



Økt 7 (av 12)

DB1100 Databaser

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

Dagens temaer

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

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

- Tilbakemelding SQL Saturday
- Status arbeidskrav
- Fra forrige økt: Resultater Kahoot
- Nytt innhold: Datamodellering med ER

Tilbakemelding på SQL Saturday, 1. september

- Hei, for en drøy måned siden (1. september) var det SQL Saturday på skolen.
- De som stod for arrangementet vil gjerne ha tilbakemeldinger fra studenter som deltok!
- Derfor: **Var du med på SQL Saturday på Fjerdingsgen den 1. september?**
 - Flott om du da vil sende en kort tilbakemelding til event@kristiania.no! :-)

Status arbeidskrav

- Alle dere som *ikke* leverte arbeidskrav:
 - Se bort fra denne sliden. :-)
- Frist for peer review av arbeidskrav #1 var i går.
 - Dere får en bekreftelse på levert peer review snart (i dag eller i morgen).
 - Dere vil få tilbakemelding (peer review) av deres egen video.
- Dere som har levert peer review:
 - Dere får snart (når jeg har fått sendt ut bekreftelser på peer review, over) utdelt arbeidskrav #2, inklusive info om innhold og frist for dette!

Resultat, forrige Kahoot

- Forrige gang slo dere fjorårets studenter hårfint!
 - De klarte 64,17 % riktig svar.
- Dere: **64,66 % riktig svar!**
 - Dere er helt rå! B-)

Game Over	
Show feedback	
MMJ	13,025
preb\$\$\$	12,687
Noldus	12,040
😊	11,797
Untzuntz	11,464

DB1100 - økt 6	
Played on	4 Oct 2018
Hosted by	Tomas_Sandnes
Played with	75 players
Played	12 of 12 questions

Overall Performance	
Total correct answers (%)	64,66%
Total incorrect answers (%)	35,34%
Average score (points)	6991,37 points

Play Again

New Game

Lærdom fra Kahoot

Spm. 7: Hvilket av følgende alternativer er ikke en gyldig datatype i MySQL?

- A) "varchar" (8 svar)
- B) "date" (23 svar)
- C) "int" (6 svar)
- D) "Ingen av de andre alternativene" (34 svar)

"date" er absolutt en gyldig datatype i MySQL! (En annen gyldig datatype er "datetime", kanskje noen dermed tenkte at "date" ikke var en gyldig type?)

For oversikt over datatyper, se f.eks.:

- <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>
- https://www.w3schools.com/sql/sql_datatypes.asp

Lærdom fra Kahoot, #2

Spm. 8: Verdien NULL kan forklares slik:

- A) NULL er det samme som 0 | 0 svar
- B) NULL er det samme som en blank ("space") | 37 svar
- C) NULL er det samme som et linjeskift ("enter") | 0 svar
- D) Ingen av alternativene ovenfor | 34 svar

Kanskje var det vanskelig formulert, å skjønne at med alternativ B 'blank ("space")' så mente jeg mellomromstegnet?

- *Uansett: NULL er en form for "ingenting"! Ikke 0, ikke mellomrom, ikke linjeskift. :-)*

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.

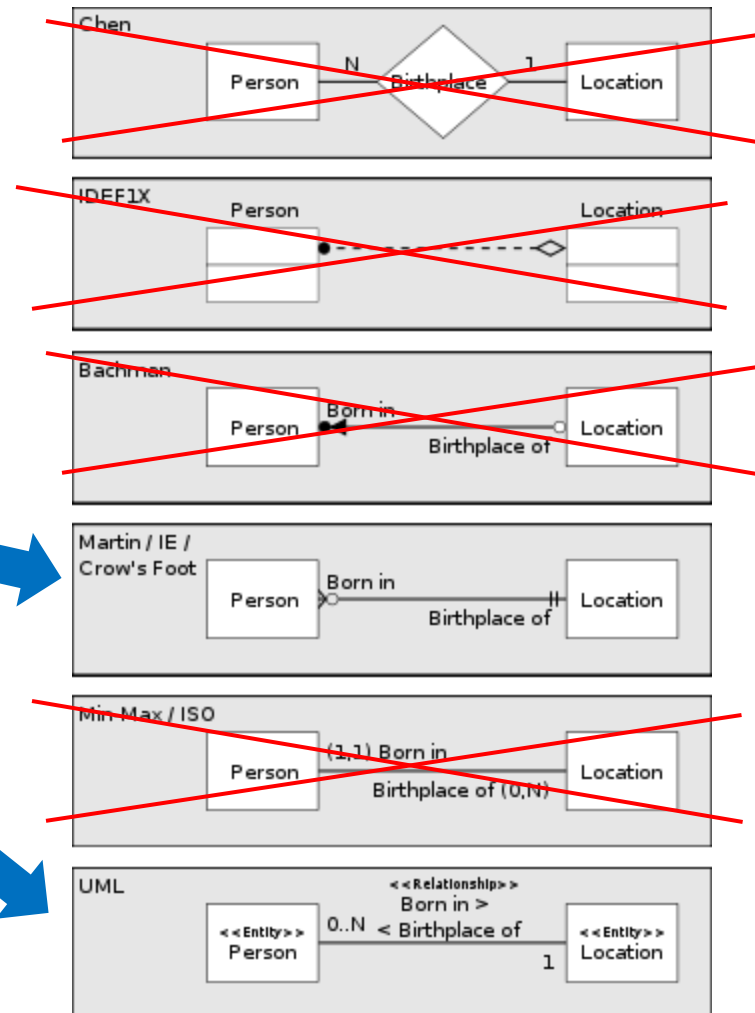
Modellering – forts.

- ER modellering:
 - ER = Entity Relationship
- NB: 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*.
- Merk: MySQL Workbench bruker uttrykket EER modell (Extended Entity Relationship).
 - Du finner info om EER-modell i læreboka, kap. 7.6.4.

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.
- LearnER, MySQL Workbench og læreboka bruker kråkefot.
- På mine slides bruker jeg UML.
- Dette er de vanligste notasjonene.
 - Vi tar *ikke* for oss resten i DB1100.

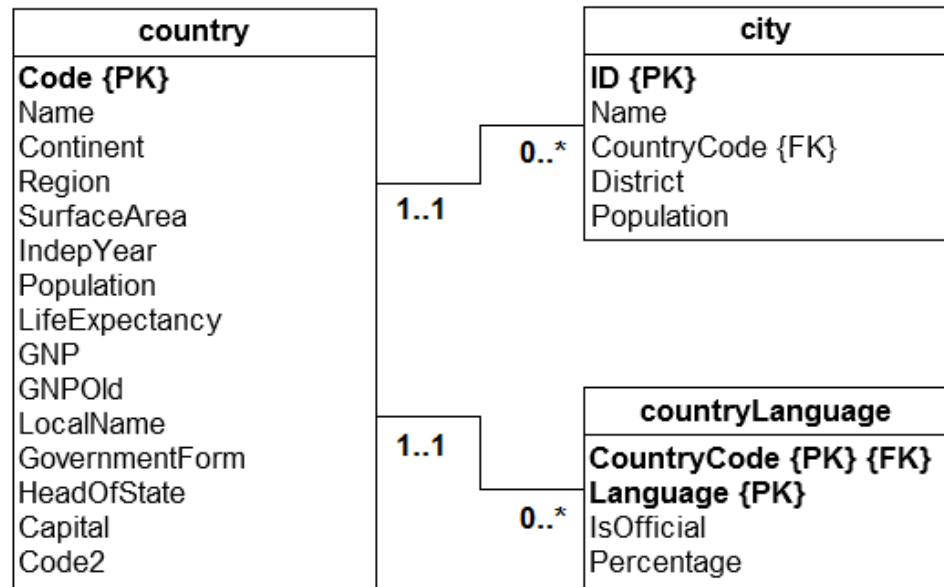
(Kilde: [Wikipedia](https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model))



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 [Draw](#) er gode alternativer.
- Velg selv, men velg gjerne noe som er gratis. :-)



Entitet, Relasjon (forhold) ogAttributt

- **Entitetstype**: En gruppe objekter med samme attributter der alle objekter har en selvstendig eksistens.
 - **Entitetsforekomst**: Et unikt identifiserbart objekt av en entitetstype.
- **Relasjonstype**: En mengde meningsfulle assosiasjoner mellom entitetstyper.
 - **Relasjonsforekomst**: En unik identifiserbar assosiasjon som inkluderer én forekomst fra hver representerte entitetstype.
- **Attributt**: En egenskap til en entitets- eller relasjonstype.

Begrepsmessig (konseptuell) og logisk modell

- Begrepsmessig (konseptuell) modell omtales i kapittel 7 i læreboka.
 - Dette er den enkleste modellen.
 - Når vi benytter LearnER (mer info om dette verktøyet i dagens øvingsoppgaver) begynner vi evt. med konseptuell fase før vi går til den neste:
- Logisk modell beskrives i kapittel 8.1 i læreboka.
 - I LearnER er alltid logisk fase med.

Svake entiteter og identifiserende forhold

- Når vi benytter kråkefot-notasjon kan vi 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.

Eksempel case: prosjektstyring

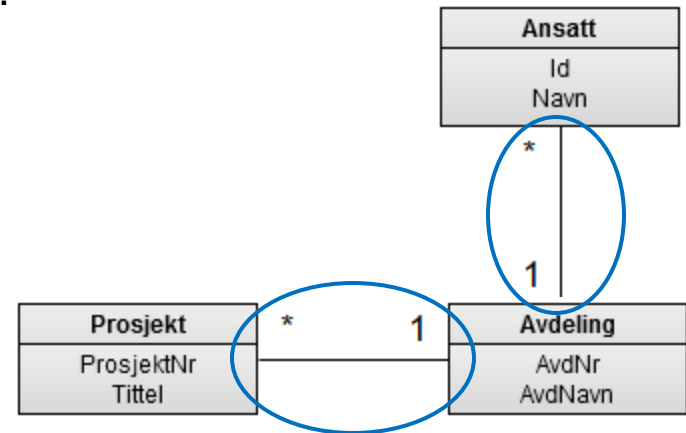
- Et firma ønsker å få oversikt over sine prosjekter. De har leid oss inn for å lage en databaseløsning som ordner dette.
- De ønsker spesifikt å få oversikt over følgende:
 - Hvilken avdeling (nummer, navn) eier hvert prosjekt?
 - Hvilke prosjekter (nummer, tittel) involverer hvilke ansatte?
 - Hvor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

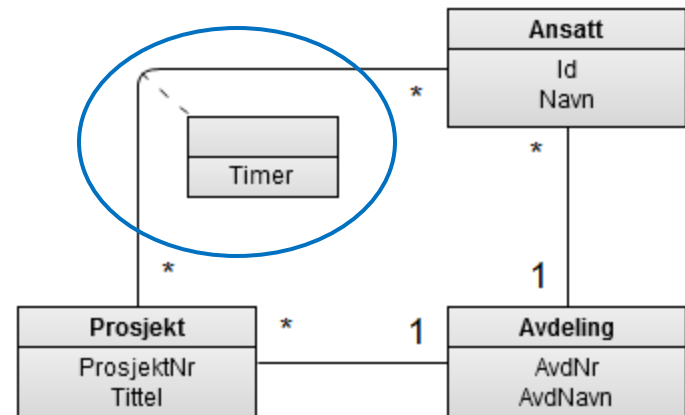
Relasjoner

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



Relasjoner – forts.

- 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 relasjonen prosjekt & ansatt!



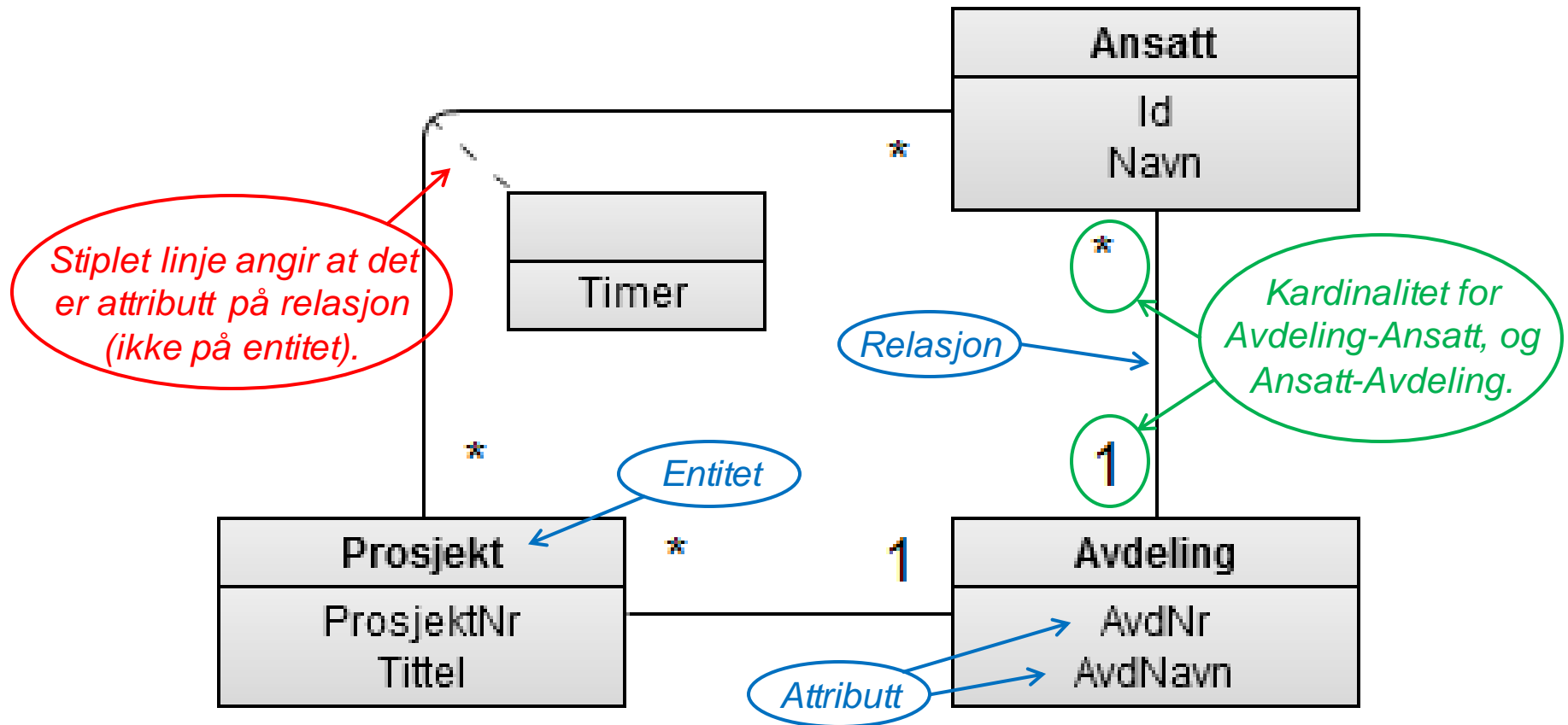
Koblingsentiteter

- Vi la inn et attributt på relasjonen mellom Ansatt og Prosjekt.
 - Relasjonen 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 7.5 i boka forklarer dette godt! :-)

Relasjoner, 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 en gitt relasjon kan ha**.
 - Eksempel: En ansatt kan maks tilhøre 1 avdeling. Kardinaliteten til ansatt i "ansatt-avdeling relasjonen" er derfor 1. (For avdeling i samme relasjon 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

Videre arbeid i dag (og kort om neste gang)

- Neste gang (torsdag om 1 uke), Modellering ER del 2:
 - Pensum er samme som til i dag : Kapittel 7 & 8.1
- Nå: (om du ikke alt har gjort det)
 - Lese kap. 7 & 8.1 i pensumboka.
 - Begynne på øvingsoppgaver!
- Etterpå:
 - Øving (flipped) 12:15 – 14:00, sjekk TimeEdit for rom.
 - Teori (flipped) 14:15 – 16:00, auditoriet.