Database Management Systems

Module 26: Relational Database Design/6: Normal Forms

Partha Pratim Das

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

ppd@cse.iitkgp.ac.in

Week Recap

- Identified the features of good relational design
- Familiarized with the First Normal Form
- Introduced the notion and the theory of functional dependencies
- Discussed issues in "good" design in the context of functional dependencies
- Studied Algorithms for Properties of Functional Dependencies
- Understood the Characterization for and Determination of Lossless Join and Determination of Dependency Preservation

Module Objectives

- To Understand the Normal Forms and their Importance in Relational Design

Module Outline

- Normal Forms

Normal Forms

Normalization or Schema Refinement

- Normalization or Schema Refinement is a technique of organizing the data in the database
- A systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics
  - Insertion Anomaly
  - Update Anomaly
  - Deletion Anomaly
- Most common technique for the Schema Refinement is decomposition.
  - Goal of Normalization: Eliminate Redundancy
- Redundancy refers to repetition of same data or duplicate copies of same data stored in different locations
- Normalization is used for mainly two purpose:
  - Eliminating redundant (useless) data
  - Ensuring data dependencies make sense, that is, data is logically stored

Anomalies

a) **Update Anomaly:** Employee 519 is shown as having different addresses on different records

Employees' Skills

Employee ID	Employee Address	Skill
426	87 Sycamore Grove	Typing
426	87 Sycamore Grove	Shorthand
519	94 Chestnut Street	Public Speaking
519	96 Walnut Avenue	Carpentry

Resolution: Decompose the Schema

a) *Update*: (ID, Address), (ID, Skill)

b) *Insert*: (ID, Name, Hire Date), (ID, Code)

c) *Delete*: (ID, Name, Hire Date), (ID, Code)

b) **Insertion Anomaly**: Until the new faculty member, Dr. Newsome, is assigned to teach at least one course, his details cannot be recorded

Faculty and Their Courses

Faculty ID	Faculty Name	Faculty Hire Date	Course Code
389	Dr. Giddens	10-Feb-1985	ENG-206
407	Dr. Saperstein	19-Apr-1999	CMP-101
407	Dr. Saperstein	19-Apr-1999	CMP-201
424	Dr. Newsome	29-Mar-2007	?

c) **Deletion Anomaly**: All information about Dr. Giddens is lost if he temporarily ceases to be assigned to any courses.

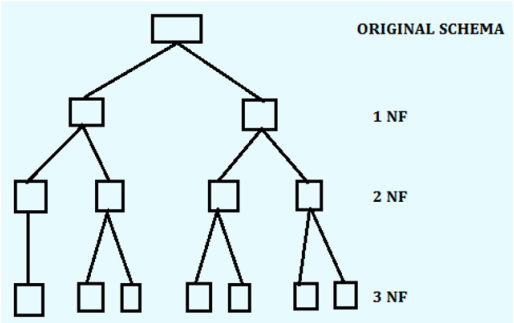
Faculty and Their Courses

Faculty ID	Faculty Name	Faculty Hire Date	Course Code
389	Dr. Giddens	10-Feb-1985	ENG-206
407	Dr. Saperstein	19-Apr-1999	CMP-101
407	Dr. Saperstein	19-Apr-1999	CMP-201

DELETE

Desirable Properties of Decomposition

- Lossless Join Decomposition Property
  - It should be possible to reconstruct the original table
- Dependency Preserving Property
  - No functional dependency (or other constraints should get violated)



Normalization and Normal Forms

- A normal form specifies a set of conditions that the relational schema must satisfy in terms of its constraints – they offer varied levels of guarantee for the design
- Normalization rules are divided into various normal forms. Most common normal forms are:
  - First Normal Form (1NF)
  - Second Normal Form (2NF)
  - Third Normal Form (3NF)
- Informally, a relational database relation is often described as "normalized" if it meets third normal form. Most 3NF relations are free of insertion, update, and deletion anomalies

Normalization and Normal Forms

- Additional Normal Forms
  - Elementary Key Normal Form (EKNF)
  - **Boyce-codd Normal Form (BCNF)**
  - **Multivalued Dependencies And Fourth Normal Form (4NF)**
  - Essential Tuple Normal Form (ETNF)
  - Join Dependencies and Fifth Normal Form (5NF)
  - Sixth Normal Form (6NF)
  - Domain/Key Normal Form (DKNF)

1NF: First Normal Form

- A relation is in First Normal Form if and only if all underlying domains contain atomic values only (doesn't have multivalued attributes (MVA))
- **STUDENT(Sid, Sname, Cname)**

Students

SID	Sname	Cname
S1	A	C,C++
S2	B	C++, DB
S3	A	DB

SID : Primary Key

MVA exists ⇒ Not in 1NF

Students

SID	Sname	Cname
S1	A	C
S1	A	C++
S2	B	C++
S2	B	DB
S3	A	DB

SID, Cname : Primary Key

No MVA ⇒ In 1NF

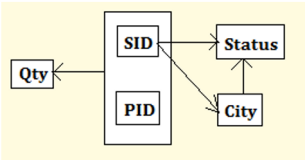
Source: http://www.edugrabs.com/normal-forms/#fnf

- Example: **Supplier(SID, Status, City, PID, Qty)**

Supplier:

SID	Status	City	PID	Qty
S1	30	Delhi	P1	100
S1	30	Delhi	P2	125
S1	30	Delhi	P3	200
S1	30	Delhi	P4	130
S2	10	Karnal	P1	115
S2	10	Karnal	P2	250
S3	40	Rohtak	P1	245
S4	30	Delhi	P4	300
S4	30	Delhi	P5	315

Key : (SID, PID)



Drawbacks:

- Deletion Anomaly: If we delete <S3,40,Rohtak,P1,245>, then we lose the information that S3 lives in Rohtak.
- Insertion Anomaly: We cannot insert a Supplier S5 located in Karnal, until S5 supplies at least one part.
- Update Anomaly: If Supplier S1 moves from Delhi to Kanpur, then it is difficult to update all the tuples having SID as S1 and City as Delhi.

Normalization is a method to reduce redundancy. However, sometimes 1NF increases redundancy.

1NF (3): Possible Redundancy

- **When LHS is not a Superkey** :
  - Let $X \rightarrow Y$ be a non trivial FD over R with X is not a superkey of R, then redundancy exist between X and Y attribute set.
  - Hence in order to identify the redundancy, we need not to look at the actual data, it can be identified by given functional dependency.
  - Example : $X \rightarrow Y$ and X is not a Candidate Key
    $\Rightarrow X$ can duplicate
    $\Rightarrow$ Corresponding Y value would duplicate also.

X	Y
1	3
1	3
2	3
2	3
4	6

- **When LHS is a Superkey** :
  - If $X \rightarrow Y$ is a non trivial FD over R with X is a superkey of R, then redundancy does not exist between X and Y attribute set.
  - Example : $X \rightarrow Y$ and X is a Candidate Key
    $\Rightarrow X$ cannot duplicate
    $\Rightarrow$ Corresponding Y value may or may not duplicate.

X	Y
1	4
2	6
3	4

2NF: Second Normal Form

- Relation $\boldsymbol{R}$ is in Second Normal Form (2NF) only iff :
  - $\boldsymbol{R}$ is in 1NF and
  - $\boldsymbol{R}$ contains no Partial Dependency

Partial Dependency:

Let $\boldsymbol{R}$ be a relational Schema and $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{A}$ be the attribute sets over $\boldsymbol{R}$ where $\boldsymbol{X}$: Any Candidate Key, $\boldsymbol{Y}$: Proper Subset of Candidate Key, and $\boldsymbol{A}$: Non Prime Attribute

If $Y \rightarrow A$ exists in R, then R is not in 2NF.

($\boldsymbol{Y} \rightarrow \boldsymbol{A}$) is a Partial dependency only if

- $\boldsymbol{Y}$: Proper subset of Candidate Key
- $\boldsymbol{A}$: Non Prime Attribute

A **prime attribute** *of a relation is an attribute that is a part of a candidate key of the relation*

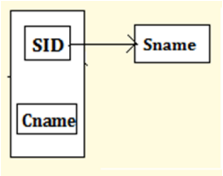
2NF (2)

- **STUDENT(Sid, Sname, Cname)** (already in 1NF)

Students:

SID	Sname	Cname
S1	A	C
S1	A	C++
S2	B	C++
S2	B	DB
S3	A	DB

(SID, Cname): Primary Key



Functional Dependencies:
$\{SID, Cname\} \rightarrow Sname$
$SID \rightarrow Sname$

Partial Dependencies:
$SID \rightarrow Sname$ (as SID is a Proper Subset of Candidate Key $\{SID, Cname\}$)

- Redundancy?
  - Sname
- Anomaly?
  - Yes

Key Normalization

R1:

SID	Sname
S1	A
S2	B
S3	A

{SID}: Primary Key

R2:

SID	Cname
S1	C
S1	C++
S2	C++
S2	DB
S3	DB

{SID,Cname}: Primary Key

The above two relations R1 and R2 are
1.Lossless Join
2.2NF
3.Dependency Preserving

Source: http://www.edugrabs.com/2nf-second-normal-form/

2NF (3): Possible Redundancy

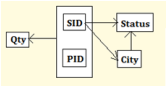
- **Supplier(SID, Status, City, PID, Qty)**

Supplier:

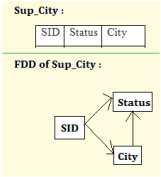
SID	Status	City	PID	Qty
S1	30	Delhi	P1	100
S1	30	Delhi	P2	125
S1	30	Delhi	P3	200
S1	30	Delhi	P4	130
S2	10	Karnal	P1	115
S2	10	Karnal	P2	250
S3	40	Rohtak	P1	245
S4	30	Delhi	P4	300
S4	30	Delhi	P5	315

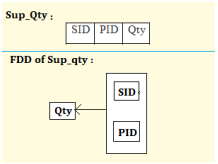
Key : (SID, PID)

Partial Dependencies:
$SID \rightarrow Status$
$SID \rightarrow City$



Post Normalization





Drawbacks:

- **Deletion Anomaly**: If we delete a tuple in *Sup_City*, then we not only loose the information about a supplier, but also loose the status value of a particular city.
- **Insertion Anomaly**: We cannot insert a City and its status until a supplier supplies at least one part.
- **Update Anomaly**: If the status value for a city is changed, then we will face the problem of searching every tuple for that city.

Source: http://www.edugrabs.com/2nf-second-normal-form/

3NF: Third Normal Form

Let $\boldsymbol{R}$ be the relational schema.

- [E. F. Codd,1971] $\boldsymbol{R}$ is in 3NF only if:
  - $\boldsymbol{R}$ should be in 2NF
  - $\boldsymbol{R}$ should not contain transitive dependencies (OR, Every non-prime attribute of R is non-transitively dependent on every key of R)
- [Carlo Zaniolo, 1982] Alternately, $\boldsymbol{R}$ is in 3NF iff for each of its functional dependencies $X \rightarrow A$, at least one of the following conditions holds:
  - X contains A (that is, A is a subset of X, meaning $X \rightarrow A$ is trivial functional dependency), or
  - X is a superkey, or
  - Every element of $A - X$, the set difference between A and X, is a *prime attribute* (i.e., each attribute in $A - X$ is contained in some candidate key)
- [Simple Statement] A relational schema $\boldsymbol{R}$ is in 3NF if for every FD $\boldsymbol{X \rightarrow A}$ associated with R either
  - $\boldsymbol{A \subseteq X}$ (that is, the FD is trivial) or
  - $\boldsymbol{X}$ is a superkey of $\boldsymbol{R}$ or
  - $\boldsymbol{A}$ is part of some candidate key (not just superkey!)
- **A relation in 3NF is naturally in 2NF**

3NF (2): Transitive Dependency

- A **transitive dependency** is a functional dependency which holds by virtue of transitivity. A transitive dependency can occur only in a relation that has three or more attributes.
- Let A, B, and C designate three distinct attributes (or distinct collections of attributes) in the relation. Suppose all three of the following conditions hold:
  - $A \rightarrow B$
  - It is not the case that $B \rightarrow A$
  - $B \rightarrow C$
- Then the functional dependency $A \rightarrow C$ (which follows from 1 and 3 by the axiom of transitivity) is a transitive dependency

3NF (3): Transitive Dependency

- Example of **transitive dependency**
- The functional dependency $\{Book\} \rightarrow \{Author\ Nationality\}$ applies; that is, if we know the book, we know the author's nationality. Furthermore:
  - $\{Book\} \rightarrow \{Author\}$
  - $\{Author\}$ does not $\rightarrow \{Book\}$
  - $\{Author\} \rightarrow \{Author\ Nationality\}$
- Therefore $\{Book\} \rightarrow \{Author\ Nationality\}$ is a transitive dependency.
- Transitive dependency occurred because a non-key attribute (Author) was determining another non-key attribute (Author Nationality).

Book	Genre	Author	Author Nationality
Twenty Thousand Leagues Under the Sea	Science Fiction	Jules Verne	French
Journey to the Center of the Earth	Science Fiction	Jules Verne	French
Leaves of Grass	Poetry	Walt Whitman	American
Anna Karenina	Literary Fiction	Leo Tolstoy	Russian
A Confession	Religious Autobiography	Leo Tolstoy	Russian

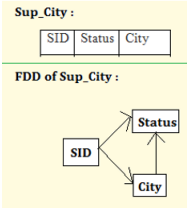
3NF (4): Example

- Example:
  Sup_City(SID, Status, City) (already in 2NF)

Sup_City:

SID	Status	City
S1	30	Delhi
S2	10	Karnal
S3	40	Rohtak
S4	30	Delhi

SID: Primary Key



- Redundancy?
  - Status
- Anomaly?
  - Yes

Functional Dependencies:
SID → Status,
SID → City,
City→ Status
Transitive Dependency :
SID → Status
{As SID → City and City → Status}

Post Normalization

SC:

SID	City
S1	Delhi
S2	Karnal
S3	Rohtak
S4	Delhi

SID: Primary Key

CS:

City	Status
Delhi	30
Karnal	10
Rohtak	40

City: Primary Key

The above two relations SC and CS are

- Lossless Join
- 3NF
- Dependency Preserving

3NF (5): Example

- Relation ***dept_advisor***(***s_ID***, ***i_ID***, ***dept_name***)
- $F = \{$***s_ID***, ***dept_name*** $\rightarrow$ ***i_ID***, ***i_ID*** $\rightarrow$ ***dept_name***$\}$
- Two candidate keys: ***s_ID***, ***dept_name***, and ***i_ID***, ***s_ID***
- R is in 3NF
  - ***s_ID***, ***dept_name*** $\rightarrow$ ***i_ID***
    - ***s_ID***, ***dept_name*** is a superkey
  - ***i_ID*** $\rightarrow$ ***dept_name***
    - ***dept_name*** is contained in a candidate key

A relational schema ***R*** is in 3NF if for every FD ***X*** $\rightarrow$ ***A*** associated with ***R*** either

- ***A*** $\subseteq$ ***X*** (i.e., the FD is trivial) or
- ***X*** is a superkey of ***R*** or
- ***A*** is part of some key (not just superkey!)

3NF (6): Redundancy

- **There is some redundancy in this schema**
- Example of problems due to redundancy in 3NF ($J : s\_ID, L : i\_ID, K : dept\_name$)
  - $R = (J, L, K)$. $F = \{JK \rightarrow L, L \rightarrow K\}$

J	L	K
j_1	l_1	k_1
j_2	l_1	k_1
j_3	l_1	k_1
null	l_2	k_2

- Repetition of information (for example, the relationship l_1, k_1)
  - (i_ID, $dept\_name$)
- Need to use null values (for example, to represent the relationship l_2, k_2 where there is no corresponding value for J).
  - (i_ID, $dept\_name$) if there is no separate relation mapping instructors to departments

Module Summary

- Studied the Normal Forms and their Importance in Relational Design – how progressive increase of constraints can minimize redundancy in a schema

Slides used in this presentation are borrowed from http://db-book.com/ with kind permission of the authors.
Edited and new slides are marked with "PPD".