



Oppgave 1.

a) Commit brukes for å lagre endringer ^{permanent} i en database, mens rollback brukes for å "UNDO" på reversere endringene i databasen fram til siste commit.

Select * From Test;

A	B	C
-	-	-

Insert into Test (A,B,C)
Values (1,2,3);

A	B	C
1	2	3

Rollback;

A	B	C
-	-	-

Select * From Test;

A	B	C
-	-	-

Insert into Test (A,B,C)
Values (1,2,3);

A	B	C
1	2	3

Commit;

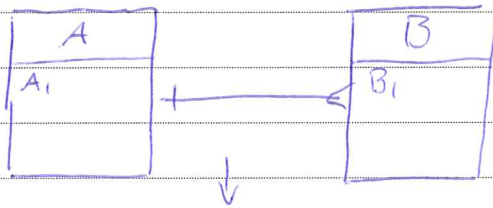
Rollback;

A	B	C
1	2	3

Det vil si man vil gjer endringer i en database, kan disse gjeres om ved hjelp av Rollback, men hvis vi commiter dem så Rollback "UNDO"er endringene fram til commit.



b) Vi implementeres en-til-mange ^{sammenheng} ved å ta primær
nøkkel fra "en side" og legger den inn som fremmed
nøkkel på "mange sider".

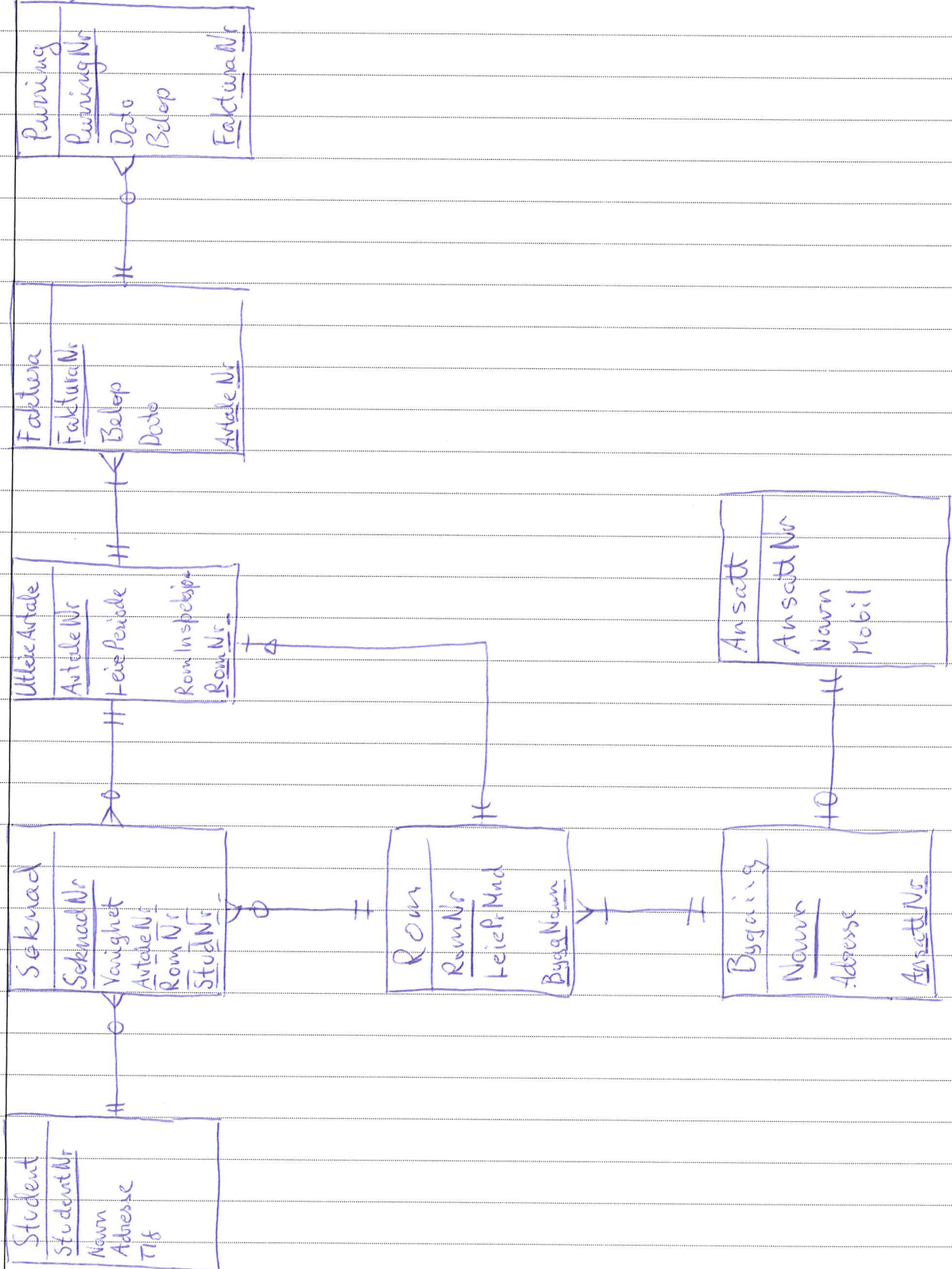


c) Index er et tabell som inneholder arkiv om data for
raskere gjennomføring, dvs den brukes for å kunne finne
ting raskere. Index brukes vanligvis for tabeller som
har over 1000 rader. Index kan deles i B-tree og
Bitmap, hvor Bitmap brukes når attributten har veldig
få verdier. Primær nøkkel blir automatisk indexert.
Kan også nevnes at hvert index fører til ekstra
prosessering ved Updates, og burde ikke brukes overdrevent,
(vurdere hva som gir mest utbytte)



Emnekode : 15 201
 Kandidatnr. : 3138
 Dato : 26.11.10
 Ark nr. : 3 av 7

Oppgave 2.

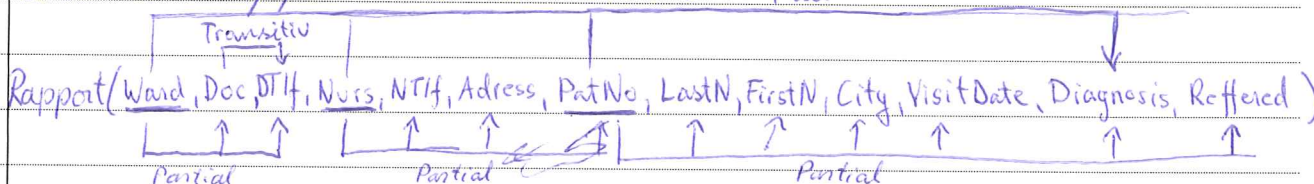




Emnekode : 15201
 Kandidatnr. : 3138
 Dato : 26.11.10
 Ark nr. : 4 av 7

Oppgave 3.

Denne rapporten er i UNF. Full



1 NF

Rapport (WardNo, DoctorInCharge, Phone, Nurs, PhoneNumber, PatientNo, LastName, FirstName, HomeCity, VisitDate, Diagnosis, RefferedTo)

PK (sammensatt) = WardNo, Nurs, PatientNo

2 NF

Ward (WardNo, DoctorInCharge, Phone)

Nurs (Nurs, PhoneNumber, Adress, WardNo)

Patient (PatientNo, LastName, FirstName, HomeCity, VisitDate, Diagnosis, RefferedTo, WardNo, Nurs)



Emnekode : 15 201
Kandidatnr. : 3138
Dato : 26.11.10
Ark nr. : 5 av 7

3NF

~~Siden jeg ikke finner noen transitiv uavhengighets så er
2NF formen 3NF også.~~

Ward (WardNo, DoctorInCharge)

Doctor (DoctorInCharge, Phone)

Nurse og Patient ifra 2NF er allerede i 3NF

Oppgave 4.

a) Constraints er gitte betingelser i databasen, som brukeren ikke kan påvirke. De holder data integriteten på plass i databasen og kan være alt fra PK, FK, CHECK, til å med attributt definisjoner er et slags constraints.

Alter table ordredetalj

```
add constraint ordrens_pk primary key (ordrens)
add constraint produktns_fk foreign key (produktns)
references produkt(produktns);
```

b) Select produktnavn

```
From (produkt inner join ordredetalj using produktns)
inner join ordre using ordrens
Where ordre_dato LIKE ('2000-10-%');
```



Emnekode : 1S 201
Kandidatnr. : 3138
Dato : 26.11.10
Ark nr. : 6 av 7

c) Select kundens, kunde_navn
From kunde
Where kundens NOT IN (
Select kundens From ordre);

d) Select kundens, kunde_navn, SUM(antall_bestilt),
SUM(antall_bestilt * enhetspris)
From (kunde inner join ordre using kundens)
inner join ordredetalj using ordrens
inner join produkt using produktus
Group by kundens, kunde_navn;

e) Where postus BETWEEN (4600 AND 4699)
AND antall_bestilt > 5

Denne kode biten legges inn i oppgave d) etter siste inner join, for Group by.



Emnekode : 15 201
Kandidatnr. : 3138
Dato : 26.11.10
Ark nr. : 7 av 7

f) Et database view er et virtuelt tabell ^{laget} av et spørring. Den holder seg alltid oppdatert ved at hvert gang vi kjører et view blir dens spørring også utført. View kan brukes for å begrense adgang til data og dermed beskytte databasen ~~og~~ fra uønsket innsyn.

```
Create View test as  
Select kundemr, kundemavn  
From kunde;
```