



Databasesystemer

fra forskellige synsvinkler

Kim Skak Larsen

`kslarsen@imada.sdu.dk`

IMADA



Oversigt

- Introduktion
- Del 1: en designers synsvinkel
- Del 2: en udviklers synsvinkel



Introduktion

- Organisation af store ensartede datamængder.
Ex: flyreservationer, CPR-registre,
nummeroplysninger, bibliotekssystemer,
karakterregistrering, finansverdenen, . . .
- Underliggende teknologi for mange
Internet-servicesider.
- Databasesprog \neq programmeringssprog.
 - sikkerhed
 - beregningssikkerhed
 - adgangssikkerhed
 - effektivitet



Del 1

Databasesystemer

— set fra en designers synsvinkel



Datamodellering: indsamling

Eksempelopgave

Lav et system til håndtering af information omkring institutter, deres ansatte og studerende.

Arbejdsgang

- Udvikl sproglig specifikation.
- Formalisér.
- Skab rammerne for datarepræsentation.
- Forbered udtræk af data til almindelige brugere.



Datamodellering: specifikation

- Institutter får en entydig institut-id. Deres postadresse, hovedtelefonnummer og årsbudget registreres.
- Medarbejdere registreres ved deres CPR-nummer med oplysninger om navn og kategori (VIP, DVIP, TAP, IL).
- En medarbejder er ansat på et eller flere institutter. Der er en ansættelsesdato for hvert ansættelsesforhold.
- Et institut har et antal forskningsledere. En forskningsleder er altid en af medarbejderne, og personen har ansvaret for et bestemt forskningsområde på instituttet.



Datamodellering: specifikation

- Studerende registreres ved deres CPR-nummer og navn sammen med oplysninger om immatrikulationsår og studieretning.
- Alle studerende har ét hjemmeinstitut.
- Nogle studerende er instruktorer. De får altid en kontorplads, som skal registreres.
- For de studerende, som er specialestuderende, registreres specialeemnet samt forventet afleveringsdato. En specialestuderende kan også være instruktør.



Datamodellering: specifikation

- Alle specialestuderende har én hovedvejleder, som er en af medarbejderne. Desuden kan en specialestuderende have et antal medvejledere.
- For kurser opbevares kode, titel, tidspunkt og lokale.
- Hvert kursus kan have et antal øvelseshold, som nummeres 1, 2, 3, osv. separat for hvert kursus. For de enkelte hold gemmes lokale og tidspunkt.
- Nogle kurser er projektkurser, hvor en studerende kan lave et projekt med en medarbejder som projektvejleder. Projektets omfang, startdato og slutdato registreres.



ER-diagrammer

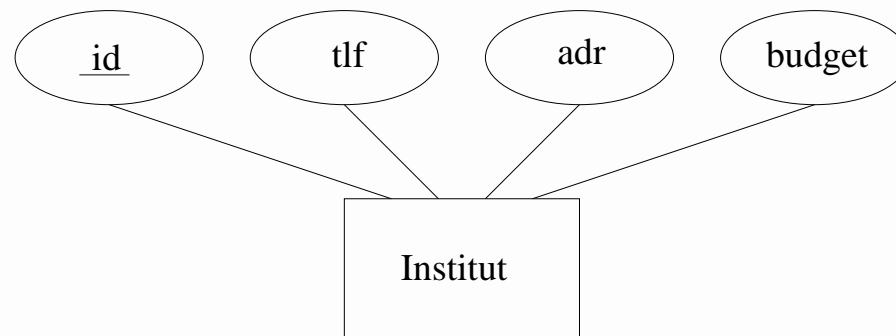
Der findes flere formalismer til beskrivelse af datasammenhænge.

En af disse er

Entity-Relationship modellen.

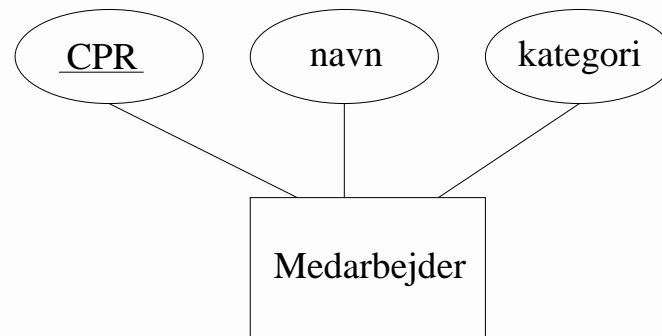
ER-diagrammer: entiteter

Institutter får en entydig institut-id. Deres postadresse, hovedtelefonnummer og årsbudget registreres.



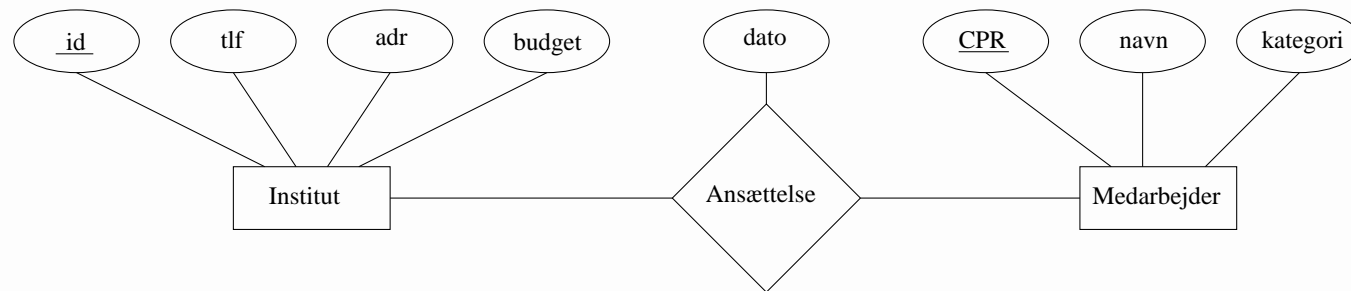
ER-diagrammer: entiteter

Medarbejdere registreres ved deres CPR-nummer med oplysninger om navn og kategori.



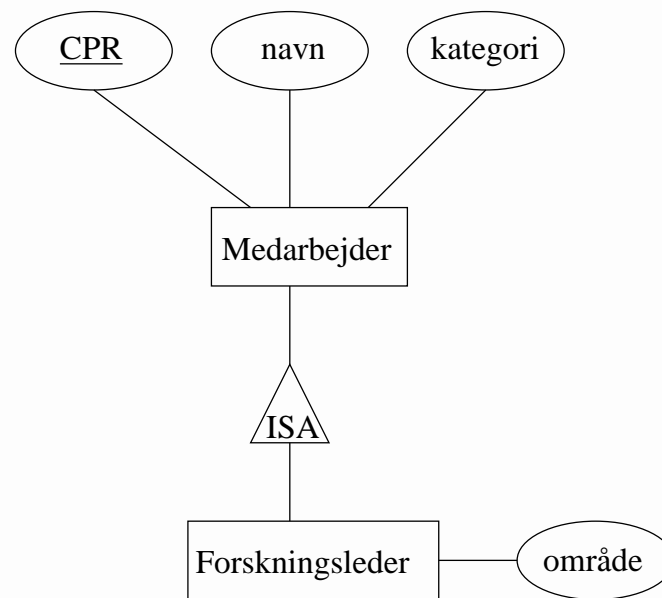
ER-diagrammer: relationer

En medarbejder er ansat på et eller flere institutter. Der er en ansættelsesdato for hvert ansættelsesforhold.



ER-diagrammer: ISA

Et institut har et antal forskningsledere. En forskningsleder er altid en af medarbejderne, og personen har ansvaret for et bestemt forskningsområde på instituttet.





Databasesprog

Databasesystemer

Der findes mange databasesystemer.

Ex: Postgres, DB2, Oracle, Access

Databasesprog

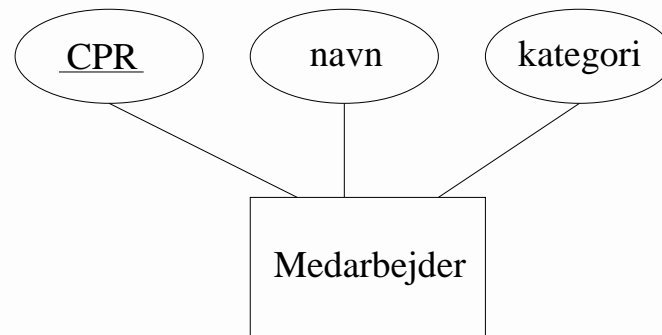
Der findes en del både traditionelle og nyere.

Men SQL

Structured Query Language

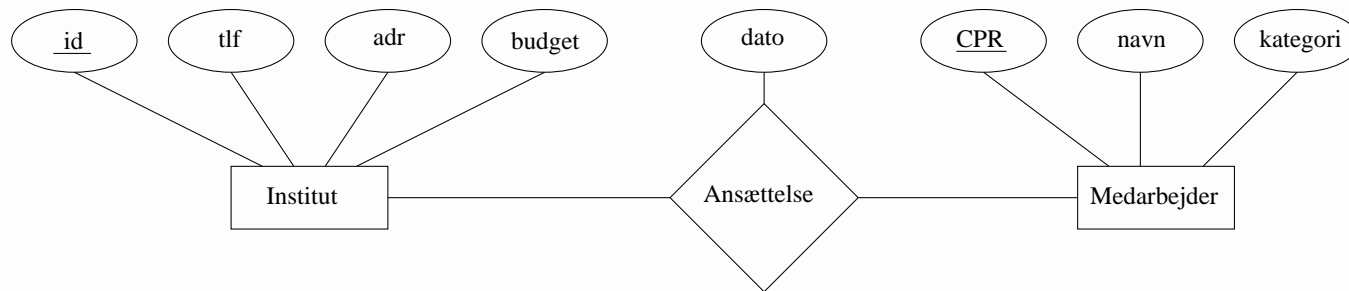
er de facto standard.

Tabeller i SQL: entiteter



```
create table Medarbejder ( CPR      char(10),  
                           navn      char(50),  
                           kategori  char(4),  
                           primary key (CPR) )
```

Tabeller i SQL: relationer



```
create table Ansæt ( CPR char(10),  
                    id integer,  
                    dato date,  
                    primary key (CPR, id),  
                    foreign key (CPR) references Medarbejder,  
                    foreign key (id) references Institut )
```


Tabeleksempel

Medarbejder

CPR	navn	kategori
1111111111	Joan Boyar	VIP
2222222222	Peter Schneider-Kamp	VIP
3333333333	Claus Michelsen	IL
4444444444	Kim Skak Larsen	VIP
5555555555	Lone Petterson	TAP
6666666666	Arun Vadiveal	DVIP
7777777777	Uffe Thorsen	DVIP



Terminologi

- En **tabel** kaldes også en **relation**.
- Databasesystemer opbygget omkring dette koncept kaldes ofte **relationelle** databaser.
- En søjles navn kaldes en **attribut**.
- En række i en tabel kaldes også et **tupel**.
- En **nøgle** er en minimal delmængde af attributter, der entydigt bestemmer tuplet.

Forespørgsler i SQL

CPR	navn	kategori
1111111111	Joan Boyar	VIP
2222222222	Peter Schneider-Kamp	VIP
3333333333	Claus Michelsen	IL
4444444444	Kim Skak Larsen	VIP
5555555555	Lone Petterson	TAP
6666666666	Arun Vadiveal	DVIP
7777777777	Uffe Thorsen	DVIP

```
select M.navn  
from Medarbejder M  
where M.kategori = 'VIP'
```

navn
Joan Boyar
Peter Schneider-Kamp
Kim Skak Larsen

Forespørgsler i SQL

Find Kim Skak Larsens instituts telefonnummer

```
select I.tlf
from Medarbejder M, Ansættelse A, Institut I
where M.CPR = A.CPR and A.id = I.id
      and M.navn = 'Kim Skak Larsen'
```

tlf

2387

Forespørgsler i SQL

Lav kategorioversigt

```
select M.kategori, count (*) as antal
from Medarbejder M
group by M.kategori
```

kategori	antal
VIP	3
IL	1
TAP	1
DVIP	2

Skemadesign: anomalier

navn	stillingskategori	timeløn i kr
Anita Jensen	rengøringsassistent	95
Cristiano Ronaldo	fodboldspiller	50.000
Jørgen Pryds	rengøringsassistent	95
Piotr Nowak	bygningarbejder	40
Pawel Kowalski	datalog	330
Børge Vestergaard	rengøringsassistent	95
Melisa Thomsen	datalog	330

Anomalier

- indsættelse: kan ikke registrere ny revisor.
- sletning: timelønnen slettes med sidste person.
- opdatering: inkonsistens, hvis ikke alle felter opdateres.

Skemadesign: afhængigheder

navn	stillingskategori	timeløn i kr
Anita Jensen	rengøringsassistent	95
Cristiano Ronaldo	fodboldspiller	50.000
Jørgen Pryds	rengøringsassistent	95
Piotr Nowak	bygningssarbejder	40
Pawel Kowalski	datalog	320
Børge Vestergaard	rengøringsassistent	95
Melisa Thomsen	datalog	320

Problem

Stillingskategorien bestemmer entydigt timelønnen.

Skemadesign: dekomposition

navn	stillingskategori
Anita Jensen	rengøringsassistent
Cristiano Ronaldo	fodboldspiller
Jørgen Pryds	rengøringsassistent
Piotr Nowak	bygningsarbejder
Pawel Kowalski	datalog
Børge Vestergaard	rengøringsassistent
Melisa Thomsen	datalog

stillingskategori	timeløn i kr
rengøringsassistent	95
fodboldspiller	50.000
bygningsarbejder	40
datalog	320



Skemadesign: normalformer

Ex: 3. normalform

Antag, at en delmængde af tabellens attributter X entydigt bestemmer en attribut A .

Vi kræver

- $A \in X$, eller
- X er en supernøgle for tabellen, eller
- A indgår i en nøgle for tabellen.

Check, og til dels løsning af normaliseringsproblemer, kan automatiseres.



Del 2

Databasesystemer

— set fra en udviklers synsvinkel



Evaluering af en forespørgsel

- En forespørgsel er en **specifikation**.
- Data organiseres.
- Evalueringsplaner vurderes.
- En god plan vælges og udføres.

En god plan er en, der minimerer disktilgang.

Evalueringsplaner: ombytning

Datainformation

1000 medarbejdere fordelt på 20 institutter, hvor af 3 har budget på mindst 100.000.000 kr.

Opgave

Find alle institutledere på institutter med budgetter på mindst 100.000.000 kr.

```
select M.navn
from Medarbejder M, Ansættelser A, Institutter I
where M.CPR = A.CPR and A.id = I.id and
        I.budget  $\geq$  100.000.000 and M.kategori = 'IL'
```



Evalueringsplaner: ombytning

Datainformation

1000 medarbejdere fordelt på 20 institutter, hvor af 3 har budget på mindst 100.000.000 kr.

Plan 1

1. Kombinér alle tabelindgangene.
2. Gennemløb og udvælg.

Resultat

Mindst $20 \times 1000 = 20.000$ operationer.



Evalueringsplaner: ombytning

Datainformation

1000 medarbejdere fordelt på 20 institutter, hvor af 3 har budget på mindst 100.000.000 kr.

Plan 2

1. Gennemløb og udvælg alle IL fra Medarbejder
2. Gennemløb og udvælg alle institutter, der har budget på mindst 100.000.000 kr, fra Institut.
3. Kombinér.

Resultat

Et sted mellem 1000 og 2000 operationer.



Evalueringsplaner: formulering

Man bruger et mindre antal relationelle operatører (relationel algebra) til at udtrykke en evalueringsplan.

Blandt andre:

- **select*** til at udvælge tupler.
- **project** til at udvælge attributter.
- **join** til at kombinere tabeller.

* Ikke det samme som nøgleordet select i SQL.

select

$B \leftarrow \text{select from } A \text{ where } X < 42 \text{ or } Y = 7$

A:

X	Y
3	19
51	9
52	7
33	20
63	15
14	7

B:

X	Y
3	19
52	7
33	20
14	7

project

$B \leftarrow \text{project } Y, Z \text{ from } A$

A:

X	Y	Z	Q
1	2	3	4
2	3	5	8
2	3	5	7
2	4	8	16

B:

Y	Z
2	3
3	5
4	8

join

$C \leftarrow \text{join } A \text{ and } B$

A:

X	Y
3	4
8	5
5	7
8	16

B:

Z	Q
2	3
3	5

C:

X	Y	Z	Q
3	4	2	3
8	5	2	3
5	7	2	3
8	16	2	3
3	4	3	5
8	5	3	5
5	7	3	5
8	16	3	5

join

$C \leftarrow \text{join } A \text{ and } B \text{ where } X = Q \text{ or } Y = Q$

A:

X	Y
3	4
8	5
5	7
8	16

B:

Z	Q
2	3
3	5

C:

X	Y	Z	Q
3	4	2	3
8	5	3	5
5	7	3	5

Evalueringsplaner: ombytning

Plan 1

A ← join Medarbejder and Ansættelser

B ← join A and Institutter

C ← select from B where M.CPR = A.CPR and A.id = I.id
and budget \geq 100.000.000 and kategori = 'IL'

D ← project navn from C

Evalueringsplaner: ombytning

Plan 2

A ← **select from** Medarbejder **where** kategori = 'IL'

B ← **select from** Institutter **where** budget \geq 100.000.000

C ← **join** A **and** Ansættelser

D ← **join** C **and** B

E ← **select from** D **where** M.CPR = A.CPR **and** A.id = I.id

F ← **project** navn **from** E



Evaluering og optimering

- operationsrækkefølge.
- valg af konkrete metode for operatorerne.
- index: find få udvalgte informationer hurtigt.
- pipelining.
- . . .



Et optimeringseksempel

Efter en forespørgsel ønsker vi ikke at se (og slet ikke at gemme) dubleret information (dvs. flere ens tupler i resultatet).

Derfor må vi lave et stort arbejde for at fjerne dubletter.

Vi vil helst kun gøre det, når det er nødvendigt.

Kan vi afgøre det på forhånd? Dvs. blot ved at se på forespørgslen. . .

Optimering: dubletter

Kasser med højde og bredde.

```
select distinct 2*(H+B) as D1 , B>5 as D2
from Kasser
```

Dubletter?

H	B
3	5
4	7
2	5
4	4

→

D1	D2

Optimering: dubletter

Kasser med højde og bredde.

```
select distinct 2*(H+B) as D1 , B>5 as D2
from Kasser
```

Dubletter?

H	B
3	5
4	7
2	5
4	4

→

D1	D2
16	false
22	true
14	false
16	false

Optimering: dubletter

Kasser med højde og bredde.

```
select distinct 2*(H+B) as D1, (H+B)*H as D2, B>5 as D3
from Kasser
```

Dubletter?

H	B
3	5
4	7
2	5
4	4

→

D1	D2	D3

Optimering: dubletter

Kasser med højde og bredde.

```
select distinct 2*(H+B) as D1, (H+B)*H as D2, B>5 as D3
from Kasser
```

Dubletter?

H	B
3	5
4	7
2	5
4	4

→

D1	D2	D3
16	24	false
22	44	true
14	14	false
16	32	false



Optimering: dubletter

Kasser med højde og bredde.

```
select distinct 2*(H+B) as D1, (H+B)*H as D2, B>5 as D3  
from Kasser
```

Dubletfri ved et tilfælde?

Optimering: dubletter

Kasser med højde og bredde.

select distinct $2*(H+B)$ as D1, $(H+B)*H$ as D2, $B > 5$ as D3
from Kasser

Dubletfri ved et tilfælde?

H	B		D1	D2	D3
?	?	→	16	24	false

$$2 * (H + B) = 16 \Rightarrow H + B = 8$$

$$(H + B)H = 24 \wedge H + B = 8 \Rightarrow H = 3$$

$$H + B = 8 \wedge H = 3 \Rightarrow B = 5$$



Beregningsmæssig sikkerhed

Databasesystemer er fundamentet for den finansielle verden! Fokus på **transaktioner**.

Mange andre aspekter af sikkerhed. Ex:

- Korrekthed af programmer.
- Sikker opbevaring af data.
- Sikker overførsel og autifikation.

Concurrency Control

Automat	Din konto
	700
“hæv 300”	
aflæs (700)	
beregn (700–300)	
gem (400)	400

Concurrency Control

Automat	Din konto	SU-styrelsen
	700	
“hæv 300” aflæs (700)		“indbetal 4500” aflæs (700)
beregn (700–300)		beregn (700 + 4500)
	5200	gem (5200)
gem (400)	400	



Concurrency Control

- Transaktioner skal være udelelige.
- Men vi ønsker stor parallellitet.

Transaktioner skal være

serialiserbare

Crash Recovery

SU-styrelsen	Din konto	
	RAM	Disk
aflæs...		700
	700	
aflæst (700)		
beregn (700 + 4500)		
gem (5200)		
	5200	
		5200

Crash Recovery

SU-styrelsen	Din konto	
	RAM	Disk
aflæs...		700
	700	
aflæst (700)		
beregn (700 + 4500)		
gem (5200)		
	5200	
	CRASH!	
		5200



Transaktionsbegreber

- Serialiserbarhed.
- Commit.
- Låseprotokoller.
- Write-ahead-log.
- Roll-back (undo).



Problemer med låse

Overførsler fra en konto til en anden.

A til B

B til C

C til A



Problemer med låse

Overførsler fra en konto til en anden.

A til B

B til C

C til A

lock_request(A)

Problemer med låse

Overførsler fra en konto til en anden.

A til B

B til C

C til A

A til B	B til C	C til A
lock_request(A)		
	lock_request(B)	

Problemer med låse

Overførsler fra en konto til en anden.

A til B	B til C	C til A
lock_request(A)		
	lock_request(B)	
A locked	B locked	lock_request(C)

Problemer med låse

Overførsler fra en konto til en anden.

A til B

B til C

C til A

lock_request(A)		
A locked	lock_request(B)	
lock_request(B)	B locked	lock_request(C)
		C locked

Problemer med låse

Overførsler fra en konto til en anden.

A til B	B til C	C til A
lock_request(A)		
A locked	lock_request(B)	
lock_request(B)	B locked	lock_request(C)
	lock_request(C)	C locked
		lock_request(A)

Problemer med låse

Overførsler fra en konto til en anden.

A til B	B til C	C til A
lock_request(A)		
A locked	lock_request(B)	
lock_request(B)	B locked	lock_request(C)
	lock_request(C)	C locked
		lock_request(A)

Deadlock!



Databasesystemer

En tværdatalogisk disciplin

- maskinarkitektur
- operativsystemer
- distribuerede systemer
- programmeringssprog
- algoritmer & datastrukturer
- kompleksitetsteori
- . . .