

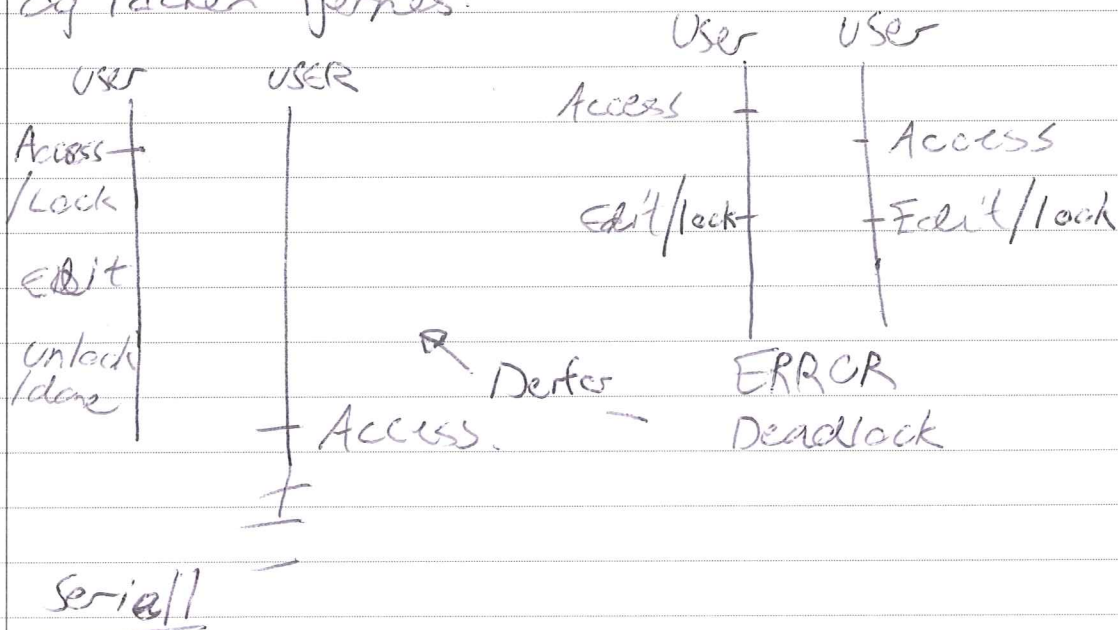


Oppgave 1

a) Concurrency Controler mekanismen som konflikthåndterer ~~brøker~~<sup>endringer</sup> i databaser.

Når ~~en~~<sup>en</sup> brukere foretaker å endre noe setter det en "lock". Locken har ulike grader (levels) fra å lese påvarende kolonne til å lese rader, tabeller og hele databaser. Det er to lock-typer: shared, gjerne kalt "s" hvor kun endringer låses ut forstående (andre som også vil endre), men kan fortsatt vises (read only) Vi har også en "eksklusiv"-type (x) som ~~ikke~~<sup>ikke</sup> tillater "read only" for andre.

Deadlock oppstår når to brukere forsøker å ~~gjøre~~<sup>gjøre</sup> endre underveis samtidig. Derfor benyttes seriell tilgang (serializing) og locken utføres i del brukeren søker tilgang, i stedet for når endringene utføres. Dette vil si at de andre brukerne må vente i "kjø" til den andre brukeren er ferdig og locken fjernes.



Seriell



Emnekode : 1S 201  
Kandidatnr. : 2138  
Dato : 24/11 09  
Ark nr. : 2 av 8

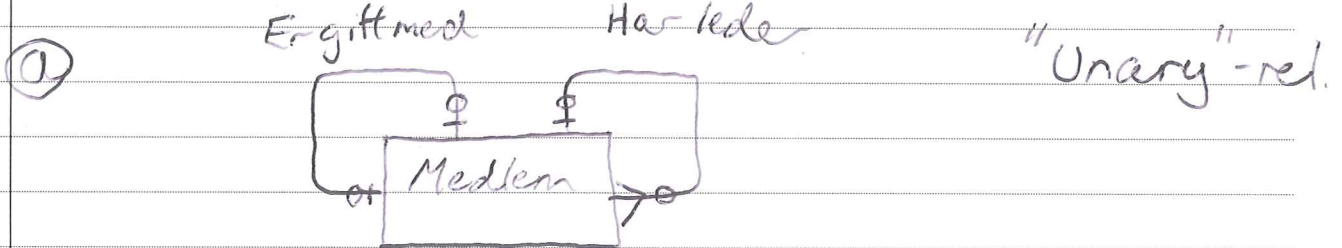
## Oppgave 1

(b)

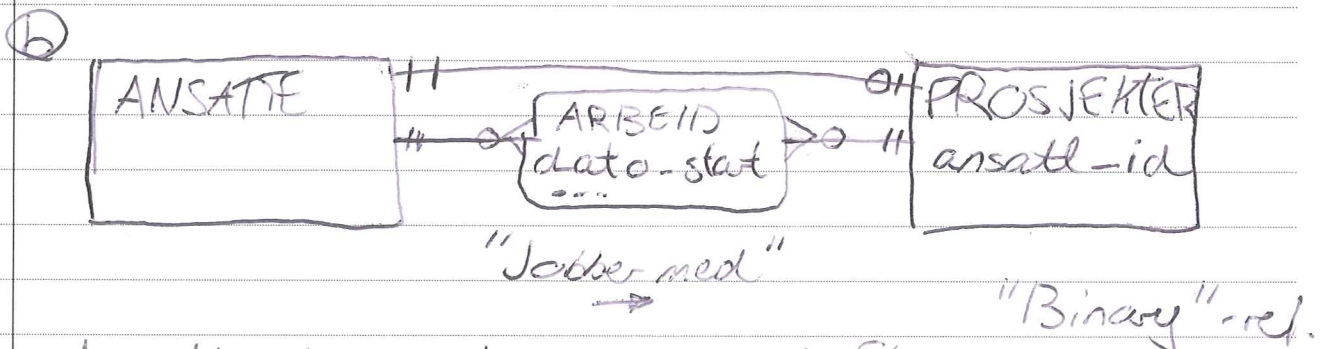
Data-dictionary sitter på informasjonen som definerer databasen. Det vil si informasjon om tabeller, attributter, typer, constraints, views, indexter og mer. Eksempel er en tabell, hvor "meta-dataen" rundt denne tabellen er lagret i data-dictionary. Hver attributt (ID integer(6) NOT NULL, NAME varchar2(25), ~~rest~~ hver regel (constraint ~~er~~ sjekk check (...), constraint nøkkel primary key (...)).



### Oppgave 2



Kardinalitet : • Et medlem kan kun være gift med ett annet medlem  
 • Et medlem kan ha en leder, men medlem kan være leder for mange andre medlemmer



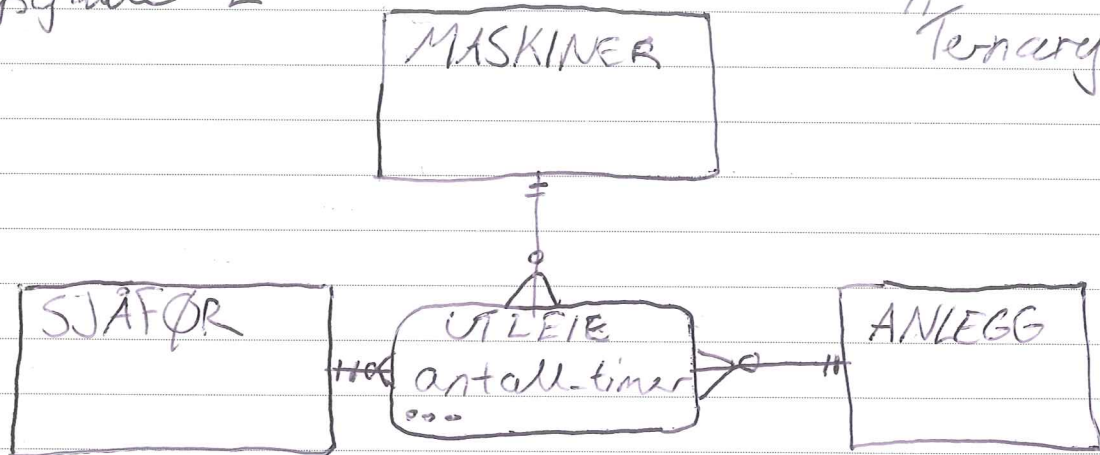
• Ansatte kan arbeide med flere prosjekter samtidig  
 • Prosjekter kan ha flere ansatte, men må ha en ansvarlig leder.  
 Forutsetninger : Jeg velger å la ansatt kun være sjef for ett prosjekt om gangen



Emnekode : 18-201  
 Kandidatnr. : 2738  
 Dato : 24/11 09  
 Ark nr. : 4 av 8

## Oppgave 2

c



- Sjøfører kan jobbe på flere maskiner på ulike anlegg.
- For at et utleie skal oppstå, må det tilhøre en sjøfører <sup>og en</sup> maskin og et anlegg.

## Oppgave 3

a

Tabellen er unormalisert (UNF). Dette kan vi se fordi den inneholder fler-verdi-attributter (multi-value).

① Empno → Empname, Jobclass

Projno → Projname, Projleaderno, Projleadername, subtotal

Projno, Empno → ChargeHR, HRbilled, totalcharge

MERK HER: • ChargeHR kan tilhøre Empno →, men jeg går ut i fra at lønna varierer etter prosjekt! (i stedet for "job class") \*viktig! (Påvirker videre utvikling)



Emnekode : 15 201  
 Kandidatnr. : 2138  
 Dato : 24/11 09  
 Ark nr. : 5 av 8

### Oppgave 3 (b) Forts.

- Jeg har også valgt å fjerne "Total" før 1NF er oppnådd. Dette er fordi denne deriverte attributter skaper ekstremt mye gjentakelse (redundancy) sett sammen med de andre deriverte att: "total charge" og "subtotal", som fjernes under prosessen mot 2NF.
- Jeg holder Pldrname og Emprname som to attributter, istedet for å oppdele dem som (first, last) Dette utføres jeg i 2NF-fasen.

### © 1NF

<u>Projno</u>	Projname	Pldrno	Pldrname	<u>Emprno</u>	Emprname	Jobclass	C/HR	HR	Tot CHG	SUB Toc

(Ingen fler-vedi-attributter)

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

Vi deklarerer primærnøkkel basert på kandidaten  
 (Projno, Emprno)

Merk: Jeg går ut i fra at en ansatt kan  
 kan tilhøre en "jobclass"  
 \* Viktig!

### 2NF

Her fjerner vi delvis-funksjonelle avhengigheter  
 og fjerner alle derivert attributter (den kan kalkuleres  
 ved hjelp av  $C/HR * HR$ )



Emnekode : 1S 207  
 Kandidatnr. : 2138  
 Dato : 29/11 09  
 Ark nr. : 6 av 8

Oppgave 3 © forts.

2NF  
 EMPLOYEE

Empno, Emp-first-name, Emp-last-name, job-class

PROJECT

Projno, Projname, leader-no

WORK

Projno, Empno, charge-HR, Hours

Merk: For å unngå kompositt PK i "Work", hadde et alternativ vært å ta i bruk en "surrogate key" fets. ~~Workno~~ "Workno"

3NF

Her fjerner vi transitive avhengigheter (funksjonelt avhengige attributter som avhenger av en attributt som ikke er ~~PK~~ PK)

EMPLOYEE (~~Empno~~ Empno, Emp-first-name, Emp-last-name, Jobno)

JOB (~~Jobno~~ Jobno, Jobtitle)

MERK: Som nevnt tidligere kunne "Job" ~~ha~~ hatt attributtene: "charge-HR", men jeg har tatt det slik at timelønn avhenger av prosjekt.



Emnekode : 18-201  
Kandidatnr. : 2138  
Dato : 24/11 09  
Ark nr. : 7 av 8

## Oppgave 4

a) Alter table utleie

constraint pk-utleie primary key (klientnr, eiendomNr),  
constraint fk-utleieklient foreign key (klientnr)  
references Klient,  
constraint fk-utleie-eiendom foreign key (eiendomNr)  
references EiendomTilLeie;

b) CREATE VIEW ~~EiendommerOgEier~~ AS  
SELECT eiendomNr, eAdresse, pris, eierNr, navn  
FROM EiendomTilLeie ~~INNER~~ JOIN EiendomEier  
ON EiendomTilLeie.eierNr = EiendomEier.eierNr  
WITH READ ONLY;

c) SELECT navn, sum(pris)  
FROM EiendomEier INNER JOIN EiendomTilLeie  
ON Eiendom.eierNr = EiendomTilLeie.eierNr  
GROUP BY navn  
ORDER by navn;

e) SELECT navn  
FROM klient  
MINUS  
SELECT distinct navn  
FROM klient INNER JOIN utleie  
ON klient.klientnr = utleie.klientnr;



Emnekode : 15 207  
Kandidatnr. : 21 38  
Dato : 24/11 09  
Ark nr. : 8 av 8

## Oppgave 4

d) SELECT EiendomsTilLeie.eiendomNr,  
to-number(utleieSlutt)-to-number(utleieStart)  
dagerLeiet, sum(dagerLeiet\*(pris/30))  
FROM klient INNER JOIN utleie ON klientNr=  
... klientNr INNER JOIN EiendomsTilLeie on utleie.  
eiendomNr = EiendomsTilLeie.eiendomNr  
WHERE lower(navn)=lower('Cynthia Hagen')  
GROUP by eiendomNr;