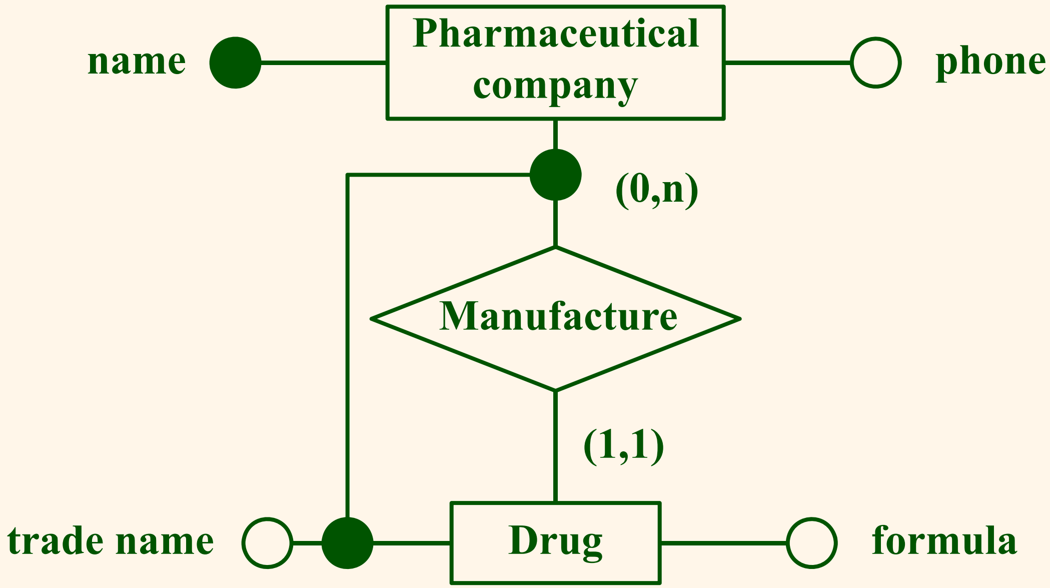
CS2102 Database Systems

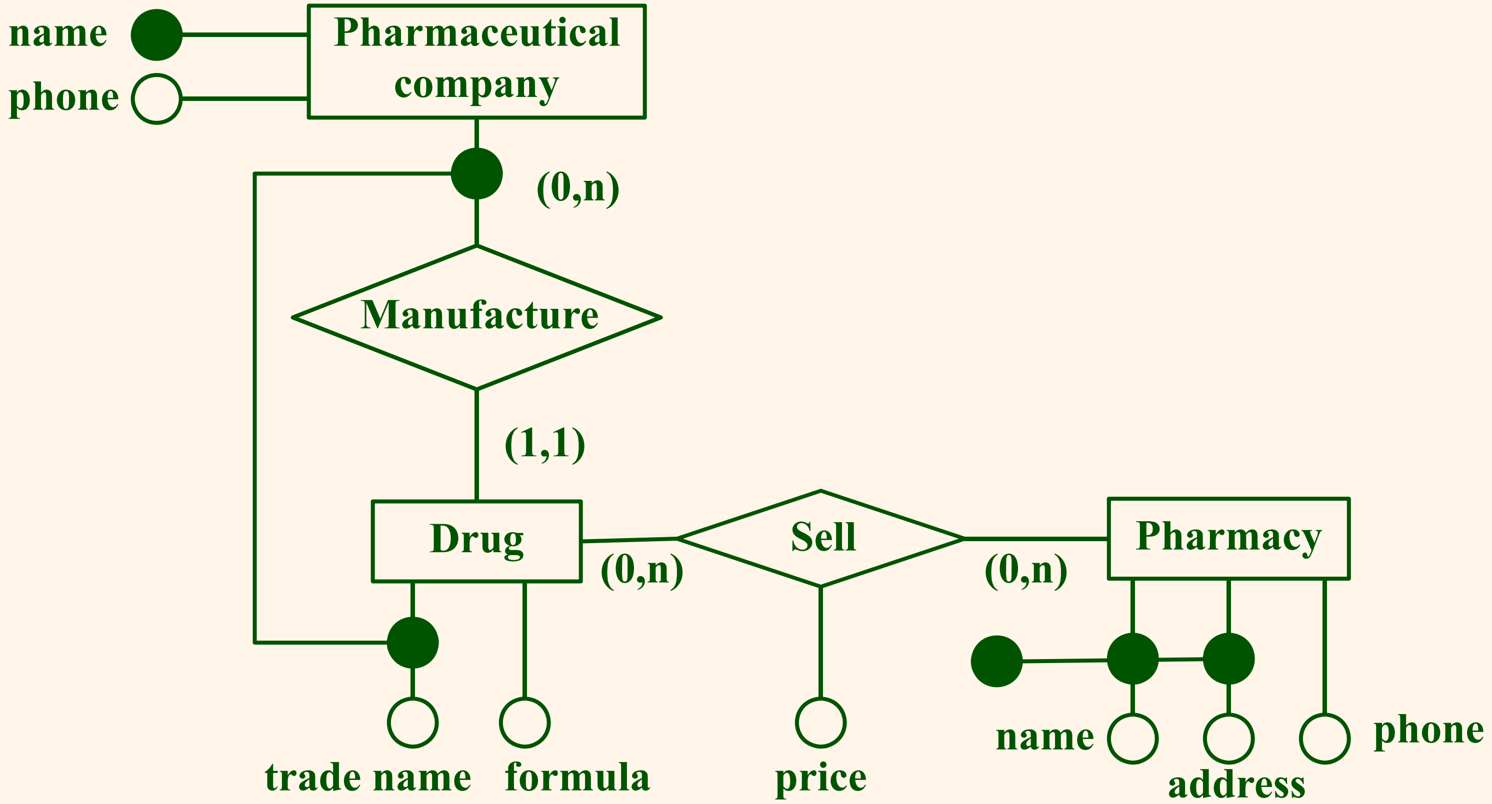
REVISION

ER Model

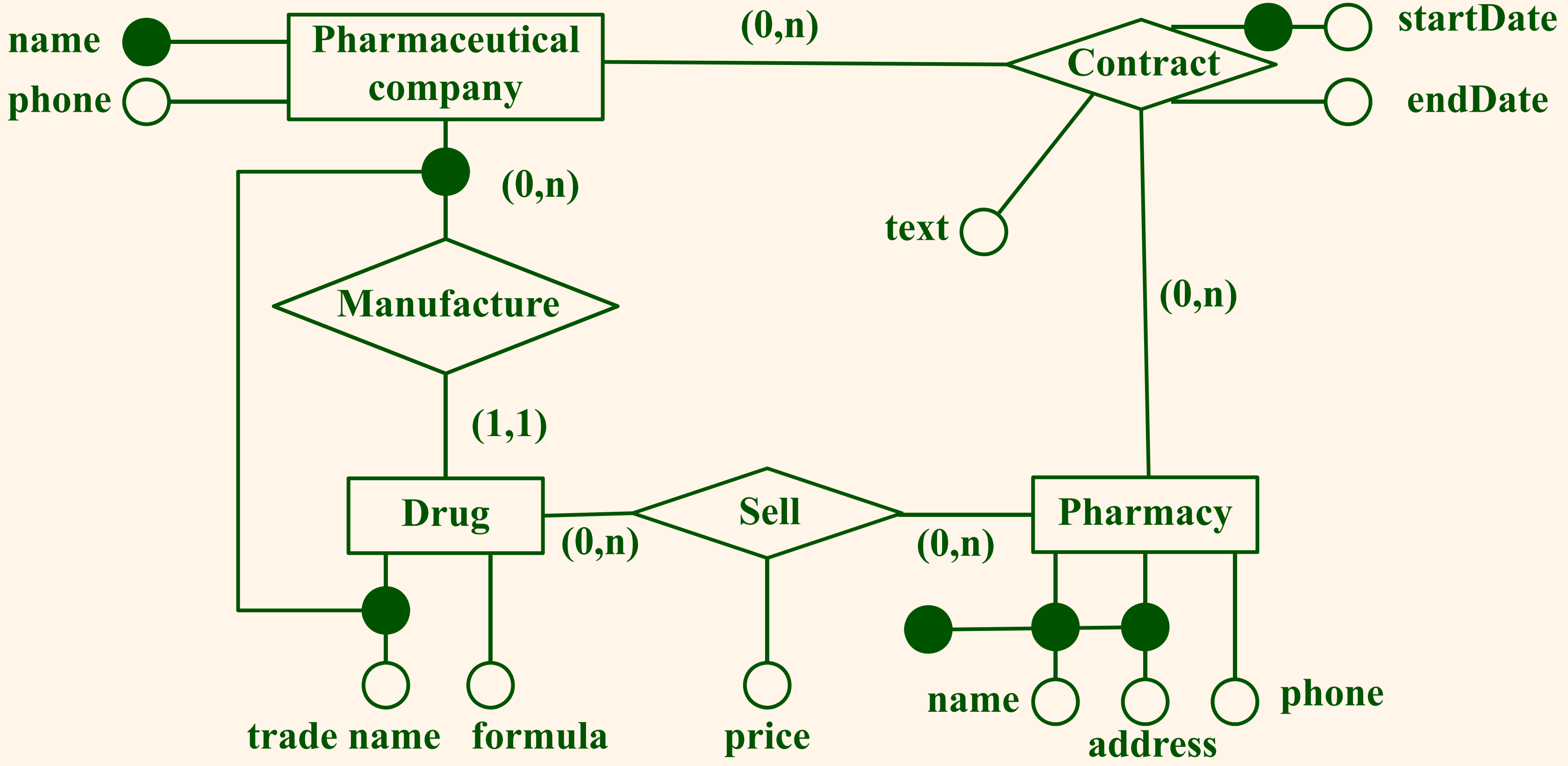
- Each pharmaceutical company is identified by name and has a phone number.
- For each drug, the trade name and formula must be recorded. Each drug is manufactured by a pharmaceutical company, and the trade name identifies a drug uniquely from the other products of that company. If a pharmaceutical company is deleted, we need not keep track of its products any more.



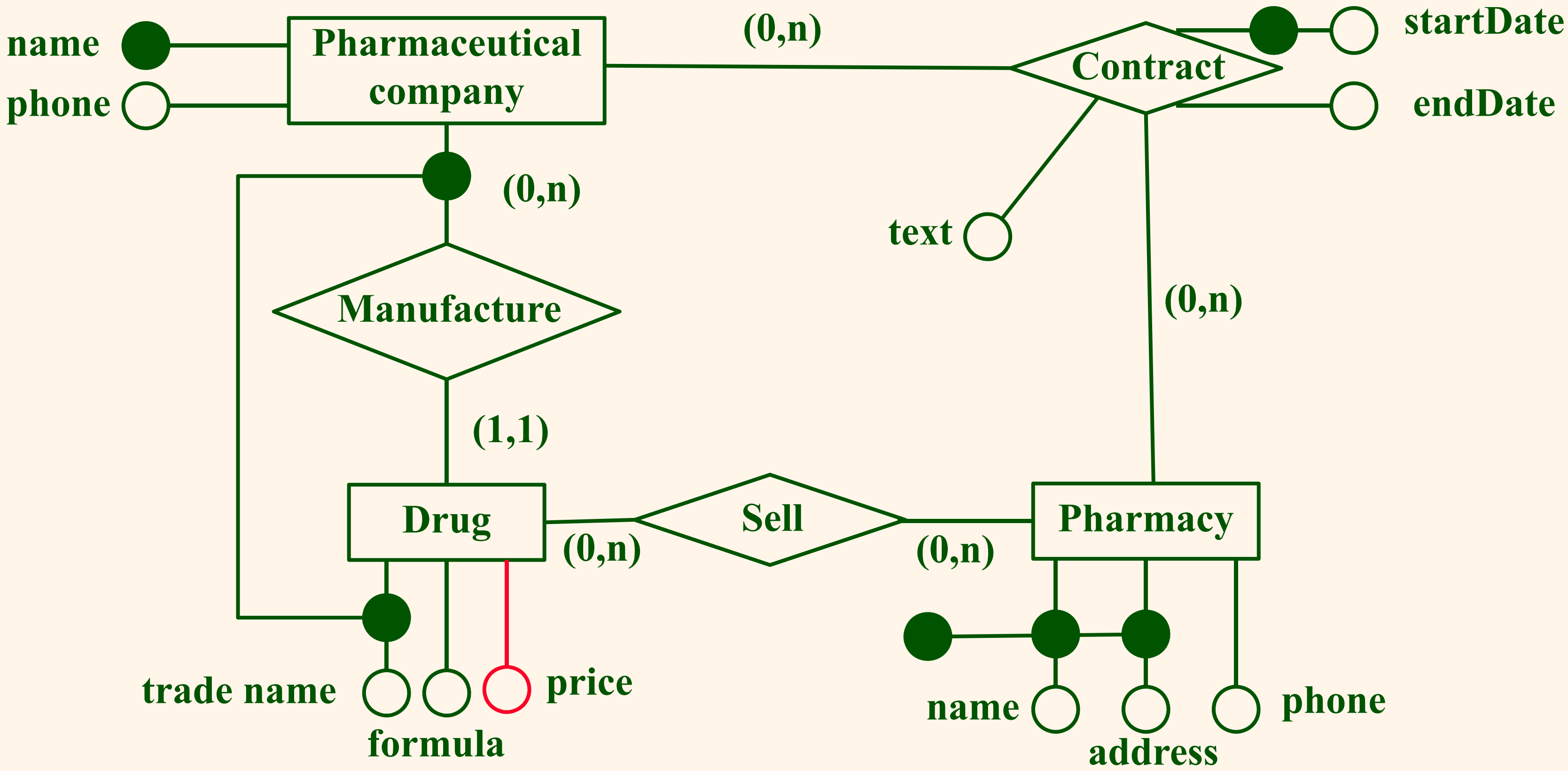
- Each pharmacy has a name, address and phone number.
- Each pharmacy sells several drugs and has a price for each. A drug could be sold at several pharmacies, and the price could vary from one pharmacy to another.



- Pharmaceutical companies have long term contracts with pharmacies. A pharmaceutical company can contract with several pharmacies, and a pharmacy can contract with several pharmaceutical companies. For each contract, you have to store a start date, an end date, and the text of the contract.



❖ If drugs are sold at a fixed price by all pharmacies:



SQL DDL

PharmaCompany(<u>cname, </u> cphone)

Drug(<u>cname, tname</u>, formula)

Pharmacy (<u>pname, address, </u> phone)

```
CREATE TABLE PharmaCompany (
        cname   VARCHAR(20) PRIMARY KEY ,
        cphone  VARCHAR(20))

CREATE TABLE Drug (
        cname   VARCHAR(20),
        tname   VARCHAR(20), formula VARCHAR(20),
        PRIMARY KEY (cname, tname),
        FOREIGN KEY (cname) REFERENCES PharmaCompany(cname))

CREATE TABLE Pharmacy (
        pname   VARCHAR(20),
        address VARCHAR(50),
        phone   VARCHAR(20),
        PRIMARY KEY (pname,address))
```

Sell (<u>cname, tname, pname, address</u>, price)
Contract (<u>cname, pname, startDate</u>, endDate,text)

```
CREATE TABLE Sell (
        cname   VARCHAR(20),  tname  VARCHAR(20),
        pname   VARCHAR(20), address VARCHAR(50),
        price   NUMERIC,
        PRIMARY KEY (cname, tname, pname, address),
        FOREIGN KEY (cname, tname) REFERENCES Drug(cname, tname),
        FOREIGN KEY (pname, address) REFERENCES
                                        Pharmacy(pname,address))
CREATE TABLE Contract (
        cname           VARCHAR(20),
        pname           VARCHAR(20), address VARCHAR(50),
        startDate       DATE, endDate DATE,
        text            VARCHAR(200),
        PRIMARY KEY (cname, pname, address, startDate),
        FOREIGN KEY (cname) REFERENCES PharmaCompany(cname),
        FOREIGN KEY (pname,address) REFERENCES
                                        Pharmacy(pname,address))
```

Relational Model

Part (pno, pname, cost, sname)

ComplexPart (pno, laborCost)

SubPart (pno, subPartOf, qty)

```
CREATE TABLE Part (   pno  NUMERIC PRIMARY KEY,
                      pname   VARCHAR(20),
                      cost      NUMERIC,
                      sname  VARCHAR(20) )

CREATE TABLE ComplexPart (
        pno NUMERIC PRIMARY KEY REFERENCES Part(pno),
        laborCost NUMERIC )

CREATE TABLE SubPart( pno NUMERIC,
        subPartOf NUMERIC,   qty NUMERIC,
        PRIMARY KEY (pno, subPartOf),
        FOREIGN KEY (pno) REFERENCES Part(pno),
        FOREIGN KEY (subPartOf) REFERENCES Part(pno) )
```

SQL

Part (pno, pname, cost, sname)

ComplexPart (pno, laborCost)

SubPart (pno, subPartOf, qty)

Q1. List the part numbers and part names of all basic parts whose cost is more than $10.

```
SELECT pno, pname
FROM Part
WHERE cost > 10
AND pno NOT IN (SELECT pno FROM ComplexPart);
```

Part (<u>pno</u>, pname, cost, sname)

ComplexPart (<u>pno</u>, laborCost)

SubPart (<u>pno, subPartOf,</u> qty)

Q2. Find all the pairs of complex parts that have the same labor cost.

```
SELECT P1.pno, P2.pno
FROM ComplexPart P1, ComplexPart P2
WHERE P1.pno <> P2.pno
AND P1.laborCost = P2.laborCost
```

Q3. Find the names of the suppliers that supplies at least two parts, with the average cost of these parts.

```
SELECT sname, AVG(cost)
FROM Part
GROUP BY sname
HAVING Count (*) >= 2;
```

Relational Algebra

Part (pno, pname, cost, sname)

ComplexPart (pno, laborCost)

SubPart (pno, subPartOf, qty)

Q4. List the names of suppliers who supply all complex parts whose labor cost is more than $100.

$$\pi_{\text{sname, pno}}(\text{Part}) \,/\, \pi_{\text{pno}}(\sigma_{\text{laborCost} > 100}(\text{ComplexPart}))$$

Q5. List the names of suppliers who supply at least two parts.

ρ (R1, Part)

ρ (R2, Part)

$$\pi_{\text{R1.sname}}(\sigma_{\text{R1.sname = R2.sname} \wedge \text{R1.pno <> R2.pno}}(\text{R1 x R2}))$$

Relational Calculus

Part (<u>Pno</u>, Pname, Cost, Sname)

ComplexPart (<u>Pno</u>, LaborCost)

SubPart (<u>Pno, SubPartOf,</u> Qty)

Q6. Find the name of the cheapest part.

TRC:

$\{ T \mid \exists P1 \in Part\ (\neg\ (\exists P2 \in Part\ (P2.Cost < P1.Cost)) \wedge$
$T.Pname = P1.Pname) \}$

DRC:

$\{ \langle X \rangle \mid \exists P, C, S\ (\langle P, X, C, S \rangle \in Part \wedge$
$\neg\ (\exists P2, X2, C2, S2\ (\langle P2, X2, C2, S2 \rangle \in Part \wedge C2 < C))) \}$

Part (<u>Pno</u>, Pname, Cost, Sname)

ComplexPart (<u>Pno</u>, LaborCost)

SubPart (<u>Pno, SubPartOf,</u> Qty)

Q7. Find the name of the cheapest **basic** part.

TRC:

$\{ T \mid \exists P1 \in Part (\neg (\exists P2 \in Part (P2.Cost < P1.Cost)) \wedge$
$\quad \neg (\exists P3 \in ComplexPart (P3.Pno = P1.Pno)) \wedge$
$\quad T.Pname = P1.Pname)\}$

DRC:

$\{ <X> \mid \exists P, C, S \ (<P, X, C, S> \in Part \wedge$
$\quad \neg (\exists P2, X2, C2, S2 (<P2, X2, C2, S2> \in Part \wedge C2 < C)) \wedge$
$\quad \neg (\exists P3, X3, C3, S3 (<P3, X3, C3, S3> \in ComplexPart \wedge P3 = P))) \}$

Part (<u>Pno</u>, Pname, Cost, Sname)

ComplexPart (<u>Pno</u>, LaborCost)

SubPart (<u>Pno, SubPartOf,</u> Qty)

Q8. List the part numbers that are first and second level subparts of part number p200.

TRC:

{ T | (∃ P1 ∈ SubPart (P1.SubPartOf = 'p200') ∧ T.pno = P1.pno) ∨
(∃ P2 ∈ SubPart ∃ P3 ∈ SubPart (P2.SubPartOf = P3.Pno ∧
P3.SubPartOf = 'p200') ∧ T.pno = P2.pno)}

DRC:

{<X> | ∃ Q (<X, 'p200', Q> ∈ SubPart) ∨
∃P, Q1, Q2 (<X, P, Q1> ∈ SubPart ∧ <P, 'p200', Q2> ∈ SubPart) }

Schema Refinement

AB→C, C→A, BC→D, ACD→B, D→E, D→G, BE→G, CG→B, CG→D, CE→A, CE→G

Q1. Find all the candidate keys of R

$A^+=$ **A**	$AB^+=$ **ABCDEG**	$DE^+=$ **DEG**	$ADE^+=$ **ADEG**
$B^+=$ **B**	$AC^+=$ **AC**	$DG^+=$ **DEG**	$ADG^+=$ **ADEG**
$C^+=$ **AC**	$AD^+=$ **ADEG**	$EG^+=$ **EG**	$AEG^+=$ **AEG**
$D^+=$ **DEG**	$AE^+=$ **AE**		$DEG^+=$ **DEG**
$E^+=$ **E**	$AG^+=$ **AG**		
$G^+=$ **G**	$BC^+=$ **ABCDEG**		$ADEG^+=$ **ADEG**
	$BD^+=$ **ABCDEG**		
	$BE^+=$ **ABCDEG**		
	$BG^+=$ **BG**		
	$CD^+=$ **ABCDEG**		
	$CE^+=$ **ABCDEG**		
	$CG^+=$ **ABCDEG**		

So, the candidate keys are:

AB, BC, BD, BE, CD, CE, CG

Q2. Find an extended minimal cover of F.

Step 1. Simplify the right hand side

N.A.

Q2. Find an extended minimal cover of F.

Step 2. Eliminate redundant attributes

$AB \rightarrow C$	$AB \rightarrow C$
$BC \rightarrow D$	$BC \rightarrow D$
$ACD \rightarrow B$	**$CD \rightarrow B$**
$BE \rightarrow G$	$BE \rightarrow G$
$CG \rightarrow B$	$CG \rightarrow B$
$CG \rightarrow D$	$CG \rightarrow D$
$CE \rightarrow A$	**$C \rightarrow A$**
$CE \rightarrow G$	$CE \rightarrow G$
$D \rightarrow E$	$D \rightarrow E$
$D \rightarrow G$	$D \rightarrow G$
$C \rightarrow A$	

Can we replace $AB \rightarrow C$ with $A \rightarrow C$ or $B \rightarrow C$?

No, since $A^+=\{A\}$ and $B^+=\{B\}$

Can we replace $CE \rightarrow A$ with $C \rightarrow A$ or $E \rightarrow A$?

Yes, since we have $C \rightarrow A$

Can we replace $ACD \rightarrow B$ with $AC \rightarrow B$ or $AD \rightarrow B$ or $CD \rightarrow B$ or $A \rightarrow B$ or $C \rightarrow B$ or $D \rightarrow B$?

Yes, since $CD^+=\{ABCDEG\}$

Q2. Find an extended minimal cover of F.

Step 3. Eliminate redundant functional dependencies

$AB \rightarrow C$
$BC \rightarrow D$
$CD \rightarrow B$
$BE \rightarrow G$
$CG \rightarrow B$
$CG \rightarrow D$
$C \rightarrow A$
$CE \rightarrow G$
$D \rightarrow E$
$D \rightarrow G$

Can we eliminate $AB \rightarrow C$?

Compute AB^+ by using the other FDs

$AB^+=\{AB\}$

Does not contain C. **NO!**

Q2. Find an extended minimal cover of F.

Step 3. Eliminate redundant functional dependencies

$AB \to C$
$BC \to D$
$\mathbf{CD \to B}$
$BE \to G$
$CG \to B$
$CG \to D$
$C \to A$
$CE \to G$
$D \to E$
$D \to G$

Can we eliminate $CD \to B$?

Compute CD^+ by using the other FDs

$CD^+=\{ABCDEG\}$

CD^+ contains B. **YES!**

Q2. Find an extended minimal cover of F.

Step 3. Eliminate redundant functional dependencies

$AB \rightarrow C$
$BC \rightarrow D$
$CD \rightarrow B$
$BE \rightarrow G$
$CG \rightarrow B$
$\mathbf{CG \rightarrow D}$
$C \rightarrow A$
$CE \rightarrow G$
$D \rightarrow E$
$D \rightarrow G$

Can we eliminate $CG \rightarrow D$?

Compute CG^+ by using the other FDs

$CG^+=\{ABCDEG\}$

CG^+ contains D. **YES!**

Q2. Find an extended minimal cover of F.

Step 3. Eliminate redundant functional dependencies

AB→C
BC→D
CD→B
BE→G
CG→B
CG→D
C→A
CE→G
D→E
D→G

AB→C
BC→D
BE→G
CG→B
C→A
CE→G
D→E
D→G

Minimal Cover

Q2. Find an extended minimal cover of F.

Step 4. Group all dependencies with the same left hand side into one

$AB \rightarrow C$	$AB \rightarrow C$
$BC \rightarrow D$	$BC \rightarrow D$
$BE \rightarrow G$	$BE \rightarrow G$
$CG \rightarrow B$	$CG \rightarrow B$
$C \rightarrow A$	$C \rightarrow A$
$CE \rightarrow G$	$CE \rightarrow G$
$\mathbf{D \rightarrow E}$	$\mathbf{D \rightarrow EG}$
$\mathbf{D \rightarrow G}$	
	Extended Minimal Cover

Q3. Is R in BCNF ?

$AB \rightarrow C$	**Super key**
$BC \rightarrow D$	**Super key**
$BE \rightarrow G$	**Super key**
$CG \rightarrow B$	**Super key**
$C \rightarrow A$	**??? Violate BCNF**
$CE \rightarrow G$	
$D \rightarrow E$	
$D \rightarrow G$	

The candidate keys are:

AB, BC, BD, BE, CD, CE, CG

R is not in BCNF.

Q4. Is R in 3NF ?

$AB \rightarrow C$	**Super key**
$BC \rightarrow D$	**Super key**
$BE \rightarrow G$	**Super key**
$CG \rightarrow B$	**Super key**
$C \rightarrow A$	**A is part of candidate key**
$CE \rightarrow G$	**Super key**
$D \rightarrow E$	**E is part of candidate key**
$D \rightarrow G$	**G is part of candidate key**

The candidate keys are:

AB, BC, BD, BE, CD, CE, CG

R is in 3NF.