



Økt 9 (av 12)

DB1100 Databaser

(Tomas Sandnes / tomas.sandnes@kristiania.no)

Dagens temaer

Dagens pensum: [Læreboka, kapittel 8.2.1 - 8.2.4](#)

- Status arbeidskrav del 2
- Fra forrige økt:
 - Resultater Kahoot
 - Repetisjon enkelte elementer fra modellering ER
- Dagens tema: [Normalisering](#)
 - Hensikt
 - 1. til 3. normalform (1NF, 2NF og 3NF)

Status arbeidskrav del 2

- Alle dere som *ikke leverte arbeidskrav*:
 - Se bort fra denne sliden. :-)
- Fristen for å levere video for arbeidskrav #2 var søndag 21. oktober.
 - Dere får snart (senest kommende helg!) en bekreftelse på levert video samt link til video dere nå skal gjøre peer review for (omtrent som sist).

Resultat, forrige Kahoot

- Forrige gang slo dere fjorårets studenter igjen!
 - Stillingen er 5-1 til dere. :-)
- Dere: **67,79 % riktig svar!**
 - (I fjor: 66,95 % riktig.)

Game Over	
Show feedback	
RealTafk	12,448
Jonas	11,867
Ensomjente_99	11,175
Modellering 🙄	11,139
Eir	11,111

DB1100 - økt 8 - Modellering2	
Played on	18 Oct 2018
Hosted by	Tomas_Sandnes
Played with	106 players
Played	10 of 10 questions

Overall Performance	
Total correct answers (%)	67,79%
Total incorrect answers (%)	32,21%
Average score (points)	5724,92 points

Play Again 

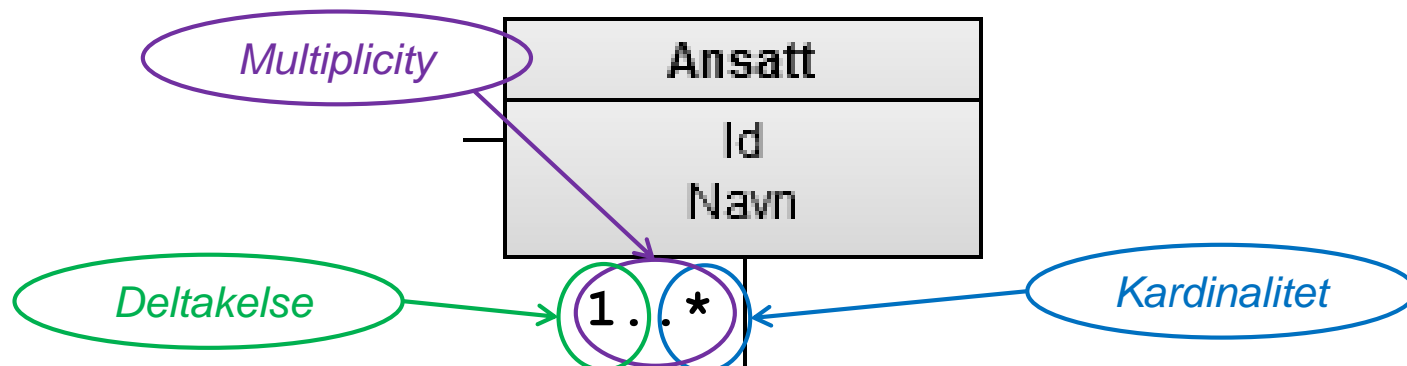
New Game 

Lærdom fra Kahoot

Spm. 4: Hva legger vi i begrepet "deltakelse" i forbindelse med ER-modellering?

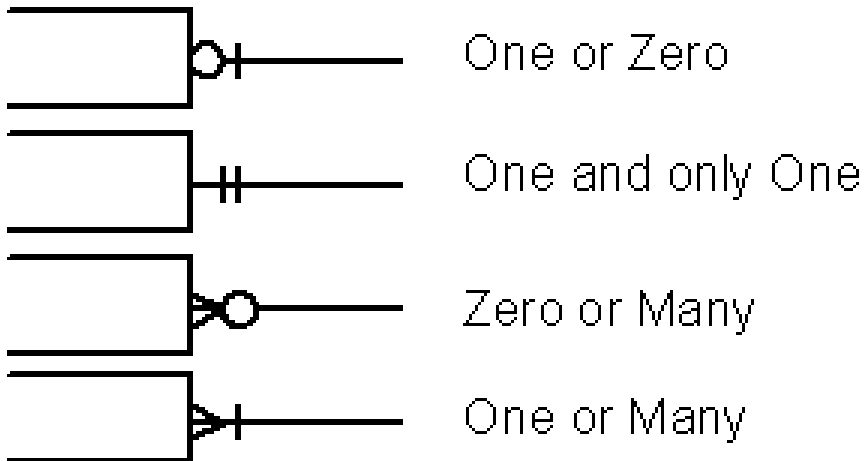
- A) "Om kunden deltar i utforming av modellen vår." (2 svar)
- B) "Om en entitet er med i modellen vår." (30 svar)
- C) "Det minste antall koplinger en entitet i en relasjon kan ha" (38 svar)
- D) "Det største antall koplinger en entitet i en relasjon kan ha" (25 svar)

*"Deltakelse" er et begrep for minste antall koplinger! (Det største antall koplinger heter "**kardinalitet**", minste + største til sammen heter "**multiplicity**".)*



Kråkefot (og UML), repetisjon

Summary of Crow's Foot Notation



UML Notation

0..1

1..1 (*evt. '1'*)

0..*

1..*

Kilde: tdan.com/crows-feet-are-best/7474

Konseptuell og logisk modell, repetisjon

- **Konseptuell (begrepsmessig) modell** beskrives i kapittel 7 i læreboka.
 - Dette er den enkleste modellen.
 - Den angir ikke fremmednøkler.
 - Når vi benytter enkel kråkefot eller UML notasjon angir den heller ikke attributter.
 - I LearnER begynner vi eventuelt med konseptuell modell før vi går til den neste:
- **Logisk modell** beskrives i kapittel 8.1 i læreboka.
 - Dette er en mer utfyllende modell.
 - Den har alltid med nøkler (PK og FK).
 - Den har alltid med attributter.
 - I LearnER er logisk modell alltid med.

ER diagram med Kråkefot og UML, repetisjon

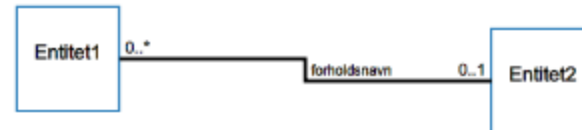
- Vi kan vise modeller av databasen som [ER diagram](#).
 - Dette kan gjøres på litt forskjellige måter. Her vist med noen LearnER eksempler:

- **Konseptuell modell**

- Enkel Kråkefot

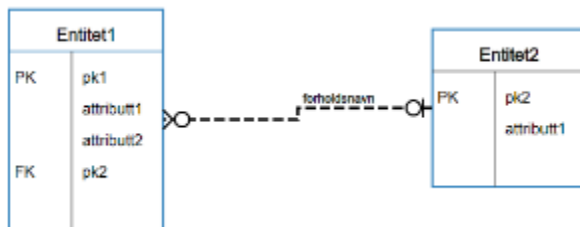


- UML

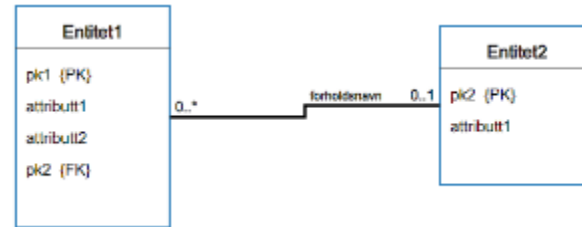


- **Logisk modell:**

- Kråkefot

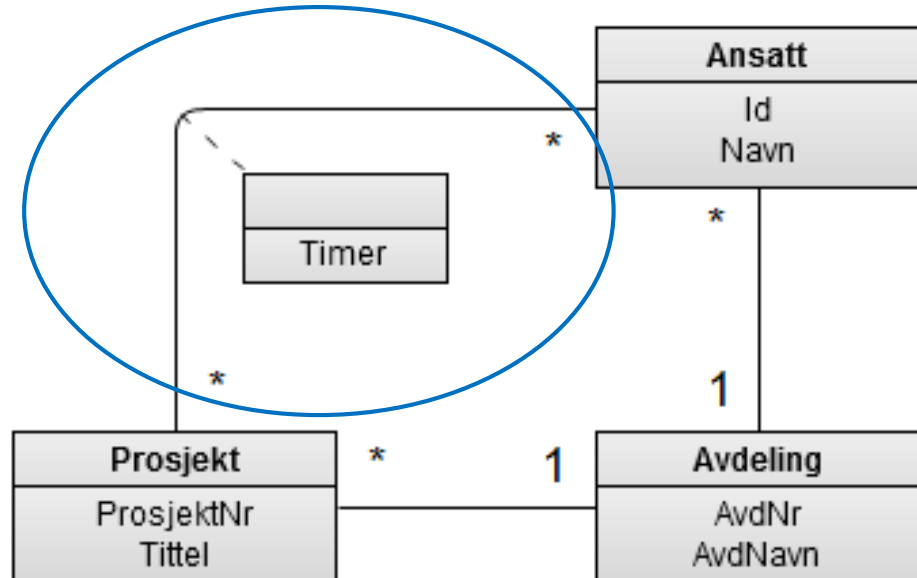


- UML



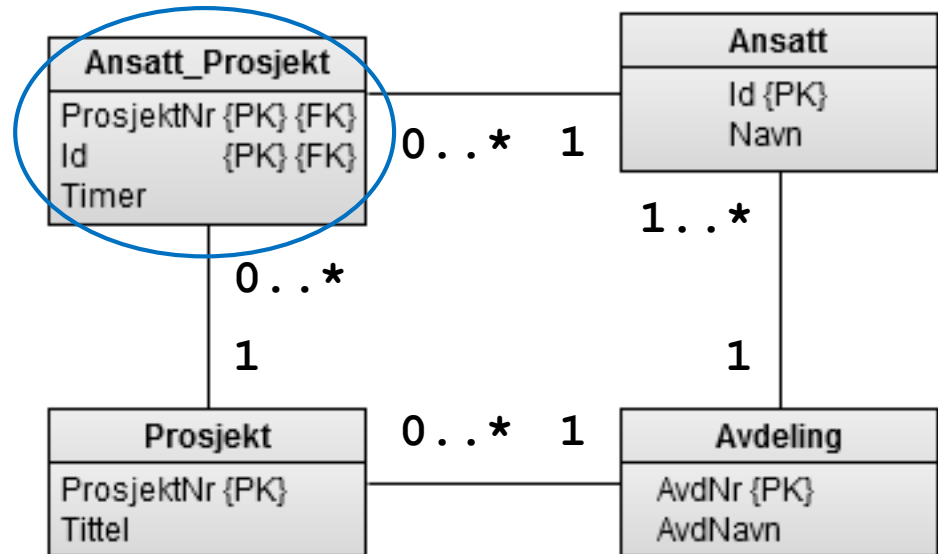
Mange-til-mange (M:M) relasjon, repetisjon

- Fra forrige ukes eksempel, relasjonen prosjekt & ansatt:
 - Et prosjekt kan bemannes av mange (*) ansatte.
 - En ansatt kan jobbe parallelt på mange (*) prosjekter.
- **Ansatt – Prosjekt** har en **mange-til-mange relasjon!**
- I tillegg trenger vi å vite "tid per ansatt per prosjekt".
 - Får attributtet "**Timer**" og plasseres på relasjonen prosjekt & ansatt!



Koblingsentiteter, repetisjon

- Når vi har et **mange-til-mange-forhold** introduserer vi en **koblingsentitet**, og **eventuelle relasjonsattributter** legges i denne entiteten.
 - Kapittel 7.5 i boka forklarer dette godt.
- For eksempelet vårt: Vi fjerner (realiserer) mange-til-mange relasjonen mellom Ansatt og Prosjekt ved å **lage** koblingsentiteten **Ansatt_Prosjekt**.
 - Koblingsentiteten skal inneholde **PK fra de to entitetene** den kopler sammen, samt **eventuelle relasjonsattributter**.
 - (Timer i vårt tilfelle.)




ER-skjema til database, repetisjon

- ER til database, punkt 1/6:
 - Markér PK for alle entiteter.
 - Fjern mange-til-mange relasjoner ved å lage nye koplingsentiteter.
 - Koplingsentitetene skal inneholde PK fra de to entitetene de kopler sammen, samt eventuelle relasjonsattributter.
- ER til database, punkt 2/6:
 - Hver entitet blir en tabell med samme navn.
 - Attributtene til en entitet blir kolonner i tabellen.
 - I UML er det vanlig å angi primary key ved å sette {PK} bak navnet.
 - (Underline er en annen vanlig primary key notasjon.)

ER-skjema til database, repetisjon – forts.

- ER til database, punkt 3/6:
 - Én til mange relasjoner realiseres ved at PK på én-siden kopieres til tabellen på mange-siden, og blir fremmednøkkel (FK) der.
 - UML notasjon, foreign key: {FK}.
 - (Annen vanlig notasjon: stiplet linje.)
- ER til database, punkt 4/6:
 - Én til én relasjoner kan realiseres på flere måter.
 - Sjekk kapittel 8.1.5 i boka.
- ER til database, punkt 5/6 & 6/6:
 - Normalisering og denormalisering av databasen.



Dette (punkt 5 og 6)
kommer i dag (nå!)
og i morgen!

NORMALISERING

Hensikten med normalisering

- Hensikten med **normalisering** er å finne de mest gunstige relasjoner for en gitt database.
- **Kriterier vi etterstreber** ved normalisering:
 - Skape **lavest antall nødvendige attributter** (kolonner) for å ta vare på de relevante data.
 - Lagre attributter som er **logisk knyttet sammen** innen en tabell.
 - **Minimal dobbellagring** (redundans), så attributter kun lagres ett sted.
Unntak for Foreign keys: Disse trenger nødvendigvis å lagres flere steder, da de skal koble flere entiteter (tabeller) sammen.
- Fra læreboka:
 - «Normalisering går ut på å gjenta normaliseringssteg helt til all redundans er fjernet»

Problemer med dobbellagring

- Tabellen `ansatt_filial` (med primærnøkkel `anr`) inneholder **dobbellagring** (**redundans**) i flere kolonner.
 - FNR, FADRESSE og FPST: filialnummer, filialadresse og filialpostnummer.

ANR	NAVN	AADRESSE	APST	STILLING	LOENN	FNR	FADRESSE	FPST
3	Jon Hvit	Bruveien 7	4000	Leder	30000	1	Utleieveien 15	4000
4	Anne Strand	Strandgaten	2000	Megler	12000	1	Utleieveien 15	4000
20	Olav Gautesen	Galmannsveien 4	3000	Smegler	26000	1	Utleieveien 15	4000
5	David Opalsen	Gulerleveien 43	2000	Sekretær	18000	1	Utleieveien 15	4000
2	Marie Hovland	Strilegaten 8	5000	Leder	13000	2	Smuglerstien 67	5000
23	Ole Ås	Mor Åseveien 56	4000	Megler	17000	2	Smuglerstien 67	5000
21	Per Pollesen	Podlestadvveien 5	5000	Sekretær	15000	2	Smuglerstien 67	5000
7	Karl Hansen	Olavsgt 7	2000	Leder	25000	3	Snusveien 7	7000

- *Hvilke problemer gir denne tabellen oss, med tanke på innlegging, endring og sletting?*

INSERT-problemer

- Legge inn ny ansatt, problemer:
 - Må samtidig legge inn allerede registrerte data på nytt for filial. (Filialnummer, adresse og postnummer).
 - Ved feiltasting får man problemer med at adressen blir forskjellig for samme filial (får en inkonsistent database).

ANR	NAVN	AADRESSE	APST	STILLING	LOENN	FNR	FADRESSE	FPST
3	Jon Hvit	Bruveien 7	4000	Leder	30000	1	Utleieveien 15	4000
4	Anne Strand	Strandgaten	2000	Megler	12000	1	Utleieveien 15	4000

- Legge inn ny filial uten ansatte, problemer:
 - Må sette inn `NULL` på feltene som gjelder ansatte, for eksempel `anr`.
 - `anr` er primærnøkkel og `NULL` er ikke tillatt her.
 - Må dermed legge inn en dummyansatt.

DELETE-problemer

- Slette siste ansatte for en filial, problemer:
 - Mister samtidig alle opplysninger om filialen.

ANR	NAVN	AADRESSE	APST	STILLING	LOENN	FNR	FADRESSE	FPST
---	-----	-----	----	-----	-----	---	-----	----
3	Jon Hvit	Bruveien 7	4000	Leder	30000	1	Utleieveien 15	4000

UPDATE-problemer

- Endring av postnummer eller adresse til en filial, problemer:
 - Må gjøre de samme endringene i alle rader til alle ansatte for denne filialen.
 - Ved feiltasting får man problemer med at adressen blir forskjellig på samme filial (får en inkonsistent database).

ANR	NAVN	AADRESSE	APST	STILLING	LOENN	FNR	FADRESSE	FPST
3	Jon Hvit	Bruveien 7	4000	Leder	30000	1	Utleieveien 15	4000
4	Anne Strand	Strandgaten	2000	Megler	12000	1	Utleieveien 15	4000

Problemer - oppsummert

- Redundans i tabellene våre skaper problemer.
- Samlebegrep for utfordringene knyttet til bruk av en tabeller med redundans kalles *oppdateringsanomalier*.

Bedre tabellstruktur

- Løsning: Splitte ansatt_filial i to tabeller: ansatt og filial:

ANR	NAVN	ADRESSE	POST	STILLING	LOENN	FNR
---	-----	-----	----	-----	-----	---
3	Jon Hvit	Bruveien 7	4000	Leder	30000	1
4	Anne Strand	Strandgaten	2000	Megler	12000	1
20	Olav Gautesen	Galmannsveien 4	3000	Megler	26000	1
5	David Opalsen	Gulerleveien 43	2000	Sekretær	18000	1
2	Marie Hovland	Strilegaten 8	5000	Leder	13000	2
23	Ole Ås	Mor Åseveien 56	4000	Megler	17000	2
21	Per Pollesen	Podlestadveien 5	5000	Sekretær	15000	2
7	Karl Hansen	Olavsgt 7	2000	Leder	25000	3

FNR	ADRESSE	POST
---	-----	----
1	Utleieveien 15	4000
2	Smuglerstien 67	5000
3	Snusveien 7	7000

*Får vi nå problemer med
innlegging/endring/sletting?*

Bedre tabellstruktur, INSERT

- Legge inn ny ansatt:
 - Legger kun inn personopplysninger pluss korrekt filialnummer.
 - Filialadressen kan ikke bli forskjellig for en og samme filial, da filialopplysningene kun lagres i én rad i databasen. (Har en konsistent database).
- Legge inn ny filial uten ansatte:
 - Helt ok at det ikke finnes opplysninger om ansatte i filialtabellen.

Bedre struktur, DELETE/UPDATE

- Slette siste ansatte for en filial:
 - Filialen eksisterer fortsatt i filialtabellen.
- Endring av postnummer eller adresse til en filial:
 - Kun nødvendig å endre opplysningene for én rad i databasen.
(Har en konsistent database.)

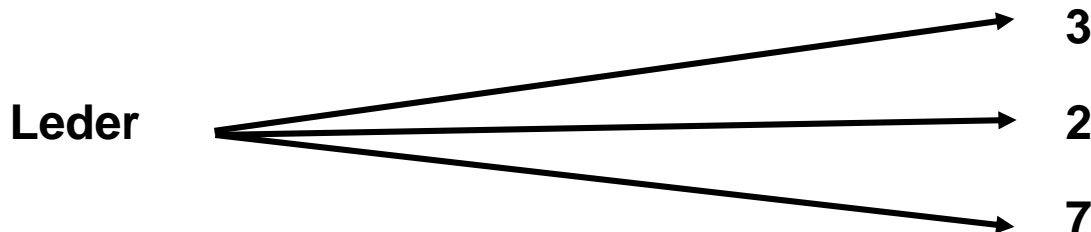
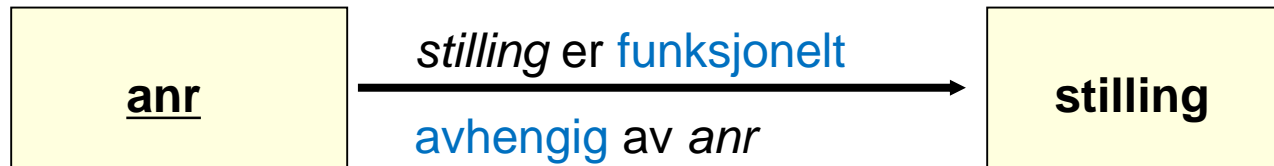
Normaliseringsprosessen

- Normalisering vil som regel si å **splitte databasen opp i flere tabeller**.
- Når en tabell skal splittes til flere, er det viktig at:
 - Vi får en tapsfri splitt: Alle rader i den opprinnelige tabellen må kunne dannes ved å koble sammen de nye tabellene. (Boka: Tapsfri dekomposisjon).
 - Vi bevarer avhengigheter: begrensninger i den originale tabellen bevares ved at vi lager tilsvarende begrensninger i de nye.
- Vi innfører begrepet "**funksjonell avhengighet**":
 - Merk at vi må forstå betydningen (semantikken) til attributtene for at dette skal ha mening.
 - Med andre ord: Vi må "forstå databasen" før vi kan normalisere den.

Definisjon, funksjonell avhengighet

- Hvis A og B er attributter i en relasjon R, så er B funksjonelt avhengig av A (kan skrives: $A \rightarrow B$) hvis det for hver verdi av A kan identifiseres kun én verdi av B, og hvis A gjentas får vi også samme verdi av B.
 - (A og B kan bestå av flere attributter).
- Altså: Har du A, så vet du B.
- Eksempel: Man kan si at 'stilling' er funksjonelt avhengig av 'ansattnummer' ('ansattnummer' \rightarrow 'stilling'). Da skal dette ansattnummeret alltid vise til samme stilling.
 - Det motsatte er ikke nødvendigvis tilfelle: Det er ikke sagt at det nødvendigvis er bare ett ansattnummer knyttet til en type stilling. Det kan være flere ledere (med hvert sitt ansattnummer, altså forskjellige personer) som leder hver sin avdeling, osv.
- *Litt vanskelige begreper, men kanskje lettere om vi ser et visuelt eksempel?*
 - *Se neste slide for eksempel for den tenkte databasen vi alt har sett på. :-)*

Funksjonell avhengighet



Determinant: viktig begrep

- Determinant:
 - En kolonne (eller fler) i en tabell som er bestemmende for minst en annen kolonne i tabellen.
- Generelt:
 - Ved $A \rightarrow B$ (B funksjonelt avhengig av A) er **A determinant for B**.
(Men B er ikke determinant for A.)
- Eksempelet på forrige side:
 - `anr` er determinant for `stilling`.
 - Men `stilling` er ikke determinant for `anr`.

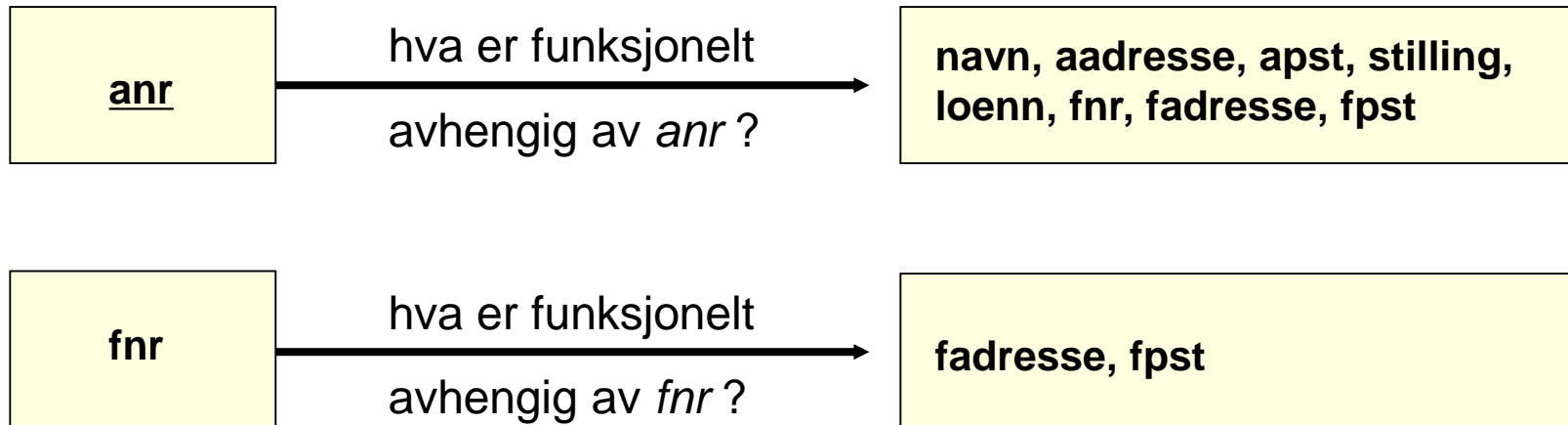
Oppgave

- Hvilke funksjonelle avhengigheter finner du i tabellen nedenfor?

ANR	NAVN	AADRESSE	APST	STILLING	LOENN	FNR	FADRESSE	FPST
3	Jon Hvit	Bruveien 7	4000	Leder	30000	1	Utleieveien 15	4000
4	Anne Strand	Strandgaten	2000	Megler	12000	1	Utleieveien 15	4000
20	Olav Gautesen	Galmannsveien 4	3000	Smegler	26000	1	Utleieveien 15	4000
5	David Opalsen	Gulerleveien 43	2000	Sekretær	18000	1	Utleieveien 15	4000
2	Marie Hovland	Strilegaten 8	5000	Leder	13000	2	Smuglerstien 67	5000
23	Ole Ås	Mor Åseveien 56	4000	Megler	17000	2	Smuglerstien 67	5000
21	Per Pollesen	Podlestadveien 5	5000	Sekretær	15000	2	Smuglerstien 67	5000
7	Karl Hansen	Olavsgt 7	2000	Leder	25000	3	Snusveien 7	7000

Funksjonell avhengighet

ANSATT_FILIAL (anr, navn, aadresse, apst, stilling, loenn, fnr, fadresse, fpst)



Full funksjonell avhengighet

- B er **fullstendig funksjonelt avhengig** av A hvis B er funksjonelt avhengig av A, men ingen delmengde ("subset") av A.
 - Du kan ikke ta bort en kolonne fra A uten at B mister funksjonell avhengighet til A.

Transitiv avhengighet

- Transitiv avhengighet er indirekte funksjonell avhengighet.
- A, B og C er attributter i en relasjon.
- $A \rightarrow B$ og $B \rightarrow C$.
- C er transitivt avhengig (indirekte funksjonelt avhengig) av A, via B.

Oppgave

- Hvilke transitive avhengigheter finner du i tabellen nedenfor?

ANR	NAVN	AADRESSE	APST	STILLING	LOENN	FNR	FADRESSE	FPST
3	Jon Hvit	Bruveien 7	4000	Leder	30000	1	Utleieveien 15	4000
4	Anne Strand	Strandgaten	2000	Megler	12000	1	Utleieveien 15	4000
20	Olav Gautesen	Galmannsveien 4	3000	Smegler	26000	1	Utleieveien 15	4000
5	David Opalsen	Gulerleveien 43	2000	Sekretær	18000	1	Utleieveien 15	4000
2	Marie Hovland	Strilegaten 8	5000	Leder	13000	2	Smuglerstien 67	5000
23	Ole Ås	Mor Åseveien 56	4000	Megler	17000	2	Smuglerstien 67	5000
21	Per Pollesen	Podlestadveien 5	5000	Sekretær	15000	2	Smuglerstien 67	5000
7	Karl Hansen	Olavsgt 7	2000	Leder	25000	3	Snusveien 7	7000

Transitiv anhengighet

ANR -----> FNR

FNR -> FADRESSE, FPST

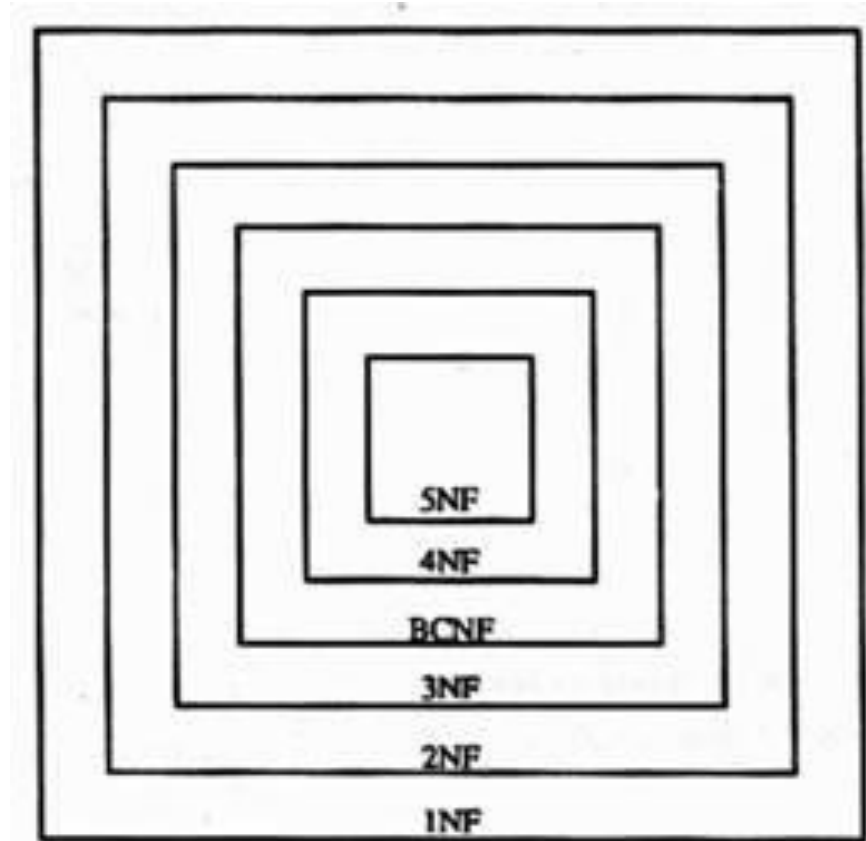
ANR -----> FADRESSE, FPST

ANR	NAVN	AADRESSE	APST	STILLING	LOENN	FNR	FADRESSE	FPST
3	Jon Hvit	Bruveien 7	4000	Leder	30000	1	Utleieveien 15	4000
4	Anne Strand	Strandgaten	2000	Megler	12000	1	Utleieveien 15	4000
20	Olav Gautesen	Galmannsveien 4	3000	Smegler	26000	1	Utleieveien 15	4000
5	David Opalsen	Gulerleveien 43	2000	Sekretær	18000	1	Utleieveien 15	4000
2	Marie Hovland	Strilegaten 8	5000	Leder	13000	2	Smuglerstien 67	5000
23	Ole Ås	Mor Åseveien 56	4000	Megler	17000	2	Smuglerstien 67	5000
21	Per Pollesen	Podlestadveien 5	5000	Sekretær	15000	2	Smuglerstien 67	5000
7	Karl Hansen	Olavsgt 7	2000	Leder	25000	3	Snusveien 7	7000

FADRESSE og FPST er **transitivt avhengig** av ANR, **via** FNR.

Normaliseringsprosessen

- I dag ser vi på 1NF til 3NF.
(Første til tredje normalform.)
- Neste gang ser vi på BCNF.
(Boyce-Codd normalform.)
- 4NF og 5NF (fjerde og femte normalform) er ikke relevant i dette emnet.



Unnormalisert form (UNF)

- Vi registrerer leieforhold som omfatter leietaker (lnr, lnavn), prospekt (pnr, padresse, fradato, tildato, leie) og eier (enr, enavn).
 - En tabell er unormalisert dersom den inneholder celler med mer enn ett dataelement.

<i>lnr</i>	<i>lnavn</i>	<i>pnr</i>	<i>padresse</i>		<i>fradato</i>	<i>tildato</i>	<i>leie</i>	<i>enr</i>	<i>enavn</i>
5	Hansen	6	Aveien	1	01-JUL-94	01-SEP-96	3500	4	Olsen
?	?	1	Bveien	8	01-SEP-96	01-JAN-98	4400	12	Larsen
9	Persen	6	Aveien	1	01-SEP-96	01-SEP-97	3500	4	Olsen
?	?	2	Dveien	6	01-SEP-97	01-SEP-98	5500	6	Alfsen
?	?	5	Eveien	3	01-SEP-98	null	5000	6	Alfsen

Første normalform (1NF)

- En tabell er på **1NF** dersom:
 - Cellene inneholder bare ett dataelement.
 - (NB: "Ett dataelement" kan godt være flere ord, f.eks. i en varchar celle.)
- Vi kan fjerne cellene med flere dataelementer ved å lage én rad per element:
 - Da inneholder hvert felt kun en verdi.
 - Primærnøkkelen blir her sammensatt av (lnr, pnr).

Leietaker_Leiekontrakt_Prospekt_Eier

lnr	lnavn	pnr	padresse	fradato	tildato	leie	enr	enavn
---	-----	---	-----	-----	-----	----	---	-----
5	Hansen	6	Aveien 1	01-JUL-94	01-SEP-96	3500	4	Olsen
5	Hansen	1	Bveien 8	01-SEP-96	01-JAN-98	4400	12	Larsen
9	Persen	6	Aveien 1	01-SEP-96	01-SEP-97	3500	4	Olsen
9	Persen	2	Dveien 6	01-SEP-97	01-SEP-98	5500	6	Alfsen
9	Persen	5	Eveien 3	01-SEP-98	null	5000	6	Alfsen

Andre normalform (2NF)

- En tabell er på **2NF** dersom:
 - Den er på **1NF**.
 - Og **ingen delmengde ("subset") av primærnøkkelen er determinant for en eller flere andre kolonner** (ingen "delvis avhengighet" til PK).
 - *Merk: Om en tabell er på 1NF og PK består av én kolonne, er tabellen også på 2NF!*

Ok: avhengighet til hele PK.

lnr	lnavn	pnr	padresse	fradato	tildato	leie	enr	enavn
5	Hansen	6	Aveien 1	01-JUL-94	01-SEP-96	3500	4	Olsen
5	Hansen	1	Bveien 8	01-SEP-96	01-JAN-98	4400	12	Larsen
9	Persen	6	Aveien 1	01-SEP-96	01-SEP-97	3500	4	Olsen
9	Persen	2	Dveien 6	01-SEP-97	01-SEP-98	5500	6	Alfsen
9	Persen	5	Eveien 3	01-SEP-98	null	5000	6	Alfsen

Ikke ok: delvis avhengighet til PK.

Andre normalform (2NF) – forts.

- Primærnøkkelen er sammensatt.
 - Vi skiller ut kolonnene som er avhengige av deler av primærnøkkelen og plasserer disse i nye tabeller:

Leietaker

lnr	lnavn
---	-----
5	Hansen
9	Persen

Leiekontrakt

lnr	pnr	fradato	tildato
---	---	-----	-----
5	6	01-JUL-94	01-SEP-96
5	1	01-SEP-96	01-JAN-98
9	6	01-SEP-96	01-SEP-97
9	2	01-SEP-97	01-SEP-98
9	5	01-SEP-98	null

Prospekt_Eier

pnr	padresse	leie	enr	enavn
---	-----	-----	---	-----
6	Aveien	1 3500	4	Olsen
1	Bveien	8 4400	12	Larsen
2	Dveien	6 5500	6	Alfsen
5	Eveien	3 5000	6	Alfsen

Tredje normalform (3NF)

- En tabell er på **3NF** dersom:
 - Den er på **2NF**.
 - Og **ingen ikke-primærnøkkel-attributt har en transitiv avhengighet til en primærnøkkel**. (Ingen $A \rightarrow B$ og $B \rightarrow C \Rightarrow A \rightarrow C$.)
- Alternativt kan vi si det slik: Alle ikke-primærnøkkel-attributter er kun funksjonelt avhengige av primærnøkkel.

pnr	adresse	leie	enr	enavn
6	Aveien	1 3500	4	Olsen
1	Bveien	8 4400	12	Larsen
2	Dveien	6 5500	6	Alfsen
5	Eveien	3 5000	6	Alfsen

Merk: Tabellene
Leietaker og
Leiekontrakt er
allerede på 3NF.

Tredje normalform (3NF) – forts.

- Vi fjerner transitivt avhengige attributter fra tabellen og plasserer de i en ny tabell.
- **Determinanten blir nøkkelen.**

Prospekt

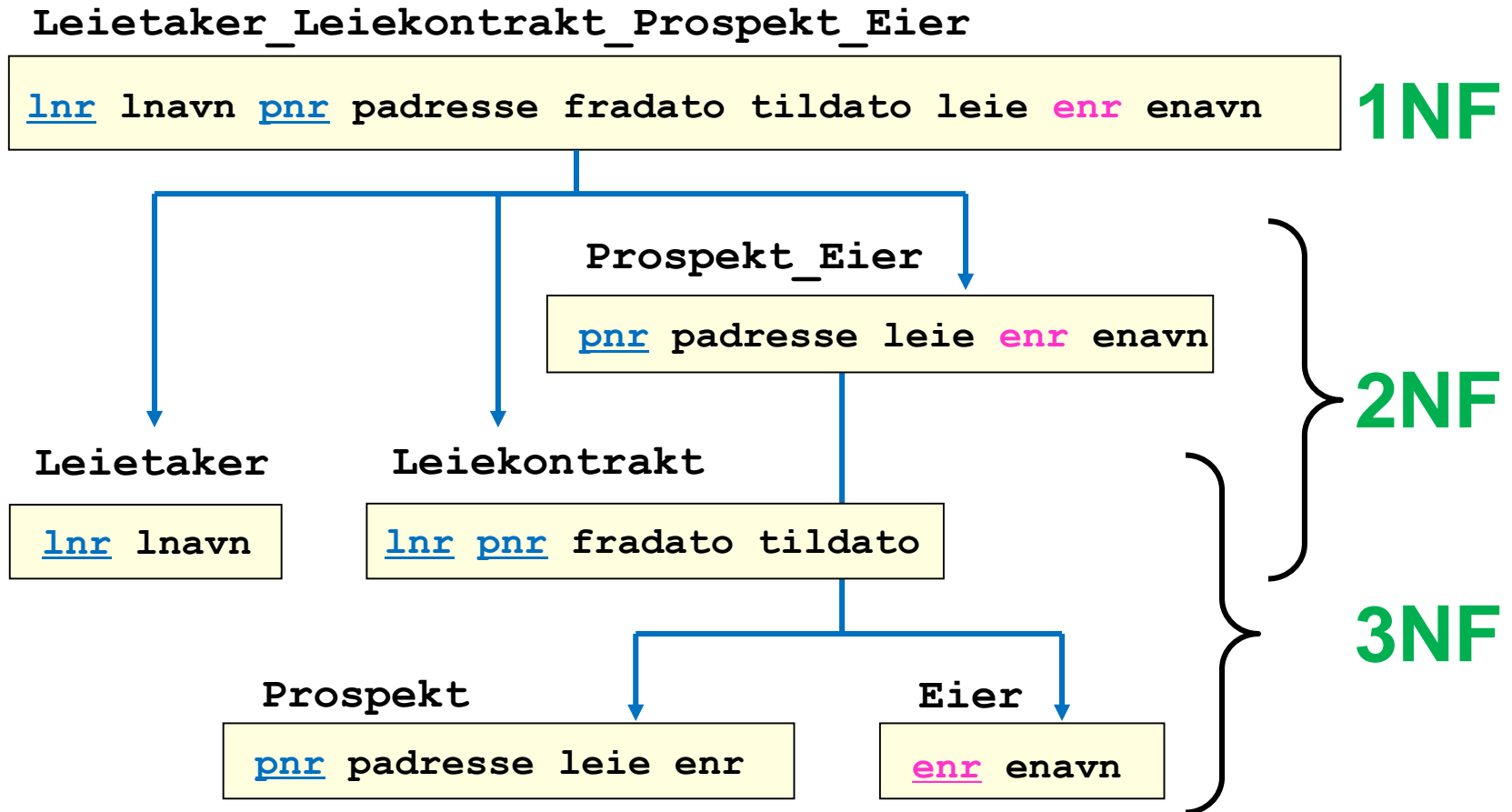
pnr	adresse	leie	enr
---	-----	----	---
6	Aveien 1	3500	4
1	Bveien 8	4400	12
2	Dveien 6	5500	6
5	Eveien 3	5000	6

Eier

enr	enavn
---	-----
4	Olsen
12	Larsen
6	Alfsen



Oppsummering: fra 1NF - 3NF



1NF til 3NF, oppsummert

- 1NF:
 - Cellene inneholder bare *ett* dataelement.
- 2NF, dersom den er på 1NF og:
 - Ingen avhengighet til *deler av primærnøkkelen*. (Ingen delmengde av PK er determinant for kolonner som ikke er del av PK.)
- 3NF, dersom den er på 2NF og:
 - Ingen kolonne utenfor primærnøkkelen har en *transitiv* (indirekte) avhengighet til *primærnøkkelen*. ($A \rightarrow B$ og $B \rightarrow C \Rightarrow A \rightarrow C$.)

Er NF, determinant, m.m. fortsatt litt hårete?

- Det kan godt være at normalisering fortsatt er et vanskelig tema!?
- I så fall – forslag:
 - Ta en titt på en (eller fler) av studentvideoene fra i fjor om normalisering!?
(Du finner de nederst i lista med eksterne ressurser for DB1100 på Canvas.)
 - Og/eller google normaliseringsvideoer på nett.
 - Og/eller les de relevante avsnittene i læreboka en gang til.
 - Og/eller spør veiledere og foreleser i øvingstimene.
- Etter dette: Gjør øvingsoppgavene!
 - (Husk: Løsningsforslag tilgjengeliggjøres når det er 1 øvingstime igjen.)
- Etter dette: Delta på forelesningen! :-)

Videre arbeid i dag (og kort om neste gang)

- Nå:
 - Lese kap. 8.2.1 - 8.2.4 i pensumboka.
 - Se opp til 6 studentvideoer om normalisering (linket nederst på Canvas).
Finner du noen favoritter?!
 - Begynne på øvingsoppgavene.
- Etterpå:
 - Øving (flipped, 2 timer) 14:15 - 16:00, sjekk TimeEdit for rom.
Teori (flipped, 2 timer) 16:15 - 18:00, auditoriet.
- Neste gang: (i morgen!)
 - Boyce-Codd normalform (BCNF) og denormalisering.
 - Pensum til i morgen: Resten av kapittel 8.