

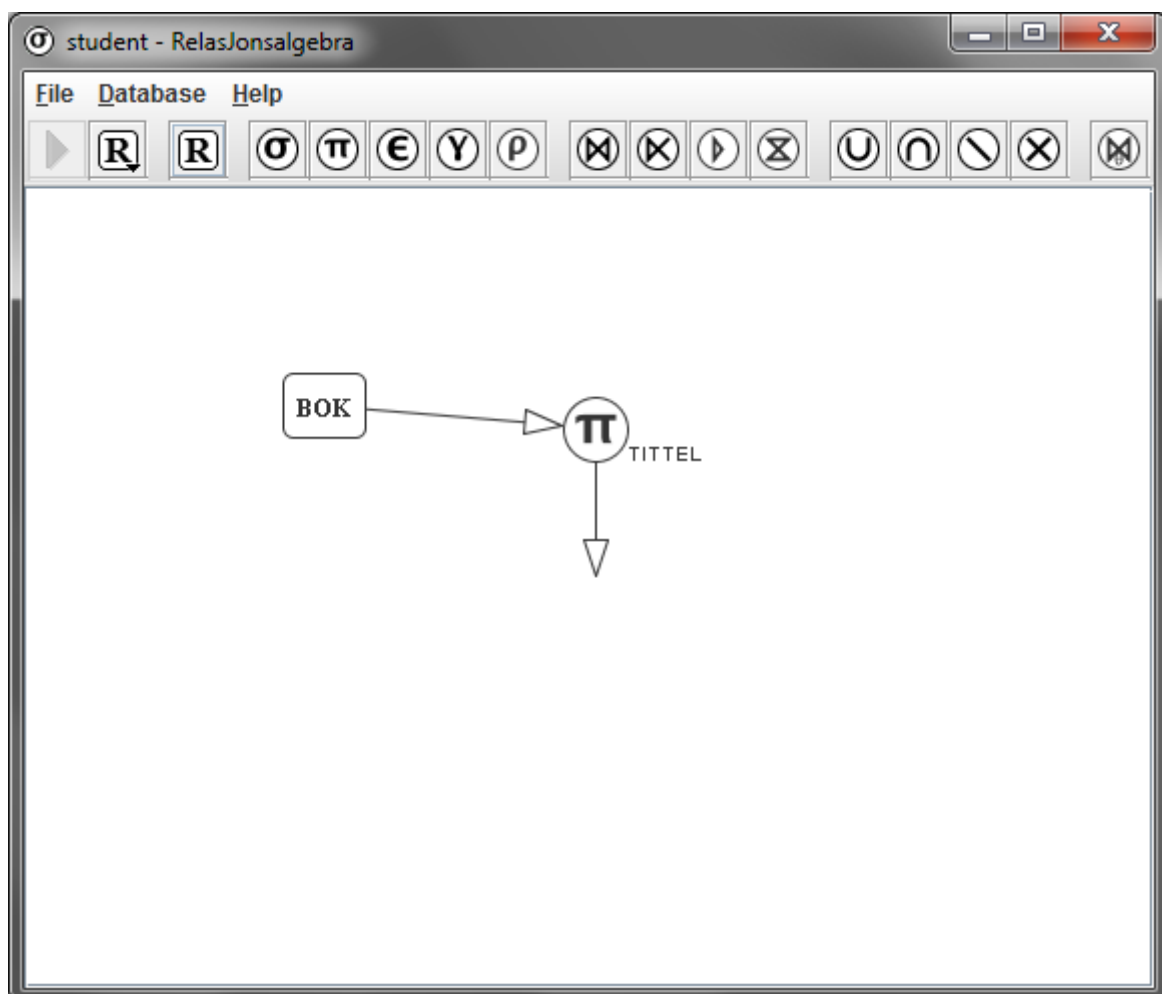
TDT4145 – Øving 3

Aleksander Hansen

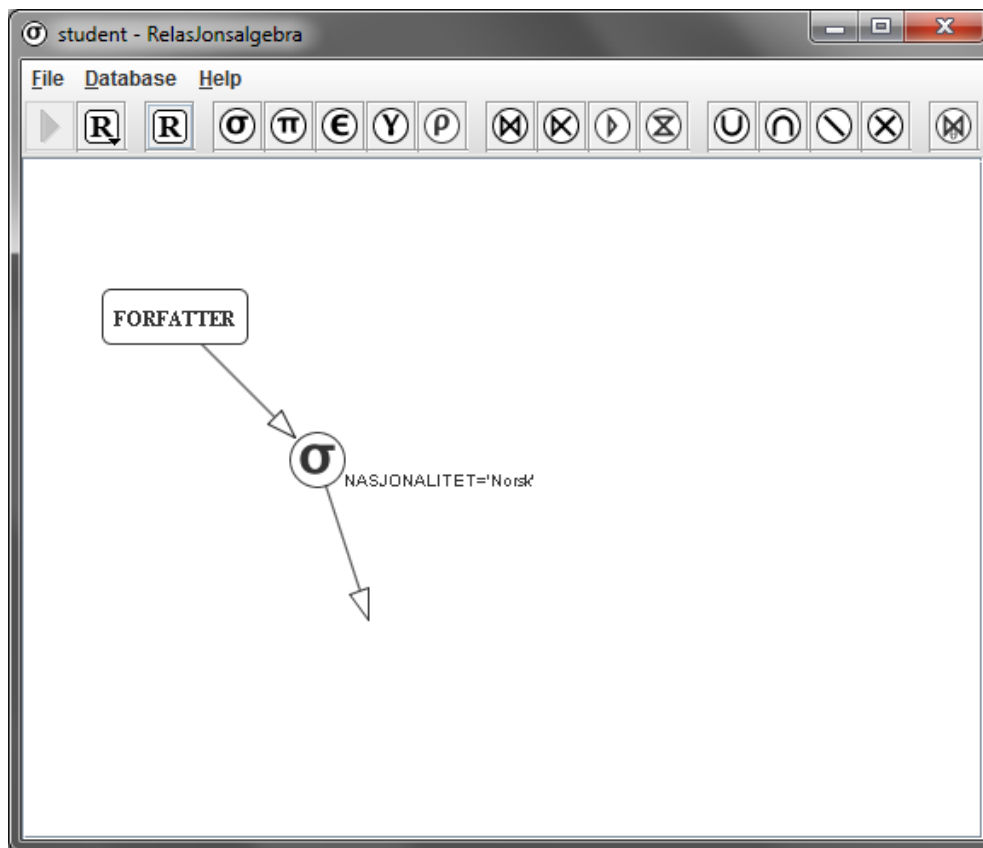
Våren 2015

Oppg1.

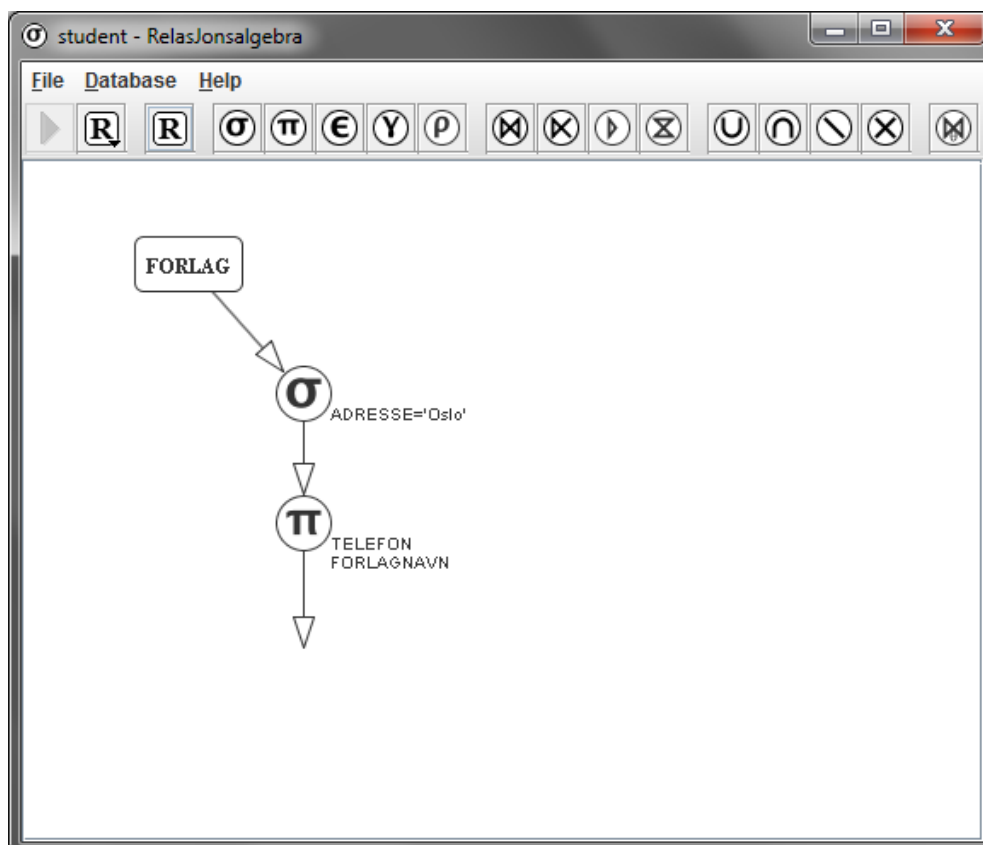
1)



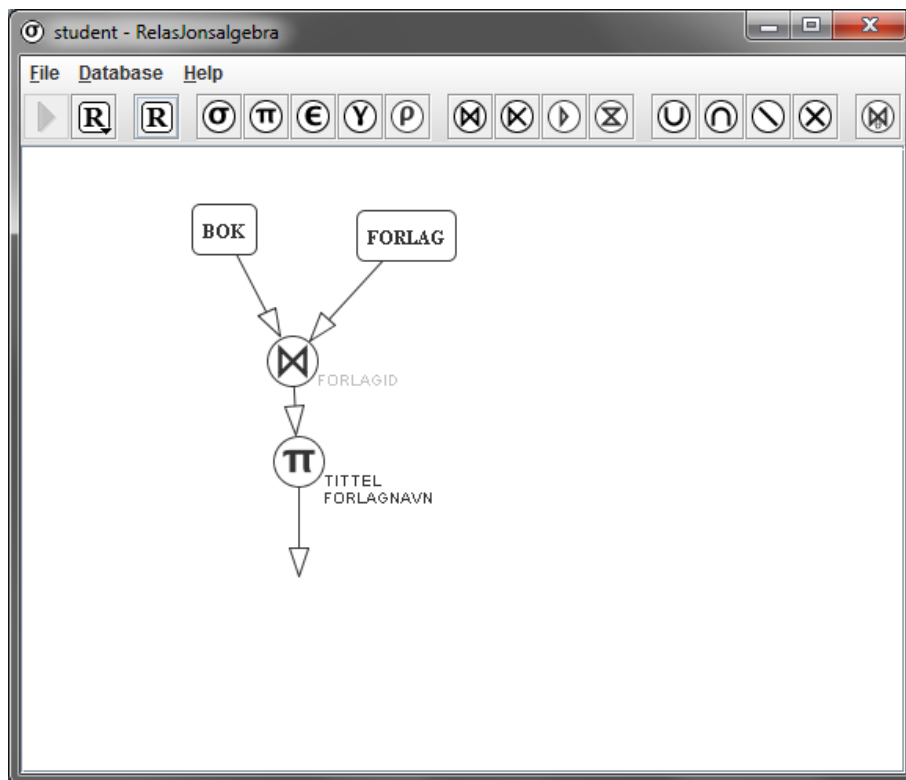
2)



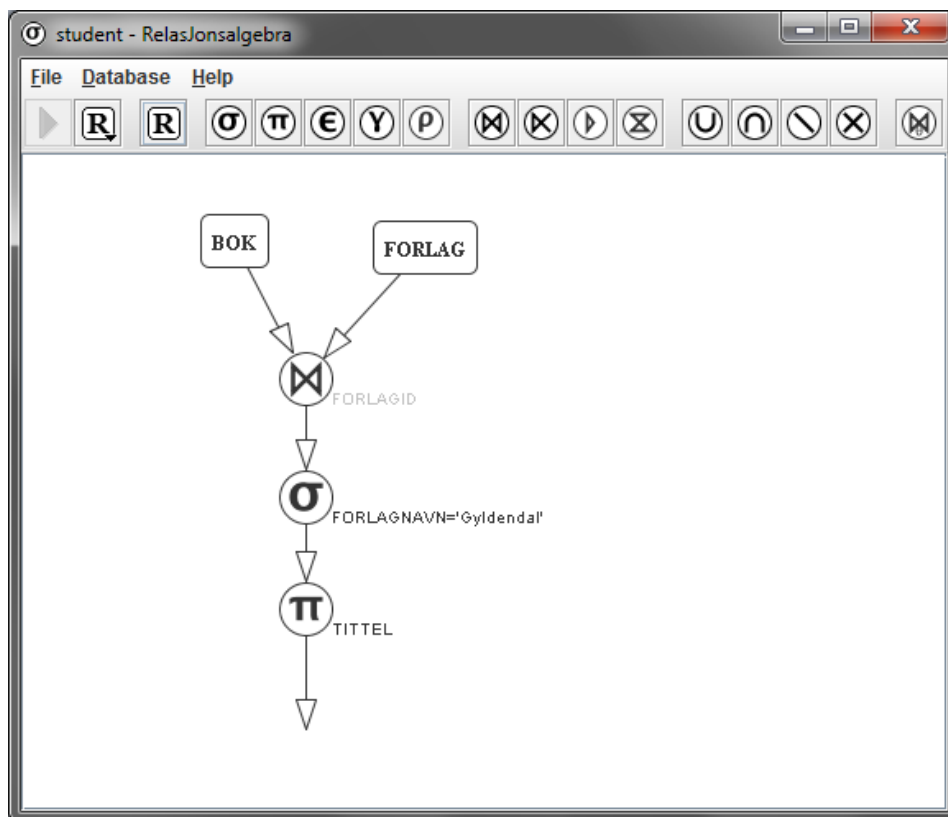
3)



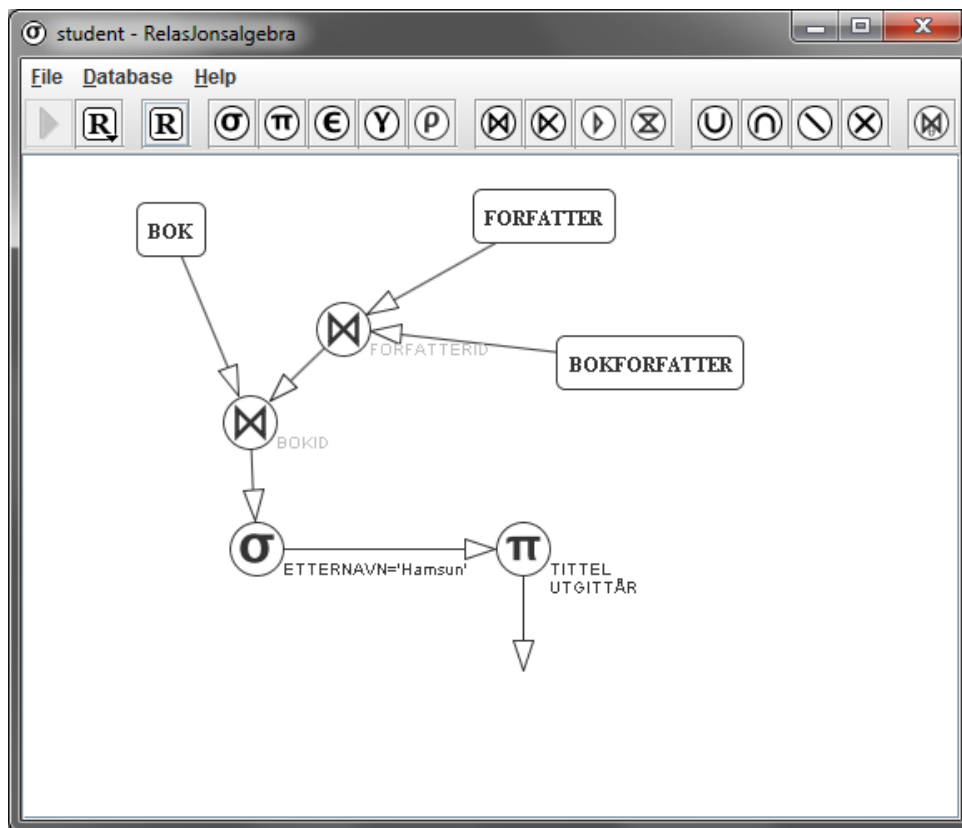
4)



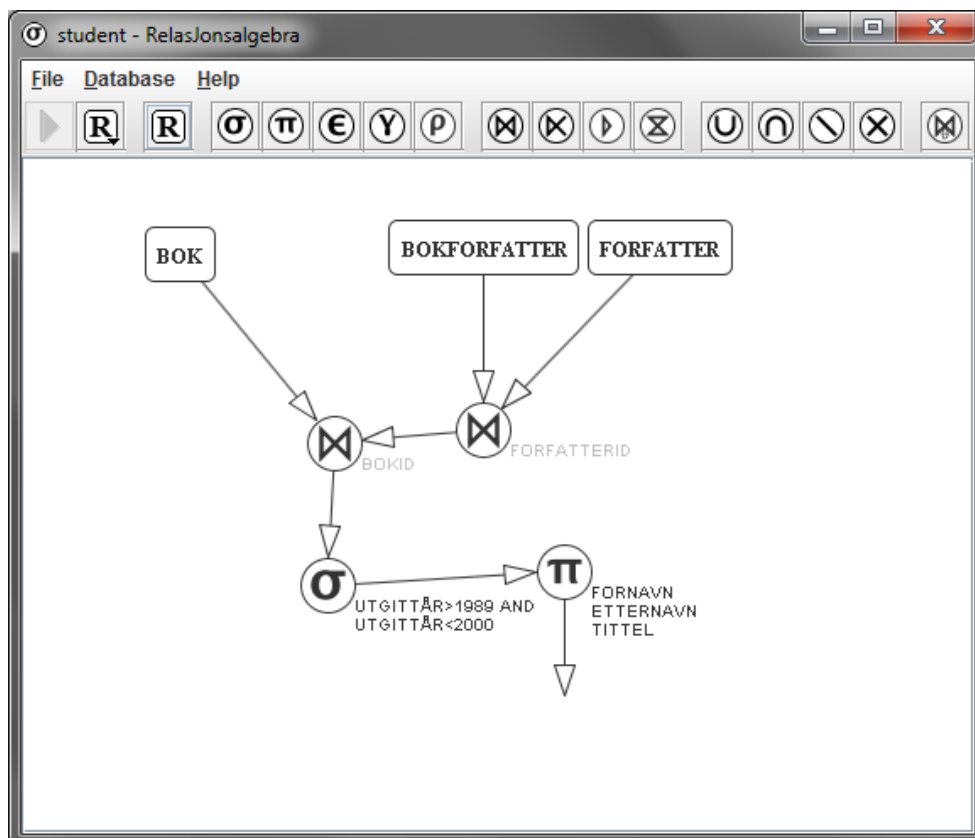
5)



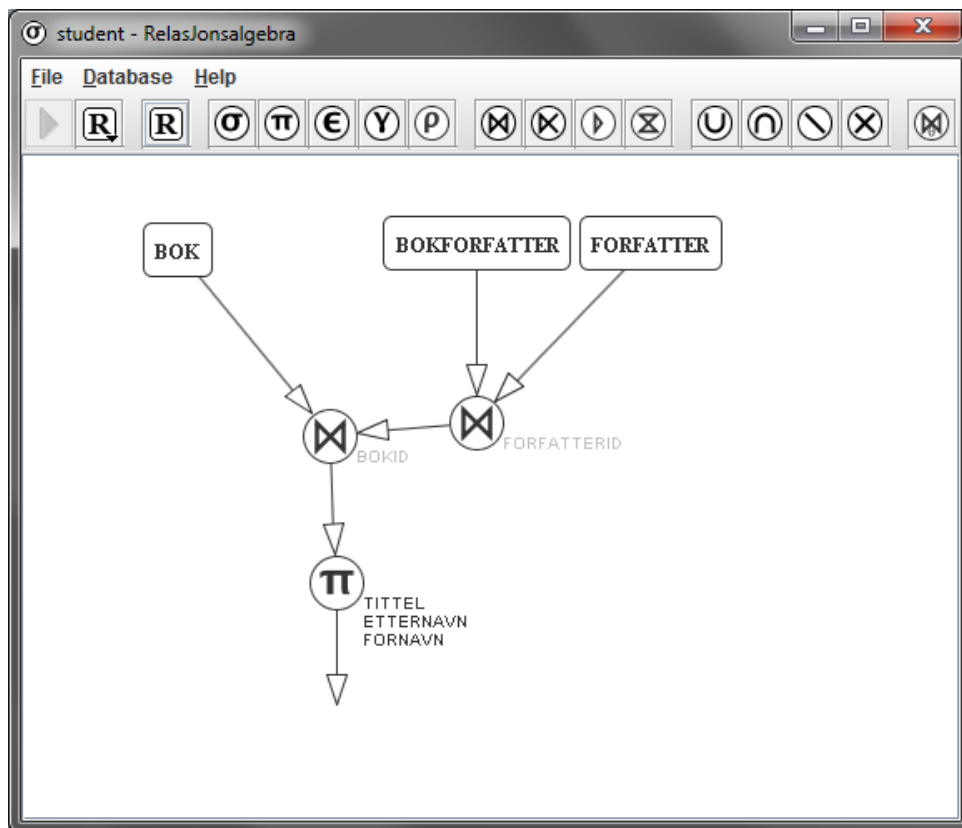
6)



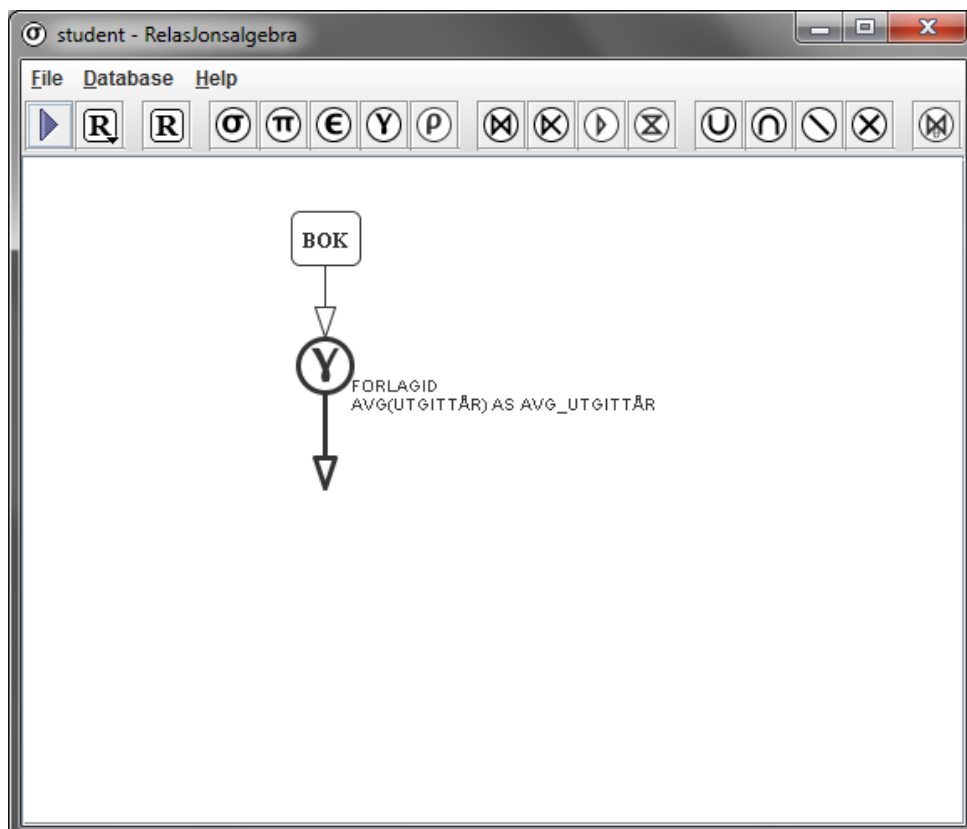
7)



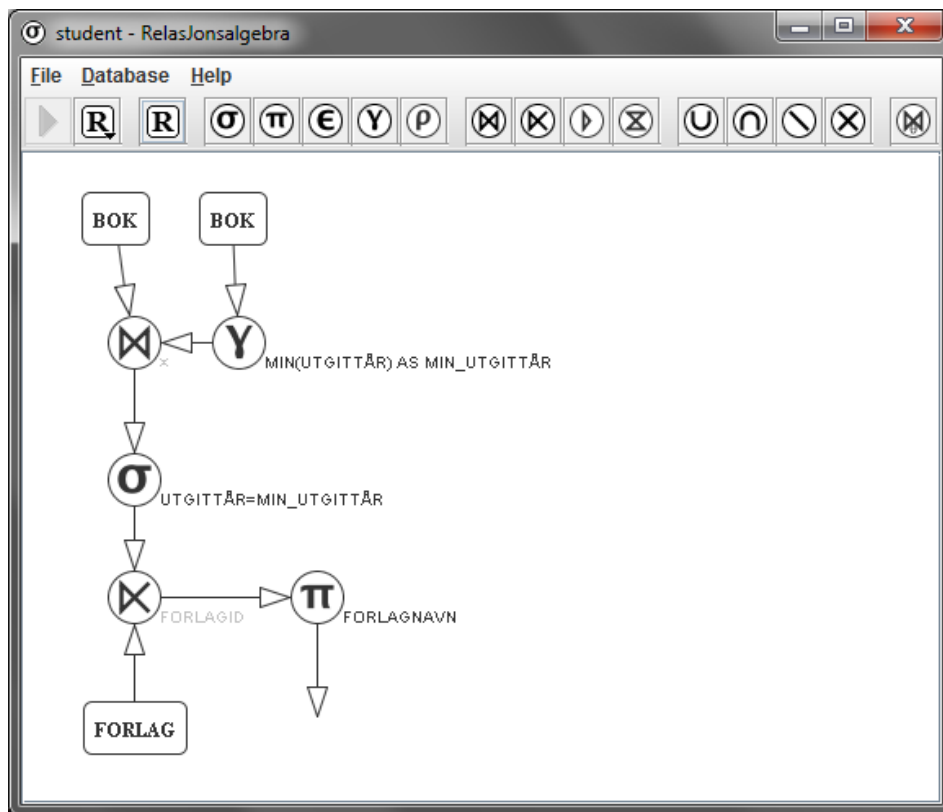
8)



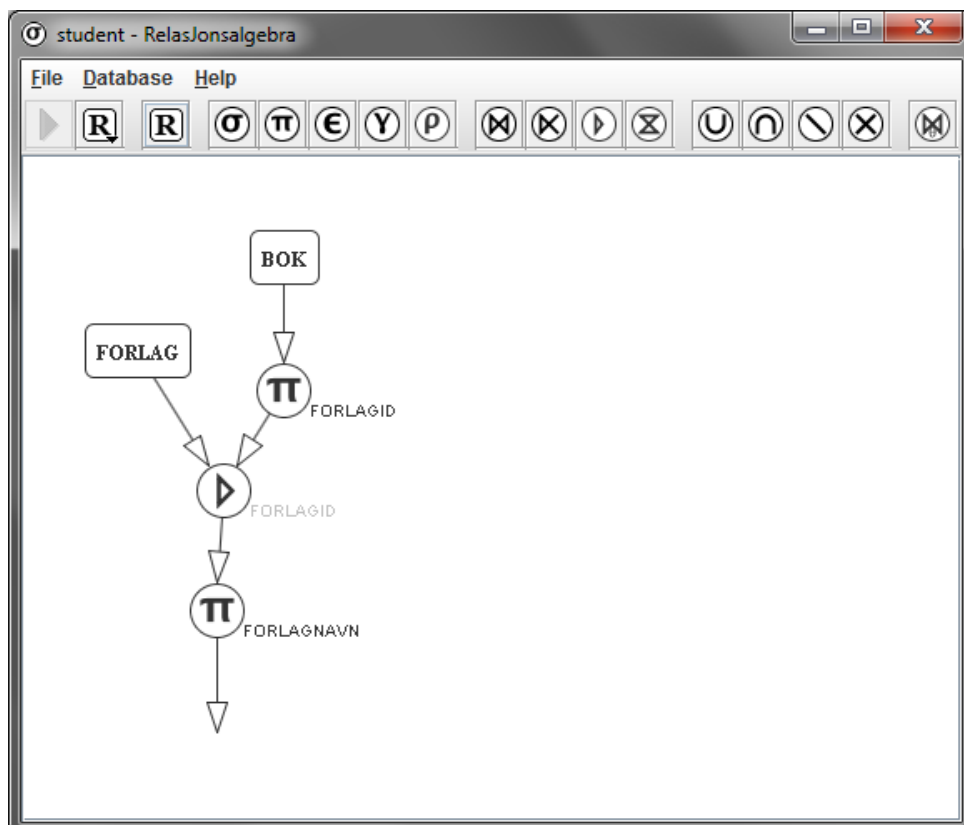
9)



10)



11)



Oppg2.

a)

```
create table KUNDE (KUNDENR int not null,  
NAVN varchar(255) not null,  
KREDITTGRENSE int not null,  
POSTNR int not null,  
primary key (KUNDENR),  
foreign key (POSTNR) references POSTSTED(POSTNR));
```

```
create table POSTSTED (POSTNR int not null,  
POSTSTED varchar(255) not null,  
primary key (POSTNR));
```

```
create table BESTILLING (ARTNR int not null,  
KUNDENR int not null,  
KVANTUM int not null,  
primary key (ARTNR, KUNDENR),  
foreign key (ARTNR) references ARTIKKEL(ARTNR),  
foreign key (KUNDENR) references KUNE(KUNDENR));
```

```
create table ARTIKKEL (ARTNR int not null,  
NAVN varchar(255) not null,  
ANT int not null,  
PRIS double not null,  
primary key (ARTNR),  
unique (NAVN));
```

b)

Vi legger til ON DELETE CASCADE ON UPDATE CASCADE til fremmednøkklene i tabellen BESTILLING. For postnummre legger vi til ON UPDATE CASCADE til KUNDE tabellen, men utelater ON DELETE CASCADE fordi vi vil beholde kunden selv om et poststed fjernes. Dette må da håndteres manuelt.

c)

```
create assertion KREDITTGRENSE check (not exist (  
select C.KUNDENR, C.KREDITTGRENSE, sum(B.KVANTUM*A.PRIS)  
from ARTIKKEL A, BESTILLING B, KUNDE C  
where B.ARTNR = A.ARTNR and B.KUNDENR = C.KUNDENR  
grouped by C.KUNDENR, C.KREDITTGRENSE  
having sum (B.KVANTUM*A.PRIS) > KREDITTGRENSE));
```

Oppg3.

a)

```
select TITTEL from bok;
```

b)

```
select * from FORFATTER  
where NASJONALITET='Norsk';
```

c)

```
select FORLAGID, FORLAGNAVN, TELEFON from FORLAG  
where ADRESSE='Oslo'  
ordered by FORLAGNAVN;
```

d)

```
select TITTEL, FORLAGNAVN from BOK, FORLAG  
where BOK.FORLAGID=FORLAG.FORLAGID;
```

e)

```
select TITTEL, UTGITTÅR from BOK, BOKFORFATTER, FORFATTER  
where BOK.BOKID=BOKFORFATTER.BOKID  
and BOKFORFATTER.FORFATTERID=FORFATTER.FORFATTERID  
and FORNAVN='Knut' and ETTERNAVN='Hamsun';
```


f)

```
select FORNAVN, ETTERNAVN, FØDEÅR
from FORFATTER
where ETTERNAVN like 'H%';
```

g)

```
select count(*) from FORLAG;
```

h)

```
select TITTEL, FORNAVN, ETTERNAVN, FORLAGNAVN
from BOK, FORLAG, FORFATTER, BOKFORFATTER
where BOK.BOKID=BOKFORFATTER.BOKID and BOK.FORLAGID=FORLAG.FORLAGID
and NASJONALITET='Britisk';
```

i)

```
select FORFATTER.FORNAVN, FORFATTER.ETTERNAVN, count(*)
from BOK, FORFATTER, BOKFORFATTER
where BOK.BOKID=BOKFORFATTER.BOKID
and BOKFORFATTER.FORFATTERID=FORFATTER.FORFATTERID
group by FORFATTER.FORNAVN, FORFATTER.ETTERNAVN
order by count(*) desc;
```

j)

```
select TITTEL, UTGITTÅR
from BOK
where UTGITTÅR=(select min(UTGITTÅR) from BOK);
```

k)

```
select F.FORLAGNAVN, count(*)
from FORLAG F, BOK B
where B.FORLAGID=F.FORLAGID
group by F.FORLAGNAVN having count(*)>2;
```

l)

```
select F.FORLAGNAVN from FORLAG F
where F.FORLAGID not in (
select F.FORLAGID from FORLAG F, BOK B where F.FORLAGID=B.FORLAGID);
```

Oppg4.**a)**

Hensikten med virtuelle tabeller er å representere dataene i en eller flere tabeller på en forenklet måte for å skjule unødvendig komplekstet. Oppdatering av virtuelle tabeller er problematisk fordi det er vanskelig å definere regler som entydig oppdaterer tabellene den virtuelle tabellen er basert på.

b)

```
create view PROJECT (PNAME, DNAME, NO_OF_EMPLS, TOTAL_HOURS) as
select PNAME, DNAME, count(*), sum(HOURS)
from PROJECT, DEPARTMENT,WORKS_ON
where DNUM = DNUMBER and PNUMBER = PNO
group by PNAME;
```

c)**Oppg5.****a)**

```
SELECT * FROM Supplier WHERE status>15;
```

b)

```
SELECT sname, s.city FROM Supplier s, SuppliesPart sp, Part p
WHERE s.sno=sp.sno AND sp.pno=p.pno AND p.pname='Screw';
```

c)

```
SELECT pno, pname FROM Part
WHERE pno IN (SELECT pno FROM SuppliesPart GROUP BY pno HAVING COUNT(*)>1);
```

d)

```
SELECT COUNT(*) FROM Supplier;
```

e)

```
SELECT s.city FROM Supplier s, Part p, SuppliesPart sp
WHERE s.sno=sp.sno AND sp.pno=p.pno AND p.weight>10;
```

f)

SELECT DISTINCT sname

WHERE sno NOT IN (

SELECT sp.sno FROM SuppliesPart sp, Part p WHERE sp.pno=p.pno AND p.pname='Screw')

ORDER BY sname;