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

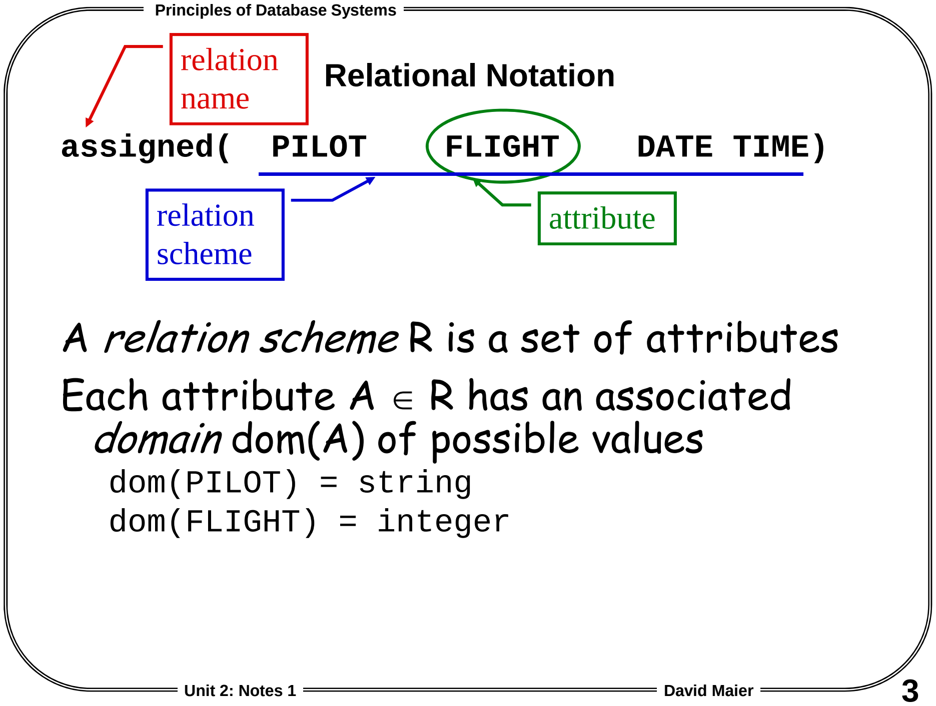
Unit 2
Dependencies and Inference

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



Relational Notation

relation name

`assigned(  PILOT     FLIGHT     DATE TIME)`

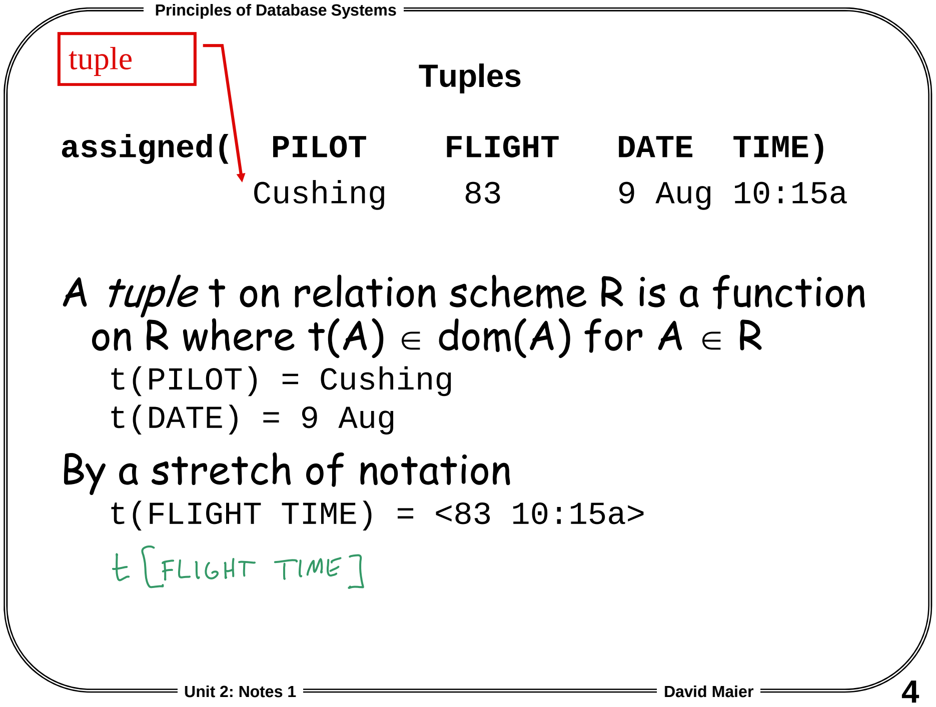
relation scheme

attribute

A *relation scheme* R is a set of attributes

Each attribute $A \in R$ has an associated *domain* dom(A) of possible values

```
dom(PILOT) = string
dom(FLIGHT) = integer
```



tuple

Tuples

```
assigned( PILOT      FLIGHT    DATE   TIME)
         Cushing      83       9 Aug  10:15a
```

A *tuple* t on relation scheme R is a function on R where $t(A) \in dom(A)$ for $A \in R$

```
t(PILOT) = Cushing
t(DATE) = 9 Aug
```

By a stretch of notation

```
t(FLIGHT TIME) = <83 10:15a>
```

t[FLIGHT TIME]

relation
instance

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

A *relation instance* r on scheme R is a set of tuples on R. Denote $r(R)$.

relation = name + scheme + instance

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
T → L ✗	Cushing	116	10 Aug	1:25p
	Chin	116	12 Aug	1:25p
DT → L ✓	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

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

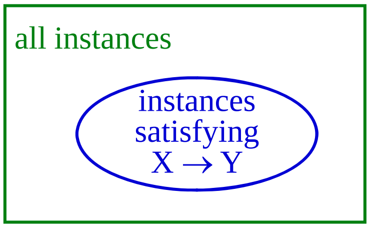
A relation instance r(R) satisfies the *functional dependency* (FD) X → Y (X and Y subsets of schema(R)), if

```
for any 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.

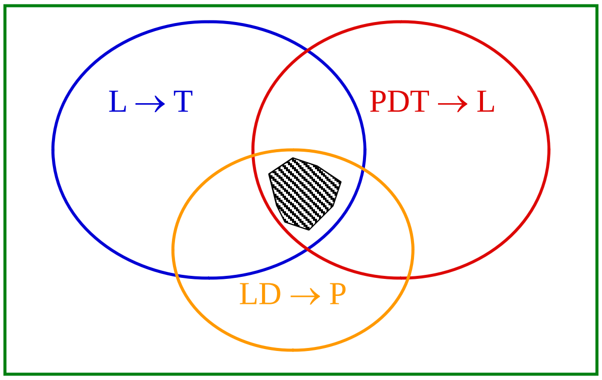
Rather, specify FDs that we expect all instances of to satisfy as part of the relation scheme



Sets of FDs

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

F = {L → T, PDT → L, LD → P}

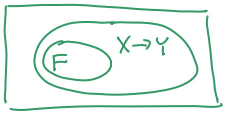


Implication

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

Say that F *implies* X → Y (F ⊨ X → Y)

Consider L → 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 → T, t(T) = s(T)
thus r also satisfies PL → T



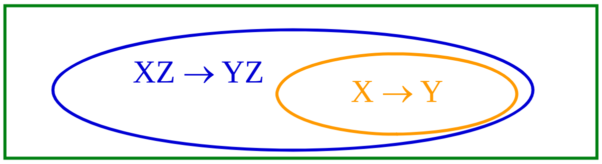
Inference Axioms

General patterns that describe implications.
For X, Y, Z, W sets of attributes

F1. Reflexivity: $X \to X$

shorthand $X \cup Z$

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



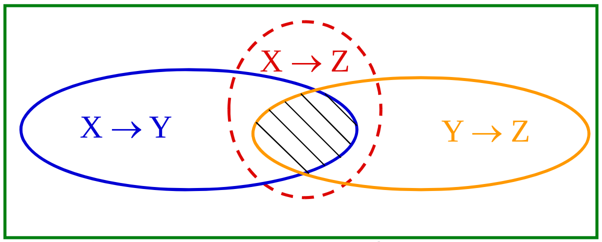
F3. Additivity: if $X \to Y$ and $X \to Z$, then
$X \to YZ$

More Axioms

$YZ = ZY$ $X \to Z$

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

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



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

X is stronger than Y

Can Do Proofs

F = {L → T, PDT → L, LD → P}

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

Derivation

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

$F \underbrace{\vdash} X \rightarrow Y$

Is implication 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 → Y and Y → Z, then X → Z

Take t,s in r, where r satisfies X → Y and Y → Z.

Suppose t(X) = s(X). Then from X → Y, must have t(Y) = s(Y).

But by Y → Z, then have t(Z) = s(Z).

So r satisfies X → Z.

Definitions

set of FDs

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

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

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

{L → T, PDT → L, LD → P}

What is $(LD)^+$ = LDTP

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

If I can't derive $X \to Y$ from F, then $\underbrace{F \not\models X \to Y}_{\text{counter example to } F \models X \to Y}$

Take any $X \to Y$ 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

bi ≠ ci

This table violates $X \to Y$.

$X^+ \supseteq X$ t(X) = s(X), but claim t(Y) ≠ s(Y)

if t(Y) = s(Y), then $Y \subseteq X^+$

then since $X \to X^+$ in F^+, so is $X \to Y$ by projectivity, a contradiction

The instance satisfies F^+

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

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

- 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 $X \rightarrow X^+Z$

Case 2, continued

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

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 → B, B → C}

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

r violates $C \to A \notin F^+$

$$\underbrace{C \to AB \qquad C \to AC \qquad C \to ABC}_{\text{all violated, too!}}$$

Armstrong Relation

C → A, C → B, B → A, BC → A

FDs in F^- with one attribute on right side

A	B	C	
a1	b1	c1	C → A
a2	b2	c1	$C^+ = C$
a3	b3	c2	C → B
a4	b4	c2	C^+
a5	b5	c3	B → A
a6	b5	c3	$B^+ = BC$
a7	b6	c4	BC → A
a8	b6	c4	$BC^+ = BC$

F = {A → B, B → C}

Join Dependency

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

Can split up

asgn1(PILOT	FLT	DATE)
Cushing	83	9 Aug
Clark	83	11 Aug
Chin	83	13 Aug
Cushing	116	10 Aug
Chin	116	12 Aug
Clark	281	8 Aug
Copley	281	9 Aug
Copley	281	13 Aug
Clark	301	12 Aug
Copley	412	15 Aug

asgn2(FLT	TIME)
83	10:15a
116	1:25p
281	5:50a
301	6:35p
412	1:25p

Can Recover with a Join

assign^{ed} = asgn1 ⋈ asgn2

assign satisfies a *join dependency* (JD) ⋈[PLD, LT]

$$\text{assign} = \pi_{PLD}(\text{assign}) \bowtie \pi_{LT}(\text{assign})$$

A “lossless” join

lossy join

What about ⋈[PL, PDT]?

violated

```
asgn3(PILOT   FLT) asgn4(PILOT    DATE   TIME)
     Cushing   83  ──>  Cushing   9 Aug 10:15a
     Cushing 116   ──>  Cushing 10 Aug   1:25p
       . . .              . . .


asgn3*asgn4(PILOT   FLT   DATE    TIME)
       ─>  Cushing   83   9 Aug   10:15a
       ─>  Cushing   83 10 Aug    1:25p X
                      . . .
```

⋈

$r \subseteq \pi_{PL}(r) \bowtie \pi_{PDT}(r)$

Not in original relation instance

Join Dependency, General Form

$R1 \cup R2 \cup \dots \cup Rn = R$

r(R), with R1, R2, …, Rn subsets of R

r satisfies ⋈[R1, R2, …, Rn] if

$r = \pi_{R1}(r) \bowtie \pi_{R2}(r) \bowtie \dots \bowtie \pi_{Rn}(r)$

notice that always have

$r \subseteq \pi_{R1}(r) \bowtie \pi_{R2}(r) \bowtie \dots \bowtie \pi_{Rn}(r)$

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

Implications of FDs and JDs

There are some axioms

$X \to Y$ on R, $Z = R - XY$
FJ1. $X \to Y$ implies $\bowtie[XY, XZ]$

Example
$L \to T$, 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.

Need to Use Tableaux

Tableau

A "template" for a relation

Has *rows* of variables instead of tuples of values

	U(P	L	D	T)
v	*p1*	*f1*	*d1*	*m1*
w	*p2*	*f1*	*d2*	*m2*

Treat tableau rows like tuples

w(L) = *f1*

Valuation

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

ρ(v) in dom(A) if v in A column

$\rho(p1)$=Chin $\rho(f1)$=86 $\rho(d1)$=1 June $\rho(m1)$=4p

$\rho(p2)$=Chao $\rho(d2)$=2 June $\rho(m2)$=4p

Extend Valuation to Rows and Tableaux

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

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

$\rho(p1\ f1\ d1\ m1)$=<Chin 86 1June 4p>

tableau

$\rho(U) = \{\rho(w) \mid w \text{ in } U\}$

ρ(U)(	P	L	D	T)
ρ(v)	Chin	86	1June	4p
ρ(w)	Chao	86	2June	4p

How About This Valuation?

$\rho(p1)$=Chin $\rho(f1)$=86 $\rho(d1)$=1 June $\rho(m1)$=4p

$\rho(p2)$=Chao $\rho(d2)$=2 June $\rho(m2)$=5p

ρ(U)(	P	L	D	T)
ρ(v)	Chin	86	1June	4p
ρ(w)	Chao	86	2June	5p

L → T

Want to Enforce L → T

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

Can we "fix" the tableau?

	U(P	L	D	T)
v	*p1*	*f1*	*d1*	*m1*
w	*p2*	*f1*	*d2*	***m1***

Have "applied" L → 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 → 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 → E, AG → C, BE → D, E → G,
DG → H}

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

Apply $E \to G$

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

Apply $AG \to C$

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

Testing FD Implication with the Chase

Warning! Method in book is wrong!

Want to test $F \models X \to Y$ on scheme
$R = \underbrace{A1\ A2\ \ldots\ An}$ columns

1. Set up a tableau with rows v, w:
   v(Ai) = ai
   w(Ai) = ai if Ai in X
   = bi otherwise
   $\}$ Tableau U_X
2. Chase U_X with FDs in F
3. If v(Y) = w(Y), then $F \models X \to Y$
   (if not, we have a counterexample)

Example

$F = \{AB \to C,\ B \to D,\ CD \to E,\ CE \to GH,\ G \to A\} \models \underline{BG \to C}$?

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

$BG \to H$

You Try It

$\{AB \overset{x}{\to} E,\ AG \overset{x}{\to} C,\ BE \overset{\checkmark}{\to} D,\ E \overset{\checkmark}{\to} G,\ DG \underset{\checkmark}{\to} H\}$

	A	B	C	D	E	G	H
	~~a1~~	~~b1~~	~~c1~~	~~d1~~	~~e1~~	~~g1~~	~~h1~~
✓	a1 1	a2 3	a3 4	a4 6	a5 7	a6 8	a7 9
w	b1 2	a2 3	b3 5	~~b4~~ 6 a4	a5 7	~~b6~~ 8 a6	~~b7~~ 9 a7

$(BE)^+ = BDEGH$

Does BE → A?

Does BE → H? (can use same chase)