MC302 – DBMS : Normalization

Goonjan Jain
Department of Applied Mathematics
Delhi Technological University

Overview

- DB design and normalization
  - Pitfalls of bad design
    - Redundancy – space, inconsistencies, updation anomalies
  - Decomposition
    - lossless join decomp.
    - dependency preserving
  - Normal forms

Decompositions

There are “bad” decompositions. Good ones are:

- lossless and
- dependency preserving

Decompositions – Lossy:

- R1(roll_no, grade, name, address)

Roll_no	Grade	Name	Address
2017/MC/24	A	Aman	Prime
2017/MC/24	A+	Aman	Prime
2017/MC/78	A	Rohit	Main

R2(course_id, grade)

Course_id	Grade
MC302	A
MC304	A+
MC306	A

Roll_no	Course_id	Grade	Name	Address
2017/MC/24	MC302	A	Aman	Prime
2017/MC/24	MC304	A+	Aman	Prime
2017/MC/78	MC306	A	Rohit	Main

Roll_no → *name, address*
Roll_no, course_id → grade

- can not recover original table with a join!

Lossy Decomposition

RollNo	Course_id	Grade	Name	address
24	MC302	A	Aman	Prime
24	MC306	A	Aman	Prime
24	304	A+	Aman	Prime
78	302	A	Rohit	Main
78	306	A	Rohit	Main

Decompositions

- Example of non-dependency preserving

S#	Address	Status
678	India	E
689	US	E
700	UK	A

S# → address, status
address → status

- Is it lossless?

S#	Address
678	India
689	US
700	UK

S# → address

S#	Status
678	E
689	E
700	A

S# → status

Decomposition Lossless

- Non additive join property – No ‘spurious tuples’ generated
- Definition:
- consider schema R, with FD “F”. R1, R2 is a lossless join decomposition of R if we **always** have:

$$r1 \bowtie r2 = r$$

- An easier criterion?

Decomposition - Lossless

- Theorem: lossless join decomposition if the joining attribute is a superkey in at least one of the new tables
- Formally:

$$R1 \cap R2 \rightarrow R1 \text{ or}$$

$$R1 \cap R2 \rightarrow R2$$

Decompositions – Lossless:

R1

Roll_no	Course_id	Grade
2017/MC/24	MC302	A
2017/MC/24	MC304	A+
2017/MC/78	MC306	A

R2

Roll_no	Name	Address
2017/MC/24	Aman	Prime
2017/MC/78	Rohit	Main

Roll_no, course_id → grade

roll_no → name, address

Roll_no	Course_id	Grade	Name	Address
2017/MC/24	MC302	A	Aman	Prime
2017/MC/24	MC304	A+	Aman	Prime
2017/MC/78	MC306	A	Rohit	Main

Roll_no → *name, address*
Roll_no, course_id → grade

Test for lossless decomposition

- Create an empty matrix
  - No. of rows = no. of sub-relations
  - No. of columns = no. of attributes in the original table
- For each row *i* and column *j*, mark ‘x’ if R_i contains attribute A_j
- For all FDs X $\rightarrow$ Y, if at least any two rows have ‘x’ in X column, set ‘x’ in the Y column of other rows.
- If any row is made up entirely of ‘x’, then the decomposition is lossless, else lossy.

Example 1

R {A, B, C, D, E, F}

F={A → B, C → DE, AC → F}

R1{B, E} R2{A, C, D, E, F}

	A	B	C	D	E	f
R1		X			X	
R2	x		x	x	x	x

No change in the matrix after applying FDs. Thus test fails and the decomposition is lossy.

Example 2

R {A, B, C, D, E, F}

F={A ⟶ B, C ⟶ DE, AC ⟶ F}

R1{A, B} R2{C, D, E} R3{A, C, F}

	A	B	C	D	E	f
R1	x	X				
R2			x	x	x	
R3	x		x			x

Example 2

R {A, B, C, D, E, F}

F={A → B, C → DE, AC → F}

R1{A, B} R2{C, D, E} R3{A, C, F}

	A	B	C	D	E	f
R1	x	X				
R2			x	x	x	
R3	x	x	x	x	x	x

Last row consists of all ‘x’, thus the decomposition is lossless.

Decompositions – Dependency Preservation

- informally: we don't want the original FDs to span two tables - counter-example:

S#	Address	Status
678	India	E
689	US	E
700	UK	A

S# → address, status
address → status

S#	Address
678	India
689	US
700	UK

S# → address

S#	Status
678	E
689	E
700	A

S# → status

Decompositions – Dependency Preservation

- Dependency preserving decomposition

S#	Address	Status
678	India	E
689	US	E
700	UK	A

S# → address, status
address → status

S#	Address
678	India
689	US
700	UK

S# → address

Address	Status
India	E
US	E
UK	A

Address → status

(But S# → status?)

Normal Forms

Why is normalization needed?

- Process of efficiently organizing data in a database that is based on the concepts of Normal Forms.
- A relational table is said to be in a normal form if it satisfies a certain set of constraints.

- **Objectives of Normalization**
  - Free relations from undesirable insertion, update and deletion dependencies.
  - Increase life span of application programs

Levels of Normalization

- In total, six Normal Forms are defined but Third Normal Form (3NF) is considered sufficient for most business database design purposes.

Insertion Anomaly

- when certain attributes cannot be inserted into the database without the presence of other attributes.
- E.g. Student (roll_no, courseid, name, address, course_title)

Roll_no	Courseid	Name	Address	Course_title
2K19/MSMA/78	MSMA312	Ajay	Delhi	Literature
2K19/MSMA/78	MSMA453	Ajay	Delhi	Psychology
2K19/MSMA/90	MSMA312	Seema	Delhi	Literature

Deletion Anomaly

- A **deletion anomaly** occurs when you **delete** a record that may contain attributes that shouldn't be deleted.
- E.g. Student (roll_no, courseid, name, address, course_title)

Roll_no	Courseid	Name	Address	Course_title
2K19/MSMA/78	MSMA312	Ajay	Delhi	Literature
2K19/MSMA/78	MSMA453	Ajay	Delhi	Psychology
2K19/MSMA/90	MSMA756	Seema	Delhi	Philosophy
2K19/MSMA/93	MSMA312	Geeta	Gurgaon	Literature

Roll_no	Courseid	Name	Address	Course_title
2K19/MSMA/78	MSMA312	Ajay	Delhi	Literature
2K19/MSMA/78	MSMA453	Ajay	Delhi	Psychology
2K19/MSMA/93	MSMA312	Geeta	Gurgaon	Literature

Updation Anomaly

- An **update anomaly** is a data inconsistency that results from data redundancy and a partial **update**.

Roll_no	Courseid	Name	Address	Course_title
2K19/MSMA/78	MSMA312	Ajay	Delhi	Literature
2K19/MSMA/78	MSMA453	Ajay	Delhi	Psychology
2K19/MSMA/90	MSMA756	Seema	Delhi	Philosophy
2K19/MSMA/93	MSMA312	Geeta	Gurgaon	Literature

First Normal Form (1NF)

- A table conforms to First Normal Form (1NF) when:
  - Domain of each attribute contains only atomic values, and
  - Value of each attribute contains only a single value from that domain.

  Cell does not contain either multivalued or composite values.

Emp_No	Name	Dependents
E103	Rahul	Raghav Seema
E134	Amit	Anil Manoj Geeta

NOT 1NF

1NF – 3 options

Option 1:

Emp_No	Name	Dependents
E103	Rahul	Raghav
E103	Rahul	Seema
E134	Amit	Anil
E134	Amit	Manoj
E134	Amit	Geeta

Option 2:

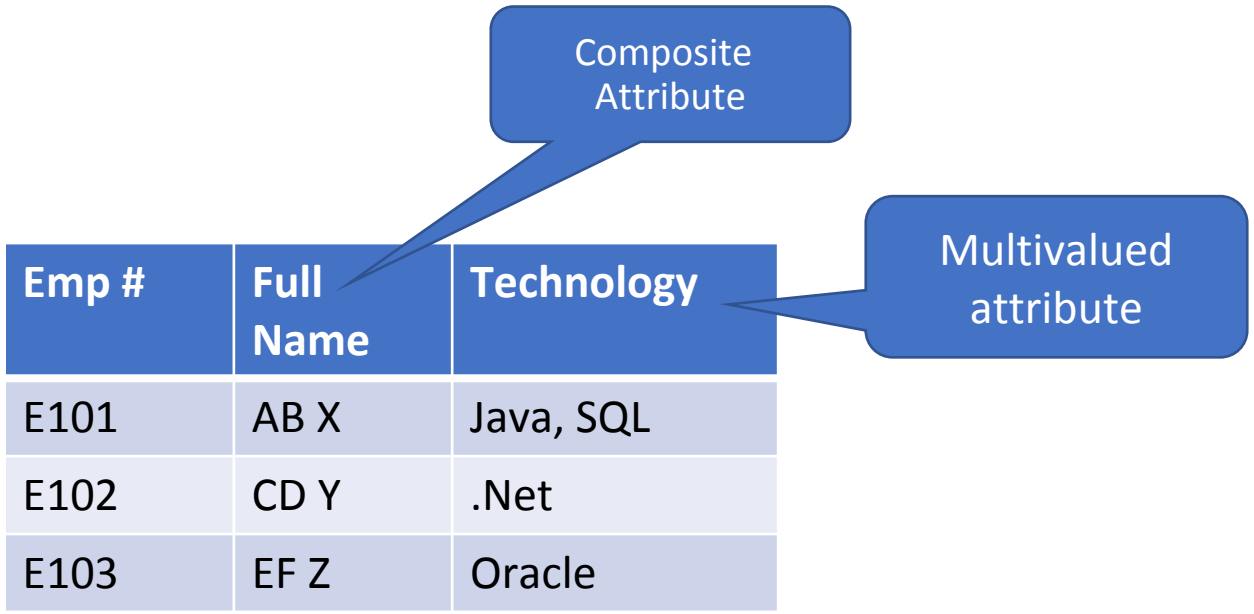
Emp_NO	Name	Dependent1	Dependent2	Dependent3
E103	Rahul	Raghav	Seems	
E134	Amit	Anil	Manoj	Geeta

Option 3:

Emp_No	Name
E103	Rahul
E134	Amit

Emp_No	Dependents
E103	Raghav
E103	Seema
E134	Anil
E134	Manoj
E134	Geeta

First Normal Form (1NF)



Emp #	Full Name	Technology
E101	AB X	Java, SQL
E102	CD Y	.Net
E103	EF Z	Oracle

Emp #	First Name	Last Name
E101	AB	X
E102	CD	Y
E103	EF	Z

Emp #	Technology
E101	Java
E101	SQL
E102	.Net
E103	Oracle

1NF to 2NF

- An relational table is in Second Normal Form (2NF), if:
  - It is in 1NF, and
  - No non-prime attribute is dependent on any proper subset of any candidate key of the table.

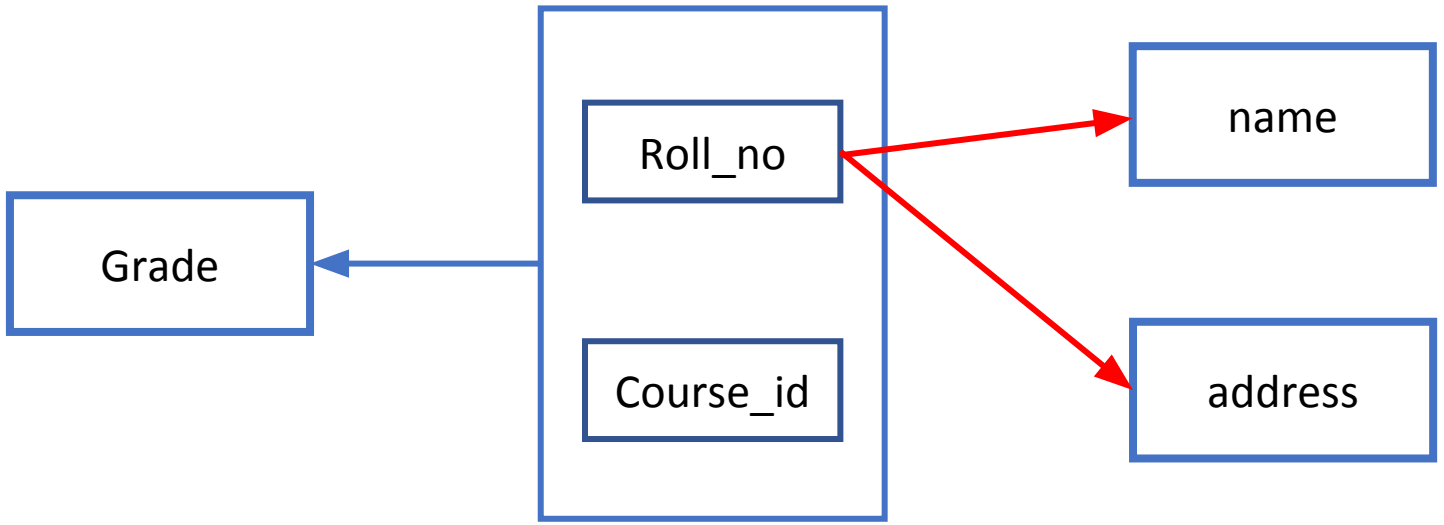
- **Steps**:
  - Identify primary key for the 1NF relation.
  - Identify functional dependencies in the relation.
  - If partial dependencies exist on the primary key remove them by placing them in a new relation along with copy of their determinant.

2NF

counter-example:

Table1(roll_no, course_id, grade, name, address)

roll_no, course_id → grade Roll_no → name, address



2NF

Roll_no	Course_id	Grade
2017/MC/24	MC302	A
2017/MC/24	MC304	A+
2017/MC/78	MC306	A

Roll_no, course_id → grade

Roll_no	Name	Address
2017/MC/24	Aman	Prime
2017/MC/78	Rohit	Main

roll_no → name, address

Second Normal Form (2NF)

Emp #	Project #	Project Loc
E101	P101	India
E102`	P102	Australia
E103	P101	India

Partially dependent on Primary key {emp #, project #}

Emp #	Project #
E101	P101
E102	P102
E103	P103

Project #	Project Loc
P101	India
P102	Australia

Third Normal Form(3NF)

- A relational table is in 3NF if
  - It is in 2NF, and
  - All the attributes in a table are determined only by the candidate keys of that table and not by any non-prime attributes.
- Reduces duplication of data and ensures referential integrity.
- No Transitive dependency.

- Formally, a rel. R with FDs “F” is in 3NF if:
- for every $a \rightarrow b$ in F:
  - it is trivial (*a* superset of *b*) or
  - *a* is a superkey or
  - *b*: part of a candidate key

Third Normal Form (3NF)

Emp # -> ZipCode,
Zip Code -> City

Emp No	House No	City	Zip Code
E101	12/1	A	123
E102	23/43	B	456
E103	67/5	A	123

Emp No	House No	Zip Code
E101	12/1	123
E102	23/43	456
E103	67/5	123

Zip Code	City
A	123
B	456

3NF

how to bring a schema to 3NF?

First one:

- start from ER diagram and turn to tables
- then we have a set of tables R1, ... Rn which are in 3NF
- for each FD (X → A) in the cover that is not preserved, create a table (X,A)

Second one (“synthesis”)

- take all attributes of R
- for each FD (X → A) in the cover, add a table (X,A)
- if not lossless, add a table with appropriate key

3NF

Example:

R: ABC

F: $A \rightarrow B$, $C \rightarrow B$

Q1: what is the cover?

Q2: what is the decomposition to 3NF?

3NF

Example:

R: ABC

F: A → B, C → B

Q1: what is the cover?

A1: ‘F’ is the cover

Q2: what is the decomposition to 3NF?

A2: R1(A,B), R2(C,B), ... [is it lossless??]

3NF

Example:

R: ABC

F: A → B, C → B

Q1: what is the cover?

A1: ‘F’ is the cover

Q2: what is the decomposition to 3NF?

A2: R1(A,B), R2(C,B), R3(A,C)

Boyce Codd Normal Form (BCNF)

- Definition: Relation R is in BCNF w.r.t F, if
  - informally: everything depends on the full key, and nothing but the key
  - semi-formally: every determinant (of the cover) is a candidate key

- Formally: for every FD $a \rightarrow b$ in F
  - $a \rightarrow b$ is trivial or
  - a is a superkey
- R with only 2 attributes is automatically in BCNF.

BCNF

- Example and counter example

Roll_no	Name	Address
2017/MC/24	Aman	Prime
2017/MC/24	Aman	Prime
2017/MC/78	Rohit	Main

Roll_no → *name, address*

Roll_no	Course_id	Grade	Name	Address
2017/MC/24	MC302	A	Aman	Prime
2017/MC/24	MC304	A+	Aman	Prime
2017/MC/78	MC306	A	Rohit	Main

Roll_no → *name, address*
Roll_no, course_id → grade

BCNF

- Theorem: given a schema R and a set of FD "F", we can always decompose it to schemas R1, … Rn, so that
  - R1, … Rn are in BCNF and
  - the decompositions are lossless.

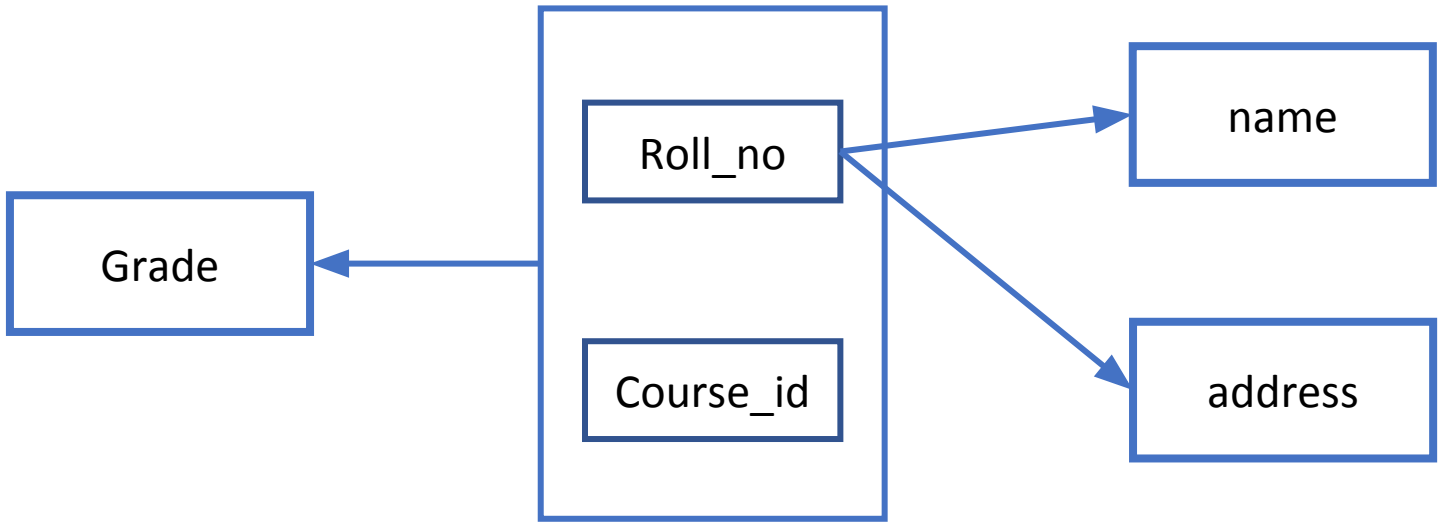
(but, some decomp. might lose dependencies)

- For a relation R
  - for every FD X → A that violates BCNF, decompose to tables (X,A) and (R-A)
  - repeat recursively

BCNF

e.g. - Table1(roll_no, course_id, grade, name, address)

roll_no, course_id → grade Roll_no → name, address



BCNF

R1

Roll_no	Course_id	Grade
2017/MC/24	MC302	A
2017/MC/24	MC304	A+
2017/MC/78	MC306	A

R2

Roll_no	Name	Address
2017/MC/24	Aman	Prime
2017/MC/24	Aman	Prime
2017/MC/78	Rohit	Main

Roll_no, course_id → grade

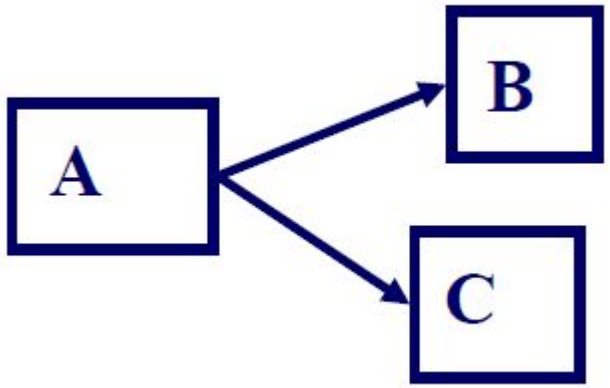
roll_no → name, address

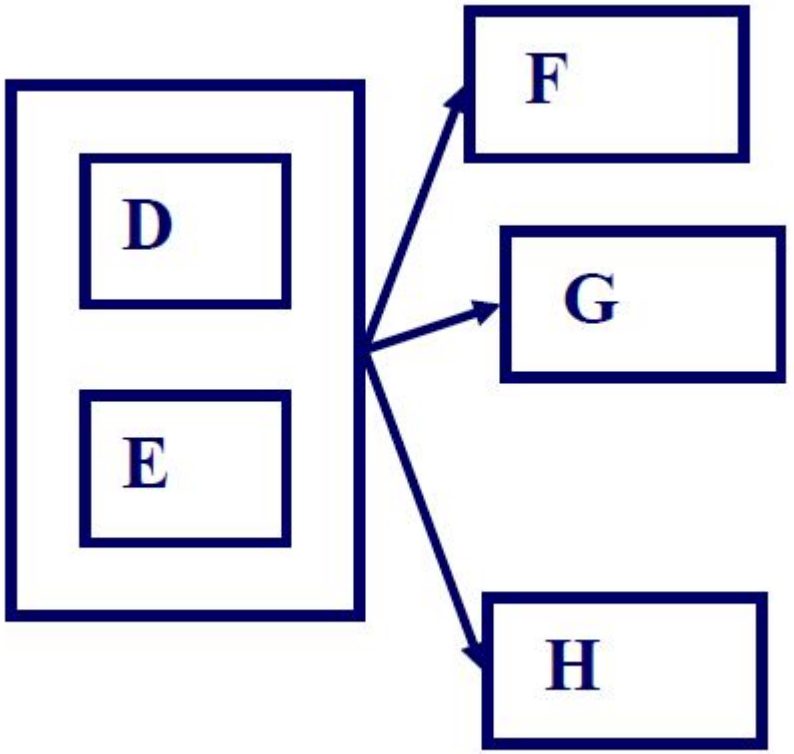
Roll_no	Course_id	Grade	Name	Address
2017/MC/24	MC302	A	Aman	Prime
2017/MC/24	MC304	A+	Aman	Prime
2017/MC/78	MC306	A	Rohit	Main

Roll_no → *name, address*
Roll_no, course_id → grade

BCNF

- Pictorially, we want a ‘star’ shape

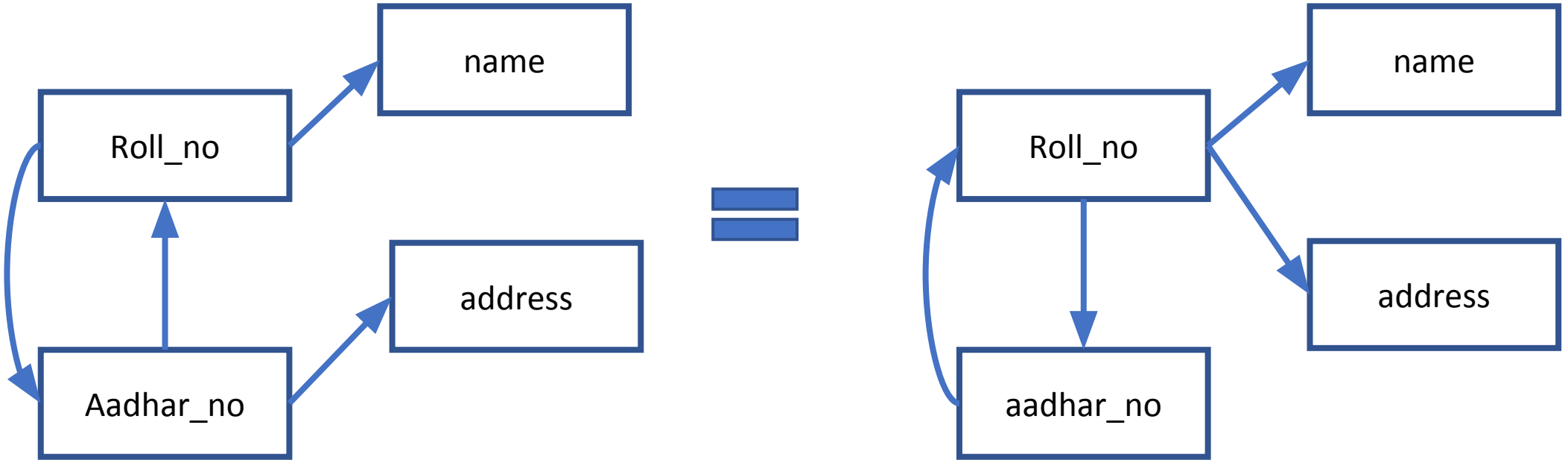




BCNF

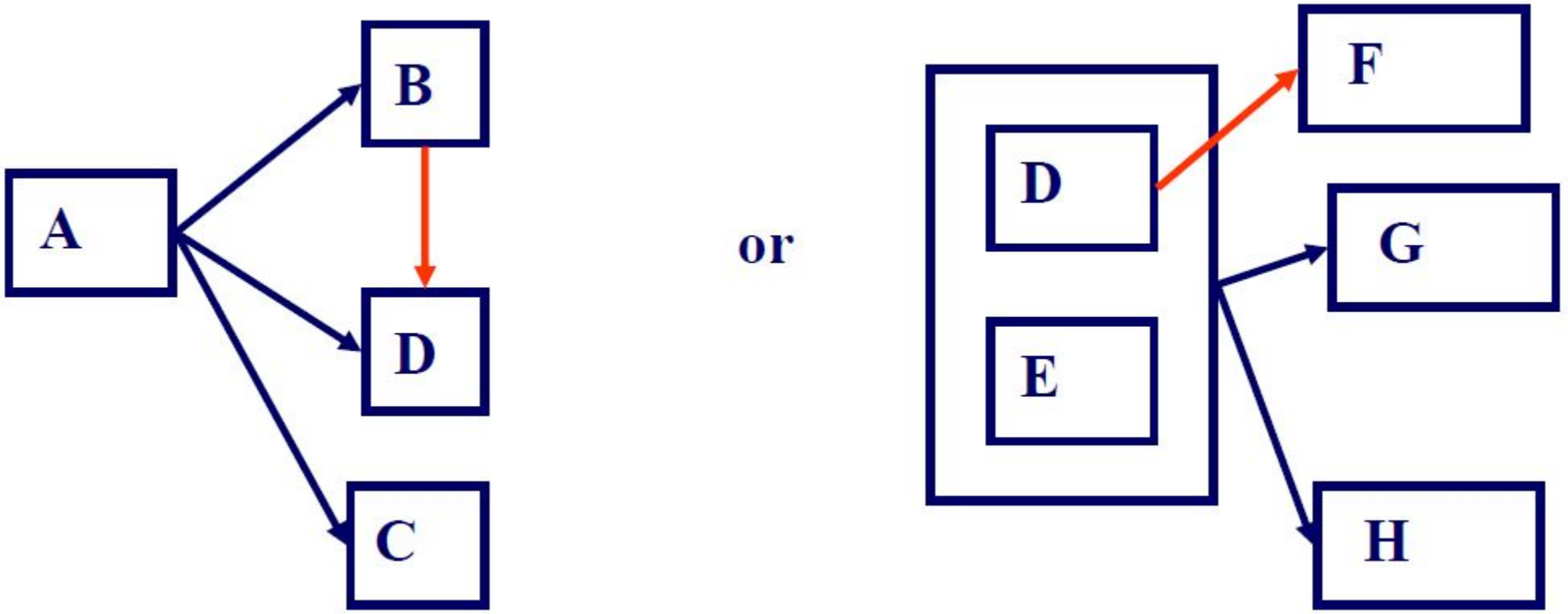
- Or a ‘star’-like (e.g. 2 candidate keys)

  STUDENT(roll_no, aadhar_no, name, address)



BCNF

- But not



3NF vs BCNF

If ‘R’ is in BCNF, it is always in 3NF (but not the reverse)

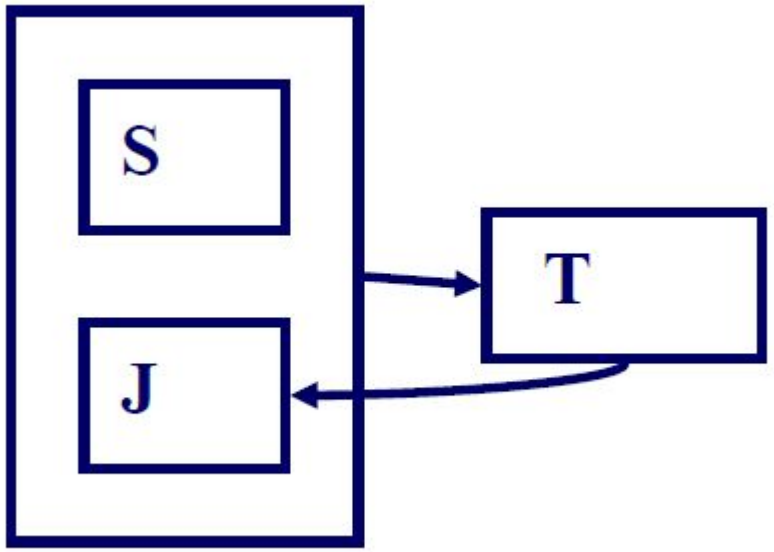
for A → B:

- 3NF:
  if B is a primary-key attribute and A is not a candidate key.
- BCNF:
  A must be candidate key

- In practice, aim for
  - BCNF; lossless join; and dependency preservation
- if impossible, we accept
  - 3NF; but insist on lossless join and dependency preservation

3NF vs BCNF

- consider the “classic” case:
- STJ(Student, Teacher, subJect)
- T → J
- S,J → T
- is it BCNF?



3NF vs BCNF

STJ(Student, Teacher, subJect)

T → J S, J → T

1) R1(T,J) R2(S,J)

(BCNF? - lossless? - dep. pres.?)

2) R1(T,J) R2(S,T)

(BCNF? - lossless? - dep. pres.?)

3NF vs BCNF

STJ(Student, Teacher, subJect)

$T \rightarrow J \quad S, J \rightarrow T$

1) R1(T,J) R2(S,J)

(BCNF? Y - lossless? N - dep. pres.? N)

2) R1(T,J) R2(S,T)

(BCNF? Y - lossless? Y - dep. pres.? N)

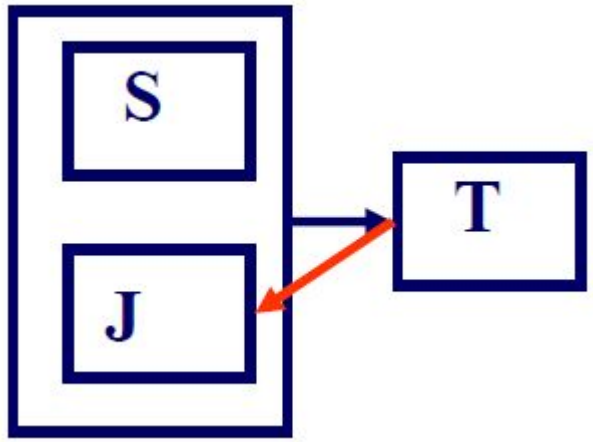
- in this case: impossible to have both
  - BCNF and
  - dependency preservation

3NF vs BCNF

STJ(Student, Teacher, subJect)

T → J S,J → T

- informally, 3NF “forgives” the red arrow



Review to Normalization

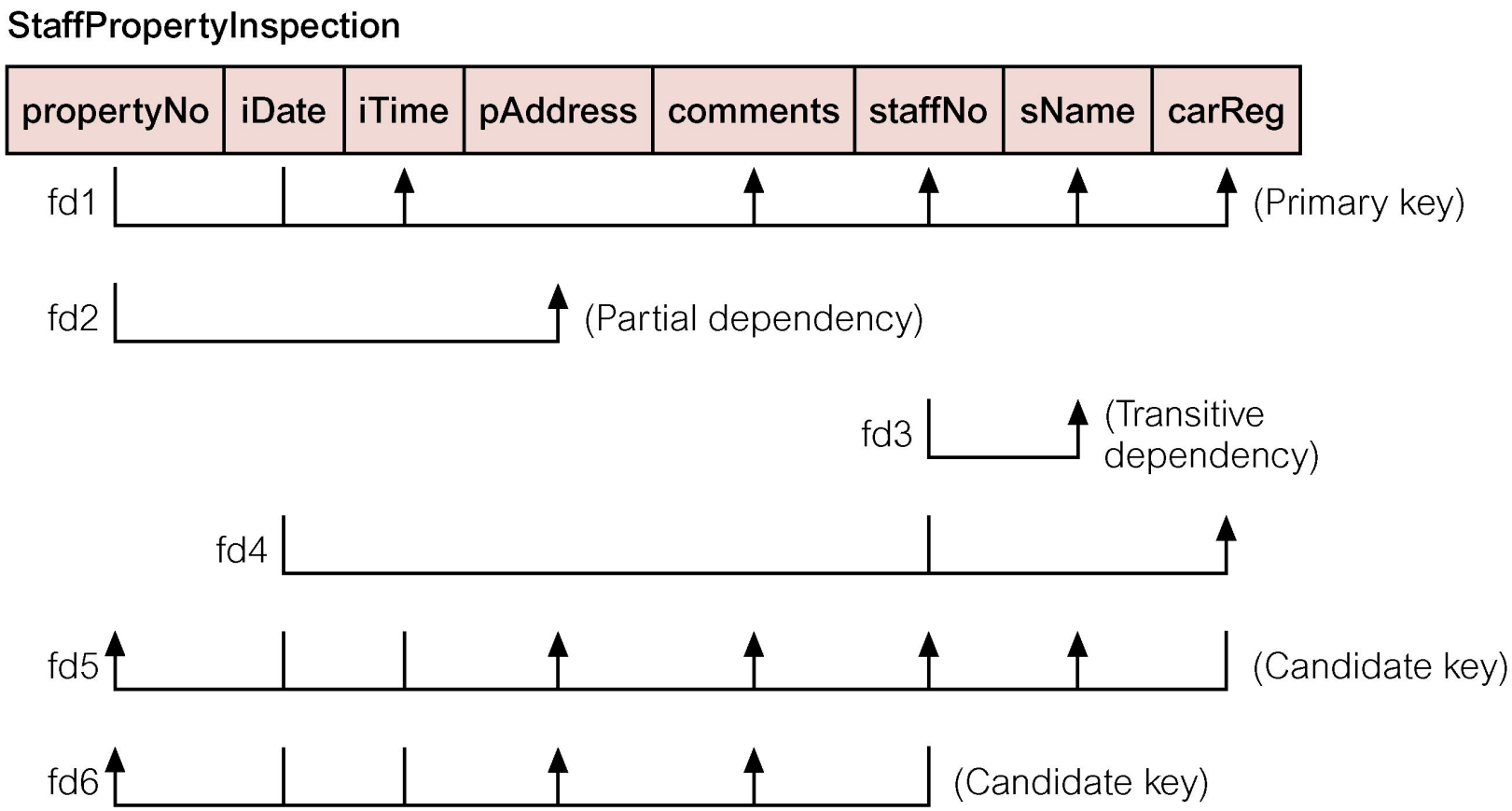
StaffPropertyInspection

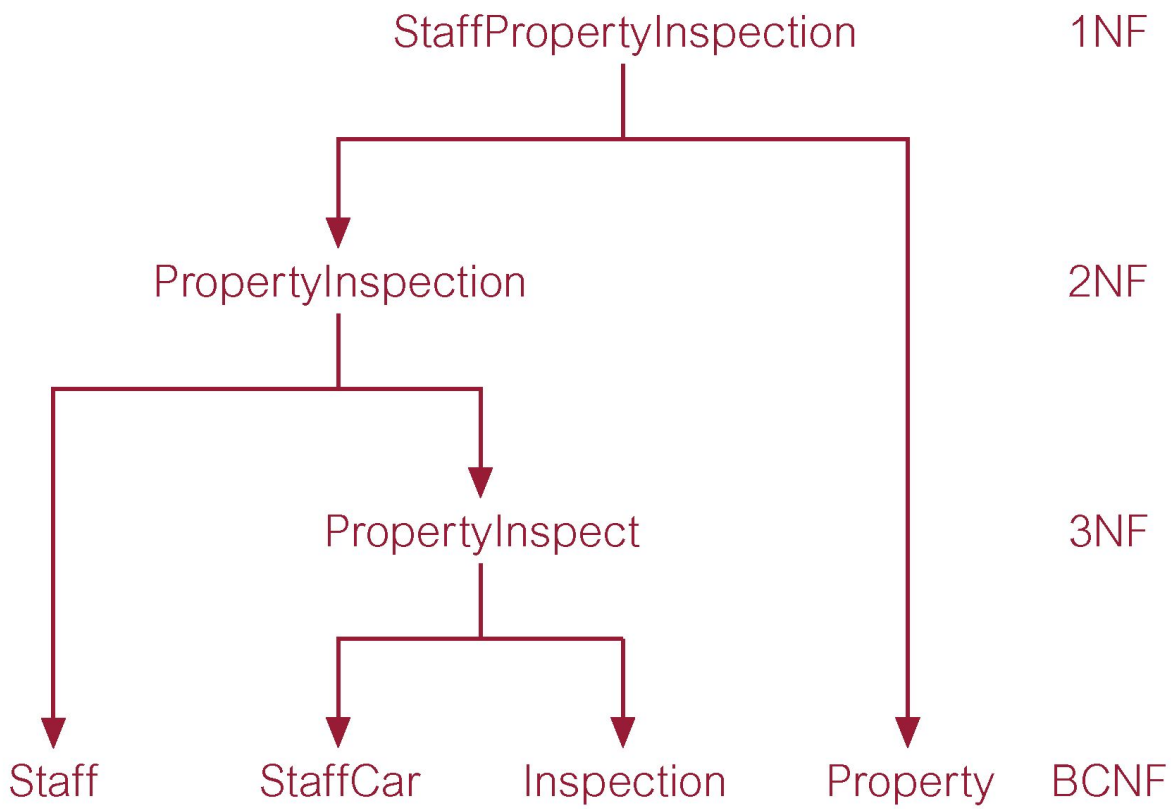
propertyNo	pAddress	iDate	iTime	comments	staffNo	sName	carReg
PG4	6 Lawrence St, Glasgow	18-Oct-00 22-Apr-01 1-Oct-01	10.00 09.00 12.00	Need to replace crockery In good order Damp rot in bathroom	SG37 SG14 SG14	Ann Beech David Ford David Ford	M231 JGR M533 HDR N721 HFR
PG16	5 Novar Dr, Glasgow	22-Apr-01 24-Oct-01	13.00 14.00	Replace living room carpet Good condition	SG14 SG37	David Ford Ann Beech	M533 HDR N721 HFR

StaffPropertyInspection

propertyNo	iDate	iTime	pAddress	comments	staffNo	sName	carReg
PG4	18-Oct-00	10.00	6 Lawrence St, Glasgow	Need to replace crockery	SG37	Ann Beech	M231 JGR
PG4	22-Apr-01	09.00	6 Lawrence St, Glasgow	In good order	SG14	David Ford	M533 HDR
PG4	1-Oct-01	12.00	6 Lawrence St, Glasgow	Damp rot in bathroom	SG14	David Ford	N721 HFR
PG16	22-Apr-01	13.00	5 Novar Dr, Glasgow	Replace living room carpet	SG14	David Ford	M533 HDR
PG16	24-Oct-01	14.00	5 Novar Dr, Glasgow	Good condition	SG37	Ann Beech	N721 HFR

Dependencies




StaffPropertyInspection
1NF
PropertyInspection
2NF
PropertyInspect
3NF
Staff
StaffCar
Inspection
Property
BCNF

Multivalued Dependency (MVD)

- Given a relation R (A, B, C, ..), such that
  - for each value of A there is a set of values for B and a set of values for C.
  - set of values for B and C are independent of each other.
- $A \twoheadrightarrow B$ and $A \twoheadrightarrow C$
- MVD are a consequence of 1NF.
- Informally – when 2 independent 1:N relationships are mixed in the same relation, an MVD may arise

MVD – trivial and non-trivial

- MVD $A \twoheadrightarrow B$ is trivial
  - $B \subseteq A$, or … (1)
  - $A \cup B = R$ … (2)
- Does not specify any significant or meaningful constraint on R

- MVD is nontrivial, if (1) and (2) are not satisfied
- Relations with nontrivial MVDs are all-key relations.

BranchStaffOwner

branchNo	sName	oName
B003	Ann Beech	Carol Farrel
B003	David Ford	Carol Farrel
B003	Ann Beech	Tina Murphy
B003	David Ford	Tina Murphy

Fourth Normal Form (4NF)

- Relation R is in 4NF, if
  - For all nontrivial MVDs $X \twoheadrightarrow Y$ in F^+, X is a superkey in R
- An all key relation is
  - in BCNF – no FDs
  - With MVD, is not in 4NF

BranchStaff

branchNo	sName
B003	Ann Beech
B003	David Ford

BranchOwner

branchNo	oName
B003	Carol Farrel
B003	Tina Murphy

4NF - Example

Questions –

1. Candidate Keys?
2. All MVDs?
3. Are pizza varieties offered affects the delivery area? Discuss for both ‘yes’ and ‘no’.
4. Is the relation in 4NF?
5. If not, transform.

Restaurant	Pizza Variety	DeliveryArea
A1	Thick Crust	S1
A1	Thick Crust	S2
A1	Thick Crust	S3
A1	Stuffed Crust	S1
Elite	Thin Crust	S3
Elite	Stuffed Crust	S3
Elite	Thin Crust	S3

Take Away – Normalize till BCNF

- Relation – Lots(Property_id#, country_name, lot#, area, price, tax_rate)

- FD1: P → CLAPrT
- FD2: CL → PAPrT
- FD3: C → T
- FD4: A → Pr
- FD5: A → C

Fifth Normal Form

- Also called Project-Join Normal Form
- a relation is in 5NF,
  - in 4NF and
  - Have no lossless decomposition into smaller tables, i.e. cannot be decomposed further.
- Table has Join Dependency –
  - Table can be recreated by joining multiple tables, and
  - each of this table have a subset of the attributes of the table
- **Trivial JD** – if one of the tables has all the attributes of T
- Relationships in JD are independent of each other

Example

Agent	Company	Product
Smith	Ford	Car
Smith	Ford	Truck
Smith	GM	Car
Smith	GM	Truck
Jones	Ford	Car
Jones	Ford	Truck
Brown	Ford	Car
Brown	GM	Car
Brown	Toyota	Car
Brown	Toyota	bus

- Can you quickly deduce business rules from this table?

Example

Agent	Company
Smith	Ford
Smith	GM
Jones	Ford
Brown	Ford
Brown	GM
Brown	Toyota

Company	Product
Ford	Car
Ford	Truck
GM	Car
GM	Truck
Toyota	Car
Toyota	bus

Agent	Product
Smith	Car
Smith	Truck
Jones	Car
Jones	Truck
Brown	Car
Brown	bus

- Jones sells cars and GM makes cars, but Jones does not represent GM
- Brown represents Ford and Ford makes trucks, but Brown does not sell trucks
- Brown represents Ford and Brown sells buses, but Ford does not make buses

Conclusion

“Everything should depend on the key, the whole key, and nothing but the key”