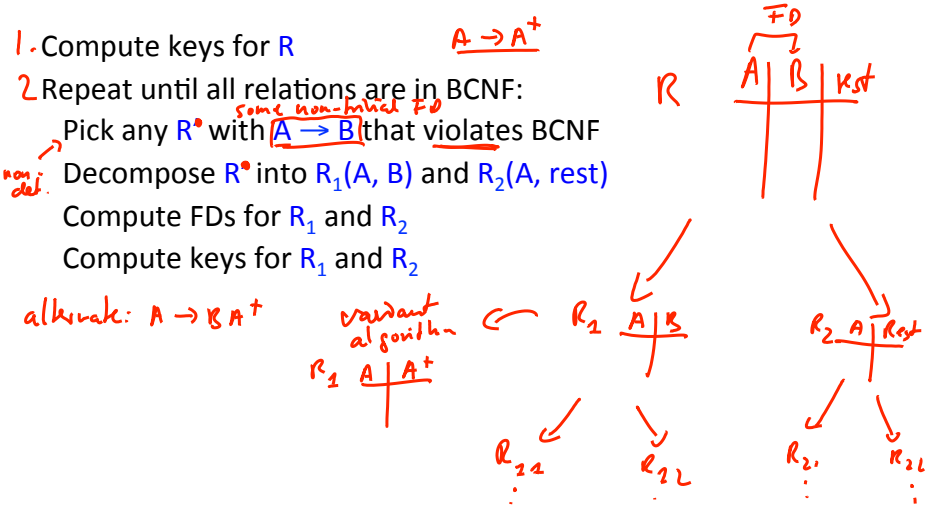
BCNF

BCNF decomposition algorithm

Input: relation R + FDs for R

Output: decomposition of R into BCNF relations with "lossless join"

1. Compute keys for R
2. Repeat until all relations are in BCNF:
   - Pick any R^* with $A \rightarrow B$ that violates BCNF
   - Decompose R^* into $R_1(A, B)$ and $R_2(A, rest)$
   - Compute FDs for R_1 and R_2
   - Compute keys for R_1 and R_2



BCNF

BCNF Decomposition Example

Student(SSN, sName, address, HScode, HSname, HScity, GPA, priority)

1. SSN → sName, address, GPA
2. HScode → HSname, HScity
3. GPA → priority

S1 (HScode, HSname, HScity) ✓

S2 (SSN, sName, addr, HScode, GPA, prio)

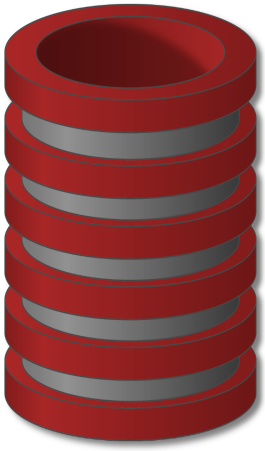
S3 (GPA, Prio) ✓

S4 (SSN, sName, addr, HScode, GPA)

S5 (SSN, sName, addr, GPA) ✓

S6 (SSN, HScode) ✓

BCNF: S1, S3, S5, S6

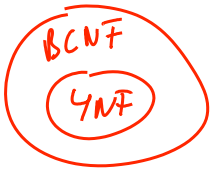


Relational Design Theory

Multivalued Dependencies & 4th Normal Form

Relational design by decomposition

- "Mega" relations + properties of the data
- System decomposes based on properties
- Final set of relations satisfies normal form
  - No anomalies, no lost information
- Functional dependencies ⇒ Boyce-Codd Normal Form
- Multivalued dependences ⇒ Fourth Normal Form



Example: College application info.

`Apply(SSN, cName, hobby)`

FDs? None

Keys? {SSN, cName, hobby} only key

BCNF? yes

Good design? No

if student with SSN x applies to N colleges
has M hobbies
↘ N×M rows to describe that
(but N+M should be enough)

Multivalued Dependency

- Based on knowledge of real world
- All instances of relation must adhere

Relation R, $\bar{A} \twoheadrightarrow \bar{B}$ ($\bar{A} = A_1 \ldots A_n$, $\bar{B} = B_1 \ldots B_n$)

$\forall$ a. $t, u \in R$ if $t[\bar{A}] = u[\bar{A}]$

then: exist $v \in R$: $v[\bar{A}] = t[\bar{A}]$
$v[\bar{B}] = t[\bar{B}]$
$v[\text{rest}] = u[\text{rest}]$

R	$\bar{A}$	$\bar{B}$	rest
t:	$\bar{a}$	$\bar{b}_1$	$\bar{r}_1$
u:	$\bar{a}$	$\bar{b}_2$	$\bar{r}_2$
v:	$\bar{a}$	$\bar{b}_1$	$\bar{r}_2$
v':	$\bar{a}$	$\bar{b}_2$	$\bar{r}_1$

MVDs aka TGDs

Apply(SSN, cName, hobby)

SSN →→ cName // SSN →→ hobby

Apply

SSN	cName	hobby
007	UCD	trumpet
007	UCSD	surfing
007	UCD	surfing
007	UCSD	trumpet

Modified example

Apply(SSN, cName, hobby)

Reveal hobbies to colleges selectively

FDs None

MVDs? None (b/c cName and hobby no longer independent)

Good design? yes, for this case

Apply

007	UCD	trumpet
007	UCSD	surfing
007	UCSD	trumpet

Expanded example

`Apply(SSN, cName, date, major, hobby)`

(1) *Reveal hobbies to colleges selectively*
(2) *Apply once to each college* one day
(3) *May apply to multiple majors*

FD: SSN, cName → date (2)

(4) FD SSN, major ?→ cName "each applicant applies to at most one college for a given major"

(5) FD SSN → major "single major"

MVD SSN, cName, date →→ major
- " - →→ hobby ↩ independent

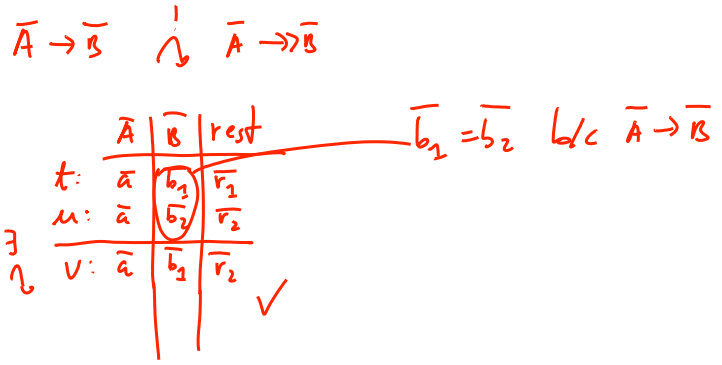
Trivial Multivalued Dependency

$\bar{A} \twoheadrightarrow \bar{B}$ $\bar{B} \subseteq \bar{A}$ or $\bar{A} \cup \bar{B}$ = all attributes

Nontrivial MVD

Rules for Multivalued Dependencies

FD-is-an-MVD rule



Rules for Multivalued Dependencies

Intersection rule

$\bar{A} \twoheadrightarrow \bar{B}$ $\bar{A} \twoheadrightarrow \bar{C}$ ⇒ $\bar{A} \twoheadrightarrow \bar{B} \cap \bar{C}$

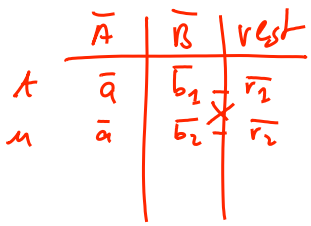
Transitive rule

$\bar{A} \twoheadrightarrow \bar{B}$, $\bar{B} \twoheadrightarrow \bar{C}$ ⇒ $\bar{A} \twoheadrightarrow \bar{C} \setminus \bar{B}$

MVDs & 4NF

Fourth Normal Form

Relation R with MVDs is in 4NF if:
For each nontrivial $A \twoheadrightarrow B$, A is a key



MVDs & 4NF

4NF decomposition algorithm

Input: relation R + FDs for R + MVDs for R
Output: decomposition of R into 4NF relations with "lossless join"

Compute keys for R
Repeat until all relations are in 4NF:
  Pick any R' with nontrivial $A \twoheadrightarrow B$ that violates 4NF
  Decompose R' into $R_1(A, B)$ and $R_2(A, \text{rest})$
  Compute FDs and MVDs for R_1 and R_2
  Compute keys for R_1 and R_2