



Økt 7 (av 12)

DB1102 Databaser

Per Lauvås / per.lauvas@kristiania.no

Yuan Lin / yuan.lin@kristiania.no

Dagens temaer

Dagens tema: [Modellering ER, del 1 \(av 2\)](#).

– Dagens pensum: (samme som neste uke!) [Læreboka, kapittel 7 & 8.1](#)

- Nytt innhold: Datamodellering med ER
- Modelleringsverktøyet LearnER

Zoom-øktene framover

Blir 15.15-17 (altså flyttet med én time).

Kahoot



Spørsmål 5

5 -Quiz Hva menes med uttrykket "Kardinalitet" i tabellsammenheng?

5 of 5 < >



Foredlingsgraden til innholdet

✗

19



Antall rader den inneholder

✓

24



Hvor mange kolonner tabellen er delt opp i

✗

16



Ingen av de andre alternativene

✗

2



No answer

✗

10

Spørsmål 8

8 -Quiz Verdien NULL kan forklares slik:

2 of 5 < >



NULL er det samme som 0

✗ ●

6



NULL er det samme som en blank ("space")

✗ █████

36



NULL er det samme som et linjeskift ("enter")

✗ |

2



Ingen av alternativene ovenfor

✓ █████

16



No answer

✗ —

11

Spørsmål 9

9 -Quiz Hva er relasjonsalgebra?

1 of 5



Bruken av matematiske operatorer (+, -, *) i SQL spørringer

✗

14



Forholdet mellom forskjellige tabeller i en database

✗

30



Det teoretiske språket som SQL er basert på

✓

12



Ingen av de andre alternativene

✗

3



No answer

✗

12

Spørsmål 10

10 -Quiz Hva menes med kartesisk produkt av R og S?

3 of 5 < >



Summen av antall rader R og kolonner S i en tabell

✗

21



Rader som forekommer både i mengdene R og S

✗

13



Mengden som kombinerer hver rad i R med hver rad i S

✓

19



Ingen av de andre alternativene

✗

4



No answer

✗

14

Kartesisk produkt (repetisjon)

- Kartesisk produkt har vi snakket om tidligere.
 - Se også læreboka, side 97 og 155.
- **Kartesisk produkt** operasjonen gir som output mengden som kombinerer hver eneste rad ("tuple") i tabell R med hver eneste rad i tabell S.
 - Formel: $R \times S$
 - Huskeregel: "*Plusse kolonner, gange rader.*"

R	
a	
b	

S	
1	
2	

R x S	
a	1
a	2
b	1
b	2

Spørsmål i Zoom

«Må en Foreign Key referere til en Primary Key?»

Nei, men den må referere til noe unikt.

[Books online \(via StackOverflow\)](#):

A FOREIGN KEY constraint does not have to be linked only to a PRIMARY KEY constraint in another table; it can also be defined to reference the columns of a UNIQUE constraint in another table.

Og det er vanskelig å finne et godt eksempel der det er en god idé.

[Datadiversity](#):

- **Reference to a unique key other than the primary key.** There's no benefit to this. This is a poor development practice.

Modellering

- Hittil har vi sett på hvordan det er å jobbe mot eksisterende schema/databaser. Da er tabellene alt definert for oss. Vi har hatt fokus på:
 - `select` queries.
 - `insert into`, `update` og `delete from`.
 - `create table`, `alter table` og `drop table`.
- Av og til ønsker vi å være arkitektene bak databasen.
 - Da kommer `ER modellering` inn i bildet.

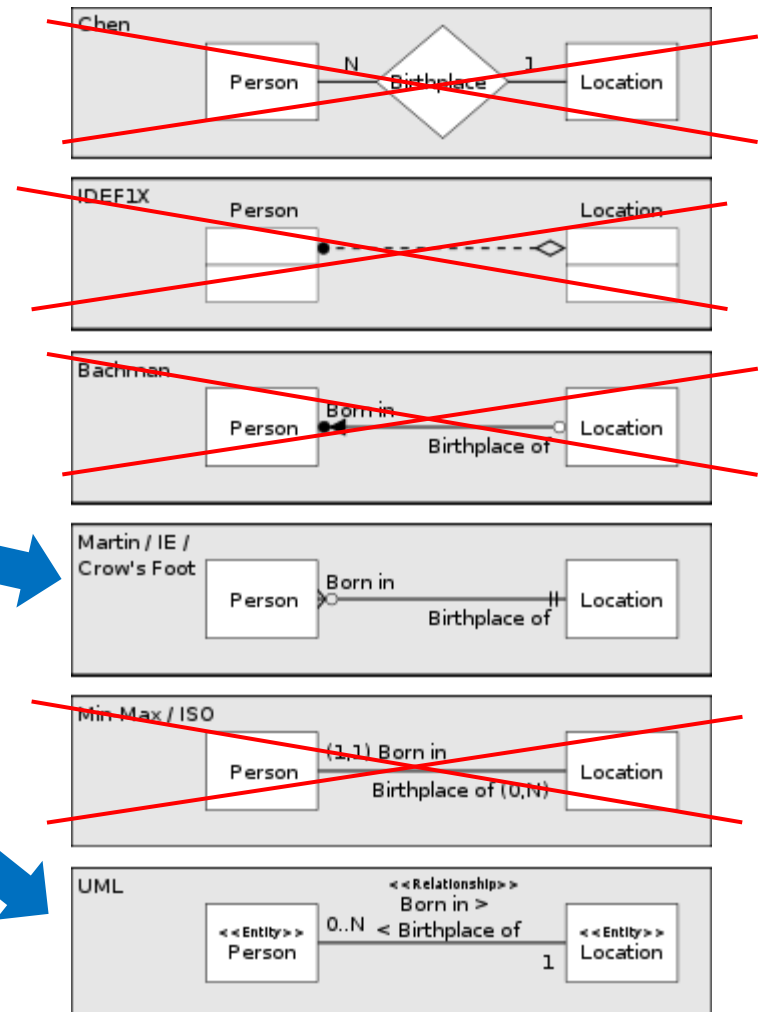
Modellering – forts.

- ER modellering:
 - ER = Entity Relationship
- NB: Relation vs. relationship - her må vi holde tunga rett i munnen:
 - "Relation" er et generelt relasjonsdatabaseuttrykk, og betyr tabell.
 - "Relationship" brukes i modelleringssammenheng, og betyr koplingen mellom to tabeller. Boka kaller dette *forhold* eller *relasjon*.
 - Jeg benytter mest begrepene *forhold* og *koplinger* når jeg oversetter relationship for å skille det fra annen DB terminologi. Men vi kommer ikke helt unna *relasjon*.
- Merk: MySQL Workbench bruker uttrykket EER modell (Extended Entity Relationship).

Om notasjoner

- Det finnes en rekke ER notasjoner.
- Du kan velge om du vil benytte kråkefot eller UML på eksamen.
 - Men bruk valget ditt konsekvent.
- MySQL Workbench og læreboka bruker kråkefot.
- På slides bruker jeg UML.
- Dette er de vanligste notasjonene.
 - Vi tar *ikke* for oss resten i DB1100.

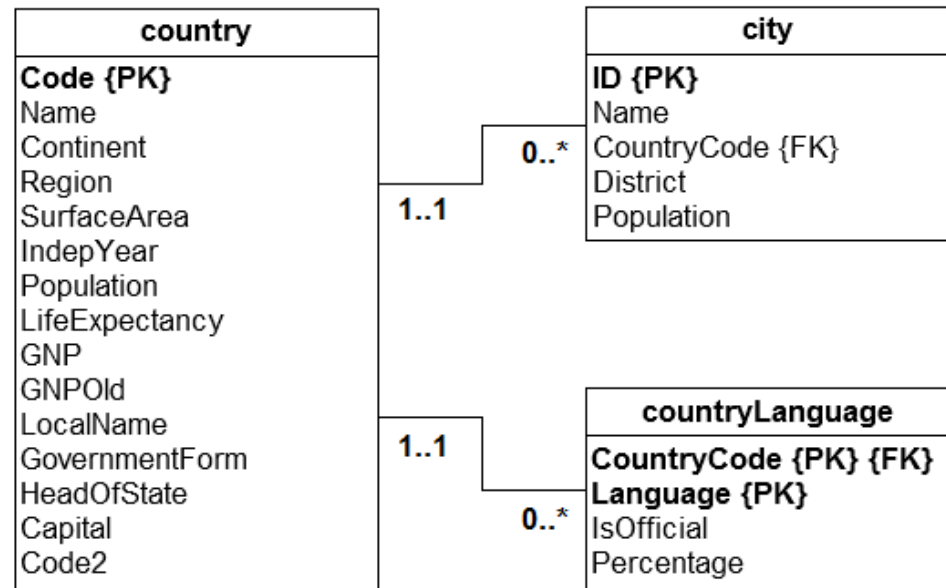
(Kilde: [Wikipedia](#))



ER diagram (modellering)

- Vi kan vise modellen av databasen *World* som et ER diagram:
 - Entity
 - Relationship

- Denne modellen er laget i Gliffy: www.gliffy.com/
- Lucidchart og diagrams.net gode alternativer.
- Velg selv, men velg gjerne noe som er gratis. :-)



Konseptuell (begrepsmessig) og logisk modell

- **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

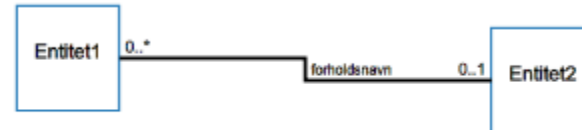
- Vi kan vise modeller av databasen som **ER diagram**.
 - Dette kan gjøres på litt forskjellige måter. Her vist med noen LearnER eksempler:

- **Konseptuell modell**

- Enkel Kråkefot

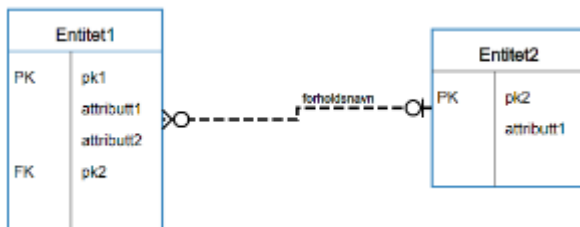


- UML

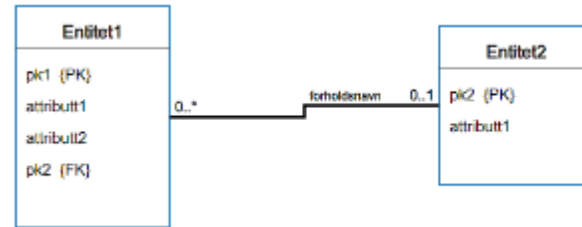


- **Logisk modell:**

- Kråkefot

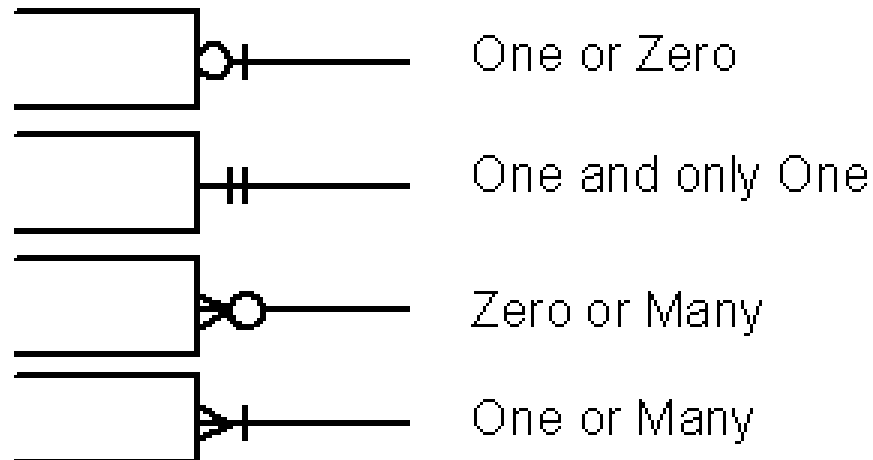


- UML



Kråkefot

Summary of Crow's Foot Notation



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

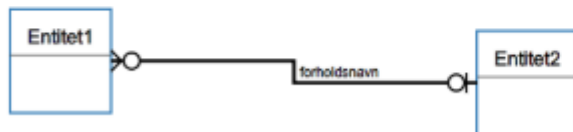
Identifiserende og ikke-identifiserende forhold

- Når vi benytter **kråkefot** notasjon (*ikke* enkel kråkefot eller UML) kan vi i den **logiske modellen** tydeliggjøre svake entiteter og identifiserende relasjoner.
- **Svake entiteter:**
 - Eksistensen til forekomster av den svake entiteten avhenger av eksistensen til forekomsten av entiteten den er knyttet til.
 - Identifikatoren til en svak entitet er helt eller delvis arvet fra andre entiteter.
- **Identifiserende forhold:**
 - Relasjoner mot svake entiteter, som skal føre til arv av identifikator, kalles **identifiserende**, og tegnes som **heltrukne linjer**.
 - Alle andre forhold kalles **ikke-identifiserende** og tegnes som **stiplede linjer**.
- Eksempel fra læreboka: (*kap. 7.4*)
 - Saga Kino og salen Saga 2.

Konseptuell + enkel kråkefot | Logisk + kråkefot

En konseptuell modell med enkel kråkefot-notasjon:

- Det velges bare entiteter og forhold
- Attributter velges ikke
- Skiller ikke mellom identifiserende og ikke-identifiserende forhold



En logisk modell med kråkefot-notasjon:

- Velger primærnøkler
- Velger fremmednøkler
- Identifiserende forhold vises med heltrukken linje og ikke-identifiserende forhold vises med stiplet linje



- La oss se på dette i LearnER! Tar en oppgave i LearnER i neste video😊

Eksempel case: prosjektstyring

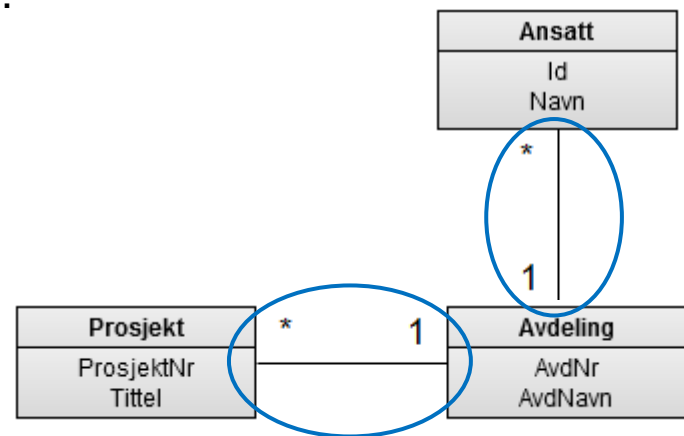
- Et firma ønsker å få oversikt over sine prosjekter. De har leid oss inn for å lage en databaseløsning som ordner dette.
- De ønsker spesifikt å få oversikt over følgende:
 - Hvilken avdeling (nummer, navn) eier hvert prosjekt?
 - Hvilke prosjekter (nummer, tittel) involverer hvilke ansatte?
 - Hvor mye tid benytter hver ansatt (id, navn) per prosjekt?
(NB: Denne siste er litt vanskelig å plassere på rett sted.)
- Modelleringsspørsmål:
 - Hvilke entiteter (kommende tabeller) må vi ha?
Og hvilke attributter skal plasseres i entitetene?
 - Hva er forholdene (relasjonene) mellom entitetene?
(Hvordan hører de sammen: En til en? En til mange? Mange til mange?)

Entiteter og attributter

- Ut fra spesifikasjonen kommer vi fram til et behov for følgende entiteter:
 - Avdeling
 - Prosjekt
 - Ansatt
- Videre trenger vi følgende attributter:
 - Avdeling: AvdNr, AvdNavn
 - Prosjekt: ProsjektNr, Tittel
 - Ansatt: Id, Navn

Forhold

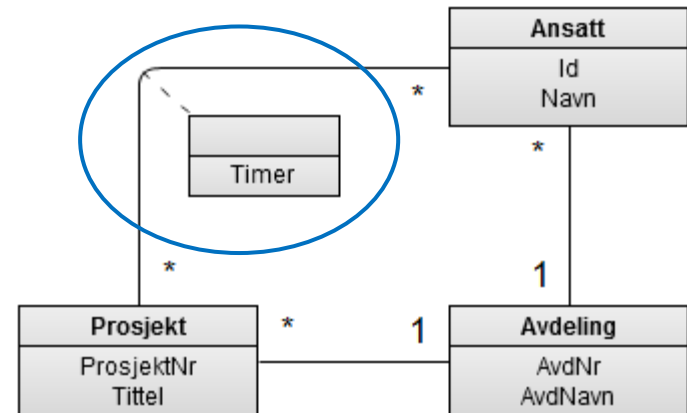
- Forholdene er ikke oppgitt i spesifikasjonen, men det er logisk(?) å anta forhold som oppgitt under.
 - (Vi bør tidlig i prosjektet få de bekreftet av kunden!)
- Forholdet (Relasjonen) avdeling & ansatt:
 - En avdeling kan ha mange (symbol: *) ansatte.
 - En ansatt tilhører én (symbol: 1) avdeling.
- Forholdet (Relasjonen) prosjekt & avdeling:
 - Et prosjekt tilhører én (symbol: 1) avdeling.
 - En avdeling kan ha mange (symbol: *) prosjekter.



Forhold – forts.

- Forholdet (Relasjonen) prosjekt & ansatt:

- Et prosjekt kan bemannes av mange (*) ansatte.
- En ansatt kan jobbe parallelt på mange (*) prosjekter.



- Modellen begynner å falle på plass! :-)

- Men vi har én attributt igjen:

- Vi trenger å vite ”tid per ansatt per prosjekt”.
- Men putte denne hvor?
- På forholdet prosjekt & ansatt!

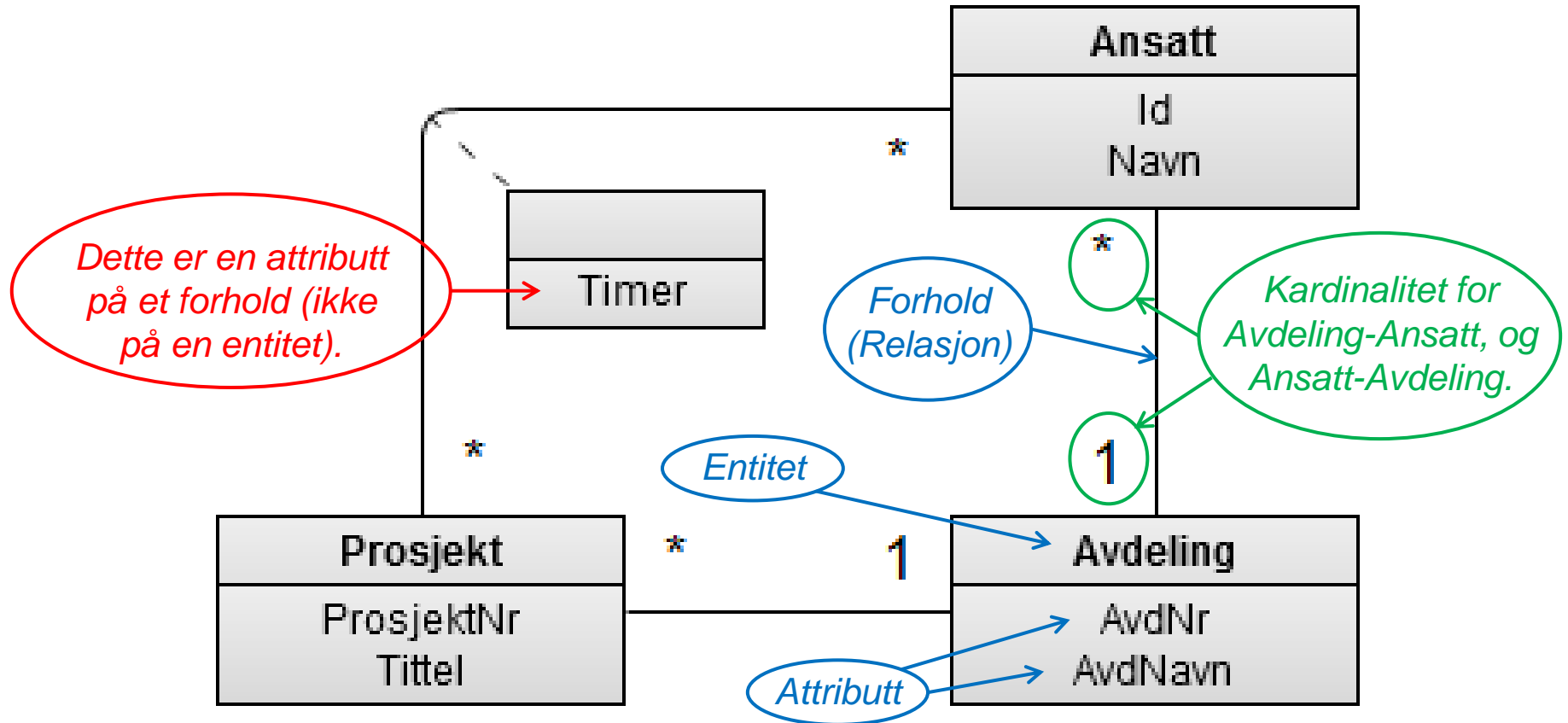
Koblingsentiteter

- Vi la inn et attributt på forholdet mellom Ansatt og Prosjekt.
 - Forholdet er i dette tilfellet et mange-til-mange-forhold: En ansatt kan delta i mange prosjekter. Et prosjekt kan ha mange prosjektdeltakere (ansatte).
- Når vi har et **mange-til-mange-forhold** introduserer vi en **koblingsentitet**, og **attributtene legges i denne entiteten**.
- Vi kaller overgangen fra mange-til-mange-forhold til koblingsentitet for *entitisering*.
- Kapittel 7.4.5 i boka forklarer dette godt! :-)

Forhold, nytt uttrykk

- Kardinalitet:
 - Vi husker fra før av at vi har kardinalitet i en tabell, og at dette betyr det antall rader tabellen inneholder.
- På liknende måte har vi **kardinalitet ifbm. ER-modellering**:
 - Kardinaliteten angir **det største antall koplinger en entitet som deltar i et gitt forhold kan ha**.
 - Eksempel: En ansatt kan maks tilhøre 1 avdeling. Kardinaliteten til ansatt i "ansatt-avdeling forholdet" er derfor 1. (For avdeling i samme forhold er den '*'.)

ER-skjema, ord/uttrykk - UML



To nye nøkkeltyper

- Supernøkkel

- En eller flere kolonner som danner en unik identifikator for en rad i en tabell.
- NB: Kan godt være flere kolonner enn det minste, unike utvalget. (Alle kolonner i en tabell til sammen er dermed alltid en supernøkkel.)

- Kandidatnøkkel

- En supernøkkel som ikke kan reduseres til færre kolonner om den fortsatt skal være en unik identifikator.

- Repetisjon, fra før har vi lært:

- **Primærnøkkel**: (PK) De(n) kolonnen(e) vi velger å bruke som vår unike identifikator for en rad i en tabell.
- **Fremmednøkkel**: Kolonne(r) som viser til primærnøkkelen i en annen tabell.

Oppgave!

- Hvilke supernøkler og kandidatnøkler har vi her?

ProsjektNr (fra Prosjekt)	Id (fra Ansatt)	Timer
1001	1	12
1002	1	44
1002	2	20
1002	3	125
1003	2	25
1004	2	5
1004	3	10
1005	1	10
1006	1	20
1006	2	125

Økt 8

- Neste gang:
 - Modellering ER, del 2.
 - Pensum er samme som til idag: Kapittel 7 & 8.1
 - Mer om arbeidskravet 😊