Løsningsforslag DAT1000B Database 1 – vår 2018

Oppgave 1

SQL-skript ligger i vedlegg nederst.

1a

```
SELECT KursNr, Dato, Romkode
FROM Eksamen
WHERE Year(Dato) = 2017 AND Month(Dato) = 12
ORDER BY Dato, KursNr;
```

1b

```
SELECT DISTINCT Student.*
FROM Student, Sensur, Eksamen
WHERE Student.StudNr = Sensur.StudNr
AND Sensur.EksNr = Eksamen.EksNr
AND Eksamen.KursNr = 101
AND Sensur.Karakter <> 'F';
```

1c

```
CREATE TABLE Sensur
(
  EksNr    INTEGER,
  StudNr   INTEGER,
  Karakter CHAR(1) NOT NULL,
  PRIMARY KEY (EksNr, StudNr),
  FOREIGN KEY (EksNr) REFERENCES Eksamen (EksNr),
  FOREIGN KEY (StudNr) REFERENCES Student (StudNr)
);
```

- Teknikk 1: Fremmednøkkel mot tabell med lovlige bokstavkarakterer.
- Teknikk 2: CHECK-regel
- Teknikk 3: TRIGGER (ikke pensum)
- Teknikk 4: ENUM datatype (ikke pensum)

1d

```
UPDATE Eksamen
SET Romkode = '5-225'
WHERE Romkode = '5-224'
AND Dato > CURDATE();
```

1e

```
SELECT KursNr, UPPER(Navn) AS KursNavn, Studiepoeng
FROM Kurs
WHERE KursNr NOT IN (SELECT KursNr FROM Eksamen);
```

1f

```
SELECT Kurs.KursNr, Navn, COUNT(*) AS AntallStudenter
FROM Kurs, Eksamen, Sensur
WHERE Kurs.KursNr = Eksamen.KursNr
AND Sensur.EksNr = Eksamen.EksNr
AND YEAR(Eksamen.Dato) = 2017
GROUP BY Kurs.KursNr, Navn
HAVING COUNT(*) > 50;
```

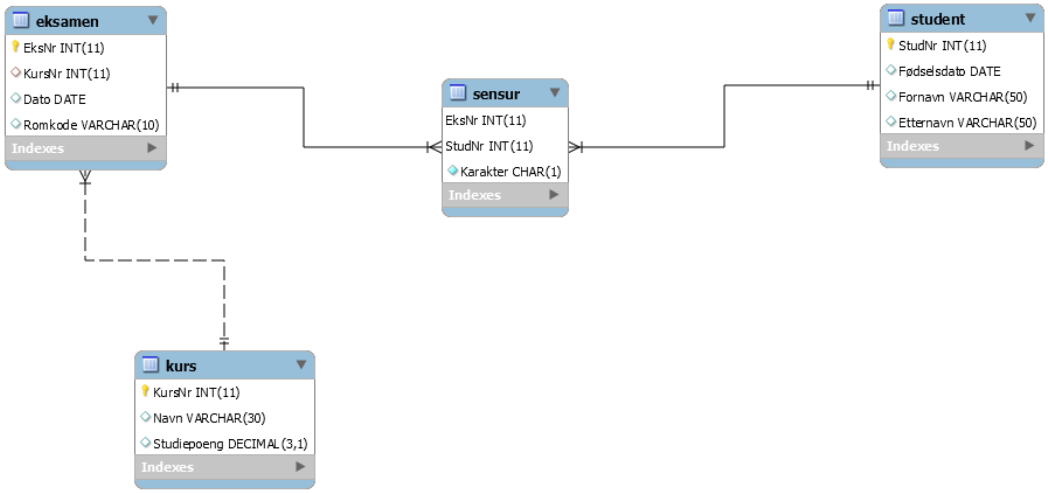
1g

```
CREATE VIEW
  Karakterutskrift(StudNr, Fornavn, Etternavn, KursNr, Karakter)
AS
  SELECT Student.StudNr, Fornavn, Etternavn, Kurs.KursNr, MIN(Karakter)
  FROM Student, Kurs, Eksamen, Sensur
  WHERE Student.StudNr = Sensur.StudNr
  AND Kurs.KursNr = Eksamen.KursNr
  AND Sensur.EksNr = Eksamen.EksNr
  GROUP BY Student.StudNr, Fornavn, Etternavn, Kurs.KursNr;
```

Oppgave 2

2a

En logisk datamodell for databasen i oppgave 1 kan man lage ved å kjøre «reverse engineering» i MySQL Workbench. Resultatet blir slik:

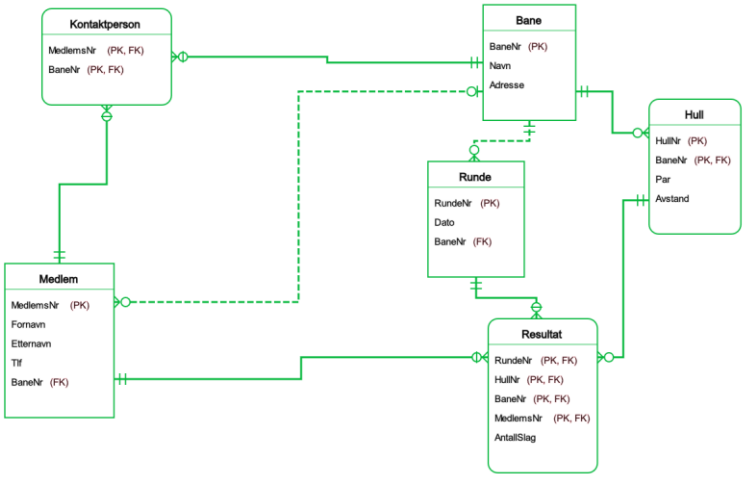


2b

Her skulle man lage en logisk datamodell. Våren 2018 ble det i første omgang kun skissert en løsning i form av følgende tabellstruktur, der understreking viser primærnøkler og stjerner viser fremmednøkler:

- Bane(BaneNr, Navn, Adresse, PostNr)
- Hull(BaneNr*, HullNr, AntallSlagPar, AntallMeter)
- Spiller(SpillerNr, Fornavn, Etternavn, Tlf, BaneNr*)
- Kontakt(BaneNr*, SpillerNr*)
- Runde(RundeNr, BaneNr*, Dato)
- Resultat(SpillerNr*, RundeNr*, HullNr, AntallSlag)
- HandicapHistorikk(SpillerNr*, Dato, Score)

Seinere er denne oppgaven lagt inn i den pedagogiske applikasjonen LearnER, sammen med et løsningsforslag – **jeg mener at oppgaveteksten skal være lik**. Både oppgavetekst og løsning fra LearnER er limt inn under.




Oppgave SQL

Du skal nå lage en datamodell til bruk for Golfforbundet i Utopia basert på beskrivelsen under.

Det finnes et antall golfbaner i Utopia. Om hver bane skal det lagres et unikt banenummer, et navn og en adresse. En golfbane kan ha mange hull, det vanligste er 9 eller 18. Hullene på en bane er nummerert fra 1 og oppover. Med «hull» menes her ikke bare hullet i bakken der golfballen skal slås oppi, men hele området fra utslagspunktet til putteområdet, det som på engelsk kalles for putting green (området med kortklipt gress). Hullets par er definert som det antall slag en god spiller skal behøve for å slå ballen i hullet. Om hvert hull skal både hullets par og avstanden fra utslagspunktet til midten av putteområdet lagres.

Databasen skal lagre fornavn, etternavn og telefonnummer til alle spillere. For å spille golf i Utopia må man være medlem i forbundet. En spiller er alltid tilknyttet én bestemt bane. Til hver bane blir en eller flere spillere utpekt som kontaktpersoner for denne banen.

En (liten) gruppe med spillere kan gå sammen for å spille en golfrunde. En golfrunde gjennomføres i løpet av én dag, slik reglene er i Utopia. En spiller kan over tid spille mange golfrunder med forskjellige medspillere. På en runde vil spillergruppen gå gjennom hele golfbanen og spille på alle hullene etter tur. For hvert hull registrerer man hvor mange slag hver av spillerne i gruppen måtte bruke for å få ballen i hullet.

En spiller har til enhver tid et såkalt golfhandicap. Dette er et tall som forteller noe om spillestyrken til vedkommende, altså hvor god denne spilleren er. Resultatene for en golfrunde må alltid registreres samme dag for å være gyldige. På slutten av hver dag så blir handicappet til spillerne oppdatert, i henhold til bestemte regler som vi ikke går nærmere inn på her. Systemet skal ta vare på handicaphistorikken til spillerne, dvs. at man fra opplysningene i databasen skal kunne avlese hva handicappet til en spiller var på slutten av en hvilken som helst dag.

Oppgave 3

3a

Eksempeltabell:

- Eksamen(EksNr, KursNr, Dato, Romkode, AntallPlasser, AnsattNr, Fornavn, Etternavn)

Tabellen er uheldig fordi den medfører redundans (dobbeltlagring): Alle rader med samme Romkode vil også ha samme verdi i AntallPlasser, og alle rader ned samme AnsattNr vil ha samme verdi i både Fornavn og Etternavn.

Funksjonelle avhengigheter:

- EksNr -> ALLE KOLONNER
- Romkode -> AntallPlasser
- AnsattNr -> Fornavn, Etternavn

Kandidatnøkkel: EksNr

Den første avhengigheten starter i kandidatnøkkelen (EksNr), så den kan vi se bort fra.

De to andre avhengighetene bryter med 3NF. Vi starter med avhengigheten fra Romkode:

- Eksamen(EksNr, KursNr, Dato, Romkode, AnsattNr, Fornavn, Etternavn)
- Rom(Romkode, AntallPlasser)

Deretter behandler vi avhengigheten fra AnsattNr, som gir resultatet:

- Eksamen(EksNr, KursNr, Dato, Romkode*, AnsattNr*)
- Rom(Romkode, AntallPlasser)
- Ansatt(AnsattNr, Fornavn, Etternavn)

Primærnøkler er understreket og fremmednøkler er merket med stjerne.

Oppgave 4

Korrekte svar på spørsmål 1–15: adcab ddccc cdcbc

Vedlegg: SQL-skript til opgave 1

```
DROP TABLE IF EXISTS Sensur;
DROP TABLE IF EXISTS Eksamen;
DROP TABLE IF EXISTS Kurs;
DROP TABLE IF EXISTS Student;

CREATE TABLE Student
(
 StudNr      INTEGER,
 Fødselsdato DATE,
 Fornavn     VARCHAR(50),
 Etternavn   VARCHAR(50),
 PRIMARY KEY (StudNr)
);

CREATE TABLE Kurs
(
 KursNr      INTEGER,
 Navn        VARCHAR(30),
 Studiepoeng DECIMAL(3,1),
 PRIMARY KEY (KursNr)
);

CREATE TABLE Eksamen
(
 EksNr   INTEGER,
 KursNr  INTEGER,
 Dato    DATE,
 Romkode VARCHAR(10),
```

```
  PRIMARY KEY (EksNr),
  FOREIGN KEY (KursNr) REFERENCES Kurs (KursNr)
);



CREATE TABLE Sensur
(
  EksNr    INTEGER,
  StudNr   INTEGER,
  Karakter CHAR(1) NOT NULL,
  PRIMARY KEY (EksNr, StudNr),
  FOREIGN KEY (EksNr) REFERENCES Eksamen (EksNr),
  FOREIGN KEY (StudNr) REFERENCES Student (StudNr)
);

INSERT INTO
  Student(StudNr, Fødselsdato, Fornavn, Etternavn)
VALUES
  (1, '1994-05-17', 'Peder', 'Aas'),
  (2, '2000-11-07', 'Kari',  'Moan'),
  (3, '1998-01-29', 'Ane',   'Liane');

INSERT INTO
  Kurs(KursNr, Navn, Studiepoeng)
VALUES
  (101, 'Lineær algebra',      7.5),
  (102, 'Tysk grammatikk',     7.5),
  (103, 'Bacheloroppgave',    15.0),
  (104, 'Spesialpensum',       5.0),
  (105, 'Geografisk analyse', 10.0);

INSERT INTO
  Eksamen(EksNr, KursNr, Dato, Romkode)
VALUES
  (51, 101, '2016-05-13', '5-322'),
  (52, 102, '2017-05-15', '2-132'),
  (53, 101, '2017-05-23', '5-323'),
  (54, 104, '2017-12-12', '5-224'),
  (55, 105, '2017-12-15', 'Storsalen');

INSERT INTO
  Sensur(EksNr, StudNr, Karakter)
VALUES
  (51, 1, 'F'),
  (51, 2, 'C'),
  (51, 3, 'A'),
  (52, 1, 'E'),
```

```
(53, 1, 'B'),
(55, 2, 'C');
```