

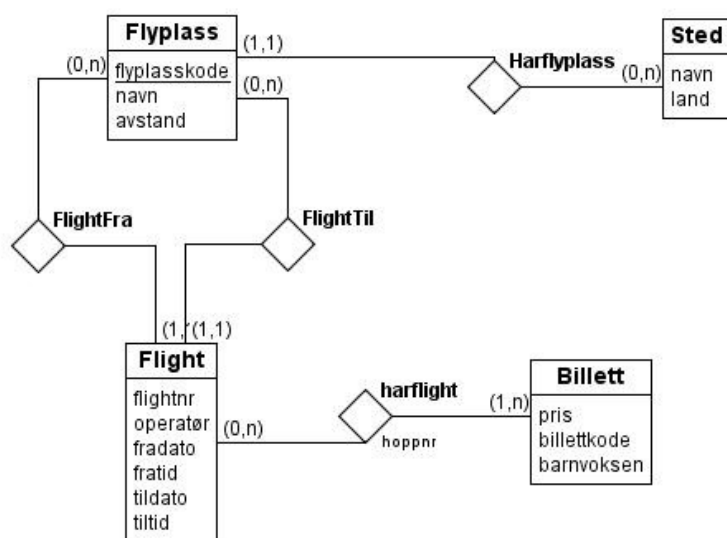
Norges teknisk-naturvitenskapelige universitet
 Institutt for datateknikk og informasjonsvitenskap



**LØSNINGSFORSLAG TIL
 EKSAMENSOPPGAVE I FAG TDT4145 – DATAMODELLERING OG
 DATABASESYSTEMER**

13. august 2008

Oppgave 1 – Datamodellering – 20 %



Oppgave 2 – Relasjonsalgebra og SQL – 20 %

- $PROJECT_studadr(SELECT_emnenavn='Objektorientert\ systemutvikling' (Emne) JOIN SELECT_karakter='A' (Eksamen) JOIN Student)$
- $PROJECT_studnr(SELECT_emnenavn='Objektorientert\ systemutvikling' (Emne)) JOIN Eksamen) INTERSECT PROJECT_studnr(SELECT_emnenavn='Systemutvikling' (Emne)) JOIN Eksamen)$

- c) *SELECT sensornavn*
FROM sensor, eksamen, student
WHERE studadr=sensoradr AND sensor.sensorid=eksamen.sensorid AND
eksamen.studnr=student.studnr;
- d) *SELECT karakter, count(*)*
FROM eksamen
WHERE emnekode='TDT4145'
GROUP BY karakter;
- e) *SELECT karakter*
FROM eksamen
GROUP BY karakter
HAVING count()>= ALL (SELECT count(*)*
FROM eksamen
GROUP BY karakter);
- f) *SELECT sensorid*
FROM eksamen
WHERE karakter='F'
GROUP BY sensorid
HAVING count()>= ALL (SELECT count(*)*
FROM eksamen
WHERE karakter='F'
GROUP BY sensorid);

Oppgave 3 – Lagring og systemkatalog – 20%

- a) Fastlengdeposter er beskrevet i læreboka i 9.7.1. Postene er pakket tett med feltene på faste posisjoner, systemkatalogen beskriver lengden på hvert felt. Variabellengdeposter er beskrevet i 9.7.2 med to forskjellige formater, enten med feltpekere eller med feltseparasjonstegn.
- b) Blokkformater er beskrevet i læreboka i 9.6. Fastlengdeposter kan lagres rett etter hverandre med antall poster til slutt, eller ”unpacked” med en bitvektor til slutt. Blokker med variabellengdeposter kan ha postene lagret ”unpacked” fra starten av blokka, mens slutten har et ”slot directory” med pekere til postene og antall pekere.
- c) Heapfiler er beskrevet i kapittel 9.5. To implementasjoner er beskrevet, en med to sammenlenkede lister av blokker. Den ene lista er med full blokker, den andre med ledige plass. Eller så kan de implementeres med ”directory of pages”, dvs det er separat sett av indeksblokker som er lenket sammen. Hver indeksblokk inneholder en rekke pekere til vanlig datablokker.
- d) Systemkatalogen er beskrevet i læreboka i kapittel 12.1, og den består av informasjon om systemet, som blokkstørrelser, antall blokker i buffer, og lignende. I tillegg er det informasjon for hver tabell, view, indeks, restriksjoner og prosedyrer og lignende. Den lagrer også statistikk om antall poster og blokker for hver tabell og indeks.

Oppgave 4 – Queryevaluerer – 5%

Læreboka øverst side 416. En queryoptimalisator bruker ”left deep plans” for å minske antall planer som vurderes og for å kunne utnytte fullt pipelinet utføring.

Oppgave 5 – Transaksjoner – 15%

- a) Læreboka side 583-4. **prevLSN** er peker til forrige loggpost i samme transaksjon, **transID** er transaksjonsidentifikatoren, **type** beskriver om det er update, insert, delete el., **pageId** er identifikatoren til den databblokka som er berørt av loggposten, **length** er lengden av endringen, **offset** er peker inn i posten for endringen, **before-image** er gammelt bilde av posten, **after-image** er nytt bilde av posten
- b) Læreboka side 584. En kompeseringsloggpost er en spesiell redologgpost som er brukt til undo av en annen loggpost. Kan være nyttig for å få til recordnivå låsing.
- c) Det er tre faser av recovery, analyse, redo og undo. Dette er beskrevet i kapittel 18.6. Analysefasen starter fra siste sjekkpunktloggpost i loggen og går framover for å finne taper- og vinnertransaksjoner. Redofasen sørger for at alle vinnertransaksjoner får gjort ferdig sine operasjoner. Undofasen ruller tilbake tapertransaksjoner ved å kompensere for de aksjonene den har gjort.

Oppgave 6 – Normalisering – 20%

- a) Navn -> Klubb, Klasse Renn -> Arena Navn, Renn -> Tid, Plassering
- b) **Innsettingsanomalier.** Det går ikke å sette inn en løper uten å sette inn resultater.
Slettingsanomalier. Hvis du sletter informasjon om en løpers renn, kan du slette all informasjon om løperen. **Oppdateringsanomalier.** Siden informasjon om klubb og klasse er redundant lagret, kan det bli inkonsistents ved oppdatering. **Redundans.** Dobbeltlagring av klubb, klasse, arena.
- c) Nøkkel er renn, navn. Klubb, Klasse, Arena er delvis avhengig av nøkkel. Da er ikke tabellen på 2. normalform. Vi antar da at tabellen er på 1.normalform.
- d) Løper(navn, klubb, klasse)
 Renn(renn,arena)
 Resultat(renn,navn,tid,plassering)