Announcements

- Upcoming events: ~ HW#2 ~ box in lecture | SmartSite
  - HW#3 (Relational Design)
  - Project #2 (XML to Relational Mapping, Queries)
- Lectures:
  - Database Design Theory & Normalization
    - Functional and multi-valued dependencies
    - Normal forms (esp. BCNF, 4th NF)
    - Textbook chapter(s) on **Design Theory / Normalization**
      - [GMUW09, Ch.3] [SKS05, Ch.7] [EN10, Ch.15]
    - DB-class.org:
      - videos, quizzes on "Relational Design Theory"

Re. Homework

- This is wrong – why?
  - ans(X,Y,Z) :- R(X,Y,Z), S(X,Y,Z), X=Y
- NOT (x<y OR x>y)
  - Same as x = y.
  - Right? Yes, but only if domain (x,y) is totally ordered (not true in general)

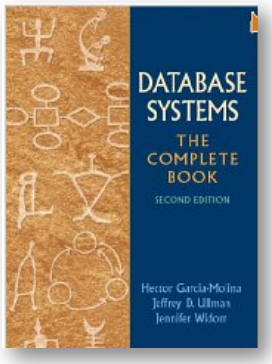
eqv. ans(X,X,Z) :- R(X,X,Z), S(X,X,Z)

not eqv. but intended? ans(X1, Z1, Z2) :- R(X1,Y1,Z1), S(X2,Y2,Z2), X1=Y2

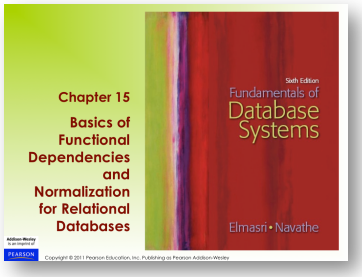
- XPath conditions are existential
  - "for **some**", e.g. …
  - $src//country[pop(…) >= $src //country//pop(..)] …
  - we need "all", not "some" here…
  - →Think XQuery constructs (XPath won't do)

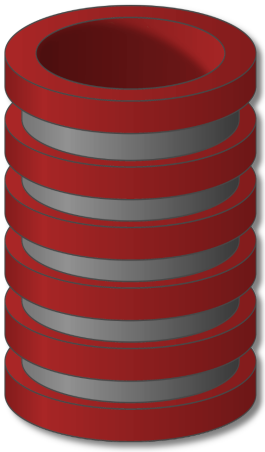
Database Design Theory & Normalization

Readings: Textbooks!



Ch. 3: Design Theory for Relational DBs





Relational Design Theory

Motivation & overview

- Watch videos (again)
- Do **online quizzes**!!

http://class2go.stanford.edu/db/Winter2013

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 …

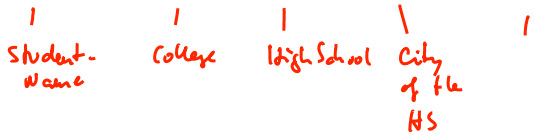
- Some designers go straight to relations
- Useful to understand why tools produce certain schemas

❖ 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, UCD.

Apply(123, Ann, Stanford, PAHS, Palo Alto, tennis)
GHS, -"-, trumpet
Stanford

2 * 4 * 2 = 16 tuples
HS college hobbies

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

Design "anomalies"

- Redundancy
- Update anomaly
- Deletion anomaly
- Insertion anomaly

Example: College application info.

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

```
Student(SSN, sName)
Apply(SSN, cName)
HighSchool(SSN, HS)
Located(HS, HScity)
Hobbies(SSN, hobby)
```

This decomposed schema has none of the above anomalies, yet allows to reconstruct the original information.

Design by decomposition

- Start with "mega" relations containing everything
- Decompose into smaller, better relations with same info.
- Can do decomposition automatically (given the dependencies among attributes)

Automatic decomposition "knowledge" about integrity constraints

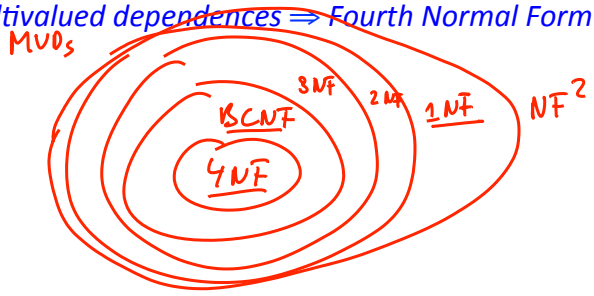
- "Mega" relations + *properties of the data*
- System decomposes based on properties
- Final set of relations satisfies *normal form*
  - *No anomalies, no lost information*

Rel. design - overview

Properties and Normal Forms

Functional dependencies ⇒ *Boyce-Codd Normal Form*

+ Multivalued dependences ⇒ *Fourth Normal Form*



"Each attribute must represent a fact about:
The Key, the Whole Key, and Nothing but the Key.
So help me Codd"

After: Edgar "Ted" Codd (1923–2003), English computer scientist, foundations of RDBMS.
[*A relational model of data for large shared data banks*, IBM,1970]., over 7300 GS citations

Rel. design - overview

Functional Dependencies and BCNF

`Apply(SSN, sName, cName)`

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

Functional Dependency `SSN → sName`

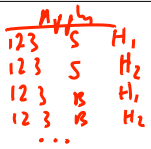
- Same `SSN` always has same `sName`
- Should store each `SSN`'s `sName` only once

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

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

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 ⇒ Boyce-Codd Normal Form
- Multivalued dependences ⇒ Fourth Normal Form