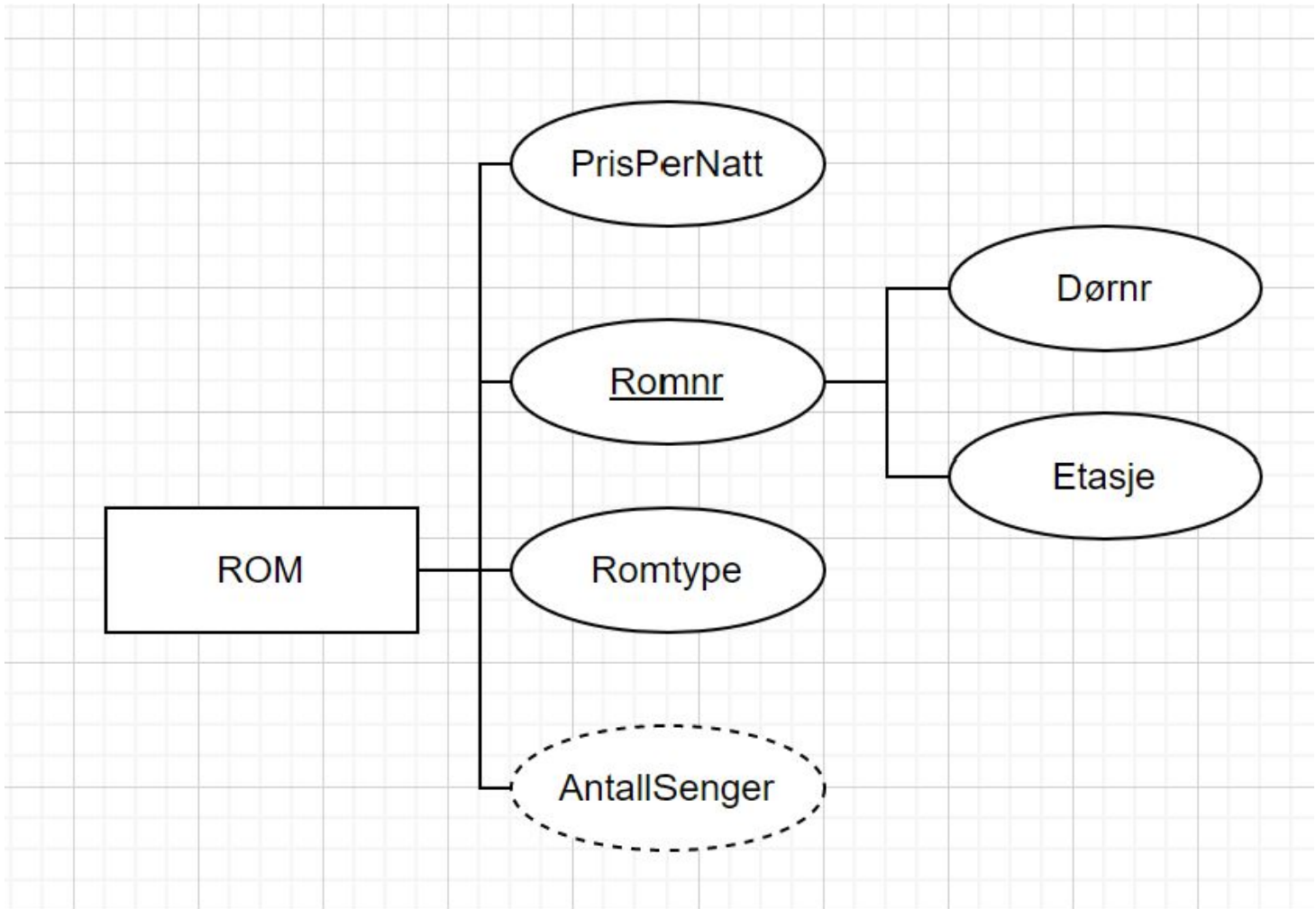
IN2090

GJENNOMGANG AV EKSAMEN 2021

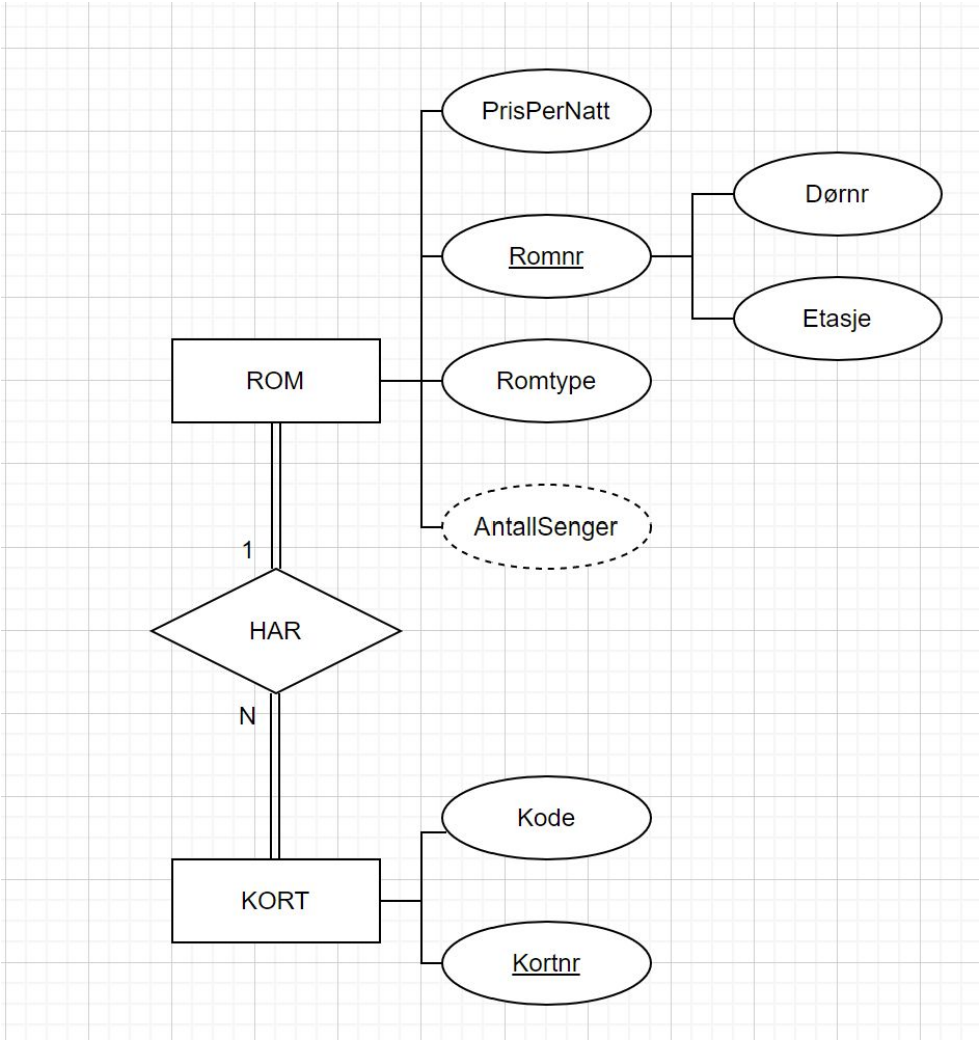
Oppgave 1 – ER: Hoteller

- Hotellet har mange rom som systemet skal lagre følgende informasjon om:
  - Et unikt romnummer, som består av etasjen rommet ligger på samt dørnummeret (f.eks. kan et romnummer være 305 hvor 3 er etasjen og 05 er dørnummeret).
  - Hvert rom har også en type (f.eks. 'dobbeltrom', 'enkeltrom', 'suite', osv.),
  - og et antall senger. Antall senger kan utledes fra romtypen.
  - Hvert rom har også en pris per natt knyttet til seg.


PrisPerNatt
Dørnr
Romnr
Etasje
ROM
Romtype
AntallSenger

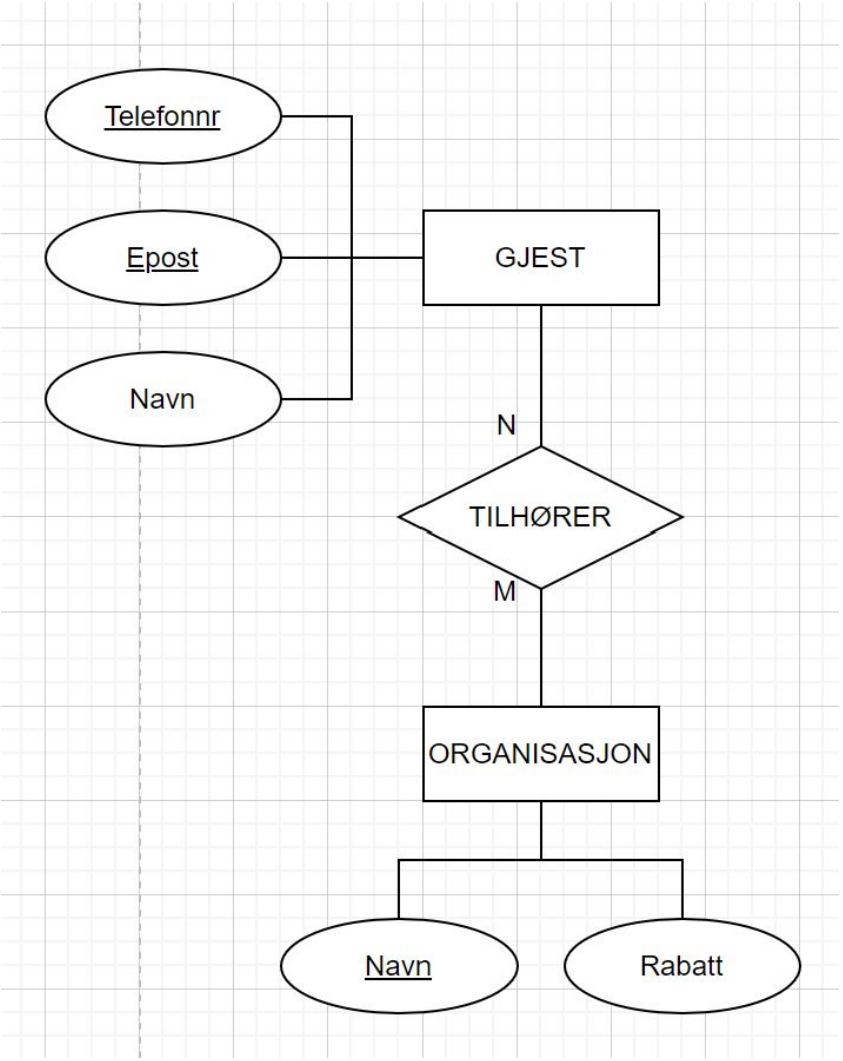
Oppgave 1 – ER: Hoteller

- Hotellet bruker adgangskort til rommene,
  - og hvert rom kan ha mange adgangskort, og hvert kort fungerer på nøyaktig ett rom.
  - Hvert rom må ha minst ett adga ngskort.
- Kortene har et unikt kortnummer og en kode.



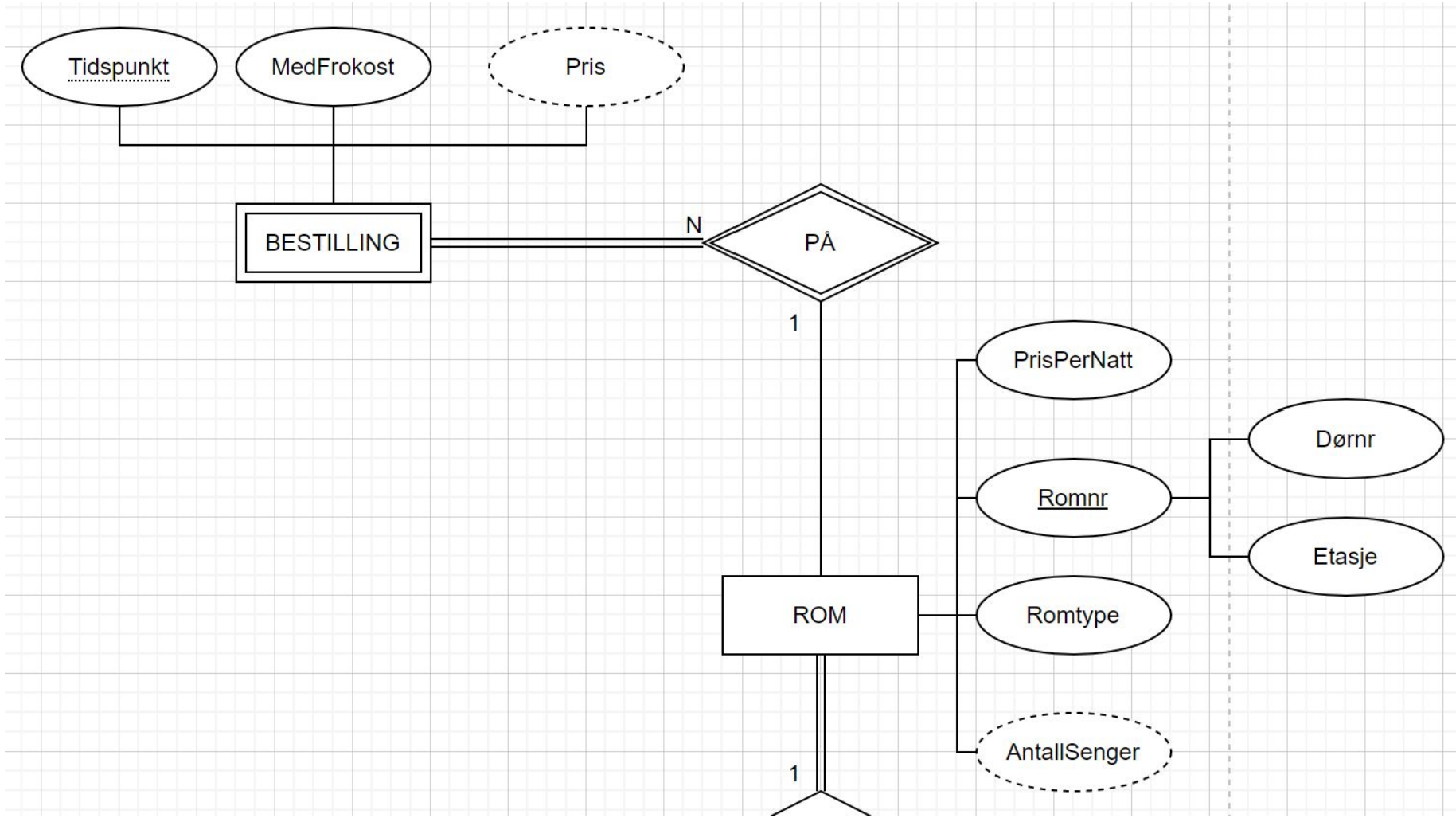
Oppgave 1 – ER: Hoteller

- Databasen skal også lagre informasjon om gjestene på hotellet. Om gjestene ønsker hotellet å
  - lagre navn,
  - samt en epostadresse
  - og et telefonnummer.
  - Både eposten og telefonnummeret er unikt for hver gjest.
- En gjest kan være tilknyttet organisasjoner, f.eks. firmaer eller offentlige organer. Hver organisasjon har
  - et unikt navn
- og en rabatt (uttrykt ved et enkelt desimaltall, f.eks. 0.1) på hotellet.
- Hver gjest kan tilhøre flere organisasjoner og en organisasjon kan ha mange gjester som tilhører den.



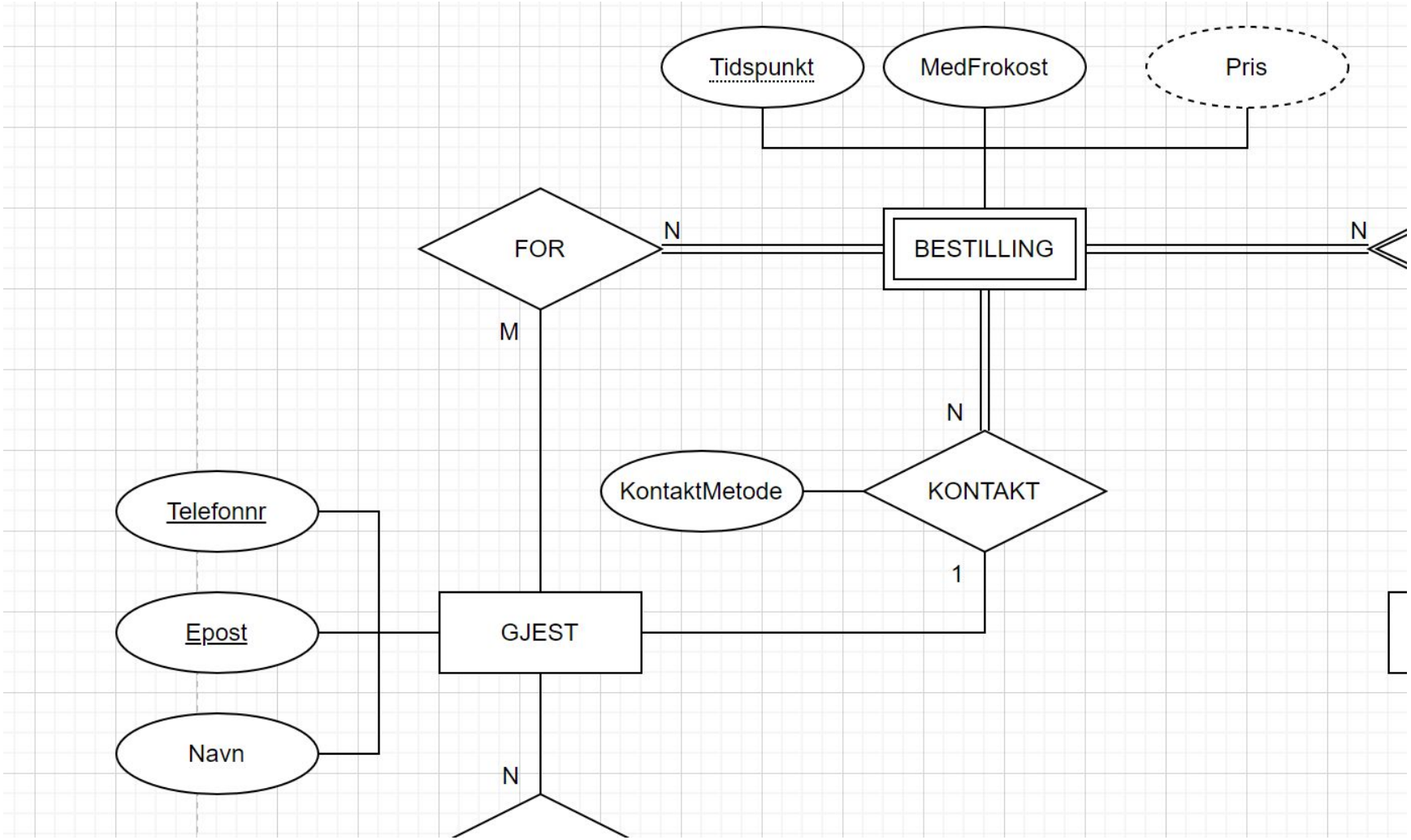
Oppgave 1 – ER: Hoteller

- En annen sentral ting som systemet skal lagre informasjon om er bestillinger.
  - En bestilling gjøres på et rom for et bestemt tidspunkt.
- Kombinasjonen av bestillingens tidspunkt og rommet bestillingen er på er unikt. Et rom kan derimot ha mange bestillinger, men kun dersom de har forskjellige tidspunkt.
  - I tillegg ønsker vi å lagre hvorvidt bestillingen inkluderer frokost eller ikke.
  - Bestillinger har også en pris, men denne kan utledes fra prisen på rommet og lengden på bestillingen.


Tidspunkt
MedFrokost
Pris
BESTILLING
N
PÅ
1
PrisPerNatt
Dørnr
Romnr
Etasje
ROM
Romtype
AntallSenger
1

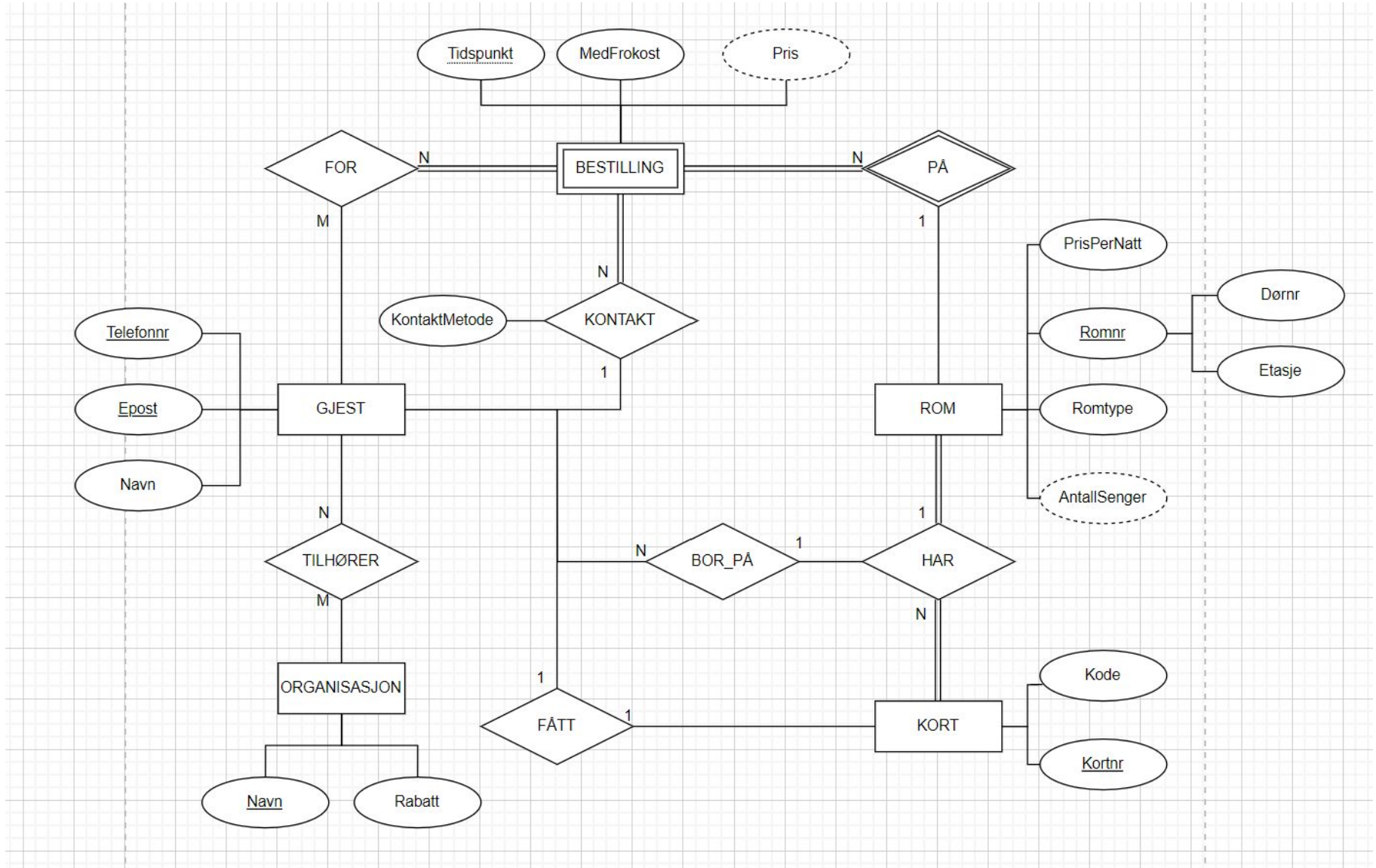
Oppgave 1 – ER: Hoteller

- En bestilling er for minst én gjest, men kan være for mange gjester,
- og en gjest kan være med på mange bestillinger.
- I tillegg har hver bestilling nøyaktig én gjest som er bestillingens kontaktperson.
  - Her ønsker vi også å lagre den foretrukne kontaktmetoden (f.eks. 'epost' eller 'telefon').
- Merk at én gjest kan godt være kontaktperson på flere bestillinger.


Tidspunkt
MedFrokost
Pris
FOR
N
BESTILLING
N
M
N
KontaktMetode
KONTAKT
Telefonnr
1
Epost
GJEST
Navn
N

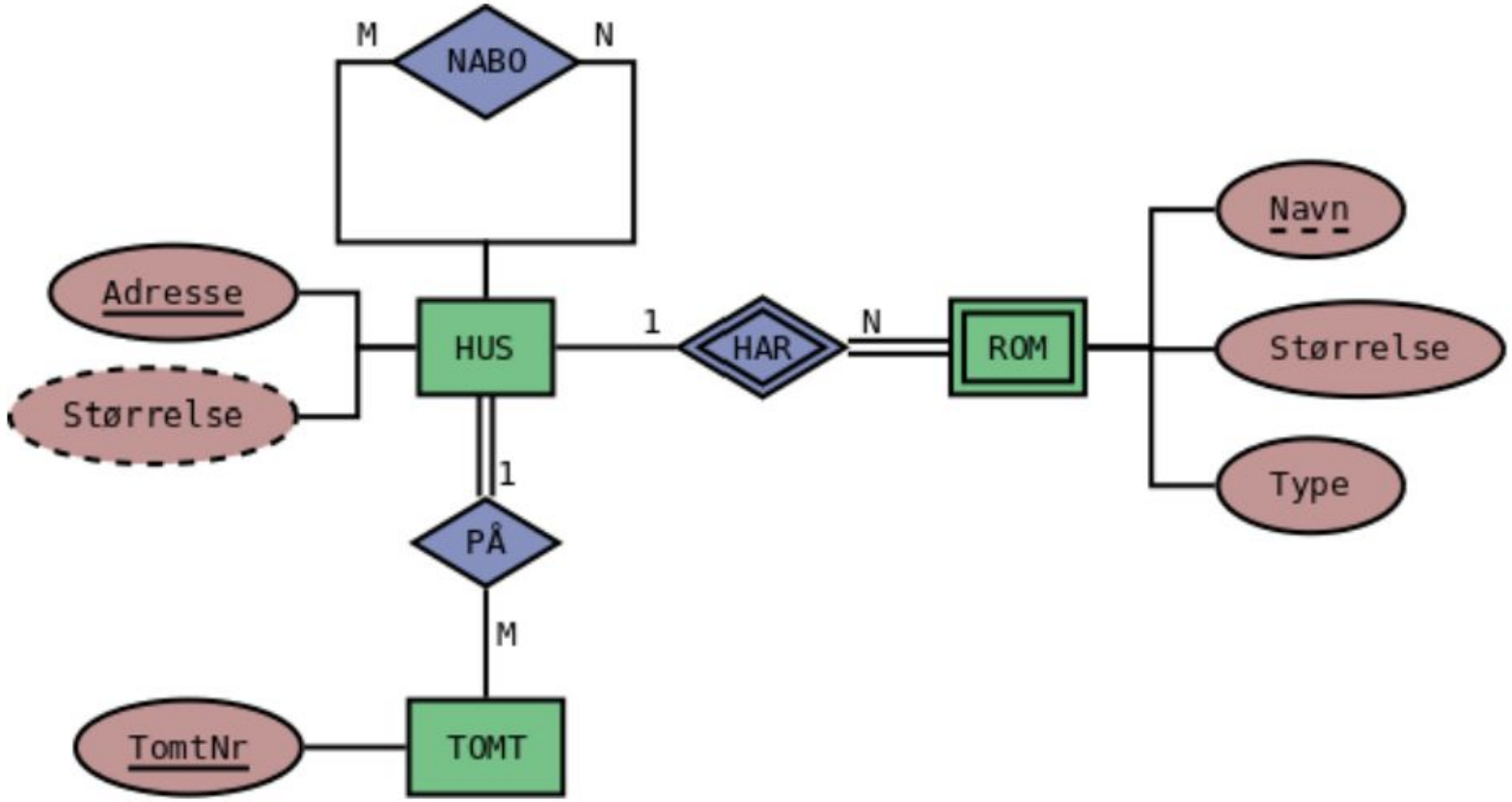
Oppgave 1 – ER: Hoteller

- Til slutt ønsker vi å lagre informasjon om hvilke gjester som nå bor på hvilke rom, og adgangskortene de har fått. Disse dataene representerer kun informasjon som er sant i øyeblikket, og ikke historisk informasjon.
- Altså kan en gjest kun bo på ett rom, men hvert rom kan ha mange gjester (for eksempel kan jo to personer bo på et dobbeltrom samtidig).
- Hver gjest får utlevert høyst ett adgangskort, og et adgangskort kan høyst gis ut til én gjest.


Tidspunkt
MedFrokost
Pris
FOR
N
BESTILLING
N
PÅ
M
1
N
PrisPerNatt
Dørnr
KontaktMetode
KONTAKT
Romnr
Telefonnr
1
Etasje
Epost
GJEST
ROM
Romtype
Navn
AntallSenger
N
1
N
BOR_PÅ
1
TILHØRER
HAR
M
N
ORGANISASJON
1
Kode
1
FÅTT
1
KORT
Kortnr
Navn
Rabatt

Oppgave 1 - ER: Hus

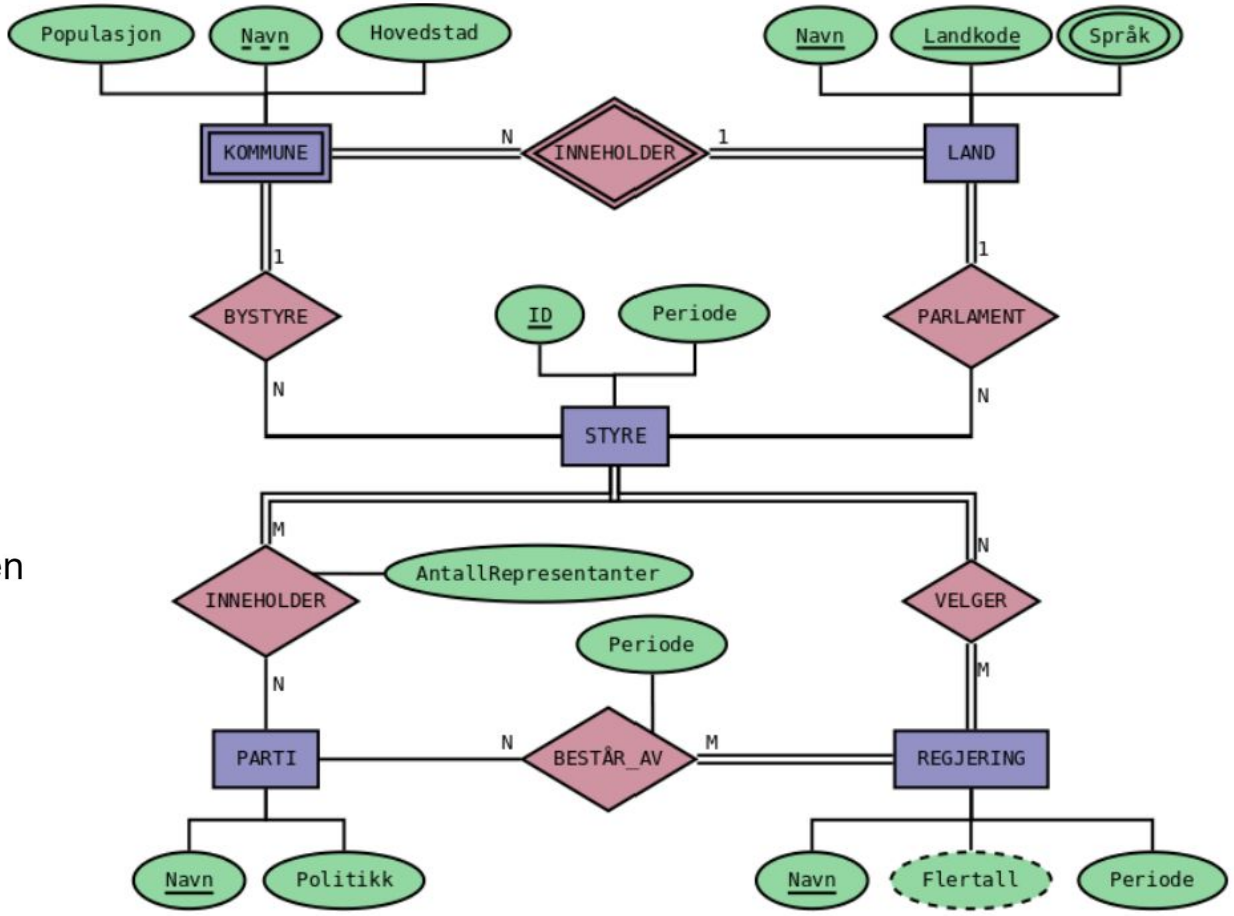
1. Et hus har minst ett rom. NEI
2. Et hus kan være på mange tomter. JA
3. Hvert hus har minst én nabo. NEI
4. Et hus kan ha mange rom med samme type. JA
5. Alle hus har en adresse. JA



Oppgave 1 - Realisering

Realisering

1. Vanlig entiteter blir tabeller
2. Svake entiteter
3. 1 : 1, tre valg:
   - flytte attributter over til en side
   - Ny tabell med alle attributter
   - Ny tabell med bare PKer og relasjonsattributt
4. 1 : N, to valg: Ny tabell eller flytte attributter over til N-siden
5. N : M, her blir relasjonen alltid en ny tabell
6. Flerverdiattributter blir alltid en ny tabell
7. Ternære



Oppgave 1 - Realisering

- Vanlige entiteter:
  - Land(**landkode**, navn)
  - Språk(land, språk)
  - Styre(ID, periode)
  - Parti(navn, politikk)
  - Regjering(navn, periode)
- Svake entiteter:
  - Kommune(navn, land, populasjon, hovedstad)
- Fremmednøkler:
  - Kommune(land) → Land(navn)
  - Språk(land) → Land(navn)

Entiteter

- Lag en tabell av entiteten, og:
  - Ta med alle vanlige attributter
  - For sammensatte attributter, ta bare med "underattributtene"
  - Velg en PK. Er det flere nøkler velg en PK og de andre som KN
- Merk:
  - Utledet/Derived (de stiplete) skal IKKE tas med
  - Flerverdiattributter skal IKKE være med (disse blir en egen tabell)

Svake entiteter

- Lag en tabell av den svake entiteten, og:
  - Ta med alle vanlige attributter
  - For sammensatte attributter, ta bare med "underattributtene"
  - Legg til PK fra den identifiserende entiteten. Da blir nøkkelen du la til fremmednøkkel til den identifiserende entiteten.
  - PK blir kombinasjonen av den svake nøkkelen og PK fra den indentifiserende entiteten
- Merk:
  - Utledet/Derived (de stiplete) skal IKKE tas med
  - Flerverdiattributter skal IKKE være med (disse blir en egen tabell)

Oppgave 1 - Realisering

- En til en relasjoner (1:1):
  - Har ingen i dette tilfellet
- En til mange/mange til en (1:N/N:1):
  - Bystyre(by, land, styre)
  - Parlament(land, styre)
- Mange til mange relasjoner (N:M):
  - Inneholder(styre, parti, antallRepresentanter)
  - Velger(styre, regjering)
  - BestårAv(parti, regjering, periode)
- Fremmednøkler:
  - Bystyre(by, land) → Kommune(navn, land)
  - Bystyre(styre) → Styre(id)
  - Parlament(land) → Land(navn)
  - Parlament(styre) → Styre(id)
  - Inneholder(styre) → Styre(id)
  - Inneholder(parti) → Parti(navn)
  - Velger(styre) → Styre(id)
  - Velger(regjering) → Regjering(navn)
  - BestårAv(styre) → Styre(id)
  - BestårAv(parti) → Parti(navn)

1 til 1 relasjoner

- 3 forskjellige måter å gjøre dette på:
  1. (Foretrukket) Velg en av sidene og ta med relasjonssattributter og PK fra motsatt side. Hvis det er en total deltakelse (dobbelt strek) med i relasjonen, velg den siden.
  2. Når begge entiteter har total deltakelse inn til relasjonen. Lag en ny tabell og ta med alle attributter fra begge entitetene + relasjonsatributter. Valgfritt hvilken av de to PKene man velger som ny PK i den nye tabellen. Den andre blir en KN
  3. "Brukes sjeldent". Formålet er å kryssreferere PKer. Lag en ny tabell med relasjonsattriibutt og begge PKene.

En til mange relasjoner (1:N eller N:1)

- To forskjellige måter å gjøre det på:
  1. Flytt relasjonssattributter (de ut fra relasjonen) til N-siden og ta med PK fra 1-siden (Foretrukket). Husk den nye Fden. PK blir den som var på N-siden fra før.
  2. Lage en ny tabell av relasjonen og ta med PK fra begge sider.

Mange til mange relasjoner (N:M)

- Lag alltid en ny tabell
- Ta med relasjonsattributter (de som går ut av relasjonen)
- Ta med PK fra begge entitetene. Disse danner en ny PK i den nye tabellen (altså felles strek under begge)

Oppgave 2 - SQL

For hver spørring under, hvor mange rader er det i svaret?

1. 0 (må bruke IS NULL og ikke = NULL for å få ut de to radene som ikke har år i oppdaget.
2. 2 (her bruker spørringen kryssprodukt, dvs alle radene i planet vil forekomme for hver kolonne i måne. Så vi har to kolonner i planet som tilfrestiller betingelsen og en I måne. Derfor blir resultatet 2x1 = 2)

1.

```
SELECT *
FROM planet
WHERE oppdaget = NULL;
```

2.

```
SELECT *
FROM planet AS p, måne AS m
WHERE p.oppdaget = 2017 AND
      m.oppdaget = 2019;
```

planet

pid	navn	masse	sid	oppdaget	oid
1	STR 002 a	0.314	4	1981	9
2	Tellus	0.597	1		
3	LHS 1723 a	1.20594	8	2017	3
4	Vega b	13.0743	2	2012	2
5	LHS 1723 b	1.37907	8	2017	3
6	Jupiter	189.249	1		
7	STR 045 c	0.717	9	2006	10
8	Neptune	9.928113	1	1846	6

måne

mid	navn	masse	pid	oppdaget	oid
1	Moon	0.007343	2		
2	Europa	0.004776	6	1610	7
3	MN 322	0.000597	1	2020	11
4	Triton	0.002388	8	1846	14
5	STR 045 c1	0.001791	7	2019	5
6	LHS1723 b3	0.005373	5	2020	5

Oppgave 2 - SQL

For hver spørring under, h vor mange rader er det i s varet?

1. Her forekommer 2017 to ganger (i planet), 2020 to ganger og 2019 en gang (i måne), vi vil da få ut 3 rader.
2. Her får vi bare med de stjerneid-ene som forekommer mer enn en gang fra planet tabellen, ser at det er 2 sid som forekommer mer enn en gang (sid = 1 og sid = 8)

3.

```
SELECT oppdaget FROM planet
WHERE oppdaget >= 2017
UNION
SELECT oppdaget FROM måne
WHERE oppdaget >= 2017;
```

4.

```
SELECT sid, count(*)
FROM planet
GROUP BY sid
HAVING count(*) > 1;
```

planet

pid	navn	masse	sid	oppdaget	oid
1	STR 002 a	0.314	4	1981	9
2	Tellus	0.597	1		
3	LHS 1723 a	1.20594	8	2017	3
4	Vega b	13.0743	2	2012	2
5	LHS 1723 b	1.37907	8	2017	3
6	Jupiter	189.249	1		
7	STR 045 c	0.717	9	2006	10
8	Neptune	9.928113	1	1846	6

måne

mid	navn	masse	pid	oppdaget	oid
1	Moon	0.007343	2		
2	Europa	0.004776	6	1610	7
3	MN 322	0.000597	1	2020	11
4	Triton	0.002388	8	1846	14
5	STR 045 c1	0.001791	7	2019	5
6	LHS1723 b3	0.005373	5	2020	5

Oppgave 2 - SQL

For hver spørring under, hvor mange rader er det i svaret?

1. Vi har oid = 13, 9, 2, 12, 8, 1, 4 i stjerne. Type = 'satellitt' for oid = 2, 9 og 11 I observator. Ville endt opp med 2 rader hvis vi hadde brukt inner join. Men siden vi bruker natural join vil vi ikke finne noen rader som er like, ettersom navn i stjerne og navn i obseravtor er representerer ulike ting.
2. Velger distinct type = 'satellitt' for da bare ut satelitt. 1 rad.

```
SELECT *
FROM observator NATURAL JOIN stjerne
WHERE type = 'satellitt';
```

6.

```
SELECT DISTINCT type
FROM observator
WHERE type = 'satellitt';
```

observator

oid	type	navn
1	observatorie	Palomar Observatory
2	satellitt	Infrared Astronomical Satellite
3	observatorie	La Silla Observatory
4	person	Sir Isaac Newton
5	observatorie	Girawali Observatory
6	person	Johann Galle et al.
7	person	Gallileo Gallilei
8	observatorie	Helsinki University Observatory
9	satellitt	Astron
10	observatorie	BESS
11	satellitt	Infrared Space Observatory
12	person	Walter Sidney Adams
13	person	John William Draper
14	person	William Lassell

stjerne

sid	navn	masse	lysstyrke	oppdaget	oid
1	Sun	198.9	1		
2	Vega	424.6	40.12	1900	13
3	Sirius a	398.1	25.4		
4	STR 002	218.6	11.4	1912	9
5	STR 312	98.1	0.56	2018	2
6	Sirius b	412.6	25.4	1915	12
7	STR 987	123.4	2.7	2001	8
8	LHS 1723	33.91245	0.00365	1982	1
9	STR 045	619.5	70.5	1701	4

Oppgave 2 - SQL

- Skriv en SQL-spørring som finner navn, masse og lysstyrke på alle stjerner som har masse større enn 200 eller lysstyrke større enn 10.

```
SELECT navn, masse, lysstyrke
FROM stjerne
WHERE masse > 200 OR
      lysstyrke > 10;
```

Oppgave 2 - SQL

- Skriv en SQL-spørring som finner navn og masse på planeter som er i samme solsystem som planeten med navn 'Tellus'. Sorter resultatet etter masse fra minst til størst.

```
SELECT p2.navn, p2.masse
FROM planet AS p1
     JOIN planet AS p2 USING (stjerne)
WHERE p1.navn = 'Tellus'
ORDER BY p2.masse;
```

Oppgave 2 - SQL

- Det viser seg at observatoriet med navn 'BESS' konsekvent har rapportert massen på alle sine observasjoner av stjerner gjort før '1990' i pund fremfor kilo. Skriv derfor en SQL-kommando som oppdaterer massen til alle stjerner observert før dette året av dette observatoriet til kilogram. Merk at ett pund er 0.45 kilogram, altså blir ny masse lik 0.45 ganger tidligere masse.

```sql
UPDATE stjerne
SET masse = 0.45 * masse
WHERE oppdaget < 1990 AND oid = (
  SELECT oid
  FROM observator
  WHERE navn = 'BESS'
);
```

Oppgave 2 - SQL

- Skriv en SQL-kommando som lager et VIEW med navn universet som inneholder to kolonner: det totale antallet astronomiske objekter i universet (altså totalt antall stjerner, planeter og måner som er i databasen), samt den totale massen til universet (altså den totale massen til alle disse objektene i databasen).

```sql
CREATE VIEW universet(antall, masse) AS
WITH
  alle_objekter AS ( -- Massen til alle objekter i universet
    SELECT masse FROM stjerne
    UNION ALL
    SELECT masse FROM planet
    UNION ALL
    SELECT masse FROM måne
  )
SELECT count(*) AS antall, sum(masse) AS masse
FROM alle_objekter;
```

Oppgave 2 - SQL

- Et solsystem er en mengde astronomiske objekter som går i bane rundt en stjerne, altså alle planetene som går i bane rundt stjernen samt alle disse planetenes måner. Skriv en SQL-spørring som finner navn på alle stjerner som er en del av et stort solsystem, hvor et stort solsystem er et solsystem med mer enn 10 planeter eller som har total masse (på både stjernen, planetene og månene til sammen) på mer enn 400.

```sql
WITH
  solsystemer AS (
    -- Knytter sid og stjernens navn til hvert
    -- objekts masse i stjernens solsystem
    SELECT sid, navn AS stjerne, masse
    FROM stjerne
    UNION ALL
    SELECT s.sid, s.navn AS stjerne, p.masse
    FROM stjerne AS s
         JOIN planet AS p USING (sid)
    UNION ALL
    SELECT s.sid, s.navn AS stjerne, p.masse
    FROM stjerne AS s
         JOIN planet AS p USING (sid)
         JOIN måne AS m USING (pid)
  )
SELECT stjerne
FROM solsystemer
GROUP BY sid, stjerne
HAVING sum(masse) > 400
UNION ALL
SELECT s.navn AS stjerne
FROM stjerne AS s
     JOIN planet AS p USING (sid)
GROUP BY s.sid, s.navn
HAVING count(*) > 10
```

Oppgave 3 - Relasjonsmodellen og normalformer

- Bruk samme skjema som for SQL-oppgavene over, og skriv en SQL-spørring som gir samme resultat (samme tupler/rader) som relasjonsalgebra-uttrykket under:

$$\pi_{navn}\left(observator \bowtie_{oid=soid} \pi_{soid}\left(\rho_{oid \to soid}\left(\sigma_{lysstyrke>50}\left(stjerne\right)\right)\right)\right)$$

```sql
SELECT o.navn
FROM observator AS o
     JOIN stjerne AS s USING (oid)
WHERE s.lysstyrke > 50;
```

```sql
SELECT navn
FROM observator
     JOIN (SELECT oid AS soid
           FROM stjerne
           WHERE lysstyrke > 50) AS t ON oid = soid;
```

Oppgave 3
- Relasjonsmodellen og normalformer

- Gitt følgende relasjon:
  R(A, B, C, D, E, F, G)
  med kandidatnøkler BCFG og BDFG.
  For hver av FDene under, anta at FDen gjelder for relasjonen R over, og bruk algoritmen for å finne normalform til å avgjøre hvilken normalform FDen (alene) tilsier at R er på.

FD	1NF	2NF	3NF	BCNF
$BD \rightarrow C$				
$AC \rightarrow D$				
$ABC \rightarrow E$				
$DEF \rightarrow C$				
$CFG \rightarrow A$				

Oppgave 3
- Relasjonsmodellen og normalformer

- R(A, B, C, D, E, F, G)
- BD → C
  - BD* = BDC, ikke supernøkkel. C er et nøkkelattributt, 3NF.
- AC → D
  - AC* = ACD, ikke supernøkkel. D er et nøkkelattributt, 3NF
- ABC → E
  - ABC* = ABCDE, ikke en supernøkkel. E er ikke et nøkkel attributt. ABC er ikke del av en kandidatnøkkel, 2NF.
- DEF →C
  - DEF* = CDEF, ikke en supernøkkel. C er et nøkkelattributt, 3NF.
- CFG → A
  - CFG* = ACDFG, ikke en supernøkkel. A er ikke et nøkkelattributt. CFG er del av en kandidatnøkkel, 1NF.

Bestemme normalform

- Først, finn alle kandidatnøkler
- For hver tabell og hver FD X → A:
  1. Er X en supernøkkel?
     Ja: BCNF sålangt, gå til neste FD
     Nei: brudd på BCNF. Gå til 2.
  2. Er A et nøkkelattributt?
     Ja: 3NF sålangt, gå til neste FD
     Nei: brudd på 3NF. Gå til 3.
  3. Er X del av en kandidatnøkkel?
     Nei: 2NF sålangt, gå til neste FD
     Ja: brudd på 2NF og skjema er på 1NF, stopp.
- Tabellen er så på den laveste normalformen vi får ut av denne algoritmen
- Skjemaet er på den laveste normalformen av tabellenes
- Med andre ord: Hvis jeg har en tabell og en FD som bryter 2NF, er skjemaet på 1NF.

Oppgave 3
- Relasjonsmodellen og normalformer

FD	1NF	2NF	3NF	BCNF
$BD \rightarrow C$			X	
$AC \rightarrow D$			X	
$ABC \rightarrow E$		X		
$DEF \rightarrow C$			X	
$CFG \rightarrow A$	X			

Oppgave 3 - Tapsfridekomposisjon

Gitt følgende relasjon: R(A, B, C, D, E, F)
med kandidatnøkkel ABD, og følgende FDer:

1. $AB \rightarrow C$
2. $B \rightarrow E$
3. $D \rightarrow E$
4. $D \rightarrow F$

Dekomponer R tapsfritt til BCNF. Vis stegene du gjør og list opp nøkler og FDer på alle relasjonene underveis.

Oppgave 3 - Tapsfridekomposisjon

R(A, B, C, D, E, F)

1. **AB → C**
2. **B → E**
3. **D → E**
4. **D → F**

- AB → C bryter med BCNF, siden AB ikke er en supernøkkel.
  - Finner tillukningen til AB: AB+ = ABCE
  - Dekomponerer R til S1(A, B, C, E) og S2(A, B, D, F).
- For S1(A, B, C, E) gjelder FD 1 og 2, og har dermed eneste kandidatnøkkkel AB.
  - FD 1 bryter da ikke med BCNF siden AB er en supernøkkel, men FD 2 bryter med BCNF siden B ikke er en supernøkkel. Dekomponerer derfor med hensyn på denne: B+ = BE Dekomponer S1 til S11(B, E) og S12(A, B, C).
- For S11 gjelder bare FD 2, og er dermed på BCNF og har eneste kandidatnøkkel B.
- For S12 gjelder kun FD 1, er dermed også på BCNF og har eneste kandidatnøkkel AB.
- S2(A, B, D, F) har kun FD 4.
  - Vi ser at ABD+ = ABDF og er dermed også (eneste) kandidatnøkkel for S2. Siden D ikke er en supernøkkel bryter dermed FDen med BCNF og vi dekomponerer med D+ = DF: S21(A, B, D) og S22(D, F).
- For S21 gjelder ingen FDer og den er dermed på BCNF (med ABD som eneste kandidatnøkkel),
- mens for S22 holder kun FD 4, og får dermed eneste kandidatnøkkel D, og bryter dermed ikke med BCNF.
- **Altså kan R dekomponeres tapsfritt til S11(B, E), S12(A, B, C), S21(A, B, D) og S22(D, F)**

Siste uke med gruppetimer

- Repetisjon i de ulike temaene:
  - ER-modellering
  - Relasjonsmodellen
  - SQL
- Gå i den gruppetimen du ønsker basert på temaet du vil repetere.
- Det er lov å møte opp i flere gruppetimer.
- Lykke til med øving til eksamen!

Gruppelærer	**Tema**	**Tidspunkt**	**Rom**
Alexander	Relasjonsmodellen	Onsdag 14:15-16:00	Aud. Smalltalk
Sindre	Relasjonsmodellen	Tirsdag 12:15-14:00	Sem. Logo
Camilla	SQL	Torsdag 10:15-12:00	Sem. C
Elise	ER-modellering	Mandag 14:15-16:00	Dat. Limbo
Jamila	ER-modellering	Onsdag 10:15-12:00	Sem. Logo
Julia	SQL	Onsdag 12:15-14:00	Sem. C