



Relational Design Theory

Functional Dependencies

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

Functional dependencies are generally useful concept

- Data storage – compression
- Reasoning about queries – optimization

keys

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**

$$\begin{aligned} \text{GPA} > 3.8 & \quad \text{priority} = 1 \\ 3.3 < \text{GPA} \leq 3.8 & \quad \text{"} = 2 \\ \text{GPA} \leq 3.3 & \quad \text{"} = 3 \end{aligned}$$

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

$$\forall t, u \in R : \\ t[A_1, \dots, A_n] = u[A_1, \dots, A_n] \Rightarrow t[B_1, \dots, B_m] = u[B_1, \dots, B_m]$$

$$\textcircled{\underline{R}} \quad \underbrace{A_1, A_2, \dots, A_n}_{\overline{A}} \rightarrow \underbrace{B_1, B_2, \dots, B_m}_{\overline{B}}$$

Functional Dependency

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

$$\overline{A} \rightarrow \overline{B} \quad R(\overline{A}, \overline{B}, \overline{C})$$

\overline{A}	\overline{B}	\overline{C}
\overline{a}	\overline{b}	\overline{c}_1
\overline{a}	\overline{b}	\overline{c}_2

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

123
123

SSN \rightarrow sName

SSN \rightarrow address \leftarrow

HScode \rightarrow HSname, HScity

HSname, HScity \rightarrow HScode

(SSN \rightarrow GPA
GPA \rightarrow priority
 \rightarrow SSN \rightarrow priority

more

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

$cName \rightarrow date$

$SSN, cName \rightarrow major$

$SSN \rightarrow state$

Functional Dependencies and Keys

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

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

key

key

	\bar{A}	\bar{B}
	\bar{a}	\bar{b}
→	\bar{a}	\bar{b}
	\vdots	\vdots

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

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

\bar{A}	\bar{B}
\swarrow	$\swarrow \swarrow \swarrow$
\swarrow	$\swarrow \swarrow \swarrow$
\vdots	\vdots

Rules for Functional Dependencies

Splitting rule

$$\rightarrow \overline{A} \rightarrow B_1, B_2, \dots, B_m \leftarrow$$

$$\Rightarrow \rightarrow \overline{A} \rightarrow B_1 \quad \overline{A} \rightarrow B_2 \quad \dots$$

Can we also split left-hand-side?

$$A_1, A_2, \dots, A_n \rightarrow \overline{B}$$

$H_{Sname} \rightarrow H_{Scode}$

? $A_1 \rightarrow \overline{B} \quad A_2 \rightarrow \overline{B}$ \nearrow

No

$H_{Sname}, H_{Scity} \rightarrow H_{Scode}$

Rules for Functional Dependencies

Combining rule

$$\begin{array}{l} \bar{A} \rightarrow B_1 \\ \bar{A} \rightarrow B_2 \\ \vdots \\ \bar{A} \rightarrow B_n \end{array}$$

$$\Rightarrow \bar{A} \rightarrow B_1, \dots, B_n$$

Rules for Functional Dependencies

Trivial-dependency rules

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

↑

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

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

↑ splitting

Rules for Functional Dependencies

Transitive rule

$$\boxed{\overline{A} \rightarrow \overline{B}} \quad \boxed{\overline{B} \rightarrow \overline{C}} \quad \leftarrow$$

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

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

Closure of Attributes

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

$$\bar{A}^+ \quad \{A_1, \dots, A_n\}^+ \quad A \rightarrow C, D \quad C \rightarrow E$$

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

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

$\{\underline{SSN}, \underline{HScode}\}^+ \rightarrow \text{all attrs.}$
key

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

Closure and Keys

Is \bar{A} a key for R ? \rightarrow FDs

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

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

③ Consider every subset of attrs
 $\bar{A}^+ \rightarrow$ all attrs
 key
 \uparrow increasing size
 \bar{A}
 $\textcircled{AB} \rightarrow$ 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

$S_2: \{ SSN \rightarrow priority \}$

★ How to test? $S_1: \{ \underline{SSN} \rightarrow GPA, GPA \rightarrow \underline{priority} \}$

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

(1) \bar{A}^+ based on \underline{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