



Fag TDT4145 Datamodellering og databasesystemer

Øving 3: Relasjonsalgebra og SQL

Oppgave 1

(Oppgaven skal gjøres ved hjelp av **RelasJonsalgebra** som finnes på *It's Learning*. Last ned bokbase.zip og RelAlg.jar. Pakk ut bokbase.zip til samme katalogen hvor RelAlg.jar ligger. Bruk kommandolinje for å bytte til denne katalogen, så kjør **java -jar RelAlg.jar**. Når applikasjonen har startet, velger man **Database -> Connect** og skriver **student** som både brukernavn og passord og databasenavnet skrives om til **bokbase**. Applikasjonen skal da kjøre. Den bør være relativt grei å forstå, men dokumentasjon ligger uansett i Relalg_doc/relalg/index.no.html.

Filstrukturen i mappen skal altså se slik ut for å kunne kjøre RelAlg:

Navn	Dato endret	Type	Størrelse
bokbase	19.02.2015 10:59	Filmappe	
derby	19.02.2015 11:00	Tekstdokument	1 kB
RelAlg	18.02.2015 11:55	Executable Jar File	2 860 kB

Og i bokbase-mappen:

Navn	Dato endret	Type	Størrelse
log	18.02.2015 11:48	Filmappe	
seg0	18.02.2015 11:48	Filmappe	
tmp	19.02.2015 10:59	Filmappe	
.DS_Store	18.02.2015 11:48	DS_STORE-fil	7 kB
db	19.02.2015 10:59	LCK-fil	1 kB
dbex	18.02.2015 11:48	LCK-fil	1 kB
derby	19.02.2015 10:59	Tekstdokument	14 kB
service	18.02.2015 11:48	PROPERTIES-fil	1 kB

Databasen består av følgende tabeller (tabellinnholdet er vedlagt bakerst i øvingen):

Bok(Bokid, Tittel, Utgittår, Forlagid)

Forlag(Forlagid, Forlagnavn, Adresse, Telefon)

Forfatter(Forfatterid, Fornavn, Etternavn, Fødeår, Dødsår, Nasjonalitet)

BokForfatter(Bokid, Forfatterid)

Gjør deg kjent med databasen ved å se på innholdet i de fire tabellene. Det skal lages relasjonsalgebra for følgende spørringer. Husk å legge ved et skjermbilde av spørringer (ikke resultatet) i besvarelsen din.

- 1) Finn alle boktitler i databasen.
- 2) Finn alle opplysninger om norske forfattere.
- 3) Finn navn og telefonnummer for alle forlag i Oslo.
- 4) Finn tittel og forlagnavn for alle bøker.
- 5) Finn tittel på alle bøker som er gitt ut av Gyldendal.
- 6) Finn tittel og utgivelsesår for alle bøker som er skrevet av Knut Hamsun
- 7) Finn tittel og forfatternavn for alle bøker som er gitt ut på 90-tallet.
- 8) Finn tittel, forfatternavn og forlagnavn for alle bøker.
- 9) Finn gjennomsnittlig utgivelsesår for alle forlag.
- 10) Finn navnet på forlaget som har gitt ut den eldste boka.
- 11) Finn hvilke forlag som ikke har gitt ut noen bøker.

I oppgave 2 i denne øvingen skal man bruke Apache Derby (JavaDB) sin syntaks til å skrive SQL-spørringer. Offisiell syntaksbeskrivelse finnes her:
<http://db.apache.org/derby/docs/10.11/ref/refderby.pdf>

Oppgave 2

Gitt følgende relasjonskjema:

KUNDE(KUNDENR, NAVN, KREDITTGRENSE, POSTNR).

POSTSTED(POSTNR, POSTSTED).

BESTILLING(ARTNR, KUNDENR, KVANTUM).

ARTIKKEL(ARTNR, NAVN, ANT, PRIS).

- Skriv datadefinisjoner for disse tabellene i et SQL-skjema. Velg selv datatyper for attributtene. Ta også med spesifisering av primær-, kandidat- og fremmednøkler.
- Når en kunde eller artikkel slettes, skal tilsvarende bestillinger slettes og når et kunde- eller artikkelnummer endres i KUNDE eller ARTIKKEL, skal tilsvarende nummer endres i BESTILLING. Hvordan kan dette uttrykkes i SQL? Vis hvordan disse kravene kan inkluderes i definisjonene fra a). Hvilke regler er naturlig å bruke for postnummer?
- Vi ønsker å legge inne en begrensning på hvor mye en kunde kan bestille. Kravet er at samlet verdi på en kundes bestillinger ikke skal overstige kundens kredittgrense. Vis hvordan denne begrensningen kan spesifiseres i SQL.

Oppgave 3

I denne oppgaven bruker vi samme eksempeldatabase som i Oppgave 1. Du skal skrive en del spørringer for denne databasen. For å gjøre dette har du tre alternative fremgangsmåter. Du må altså velge en av disse.

- Bruke **SqlClient**. SqlClient lastes ned fra it's Learning og startes på samme måte som beskrevet i Oppgave 1. Lukk RelasjonsAlgebra før du starter SqlClient.
- Bruke databasen som ligger på NTNUs servere. Måter å koble til denne er beskrevet i MySQL.pdf. Legg merke til at du her må være på NTNUs nett. (Ikke Eduroam) Se <https://innsida.ntnu.no/wiki/-/wiki/English/Install+VPN> for oppsett av VPN for å koble deg til NTNUs interne nett.
- Installere en databaseserver på din egen datamaskin. For instruksjoner om hvordan du kan gjøre dette, se MySQL.pdf, pkt 3, og pkt 5.1. Så kan du sette opp bokdatabasen ved å kjøre sql-spørringene i bokbase.sql.

Oppgaver:

- Finn alle boktitler i databasen.
- Finn alle opplysninger om norske forfattere.
- Finn navn og telefonnr for alle forlag i Oslo, sortert alfabetisk på navn.
- Finn tittel og forlagnavn for alle bøker.
- Finn tittel og utgittår for alle bøker som er skrevet av Knut Hamsun.
- Finn alle forfattere som har etternavn som begynner på H. Skriv ut fornavn, etternavn og fødselsår.
- Finn ut hvor mange forlag det er i databasen.
- Finn tittel, forfatternavn og forlagnavn for alle bøker som er skrevet av britiske forfattere.

- i) Lag en liste over alle forfattere og hvor mange bøker hver har skrevet. Sorter lista etter synkende antall bøker.
- j) Finn tittel og utgittår for den eldste boka i databasen.
- k) List ut alle forlag som har gitt ut flere enn to bøker, og antall bøker de har gitt ut.
- l) Finn alle forlag som ikke har gitt ut noen bøker.

Oppgave 4

- a) Hva er hensikten med virtuelle tabeller (views)? Hvorfor kan vi få problemer ved oppdateringer gjennom virtuelle tabeller?
- b) Gitt tabellene i eksempeldatabasen vist til slutt i øvingen. Definer en virtuell tabell (view) over disse tabellene som inkluderer prosjektnavn, avdelings-tilhørighet, antall ansatte (på prosjektet) og totalt antall timer arbeidet per uke på hvert prosjekt.
- c) Gitt følgende view-definisjon:

```
CREATE VIEW DEPT_SUMMARY(D, C, TOTAL_S, AVERAGE_S)
AS SELECT DNO, COUNT(*), SUM(SALARY), AVG(SALARY)
FROM EMPLOYEE
GROUP BY DNO
```

Hvilke av spørringene og oppdateringene under er lovlige? Vis hvordan disse utføres på basis-tabellene.

- 1) **SELECT ***
FROM DEPT_SUMMARY
- 2) **SELECT D, C**
FROM DEPT_SUMMARY
WHERE TOTAL_S > 10000
- 3) **UPDATE DEPT_SUMMARY**
SET D = 3
WHERE D = 4
- 4) **DELETE**
FROM DEPT_SUMMARY
WHERE C > 4

Oppgave 5

Vi har en database for å registrere leverandører og deler. Vi vet hvor leverandørene er (Supplier.city) og hvor de har delene sine (Part.city) og hvor mange deler de har på hvert sted. Tabellen Part er en slags produktbeskrivelse over mulige deler en leverandør har på et sted, mens SuppliesPart forteller om de faktisk har den (qty > 0).

Denne databasen består av tre tabeller, som er definert i SQL som følger:

```
CREATE TABLE Supplier (
    sno INT NOT NULL PRIMARY KEY,
    sname CHAR(5),
    status INT,
    city CHAR(6));
```

```
CREATE TABLE Part (
    pno INT NOT NULL PRIMARY KEY,
    pname CHAR(5),
    color CHAR(5),
    weight DECIMAL(3,1),
    city CHAR(6));
```

```
CREATE TABLE SuppliesPart (
    sno INT NOT NULL,
```

```

pno INT NOT NULL,
qty INT,
PRIMARY KEY (sno, pno),
FOREIGN KEY (sno) REFERENCES Supplier(sno),
FOREIGN KEY (pno) REFERENCES Part(pno));

```

- Skriv et spørsmål i SQL som finner alle leverandører med status større enn 15.
- Skriv et spørsmål i SQL som finner navn og leverandørens by for leverandører som leverer deler med navn 'Screw'.
- Skriv et spørsmål i SQL som finner nummer (pno) og navn for deler som kommer fra flere (enn **1**) leverandører.
- Skriv et spørsmål i SQL som finner det totale antallet leverandører.
- Skriv et spørsmål i SQL som finner byene for leverandører som leverer deler med vekt (weight) større enn 10.0.
- Skriv et spørsmål i SQL som finner navn for leverandører som ikke leverer deler med navn 'Screw'. Resultatet skal være sortert og hvert navn skal opptre kun en gang i resultatet.

VEDLEGG

Bok-tabellen

BOKID	TITTEL	UTGITTÅR	FORLAGID
1	Tåpenes sammensvergelse	1995	7
2	Rebecca-koden	1981	3

3	Gutter er gutter	1998	5
4	Microserfs	1995	8
5	Generation X	1991	8
6	Klosterkrønike	1982	3
7	Univers uten grenser	1988	3
8	Nålen	1978	3
9	Markens grøde	1917	2
10	Victoria	1898	2
11	Sult	1890	2
12	Benoni	1908	2
13	Rosa	1908	2
14	Ett skritt etter	1997	2
15	Den femte kvinnen	1996	2
16	Villspor	1995	2
17	Silkeridderen	1994	2
18	Den hvite løvinnen	1993	2
19	Hundene i Riga	1992	2
20	Bridget Jones dagbok	1996	5
21	Sa terapeuten min	1998	3
22	Sa mor	1996	3
23	Jubel	1995	3
24	Tatt av kvinnen	1993	3
25	NAIV.SUPER.	1996	3

Forlag-tabellen

FORLAGID	FORLAGNAVN	ADRESSE	TELEFON
1	Tapir	Trondheim	73590000
2	Gyldendal	Oslo	22220000
3	Cappelen	Oslo	22200000
4	Universitetsforlaget	Oslo	23230000
5	Aschehoug	Oslo	22000000
6	Oktober	Oslo	22002200
7	Tiden	Oslo	22232223
8	HarperCollins	USA	

Forfatter-tabellen

FORFATTERI	FORNAVN	ETTERNAVN	FØDEÅR	DØDÅR	NASJONALIT
1	John Kennedy	Toole	1937	1969	USA
2	Ken	Follett			Britisk
3	Stephen W.	Hawking			Britisk
4	Jose	Saramago	1942		Portugisisk
5	Douglas	Coupland	1961		Canadisk
6	Nick	Hornby	1957		Britisk
7	Knut	Hamsun	1859	1952	Norsk
8	Henning	Mankell	1948		Svensk
9	Helen	Fielding			Britisk
10	Hal	Sirowitz			USA
11	Lars Saabye	Christensen			Norsk
12	Erlend	Loe			Norsk

BokForfatter-tabellen

BOKID	FORFATTERI
1	1
2	2
3	6
4	5
5	5
6	4
7	3
8	2
9	7
10	7
11	7
12	7
13	7
14	8
15	8
16	8
17	8
18	8
19	8
20	9
21	10
22	10
23	11
24	12
25	12

Eksempeldatabase (til oppgave 4)

```
CREATE TABLE EMPLOYEE
(FNAME      VARCHAR(15) NOT NULL,
 MINIT      CHAR(1),
 LNAME      VARCHAR(15) NOT NULL,
 SSN        CHAR(9) NOT NULL,
 BDATE      CHAR(9),
 ADDRESS    VARCHAR(30),
 SEX        CHAR(1),
 SALARY     INT,
 SUPERSSN   CHAR(9),
 DNO        INT NOT NULL,
 PRIMARY KEY (SSN),
 FOREIGN KEY (SUPERSSN) REFERENCES EMPLOYEE (SSN),
 FOREIGN KEY (DNO) REFERENCES DEPARTMENT (DNUMBER)
);

CREATE TABLE DEPARTMENT
(DNAME      VARCHAR(15) NOT NULL,
 DNUMBER    INT NOT NULL,
 MGRSSN     CHAR(9) NOT NULL,
 MGRSTARTDATE CHAR(9),
 PRIMARY KEY (DNUMBER),
 UNIQUE (DNAME),
 FOREIGN KEY (MGRSSN) REFERENCES EMPLOYEE (SSN)
);

CREATE TABLE DEPT_LOCATIONS
(DNUMBER    INT NOT NULL,
 DLOCATION   VARCHAR(15) NOT NULL,
 PRIMARY KEY (DNUMBER, DLOCATION),
 FOREIGN KEY (DNO) REFERENCES DEPARTMENT (DNUMBER)
);

CREATE TABLE PROJECT
(PNAME      VARCHAR(15) NOT NULL,
 PNUMBER    INT NOT NULL,
 PLOCATION   VARCHAR(15),
 DNUM       INT NOT NULL,
 PRIMARY KEY (PNUMBER),
 UNIQUE (PNAME),
 FOREIGN KEY (DNO) REFERENCES DEPARTMENT (DNUMBER)
);

CREATE TABLE WORKS_ON
(ESSN      CHAR(9) NOT NULL,
 PNO       INT NOT NULL,
 HOURS     DECIMAL(3,1) NOT NULL,
 PRIMARY KEY (ESSN, PNO),
 FOREIGN KEY (ESSN) REFERENCES EMPLOYEE (SSN),
 FOREIGN KEY (PNO) REFERENCES PROJECT (PNUMBER)
);

CREATE TABLE DEPENDENT
(ESSN      CHAR(9) NOT NULL,
 DEPENDENT_NAME CHAR(15) NOT NULL,
 SEX       CHAR(1),
 BDATE     CHAR(9),
 RELATIONSHIP VARCHAR(8),
 PRIMARY KEY (ESSN, DEPENDENT_NAME),
 FOREIGN KEY (ESSN) REFERENCES EMPLOYEE (SSN)
);
```