Topic 13:

Data Normalization

"At the time, Nixon was normalizing relations with China. I figured that if he could normalize relations, then so could I."

— E.F. Codd

Motivating Data Normalization

Some key goals of attribute placement within relations:

Related Concerns:

Update Anomalies (1 / 4)

Recall this relation from Functional Dependencies:

StudentDept

Id	Name	DAbbrev	DName	DOffice
2	Phil	CSc	Computer Science	G-S 917
4	Lisa	CSc	Computer Science	G-S 917
5	Steve	Math	Mathematics	Math 108
13	Bob	CSc	Computer Science	G-S 917
14	Pat	Math	Mathematics	Math 108

Problem:

Solution:

Update Anomalies (2 / 4)

One Potential Split:

Student_Place

Name	DOffice
Phil	G-S 917
Lisa	G-S 917
Steve	Math 108
Bob	G-S 917
Pat	Math 108

Student_Department

Id	DAbbrev	DName	DOffice
2	CSc	Computer Science	G-S 917
4	CSc	Computer Science	G-S 917
5	Math	Mathematics	Math 108
13	CSc	Computer Science	G-S 917
14	Math	Mathematics	Math 108

First Question:

Update Anomalies (3 / 4)

The schema from the last slide:

Student_Place

Name	DOffice

Student_Department

Id	DAbbrev	DName	DOffice

Second Question:

Update Anomalies (4 / 4)

Let's try a different split:

Student

Id	Name	Major_Dept
2	Phil	CSc
4	Lisa	CSc
5	Steve	Math
13	Bob	CSc
14	Pat	Math

Department

Abbreviation	Name	Office
CSc	Computer Science	G-S 917
Math	Mathematics	Math 108

Review of Functional Dependencies

Recall:

Definition: Functional Determination

The set of attributes X functionally determines the set of attributes Y (denoted $X \rightarrow Y$) iff whenever any two tuples of the relation agree on their X values, they must also agree on their Y values.

Normal Forms

The justification for splitting relations is known as . . .

Definition: Normalization

1^{st} Normal Form (1 / 5)

Definition: First Normal Form

. . . .

Example(s): Consider this relation:

Employee

EmpID	Name	Children

1^{st} Normal Form (2 / 5)

Attempt #1: Let's try to achieve 1NF by *flattening* the relation:

Employee2

EmpID	Name	Child
415	Joe	Joe Jr.
415	Joe	Sally
415	Joe	Peter
667	Rhonda	Jim Bob
667	Rhonda	Bobby Ray

See any problems with this relation?

1^{st} Normal Form (3 / 5)

Attempt #2: Separate the Employee and Child information:

Employee3

EmpID	EmpName
415	Joe
667	Rhonda

Child

EmpID	ChildName
415	Joe Jr.
415	Sally
415	Peter
667	Jim Bob
667	Bobby Ray

The Good:

The Bad:

1^{st} Normal Form (4 / 5)

Attempt #3: Give each child a unique identifier:

Employee3

EmpID	EmpName
415	Joe
667	Rhonda

Child2

ChildID	EmpID	ChildName
2	415	Joe Jr.
3	415	Sally
1	415	Peter
5	667	Jim Bob
4	667	Bobby Ray

1^{st} Normal Form (5 / 5)

Notes:

- Clearly, using names as PKs isn't a good idea!
- By a strict interpretation of the relational model's definition, true relations can't have set–valued attributes (thus making 1NF relations a given)
- However, set–valued attributes are commonly permitted in DBMSes (because they are practical)

2^{nd} Normal Form (1 / 5)

Time to talk about grouping the attributes . . . with FDs!

Definition: Full Functional Dependency

. . . .

. . . .

Definition: Prime Attribute

. . . .

2^{nd} Normal Form (2 / 5)

Definition: Second Normal Form (2NF), 1 of 2

. . . .

Example(s):

Consider this schema:

First

<u>**S#**</u>	<u>**P#**</u>	**City**	**Status**	**Qty**

2^{nd} Normal Form (3 / 5)

Example(s): (Continued)

Now consider these FDs in First:

$$S\# \rightarrow \text{City} \qquad \{S\#, P\#\} \rightarrow \text{Qty}$$
$$S\# \rightarrow \text{Status} \qquad \text{City} \rightarrow \text{Status}$$

Given these FDs, is First in 2NF?

2^{nd} Normal Form (4 / 5)

How can we decompose First into multiple 2NF relations?

2^{nd} Normal Form (5 / 5)

An alternate (and more confusing!) 2NF definition:

Definition: Second Normal Form (2NF), 2 of 2

A relation R is in 2NF if, for all FDs in R of the form $X \rightarrow A$ where A is a single non–prime attribute *not contained in* X, X is *not contained in* a CK of R.

3^{rd} Normal Form (1 / 5)

Even with 2NF, we can still have redundancy:

Example(s): Consider F2 again, but with data:

F2

<u>S#</u>	City	Status
S1	London	20
S2	Paris	10
S3	Paris	10
S4	London	20

Definition: Trivial Functional Dependency (rem. T12/Reflexivity?)

An FD $X \rightarrow A$ is a <u>trivial</u> FD when $A \subseteq X$.

Example: $\{S\#, P\#\} \rightarrow S\#$ is an FD, but is trivial.

3^{rd} Normal Form (2 / 5)

Definition: Superkey

Example(s):

Definition: Third Normal Form (3NF), 1 of 2

3^{rd} Normal Form (3 / 5)

3NF catches the problem with F2:

F2

S#	City	Status
S1	London	20
S2	Paris	10
S3	Paris	10
S4	London	20

FDs: S# $\longrightarrow$ City, S# $\longrightarrow$ Status, and City $\longrightarrow$ Status.

3^{rd} Normal Form (4 / 5)

We solve this problem with decomposition:

3^{rd} Normal Form (5 / 5)

Here's alternate (and not as useful) 3NF definition.

Definition: Third Normal Form (3NF), 2 of 2

A relation R is in 3NF if R is in 2NF and every non–prime attribute of R is non-transitively dependent on every CK of R.

Boyce-Codd Normal Form (BCNF) (1 / 3)

Definition: Review: 3NF (version 1)

A relation R is in 3NF if, for every non–trivial FD $X \rightarrow A$ that holds in R, either (a) X is a superkey of R, or (b) A is a prime attribute of R.

If we drop (b), the definition becomes more restrictive.

Definition: Boyce-Codd Normal Form

BCNF (2 / 3)

Example(s):

Consider the schema R(m,n,o,p) with these FDs:

$\{m,n\} \rightarrow o$, $\{m,n\} \rightarrow p$, and $p \rightarrow n$.

Is this schema in 3NF?

Is this schema in BCNF?

BCNF (3 / 3)

Notes:

Summary of FD-Based Normalization

- Create an initial relational design
- Identify the FDs
- Construct a decomposed schema for which:
    - Natural joins do not add spurious tuples (a.k.a. The Non-Additive (or Lossless) Join Property)
    - All relations are in at least 3NF
    - All FDs are retained or can be reconstructed

Beyond Functional Dependencies

You're kidding — there are *more* normal forms?

Motivating Example (1 / 4)

Consider this schema:

Bookstore

Course	Professor	Text
Programming	{Jones,Smith}	{The Java Coloring Book, Java for the Imbecilic}
Data Structures	{Jones}	{The Java Coloring Book, Boxes + Arrows = Linked Lists, Why Trees Grow Down},

Motivating Example (2 / 4)

To achieve 1NF, we need to ‘flatten’ the relation:

Bookstore2

Course	Professor	Text
Programming	Jones	The Java Coloring Book
Programming	Jones	Java for the Imbecilic
Programming	Smith	The Java Coloring Book
Programming	Smith	Java for the Imbecilic
Data Structures	Jones	The Java Coloring Book
Data Structures	Jones	Boxes + Arrows = Linked Lists
Data Structures	Jones	Why Trees Grow Down

Motivating Example (3 / 4)

Observations about Bookstore2:

Problems with Bookstore2:

Motivating Example (4 / 4)

Let's try separating teaching from texts:

Teaches

Course	**Professor**
Programming	Jones
Programming	Smith
Data Structures	Jones

Requires

Course	**Text**
Programming	The Java Coloring Book
Programming	Java for the Imbecilic
Data Structures	The Java Coloring Book
Data Structures	Boxes + Arrows = Linked Lists
Data Structures	Why do Trees Grow Down?

Multivalued Dependencies (1 / 7)

We need a new type of dependency!

Definition: Multivalued Dependency (MVD)

Let A be the set of attributes of relation R, with $X \subseteq A$ and $Y \subseteq A$.
If two tuples $s, t \in R$ have matching X values, then the MVD $X \twoheadrightarrow Y$ exists in R when tuples u and w also exist in R such that ...

Multivalued Dependencies (2 / 7)

Let's apply the definition to a new example.

Example(s):

If Student $\twoheadrightarrow$ Class, which additional tuples must exist?

$A = \{ X, \quad Y, \quad Z \}$

	Student	Class	TA
s	Art	244	Kay
t	Art	337	Lee
u			
w			

The Definition's Conditions:

(a) all four tuples have matching X values,
(b) s and u have matching Y values,
(c) t and w have matching Y values,
(d) s and w have matching $A - Y$ values, and
(e) t and u have matching $A - Y$ values.

Multivalued Dependencies (3 / 7)

Does Bookstore2 contain an MVD?

Example(s):

Can we identify s, t, u, and w for ‘Programming’?

Bookstore2 (partial)

Course	Professor	Text
Programming	Jones	The Java Coloring Book
Programming	Jones	Java for the Imbecilic
Programming	Smith	The Java Coloring Book
Programming	Smith	Java for the Imbecilic

Multivalued Dependencies (4 / 7)

Does Bookstore2 meet the MVD definition? (Continued!)

Example(s):

But what about the ‘Data Structures’ tuples?

Bookstore2 (partial)

Course	Professor	Text
Data Structures	Jones	The Java Coloring Book
Data Structures	Jones	Boxes + Arrows = Linked Lists
Data Structures	Jones	Why Trees Grow Down

Multivalued Dependencies (5 / 7)

If $X \twoheadrightarrow Y$ comes from $Y \times Z$, does $X \twoheadrightarrow Z$ also hold?

Multivalued Dependencies (6 / 7)

This pairing requirement leads to another, quite different, definition of MVDs:

Definition: Multidependency

Let R be a relational schema, and let X, Y, and Z be subsets of R's attributes. Y is multidependent on X (or X multidetermines Y), denoted $X \twoheadrightarrow Y$, iff the set of Y values matching a given (X, Z) pair of values depends only on the X set and is independent of the Z set.

Multivalued Dependencies (7 / 7)

Summary:

- MVDs explain a more general kind of redundancy than do FDs.
- Like FDs, MVDs should be designed into a relational schema.
- Neither 1NF, 2NF, 3NF, nor BCNF covers MVDs.

Notes:

4^{th} Normal Form (1 / 3)

Remember trivial FDs?

Definition: Trivial MVDs

. . . .

At last, a normal form that addresses MVDs:

Definition: Fourth Normal Form (4NF)

. . . .

4^{th} Normal Form (2 / 3)

Is our decomposition of Bookstore2 in 4NF?

Teaches

Course	Professor
Programming	Jones
Programming	Smith
Data Structures	Jones

Requires

Course	Text
Programming	The Java Coloring Book
Programming	Java for the Imbecilic
Data Structures	The Java Coloring Book
Data Structures	Boxes + Arrows = Linked Lists
Data Structures	Why do Trees Grow Down?

(Recall: The MVDs are Course $\twoheadrightarrow$ Professor and Course $\twoheadrightarrow$ Text)

4^{th} Normal Form (3 / 3)

Notes:

- How often do non–4NF relations occur?

  A study of 40 schemas found 23% had ≥ 1 non-4NF relation(s).

  (Meaning: Stopping at 3NF/BCNF might be stopping too soon!)

- Three time-saving theorems:
  - All non-4NF relations can be decomposed into two or more 4NF relations.
  - All 4NF relations are also BCNF relations.
  - A BCNF relation whose candidate keys are all single attributes is also a 4NF relation.

5^{th} Normal Form (aka Projection-Join N.F.)

Consider (most of) SPJ:

SPJ

S#	P#	J#

- Because of the redundancies within the three pairs (S# – P#, S# – J#, and P# – J#), 5NF says that SPJ should be divided into three relations (SP, SJ, and PJ).
- Practically, doing so isn't efficient (much re-joining is required).
- And there are several other normal forms for even more esoteric problems! (E.g., 6NF, DKNF, Unnormalized Form, . . .)

Normal Form Venn Diagram