra et søk etter `l*` i det orginale søketreet og fra et søk etter `nom*` fra et reversert søketre, får vi alle ordene som begynner med `l` og slutter med `mon`.

Disse ordene kan vi så bruke til å slå opp i vår indeksering av termer som vi har konstruert tidligere.

k-Gram

Et k-Gram er et ord som består av *k* tegn.

Vi kan dele et ord opp i, f.eks., 3-gram:

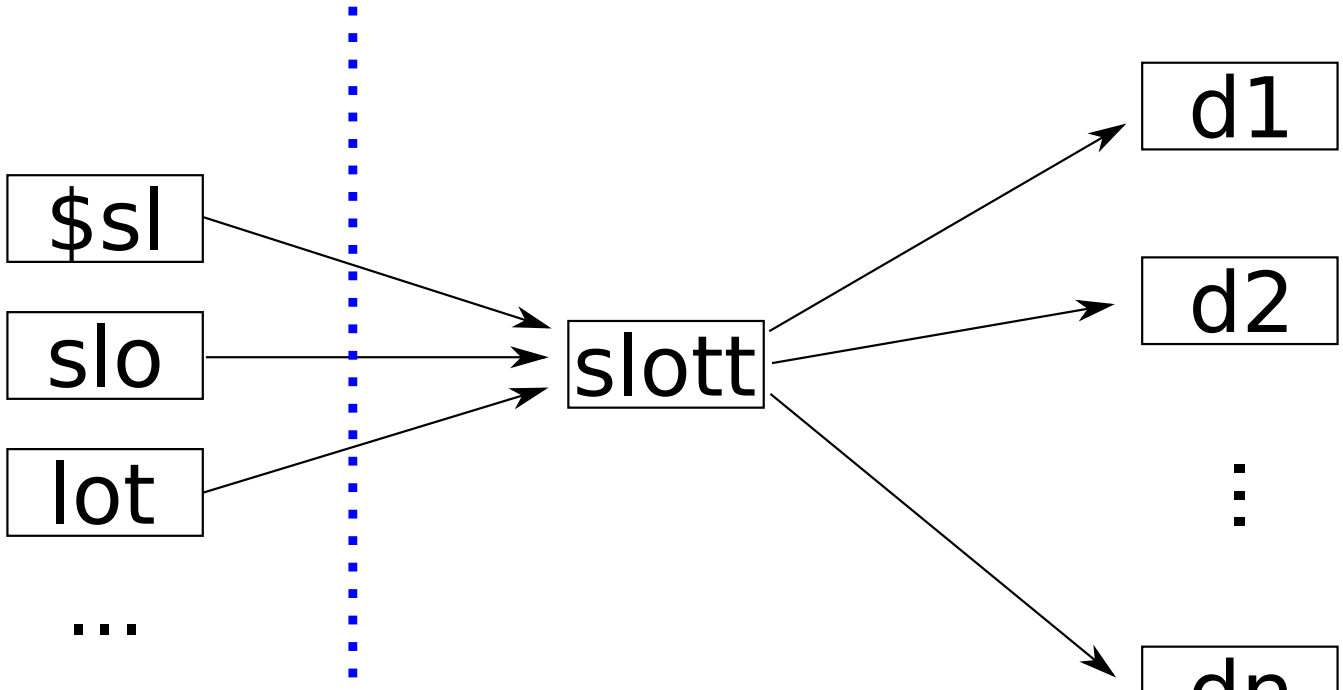
sl, slo, lot, ott, tt

Hvor $ er et spesialtegn vi bruker for å beskrive begynnelsen/slutten av et ord.

Hvis vi lager en indeks for indeksene våre bestående av k-gram, kan vi lettere søke etter tekst med jokertegn.

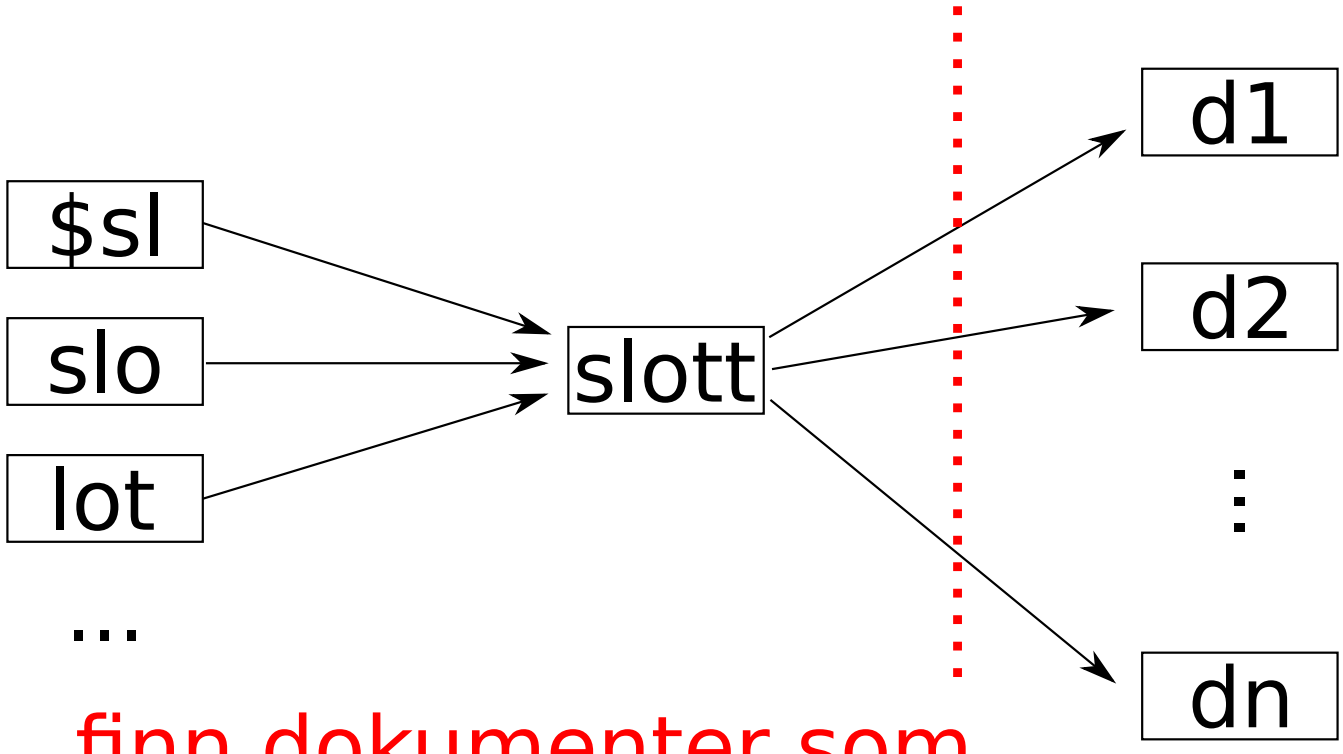
Hvis “slott” er en indeks i listen vår, vil alle 3-grammene over referere til til dette ordet.

k-Gram



finn termer som
passer med søket

k-Gram



finn dokumenter som inneholder termen (invertert termindeks)

k-Gram

Hvis vi nå ønsker å søke etter et ord med jokertegn i, deler vi denne inn i 3-grammene som passer med søketeksten.

F. eks: `red*` $\Rightarrow$ $re og red.

k-Gram

Hvis vi nå ønsker å søke etter et ord med jokertegn i, deler vi denne inn i 3-grammene som passer med søketeksten.

F. eks: `red*` $\Rightarrow$ $re og red.

Disse bruker vi for å finne alle indekser som inneholder alle disse 3-grammene.

Vil vil få: “retired”, “reduce”, ...

k-Gram

Hvis vi nå ønsker å søke etter et ord med jokertegn i, deler vi denne inn i 3-grammene som passer med søketeksten.

F. eks: `red*` $\Rightarrow$ $re og red.

Disse bruker vi for å finne alle indekser som inneholder alle disse 3-grammene.

Vil vil få: “retired”, “reduce”, ...

Denne resultatlisten må vi så etterbehandle for å luke ut de resultatene som ikke passer med søketeksten (“retired”).

Levenshtein avstand

Hvor stor forskjeller er det på “taket” og “takker”?

Levenshtein avstand

Hvor stor forskjeller er det på “taket” og “takker”?

		t	a	k	k	e	r
t							
a							
k							
e							
t							

Levenshtein avstand

Hvor stor forskjeller er det på “taket” og “takker”?

		t	a	k	k	e	r
	0	1	2	3	4	5	6
t	1						
a	2						
k	3						
e	4						
t	5						

Levenshtein avstand

$$m[i,j] = \min\{m[i-1,j-1] + s,$$
$$m[i-1,j] + 1,$$
$$m[i,j-1] + 1\}$$

$$s = \begin{cases} 0, \text{ hvis } s_1[i] = s_2[j] \\ 1, \text{ ellers } 0 \end{cases}$$

		t	a	k	k	e	r
	0	1	2	3	4	5	6
t	1	?					
a	2						
k	3						
e	4						
t	5						

Levenshtein avstand

$$m[i,j] = \min\{m[i-1,j-1] + s,$$
$$m[i-1,j] + 1,$$
$$m[i,j-1] + 1\}$$

$$s = \begin{cases} 0, \text{ hvis } s_1[i] = s_2[j] \\ 1, \text{ ellers } 0 \end{cases}$$

		t	a	k	k	e	r
	0	1	2	3	4	5	6
t	1	0	?				
a	2						
k	3						
e	4						
t	5						

Levenshtein avstand

$$m[i,j] = \min\{m[i-1,j-1] + s,$$
$$m[i-1,j] + 1,$$
$$m[i,j-1] + 1\}$$

$$s = \begin{cases} 0, \text{ hvis } s_1[i] = s_2[j] \\ 1, \text{ ellers } 0 \end{cases}$$

		t	a	k	k	e	r
	0	1	2	3	4	5	6
t	1	0	1	2	3	4	5
a	2	?					
k	3						
e	4						
t	5						

Levenshtein avstand

$$m[i,j] = \min\{m[i-1,j-1] + s,$$
$$m[i-1,j] + 1,$$
$$m[i,j-1] + 1\}$$

$$s = \begin{cases} 0, \text{ hvis } s_1[i] = s_2[j] \\ 1, \text{ ellers } 0 \end{cases}$$

		t	a	k	k	e	r
	0	1	2	3	4	5	6
t	1	0	1	2	3	4	5
a	2	1	?				
k	3						
e	4						
t	5						

Levenshtein avstand

$$m[i,j] = \min\{m[i-1,j-1] + s,$$
$$m[i-1,j] + 1,$$
$$m[i,j-1] + 1\}$$

$$s = \begin{cases} 0, \text{ hvis } s_1[i] = s_2[j] \\ 1, \text{ ellers } 0 \end{cases}$$

		t	a	k	k	e	r
	0	1	2	3	4	5	6
t	1	0	1	2	3	4	5
a	2	1	0	?			
k	3						
e	4						
t	5						

Levenshtein avstand

$$m[i,j] = \min\{m[i-1,j-1] + s,$$
$$m[i-1,j] + 1,$$
$$m[i,j-1] + 1\}$$

$$s = \begin{cases} 0, \text{ hvis } s_1[i] = s_2[j] \\ 1, \text{ ellers } 0 \end{cases}$$

		t	a	k	k	e	r
	0	1	2	3	4	5	6
t	1	0	1	2	3	4	5
a	2	1	0	1	2	3	4
k	3	2	1	0	1	2	3
e	4	3	2	1	?	?	
t	5						

Levenshtein avstand

$$m[i,j] = \min\{m[i-1,j-1] + s,$$
$$m[i-1,j] + 1,$$
$$m[i,j-1] + 1\}$$

$$s = \begin{cases} 0, \text{ hvis } s_1[i] = s_2[j] \\ 1, \text{ ellers } 0 \end{cases}$$

		t	a	k	k	e	r
	0	1	2	3	4	5	6
t	1	0	1	2	3	4	5
a	2	1	0	1	2	3	4
k	3	2	1	0	1	2	3
e	4	3	2	1	1	1	2
t	5	4	3	2	2	?	

Levenshtein avstand

$$m[i,j] = \min\{m[i-1,j-1] + s,$$
$$m[i-1,j] + 1,$$
$$m[i,j-1] + 1\}$$

$$s = \begin{cases} 0, \text{ hvis } s_1[i] = s_2[j] \\ 1, \text{ ellers } 0 \end{cases}$$

		t	a	k	k	e	r
	0	1	2	3	4	5	6
t	1	0	1	2	3	4	5
a	2	1	0	1	2	3	4
k	3	2	1	0	1	2	3
e	4	3	2	1	1	1	2
t	5	4	3	2	2	2	2