

Eksamen høst 2015 – løsningsforslag

Oppgave 1

1a

```
SELECT KundeNr, RegNr
FROM Forsikring
WHERE KmPrAar > 20000 AND ForsType = 'Kasko'
ORDER BY KundeNr;
```

1b

```
SELECT ForsType AS Forsikringtype, COUNT(*) AS Antall
FROM Forsikring
GROUP BY ForsType;
```

1c

```
SELECT F.ForsNr, COUNT(S.SaksNr) AS Antall
FROM Forsikring AS F LEFT OUTER JOIN Skadesak AS S
    ON F.ForsNr = S.ForsNr
GROUP BY F.ForsNr;
```

1d

```
CREATE TABLE Forsikring
(
    ForsNr      INT AUTO_INCREMENT,
    KundeNr     INT NOT NULL,
    RegNr       CHAR(7) UNIQUE NOT NULL,
    RegAar      INT NOT NULL,
    KmPrAar     INT NOT NULL,
    ForsType    VARCHAR(10) NOT NULL,
    Bonus       INT,
    AarsPremie  DECIMAL(8, 2),
    CONSTRAINT ForsikringPK PRIMARY KEY (ForsNr),
    CONSTRAINT ForsikringKundeFK FOREIGN KEY (KundeNr)
        REFERENCES Kunde(KundeNr)
) ENGINE=INNODB;
```

Fire alternative teknikker for å sikre at Forstype og Bonus inneholder kun de angitt lovlige verdiene (det var ikke noe krav om å skrive kode på dette punktet):

- Bruk en CHECK-regel (MySQL godtar syntaksen, men sjekker ikke).
- Lag en hjelpetabell med lovlige verdier og opprett fremmednøkkel fra Forsikring (særlig Forstype egner seg godt for denne løsningen).
- Bruk en trigger (greit alternativ til CHECK i MySQL).
- For Forstype kan man også bruke ENUM som datatype, men dette er ikke pensum.

1e

```
START TRANSACTION;

INSERT INTO
  Skadesak(ForsNr, RegDato, SkadeType)
VALUES
  (2, CURDATE(), 'Kollisjon');

UPDATE Forsikring
SET Bonus = Bonus - 10
WHERE ForsNr = 2;

COMMIT;
```

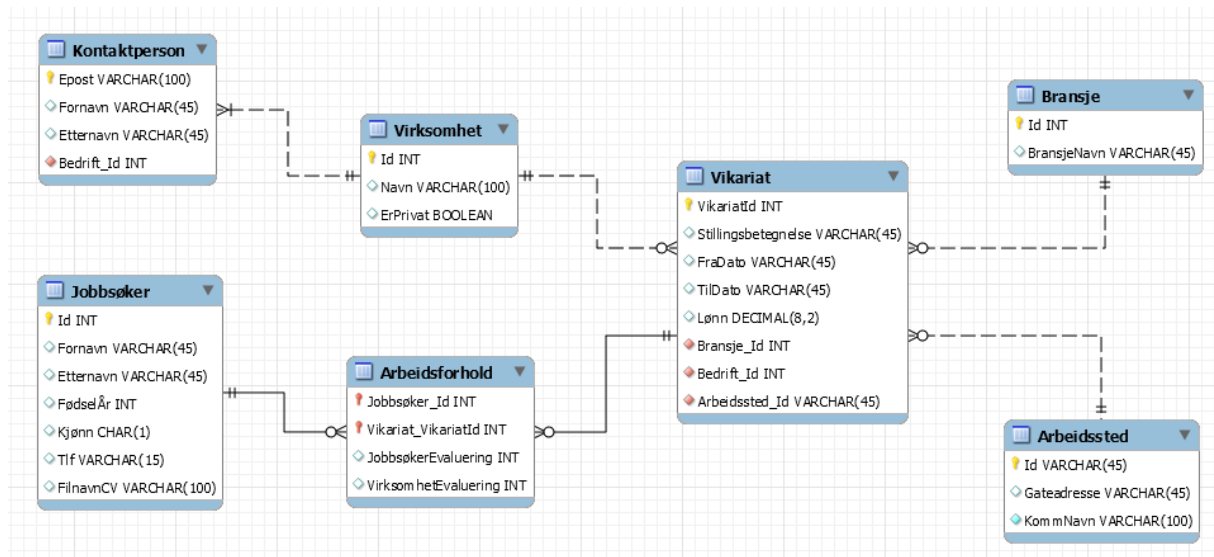
1f

```
DROP VIEW IF EXISTS NyeSkadesaker;

CREATE VIEW NyeSkadesaker AS
  SELECT S.*, K.*, ForsType, Bonus
  FROM Kunde AS K, Forsikring AS F, Skadesak AS Skadesak
  WHERE K.KundeNr = F.KundeNr
  AND F.ForsNr = S.ForsNr
  AND YEAR(RegDato) = YEAR(CURDATE());
```

Oppgave 2

NB! Diagrammet under kan forenkles – det er tilstrekkelig med et en-til-mange forhold mellom Jobbsøker og Vikariat (et vikariat blir kun besatt av én jobbsøker). Dermed forsvinner behovet for koblingsentiteten Arbeidsforhold: Attributtene JobbsøkerEvaluerings og VirksomhetEvaluerings kan legges inn i Vikariat. Noen velger å plassere disse i en egen entitet Evaluering som kobles til Vikariat med et en-til-en forhold, det er også helt ok.



3-a

Begrunnelse: Foreslått løsning er uheldig fordi kommunenavn blir gjentatt for alle kunder som bor i samme kommune, og dessuten blir fylkesnavn gjentatt for alle kunder som tilhører samme kommune.

Funksjonelle avhengigheter:

- KundeNr → ALLE KOLONNER
- KoNr → KoNavn
- KoNr → FyNr
- FyNr → FyNavn

Kandidatnøkkel: KundeNr

Resultat:

- Kunde(KundeNr, FDato, Fornavn, Etternavn, KoNr*)
- Kommune(KoNr, KoNavn, FyNr*)
- Fylke(FyNr, FyNavn)

Oppgave 3-b

Hva er en transaksjon: En logisk og kanskje sammensatt operasjon mot databasen. Kan være naturlig å ta med et eksempel, og nevne kommandoene START TRANSACTION, COMMIT og ROLLBACK. Noen vil kanskje også liste opp ACID-egenskapene.

Mekanismer for å håndtere samtidige brukere: Låser. Kan være naturlig å nevne problemer som kan oppstå i et «naivt» system, f.eks. tapt oppdatering (lost update). Man bør forklare hva en lås er, nevne skrive- og leselåser, samt tofaselåsing (en transaksjon kan ikke sette en ny lås etter at første lås er frigitt) for å sikre serialiserbarhet (korrekt resultat) og vranglås. Kanskje også optimistisk låsing.

Oppgave 4

abbcd dcbcb aacbc dbacb

SQL-skript for oppgave 1

NB! Skriptet er ikke en del av besvarelsen, og er kun tatt med for de som ønsker å teste løsningene.

```
DROP TABLE IF EXISTS Skadesak;
DROP TABLE IF EXISTS Forsikring;
DROP TABLE IF EXISTS Kunde;

CREATE TABLE Kunde
(
  KundeNr      INT AUTO_INCREMENT,
  FDate        DATE,
  Fornavn      VARCHAR(30),
  Etternavn    VARCHAR(50),
  CONSTRAINT KundePK PRIMARY KEY (KundeNr)
) ENGINE=INNODB;

CREATE TABLE Forsikring
(
  ForsNr       INT AUTO_INCREMENT,
  KundeNr      INT NOT NULL,
  RegNr        CHAR(7) UNIQUE NOT NULL,
  RegAar       INT NOT NULL,
  KmPrAar      INT NOT NULL,
  ForsType     VARCHAR(10) NOT NULL,
  Bonus        INT,
  AarsPremie   DECIMAL(8, 2),
  CONSTRAINT ForsikringPK PRIMARY KEY (ForsNr),
  CONSTRAINT ForsikringKundeFK FOREIGN KEY (KundeNr)
    REFERENCES Kunde (KundeNr)
) ENGINE=INNODB;
```

```

CREATE TABLE Skadesak
(
    SaksNr          INT AUTO_INCREMENT,
    ForsNr          INT NOT NULL,
    RegDato         DATE,
    SkadeType       VARCHAR(30),
    CONSTRAINT SkadesakPK PRIMARY KEY (SaksNr),
    CONSTRAINT SkadesakForsikringFK FOREIGN KEY (ForsNr)
        REFERENCES Forsikring(ForsNr)
) ENGINE=INNODB;

```

```

INSERT INTO
    Kunde(FDato, Fornavn, Etternavn)
VALUES
    ('1988-07-17', 'Ola',    'Hansen'),
    ('1991-11-04', 'Kari',   'Mo'),
    ('1970-02-28', 'Anette', 'Lien');

```

```

INSERT INTO
    Forsikring(KundeNr, RegNr, RegAar, KmPrAar, ForsType, Bonus, AarsPremie)
VALUES
    (1, 'LY12345', 2012, 22000, 'Kasko',    50, 4500.00),
    (1, 'AD33445', 1997, 11000, 'Ansvar',  70, 2300.00),
    (2, 'KZ99887', 2010, 18000, 'Kasko',    20, 7800.00),
    (3, 'PN29298', 2010, 25000, 'Delkasko', 50, 5200.00);

```

```

INSERT INTO
    Skadesak(ForsNr, RegDato, SkadeType)
VALUES
    (1, '2014-02-10', 'Kollisjon'),
    (2, '2014-12-16', 'Brannskade'),
    (2, '2015-07-30', 'Kollisjon'),
    (2, '2015-10-01', 'Tyveri'),
    (3, '2015-10-30', 'Glassrute');

```