Norges teknisk-naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap

LØSNINGSFORSLAG TIL
EKSAMENSOPPGAVE I FAG TDT4145 – DATAMODELLERING OG DATABASESYSTEMER

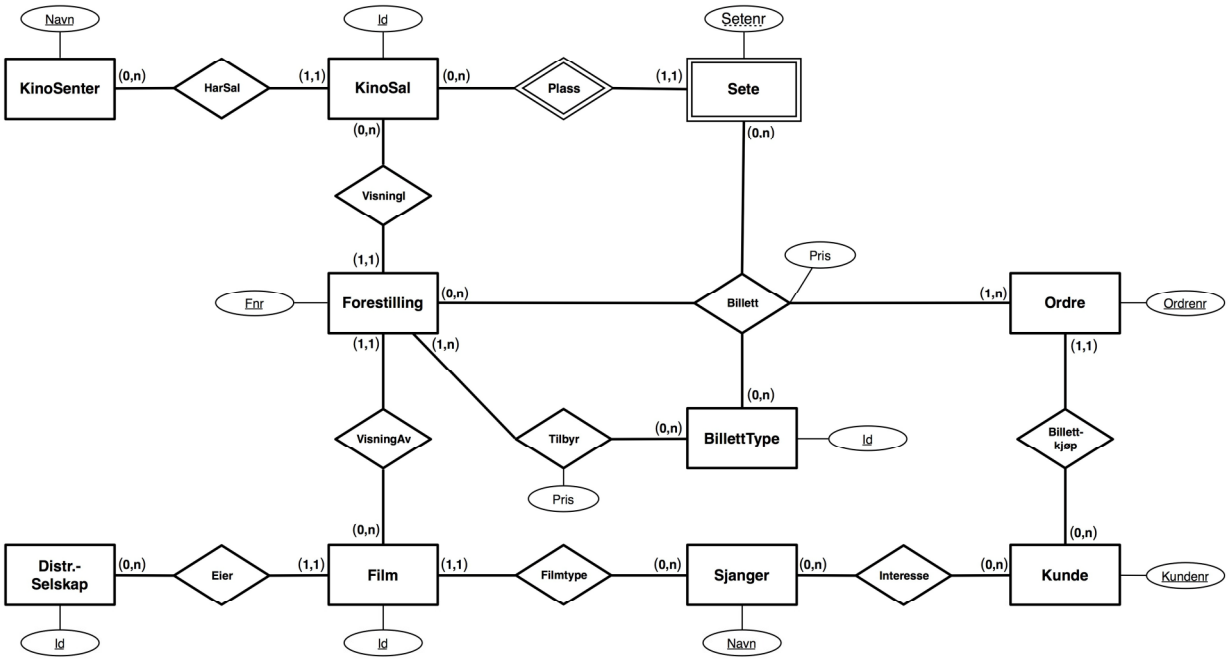
Eksamensdato: **1. juni 2011**

Eksamenstid: **09.00-13.00**

Tillatte hjelpemiddel: **D: Ingen trykte eller håndskrevne hjelpemiddel tillatt. Bestemt, enkel kalkulator tillatt.**

Oppgave 1 – Datamodellering – 25 %

ER-diagram med nøkkelattributter og attributter på relasjonsklassene:

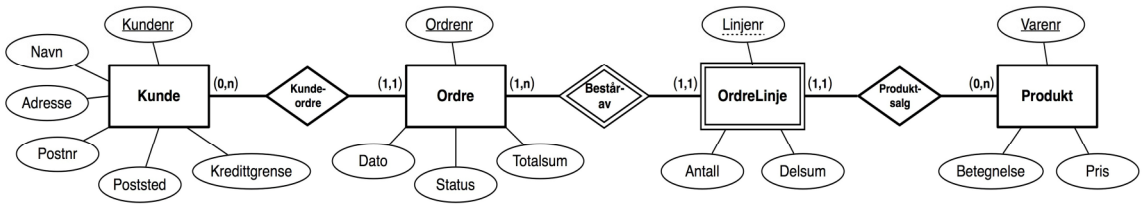


Det legges til attributter i entitetsklassene i forhold til den tekstlige beskrivelsen.

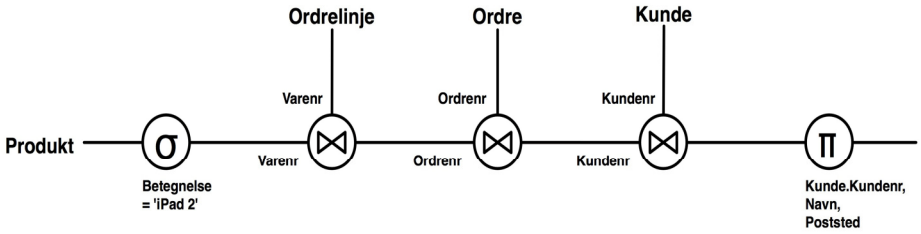
Oppgave 2 – Relasjonsalgebra og SQL – 20 %

a) Under er vist et ER-diagram der vi har gjort følgende forutsetninger:

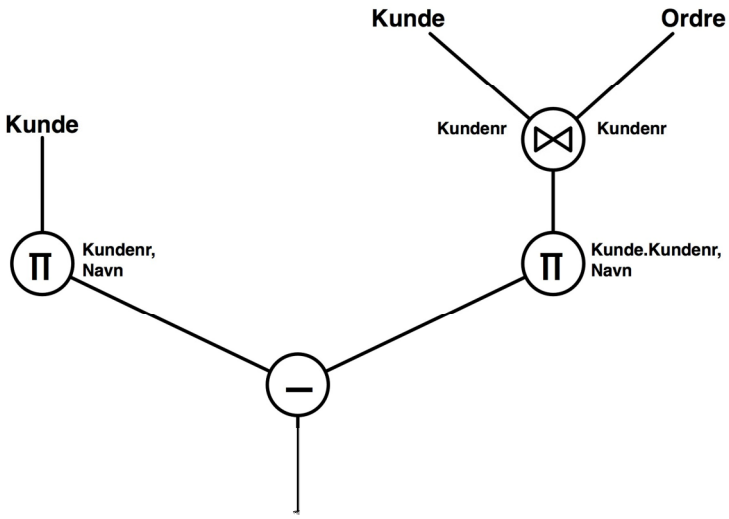
- En kunde kan eksistere uten ordrer.
- En ordre *må* ha en kunde og bestå av minst en ordrelinje.
- En ordrelinje *må* høre til en bestemt ordre og gjelde ett bestemt produkt.
- Et produkt kan eksistere uten å inngå i noen ordre(-linje).



b) Relasjonsalgebra:



c) Relasjonsalgebra:



d) SQL:

```
SELECT Kundenr, Navn, Adresse
FROM Kunde
WHERE Kundenr NOT IN (SELECT Kundenr
                      FROM Ordre
                      WHERE Totalsum >= 1000);
```

e) SQL:

```
SELECT Betegnelse, SUM(Delsum)
FROM Ordrelinje AS OL, Produkt AS P
WHERE OL.Varenr = P.Varenr
AND ((Betegnelse = 'grill') OR
     (Betegnelse = 'grillkull') OR
     (Betegnelse = 'tennvæske'))
GROUP BY Betegnelse
ORDER BY Betegnelse ASC;
```

Oppgave 3 – Normalisering – 15 %

a) Dersom alle tupler som har samme verdier for en mengde attributter (X), alltid vil ha samme verdier for en mengde attributter (Y), sier vi at Y er funksjonelt avhengig av X, $X \rightarrow Y$.

b) For at vekt $\rightarrow$ merke skal gjelde, må det være slik i mini-verdenen at alle biler med samme vekt er av samme merke. Når det finnes en Toyota som veier 1055 kilogram kan det for eksempel ikke eksistere en Volvo som veier det samme.

c) Den eneste (kandidat-)nøkkelen i tabellen er regNr. Siden nøkkelen ikke er sammensatt, kan det ikke eksistere delvise avhengigheter fra nøkkelen til et ikke-nøkkel-attributt. Dette er kravet til andre normalform som dermed er oppfylt.

Det finnes en funksjonelle avhengigheter mellom ikke-nøkkel-attributtene der venstresiden ikke er en supernøkkel og høyresiden ikke er nøkkelattributter, modell $\rightarrow$ merke, vekt. Tredje normalform (3NF) forbyr slike, tabellen er derfor ikke på 3NF.

d) Avgjør egenskapene til de tre dekomponeringene:

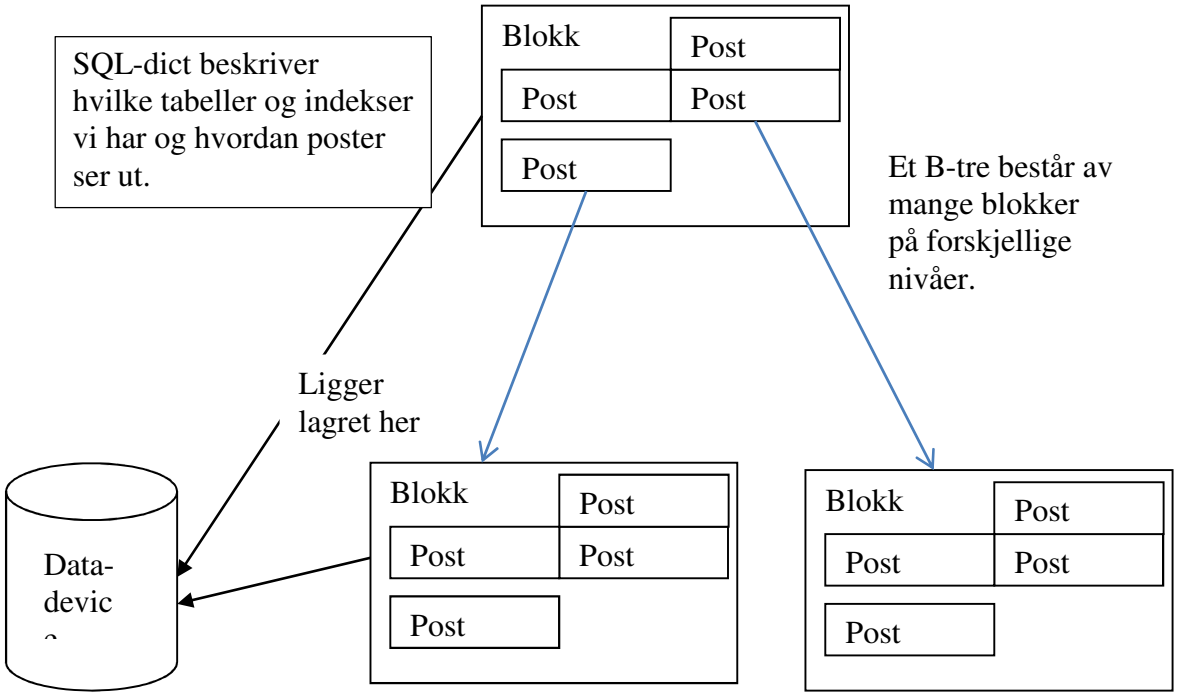
R1(a b) og R2(a c): Bevarer funksjonelle avhengigheter fordi a$\rightarrow$b kan ivaretas i R1 og har tapsløs-join-egenskapen fordi det felles attributtet (a) er en supernøkkel i R1.

R1(a b) og R2(b c): Bevarer funksjonelle avhengigheter fordi a$\rightarrow$b kan ivaretas i R1, men har ikke tapsløs-join-egenskapen siden det felles attributtet (b) ikke er en supernøkkel i verken R1 eller R2.

R1(a c) og R2(b c): Bevarer ikke funksjonelle avhengigheter siden a$\rightarrow$b ikke kan ivaretas i noen av del-tabellene, men blir en inter-tabell-avhengighet mellom R1 og R2. Dekomponeringen har heller ikke tapsløs-join-egenskapen siden det felles attributtet (b) ikke er en supernøkkel i R1 eller R2.

Oppgave 4 – Lagring, indekser og queryutføring – 20 %

a)



b)

Vi gir to eksempler her. Ett som viser en direkte aksess av søkenøkkel, og ett som viser et rangequery.

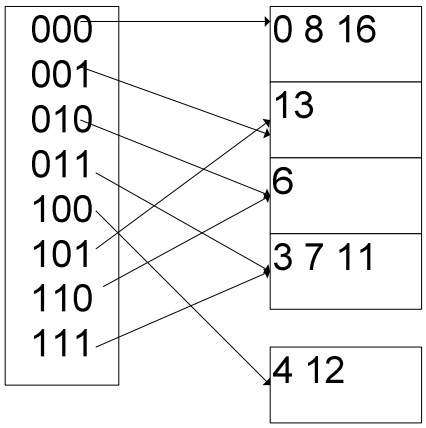
SELECT navn FROM Student WHERE studNr=123456;

Og det er en indeks på studNr, for eksempel en hashindeks. Hashindeksen gjør at stort sett er det kun ei blokk som aksesseres. Her kan tabellen være en clustered hash.

SELECT navn FROM Ansatt WHERE lønn>1000000;

Og det er en B-tre-index på lønn. Hvis queriet er selektivt, vil indeksen gi god ytelse fordi kun få poster i tabellen vil aksesseres. Her vil indeksen sannsynligvis være en sekundærindeks.

c)



Her har vi vist pekerstrukturen etter at den første blokka ha blitt splittet/rehashet.
Studenter som har svart med vanlig hashing med overløp, dvs. blokk 0 har en peker til en ekstra blokk, har også blitt den del belønnet.

d) De algoritmene som er nevnt i kap. 12 i boka er:
   1. Nested loop: For hver post/blokk i den ene tabellen, gå gjennom alle poster i den andre tabellen og se etter match.
   2. Index nested loop: For hver post i den ene tabellen, slå opp i en index for den andre tabellen.
   3. Sort-merge join: Sorter begge tabellene på join-kolonnen og så bruk fletting av tabellene.

Oppgave 5 – Transaksjoner – 20 %

a)

(1) 1->3->2. *Konfliktserialiserbar*
(2) 1->3<->2. Det er sykel mellom 2 og 3. *Ikke konfliktserialiserbar*.
(3) 1->2->3. *Konfliktserialiserbar*.
(4) 1<->2->3. Sykel mellom 1 og 2. *Ikke konfliktserialiserbar*.

b)

(1) Gjenopprettbar. 2 leser fra 3 og committer senere. Ikke ACA.
(2) Ikke gjenopprettbar. 2 leser fra 3 og committer før.
(3) Strict. Ingen leser eller skriver en ucommittet verdi. Verdier skrevet blir straks committet.
(4) Strict. Ingen leser eller skriver en ucommittet verdi. Verdier skrevet blir straks committet.

c)

Fordelen med "no-force" er at transaksjonen slipper å skrive data ved commit, det holder med å skrive loggen, som kan gå temmelig raskt.
Fordelen med "steal" er at det er mulig å "stjele" et skittent buffer. Dette gjør buffermanager mer fleksibel og man slipper at en transaksjonen har reservert hele bufferet med sine skitne blokker.

d)

En CLR kompenserer for en vanlig loggpost, dvs. den representerer undo av en loggpost. For å gjøre undo må man da redoe CLRen. Det at vi har CLRer gjør også at vi kan støtte fingranulære låser, dvs. låser på postnivå. Dette er fordi en undo ikke setter tilbake PageLSNen, men installerer CLRens LSN inn i PageLSN.