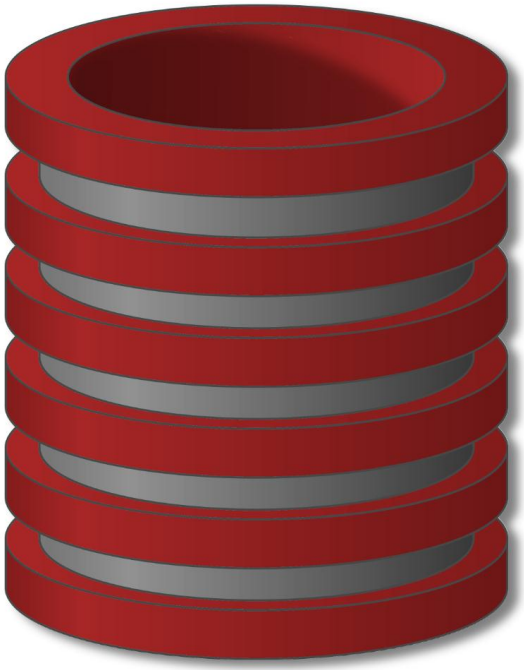


Relational Design Theory

Motivation & overview

Jennifer Widom

Designing a database schema

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

Often use higher-level design tools, but …

- Some designers go straight to relations
- Useful to understand why tools produce certain schemas

❖ Very nice theory for relational database design

Example: College application info.

- SSN and name
- Colleges applying to
- High schools attended (with city)
- Hobbies

```
Apply(SSN, sName, cName, HS, HScity, hobby)
```

Rel. design - overview

- **Apply(SSN, sName, cName, HS, HScity, hobby)**

123 Ann from PAHS (P.A.) and GHS (P.A.) plays tennis and trumpet and applied to Stanford, Berkeley, and MIT

123 Ann Stanford PAHS P.A. tennis
123 Ann Berkeley PAHS P.A. tennis
123 Ann Berkeley PAHS P.A. trumpet
⋮ ⋮ GHS ⋮

12 tuples

Apply(SSN, sName, cName, HS, HScity, hobby)

123 Ann from PAHS (P.A.) and GHS (P.A.) plays tennis and trumpet and applied to Stanford, Berkeley, and MIT

Design "anomalies"

- Redundancy

capture info. multiple times
123 Ann PAHS tennis
MIT

Apply(SSN, sName, cName, HS, HScity, hobby)

123 Ann from PAHS (P.A.) and GHS (P.A.) plays tennis and trumpet and applied to Stanford, Berkeley, and MIT

Design "anomalies"

- Redundancy
- Update anomaly

update facts differently

trumpet ~~trumpet~~ cornet

Apply(SSN, sName, cName, HS, HScity, hobby)

123 Ann from PAHS (P.A.) and GHS (P.A.) plays tennis and trumpet and applied to Stanford, Berkeley, and MIT

Design "anomalies"

- Redundancy
- Update anomaly
- Deletion anomaly

inadvertently deletion

surfing

Example: College application info.

- SSN and name
- Colleges applying to
- High schools attended (with city)
- Hobbies

No anomalies
Reconstruct orig. data

Student(SSN, sName)

Apply(SSN, cName, hobby)

HighSchool(SSN, HS, HScity)

~~Located(HS, HScity)~~

~~Hobbies(SSN, hobby)~~

Design by decomposition

- Start with "mega" relations containing everything
- Decompose into smaller, better relations with same info.
- Can do decomposition automatically

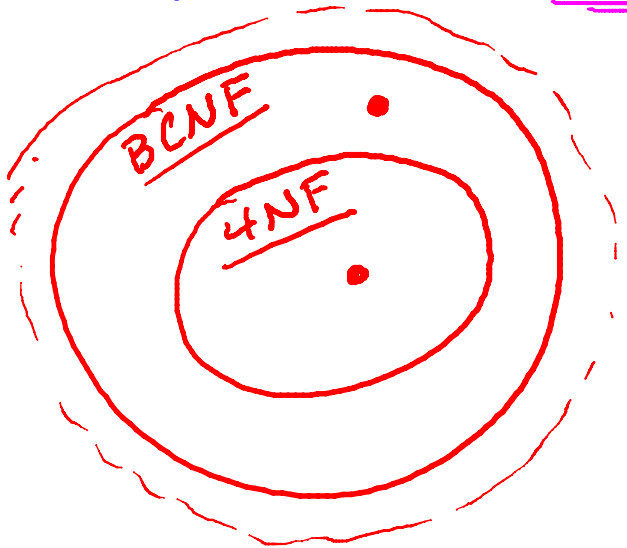
Automatic decomposition

- "Mega" relations + *properties of the data*
- System decomposes based on properties
- Final set of relations satisfies *normal form*
  - *No anomalies, no lost information*

Properties and Normal Forms

- *Functional dependencies ⇒ Boyce-Codd Normal Form*
- *+ Multivalued dependences ⇒ Fourth Normal Form*

1st
2nd
3rd



Functional Dependencies and BCNF

Apply(SSN, sName, cName) ← Not in BCNF

↑ Not a key

- Redundancy; Update & Deletion Anomalies
- Storing **SSN-sName** pair once for each college

Functional Dependency **SSN → sName**

- Same **SSN** always has same **sName**
- Should store each **SSN**'s **sName** only once

Boyce-Codd Normal Form If **A → B** then **A** is a key

Decompose: **Student(SSN, sName)** **Apply(SSN, cName)**

↑ key

Multivalued Dependencies and 4NF

Apply(SSN, cName, HS)

- Redundancy; Update & Deletion Anomalies
- Multiplicative effect
- Not addressed by BCNF: No functional dependencies

Multivalued Dependency **SSN ↠ cName**

- Given **SSN** has every combination of **cName** with **HS**
- Should store each **cName** and each **HS** for an **SSN** once

Fourth Normal Form If **A ↠ B** then **A** is a key

Decompose: **Apply(SSN, cName)** **HighSchool(SSN, HS)**

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 $\Rightarrow$ Boyce-Codd Normal Form
- Multivalued dependences $\Rightarrow$ Fourth Normal Form