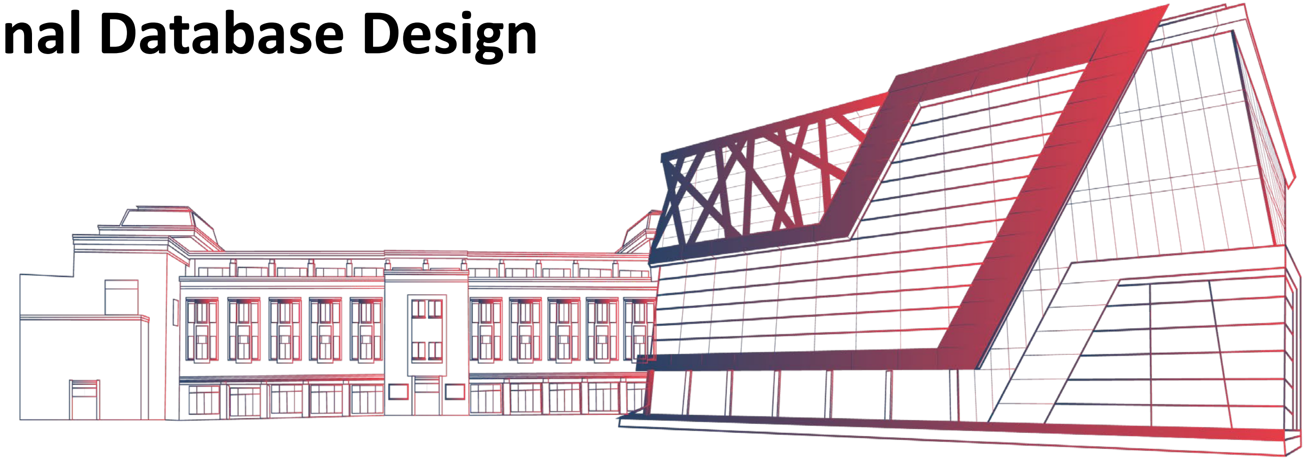
THE NORTHCAP UNIVERSITY

UNIT III

Relational Database Design



30 YEARS OF EXCELLENCE | NAAC Accredited

Lecture 19

Decomposition to Higher Normal Forms

Decomposition to Higher Normal Form Example

Consider a relation with schema R(A,B,C,D) and FDs {AB → C, C → D, D → A}.

a. Indicate all BCNF violations for R.

b. Decompose the relations into collections of relations that are in BCNF.

Solution: Attribute closure:

A → A	B → B	C → ACD	D → AD
AB → ABCD	AC → ACD	AD → AD	BC → ABCD
BD → ABCD	CD → ACD	ABC → ABCD	
ABD → ABCD	ACD → ACD	BCD → ABCD	

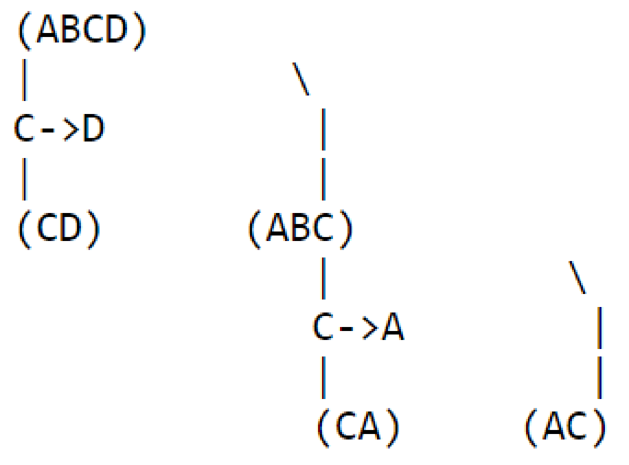
Thus, the candidate keys are: AB, BC, and BD

Decomposition to Higher Normal Form Example

a) C → D and D →A are violating BCNF

b) Decompose the relations into collections of relations that are in BCNF.

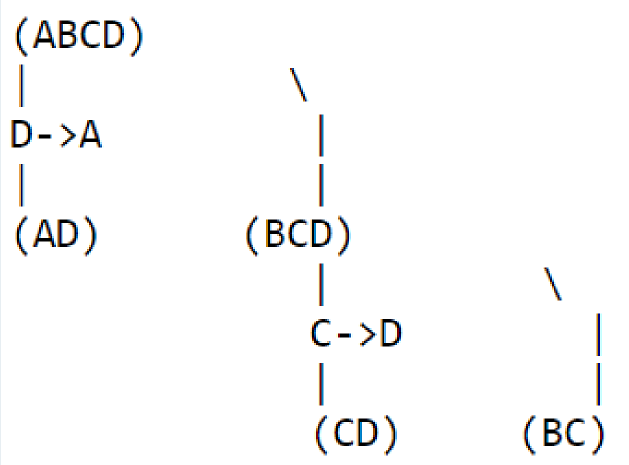
Case 1



```
(ABCD)
|            \
C->D          |
|             |
(CD)        (ABC)
              |          \
              C->A        |
              |           |
              (CA)      (AC)
```

R1(CD), R2(AC) and R3(BC) but
D → A and AB → C are not preserved

Case 2



```
(ABCD)
|            \
D->A          |
|             |
(AD)        (BCD)
              |          \
              C->D        |
              |           |
              (CD)      (BC)
```

R1(AD), R2(CD) and R3(BC) but
AB → C is not preserved

Decomposition to Higher Normal Form Drill

Decompose to highest normal form while satisfying lossless join and dependency preservation properties:

1. Relation R (ABCDE)

   $F = \{ AB \rightarrow C, C \rightarrow D, B \rightarrow E \}$

2. Relation R (ABCDEFGHIJ)

   $F =\{ AB \rightarrow C, C \rightarrow D, A \rightarrow E, E \rightarrow FG, B \rightarrow H, H \rightarrow IJ \}$

3. Relation R (ABC)

   $F = \{ AB \rightarrow C, C \rightarrow A \}$

Solution - Decomposition to Higher Normal Form Drill

1. **Relation R (ABCDE)**

   F = { AB → C, C → D, B → E }

   AB+ = A, B, C, D, E

   A+ = A

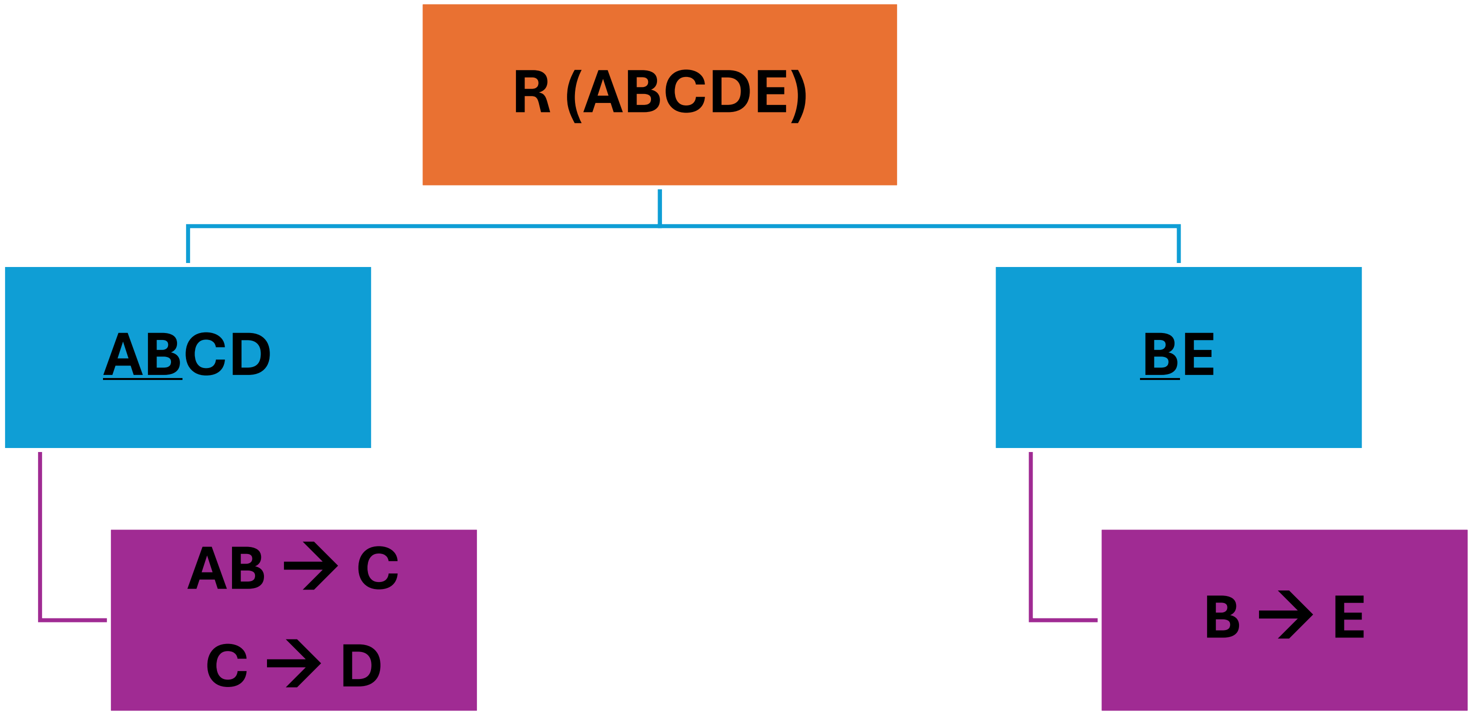
   B+ = B, E

FD	AB → C	C → D	B → E
BCNF	✓	X	X
3NF	✓	X	X
2NF	✓	✓	X
1NF	✓	✓	✓

The table is currently in 1NF but not in 2NF because of partial dependency B → E

Solution - Decomposition to Higher Normal Form Drill (Cont.)

- **2NF Decomposition**



R (ABCDE)

ABCD

AB → C

C → D

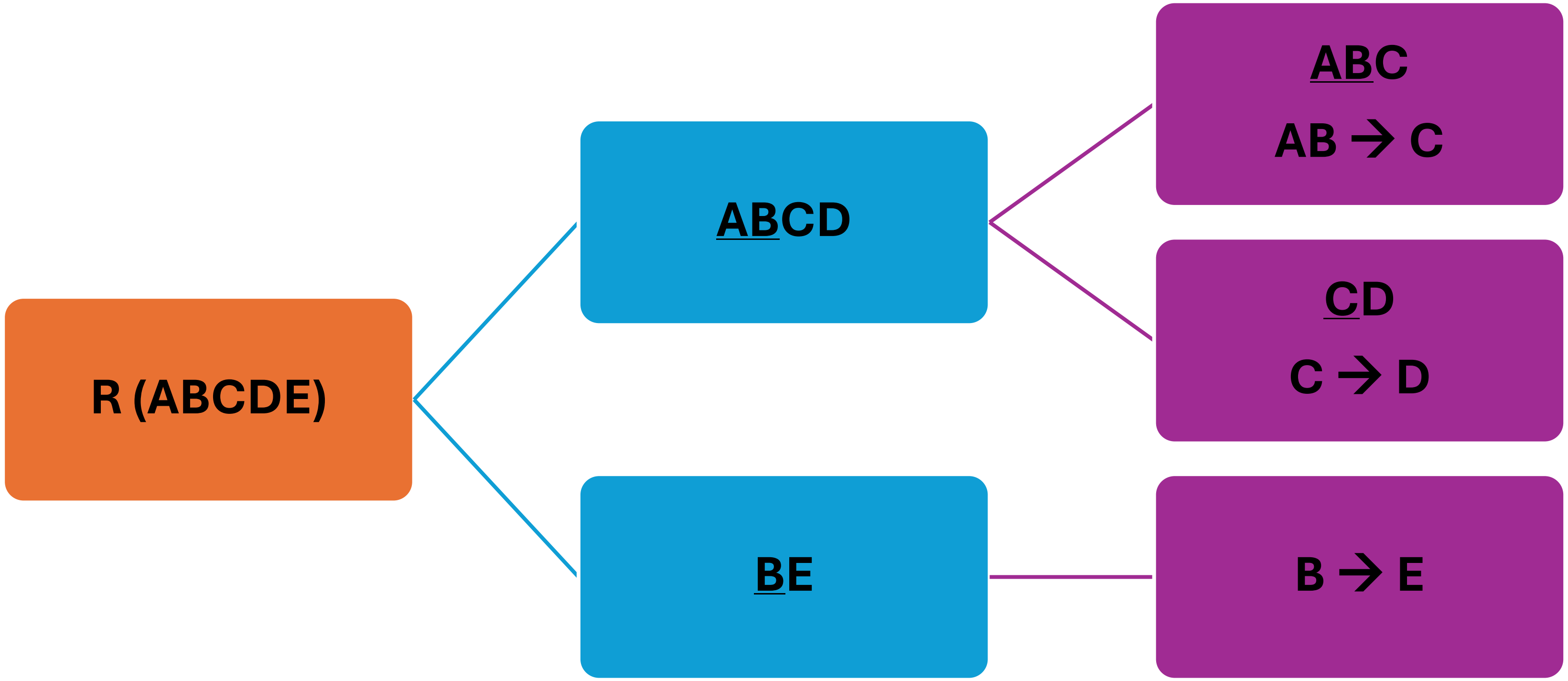
BE

B → E

2NF, Lossless Join and Dependency Preservation satisfied

Solution - Decomposition to Higher Normal Form Drill (Cont.)

- **3NF Decomposition**



3NF, BCNF, Lossless Join and Dependency Preservation satisfied

Solution - Decomposition to Higher Normal Form Drill (Cont.)

2. **Relation R (ABCDEFGHIJ)**

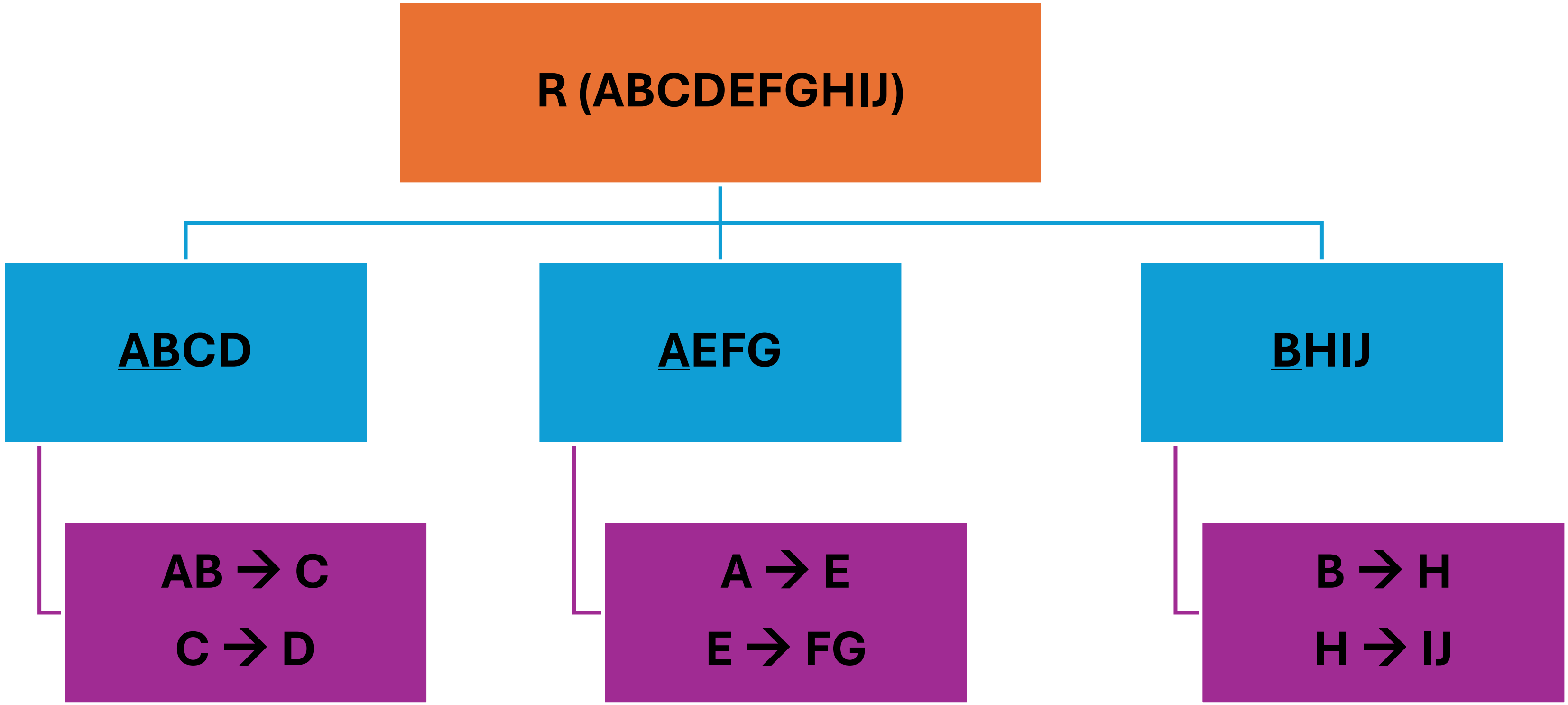
F ={ AB → C, C → D, A → E, E → FG, B → H, H → IJ }

AB+ = A, B, C, D, E, F, G, H, I, J

FD	AB → C	C → D	A → E	E → FG	B → H	H → IJ
BCNF	✓	X	X	X	X	X
3NF	✓	X	X	X	X	X
2NF	✓	✓	X		X	

Solution - Decomposition to Higher Normal Form Drill (Cont.)

- **2NF Decomposition**



R (ABCDEFGHIJ)

ABCD

AB → C

C → D

AEFG

A → E

E → FG

BHIJ

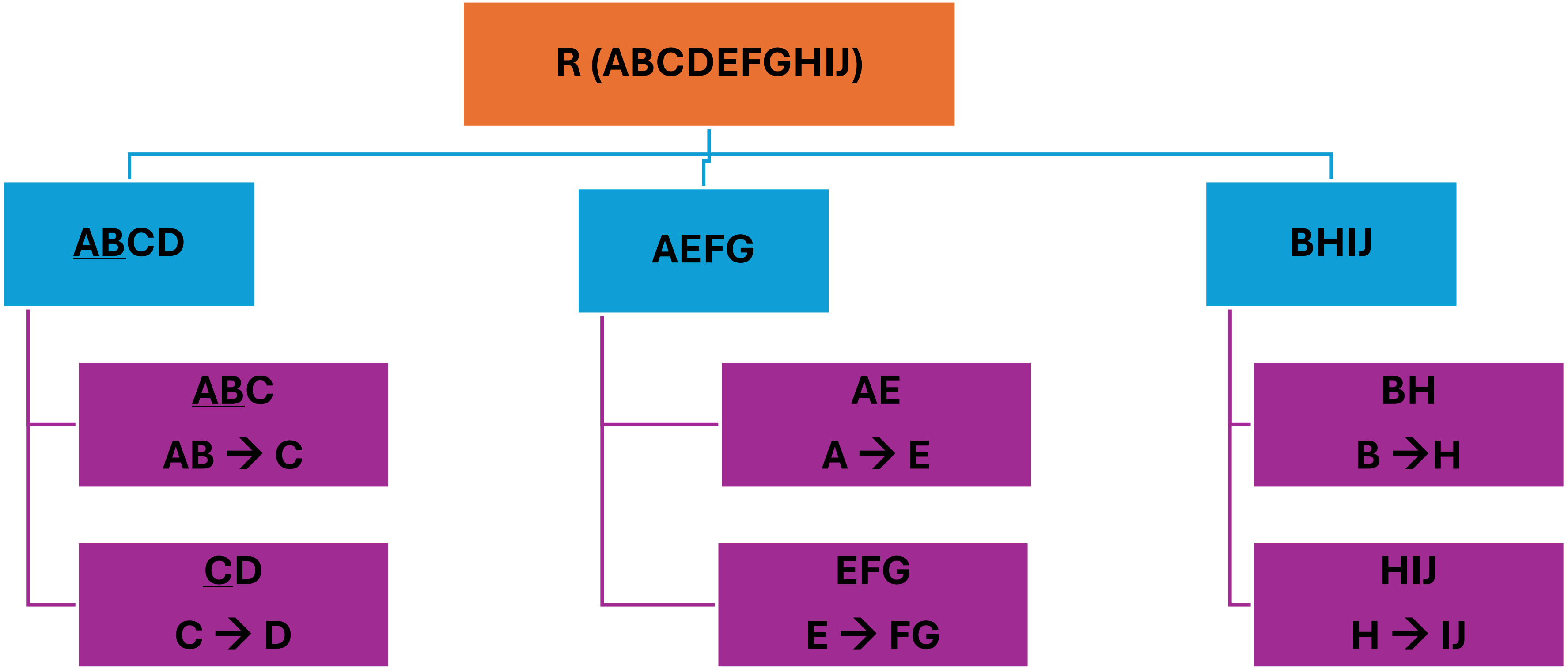
B → H

H → IJ

2NF, Lossless Join and Dependency Preservation satisfied

Solution - Decomposition to Higher Normal Form Drill (Cont.)

- **3NF Decomposition**



R (ABCDEFGHIJ)

ABCD

ABC
AB → C

CD
C → D

AEFG

AE
A → E

EFG
E → FG

BHIJ

BH
B →H

HIJ
H → IJ

3NF, BCNF, Lossless Join and Dependency Preservation satisfied

Solution - Decomposition to Higher Normal Form Drill (Cont.)

3. **Relation R (ABC)**

F = { AB → C, C → A }

AB+ = A, B, C

CB+ = B, C, A

The relation is in 3NF but not BCNF

Case1: If R1 (BC) and R2 (AC)

Lossless Join Decomposition and BCNF are satisfied but dependency preservation is violated.

Case 2: If we decompose such that dependency preservation and lossless join is satisfied, then BCNF is not satisfied.

BCNF and DP decomposition may not be possible together for all relations

Normalization Summary

DB Design Goal	1NF	2NF	3NF	BCNF
Achieve 0% Redundancy	X	X	X	✓ (over FDs)
LLJ Satisfied	✓	✓	✓	✓
DP Satisfied	✓	✓	✓	X✓

Best accurate form is 3NF because dependency preservation and lossless join properties are more important

Multi-valued Dependency (MVD)

A **multi-valued dependency** (**MVD**) ***X* →→ *Y*** specified on relation schema *R*, where *X* and *Y* are both subsets of *R*, specifies the following constraint:

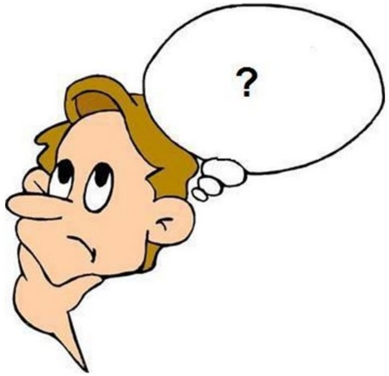
If two tuples t_1 and t_2 exist in *R* such that $t_1[X] = t_2[X]$, then two tuples t_3 and t_4 should also exist in *R* with the following properties, where we use *Z* to denote $(R - (X \cup Y)$:

$$t_4[X] = t_3[X] = t_1[X] = t_2[X]$$

$$t_3[Y] = t_1[Y] \text{ and } t_4[Y] = t_2[Y]$$

$$t_3[Z] = t_2[Z] \text{ and } t_4[Z] = t_1[Z]$$

MVD Example



Student

StudentName	CourseDiscipline	Activities
Amit	Mathematics	Singing
Akash	Mathematics	Dancing
Yuvraj	Computers	Cricket
Akash	Mathematics	Singing
Akash	Literature	Dancing
Akash	Literature	Singing

Therefore, multivalued dependency:

StudentName →→ CourseDiscipline

StudentName →→ Activities

MVD Rules

- **<u>Complementation:</u>** If X →→ Y, and Z is all the other attributes i.e. Z = (*R* - (*X* ∪ *Y*), then X →→ Z.

  Eg. R(ABCD) if A →→ B then, A →→ CD

- **<u>Trivial MVD:</u>** An MVD ***X* →→ *Y*** in *R* is called a **trivial MVD** if

  (a) Y is a subset of X, or (b) X ∪ Y = R

  Eg. R (ABCD) { AB →→ A, AB →→ CD } Trivial MVD

  { AB →→ C } Non-trivial MVD

- **<u>Split/ Merge:</u>** Non-trivial MVDs are not allowed to split or merge unlike FDs

  Eg. [X →→ YZ] ≠ [X →→ Y, X→→ Z]

  [X → YZ] = [X → Y, X → Z]

Fourth Normal Form

A relational schema R is in 4NF iff

a) X is superkey in every non-trivial FD X → Y in R (i.e. BCNF)

and

b) X is superkey in every non-trivial MVDs X →→ Y

Non-trivial FD
X → Y
X: superkey

and

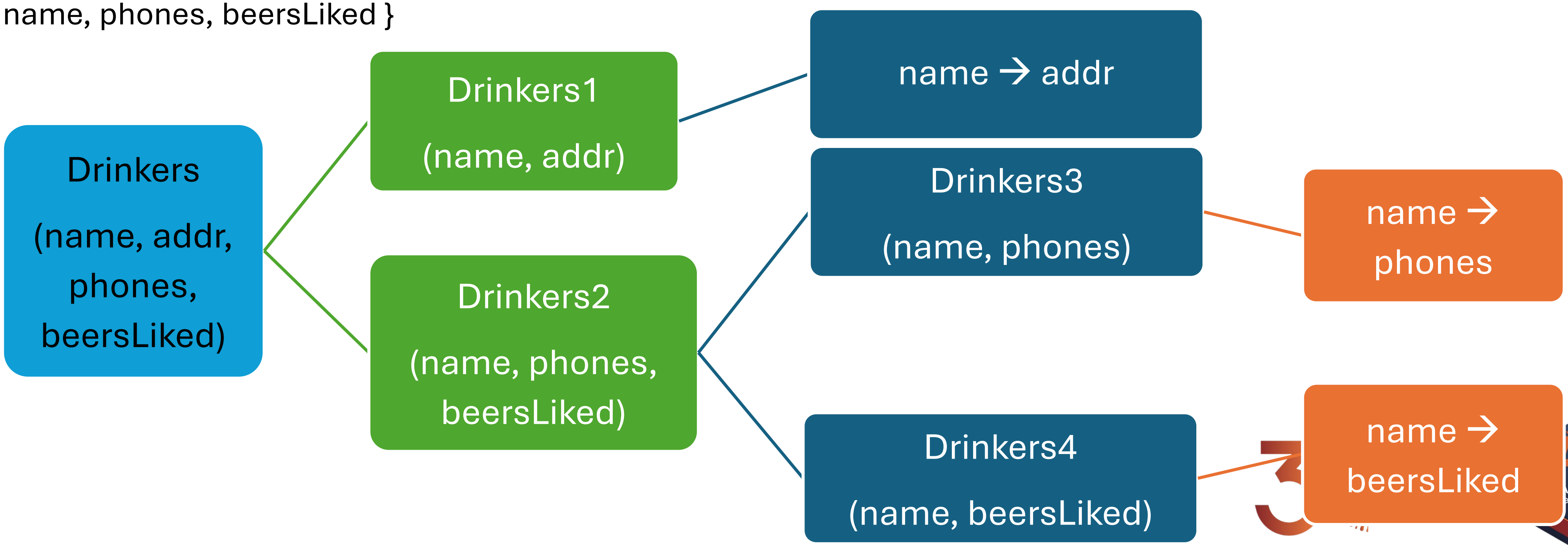
Non-trivial MVD
X →→ Y
X: superkey

Fourth Normal Form Example

- **Relation Drinkers(name, addr, phones, beersLiked)**
- FD: name → addr
- MVD's: name →→ phones

  name →→ beersLiked

- Key is { name, phones, beersLiked }



Drinkers (name, addr, phones, beersLiked)
Drinkers1 (name, addr)
Drinkers2 (name, phones, beersLiked)
name → addr
Drinkers3 (name, phones)
Drinkers4 (name, beersLiked)
name → phones
name → beersLiked

Fifth (Project - join) Normal Form (PJNF)

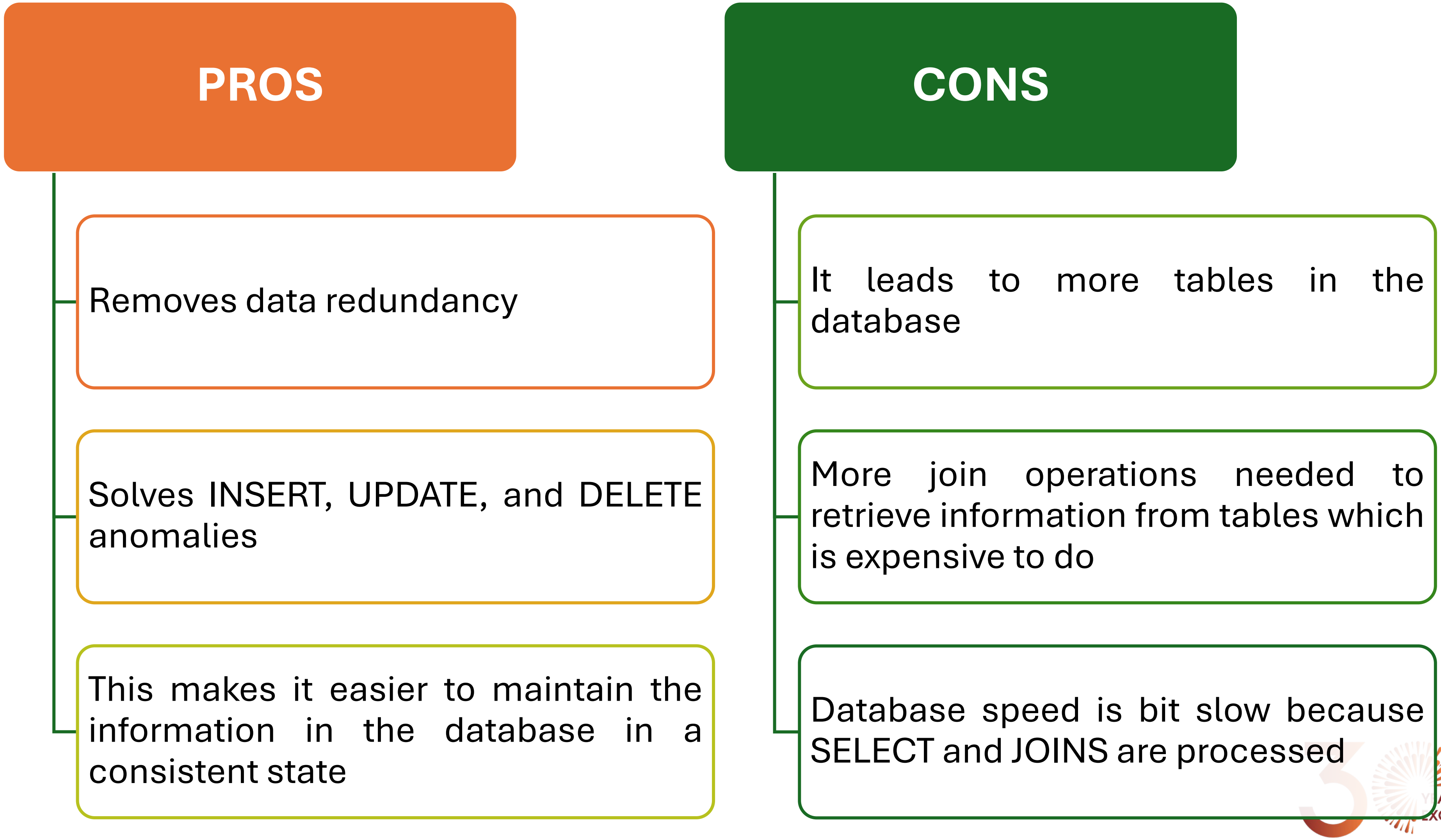
A relational schema R is in 5NF iff

a) It is in 4NF

and

b) Does not have any join dependency and joining should satisfy lossless decomposition i.e. the decomposed sub-relations can be joined in any order and all joins should be lossless.

Normalization Summarization



PROS

- Removes data redundancy
- Solves INSERT, UPDATE, and DELETE anomalies
- This makes it easier to maintain the information in the database in a consistent state

CONS

- It leads to more tables in the database
- More join operations needed to retrieve information from tables which is expensive to do
- Database speed is bit slow because SELECT and JOINS are processed

Thanks!!