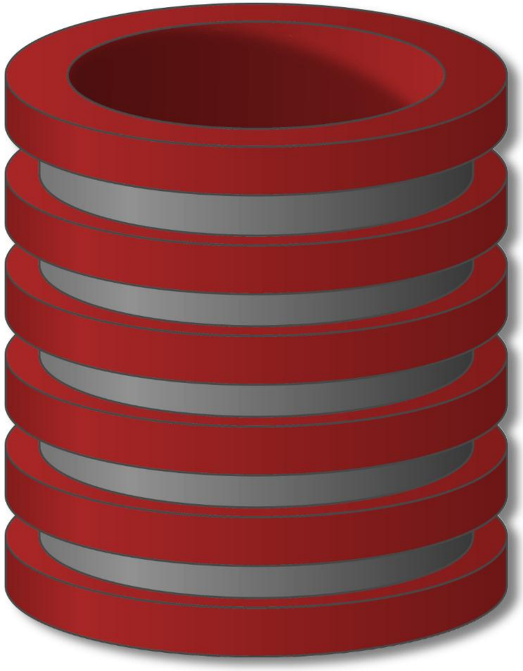


Relational Design Theory

Functional Dependencies

Jennifer Widom

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

Functional dependencies are generally useful concept

- Data storage – compression

Keys

- Reasoning about queries – optimization

Example: College application info.

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

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

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

Suppose **priority** is determined by **GPA**

$GPA > 3.8$ priority = 1

$3.3 < GPA \leq 3.8$ " = 2

$GPA \leq 3.3$ " = 3

Two tuples with same **GPA** have same **priority**

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

Two tuples with same **GPA** have same **priority**

$\forall t, u \in R:$

$t[A_1, \ldots, A_n] = u[A_1, \ldots, A_n] \Rightarrow t[B_1, \ldots, B_m] = u[B_1, \ldots, B_n]$

R

$\underbrace{A_1, A_2, \ldots, A_n}_{\bar{A}} \rightarrow \underbrace{B_1, B_2, \ldots, B_m}_{\bar{B}}$

Functional Dependency

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

$\boxed{\bar{A} \rightarrow \bar{B}}$ $R(\bar{A}, \bar{B}, \bar{C})$

$\bar{A}$	$\bar{B}$	$\bar{C}$
$\bar{a}$	$\bar{b}$	$\bar{c}_1$
$\bar{a}$	$\bar{b}$	$\bar{c}_2$

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

$\frac{\{1,2\}}{\{1,2\}}$

SSN → sName

SSN → address

HScode → HSname, HScity

HSname, HScity → HScode

SSN → GPA

GPA → priority

SSN → priority

more

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

cName → date

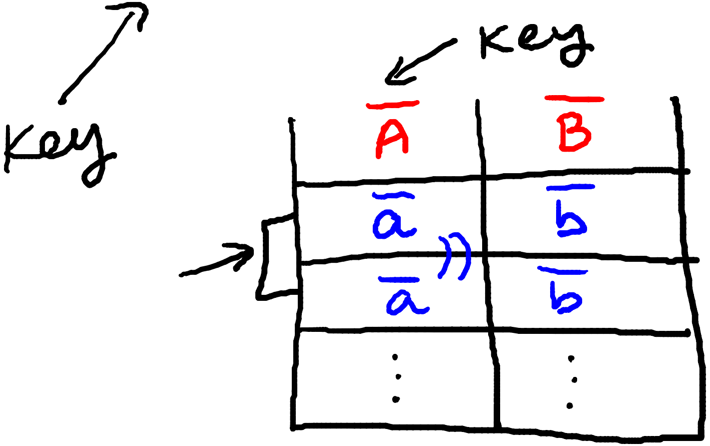
SSN, cName → major

SSN → state

Functional Dependencies and Keys

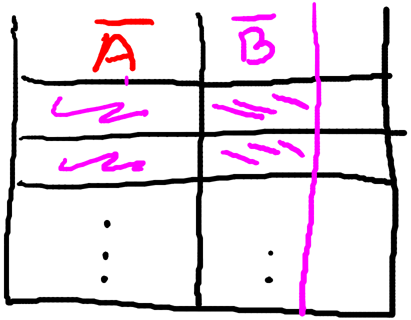
- Relation with no duplicates
- Suppose $\bar{A} \rightarrow$ all attributes

$R(\bar{A}, \bar{B})$



Trivial Functional Dependency

$\bar{A} \rightarrow \bar{B}$ $\quad \bar{B} \subseteq \bar{A}$



Nontrivial FD

$\bar{A} \rightarrow \bar{B}$ $\quad \bar{B} \not\subseteq \bar{A}$

Completely nontrivial FD

$\bar{A} \rightarrow \bar{B}$ $\quad \bar{A} \cap \bar{B} = \emptyset$

Rules for Functional Dependencies

Splitting rule

$\bar{A} \rightarrow B_1, B_2, \ldots, B_m$

$\Rightarrow \bar{A} \rightarrow B_1 \quad \bar{A} \rightarrow B_2 \quad \ldots$

Can we also split left-hand-side?

$A_1, A_2, \ldots, A_n \rightarrow \bar{B}$

? $A_1 \rightarrow \bar{B} \quad A_2 \rightarrow \bar{B}$

No

HSname, HScity → HScode

HSname → HScode

Rules for Functional Dependencies

Combining rule

$\bar{A} \rightarrow B_1$

$\bar{A} \rightarrow B_2$

$\vdots$

$\bar{A} \rightarrow B_n$

$\Rightarrow \quad \bar{A} \rightarrow B_1, \ldots, B_n$

Rules for Functional Dependencies

Trivial-dependency rules

$\bar{A} \rightarrow \bar{B}$ $\quad \bar{B} \subseteq \bar{A}$

$\bar{A} \rightarrow \bar{B}$ then $\bar{A} \rightarrow \bar{A} \cup \bar{B}$

$\bar{A} \rightarrow \bar{B}$ then $\bar{A} \rightarrow \bar{A} \cap \bar{B}$

splitting

Rules for Functional Dependencies

Transitive rule

$\bar{A} \rightarrow \bar{B}$ $\bar{B} \rightarrow \bar{C}$

then $\bar{A} \rightarrow \bar{C}$

$\bar{A}$	$\bar{B}$	$\bar{C}$	$\bar{D}$
$\bar{a}$	$\bar{b}$	$\bar{c}$	
$\bar{a}$	$\bar{b}$	$\bar{c}$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Closure of Attributes

- Given relation, FDs, set of attributes $\bar{A}$
- Find all B such that $\bar{A} \rightarrow B$

$\bar{A}^+$ $\{A_1, \ldots, A_n\}^+$

$A_4 \rightarrow C\,D$

$C \rightarrow E$

start with $\{A_1, \ldots, A_n, C, D, E\}$

repeat until no change:

if $\bar{A} \rightarrow \bar{B}$ and $\bar{A}$ in set

add $\bar{B}$ to set

Closure Example

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

SSN → sName, address, GPA

GPA → priority

HScode → HSname, HScity

$\{SSN, HScode\}^+ \rightarrow$ all attrs. Key

{SSN, HScode, sName, address, GPA, priority, HSname, HScity}

Closure and Keys

Is $\bar{A}$ a key for R ? → FDs

Compute $\bar{A}^+$ If = all attrs
then $\bar{A}$ is a key.

How can we find all keys given a set of FDs?

Consider every subset of attrs
↳ $\bar{A}$

ⓒ

$\bar{A}^+ \rightarrow$ all attrs
key

↑
increasing size

AB → all attrs

Specifying FDs for a relation

- S_1 and S_2 sets of FDs
- S_2 "follows from" S_1 if every relation instance satisfying S_1 also satisfies S_2

S_2: { SSN → priority }

S_1: { SSN → GPA, GPA → priority }

* How to test?

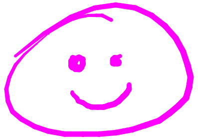
Does $\bar{A} \rightarrow \bar{B}$ follow from S ? S_1 S_2

(1) $\bar{A}^+$ based on S check if $\bar{B}$ in set.

(2) Armstrong's Axioms

Specifying FDs for a relation

Want: *Minimal set of completely nontrivial FDs such that all FDs that hold on the relation follow from the dependencies in this set*



Functional dependencies are generally useful concept

- Relational design by decomposition

  Functional dependencies $\Rightarrow$ Boyce-Codd Normal Form

- Data storage – compression
- Reasoning about queries – optimization