

Oppgavesett – relasjonsalgebra

Formål: Trening i relasjonsalgebra og optimaliseringsforståelse.

Oppgave 1 og 2 er viktigst, de andre er morsomst ☺

Oppgave 1.

Gitt relasjonene:

STUDENT (studnr, studnavn, adresse,, postnr)

KURS (kursnr, kursnavn,, nivå)

EKSAMEN (studnr, kursnr, gangnr, dato)

- a) Finn studentnr og -navn med postnr = 9000
- b) * Finn studentnr og-navn som har tatt kurs nr. 345
- c) Finn studenter (studnr, studnavn) og de kurs (kursnr) som de har tatt. Studenter uten kurs skal ikke med.
- d) Finn alle studenter (studnr, studnavn) og de kurs (kursnr) som de (evt.) har tatt
- e) Som over, men kursnavnet skal også være med.
- f) * Som over, men bare personer med postnr = 9000 og kurs på nivå 3.
- g) Finn de studenter som har tatt samme eksamen mer enn en gang

- h) Systemet gjelder en høyskole (H1) som slås sammen med to andre høyskoler (H2, H3) som har samme relasjonsoppbygging. Foreslå evt. endringer i datastrukturen og vis hvorledes du skal slå sammen relasjonene for de ulike skolene slik at det danner et felles system.
- i) Finn studenter som er registret i både H1 og H2.
- j) Finn alle studenter (alt om studenten) som ikke har tatt kurs nr. 100. Bruk ikke semijoin.
- k) Finn alle studenter (alt om studenten) som ikke har tatt kurs nr. 100. Bruk semijoin
- l) Finn de studenter (hvis noen) som har tatt alle kursene som finnes.
- m) Finn de kursene som alle studenter har tatt, hvis noen.
- n) Formuler denne spørringen, gjerne også andre spørringer i SQL. Hva er lettest for de forskjellige spørringene?

Oppgave 2.

- a) Lag en ny relasjon hvor du bare har med kursnr og kursnavn. Bruk tilordning
- b) Som over, men bruk renames (ρ).
- c) Skriv ut antall som har gått opp til eksamen i hvert av kursene (tell bare med de med gangnr = 1). Resultatet skal inneholde kursnr og antall.
- d) Skriv ut antall eksamener pr. student. Studnr, Studnavn og antall skal være med.

Oppgave 3.

Gitt en relasjon R.

- a) Hvorledes kan du bruke relasjonsalgebra til å sette inn tuppelet (a_1, a_2, \dots) ?
- b) til å slette et gitt tuppel ?

Oppgave 4.

Gitt en skog. Finn alle røtter og blader.

Oppgave 5.

Vanlige SQL-setninger med koblinger mellom f.eks. to tabeller formuleres vanligvis enten som en kobling i where-setningen eller som en delspørring, f.eks.

enten:

```
select ....  
from student, eksamen  
where student.studnr = eksamen.studnr
```

eller:

```
select .....  
from student  
where studnr in  
  (select ..... from eksamen .....)
```

Hvilke(n) join-operator(er) egner seg for å bruke delspørring? Forklar!

Oppgave 6 - optimalisering.

Ta for deg de *-merkede deloppgavene i oppgave 1. Ut fra et optimaliseringshensyn, formuler en "verst mulig" og en "god" gjennomføringsmåte for disse.

Oppgave 7.

- a) Av operatorene UNION, DIFFERENCE, PRODUCT, RESTRICT, PROJECT og JOIN, INTERSECT, DIVIDEBY er de 5 første "basale", mens de 3 siste kan avledes av de første. Vis dette. NB! Divideby er langt fra lett !
- b) Vis at hvis $\text{card}(R) > 1$ og $\text{card}(S) > 1$ så er $\pi_{\text{pk_for_r}, \text{pk_for_s}}(R) : \pi_{\text{pk_for_s}}(S) = \emptyset$. \emptyset betegner her den tomme relasjonen.
- c) Divisjonsoperatoren er ikke lukket under reelle tall (fordi $x/0$ ikke er definert). Hva med divisjonsoperatoren i relasjonsalgebra?