Purdue University Department of Computer Science

CS 44800: Introduction To Relational Database Systems

Prof. Chris Clifton

5 February 2021

Multivalued Dependencies

Indiana Center for Database Systems ™

Purdue University Department of Computer Science

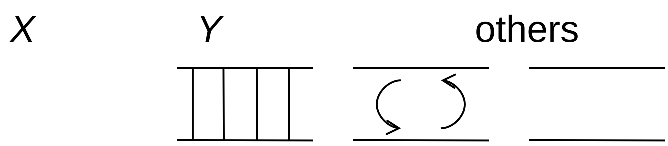
Multivalued Dependencies (MVDs)

- Suppose we record names of children, and phone numbers for parents:
  - *parent_child(parent_name, child_name)*
  - *parent_phone(parent_name, phone_number)*
- Suppose we were to combine these schemas to get
  - *parent_info(parent_name, child_name, phone_number)*
  - Example data:
    (Chris, Eric, 765-555-1234)
    (Patty, Eric, 765-555-1234)
    (Chris, Eric, 765-555-4321)
    (Chris, Denise, 765-555-1234)
    (Patty, Denise, 765-555-1234
    (Chris, Denise, 765-555-4321)
- Is this relation BCNF? 3NF?

Multivalued Dependencies

PURDUE UNIVERSITY
Department of Computer Science

The *multivalued dependency* $X \rightarrow\rightarrow Y$ holds in a relation R if whenever we have two tuples of R that agree in all the attributes of X, then we can swap their Y components and get two new tuples that are also in R.



71

Multivalued Dependencies

- Let R be a relation schema and let $\alpha \subseteq R$ and $\beta \subseteq R$. The **multivalued dependency**

  $$\alpha \rightarrow\rightarrow \beta$$

  holds on R if in any legal relation $r(R)$, for all pairs for tuples t_1 and t_2 in r such that $t_1[\alpha] = t_2[\alpha]$, there exist tuples t_3 and t_4 in r such that:

  $$\begin{aligned} t_1[\alpha] = t_2[\alpha] &= t_3[\alpha] = t_4[\alpha] \\ t_3[\beta] &= t_1[\beta] \\ t_3[R - \beta] &= t_2[R - \beta] \\ t_4[\beta] &= t_2[\beta] \\ t_4[R - \beta] &= t_1[R - \beta] \end{aligned}$$

- Note that since the behavior of Z and W are identical it follows that

  $\alpha \rightarrow\rightarrow \beta$ implies $\alpha \rightarrow\rightarrow R - \beta$

MVD -- Tabular representation

- Tabular representation of $\alpha \rightarrow\rightarrow \beta$

	α	β	$R - \alpha - \beta$
t_1	$a_1 \ldots a_i$	$a_{i+1} \ldots a_j$	$a_{j+1} \ldots a_n$
t_2	$a_1 \ldots a_i$	$b_{i+1} \ldots b_j$	$b_{j+1} \ldots b_n$
t_3	$a_1 \ldots a_i$	$a_{i+1} \ldots a_j$	$b_{j+1} \ldots b_n$
t_4	$a_1 \ldots a_i$	$b_{i+1} \ldots b_j$	$a_{j+1} \ldots a_n$

PURDUE UNIVERSITY
Department of Computer Science

Example

- In our example:

  parent_name $\rightarrow\rightarrow$ *child_name*
  parent_name $\rightarrow\rightarrow$ *phone_number*

- The above formal definition is supposed to formalize the notion that given a particular value of *Y* (*parent_name*) it has associated with it a set of values of *Z (child_name)* and a set of values of *W (phone_number)*, and these two sets are in some sense independent of each other.
- Note:
  - If $Y \rightarrow Z$ then $Y \rightarrow\rightarrow Z$
  - Indeed we have (in above notation) $Z_1 = Z_2$
    The claim follows.

Use of Multivalued Dependencies

- We use multivalued dependencies in two ways:
  1. To test relations to **determine** whether they are legal under a given set of functional and multivalued dependencies
  2. To specify **constraints** on the set of legal relations. We shall concern ourselves *only* with relations that satisfy a given set of functional and multivalued dependencies.
- If a relation *r* fails to satisfy a given multivalued dependency, we can construct a relations *r′* that does satisfy the multivalued dependency by adding tuples to *r.*

Restriction of Multivalued Dependencies

- The restriction of D to R_i is the set D_i consisting of
  - All functional dependencies in D^+ that include only attributes of R_i
  - All multivalued dependencies of the form

    $\alpha \rightarrow\rightarrow (\beta \cap R_i)$

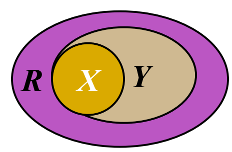
    where $\alpha \subseteq R_i$ and $\alpha \rightarrow\rightarrow \beta$ is in D^+

Purdue University
Department of Computer Science

4NF

Eliminate redundancy due to multiplicative effect of MVD's.

- Roughly: treat MVD's as FD's for decomposition, but not for finding keys.
- Formally: *R* is in Fourth Normal Form if whenever MVD $X \rightarrow\rightarrow Y$ is *nontrivial* (Y is not a subset of X, and $X \cup Y$ is not all attributes), then X is a superkey.
  - Remember, $X \rightarrow Y$ implies $X \rightarrow\rightarrow Y$, so 4NF is more stringent than BCNF.
- Decompose *R*, using 4NF violation $X \rightarrow\rightarrow Y$, into XY and $X \cup (R—Y)$.



Fourth Normal Form

- A relation schema *R* is in **4NF** with respect to a set *D* of functional and multivalued dependencies if for all multivalued dependencies in D^+ of the form $\alpha \rightarrow\rightarrow \beta$, where $\alpha \subseteq R$ and $\beta \subseteq R$, at least one of the following hold:
  - $\alpha \rightarrow\rightarrow \beta$ is trivial (i.e., $\beta \subseteq \alpha$ or $\alpha \cup \beta = R$)
  - α is a superkey for schema *R*
- If a relation is in 4NF it is in BCNF

PURDUE UNIVERSITY
Department of Computer Science

4NF Decomposition

- Schema $S = R$, $D+$ be the closure of the functional and multivalued dependencies
- While $\exists R_i \in S$ not in 4NF w.r.t. $D+$
  - Choose a nontrivial multivalued dependency $A \rightarrow\rightarrow$ B that holds on R_i, where $A \rightarrow R_i \notin D+$, and $A \cap B = \emptyset$
  - $S = (S - R_i) \cup (R_i\text{-}B) \cup (A,B)$

85

Example

- $R = (A, B, C, G, H, I)$
  $F = \{ A \rightarrow\rightarrow B$
  $B \rightarrow\rightarrow HI$
  $CG \rightarrow\rightarrow H \}$
- R is not in 4NF since $A \rightarrow\rightarrow B$ and A is not a superkey for R
- Decomposition
  a) $R_1 = (A, B)$ — (R_1 is in 4NF)
  b) $R_2 = (A, C, G, H, I)$ — (R_2 is not in 4NF, decompose into R_3 and R_4)
  c) $R_3 = (C, G, H)$ — (R_3 is in 4NF)
  d) $R_4 = (A, C, G, I)$ — (R_4 is not in 4NF, decompose into R_5 and R_6)
  - $A \rightarrow\rightarrow B$ and $B \rightarrow\rightarrow HI \rightarrow A \rightarrow\rightarrow HI$, (MVD transitivity), and
  - and hence $A \rightarrow\rightarrow I$ *(MVD restriction to R_4)*

  e) $R_5 = (A, I)$ — (R_5 is in 4NF)
  f) $R_6 = (A, C, G)$ — (R_6 is in 4NF)

PURDUE UNIVERSITY
Department of Computer Science

Final Notes (Date & Fagin '92)

- If a relation is in BCNF, and it has a key consisting of a single attribute, it is also in 4NF
  - *And you don't need to see if there are multivalued dependencies*
  - But if all keys have 2+ attributes, look for possible MVDs
- Fifth-Normal Form (also called project-join normal form)
  - Enforces lossless join in terms of dependencies
  - Defined over entire schema, not a single relation

Overall Database Design Process

We have assumed schema *R* is given

- *R* could have been generated when converting E-R diagram to a set of tables.
- *R* could have been a single relation containing *all* attributes that are of interest (called **universal relation**).
- Normalization breaks *R* into smaller relations.
- *R* could have been the result of some ad hoc design of relations, which we then test/convert to normal form.

Other Design Issues

- Some aspects of database design are not caught by normalization
- Examples of bad database design, to be avoided:

  Instead of *earnings* (*company_id, year, amount*), use
  - *earnings_2004, earnings_2005, earnings_2006*, etc., all on the schema (*company_id, earnings*).
    - Above are in BCNF, but make querying across years difficult and needs new table each year
  - *company_year* (*company_id, earnings_2004, earnings_2005, earnings_2006*)
    - Also in BCNF, but also makes querying across years difficult and requires new attribute each year.
    - Is an example of a **crosstab**, where values for one attribute become column names
    - Used in spreadsheets, and in data analysis tools

Modeling Temporal Data

- **Temporal data** have an association time interval during which the data are *valid.*
- A **snapshot** is the value of the data at a particular point in time
- Several proposals to extend ER model by adding valid time to
  - attributes, e.g., address of an instructor at different points in time
  - entities, e.g., time duration when a student entity exists
  - relationships, e.g., time during which an instructor was associated with a student as an advisor.
- But no accepted standard
- Adding a temporal component results in functional dependencies like

  $ID \rightarrow street, city$

  not holding, because the address varies over time
- A **temporal functional dependency** $X \rightarrow Y$ holds on schema *R* if the functional dependency X → Y holds on all snapshots for all legal instances r (*R*).

Modeling Temporal Data (Cont.)

- In practice, database designers may add start and end time attributes to relations
  - E.g., *course*(*course_id, course_title*) is replaced by

    course(*course_id, course_title, start, end*)
  - Constraint: no two tuples can have overlapping valid times
    - Hard to enforce efficiently
- Foreign key references may be to current version of data, or to data at a point in time
  - E.g., student transcript should refer to course information at the time the course was taken

Design Goals

- Goal for a relational database design is:
  - BCNF.
  - Lossless join.
  - Dependency preservation.
- If we cannot achieve this, we accept one of
  - Lack of dependency preservation
  - Redundancy due to use of 3NF
- Interestingly, SQL does not provide a direct way of specifying functional dependencies other than superkeys.

  Can specify FDs using assertions, but they are expensive to test, (and currently not supported by any of the widely used databases!)
- Even if we had a dependency preserving decomposition, using SQL we would not be able to efficiently test a functional dependency whose left hand side is not a key.