



Emnekode : 15-201
Kandidatnr. : 5145
Dato : 08.12.2011
Ark nr. : 1 av 8

Opg 1

1 En database er en strukturert logisk samling av data. Dataene som lagres i databasen er logisk relatert til hverandre innenfor et område. For eksempel vil en bilforandlers database inneholde data om biler og salg, men ikke ting som ikke ~~har~~ har med ~~bedriften~~ bedriften å gjøre, f.eks blomster. I en relasjonsdatabase er dataene strukturert i tabeller og tabellene ~~relatert~~ kodes til hverandre gjennom primær- og fremmed-nøkler. Kort fortalt er altså databaser det stedet hvor ønsket data lagres, og den ligger gjerne på f.eks en server. DBMS (Database Management System) er derimot databasehåndterings-systemet. Det er altså softwaren som lar oss benytte databasene. DBMS inneholder en motor som lar oss opprette, endre eller slette innhold i databasen. I tillegg har DBMS et språk som baserer seg på SQL. Eksempell på DBMS er MySQL og Oracle. DBMS er altså linke ~~en~~ mellom brukeren og databasen.

2 DD står for Data Definition. DD er hvordan datan i databasen er definert. Det er f.eks tabellene som blir definert. Eksempler på hva en tabells DD består av er navn, hvilke attributter den har, hvilke constraints som finnes osv. Eksempler på kommandoer som endrer DD (altså DDL kommandoer) er CREATE, ALTER og DROP. F.eks:

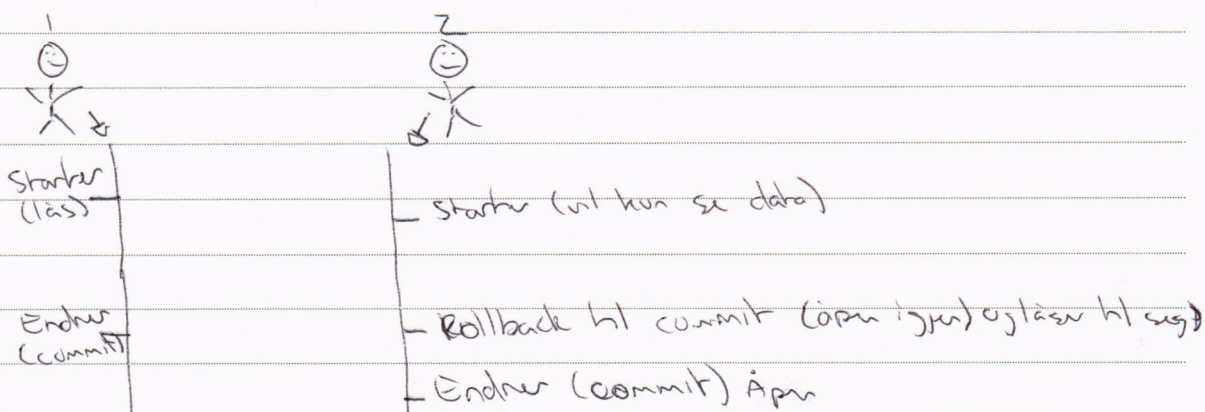
```
CREATE TABLE Bilmerke (  
  navn varchar(45) not null;  
  osv.
```



Emnekode : 15-201
 Kandidatnr. : 5145
 Dato : 08.12.2011
 Ark nr. : 2 av 8

3 DBMS bruger læsning af data (kun læse tabeller, kolonner eller rader) for at håndtere mange samtidige brugere. Grunden til dette er at kun én om gangen skal kunne udføre ændringer. På en konto skal ikke to personer kunne gå ind samtidig, og hente ut penge da databasen ikke vil ha mulighed til at opdatere rigtig. Når en bruger starter brug af databasen læser de aktuelle dataene. Næstemand får da se dataene, men har ikke mulighed til at ændre noe, siden det er låst. Når førstemand er færdig, blir dataene commitet, altså det blir lagret, og læser åbnes. For person nr. 2 starter det en rollback til forrige commit (altså den nr. 1 udførte) og dataene er åbne for at ændres. Dersom begge prøver at ændre samtidig ~~kan~~ det bli en vængløs, så ingen får gjort noe. Dette ^{kan} løses ved at det er en timer som timer ut etter en viss tid. Da åpnes dataene igjen.

Eksempel:



Eksempler på concurrency control kommandoer er COMMIT, ROLLBACK og AUTOCOMMIT. Commit lagrer, rollback ruller dataen tilbake til siste commit og autocommit er en innstilling om commit skal skje



Emnekode : 15-201
Kandidatnr. : 5145
Dato : 08.12.2011
Ark nr. : 3 av 8

3 fortsettelse:

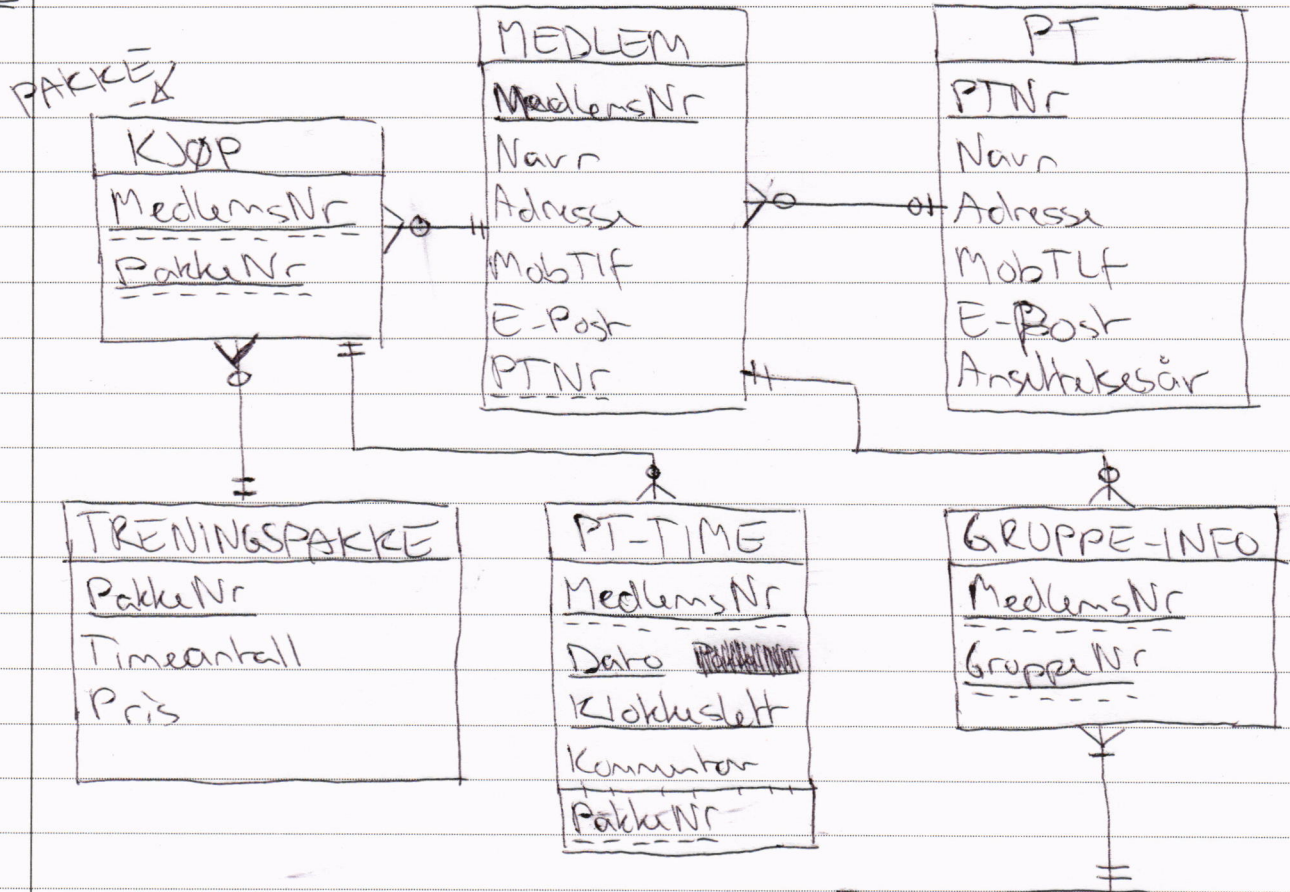
automatiske eller ikke.

4 Hvis man ikke indleser i store tabeller vil det oppstå gå sikkerstullt. Søket vil altså starte på toppen av tabellen og søke seg nedover til den finner det som leies etter. Hvis det fins veldig mange rader vil denne prosessen sinke søkene. Hvis man heller setter indreer på tabellene vil det gå betraktelig raskere.

Når man setter en index er det på n måte som et bokmerke. Indexen refererer til en rad i tabellen og legges i en egen tabell. Når det søkes vil indexen bli spikket og man finner raden ut i fra "bokmerket".



Oppg 2



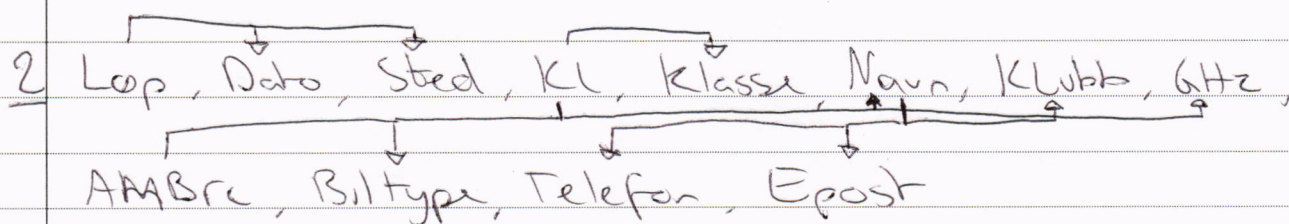
Forklaring/forutsetninger:

- Hver tabell har fått en primærnøkkel i form av et nummer. Feks PakkeNr og PTNr.
- Går ut ifra at et medlem ikke må ha en PT
- PT-TIME får info om hvilken PT det gjelder via medlemsnummeret
- Går ut ifra at Gruppetreninger har et navn og en veileder i tillegg til en egen pris.
- PT-TIME har klokketilt, i tillegg til MedlemsNr og Dato som primær-nøkkel da et medlem må kunne trene flere ganger om dagen.



Oppg 3

1 Rapporten er på UNF. UNF er steg for INF. Grunnen til at rapporten er på UNF er at den inneholder fler-verdi-attributter. Disse kan man se i eksempelvis kolonnene "KL" og "Klasse". For å få tabellen til INF kan man fylle ut de tomme feltene. På den måten fjerner man fler-verdi-attributtene.



Disse funksjonelle avhengighetene finnes i rapporten:

Løp → Dato, Sted

KL → Klasse

Navn → Epost, Telefon

KL, AMBrc → Biltype, GtZ, Navn

Funksjonell avhengighet vil si at dersom du f.eks har et løpsnavn så er det kun en mulig dato og ett mulig sted. Ut ifra løpsnavnet vet du altså både dato og sted.



Emnekode : 15-201
 Kandidatnr. : 5145
 Dato : 08.12.2011
 Ark nr. : 6 av 8

Oppg 3 1NF:

Løp

3 (Løp, Dato, Sted, KL, Klasse, Navn, Klubb, GHz, AMBRc, Biltype, Telefon, Epost)

2NF:

Klasse (KL, Klasse) (3NF-se løp) Transitiv avhengighet
~~Klasse~~ (KL, AMBRc, ~~KL~~, Navn, Klubb, GHz, Biltype, Telefon, Epost)
 Bil Løp →

Løp (Løp, Dato, Sted) (Allerede 3NF da det ikke er noen transitive avhengigheter)

3NF:

Løp (Løp, Dato, Sted)

Klasse (KL, Klasse)

Bil (AMBRc, KL, GHz, Biltype, Navn, Løp)

Deltaker (Navn, Klubb, Telefon, ~~Epost~~)

- Fra 1NF → 2NF fjernes flerverdigrattributter og primary key settes
- Fra 2NF → 3NF fjernes delvis avhengighet. Non-key attributter må være avhengig av hele primary key'en.
- Fra 3NF → 4NF fjernes de transitive avhengighetene. Et non-key attributt kan ikke være avhengig av et annet non-key attributt.



Emnekode : 1S-201
Kandidatnr. : 5145
Dato : 08.12.2011
Ark nr. : 7 av 8

Ops 4

1 SELECT fFnavn, fEnavn, tittel, år
FROM bok, forfatter
WHERE
bok.isbn = forfatter.isbn
AND
fFnavn IS "Salman"
AND
fEnavn IS "Rushdie";

2 CREATE VIEW bokoversikt AS
(SELECT fFnavn, fEnavn, tittel, år, forlNavn
FROM forlag, bok, forfatter
WHERE
forlag.forlNavn = bok.forlag
AND
bok.isbn = forfatter.isbn
ORDER BY
fEnavn, tittel);



Emnekode : 1S-201
Kandidatnr. : 5145
Dato : 08.12.2011
Ark nr. : 8 av 8

Oppg 4

3

```
SELECT lFnavn, lEnavn, fFnavn, fEnavn, count(fFnavn, fEnavn)
FROM lantaker, utlan, bok, forfatter
WHERE lantaker.medlemsld = utlan.medlemsld
AND utlan.isbn = bok.isbn
AND bok.isbn = forfatter.isbn
GROUP BY fFnavn, fEnavn, lFnavn, lEnavn lEnavn
HAVING lFnavn IS "Jenny"
AND lEnavn IS "Hansen"
AND count(fFnavn, fEnavn) > 1;
```

4

```
SELECT bok.isbn, tittel
FROM bok
LEFT JOIN utlan
ON bok.isbn = utlan.isbn
WHERE utlan.isbn IS null;
```

5

```
SELECT tittel, count(utlan.isbn)
FROM bok, utlan
WHERE bok.isbn = utlan.isbn
GROUP BY tittel
ORDER BY count(utlan.isbn) DESC
LIMIT 1;
```