Universitetet i Sørøst-Norge

Sensorveiledning

Emnekode	Emnenavn	Eksamensdato	Campus	Faglærers navn
DAT1000	Database 1	06.12.2024	Bakkenteigen	Bjørn Kristoffersen

Oppgave 1

1-a

```
SELECT KNr, Fornavn, Etternavn FROM Kunde
WHERE UPPER(Fornavn) LIKE 'O%' OR UPPER(Etternavn) LIKE 'O%';
```

1-b

```
SELECT * FROM Bolig
WHERE Kvm BETWEEN 80 AND 100
ORDER BY Adresse;
```

1-c

```
SELECT A.*, B.Adresse, K.Fornavn, K.Etternavn
FROM Avtale AS A
INNER JOIN Bolig AS B ON A.BNr = B.BNr
INNER JOIN Kunde AS K ON B.KNr = K.KNr
WHERE A.Ukedag = 'onsdag';
```

1-d

```
SELECT KNr, COUNT(*) AS AntallBoliger
FROM Bolig
GROUP BY KNr
HAVING COUNT(*) > 1;
```

1-e

Endringer i tabelldefinisjonene for Bolig og Kunde:

```
-- Adressen til boliger må alltid fylles ut.
Adresse VARCHAR(100) NOT NULL

-- Mobilnummer til kunder skal være unikt.
Mobil VARCHAR(15) UNIQUE

-- Antall kvadratmeter skal være et positivt tall.
CONSTRAINT KvmSjekk CHECK (Kvm > 0)
```

1-f

```
CREATE USER 'bruker' IDENTIFIED BY 'passord';
GRANT SELECT ON eksamen.Avtale TO 'bruker';
GRANT SELECT ON eksamen.Avtaletype TO 'bruker';
```

1-g

```
SELECT *
FROM Kunde
WHERE KNr NOT IN
  (
    SELECT DISTINCT KNr
    FROM Bolig AS B
    INNER JOIN Avtale AS A ON B.BNr = A.BNr
  );
```

1-h

```
CREATE VIEW
  AvtaleView
AS
  SELECT A.ANr, K.Fornavn, K.Etternavn, T.Beskrivelse,
    B.Kvm*T.KvmPris*A.DagerPrMnd AS Månedspris
  FROM Avtale AS A
  INNER JOIN Bolig AS B ON A.BNr = B.BNr
  INNER JOIN Kunde AS K ON B.KNr = K.KNr
  INNER JOIN Avtaletype AS T ON A.TNr = T.TNr;
```

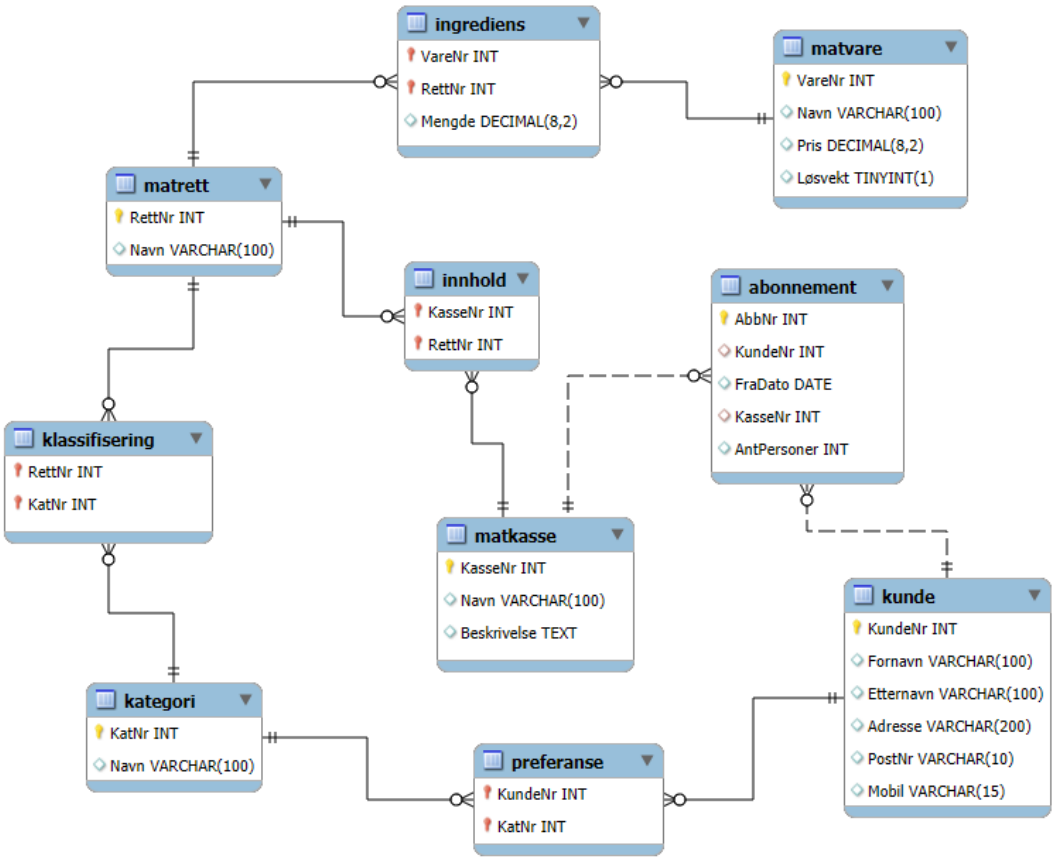
Oppgave 2

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:

- Turisme(KNr, KNavn, ANr, ANavn, PID, PNavn, Dato)

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 kommune blir kommunenavnet gjentatt.
- For gjentatte forekomster av samme attraksjon blir navnet på attraksjonen og kommunen den ligger i gjentatt.
- For gjentatte forekomster av samme person blir personens navn gjentatt.

Følgende funksjonelle avhengigheter gjelder:

- KNr $\rightarrow$ KNavn
- ANr $\rightarrow$ ANavn, KNr
- PID $\rightarrow$ PNavn

Kandidatnøkkel: ANr + PID + Dato

Vi splitter tabellen med hensyn på den første avhengigheten KNr → KNavn:

- Kommune(KNr, KNavn)
- Turisme2(ANr, ANavn, KNr, PID, PNavn, Dato)

Vi splitter Turisme2 med hensyn på avhengigheten ANr → ANavn, KNr:

- Kommune(KNr, KNavn)
- Attraksjon(ANr, ANavn, KNr)
- Turisme3(ANr, PID, PNavn, Dato)

Vi splitter Turisme3 med hensyn på avhengigheten PID → PNavn og får sluttresultatet med primærnøkler vist med understreking og fremmednøkler merket med en stjerne:

- Kommune(KNr, KNavn)
- Attraksjon(ANr, ANavn, KNr*)
- Person(PID, PNavn)
- Besøk(ANr*, PID*, Dato)

Man kan argumentere for at PNavn inneholder ikke-atomære verdier og splitte denne kolonnen i Fornavn og Etternavn, noe som vil gi en alternativ definisjon av tabell Person:

- Person(PID, Fornavn, Etternavn)

Begge løsninger gir full uttelling.

Oppgave 4

Korrekte alternativ: ccbcb aadbd baccd