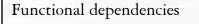


## Announcements (September 12)

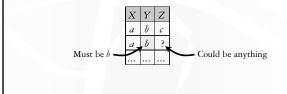
- ✤ Homework #1 due next Tuesday
- \* Help session this Wednesday
  - 4:30pm or 5:30pm?
  - D344 LSRC
  - Email reminder tonight
- Course project assigned today
  - Choice of "standard" or "open"
  - Milestone 1 right after fall break
    But plan/start early!!!

Motivation			3
	SID name	CID	
	142 Bart	CPS116	
	142 Bart	CPS114	
	857 Lisa	CPS116	
	857 Lisa	CPS130	
<ul> <li>How do we tel StudentEnroll (S</li> </ul>	0	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
	· · · · · · · · · · · · · · · · · · ·	ecause the name of a stu for each course the stude	
<ul> <li>How about a synchrony removing reduced</li> </ul>		oach to detecting an gns?	d

Dependencies, decompositions, and normal forms



- \* A functional dependency (FD) has the form  $X \to Y$ , where X and Y are sets of attributes in a relation R
- \*  $X \to Y$  means that whenever two tuples in R agree on all the attributes in X, they must also agree on all attributes in Y



# FD examples Address (street\_address, city, state, zip)

## Keys redefined using FD's

A set of attributes K is a key for a relation R if

- $\bigstar K \rightarrow$  all (other) attributes of R
  - That is, K is a "super key"
- \* No proper subset of K satisfies the above condition
  - That is, K is minimal

### Reasoning with FD's

Given a relation R and a set of FD's  $\mathcal{F}$ 

- $\bullet$  Does another FD follow from  $\mathcal{F}$ ?
  - Are some of the FD's in  $\mathcal{F}$  redundant (i.e., they follow from the others)?
- \* Is K a key of R?
  - What are all the keys of R?

## Attribute closure

 $\diamond$  Given *R*, a set of FD's  $\mathcal{F}$  that hold in *R*, and a set of attributes *Z* in *R*:

The closure of Z (denoted  $Z^+$ ) with respect to  $\mathcal{F}$  is the set of all attributes  $\{A_1, A_2, \ldots\}$  functionally determined by Z (that is,  $Z \rightarrow A_1 A_2 \ldots$ )

- \* Algorithm for computing the closure
  - Start with closure = Z
  - If  $X \to Y$  is in  $\mathcal{F}$  and X is already in the closure, then also add Y to the closure
  - Repeat until no more attributes can be added

# A more complex example

StudentGrade (SID, name, email, CID, grade)

(Not a good design, and we will see why later)

# Example of computing closure

#### ${\boldsymbol{\ast}} \; {\mathcal{F}} \, {\rm includes} :$

- SID  $\rightarrow$  name, email
- $email \rightarrow SID$
- SID, CID  $\rightarrow$  grade
- $\{ CID, email \}^+ = ?$
- $\bullet$  email  $\rightarrow$  SID
  - Add SID; closure is now { CID, email, SID }
- $\texttt{SID} \rightarrow \textit{name}, \textit{email}$ 
  - Add name, email; closure is now { CID, email, SID, name }
- $\texttt{SID}, CID \rightarrow grade$ 
  - Add grade; closure is now all the attributes in StudentGrade

11

12

## Using attribute closure

Given a relation R and set of FD's  ${\cal F}$ 

 $\diamond$  Does another FD  $X \to Y$  follow from  $\mathcal{F}$ ?

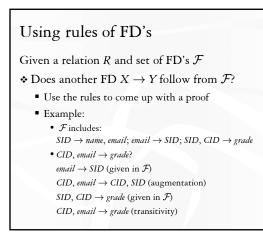
- ${\ensuremath{\,^{-}}}$  Compute  $X^+$  with respect to  ${\ensuremath{\mathcal F}}$
- If  $Y \subseteq X^+$ , then  $X \to Y$  follow from  $\mathcal{F}$

 $\bullet$  Is K a key of R?

## Rules of FD's

#### \* Armstrong's axioms

- Reflexivity: If  $Y \subseteq X$ , then  $X \to Y$
- Augmentation: If  $X \to Y$ , then  $XZ \to YZ$  for any Z
- Transitivity: If  $X \to Y$  and  $Y \to Z$ , then  $X \to Z$
- Rules derived from axioms
  - Splitting: If  $X \to YZ$ , then  $X \to Y$  and  $X \to Z$
  - Combining: If  $X \to Y$  and  $X \to Z$ , then  $X \to YZ$



13

14

15

## Non-key FD's

- ♦ Consider a non-trivial FD  $X \rightarrow Y$  where X is not a super key
  - Since *X* is not a super key, there are some attributes (say *Z*) that are not functionally determined by *X*

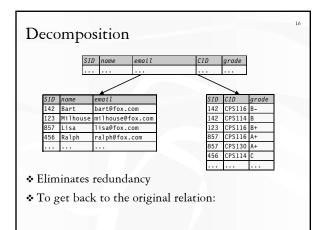


That a is always associated with b is recorded by multiple rows: redundancy, update anomaly, deletion anomaly

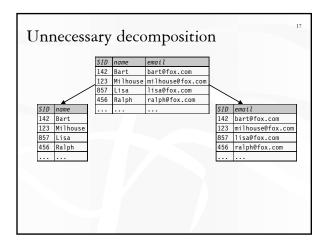
# Example of redundancy

- StudentGrade (SID, name, email, CID, grade)
- $\Rightarrow$  SID  $\rightarrow$  name, email

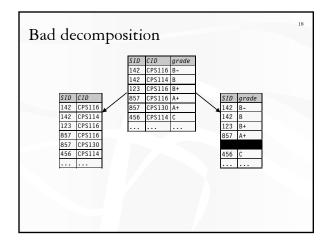
142 142	Bart Bart	bart@fox.com bart@fox.com	CPS116 CPS114	B
123			CPS114	
357	Lisa	lisa@fox.com	CPS116	A+
857	Lisa	lisa@fox.com	CPS130	A+
456	Ralph	ralph@fox.com	CPS114	С

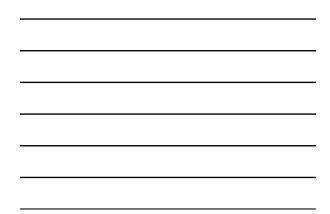






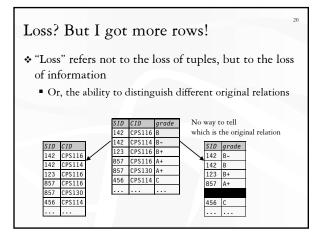






### Lossless join decomposition

- \* Decompose relation R into relations S and T
  - $attrs(R) = attrs(S) \cup attrs(T)$
  - $S = \pi_{attrs(S)}(R)$
  - $T = \pi_{attrs(T)}(R)$
- ♦ The decomposition is a lossless join decomposition if, given known constraints such as FD's, we can guarantee that  $R = S \bowtie T$
- Any decomposition gives R ⊆ S ⋈ T (why?)
  A lossy decomposition is one with R ⊂ S ⋈ T



## Questions about decomposition

- \* When to decompose
- How to come up with a correct decomposition (i.e., lossless join decomposition)

21

#### An answer: BCNF

 $\clubsuit$  A relation R is in Boyce-Codd Normal Form if

• For every non-trivial FD  $X \to Y$  in R, X is a super key

22

24

• That is, all FDs follow from "key  $\rightarrow$  other attributes"

#### \* When to decompose

- As long as some relation is not in BCNF
- How to come up with a correct decomposition
  - Always decompose on a BCNF violation (details next)
     Then it is guaranteed to be a lossless join decomposition!

## BCNF decomposition algorithm

- \* Find a BCNF violation
  - That is, a non-trivial FD  $X \to Y$  in R where X is not a super key of R
- \* Decompose R into  $R_1$  and  $R_2$ , where
  - $R_1$  has attributes  $X \cup Y$
  - $R_2$  has attributes  $X \cup Z$ , where Z contains all attributes of R that are in neither X nor Y

\* Repeat until all relations are in BCNF

## BCNF decomposition example

StudentGrade (SID, name, email, CID, grade) BCNF violation:  $SID \rightarrow name$ , email

## Another example

StudentGrade (SID, name, email, CID, grade) BCNF violation:  $email \rightarrow SID$ 

## Why is BCNF decomposition lossless

- Given non-trivial  $X \rightarrow Y$  in R where X is not a super key of R, need to prove:
- ♦ Anything we project always comes back in the join:  $R \subseteq \pi_{XY}(R) \bowtie \pi_{XZ}(R)$ 
  - Sure; and it doesn't depend on the FD
- Anything that comes back in the join must be in the original relation:
  - $R\supseteq\pi_{XY}(R)\bowtie\pi_{XZ}(R)$
  - Proof makes use of the fact that  $X \to Y$

#### Recap

- Functional dependencies: a generalization of the key concept
- Non-key functional dependencies: a source of redundancy
- BCNF decomposition: a method for removing redundancies
  - BNCF decomposition is a lossless join decomposition
- BCNF: schema in this normal form has no redundancy due to FD's