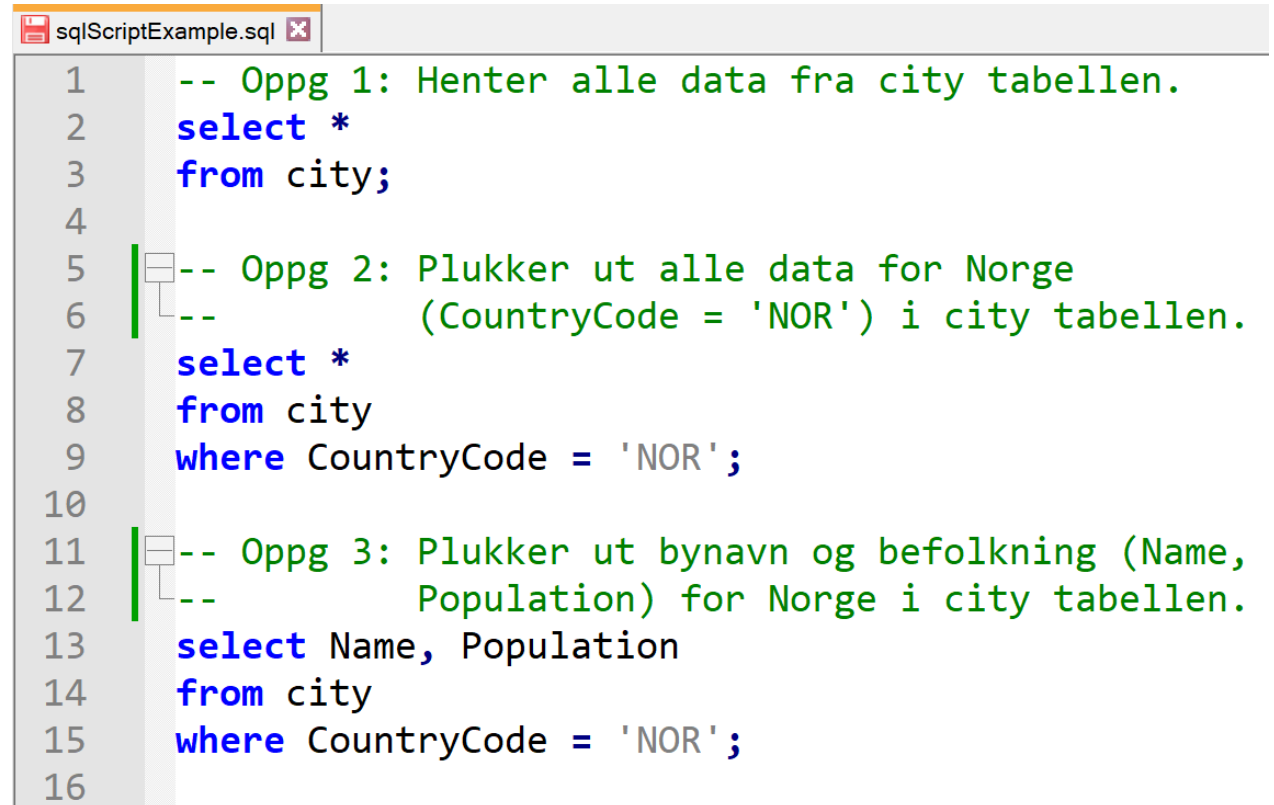


ER modellering, del 2

Modul 9 (uke 7) innhold

Mer om å levere checkpoint som sql-script

- Tar utgangspunkt i gårsdagens løsningsforslag. Skulle jeg gjort de oppgavene og levert det som sql-script kunne starten vært som fila her. →
- NB: Enkleste måten å lage dette på er kanskje å bruke Notepad++ (eller vanlig Notepad) ved siden av pgAdmin, og lime inn SQL statements der etter hvert som man løser oppgavene.



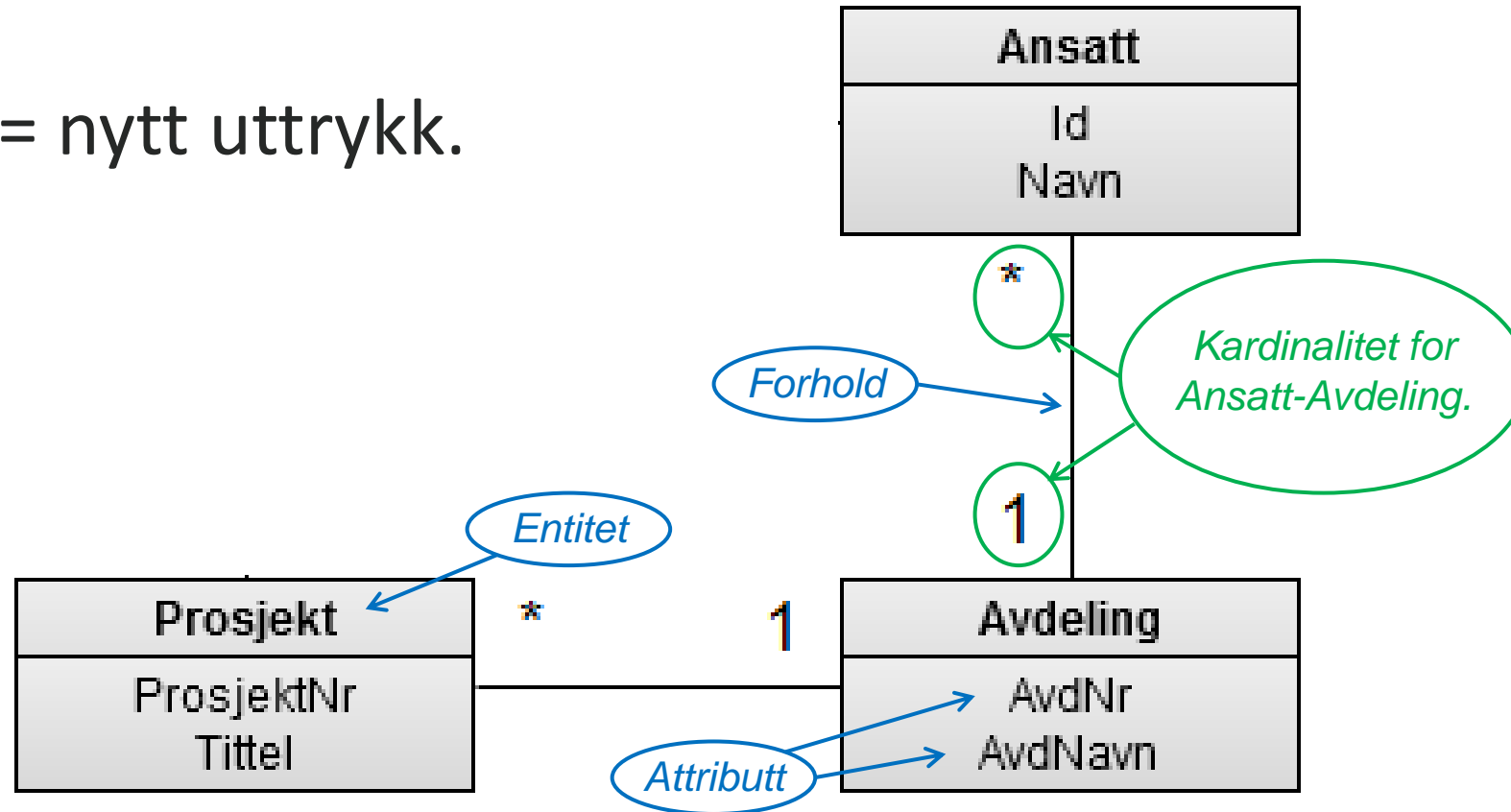
```
1  -- Oppg 1: Henter alle data fra city tabellen.
2  select *
3  from city;
4
5  -- Oppg 2: Plukker ut alle data for Norge
6  --           (CountryCode = 'NOR') i city tabellen.
7  select *
8  from city
9  where CountryCode = 'NOR';
10
11 -- Oppg 3: Plukker ut bynavn og befolkning (Name,
12 --           Population) for Norge i city tabellen.
13 select Name, Population
14 from city
15 where CountryCode = 'NOR';
16
```

Kardinalitet, nytt uttrykk

- **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

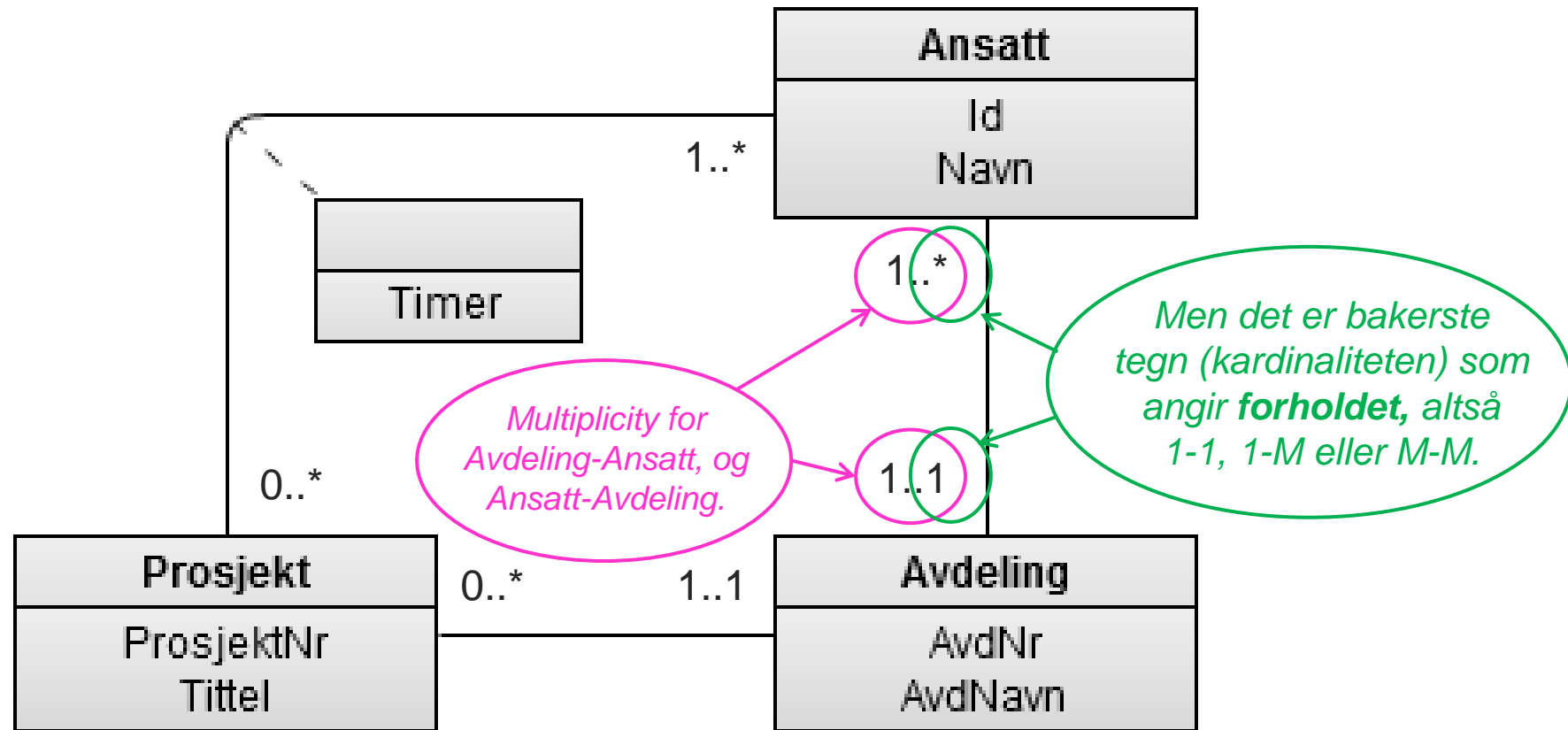
Grønn farge = nytt uttrykk.



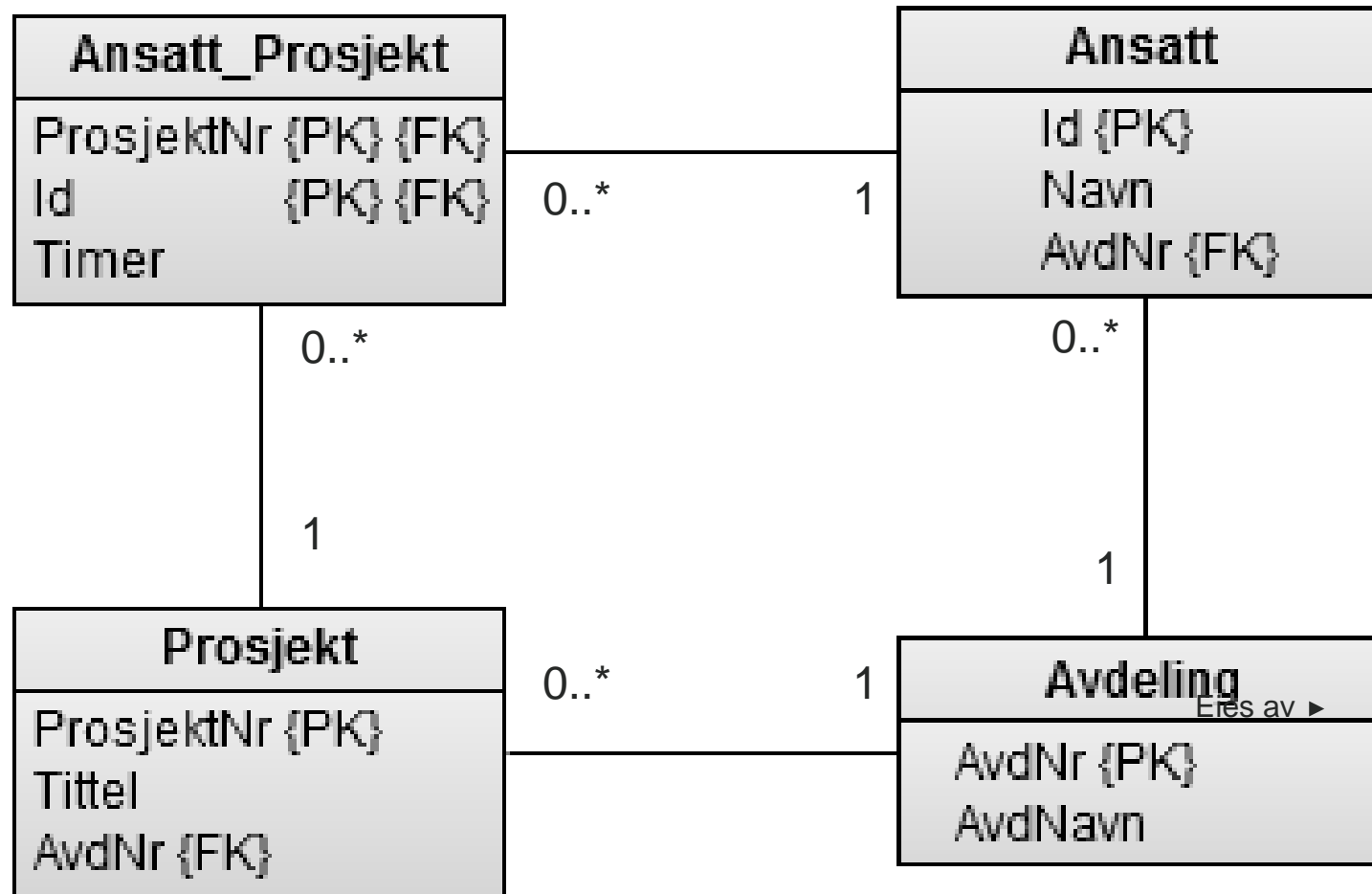
Forhold – nye begreper (pluss litt repetisjon)

- På forrige slide snakket vi om kardinalitet ifbm. ER-modellering. I tillegg til kardinalitet kan vi definere:
- **Deltakelse:** Der kardinalitet angir det største antall koplinger en entitet kan ha, angir deltakelse det minste antall koplinger en entitet kan ha.
 - I praksis verdiene 0 eller 1.
- **Multiplicity:** Deltakelse og kardinalitet satt sammen utgjør multiplicity.
 - Eksempel: En ansatt kan ha fra 1 til 1 avdeling. (Og en avdeling kan ha fra 1 til mange ansatte.)
 - Multiplicity skrives i UML: "0..1", "1..1" (evt bare "1"), "0..*", "1..*".

ER modell med UML



Databasemodell (har koplingstabell, PK og FK)



Surrogatnøkkel

- I eksempelet foran danner attributtene ProsjektNr og Id sammen primærnøkkelen.
- I stedet for en sammensatt nøkkel, kunne vi benyttet en **surrogatnøkkel**:
 - Surrogatnøkkel = **Et løpenummer uten "betydning" som kun brukes for entydig identifikasjon.**
- Hva lønner seg å benytte? Min (Tomas') erfaring:
 - Jeg har av og til angret på å *ikke* ha laget surrogatnøkkel for tabeller med sammensatte nøkler.
- Jeg (Tomas) anbefaler derfor:
 - **Alltid** surrogatnøkkel fremfor sammensatt nøkkel.
 - **Av og til** surrogatnøkkel fremfor domenespesifikk enkeltnøkkel. (Heller Id enn Email eller Username som PK!)
- Men det er ikke noe entydig fasitsvar, les f.eks. mer her:
 - stackoverflow.com/questions/1029485/use-composite-keys-or-always-use-surrogate-keys
 - stackoverflow.com/questions/23850396/composite-vs-surrogate-keys-for-referential-integrity-in-6nf

Liten tankeoppgave

- Dere har hørt om **GDPR**?
 - Dette setter bl.a. krav til at personer skal få slettet sine personlige data.
 - Samtidig er det et krav at f.eks. nettbutikker må ta vare på transaksjonshistorikk.
- Hvordan løser vi begge disse samtidig?

Hacking av databaser, teaser

- En enkel (og vanlig) form for databaseangrep ("hacking") kalles SQL injection.
- Konseptet for SQL injection, veldig kort forklart:
 - DB forventer en type input (f.eks. brukernavn / passord).
 - Vi gir den spesifikt innhold som er noe helt annet.
 - Ved å være litt kreative med dette "noe helt annet", får vi spørringen til å droppe hele brukernavn / passord sjekken!

Hacking av databaser, teaser – del 2

- Dette er spørringen:

```
SELECT Id  
FROM Students  
WHERE Name = 'brukerinntastet navn'  
AND Password = 'brukerinntastet passord';
```

- *Spm:* Hva kan vi skrive som input ("text") her for å ikke bry oss om brukernavn eller passord?
 - Svar kommer på slutten av dagen på torsdag (eller i morgen).

