NORMALIZATION

BSC(PS)IInd Sem (DBMS)

Deepak mittal

Objective

- Normalization presents a set of rules that tables and databases must follow to be well structured.
- Historically presented as a sequence of normal forms

First Normal From

- A table is in the first normal form iff
  - The domain of each attribute contains only ***atomic values***, and
  - The value of each attribute contains only a ***single value*** from that domain.

In layman's terms. it means every column of your table should only contain *single values*

Example

- For a library

Patron ID	Borrowed books
C45	B33, B44, B55
C12	B56

1-NF Solution

Patron ID	Borrowed book
C45	B33
C45	B44
C45	B33
C12	B56

Example

- For an airline

Flight	Weekdays
UA59	Mo We Fr
UA73	Mo Tu We Th Fr

1NF Solution

Flight	Weekday
UA59	Mo
UA59	We
UA59	Fr
UA73	Mo
UA73	We
...	...

Implication for the ER model

- Watch for entities that can have multiple values for the same attribute
  - Phone numbers, …
- What about course schedules?
  - MW 5:30-7:00pm
    - Can treat them as *atomic time slots*

Functional dependency

Let X and Y be *sets* of attributes in a table *T*

- Y is **functionally dependent** on *X* in *T* **iff** for each set x ∈ R.X there is precisely one corresponding set y∈ R.Y
- Y is **fully functional dependent** on *X* in *T* if *Y* is functional dependent on X and Y is not functional dependent on any proper subset of *X*

Example

- Book table

BookNo	Title	Author	Year
B1	Moby Dick	H. Melville	1851
B2	Lincoln	G. Vidal	1984

Author attribute is:

- ***functionally dependent*** on the pair { BookNo, Title}
- ***fully functionally dependent*** on BookNo

Why it matters

- table BorrowedBooks

BookNo	Patron	Address	Due
B1	J. Fisher	101 Main Street	3/2/15
B2	L. Perez	202 Market Street	2/28/15

Address attribute is

- ***functionally dependent*** on the pair { BookNo, Patron}
- ***fully functionally dependent*** on Patron

Problems

- Cannot insert new patrons in the system until they have borrowed books
  - ***Insertion anomaly***
- Must update all rows involving a given patron if he or she moves.
  - ***Update anomaly***
- Will lose information about patrons that have returned all the books they have borrowed
  - ***Deletion anomaly***

Armstrong inference rules (1974)

- ***Axioms:***
  - Reflexivity: if $Y \subseteq X$, then $X \rightarrow Y$
  - Augmentation: if $X \rightarrow Y$, then $WX \rightarrow WY$
  - Transitivity: if $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$
- ***Derived Rules:***
  - Union: if $X \rightarrow Y$ and $X \rightarrow Z$, the $X \rightarrow YZ$
  - Decomposition: if $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$
  - Pseudotransitivity: if $X \rightarrow Y$ and $WY \rightarrow Z$, then $XW \rightarrow Z$

Armstrong inference rules (1974)

- Axioms are both
  - ***Sound:***
    when applied to a set of functional dependencies they only produce dependency tables that belong to the transitive closure of that set
  - ***Complete:***
    can produce all dependency tables that belong to the transitive closure of the set

Armstrong inference rules (1974)

- Three last rules can be derived from the first three (the axioms)
- Let us look at the ***union rule***:

  if $X \rightarrow Y$ and $X \rightarrow Z$, the $X \rightarrow YZ$
- Using the first three axioms, we have:
  - if $X \rightarrow Y$, then $XX \rightarrow XY$ same as $X \rightarrow XY$ (2nd)
  - if $X \rightarrow Z$, then $YX \rightarrow YZ$ same as $XY \rightarrow YZ$ (2nd)
  - if $X \rightarrow XY$ and $XY \rightarrow YZ$, then $X \rightarrow YZ$ (3rd)

Second Normal Form

- A table is in 2NF iff
  - It is in 1NF and
  - no non-prime attribute is dependent on any proper subset of any candidate key of the table
- A ***non-prime attribute*** of a table is an attribute that is not a part of any candidate key of the table
- A ***candidate key*** is a minimal superkey

Example

- Library allows patrons to request books that are currently out

BookNo	Patron	PhoneNo
B3	J. Fisher	555-1234
B2	J. Fisher	555-1234
B2	M. Amer	555-4321

Example

- Candidate key is {BookNo, Patron}
- We have
  - Patron $\rightarrow$ PhoneNo
- Table is not 2NF
  - Potential for
    - Insertion anomalies
    - Update anomalies
    - Deletion anomalies

2NF Solution

- Put telephone number in separate Patron table

BookNo	Patron
B3	J. Fisher
B2	J. Fisher
B2	M. Amer

Patron	PhoneNo
J. Fisher	555-1234
M. Amer	555-4321

Third Normal Form

- A table is in 3NF iff
  - it is in 2NF and
  - all its attributes are determined only by its candidate keys and not by any non-prime attributes

Example

- Table BorrowedBooks

BookNo	Patron	Address	Due
B1	J. Fisher	101 Main Street	3/2/15
B2	L. Perez	202 Market Street	2/28/15

- Candidate key is BookNo
- Patron $\rightarrow$ Address

3NF Solution

- Put address in separate Patron table

BookNo	Patron	Due
B1	J. Fisher	3/2/15
B2	L. Perez	2/28/15

Patron	Address
J. Fisher	101 Main Street
L. Perez	202 Market Street

Another example

- Tournament winners

Tournament	Year	Winner	DOB
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 Sept. 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975

- Candidate key is {Tournament, Year}
- Winner $\rightarrow$DOB

Lossless Decomposition

General Concept

- If R(A, B, C) satisfies $A \rightarrow B$
  - We can project it on A,B and A,C ***without losing information***
  - Lossless decomposition
- $R = \pi_{AB}(R) \bowtie \pi_{AC}(R)$
  - $\pi_{AB}(R)$ is the projection of R on AB
  - $\bowtie$ is the natural join operator

Example

R

Course	*Instructor*	*Text*
4330	Paris	none
4330	Cheng	none
3330	Hillford	Patterson & Hennessy

- Observe that Course $\rightarrow$ Text

A lossless decomposition

$\pi_{\text{Course, Text}}(R)$

Course	*Text*
4330	none
3330	Patterson & Hennessy

$\pi_{\text{Course, Instructor}}(R)$

Course	*Instructor*
4330	Paris
4330	Cheng
3330	Hillford

A different case

R

Course	*Instructor*	*Text*
4330	Paris	Silberschatz and Peterson
4330	Cheng	none
3330	Hillford	Patterson & Hennessy

- Now Course $\not\twoheadrightarrow$ Text
- R cannot be decomposed

A lossy decomposition

$\pi_{\text{Course, Text}}(R)$

Course	*Text*
4330	none
4330	Silberschatz & Peterson
3330	Patterson & Hennessy

$\pi_{\text{Course, Instructor}}(R)$

Course	*Instructor*
4330	Paris
4330	Cheng
3330	Hillford

An Example

Normalisation Example

- We have a table representing orders in an online store
- Each row represents an item on a particular order
- Primary key is {Order, Product}

- Columns
  - <u>Order</u>
  - <u>Product</u>
  - Quantity
  - UnitPrice
  - Customer
  - Address

Functional Dependencies

- Each order is for a single customer:
  - Order $\rightarrow$ Customer
- Each customer has a single address
  - Customer $\rightarrow$ Address
- Each product has a single price
  - Product $\rightarrow$ UnitPrice
- As Order $\rightarrow$ Customer and Customer $\rightarrow$ Address
  - Order $\rightarrow$ Address

2NF Solution (I)

- ***First decomposition***
  - First table

Order	Product	Quantity	UnitPrice

  - Second table

Order	Customer	Address

2NF Solution (II)

- ***Second decomposition***
  - First table

Order	Product	Quantity

  - Second table

Order	Customer	Address

  - Third table

Product	UnitPrice

3NF

- In second table

<u>Order</u>	Customer	Address

  - Customer → Address
- Split second table into

<u>Order</u>	Customer

<u>Customer</u>	Address

Normalisation to 2NF

- Second normal form means no partial dependencies on candidate keys
  - {Order} $\rightarrow$ {Customer, Address}
  - {Product} $\rightarrow$ {UnitPrice}
- To remove the first FD we project over

  {Order, Customer, Address} (R1)

and

{Order, Product, Quantity, UnitPrice} (R2)

Normalisation to 2NF

- R1 is now in 2NF, but there is still a partial FD in R2

  {Product} $\rightarrow$ {UnitPrice}

- To remove this we project over

{Product, UnitPrice} (R3)

and

{Order, Product, Quantity} (R4)

Normalisation to 3NF

- R has now been split into 3 relations - R1, R3, and R4
  - R3 and R4 are in 3NF
  - R1 has a transitive FD on its key
- To remove

  {Order} → {Customer} → {Address}
- we project R1 over
  - {Order, Customer}
  - {Customer, Address}

Normalisation

- 1NF:
  - {<u>Order, Product</u>, Customer, Address, Quantity, UnitPrice}
- 2NF:
  - {<u>Order</u>, Customer, Address}, {<u>Product</u>, UnitPrice}, and {<u>Order, Product</u>, Quantity}
- 3NF:
  - {<u>Product</u>, UnitPrice}, {<u>Order, Product</u>, Quantity}, {<u>Order</u>, Customer}, and {<u>Customer</u>, Address}