

IN2090 NOTATER - MODELLERING



Mohammad Fadel Al Khafaji

Introduksjon

Relasjonelle databaser: Forenklet beskrivelse

- ◆ En relasjonell database er en samling tabeller
- ◆ En tabell er også kalt en relasjon
- ◆ En tabell har
 - ◆ et navn,
 - ◆ en samling kolonner
 - ◆ og en samling rader (som er dataene)
- ◆ En kolonne har
 - ◆ et navn,
 - ◆ og en type

Customers			
CustomerID (int)	Name (text)	Birthdate (date)	NrProducts (int)
0	Anna Consuma	1978-10-09	19
1	Peter Young	2009-03-01	1
2	Carla Smith	1986-06-14	8
3	Sam Penny	1961-01-09	14
4	John Mill	1989-11-16	8
5	Yvonne Potter	1971-04-12	6

Supernøkkel: Mengde med attributter som alltid har unike verdier i en relasjon.

Kan bruke en supernøkkel for å unikt identifisere en rad (altså en ting). Hvis vi har en supernøkkel, vil alle utvidelser også være en supernøkkel. Mengden av alle attributter for en relasjon er alltid en supernøkkel.

Kandidatnøkler: En minimal supernøkkel. Supernøkler gir oss det vi trenger, men kan inneholde mye unødvendig. Supernøkkel er en kandidatnøkkel dersom det ikke går an å fjerne en attributt og fortsatt ha en supernøkkel.

Primærnøkkel: Gir mer mening å bruke kandidatnøkklene for å identifisere ting. En relasjon kan ha flere kandidatnøkler F.eks. {Fagkode, Emnenummer} og {Tittel} for Kurs. Vi velger derfor ut én som vi ønsker å bruke. Merk: Hvis en relasjon bare har én kandidatnøkkel må denne bli primærnøkkel.

Fremmednøkkel: Alle ting som representeres i databasen har en primærnøkkel. Kan så bruke verdiene fra denne primærnøkkel i andre relasjoner. Dette danner da en referanse fra en relasjon til en annen.

- Skriver ofte fremmednøkler med en pil F.eks.: Karakter(Student) -> Student(Brnavn)

Markere nøkler

- ◆ I forrige video lærte vi om signaturer, f.eks.:

Kurs(Fagkode, Emnenummer, Tittel, AntSP)

- ◆ Bruker gjerne understreking for å markere kandidatnøkler, f.eks.:

Kurs(Fagkode, Emnenummer, Tittel, AntSP)

- ◆ Hvis det er mer enn én kandidatnøkkel bruker vi gjerne dobbel understrek for primærnøkkel:

Kurs(Fagkode, Emnenummer, Tittel, AntSP)

- ◆ Kan også bare skrive dem etter signaturen med vanlig tekst, f.eks:

Kurs(Fagkode, Emnenummer, Tittel, AntSP)

Kandidatnøkler: {Fagkode, Emnenummer}, {Tittel}

Primærnøkkel: {Fagkode, Emnenummer}

eller

Kurs(Fagkode, Emnenummer, Tittel, AntSP)

Kandidatnøkler: Kurs(Fagkode, Emnenummer), Kurs(Tittel)

Primærnøkkel: Kurs(Fagkode, Emnenummer)

Relasjonsalgebra

Tilsvare SELECT

Projeksjon (π)

$$\pi_{\text{Brnavn, Etternavn}} \left(\begin{array}{|c|c|c|c|} \hline \text{Brnavn} & \text{Navn} & \text{Etternavn} & \text{Adresse} \\ \hline \text{evgenit} & \text{Evgenij} & \text{Thorstensen} & \text{Addr1} \\ \text{peternl} & \text{Petter} & \text{Nilsen} & \text{Addr2} \\ \text{leifhka} & \text{Leif H.} & \text{Karlsen} & \text{Addr3} \\ \hline \end{array} \right) = \begin{array}{|c|c|} \hline \text{Brnavn} & \text{Etternavn} \\ \hline \text{evgenit} & \text{Thorstensen} \\ \text{peternl} & \text{Nilsen} \\ \text{leifhka} & \text{Karlsen} \\ \hline \end{array}$$

- ◆ π er unær (tar ett argument)
- ◆ Har en mengde med attributter som subskrift
- ◆ Returnerer ny relasjon med kun attributtene listet opp
- ◆ Den velger altså ut attributter
- ◆ Merk: Operasjonene *endrer* ikke de originale relasjonene i databasen, bare returnerer en ny relasjon
- ◆ Merk: Resultatet er en relasjon uten navn. Kan gi den et navn slik:

$$\text{StudentNavn} := \pi_{\text{Brnavn, Etternavn}}(\text{Student})$$

Tilsvare WHERE

Seleksjon (σ)

$$\sigma_{\text{Emnernummer} \geq 3000 \wedge \text{AntSP} = 10} \left(\begin{array}{|c|c|c|c|} \hline \text{Fagkode} & \text{Emnernummer} & \text{Tittel} & \text{AntSP} \\ \hline \text{IN} & 2090 & \text{Databaser og datamodellering} & 10 \\ \text{IN} & 2010 & \text{Algoritmer og datastrukturer} & 10 \\ \text{AST} & 3220 & \text{Kosmologi 1} & 10 \\ \text{MAT} & 2000 & \text{Prosjektarbeid} & 10 \\ \text{BIOS} & 9312 & \text{Alpine Ecology} & 5 \\ \text{IN} & 4070 & \text{Logikk} & 10 \\ \hline \end{array} \right)$$
$$= \begin{array}{|c|c|c|c|} \hline \text{Fagkode} & \text{Emnernummer} & \text{Tittel} & \text{AntSP} \\ \hline \text{AST} & 3220 & \text{Kosmologi 1} & 10 \\ \text{IN} & 4070 & \text{Logikk} & 10 \\ \hline \end{array}$$

- ◆ σ er også unær (tar ett argument)
- ◆ Har et uttrykk med attributt-navnene som variable som subskrift
 - ◆ Bruker symbolene $\wedge, \vee, \neg, \geq, \leq, >, <, =$ og konstanter (3000, 10, IN, osv.)
- ◆ Returnerer ny relasjon med kun de tuplene som tilfredstiller uttrykket
- ◆ Men samme attributter
- ◆ Den velger altså ut de radene vi er interessert i

Omdøping (ρ)

$$\rho_{\text{Tittel} \rightarrow \text{Navn}, \text{AntSP} \rightarrow \text{Poeng}} \left(\begin{array}{|c|c|c|c|} \hline \text{Fagkode} & \text{Emnernummer} & \text{Tittel} & \text{AntSP} \\ \hline \text{IN} & 2090 & \text{Databaser og datamodellering} & 10 \\ \text{IN} & 2010 & \text{Algoritmer og datastrukturer} & 10 \\ \text{AST} & 3220 & \text{Kosmologi 1} & 10 \\ \text{MAT} & 2000 & \text{Prosjektarbeid} & 10 \\ \text{BIOS} & 9312 & \text{Alpine Ecology} & 5 \\ \text{IN} & 4070 & \text{Logikk} & 10 \\ \hline \end{array} \right)$$

$$= \begin{array}{|c|c|c|c|} \hline \text{Fagkode} & \text{Emnernummer} & \text{Navn} & \text{Poeng} \\ \hline \text{IN} & 2090 & \text{Databaser og datamodellering} & 10 \\ \text{IN} & 2010 & \text{Algoritmer og datastrukturer} & 10 \\ \text{AST} & 3220 & \text{Kosmologi 1} & 10 \\ \text{MAT} & 2000 & \text{Prosjektarbeid} & 10 \\ \text{BIOS} & 9312 & \text{Alpine Ecology} & 5 \\ \text{IN} & 4070 & \text{Logikk} & 10 \\ \hline \end{array}$$

- ◆ ρ er også unær (tar ett argument)
- ◆ Har en mengde med attributt-pil-attributt som subskrift
- ◆ $A \rightarrow B$ sier at A skal omdøpes til B
- ◆ Returnerer ny relasjon med nye attributt-navn ihht. omdøpingene
- ◆ Endrer ellers ingenting
- ◆ Enkelte kilder bruker en litt annen syntaks

Eksempel: Unære operasjoner

Kurs			
Fagkode	Emnernr	Tittel	AntSP
IN	2090	Databaser og datamodellering	10
IN	2010	Algoritmer og datastrukturer	10
AST	3220	Kosmologi 1	10
MAT	2000	Prosjektarbeid	10
IN	4070	Logikk	10
BIOS	9312	Alpine Ecology	5
IN	5800	Declarative data engineering	10
MAT	4500	Toplogi	10
IN	4230	Nettverk	10

Finn tittel på alle informatikk-masterkurs:

$$\rho_{\text{Tittel} \rightarrow \text{IfiMasterKurs}} \left(\pi_{\text{Tittel}} \left(\sigma_{\text{Fagkode} = 'IN' \wedge \text{Emnenummer} \geq 4000} (\text{Kurs}) \right) \right)$$

$$= \begin{array}{|c|} \hline \text{IfiMasterKurs} \\ \hline \text{Logikk} \\ \text{Declarative data engineering} \\ \text{Nettverk} \\ \hline \end{array}$$

Tilsvare CROSS JOIN

Kartesisk produkt (\times)

Brnavn	Navn	Etternavn	Adresse
evgenit	Evgenij	Thorstensen	Gateveien 1a
peternl	Petter	Nilsen	Stedplassen 23
leifhka	Leif H.	Karlsen	Bergfjellet 42

\times

Student	Fagkode	Emnennummer	Kara
evgenit	IN	2090	B
peternl	AST	3220	A
evgenit	IN	2010	B
leifhka	IN	2090	B
leifhka	MAT	2000	C

=

Brnavn	Navn	Etternavn	Adresse	Student	Fagkode	Emnennummer	Kara
evgenit	Evgenij	Thorstensen	Gateveien 1a	evgenit	IN	2090	B
evgenit	Evgenij	Thorstensen	Gateveien 1a	peternl	AST	3220	A
evgenit	Evgenij	Thorstensen	Gateveien 1a	evgenit	IN	2010	B
evgenit	Evgenij	Thorstensen	Gateveien 1a	leifhka	IN	2090	B
evgenit	Evgenij	Thorstensen	Gateveien 1a	leifhka	MAT	2000	C
peternl	Petter	Nilsen	Stedplassen 23	evgenit	IN	2090	B
peternl	Petter	Nilsen	Stedplassen 23	peternl	AST	3220	A
peternl	Petter	Nilsen	Stedplassen 23	evgenit	IN	2010	B
peternl	Petter	Nilsen	Stedplassen 23	leifhka	IN	2090	B
peternl	Petter	Nilsen	Stedplassen 23	leifhka	MAT	2000	C
leifhka	Leif H.	Karlsen	Bergfjellet 42	evgenit	IN	2090	B
leifhka	Leif H.	Karlsen	Bergfjellet 42	peternl	AST	3220	A
leifhka	Leif H.	Karlsen	Bergfjellet 42	evgenit	IN	2010	B
leifhka	Leif H.	Karlsen	Bergfjellet 42	leifhka	IN	2090	B
leifhka	Leif H.	Karlsen	Bergfjellet 42	leifhka	MAT	2000	C

- ◆ \times er binær (tar to argumenter)
- ◆ Returnerer ny relasjon med:
 - ◆ Alle attributtene til begge relasjonene
 - ◆ Alle kombinasjoner av tupler fra de to relasjonene

Tilsvare INNER JOIN / JOIN

Join (\bowtie)

Brnavn	Navn	Etternavn	Adresse
evgenit	Evgenij	Thorstensen	Gateveien 1a
peternl	Petter	Nilsen	Stedplassen 23
leifhka	Leif H.	Karlsen	Bergfjellet 42

$\bowtie_{\text{Brnavn=Student}}$

Student	Fagkode	Emnennummer	Kara
evgenit	IN	2090	B
peternl	AST	3220	A
evgenit	IN	2010	B
leifhka	IN	2090	B
leifhka	MAT	2000	C

=

Brnavn	Navn	Etternavn	Adresse	Student	Fagkode	Emnennummer	Kara
evgenit	Evgenij	Thorstensen	Gateveien 1a	evgenit	IN	2090	B
evgenit	Evgenij	Thorstensen	Gateveien 1a	evgenit	IN	2010	B
peternl	Petter	Nilsen	Stedplassen 23	peternl	AST	3220	A
leifhka	Leif H.	Karlsen	Bergfjellet 42	leifhka	IN	2090	B
leifhka	Leif H.	Karlsen	Bergfjellet 42	leifhka	MAT	2000	C

- ◆ \bowtie er binær (tar to argumenter)
- ◆ Har en mengde med attributt=attributt som subskrift
- ◆ $A = B$ sier at A skal være lik B i resultatet
- ◆ Kan også ha mer generelle uttrykk (slik som for σ)
- ◆ Returnerer ny relasjon med alle kombinasjoner av tupler fra de to relasjonene som tilfredstiller uttrykket (med alle attributtene fra begge relasjonene)
- ◆ Merk: $R \bowtie_e P$ er ekvivalent med $\sigma_e(R \times P)$.

Naturlig join (\star)

Student	Fagkode	Emnennummer	Kara		Fagkode	Emnennummer	Tittel	AntSP
evgenit	IN	2090	B	\star	IN	2090	Databaser og datamodellering	10
peternl	AST	3220	A		IN	2010	Algoritmer og datastrukturer	10
evgenit	IN	2010	B		AST	3220	Kosmologi 1	10
leifhka	IN	2090	B		MAT	2000	Prosjektarbeid	10
leifhka	MAT	2000	C		BIOS	9312	Alpine Ecology	5
					IN	4070	Logikk	10

Student	Fagkode	Emnennummer	Kara	Tittel	AntSP
evgenit	IN	2090	B	Databaser og datamodellering	10
peternl	AST	3220	A	Kosmologi 1	10
evgenit	IN	2010	B	Algoritmer og datastrukturer	10
leifhka	IN	2090	B	Databaser og datamodellering	10
leifhka	MAT	2000	C	Prosjektarbeid	10

- ◆ \star er binær (tar to argumenter)
- ◆ Utfører en join på alle attributter med likt navn (og velger ut én kopi av hver)
- ◆ Dersom ingen attributter har likt navn får man det kartesiske produktet (\times)
- ◆ Må bruke ρ dersom enten:
 - ◆ Man ønsker å joine på attributter med ulikt navn (bruker ρ for å gjøre dem like)
 - ◆ Man ønsker ikke å joine på to attributter som har likt navn (bruker ρ for å gjøre dem ulike)

Snitt, union og differanse (\cap , \cup , \setminus)

Bachelorkurs			
Fagkode	Emnernr	Tittel	AntSP
IN	2090	Databaser og datamodellering	10
IN	2010	Algoritmer og datastrukturer	10
AST	3220	Kosmologi 1	10
MAT	2000	Prosjektarbeid	10

Masterkurs			
Fagkode	Emnernr	Tittel	AntSP
IN	4070	Logikk	10
BIOS	9312	Alpine Ecology	5
IN	5800	Declarative data engineering	10
MAT	4500	Toplogi	10

Bachelorkurs \cup Masterkurs =

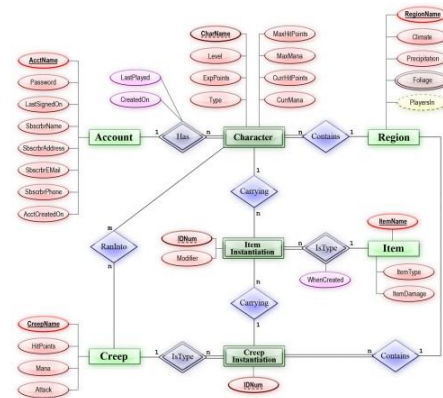
Fagkode	Emnernr	Tittel	AntSP
IN	2090	Databaser og datamodellering	10
IN	2010	Algoritmer og datastrukturer	10
AST	3220	Kosmologi 1	10
MAT	2000	Prosjektarbeid	10
IN	4070	Logikk	10
BIOS	9312	Alpine Ecology	5
IN	5800	Declarative data engineering	10
MAT	4500	Toplogi	10

- ◆ \cup , \cap og \setminus er alle binær (tar to argumenter)
- ◆ Utfører vanlig union, snitt og differanse på mengden av tupler
- ◆ Krever at begge argument-relasjonene har samme signatur (attributt-navn og typer)

ER-modellering

ER

- ◆ ER (Entity-Relationship) er et visuelt modelleringsspråk
- ◆ Enkelt diagramspråk: bokser, ovaler, diamanter og streker
- ◆ Har 3 byggeklosser og streker mellom disse: Entiteter, attributter og relasjoner
- ◆ Kan uttrykke ulike egenskaper ved disse vha. visuelle markeringer
- ◆ Inneholder kun det som er relevant for relasjonsmodellen

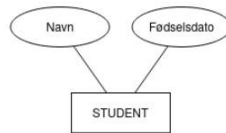


Entiteter

STUDENT

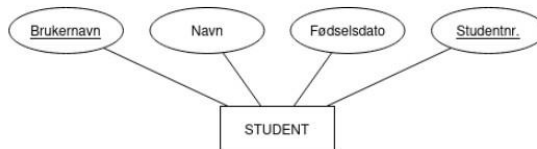
- ◆ Entitet: En konkret ting (Kari Eriksen, IN2090, Ole Johan Dahlshus, osv.)
- ◆ Entitetstype: Mengde entiteter med samme egenskaper/attributter
- ◆ Bruker ofte ordet "entitet" og "entitetstype" om hverandre
- ◆ Representeres i ER med et rektangel
- ◆ Har et unikt navn
- ◆ Har verdier knyttet til seg: attributter

Attributter



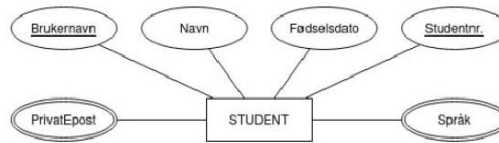
- ◆ Representeres ved ovaler i ER
- ◆ Knyttes til nøyaktig én entitetstype med en strek
- ◆ Har et unikt navn innenfor samme entitet
- ◆ Alle entiteter *kan* ha en verdi knyttet til attributten
- ◆ Alle entiteter *kan kun* ha verdier knyttet til en av entitetstypens attributter

Nøkkel-attributt



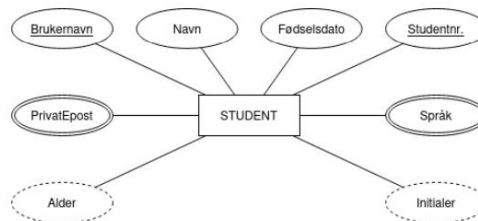
- ◆ Unike attributter kalles *nøkler*
- ◆ Markeres med en understrek under navnet
- ◆ Alle entiteter *må* ha en verdi for hver nøkkel

Flerverdi-attributt



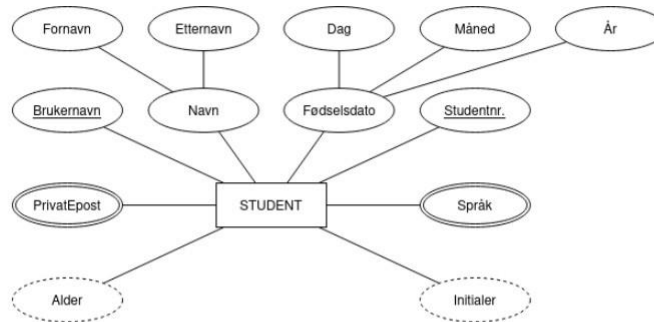
- ◆ Attributter hvor hver entitet kan ha flere verdier kalles *flerverdi*-attributter
- ◆ Markeres med dobbel-oval
- ◆ Kan ikke være nøkkel

Utledbar attributt



- ◆ Attributter hvis verdi kan utledes fra andre attributters verdi kalles *utledbar*
- ◆ Markeres med stiplet oval
- ◆ Merk: Sier ikke i ER-diagrammet *hvordan* den kan utledes!
- ◆ Litt vagt, men overordnet: Kan utledes vha.
 - ◆ Andre attributter i diagrammet
 - ◆ Vanlige operasjoner (+, -, ·, ÷ på tall, split, concat på tekst, osv.)
 - ◆ Vanlige konstanter og bakgrunnskunnskap (tall og bokstaver, nå, dagens dato, dollarkurs, osv.)

Sammensatt attributt

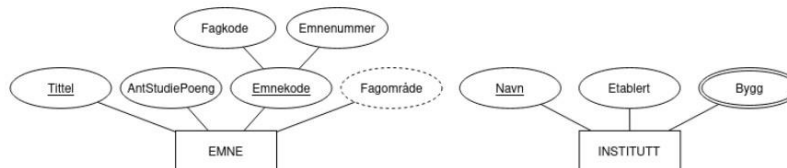


- ◆ En attributt kan bestå av flere attributter
- ◆ Kalles da *sammensatt*
- ◆ En sammensatt attributt kan bestå av sammensatte attributter
- ◆ Danner et tre
- ◆ Dersom en sammensatt attributt er nøkkel er det kombinasjonen av attributtene den består av som er unik

Eksempel

Lag en modell som modellerer følgende:

- ◆ Et emne har en unik tittel (f.eks. *Logikk*) og et antall studiepoeng (f.eks. *10*)
- ◆ Emner har også en unik emnekode (f.eks. *IN2090*)
- ◆ Emnekoden består igjen av en fagkode (f.eks. *IN*) og et emnenummer (f.eks. *2090*)
- ◆ Emner har et fagområde (f.eks. *informatikk*) som kan utledes fra fagkoden
- ◆ Et institutt har et unikt navn og et år det ble etablert
- ◆ Institutter har også en eller fler navn på bygg hvor de holder til



Relasjoner

Relasjoner er forhold som holder mellom entiteter. Representeres med en diamant i ER. Har et unikt navn. Relaterer vanligvis to entiteter, men kan relatere flere. Strekene mellom relasjonen og entitetstypene kalles deltakelser

Attributter på relasjoner: I likhet med entitetstyper kan også relasjoner ha attributter. Disse gir verdier som knyttes til forholdet mellom entitetene. Kan her ikke ha nøkler.

Øvre skranker

Et viktig aspekt ved relasjoner er hvor mange entiteter en entitet kan være relatert til. Vi skiller da kun på om det er én eller mange (altså potensielt flere enn 1). Mange representeres med en bokstav, slik som N eller M. Tallet/bokstaven lengst vekk fra en entitet sier hvor mange den entiteten høyst kan være relatert til. Disse kalles øvre skranker.

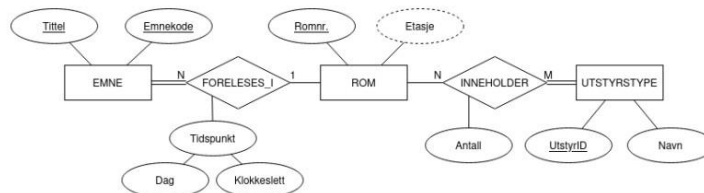
Nedre skranker

Tilsvarende, er et viktig aspekt ved relasjoner hvor mange entiteter en entitet må være relatert til. Vi skiller da kun på om det er minst én eller minst null. Minst null representeres med enkel linje, kalles partiell deltakelse. Minst én representeres med dobbel linje, kalles total deltakelse. Streken nærmest en entitet sier antall den entiteten minst må være relatert til. "Nøyaktig én" representeres med total deltakelse og 1

Eksempel

Lag en modell som modellerer følgende:

- ◆ Et rom har et unikt romnummer, samt en etasje (etasjen rommet befinner seg på), men dette kan utledes fra romnummeret
- ◆ En utstyrstype har en unik utstyrsID og et navn (slik som "tavle", "prosjektor", osv.)
- ◆ Et rom kan inneholde mange utstyr (altså flere typer utstyr) og en utstyrstype kan være inneholdt i mange rom, men vi lagrer kun de utstyrstypene som er inneholdt i minst ett rom
- ◆ Et rom kan inneholde flere ting av samme utstyrstype, og vi ønsker også å lagre dette antallet
- ◆ Et emne (likt som tidligere i denne videoen) foreleses i et rom på et fast ukentlig tidspunkt bestående av dag og klokkeslett
- ◆ Et emne foreleses i nøyaktig ett rom (altså minst og høyst ett), men et rom kan ha mange emner som foreleses i det rommet, og et rom trenger ikke brukes til forelesninger

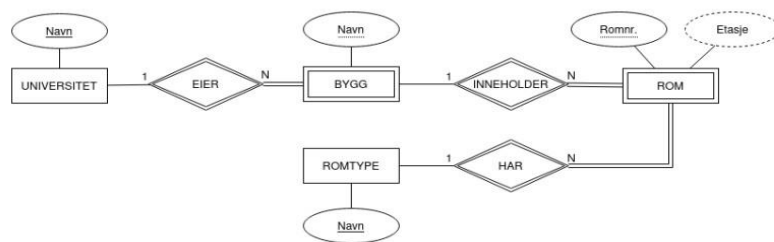


Svake entiteter

En svak entitet er en entitet som har en nøkkel som avhenger av en annen entitets nøkkel. En svak entitet har en nøkkel som kun er unik i en kontekst (gitt ved en relasjon til en annen entitet). Egen notasjon for dette i ER: Stiplet nøkkel-markering og dobbel boks

Identifiserende entitet og relasjon: Men vi må også markere hvilken relasjon det er som angir denne konteksten. Gjøres med dobbel diamant og kalles identifiserende relasjon/entitet. Nødvendig fordi den svake entiteten kan være relatert til mange andre entiteter. Merk: Svake entiteter må alltid være relatert til nøyaktig én via den identifiserende relasjonen.

Flere identifiserende entiteter

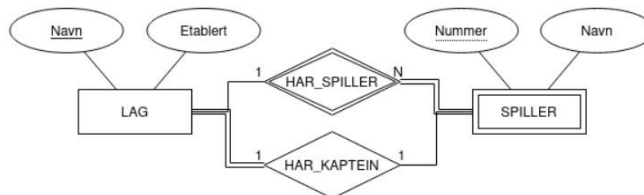


- ◆ En svak entitet kan også avhenge av flere enn én annen entitet
- ◆ Kan også ha flere nivåer med svake entiteter
- ◆ Her vil ROM ha en total nøkkel bestående av
 - ◆ UNIVERSITETets Navn
 - ◆ BYGGets Navn
 - ◆ ROMTYPEns Navn
 - ◆ ROMets Romnr.

Eksempel

Lag en modell som modellerer følgende:

- ◆ Et lag har et unikt navn og et år laget ble etablert
- ◆ En spiller spiller på nøyaktig ett lag, men et lag kan ha mange spillere (men trenger ikke ha noen)
- ◆ Et spiller har et navn, og et draktnummer som er unikt innad i laget spilleren spiller på
- ◆ Et lag kan også ha en kaptein, og et lag har nøyaktig én kaptein og en spiller kan høyst være kaptein for ett lag (men må ikke være kaptein for noe lag)



Ternær og n-ær relasjon

En ternær relasjon er en relasjon som relaterer 3 entiteter av gangen. N-ær relasjon relaterer N entiteter av gangen (for et tall N). Uvanlige, men nyttige i noen sammenhenger. Fokuserer på ternær, men prinsippene er like for $N > 3$. Ellers likt som binære.

- Kombinasjonen av de relaterte elementene er unike
- Kan ha attributter på relasjonen

Øvre skranker

Øvre skranker sier hvor mange entiteter vi kan ha av én entitetstype, gitt én kombinasjon av entiteter fra de andre entitetstypene.

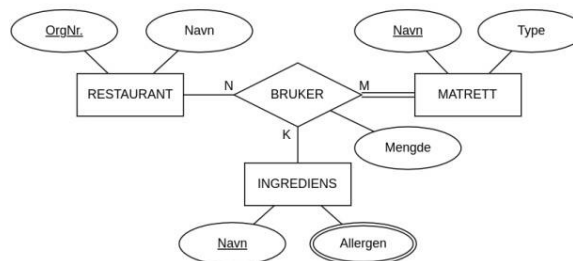
Nedre skranker

Nedre skranker sier hvor mange entiteter vi må ha av én entitetstype i relasjonen.

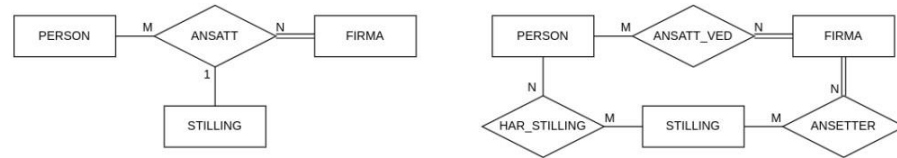
Eksempel

Lag en modell som modellerer følgende:

- ◆ En restaurant har et unikt organisasjonsnummer og et navn
- ◆ En matrett har et unikt navn og en type (f.eks. dessert, hovedrett, forrett, osv.)
- ◆ En ingrediens har et unikt navn, samt en mengde allergener (f.eks. gluten, soya, osv.)
- ◆ En restaurant bruker ingredienser i matretter, hvor hver matrett må inneholde minst én ingrediens på minst én restaurant, og:
 - ◆ Gitt en matrett og en ingrediens kan det være mange (N) restauranter;
 - ◆ gitt en ingrediens og en restaurant kan det være mange (M) matretter;
 - ◆ og gitt en restaurant og en matrett kan det være mange (K) ingredienser
- ◆ Vi ønsker å lagre mengden (gram) av en ingrediens en restaurant bruker i en matrett



Binær vs. ternær

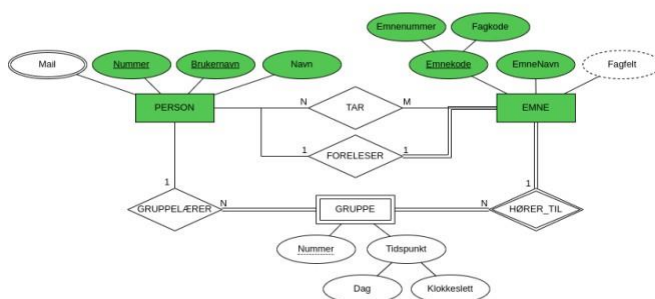


- ◆ Ternærrelasjoner er ikke ekvivalente med 3 binære
- ◆ De to modellene uttrykker forskjellige ting
- ◆ I modellen til høyre kan vi f.eks. ikke vite hvilken stilling en person har i et firma
- ◆ Også forskjellig hvilke skranker vi kan sette
- ◆ Må velge det som passer med informasjonen man ønsker å uttrykke

Datamodellering: Realisering

Realisering er en prosess/algoritme som oversetter en ER-model til et relasjonsskjema. Krever mange valg og ekstra input fra mennesker. Vi gjør derfor prosessen manuelt. I valgene ønsker vi å minimere antall relasjoner og antall NULL-verdier. Prosessen oversetter de ulike delene av en ER-model i en bestemt rekkefølge. Rekkefølge: Entiteter, svake entiteter, 1-1-relasjoner, 1-N-relasjoner, M-N-relasjoner, flerverdi-attributter, ternære/N-ære relasjoner.

Realisering av (normale) entiteter

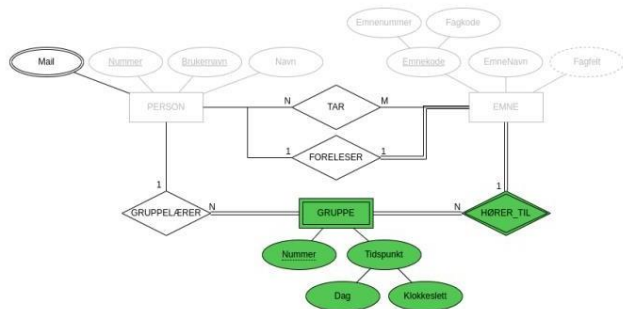


Person(nummer, brukernavn, navn)
- KN: {nummer}, {brukernavn}
- PN: {brukernavn}

Emne(emnenummer, fagkode, emnenavn)
- KN/PN: {emnenummer, fagkode}

- ◆ Entiteter blir relasjoner
- ◆ Ignorerer utledbare og venter med flerverdi
- ◆ Attributter blir relasjons-attributter (kun løv-noder for sammensatte)
- ◆ Nøkler blir kandidatnøkler (KN), må velge én primærnøkkel (PN)

Realisering av svake entiteter



Person(nummer, brukernavn, navn)

- KN: {nummer}, {brukernavn}

- PN: {brukernavn}

Emne(emnenummer, fagkode, emnenavn)

- KN/PN: {emnenummer, fagkode}

Gruppe(nummer, dag, klokkeslett, emnenummer, fagkode)

- KN/PN: {nummer, emnenummer, fagkode}

- FN: (emnenummer, fagkode) -> Emne(emnenummer, fagkode)

- ◆ Svake entiteter blir også relasjoner
- ◆ Får også PN til identifiserende entitet(er)s relasjon(er) som attributter
- ◆ PN blir identifiserendes PN + svak nøkkel
- ◆ Realiserer så den identifiserende relasjonen(e) via FN

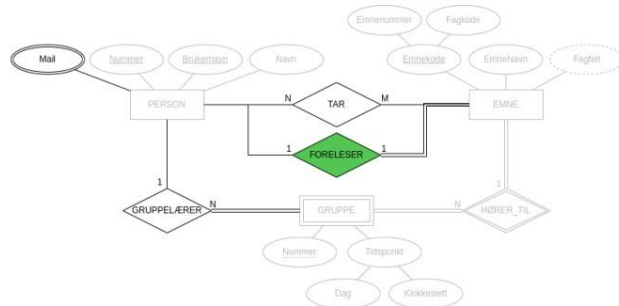
Realisering av 1-1-relasjoner

Fire alternativer

1. Sammensmeltning (merge)
2. FN fra Person
3. FN fra Emne
4. Ny relasjon

1. Sammensmeltning (merge)

Realisering av 1-1-relasjoner: Sammensmeltning



Gruppe(nummer, dag, klokkeslett, brukernavn)

- KN/PN: {nummer, brukernavn}
- FN: (brukernavn) -> Foreleser(brukernavn)

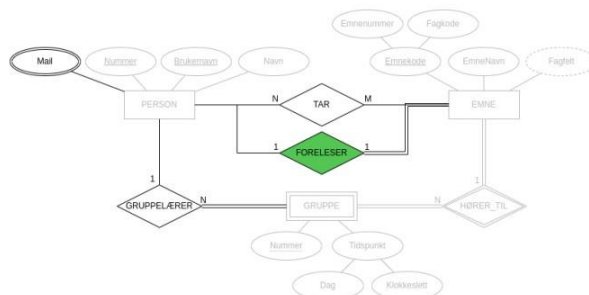
Foreleser(nummer, brukernavn, navn, emnenummer, fagkode, emnenavn)

- KN: {nummer}, {brukernavn}, {emnenummer, fagkode}
- PN: {brukernavn}

- ◆ Lager en ny relasjon som er sammensmeltningen av Person og Emne
- ◆ Får da alle deres attributter (må velge PN)
- ◆ Fjerner de gamle (og oppdaterer FNER)
- ◆ Kan kun brukes hvis begge deltakelsene er totale

2. FN fra Person

Realisering av 1-1-relasjoner: FN fra Person



Person(nummer, brukernavn, navn, emnenummer, fagkode)

- KN: {nummer}, {brukernavn}
- PN: {brukernavn}
- FN: (emnenummer, fagkode) -> Emne(emnenummer, fagkode)

Emne(emnenummer, fagkode, emnenavn)

- KN/PN: {emnenummer, fagkode}

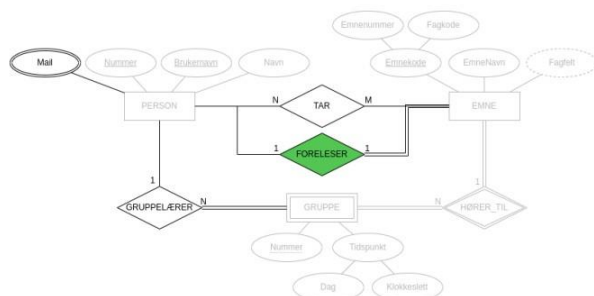
Gruppe(nummer, dag, klokkeslett, emnenummer, fagkode)

- KN/PN: {nummer, emnenummer, fagkode}
- FN: (emnenummer, fagkode) -> Emne(emnenummer, fagkode)

- ◆ Legger til FN fra Person til Emnes PN
 1. Legger til attributter tilsvarende Emnes PN
 2. Markerer dem som FN
 3. Ville blitt KN hvis PERSON hadde total deltakelse

3. FN fra Emne

Realisering av 1-1-relasjoner: FN fra Emne



Person(nummer, brukernavn, navn)

- KN: {nummer}, {brukernavn}

- PN: {brukernavn}

Emne(emnenummer, fagkode, emnenavn, foreleser)

- KN/PN: {emnenummer, fagkode}, {foreleser}

- FN: {foreleser} → Person(brukernavn)

Gruppe(nummer, dag, klokkeslett, emnenummer, fagkode)

- KN/PN: {nummer, emnenummer, fagkode}

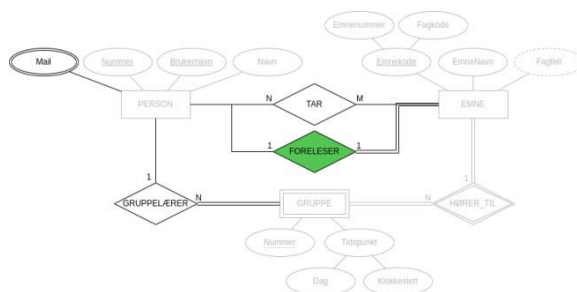
- FN: {emnenummer, fagkode} → Emne(emnenummer, fagkode)

◆ Legger til FN fra Emne til Persons PN

1. Legger til attributter tilsvarende Persons PN
2. Markerer dem som FN
3. Blir KN siden EMNE har total deltakelse

4. Ny relasjon

Realisering av 1-1-relasjoner: Ny relasjon



Person(nummer, brukernavn, navn)

- KN: {nummer}, {brukernavn}

- PN: {brukernavn}

Emne(emnenummer, fagkode, emnenavn)

- KN/PN: {emnenummer, fagkode}

Gruppe(nummer, dag, klokkeslett, emnenummer, fagkode)

- KN/PN: {nummer, emnenummer, fagkode}

- FN: {emnenummer, fagkode} → Emne(emnenummer, fagkode)

Foreleser(brukernavn, emnenummer, fagkode)

- KN: {brukernavn}, {emnenummer, fagkode}

- PN: {brukernavn}

- FN: {brukernavn} → Person(brukernavn)

{emnenummer, fagkode} → Emne(emnenummer, fagkode)

- ◆ Lager ny relasjon med referenser til entitetenes relasjoners PNER
- ◆ Får KN lik hver av disse PNene (og velger PN)
- ◆ Legger til FNER til disse PNene

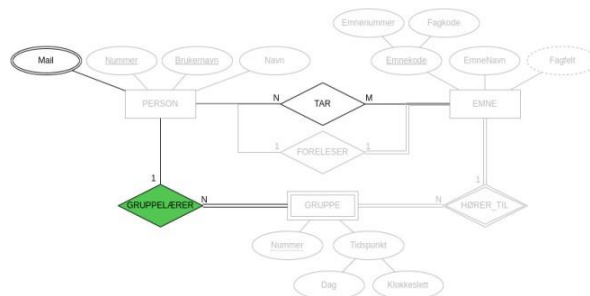
Realisering av 1-N-relasjoner

To valg:

1. Legge FN inn i N-siden (GRUPPE)
2. Ny relasjon

1. Legge FN inn i N-siden (GRUPPE)

Realisering av 1-N-relasjoner: FN i N-siden



Person(nummer, brukernavn, navn)
- KN: {nummer}, {brukernavn}
- PN: {brukernavn}

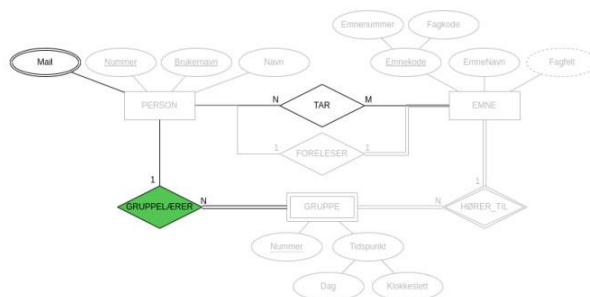
Emne(emnenummer, fagkode, emnenavn, foreleser)
- KN/PN: {emnenummer, fagkode}, {foreleser}
- FN: {foreleser} → Person(brukernavn)

Gruppe(nummer, dag, klokkeslett, emnenummer, fagkode, grlærer)
- KN/PN: {nummer, emnenummer, fagkode}
- FN: {emnenummer, fagkode} → Emne(emnenummer, fagkode)
(grlærer) → Person(brukernavn)

- ◆ Tilsvarende som FN for 1-1

2. Ny relasjon

Realisering av 1-N-relasjoner: Ny relasjon



Person(nummer, brukernavn, navn)
- KN: {nummer}, {brukernavn}
- PN: {brukernavn}

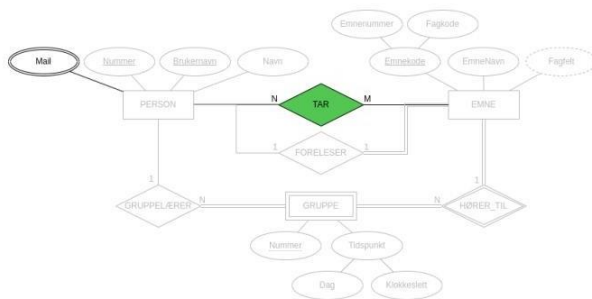
Emne(emnenummer, fagkode, emnenavn, foreleser)
- KN/PN: {emnenummer, fagkode}, {foreleser}
- FN: {foreleser} → Person(brukernavn)

Gruppe(nummer, dag, klokkeslett, emnenummer, fagkode)
- KN/PN: {nummer, emnenummer, fagkode}
- FN: {emnenummer, fagkode} → Emne(emnenummer, fagkode)

Gruppelærer(brukernavn, nummer, emnenummer, fagkode)
- KN/PN: {nummer, emnenummer, fagkode}
- FK: {brukernavn} → Person(brukernavn)
(nummer, emnenummer, fagkode) →
Gruppe(nummer, emnenummer, fagkode)

- ◆ Tilsvarende som for 1-1
- ◆ N-sidens attributter blir PN

Realisering av N-M-relasjoner



Person(nummer, brukernavn, navn)
- KN: {nummer}, {brukernavn}
- PN: {brukernavn}

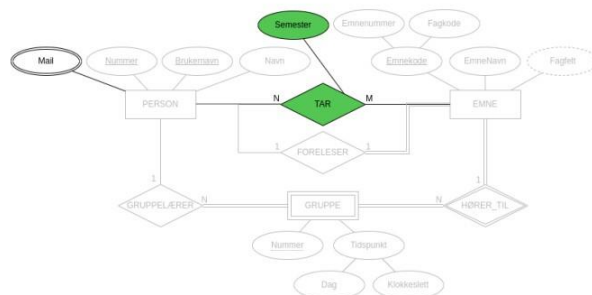
Emne(emnenummer, fagkode, emnenavn, foreleser)
- KN/PN: {emnenummer, fagkode}, {foreleser}
- FN: {foreleser} -> Person(brukernavn)

Gruppe(nummer, dag, klokkeslett, emnenummer, fagkode, grlærer)
- KN: {nummer, emnenummer, fagkode}, {grlærer}
- FN: (emnenummer, fagkode) -> Emne(emnenummer, fagkode)
(grlærer) -> Person(brukernavn)

Tar(brukernavn, emnenummer, fagkode)
- KN/PN: {brukernavn, emnenummer, fagkode}
- FN: (brukernavn) -> Person(brukernavn)
(emnenummer, fagkode) -> Emne(emnenummer, fagkode)

- ◆ Bare ett valg: Ny relasjon
- ◆ Tilsvarende 1-1 og 1-N
- ◆ PN lik alle attributter

Realisering av relasjoner med attributter



Person(nummer, brukernavn, navn)
- KN: {nummer}, {brukernavn}
- PN: {brukernavn}

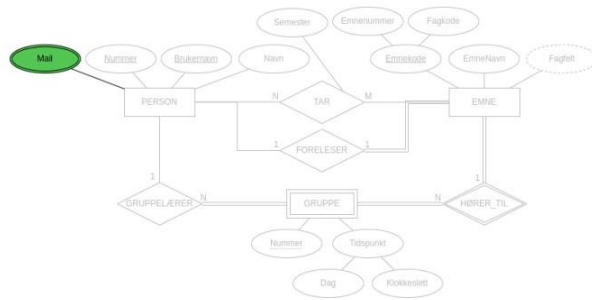
Emne(emnenummer, fagkode, emnenavn, foreleser)
- KN/PN: {emnenummer, fagkode}, {foreleser}
- FN: {foreleser} -> Person(brukernavn)

Gruppe(nummer, dag, klokkeslett, emnenummer, fagkode, grlærer)
- KN: {nummer, emnenummer, fagkode}, {grlærer}
- FN: (emnenummer, fagkode) -> Emne(emnenummer, fagkode)
(grlærer) -> Person(brukernavn)

Tar(brukernavn, emnenummer, fagkode, semester)
- KN/PN: {brukernavn, emnenummer, fagkode}
- FN: (brukernavn) -> Person(brukernavn)
(emnenummer, fagkode) -> Emne(emnenummer, fagkode)

- ◆ Attributter på relasjoner blir bare attributter i relasjonens realisering
- ◆ Tilsvarende som for entiteter
- ◆ Merk: Blir ikke del av KN/PN!

Realisering av flerverdi-attributter



- ◆ Blir egen relasjon som relaterer entitetens relasjons PN til attributten
- ◆ KN/PN lik alle attributtene i relasjonen
- ◆ FN til entitetens relasjons PN

Person(nummer, brukernavn, navn)
 - KN: {nummer}, {brukernavn}
 - PN: {brukernavn}

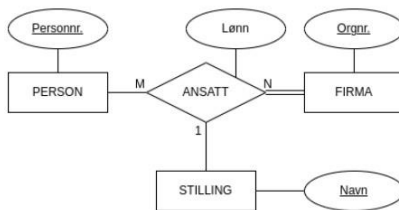
Emne(emnenummer, fagkode, emne navn, foreleser)
 - KN/PN: {emnenummer, fagkode}, {foreleser}
 - FN: {foreleser} → Person(brukernavn)

Gruppe(nummer, dag, klokkeslett, emnenummer, fagkode, grlærer)
 - KN: {nummer, emnenummer, fagkode}, {grlærer}
 - FN: (emnenummer, fagkode) → Emne(emnenummer, fagkode)
 (grlærer) → Person(brukernavn)

Tar(brukernavn, emnenummer, fagkode, semester)
 - KN/PN: {brukernavn, emnenummer, fagkode}
 - FN: (brukernavn) → Person(brukernavn)
 (emnenummer, fagkode) → Emne(emnenummer, fagkode)

Mail(brukernavn, mail)
 - KN/PN: {brukernavn, mail}
 - FN: (brukernavn) → Person(brukernavn)

Realisering av ternære relasjoner



- ◆ Tilsvarende som for M-N-relasjoner (uavhengig av kardinaliteter)
- ◆ Altså, alltid egen relasjon
- ◆ Må sette KN/PN i henhold til kardinalitetene
- ◆ Helt likt for alle N-ære relasjoner

Person(personnr)
 - KN/PN: {personnr}

Firma(orgnr)
 - KN/PN: {orgnr}

Stilling(navn)
 - KN/PN: {navn}

Ansatt(person, firma, stilling)
 - KN/PN: {person, firma}
 - FN: (person) → Person(personnr)
 (firma) → Firma(orgnr)
 (stilling) → Stilling(navn)