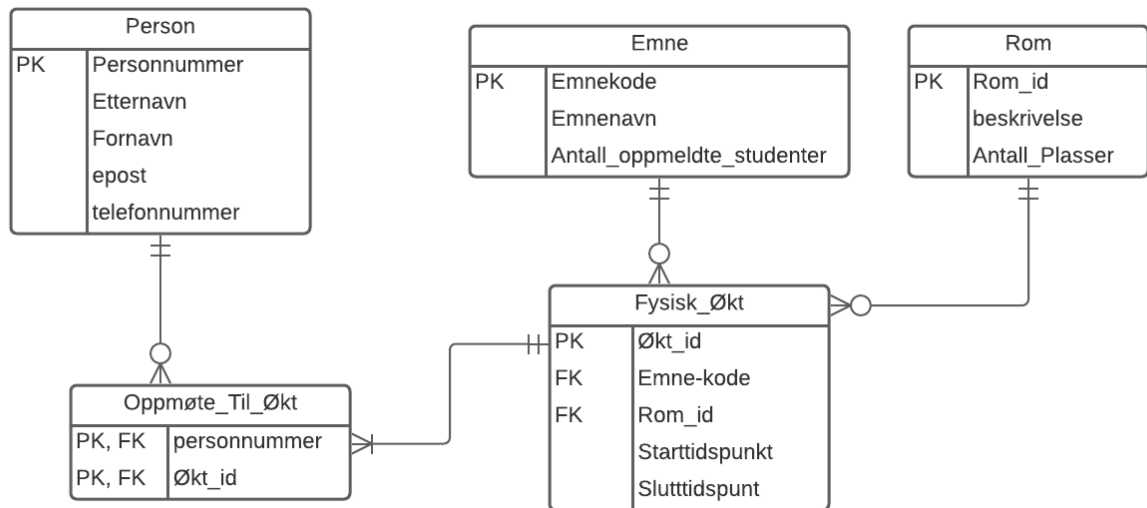


## Oppgave 1 – modellering

Smittesporing:



## Oppgave 2 – MySQL

Konferanse:

-- 2a)

```
SELECT *  
FROM deltaker  
ORDER BY etternavn ASC, fornavn ASC;
```

	DNr	Fornavn	Etternavn	EPost
►	15	Anders	Andersen	anders@andersen.no
	6	Benny	Ball	benny@benny.no
	10	Billy	Betong	billy@ppbb.no
	4	Eva	Dahl	eva@dahl.no
	12	Frida	Frosk	frida@ppbb.no
	1	Hans	Hansen	hans@hansen.no
	8	Hans	Jensen	hj@jensen.no
	17	Igor	Olsen	igor@olsen.no
	3	Jens	Jensen	jens@jensen.no
	16	Julie	Jensen	julie@jensen.no
	2	Kari	Normann	kari@normann.no
	13	Leon	Latex	leon@ppbb.no
	5	Ole	Olsen	ole@olsen.no
	7	Oline	Jensen	o-j@jensen.no
	11	Pelle	Parafin	pelle@ppbb.no
	14	Ragna	Rekkverk	ragna@ppbb.no
	9	Sandra	Salamander	sandra@ppbb.no
	18	Svetlana	Iversen	svetlana@olsen.no

-- 2b)

```
SELECT fornavn, etternavn  
FROM deltaker  
WHERE EPost LIKE "%@ppbb.no";
```

	fornavn	etternavn
►	Sandra	Salamander
	Billy	Betong
	Pelle	Parafin
	Frida	Frosk
	Leon	Latex
	Ragna	Rekkverk

-- 2c)

```
SELECT dagnr, sum(måltidpris) AS Sum_måltidspris_for_de_ulike_dagene
FROM måltid
GROUP BY dagnr;
```

	dagnr	Sum_måltidspris_for_de_ulike_dagene
►	1	278
	2	278

-- 2d)

```
SELECT *
FROM deltaker
WHERE dnr NOT IN (SELECT dnr FROM forfatter);
```

	DNr	Fornavn	Etternavn	EPost
►	15	Anders	Andersen	anders@andersen.no
	16	Julie	Jensen	julie@jensen.no
	17	Igor	Olsen	igor@olsen.no
	18	Svetlana	Iversen	svetlana@olsen.no

-- 2e)

```
SELECT d.fornavn, d.etternavn, COUNT(dt.TemaNr) as AntallTemaer
FROM deltaker AS d
LEFT JOIN deltakertema AS dt ON dt.dnr = d.dnr
GROUP BY d.dnr
ORDER BY AntallTemaer DESC;
```

	fornavn	etternavn	AntallTemaer
►	Hans	Hansen	5
	Billy	Betong	5
	Oline	Jensen	4
	Ole	Olsen	3
	Hans	Jensen	3
	Pelle	Parafin	3
	Frida	Frosk	3
	Kari	Normann	2
	Eva	Dahl	2
	Leon	Latex	2
	Jens	Jensen	1
	Benny	Ball	1
	Sandra	Salamander	1
	Ragna	Rekkverk	0
	Anders	Andersen	0
	Julie	Jensen	0
	Igor	Olsen	0
	Svetlana	Iversen	0

-- 2f)

```
INSERT INTO måltidbestilling
VALUES
  (1, "lunsj", 1),
  (1, "middag", 1),
  (2, "lunsj", 1),
  (2, "middag", 1),
  (3, "lunsj", 1);
```

✓ 1140 15:07:34 INSERT INTO måltidbestilling VALUE... 5 row(s) affected Records: 5 Duplicates: 0 Warnings... 0.000 sec

Kontroll:

```
SELECT *
FROM måltidbestilling;
```

	DNr	MåltidType	DagNr
▶	1	Middag	1
	2	Middag	1
	1	Lunsj	1
	2	Lunsj	1
	3	Lunsj	1

-- 2g)

```
UPDATE deltaker
SET epost = "svetlana@iversen.no"
WHERE dnr = 18;
```

✓ 1143 15:09:18 UPDATE deltaker SET epost = "svetlana@iversen.no" 1 row(s) affected Rows matched: 1 Changed: 1 War... 0.000 sec

Kontroll:

```
SELECT *
FROM deltaker
WHERE dnr = 18;
```

	DNr	Fornavn	Etternavn	EPost
▶	18	Svetlana	Iversen	svetlana@iversen.no

-- 2h)

```
SELECT etternavn, COUNT(*) AS Antall
FROM deltaker
GROUP BY etternavn
HAVING Antall > 1;
```

	etternavn	Antall
▶	Jensen	4
	Olsen	2

-- 2i)

```
CREATE TABLE transport(  
  dnr int not null UNIQUE,  
  fraFlyplassen ENUM ("ja", "nei", "ikke oppgitt") DEFAULT("ikke oppgitt"),  
  tilFlyplassen ENUM ("ja", "nei", "ikke oppgitt") DEFAULT("ikke oppgitt"),  
  PRIMARY KEY (dnr),  
  FOREIGN KEY (dnr) REFERENCES deltaker (dnr)  
);
```

Kontroll:

```
SELECT *  
FROM transport;
```

	dnr	fraFlyplassen	tilFlyplassen
*	NULL	NULL	NULL

-- 2j)

```
CREATE OR REPLACE VIEW infoPresentasjoner_view AS  
SELECT r.romnr,  
       concat(DATE_FORMAT(p.starttid, "%d. %M kl %H.%i"), "- ",  
DATE_FORMAT(DATE_ADD(p.starttid, INTERVAL p.varighet MINUTE), "%H.%i")) AS tidspunkt,  
       p.tittel, concat(d.fornavn, " ", d.etternavn) AS foredragsholder, t.temanavn, r.antPlasser  
FROM presentasjon AS p  
  JOIN rom AS r ON r.romnr = p.romnr  
  JOIN deltaker AS d ON d.dnr = p.dnr  
  JOIN tema AS t ON t.temanr = p.temanr  
ORDER BY r.romnr ASC, p.starttid ASC;
```

Kontroll:

```
SELECT *  
FROM infoPresentasjoner_view;
```

	romnr	tidspunkt	tittel	foredragsholder	temanavn	antPlasser
▶	A1	24. november kl 09.45-10.05	Feasability of Optimizations Requiring Bounded ...	Hans Hansen	Performance and Optimization	100
	A1	24. november kl 10.15-10.35	Evaluation of graph algorithm frameworks for m...	Kari Normann	Performance and Optimization	100
	F1	24. november kl 09.45-10.05	IT students perceptions of mandatory coursework	Jens Jensen	IT didactics	50
	F1	24. november kl 10.15-10.35	Introducing ePortfolios to IT students: The sup...	Eva Dahl	IT didactics	50
	F1	24. november kl 10.45-11.05	Teaching AI Ethics: Observations and Challenges	Ole Olsen	IT didactics	50
	F1	24. november kl 11.15-11.35	The Live Programming Lecturing Technique: A S...	Benny Ball	IT didactics	50
	F2	25. november kl 09.45-10.05	INERTIA AND CHANGE IN TRANSFORMATION ...	Oline Jensen	Digital transformation	40
	F2	25. november kl 10.15-10.35	DIGITAL TRANSFORMATION UNDER A PANDEM...	Hans Jensen	Digital transformation	40
	F2	25. november kl 10.45-11.05	Exploring the Impact of Mob Programming on th...	Sandra Salamander	Digital transformation	40
	F2	25. november kl 11.15-11.35	Exploring the Hiring Process of a Norwegian Mu...	Billy Betong	Digital transformation	40

Brukte følgende for å endre visning av månedsnavn fra engelsk til norsk:

```
set lc_time_names = "no_NO";
```

## Oppgave 3 – Normalisering

Bedrift:

Filial		
Filialnr (PK)	Adresse	Sted
1	Smalveien 1	Oslo
2	Brygga 2	Oslo

Rom		
Romnavn (PK)	Filialnr (FK, PK)	Etasje
12	1	1
22	1	2
Gløtt	2	5
Regn	2	5

Arbeidsplass		
Ansattnr (FK, PK)	Romnavn (FK)	Filialnr (FK)
123456	12	1
234567	12	1
345678	22	1
445544	Gløtt	2
554455	Gløtt	2
989898	Regn	2
323232	Regn	2

Ansatt			
Ansattnr (PK)	Fnavn	Enavn	Tlfnr
123456	Jens	Jensen	55555555
234567	Kari	Normann	66666666
345678	Ola	Olsen	77777777
445544	Lise	Olsen	88888888
554455	Per	Persen	88668866
989898	Eva	Jensen	45454545
323232	Nils	Nilsen	23343223

Utstyrsfordeling	
Ansattnr (FK)	Utstyrs_ID (PK, FK)
123456	1
123456	2
123456	3
234567	4
234567	5
234567	6
234567	7

Utstyr				
Utstyrs_ID (PK)	Utstyr	Innkjøpspris	Innkjøpsdato	Type
1	PC	10000	02.01.2019	Lenovo
2	Mobil	9000	02.01.2019	Iphone
3	Stol	3490	12.12.2018	Zareto
4	Mac	13900	05.05.2017	Macbook Pro
5	Mobil	9900	05.05.2019	Samsung
6	Stol	3900	05.07.2017	Watford
7	Headsett	2900	05.08.2017	Boss

Ansattnr, Utstyr og innkjøpsdato som sammensatt primærnøkkel er en dårlig primærnøkkel, fordi vi ikke kan registrere et utstyr hvis innkjøpsdato er ukjent eller hvis den ikke er tilknyttet en ansatt. Den er heller ikke unik dersom utstyret er kjøpt på samme dato og brukes av samme ansatt. Vi oppretter derfor en egen unik utstyrsid som primærnøkkel for utstyr.

Jeg opprettet filialnr som PK for filialer. Jeg antar at det ikke finnes rom med samme romnavn i samme filial.

Databasen er nå i 3NF:

- Hvert felt inneholder kun ett dataelement
- Ingen delvis funksjonell avhengighet til PK. I den opprinnelige databasen var det i den første tabellen tilfelle av delvis avhengighet:
  - Ansattnr → Fnavn, Enavn og Tlfnr
- Ingen indirekte avhengighet til PK ved å splitte ansatt, utstyr rom og avdelingskontor til egne tabeller



#### Oppgave 4 – diverse

Tiltak som kan bidra til å øke spørringshastigheten mot en relasjonsdatabase, særlige for spørringer som brukes mye:

1. Vi kan lage indeks for kolonnene som brukes mye i spørringene våre, spesielt de kolonnene vi søker gjennom. Typisk vil være kolonner som inngår i WHERE-clause. Hvis vi ikke bruker indeks, søkes det gjennom hver rad i kolonnen, og dette kan ta la lang tid hvis det er et stort datasett. Når vi indekserer lagrer vi en indeksliste for sortering av innholdet i kolonnen, og når vi gjør søk i denne vil den bruke indekslisten, nesten som en stikkordsliste i en bok, for å hente dataene i spørringen, og da kan spørringen finne dataene det søkes etter uten at den må se på hver eneste rad. I noen tilfeller kan hastighetene øke med flere hundre ganger. Dette er det første tiltaket man bør se på, da det er enkelt og billig å implementere. Bakdelene med indeksering er at oppdatering av data i tabellen tar lengre tid. Dette skyldes at indekslisten må oppdateres hver gang vi gjør en endring i tabellen.
2. Vi kan denormalisere databasen, som er det motsatte av å normalisere. Dette innebærer at vi legger sammen tabeller. For systemet er join mellom tabeller en ressurskrevende prosess. Færre joins vil derfor føre til raskere spørringer, men dette på bekostning av økt redundans.
3. Kjøpe bedre utstyr og raskere internett og fysisk nærhet til serveren. Man bør først vurdere de andre tiltakene, da dette er en dyr løsning.