ProfessorDep

ProfID	Name	Address	DepID	DepName	DHID
			1	CS	
		...			
			4	MIE	

$t[DepID] = t'[DepID] \Rightarrow t[Dname] = t'[DName]$

Functional Depency

Decomposition: For a table that is not "good"

- Decompose it to multiple "good" tables
- Make sure the Decomposition is lossless

$R: R_1, R_2$

$$\boxed{R_1 \infty R_2 = R}$$

join

Professor

Name	ID	Address	DepID

Department

ID	Name	Dep Head ID

Professor ∞ Department = ProfessorDep

P_1

ID	Name	Address

P_2

ID	DepID

Functional Dependency $A \to B$ Exists if the values of the Attributes A Uniquely identify the values on B

Can be a set of attributes

e.g., P_1:

ID → Name
ID → Address
ID → {Name, Address} ← Example 1
{ID, Name} → ID   {ID, Name} → Name
{ID, Name} → Address
{ID, Address} → Name ...

Goal: Is it possible to find a small set of functional Dependencies that enforcing them would enforce *all* of the FDs.

The Canonical Set

FD Rules

- Rule 1: (Reflexivity):

  if $X \subseteq Y$ then $Y \to X$

- Rule 2: (Augmentation):

  if $X \to Y$, then $ZX \to ZY$

- Rule 3: (Transitivity)

  if $X \to Y$ and $Y \to Z$, then $X \to Z$

Primitive Rules

- Rule 4: (Decomposition)

  if $X \to YZ$, then $X \to Y$ and $X \to Z$

- Rule 5: (Union)

  if $X \to Y$ and $X \to Z$, then $X \to YZ$

- Rule 6: (Pseudo Transitivity)

  if $X \to Y$ and $WY \to Z$ then $WX \to Z$

Derived Rules

Closure:

Let F be a set of FDs. the Closure of F, shown as

$$F^+$$

is the set of FDs that can derive from F, by recursively applying Rules 1-3.

e.g., Example 1.

Let $F = \{ID \xrightarrow{F1} name,\ ID \xrightarrow{F_2} Address\}$

what is F^+

Rule 2:

F1 $\rightsquigarrow$ F3 : {ID, Address} → {Name, Address}

F2 $\rightsquigarrow$ F4 : {ID, Name} → {Name, address}

Rule 4:

$F_4 \rightsquigarrow F_5$

{ID, name} → ID

F4 $\rightsquigarrow$ F6

{ID, Name} → name

$F_3 \rightsquigarrow F_7$

{ID, Address} → ID

F3 $\rightsquigarrow$ F8

{ID, Address} → Address

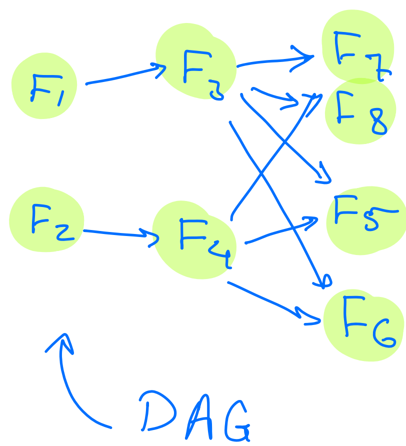
...

F is a Canonical Set for F^+,

if $\nexists\ F' \subseteq F$, s.t.

$$(F')^+ = F^+$$

* The DBMS only needs to check for the Canonical set of functional Dependencies



the set of nodes with Indegree = ∅ is the Canonical set

Normal Forms:

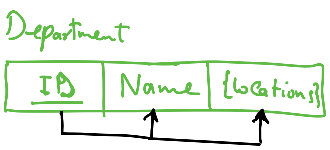
DB design Standards that describe the "goodness" of a data base

- 1NF
- 2NF
- 3NF
- BCNF
- 4NF
- 5NF

"Goodness" increases

1NF: A database Design is in 1NF if No table has any Composite or Multivalued Attributes

e.g.,

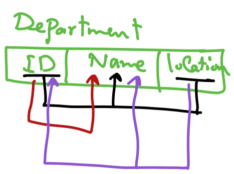


X 1NF

Resolution

Department ✓ 1NF

ID	Name	Location
1	A	X
2	B	Y
1	~~A~~ C	Z

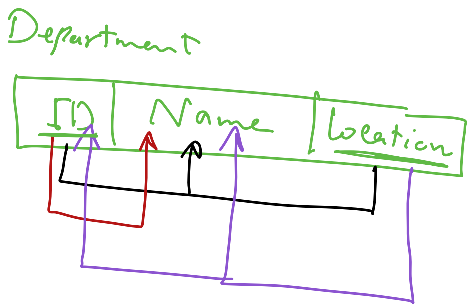


Candidate Keys = {{ID, location}}

Full Functional Dependencies (FFD)

we Say $X \to Y$ is a FFD if $\forall X' \subset X,\ X' \not\to Y$

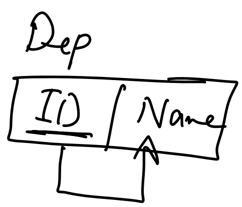
2NF: A DB design is in 2NF if all FDs are FFD

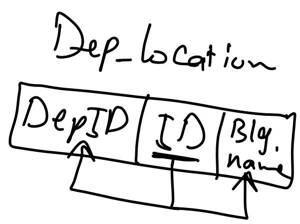


X : Not in 2NF.

{ID, location} → name is Not a FFD

Resolution





✓ 2NF

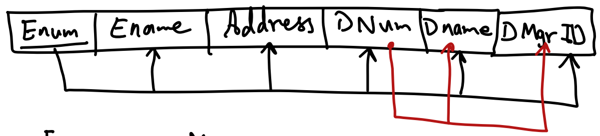
Dep

ID	Name
1	A
2	B

Dep_location

DepID	ID	Blg.
1	1	X
2	2	Y
1	3	Z

Employee



Enum → X

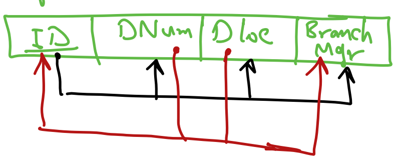
DNum → {Dname, DMgr ID} ← FFD

√2NF

3NF

An attribute A is a Prime attribute if it is part of a Candidate Key.

Dep_loc



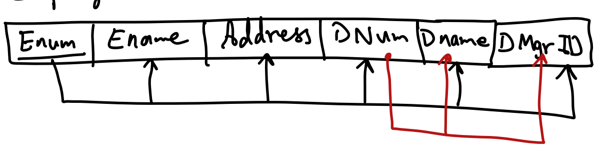
Prime attributes = {ID, DNum, Dloc}

3NF: A database Design is in 3NF if there is no Transitive FD for Non_Prime attributes

Transitivity: X → Y → Z

- Should not be Non_Prime
- Z is Part of a Candidate Key

Employee

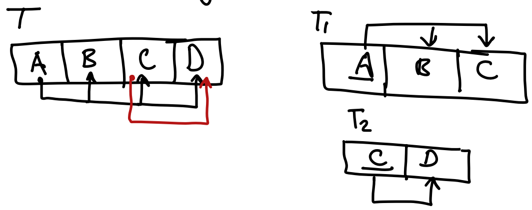


Enum → DNum → DName X 3NF

not Prime

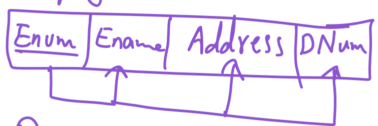
Decomposition to 3NF:

Break the Table to Multiple Tables according to FDs

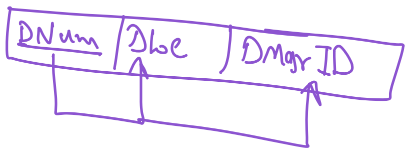


e.g.,

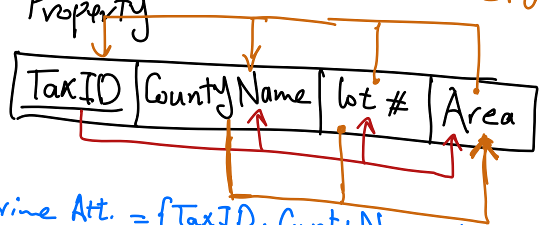
Employee



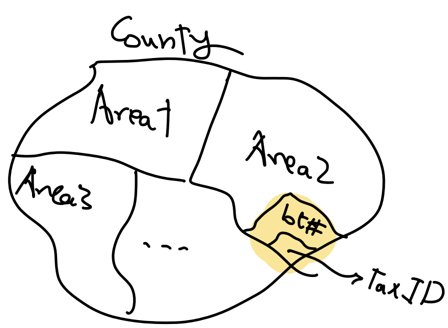
Department



Property 3rd ✓



Prime Att. = {TaxID, County Name, lot #}



Assumption: Lot #s are Unique within each County

BCNF:

Boyce Codd

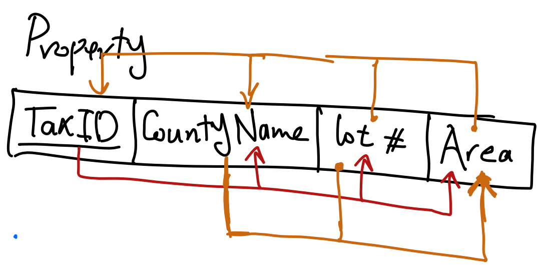
For any FD $X \to Y$

either

$Y \subseteq X$

or

X is a super key



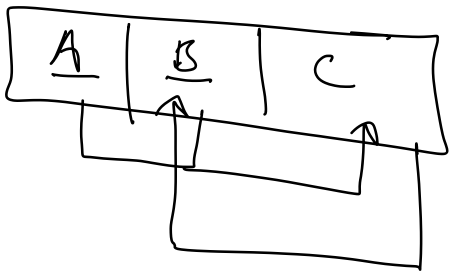
$\{CountyName, lot\ \#\} \to Area$

✗ BCNF

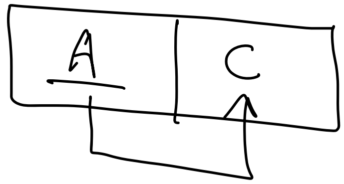
Bcz {CountyName, lot #} is not a key

BCNF Decomposition

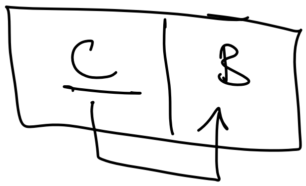
T 3NF



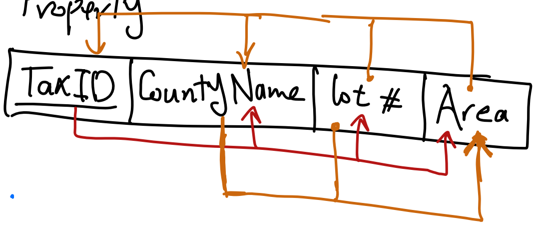
T_1



T_2



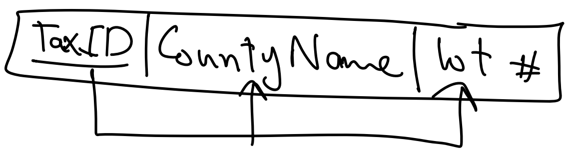
Property



TaxID → {CountyName, lot #} → Area

C → B

Property



Area

√BCNF