Q1.

a.

$$\pi_{grade}(Grades \bowtie (\sigma_{age=20}Students) \bowtie (\sigma_{credits=4}Courses))$$

b.

$$\pi_{sname}((\pi_{sid}(Grades \bowtie (\sigma_{cname="Calculus"}Courses)) - \pi_{sid}(\sigma_{grade="C"}Grades)) \bowtie Students)$$

c.

$$\pi_{sname}\left((\pi_{cid}(\sigma_{credits=3}Courses)) \bowtie (\sigma_{grade="A"}Grades) \bowtie Students\right) \cup \pi_{sname}\left((\pi_{sid}(\sigma_{grade="B"}Grades)) \bowtie Students\right)$$

d.

$$\rho\left(TMP1, \left((\pi_{sid}((\sigma_{cname="Calculus"}Courses) \bowtie Grades)) \bowtie Students\right)\right)$$

$$\rho(TMP2, TMP1)$$

$$\rho(TMP3, \pi_{sid}TMP1 - \pi_{TMP1.sid}(TMP1 \bowtie_{TMP1.age<TMP2.age} TMP2))$$

$$\pi_{age}(TMP3 \bowtie Students)$$

Q2.

a.

```
SELECT DISTINCT S.age
FROM Students S, Courses C, Grades G
WHERE S.sid = G.sid AND G.cid = C.cid AND C.cname = 'CS310'
```

b.

```
SELECT S.sname
FROM Students S, Grades G
WHERE S.sid = G.sid AND S.sid NOT IN
        (SELECT G1.sid
        FROM Grades G1, Courses C
        WHERE G1.cid = C.cid AND C.credits <>4)
```

c.

```
SELECT G.cid, AVG(G.grade)
FROM Grades G
GROUP BY G.cid
HAVING 10 <= (
        SELECT COUNT(*)
        FROM Students S, Grades G1
        WHERE S.sid = G1.sid AND S.age >= 25 AND G1.cid = G.cid
        )
```

d.

```
SELECT S.sname
FROM Students S
WHERE NOT EXISTS (
        SELECT C.cid FROM Courses C WHERE C.credits = 4
        MINUS
        SELECT G.cid FROM Grades G WHERE G.sid = S.sid)
```

e.

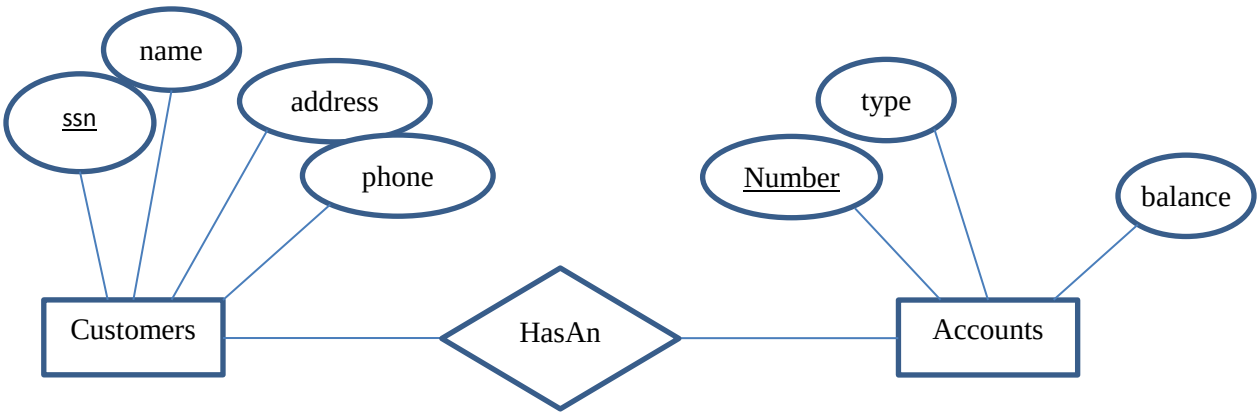
```
SELECT G.sid,G.cid
FROM Grades G
WHERE G.grade =
        (SELECT MAX(G1.grade) FROM Grades G1 where G1.cid=G.cid);
```

Q3.

```
CREATE VIEW TopStudents (ID, Name, GPA) AS
        SELECT S.sid, S.sname, AVG(G.grade)
        FROM Student S, Grades G
        WHERE S.sid = G.sid
        GROUP BY S.sid, S.sname
        HAVING AVG(G.grade) > 3.0;
```

Q4.

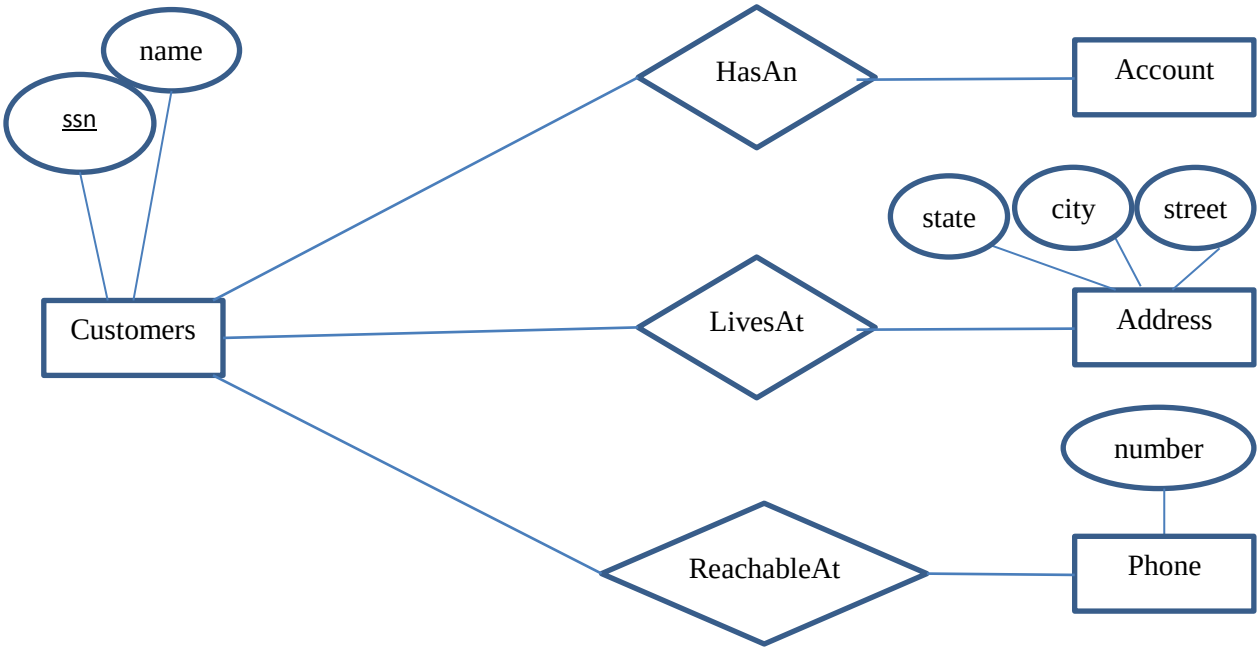
a)



b)

c)

d)



Q5.

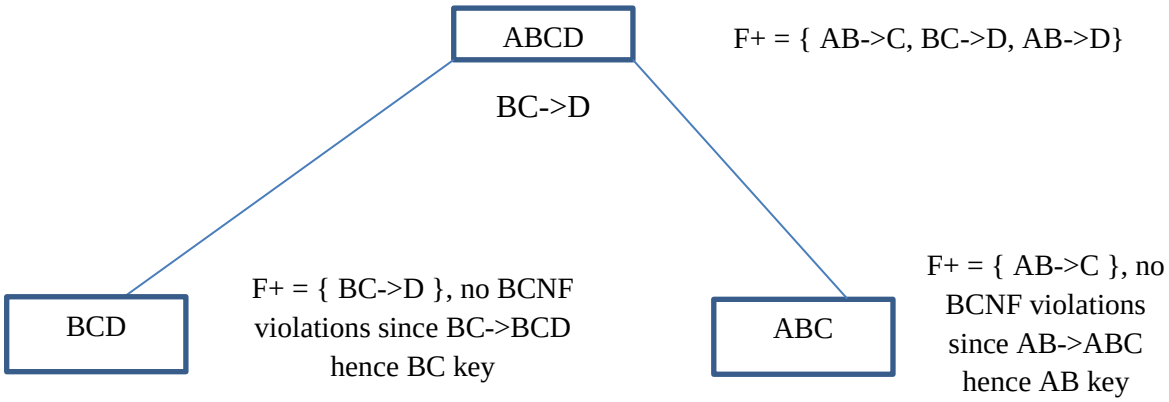
After building the table with attribute closure for each attribute combination, we get as key AB.

F+ is {AB->C, AB->D, BC->D}

First two FDs are not violations, because they contain AB on the LHS. Last one is a violation of BCNF as BC does not include a key, and also a violation of 3NF as D is not included in any key (only key is AB). So the relation is neither BCNF, nor 3NF.

Decomposition in BCNF

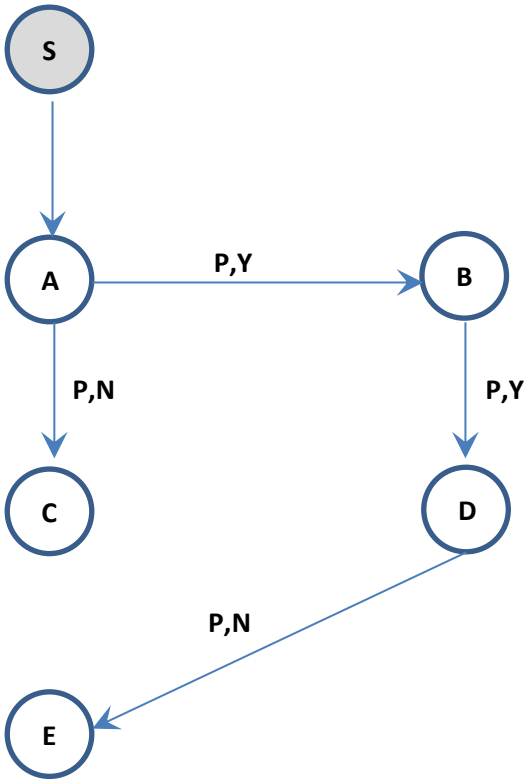
We decompose according to violating FD BC->D.



BCNF decomposition is ABC, BCD. It can be shown immediately that it is also dependency-preserving through transitivity.

Q6

After Step 4:



After Step 5:

C can still exercise P, E not.