

Relational Database Design

CPS 216
Advanced Database Systems

Announcements

- Homework #1 out today
 - Due next Thursday in class
- Sign up to present a research paper
 - Sign-up sheet available in my office (D327) during my office hours
 - First-come, first-serve
 - Participation is voluntary
 - Allows you to drop your lowest homework grade
 - In groups of 2-4

2

Relational model: a review

- A database is a collection of relations (or tables)
- Each relation has a list of attributes (or columns)
- Each attribute has a domain (or type)
- Each relation contains a set of tuples (or rows)

3

Keys

- A set of attributes K is a key for a relation R if
 - In no instance of R will two different tuples agree on all attributes of K
 - That is, K is a “tuple identifier”
 - No proper subset of K satisfies the above condition
 - That is, K is minimal
- Example: *Student* (SID , $name$, age , GPA)

4

More examples of keys

- *Enroll* (SID , CID)
- *Address* ($street_address$, $city$, $state$, zip)

5

Schema versus data

Student

<i>SID</i>	<i>name</i>	<i>age</i>	<i>GPA</i>
142	Bart	10	2.3
123	Milhouse	10	3.1
857	Lisa	8	4.3
456	Ralph	8	2.3
...

- Is *name* a key of *Student*?
 - Yes?
 - No!
- Key declarations are part of the schema

6

Usage of keys

- More constraints on data, fewer mistakes
 - Look up a row by its key value
 - Many selection conditions are “key = value”
 - “Pointers”
 - Example: *Enroll (SID, CID)*
- Many join conditions are “key = key value stored in another table”

7

Functional dependencies

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

8

FD examples

Address (street_address, city, state, zip)

9

Keys redefined using FDs

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

- - That is, K is a “super key”
- No proper subset of K satisfies the above condition
 - That is, K is minimal

10

Reasoning with FDs

Given a relation R and set of FDs F

- Does another FD follow from F ?
 - Are some of the FDs in F redundant (because they follow from the others)?
- Is K a key of R ?
 - What are all the keys of R ?

11

Attribute closure

- Given R , a set of FDs F that holds in R , and a set of attributes Z in R : The closure of Z with respect to F (denoted Z^+) is the set of all attributes functionally determined by Z
- Algorithm for computing the closure

12

A more complex example

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

- $SID \rightarrow name, email$
- $email \rightarrow SID$
- $SID, CID \rightarrow grade$
- Not a good design, and we will see why later

13

Example of computing closure

- $\{ CID, email \}^+ = ?$

14

Using attribute closure

Given a relation R and set of FDs F

- Does another FD $X \rightarrow Y$ follow from F ?
- Is K a key of R ?

15

Rules of FDs

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

16

Using rules of FDs

Given a relation R and set of FDs F

- Does another FD $X \rightarrow Y$ follow from F ?
 - Use the rules to come up with a proof
 - Example: $CID, email \rightarrow grade$?

17

Non-key FDs

- 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

X	Y	Z
a	b	c1
a	b	c2
...

The fact that “a” is always associated with “b” is recorded in multiple rows: redundancy!

18

Problems with redundancy

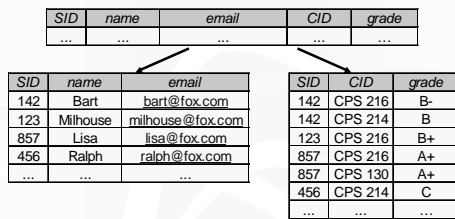
StudentGrade (SID, name, email, CID, grade)

SID → *name, email*

<i>SID</i>	<i>name</i>	<i>email</i>	<i>CID</i>	<i>grade</i>
142	Bart	bart@fox.com	CPS 216	B-
142	Bart	bart@fox.com	CPS 214	B
123	Milhouse	milhouse@fox.com	CPS 216	B+
857	Lisa	lisa@fox.com	CPS 216	A+
857	Lisa	lisa@fox.com	CPS 130	A+
456	Ralph	ralph@fox.com	CPS 214	C
...

19

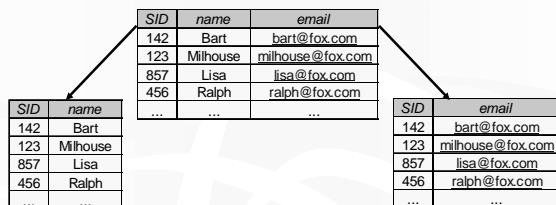
Decomposition



- Eliminates redundancy
- To get back to the original relation:

20

Unnecessary decomposition



21

Bad decomposition

<i>SID</i>	<i>CID</i>	<i>SID</i>	<i>CID</i>	<i>grade</i>	<i>SID</i>	<i>grade</i>
142	CPS 216	142	CPS 216	B-	142	B-
142	CPS 214	142	CPS 214	B	142	B
123	CPS 216	123	CPS 216	B+	123	B+
857	CPS 216	857	CPS 216	A+	857	A+
857	CPS 130	857	CPS 130	A+	857	A+
456	CPS 214	456	CPS 214	C	456	C
...

22

Lossless join decomposition

- Suppose that R is decomposed into S and T

$$\text{attrs}(R) = \text{attrs}(S) \cup \text{attrs}(T)$$

$$S = \pi_{\text{attrs}(S)}(R)$$

$$T = \pi_{\text{attrs}(T)}(R)$$
- It is a lossless join decomposition if, given constraints such as FDs, we can guarantee
$$R = S \bowtie T$$

23

Loss? But I got more rows!

- “Loss” refers not to the loss of tuples, but to the loss of information
 - Or, the ability to distinguish different original relations

<i>SID</i>	<i>CID</i>	<i>SID</i>	<i>CID</i>	<i>grade</i>	<i>SID</i>	<i>grade</i>
142	CPS 216	142	CPS 216	B-	142	B-
142	CPS 214	142	CPS 214	B	142	B
123	CPS 216	123	CPS 216	B+	123	B+
857	CPS 216	857	CPS 216	A+	857	A+
857	CPS 130	857	CPS 130	A+	857	A+
456	CPS 214	456	CPS 214	C	456	C
...

24

Questions about decomposition

- When to decompose
- How to come up with a correct decomposition

25

An answer: BCNF

- A relation R is in Boyce-Codd Normal Form if
 - For every non-trivial FD $X \rightarrow Y$ in R , X is a super key
 - 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
 - Then it’s a lossless join decomposition!

26

BCNF decomposition algorithm

- Find a BCNF violation
 - That is, a non-trivial FD $X \rightarrow Y$ 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 $X \cup Z$ (Z contains all attributes of R that are in neither X nor Y)
- Repeat until all relations are in BCNF

27

BCNF decomposition example

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

28

Another example

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

29

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:
 $R \supseteq \pi_{XY}(R) \bowtie \pi_{XZ}(R)$

30

Yet another example

- *Address* (*street_address*, *city*, *state*, *zip*)
 - *street_address*, *city*, *state* → *zip*
 - *zip* → *city*, *state*
- Keys
- BCNF?

31

To decompose, or not to decompose

*Address*₁ (*zip*, *city*, *state*)

*Address*₂ (*street_address*, *zip*)

- FDs in *Address*₁
- FDs in *Address*₂

32

“Elegant” solution

- Define the problem away!
- *R* is in Third Normal Form (3NF) if for every non-trivial FD $X \rightarrow A$, either
 - *X* is super key of *R*, or
 - *A* is a member of at least one key of *R*
- So *Address* is already in 3NF
- Tradeoff

33

Recap

- Identifying tuples: keys
- Generalizing the key concept: FDs
- Non-key FDs: redundancy
- Avoiding redundancy: BCNF decomposition
- Preserving FDs: 3NF

34

What's next

- Another kind of dependency and normal form
- A comprehensive design example
- SQL basics

35
