

## Sensorveiledning

Emnekode	Emnenavn	Eksamensdato	Campus	Faglærers navn
DAT1000	Database 1	14.12.2023	Bø	Bjørn Kristoffersen

### Oppgave 1

1-a

```
SELECT *
FROM Fottur
WHERE YEAR(StartDato) = 2023
AND AntDager = 3;
```

1-b

```
SELECT *
FROM Hytte
WHERE AntSenger > 10
AND Navn LIKE '%bu%'
ORDER BY AntSenger DESC;
```

1-c

```
SELECT Påmelding.*, CONCAT(Deltaker.Fornavn, ' ', Deltaker.Etternavn) AS Navn
FROM Deltaker INNER JOIN Påmelding ON Deltaker.DNr = Påmelding.DNr
WHERE FNr = 1;
```

1-d

```
SELECT FNr, COUNT(*) AS Antall
FROM Påmelding
WHERE YEAR(RegDato) = 2023
GROUP BY FNr;
```

1-e

Endringer som må gjøres i SQL-skriptet:

- Fornavn NOT NULL
- Etternavn NOT NULL
- EPost NOT NULL UNIQUE
- CHECK (Turtype IN ('Enkel', 'Middels', 'Krevende'))
- CHECK (Pris BETWEEN 0 AND 50000)

1-f

```
SELECT *
FROM Deltaker
WHERE DNr NOT IN (SELECT DISTINCT DNr FROM Påmelding);
```

1-g

```
CREATE ROLE 'turleder';
GRANT SELECT ON * TO 'turleder';
GRANT INSERT, UPDATE, DELETE ON Fottur TO 'turleder';
CREATE USER 'kari' IDENTIFIED BY 'hemmelig';
GRANT 'turleder' to 'kari';
```

1-h

```
CREATE VIEW PåmeldingDenneMnd AS
SELECT P.*, D.Fornavn, D.Etternavn, F.StartDato
FROM Påmelding AS P
INNER JOIN Deltaker AS D ON P.DNr = D.DNr
INNER JOIN Fottur AS F ON P.FNr = F.FNr
WHERE YEAR(RegDato) = YEAR(CURDATE())
AND MONTH(RegDato) = MONTH(CURDATE());
```

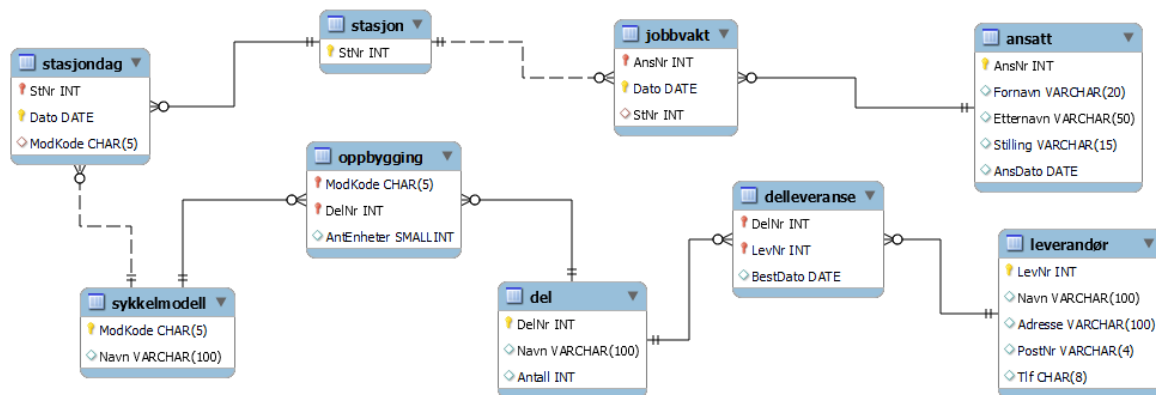
## Oppgave 2

Besvarelser på oppgaver i datamodellering blir vurdert ut fra flere og sammensatte kriterier. Det er for det første snakk om å få med alle momentene som blir nevnt i oppgaveteksten, men man skal på den andre siden heller ikke ta med unødvendige ting. Videre er det viktig at man får samlet de ulike opplysningene (attributtene/kolonnene) med fornuftige datatyper i hensiktsmessige entiteter (tabeller) og får opprettet forhold mellom entiteter der det er hensiktsmessig. Det er samtidig viktig å mestre de «tekniske» sidene ved ER: alle entiteter bør ha en fornuftig primærnøkkel, man bør ta hensiktsmessige valg når det gjelder en-til-mange, en-til-en og mange-til-mange forhold samt når det gjelder identifiserende og ikke-identifiserende forhold. Attributter bør gis naturlige datatyper og være definert med **NOT NULL** der det er aktuelt. Blant annet.

Det er vanskelig å beskrive presist hva som skal til for at en datamodell skal bli vurdert til f.eks. en A eller C eller F, for noen besvarelser kan ha med nesten alle momentene fra oppgaveteksten, men inneholde flere store, tekniske feil, mens andre besvarelser kan være teknisk sett gode, men ikke ha med mange av problemstillingene. Noen ganger kan det være krevende å tolke datamodeller riktig (slik det var ment), fordi ulike personer (studenter) velger forskjellige ord for å beskrive samme ting. Som regel vil det dessuten finnes flere, ulike og fullgode løsninger på en gitt datamodelleringsoppgave.

Gitt alle disse forbeholdene: For å få uttelling tilsvarende karakteren C bør man i hovedsak mestre de tekniske sidene ved ER og også ha med de viktigste momentene. Men man kan altså gjøre noen feil, kanskje til og med en grov feil som f.eks. å «snu» et en-til-mange forhold – hvis man gjort dette riktig andre steder i modellen. For å få karakteren A bør (så å si) alle momentene være med og det bør kun være små tekniske feil. For å få karakteren E kan det være flere grove feil og store mangler, men man må likevel levere et *ER-diagram* som viser entiteter, attributter og forhold for den aktuelle problemstillingen.

Et løsningsforslag er vist under.



### Oppgave 3

Vi ser altså på følgende tabell:

- HytteUtvidet(HNr, Navn, Telefon, AntSenger, PNr, PNavn, AntEnheter)

Tabellen inneholder redundans (dobbeltlagring). Dette er generelt uheldig fordi det sløser med lagringsplass og kan ha uheldige konsekvenser når man skal gjøre endringer (oppdateringsanomalier).

Eksempler på redundans:

- For gjentatte forekomster av samme hyttenummer blir navn, telefonnummer og antall senger for denne hytta gjentatt.
- For gjentatte forekomster av samme produktnummer blir navnet på produktet gjentatt.

Følgende funksjonelle avhengigheter gjelder:

- HNr → Navn, Telefon, AntSenger
- PNr → PNavn
- HNr, PNr → AntEnheter

Kandidatnøkkel: HNr + PNr

Den siste avhengigheten starter i kandidatnøkkel, så den kan vi se bort fra.

Vi splitter tabellen med hensyn på den første avhengigheten og får:

- T1(HNr, PNr, PNavn, AntEnheter)
- Hytte(HNr, Navn, Telefon, AntSenger)

Vi splitter T1 videre med hensyn på den andre avhengigheten og får sluttresultatet – med primærnøkler og fremmednøkler:

- Hytte(HNr, Navn, Telefon, AntSenger)
- Produkt(PNr, PNavn)
- Lager(HNr\*, PNr\*, AntEnheter)

#### Oppgave 4

Korrekte alternativ: cadbc bdccb aadba