

Unit 2 Dependencies and Inference

Model Theoretic View of Databases

A database schema is a theory or specification

The database state is a model or instance that satisfies the specification

- When is this view useful?
 - discussing constraints and implication
 - analyzing representation equivalence of two schemata
 - modeling incomplete information

CS 589, Principles of Database Systems, Unit 2, Notes 1

Principles of Database Systems

“Specification” Part

assigned(PILOT FLIGHT DATE TIME)

relation name (points to assigned)

relation scheme (points to PILOT)

attribute (points to FLIGHT)

Each attribute A has an $\text{dom}(A)$
So the relation scheme defines attributes and allowable values for a conforming relation instance

Unit 2: Notes 1 David Maier 3

Principles of Database Systems

Relation Instance

assigned(PILOT FLIGHT DATE TIME)

Cushing	83	9 Aug	10:15a
Cushing	116	10 Aug	1:25p
Clark	281	8 Aug	5:50a
Clark	301	12 Aug	6:35p
Clark	83	11 Aug	10:15a
Chin	83	13 Aug	10:15a
Chin	116	12 Aug	1:25p
Copley	281	9 Aug	5:50a
Copley	281	13 Aug	5:50a
Copley	412	15 Aug	1:25p

relation instance (points to the table)

Expect a relation instance to satisfy requirements of the schema
Data dependencies: further constraints that the schema puts on a satisfying instance

Unit 2: Notes 1 David Maier 4

Functional Dependency

assigned(PILOT	FLIGHT	DATE	TIME)
	Cushing	83	9 Aug	10:15a
	Clark	83	11 Aug	10:15a
	Chin	83	13 Aug	10:15a
	Cushing	116	10 Aug	1:25p
	Chin	116	12 Aug	1:25p
	Clark	281	8 Aug	5:50a
	Copley	281	9 Aug	5:50a
	Copley	281	13 Aug	5:50a
	Clark	301	12 Aug	6:35p
	Copley	412	15 Aug	1:25p

TIME →
FLIGHT
is not satisfied

A generalization of keys: when tuples agree on certain attributes, must agree on others

FLIGHT → TIME (L → T)
 PILOT DATE TIME → FLIGHT (PDT → L)
 FLIGHT DATE → PILOT (LD → P)

Functional Dependency Definition

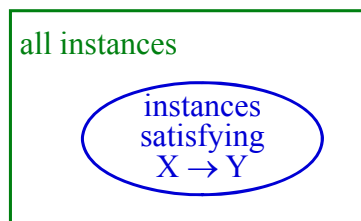
A relation instance $r(R)$ satisfies the *functional dependency* (FD) $X \rightarrow Y$ (X and Y subsets of $\text{schema}(R)$), if

for every two tuples t, s in r ,
 if $t[X] = s[X]$,
 then $t[Y] = s[Y]$

FDs as Specification

Usually, we don't ask which FDs a relation instance satisfies.

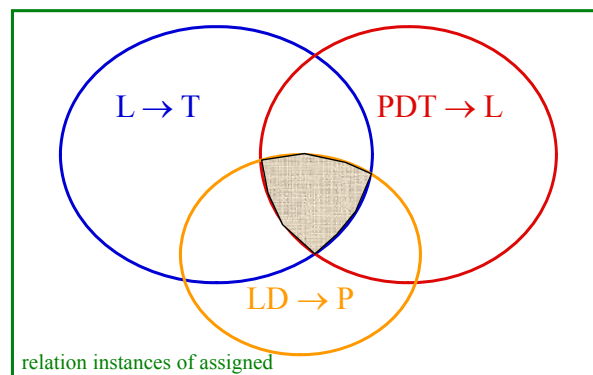
Rather, specify FDs that we expect all instances of to satisfy as part of the relation schema



Sets of FDs

A relation instance r satisfies a set of FDs F if it satisfies every FD in F

$$F = \{L \rightarrow T, PDT \rightarrow L, LD \rightarrow P\}$$



Implication

If an instance r satisfies FDs F , there may be another FD $X \rightarrow Y$ it necessarily satisfies

Say that F *implies* $X \rightarrow Y$ ($F \models X \rightarrow Y$)

Consider $L \rightarrow T$

Let r be a relation instance satisfying it

take t, s in r where $t[PL] = s[PL]$

so $t[L] = s[L]$

by $L \rightarrow T$, $t[T] = s[T]$

thus r also satisfies $PL \rightarrow T$

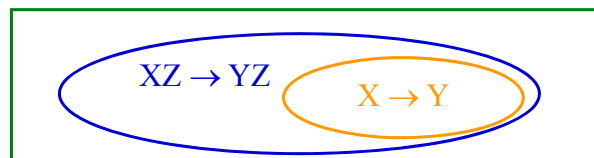
Inference Axioms

General patterns that describe implications.

For X, Y, Z, W sets of attributes

F1. Reflexivity: $X \rightarrow X$

F2. Augmentation: if $X \rightarrow Y$, then $XZ \rightarrow YZ$

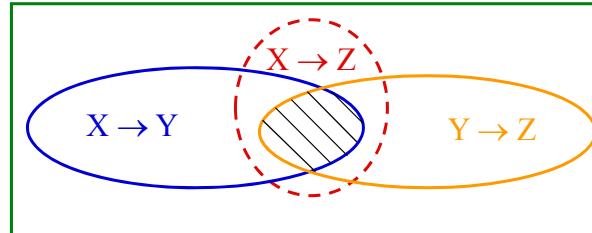


F3. Additivity: if $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$

More Axioms

F4. Projectivity: if $X \rightarrow YZ$, then $X \rightarrow Y$

F5. Transitivity: if $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$



F6. Pseudotransitivity: if $X \rightarrow Y$ and $YZ \rightarrow W$, then $XZ \rightarrow W$

Can Do Proofs

$F = \{L \rightarrow T, PDT \rightarrow L, LD \rightarrow P\}$

1. $L \rightarrow T$ (in F)
2. $LD \rightarrow TD$ (augmentation from 1.)
3. $LD \rightarrow P$ (in F)
4. $LD \rightarrow PTD$ (additivity from 2. and 3.)

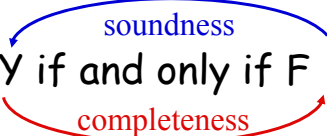
Derivation

Say that F *derives* $X \rightarrow Y$ if there is a proof of $X \rightarrow Y$ from FDs in F using inference axioms F1. - F6.

$F \vdash X \rightarrow Y$

Is implies the same as derives?

$F \models X \rightarrow Y$ if and only if $F \vdash X \rightarrow Y$



Soundness

Prove that each inference axiom is correct. That is, it holds in any relation instance.

F5. If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$

Take any tuples t, s in r , where r satisfies $X \rightarrow Y$ and $Y \rightarrow Z$.

Suppose $t[X] = s[X]$. Then from $X \rightarrow Y$, must have $t[Y] = s[Y]$.

But by $Y \rightarrow Z$, then have $t[Z] = s[Z]$.

So r satisfies $X \rightarrow Z$.

Definitions

Closure of F , F^+ , is F plus all FDs that can be derived from F by F1. - F6.

So $F \vdash X \rightarrow Y$ means $X \rightarrow Y$ in F^+

Closure of X , X^+ , is largest Z such that $X \rightarrow Z$ is in F^+

$\{L \rightarrow T, PDT \rightarrow L, LD \rightarrow P\}$

What is $(LD)^+ = LDPT$

What is $T^+ = T$

Completeness

If it's true, you can prove it

or

If you can't prove it, it's not true.

Instance to show not true

Take any $X \rightarrow Y$ not in F^+ over R .
 Consider X^+ and let $X^- = R - X^+$.

	X+ atts		X- atts	
t	a1 a2 ... an	b1 b2 ... bm		
s	a1 a2 ... an	c1 c2 ... cn		
				$b_i \neq c_i$

violates
 $X^+ \rightarrow X^-$

This table violates $X \rightarrow Y$.
 $t[X] = s[X]$, but claim $t[Y] \neq s[Y]$
 if $t[Y] = s[Y]$, then $Y \subseteq X^+$
 then since $X \rightarrow X^+$ in F^+ , so is $X \rightarrow Y$ by
 projectivity, a contradiction

The instance satisfies F^+

Take any $W \rightarrow Z$ in F^+

Case 1. $W \not\subseteq X^+$.

Then $t[W] \neq s[W]$, no problem

Case 2. $W \subseteq X^+$.

Then $t[W] = s[W]$

Have $X \rightarrow X^+$ in F^+ , $X^+ \rightarrow W$ by reflexivity
 and projectivity, and $W \rightarrow Z$ in F^+ .

That gives $X \rightarrow Z$ by transitivity applied
 twice

Additivity on $X \rightarrow X^+$ and $X \rightarrow Z$ gives
 $X \rightarrow X^+Z$

$X^+ \rightarrow X^+$

Case 2, continued

We have proved $X \rightarrow X^+Z$ is in F^+

$$X^+Z = X^+$$

But wait! X^+ is supposed to be the maximum set of attributes that X determines.

Must be that $Z \subseteq X^+$.

So $t[Z] = s[Z]$

Thus our instance satisfies $W \rightarrow Z$.

Hence $F \not\models X \rightarrow Y$

Armstrong Relation for F

An "exact model" for F :

- Satisfies F^+
- Violates every FD in F^-

Use the 2-tuple instances in the last proof, combined into one big instance

Example $R = ABC$, $F = \{A \rightarrow B, B \rightarrow C\}$

Only need FDs in F^- with one attribute on right side

Armstrong Relation

$C \rightarrow A, C \rightarrow B, B \rightarrow A, BC \rightarrow A$

A	B	C
a1	b1	c1
a2	b2	c1

a3	b3	c2
a4	b4	c2

a5	b5	c3
a6	b5	c3

a7	b6	c4
a8	b6	c4

Join Dependency

Note the redundancy in the assigned table:
repeat time for each flight.

Can split up

asgn1(PILOT	FLT	DATE)	asgn2(FLT	TIME)
Cushing	83	9 Aug	83	10:15a
Clark	83	11 Aug	116	1:25p
Chin	83	13 Au	281	5:50a
Cushing	116	10 Aug	301	6:35p
Chin	116	12 Aug	412	1:25p
Clark	281	8 Aug		
Copley	281	9 Aug		
Copley	281	13 Aug		
Clark	301	12 Aug		
Copley	412	15 Aug		

Can Recover with a Join

$assigned = asgn1 \bowtie asgn2$

assigned satisfies a join dependency (JD)

$\bowtie [PLD, LT]$

$assigned \stackrel{\subseteq \text{ always}}{=} \pi_{PLD}(assigned) \bowtie \pi_{LT}(assigned)$

A "lossless" join

What about $\bowtie [PL, PDT]$?

$asgn3(PILOT \quad FLT)$	$asgn4(PILOT \quad DATE \quad TIME)$
Cushing 83	Cushing 9 Aug 10:15a
Cushing 116	Cushing 10 Aug 1:25p
...	...

$asgn3 \bowtie asgn4(PILOT \quad FLT \quad DATE \quad TIME)$	<i>lossy join</i>
Cushing 83 9 Aug 10:15a	
Cushing 83 10 Aug 1:25p	
...	

Join Dependency, General Form

$r(R)$, with R_1, R_2, \dots, R_n subsets of R
 r satisfies $\bowtie[R_1, R_2, \dots, R_n]$ if
 $r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \dots \bowtie \pi_{R_n}(r)$

notice that always have

$$r \subseteq \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \dots \bowtie \pi_{R_n}(r)$$

A multivalued dependency is a special case where $n = 2$.

Implications of FDs and JDs

There are some axioms

$X \rightarrow Y$ on R , $Z = R - XY$

FJ1. $X \rightarrow Y$ implies $\bowtie[XY, XZ]$

lossless decomposition

Example

$L \rightarrow T$, so $\bowtie[LT, LPD]$

However, there is no finite, complete set of axioms for just FDs and JDs (or JDs alone).

Are We Stuck?

No

- There are complete axiom sets for FDs and multi-valued dependencies.
- There are complete axiom sets for classes of constraints that include FDs and JDs.
- There is an *inference procedure* for FD and JD implication.
not based on axioms

Need to Use Tableaux

A "template" for a relation

Has *rows* of variables instead of tuples of values

	<u>P</u>	<u>L</u>	<u>D</u>	<u>T</u>
v	p1	f1	d1	m1
w	p2	f1	d2	m2

Treat tableau rows like tuples

$$w(L) = f1$$

Valuation

A valuation ρ for tableau U maps its variables to domain values.

$\rho(v)$ must be in $\text{dom}(A)$ if v in A column

$\rho(p1)=\text{Chin}$ $\rho(f1)=86$ $\rho(d1)=1$ June $\rho(m1)=4p$
 $\rho(p2)=\text{Chao}$ $\rho(d2)=2$ June $\rho(m2)=4p$

Extend Valuation to Rows and Tableaux

$\rho(w)$ = apply ρ to each variable in w

$\rho(w)(A) = \rho(w(A))$

$\rho(p1\ f1\ d1\ m1) = \langle \text{Chin}\ 86\ 1\text{June}\ 4p \rangle$

$\rho(U) = \{\rho(w) \mid w \text{ in } U\}$ — some tuples could collapse

$\rho(U)$	<u>P</u>	<u>L</u>	<u>D</u>	<u>T</u>
$\rho(v)$	Chin	86	1June	4p
$\rho(w)$	Chao	86	2June	4p

How About This Valuation?

$\rho(p1) = \text{Chin}$ $\rho(f1) = 86$ $\rho(d1) = 1$ June $\rho(m1) = 4p$
 $\rho(p2) = \text{Chao}$ $\rho(d2) = 2$ June $\rho(m2) = 5p$

$\rho(U)$	<u>P</u>	<u>L</u>	<u>D</u>	<u>T</u>		
$\rho(v)$	Chin	86	1	June	4p	$\left. \begin{array}{l} \text{---} \\ \text{---} \end{array} \right\} \neq$ $L \rightarrow T$
$\rho(w)$	Chao	86	2	June	5p	

Want to Enforce $L \rightarrow T$

Need to make sure that $\rho(m1) = \rho(m2)$.

Can we "fix" the tableau?

U	<u>P</u>	<u>L</u>	<u>D</u>	<u>T</u>
v	p1	f1	d1	m1
w	p2	f1	d2	m1

Have "applied" $L \rightarrow T$ to tableau U.

The Chase

Apply FDs and JDs to a tableau, to reason about an example relation (or sub-relation)

Tableau rule for FD $X \rightarrow Y$

For rows v and w

If $v[X] = w[X]$, make $v[Y] = w[Y]$ by equating variables in Y

[Book gives rules for replacing variables]

FD chase of tableau U for F : Apply FD rules for FDs in F to U until no change.

Example

$F =$
 $\{AB \rightarrow E, AG \rightarrow C, BE \rightarrow D, E \rightarrow G,$
 $DG \rightarrow H\}$

A	B	C	D	E	G	H
$a1$	$a2$	$a3$	$a4$	$a5$	$a6$	$a7$
$a1$	$b2$	$b3$	$b4$	$a5$	$b6$ $a6$	$b7$

Apply $E \rightarrow G$

A	B	C	D	E	G	H
$a1$	$a2$	$a3$	$a4$	$a5$	$a6$	$a7$
$a1$	$b2$	$b3$	$b4$	$a5$	<u>$a6$</u>	$b7$

Apply $AG \rightarrow C$

A	B	C	D	E	G	H
a1	a2	a3	a4	a5	a6	a7
a1	b2	a3	b4	a5	a6	b7

Testing FD Implication with the Chase

Method in book is ambiguous

Want to test $F \models X \rightarrow Y$ on scheme

$R = A_1 A_2 \dots A_n$

1. Set up a tableau U_X with rows v, w :
 $v(A_i) = a_i$
 $w(A_i) = a_i$ if A_i in X
 $= b_i$ otherwise
2. Chase U_X with FDs in F
3. If $v[Y] = w[Y]$, then $F \models X \rightarrow Y$
 (if not, we have a counterexample)

CS 589, Principles of Database Systems, Unit 2, Notes 1

Principles of Database Systems

Example

$F = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A\} \not\models BG \rightarrow C?$

A	B	C	D	E	G	H
a1	a2	a3	a4	a5	a6	a7
b1 a1	a2	b3 a3	b4 a4	b5 a5	a6	b7 a7

$BG \rightarrow A$

Unit 2: Notes 1

David Maier

37

Principles of Database Systems

You Try It

$F = \{AB \rightarrow E, AG \rightarrow C, BE \rightarrow D, E \rightarrow G, DG \rightarrow H\}$

A	B	C	D	E	G	H
a1	b1	c1	d1	e1	g1	h1
a2	b1	c2	d2 a1	e1	g2 g1	h2h1

$F \models C \rightarrow A$

Does $F \models BE \rightarrow A$? no

Does $F \models BE \rightarrow H$? (can use same chase)
yes

Unit 2: Notes 1

David Maier

38