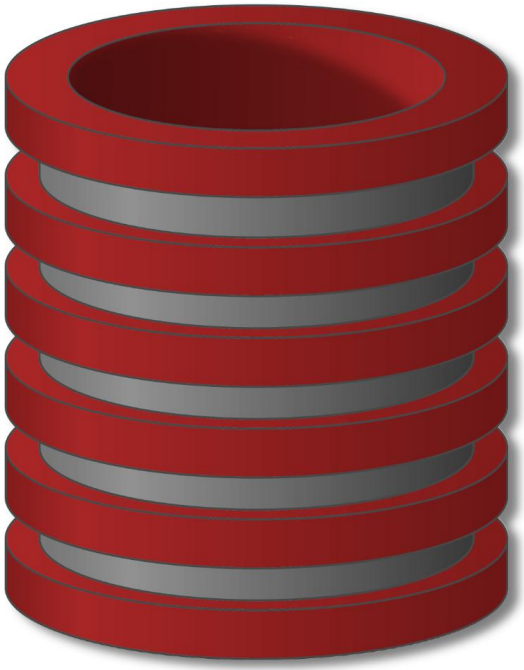


Relational Design Theory

Shortcomings of BCNF/4NF

Jennifer Widom

Boyce-Codd Normal Form

Relation R with FDs is in BCNF if:

For each $A \rightarrow B$, A is a key

Fourth Normal Form

Relation R with MVDs is in 4NF if:

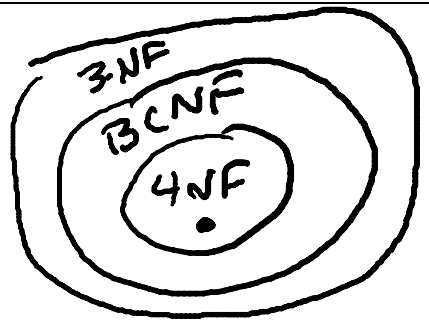
For each nontrivial $A \twoheadrightarrow B$, A is a key

Example: College application info.

Apply(SSN, cName, date, major)

Can apply to each college once for one major ✓

Colleges have non-overlapping application dates ✓



FDs: SSN, cName → date, major

date → cName

Keys: SSN, cName

BCNF: No.

A1 (date, cName)

A2 (SSN, date, major)

?

Good design? Not necessarily. 3rd Normal Form

Example #2

Student(SSN, HSname, GPA, priority)

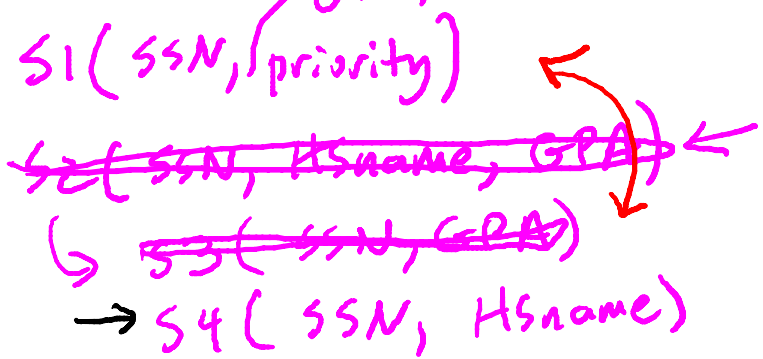
Multiple HS okay, priority determined from GPA

FDs: SSN → GPA GPA → priority SSN → priority

SSN → GPA, priority

Keys: SSN, HSname

BCNF: No. →



Good design?

Not necessarily.

Boyce-Codd Normal Form

Relation R with FDs is in BCNF if:

For each $A \rightarrow B$, A is a key

Fourth Normal Form

Relation R with MVDs is in 4NF if:

For each nontrivial $A \twoheadrightarrow B$, A is a key

After decomposition, no guarantee dependencies can be checked on decomposed relations

Example #3

Scores(SSN, sName, SAT, ACT) *

"Denormalized" relation

Multiple SATs and ACTs allowed

All queries return name + composite score for SSN

FDs + keys: SSN → sName. No Key.

MVDs: SSN, sName →→ SAT * "rest" (ACT)

4NF: No.

~~S1 (SSN, sName, SAT)~~

~~S2 (SSN, sName, ACT)~~

4NF

S3 (SSN, sName)

S4 (SSN, SAT)

S5 (SSN, ACT)

Example #4

College(cName, state)
CollegeSize(cName, enrollment)
CollegeScores(cName, avgSAT)
CollegeGrades(cName, avgGPA)
...

"Too decomposed"

BCNF/4NF? Yes.

Good Design? Not necessarily.

Designing a database schema

- Usually many designs possible
- Some are (much) better than others!
- How do we choose?

❖ Very nice theory for relational database design

- Normal forms – "good" relations
- Design by decomposition
- Usually intuitive and works well
- Some shortcomings
  - Dependency enforcement ✓
  - Query workload ✓
  - Over-decomposition ✓