

Løsningsforslag 6102/6102N Databaser høst 2016

Oppgave 1

-- 1a

```
SELECT *
FROM Tur
WHERE Pris < 8000 AND MONTH(StartDato) = 7
ORDER BY Pris, StartDato;
```

-- 1b

```
SELECT DISTINCT Medlem.*
FROM Medlem, Påmelding, Tur
WHERE Medlem.MNr = Påmelding.MNr
AND Tur.TNr = Påmelding.TNr
AND Tur.StartHytte = 2;
```

-- 1c

```
SELECT Tur.TNr, Tur.Beskrivelse, Tur.StartDato, COUNT(*)
FROM Tur, Påmelding
WHERE Tur.TNr = Påmelding.TNr
GROUP BY Tur.TNr, Tur.Beskrivelse, Tur.StartDato;
```

-- 1d

```
CREATE TABLE Hytte
(
  HNr          INT AUTO_INCREMENT,
  Navn         VARCHAR(50) NOT NULL,
  AntSenger   INT,
  HytteType   VARCHAR(20),
  CONSTRAINT HyttePK PRIMARY KEY (HNr)
) ENGINE=INNODB;
```

-- For å sikre innhold i HytteType:

-- a) Lage hjelpetabell for hyttetype med fremmednøkkel.

-- b) Legge på CHECK-regel

-- c) Lage trigger

-- d) (Bruke enum, ikke forventet)

```

-- 1e

START TRANSACTION;

INSERT INTO
  Medlem(Fornavn, Etternavn, Tlf)
VALUES
  ('Kari', 'Li', '55664477');

INSERT INTO
  Påmelding(TNr, MNr)
VALUES
  (3, LAST_INSERT_ID());

COMMIT;

-- 1f

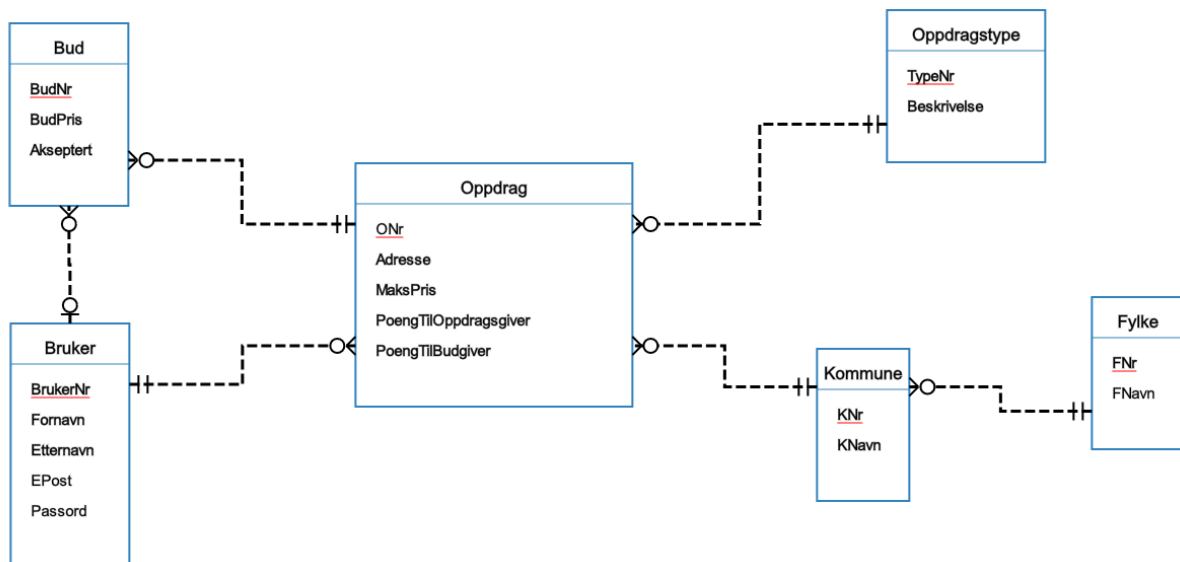
SELECT *
FROM Medlem
WHERE MNr NOT IN (SELECT MNr FROM Påmelding);

```

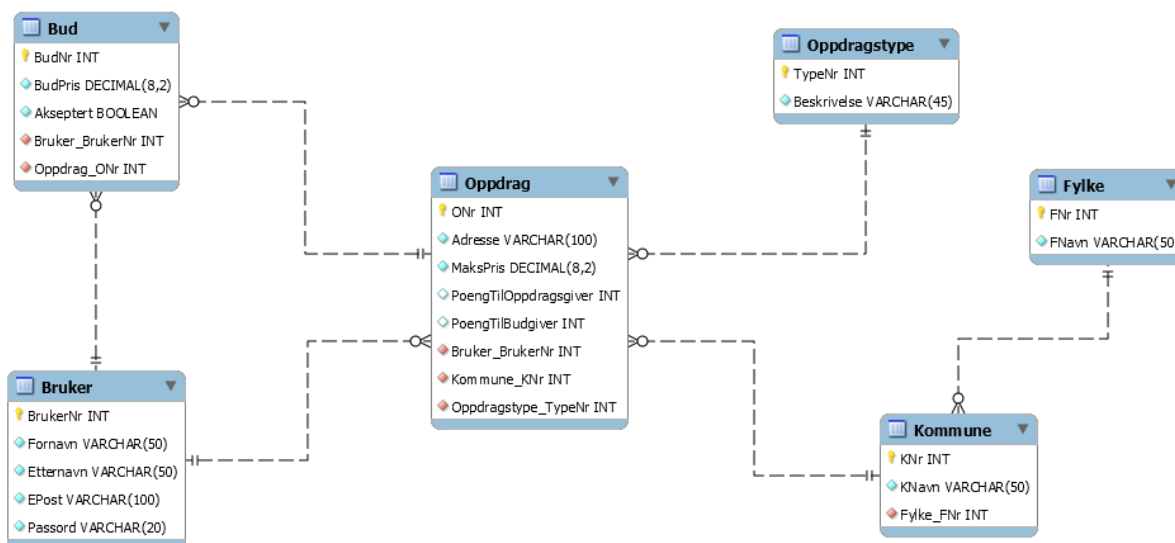
Oppgave 2

Man kunne velge å tegne datamodellen som vist i læreboken eller som i MySQL Workbench.

Begrepsmessig datamodell som tegnet i læreboken (kapittel 7):

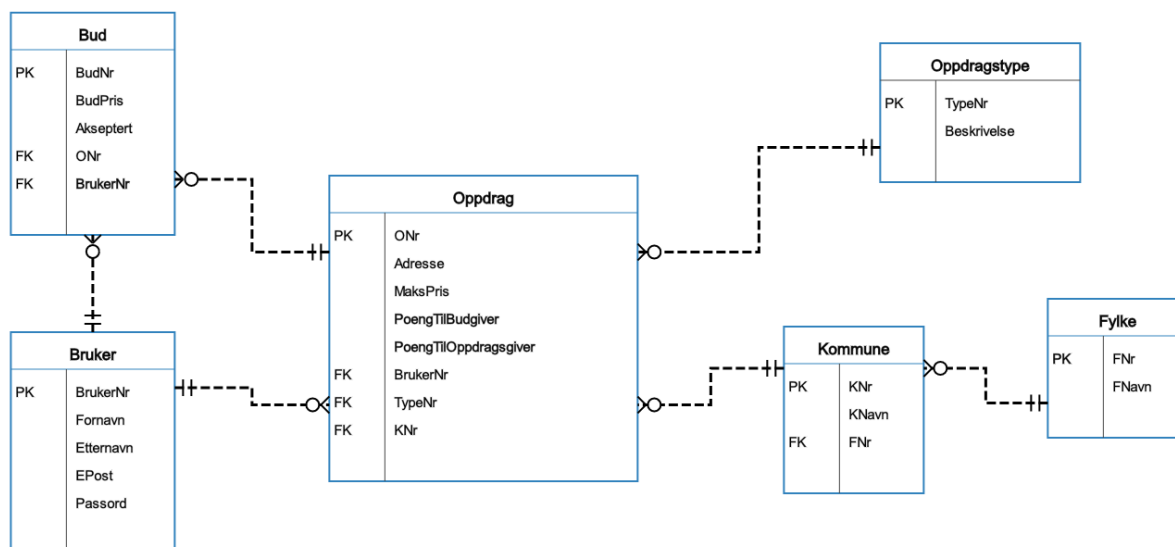


Logisk datamodell som tegnet i MySQL Workbench:



Det var ikke nødvendig å ta med datatyper og det var heller ikke noe krav å vise hvilke attributter som må fylles ut (diamanter med og uten fyll i MySQL Workbench). For å forenkle håndtegnning på papir (uten tilgang på fargekoding) kunne man symbolisere primærnøkler med understreking og merke fremmednøkler med stjerner.

En stilisert utgave av MySQL Workbench sine datamodeller svarer for øvrig til logiske datamodeller som tegnet i læreboken (kapittel 8):



Oppgave 3

3-a

Problemet med tabellen TurUtvidet er at den fører til redundans, eller dobbeltlagring. For en tur som er innom 4 hytter, så vil man registrere beskrivelsen og prisen 4 ganger. Dette øker mulighetene for feilregistrering og kan gjøre databasen inkonsistent, og det er også sløsing med lagringsplass.

Funksjonelle avhengigheter:

- TNr → Beskrivelse
- TNr → Pris
- TNr + Dato → Hytte

Kandidatnøkkel: TNr + Dato

Det betyr at tabeller bryter med 2NF på grunn av de to avhengighetene fra TNr. Ved å skille ut TNr, Beskrivelse og Pris i en egen tabell, så får vi:

- Tur(TNr, Beskrivelse, Pris)
- Rute(TNr*, Dato, Hytte)

3-a

Løsningsforslag er ikke laget.

3-b

Løsningsforslag er ikke laget.

Oppgave 4

bccbc – cdbbb – adbdc – dbabc

Vedlegg. SQL-skript til oppgave 1

Skriptet er kun tatt med for å lette testing og var ikke en del av oppgaven.

```
-- Tabellstruktur:
-- Medlem(MNr, Fornavn, Etternavn, Tlf)
-- Hytte(HNr, Navn, AntSenger, HytteType)
-- Tur(TNr, Beskrivelse, Pris, StartDato, StartHytte*)
-- Påmelding(TNr*, MNr*)

DROP TABLE IF EXISTS Påmelding;
DROP TABLE IF EXISTS Tur;
DROP TABLE IF EXISTS Hytte;
DROP TABLE IF EXISTS Medlem;

CREATE TABLE Medlem
(
  MNr          INT AUTO_INCREMENT,
  Fornavn     VARCHAR(50) NOT NULL,
  Etternavn   VARCHAR(50) NOT NULL,
  Tlf         VARCHAR(20) NOT NULL,
  CONSTRAINT MedlemPK PRIMARY KEY (MNr)
) ENGINE=INNODB;

CREATE TABLE Hytte
(
  HNr          INT AUTO_INCREMENT,
  Navn        VARCHAR(50) NOT NULL,
  AntSenger   INT,
  HytteType   VARCHAR(20),
  CONSTRAINT HyttePK PRIMARY KEY (HNr)
) ENGINE=INNODB;

CREATE TABLE Tur
(
  TNr          INT AUTO_INCREMENT,
  Beskrivelse VARCHAR(255),
  Pris        DECIMAL(8, 2) NOT NULL,
  StartDato   DATE NOT NULL,
  StartHytte  INT NOT NULL,
  CONSTRAINT TurPK PRIMARY KEY (TNr),
  CONSTRAINT TurHytteFK FOREIGN KEY (StartHytte) REFERENCES Hytte
(HNr)
) ENGINE=INNODB;
```

```
CREATE TABLE Påmelding
(
    TNr          INT,
    MNr          INT,
    CONSTRAINT PåmeldingPK PRIMARY KEY (TNr, MNr),
    CONSTRAINT PåmeldingMedlemFK FOREIGN KEY (MNr) REFERENCES Medlem
(MNr),
    CONSTRAINT PåmeldingTurFK FOREIGN KEY (TNr) REFERENCES Tur (TNr)
) ENGINE=INNODB;
```

```
INSERT INTO
    Medlem(MNr, Fornavn, Etternavn, Tlf)
VALUES
    (1, 'Ola',      'Hansen', '12345678'),
    (2, 'Kari',    'Mo',      '87654321'),
    (3, 'Anette',  'Lien',    '22334455'),
    (4, 'Johan',   'Åsen',    '88776655');
```

```
INSERT INTO
    Hytte(HNr, Navn, AntSenger, HytteType)
VALUES
    (1, 'Furubu',      25, 'Betjent'),
    (2, 'Blåsbortstua', 40, 'Selvbetjent'),
    (3, 'Steinbua',    10, 'Ubetjent');
```

```
INSERT INTO
    Tur(TNr, Beskrivelse, Pris, StartDato, StartHytte)
VALUES
    (1, 'Krevende topptur', 7500, '2017-04-27', 2),
    (2, 'Kort familietur', 4200, '2017-07-28', 1),
    (3, 'Brevandring',     9400, '2017-08-04', 2),
    (4, 'Klassisk fjelltur', 6500, '2017-08-11', 1);
```

```
INSERT INTO
    Påmelding(TNr, MNr)
VALUES
    (1, 2),
    (1, 3),
    (2, 3),
    (3, 1),
    (4, 1),
    (4, 2),
    (4, 3);
```

```
COMMIT;
```