

---

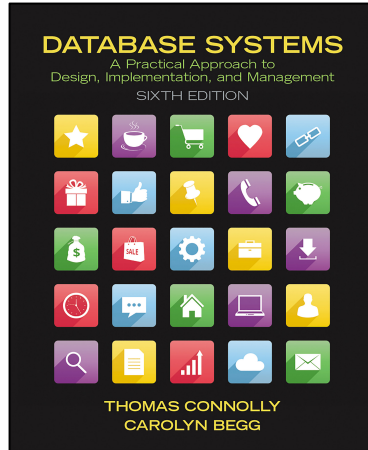
# Normal Form

---

Topic 2 Lesson 8  
Removing redundancy from a data schema

# Chapter 14 14.5-14.9 Connolly and Begg

---



# Normalization process

---

Formal technique for analysing a relation based on its primary key and the functional dependencies between the attributes of that relation.

We use the normalization process as a validation technique for the defined relations.

It is a crucial step in the **logical database design process**.

# Normalization

---

Normalization identifies redundancy that leads to functional dependency.

The normalization process is a series of tests that help identify the optimal grouping of attributes to relations.

GOAL: reduce data redundancy

# Normal form addresses dependencies

---

- GOAL: Free the collection of relations from undesirable insertion, modification and deletion **dependencies**
  - If schema has duplicated data in multiple rows
    - Forced to update/delete all copies of a piece of data
    - How do you know you got all copies of it?
- Address the flaws in the current design

# Normal form leads to a cleaner schema

---

- Reduce the need for restructuring the collection of relations
  - Build an extensible design now as opposed to later
- Make the relational model more informative to users
  - Cleaner model should be easier to understand
- Make the collection of relations neutral to the query statistics
  - Designed for general purpose querying

# Unnormalized form

---

- No primary key or NULL values in the primary key fields
- A table that contains an attribute with one or more repeating groups or contains a set
  - **A repeating group** is a set of logically related fields or values that occur multiple times in one record.
- Attributes need not be atomic

# Unnormalized form

Table to track  
birth mother to  
child

Set of children  
to a mother

One field to  
represent all  
children

UNNORMALIZED FORM (UNF) – DUPLICATES ENTITIES

Mother Id	Mother Name	Children
1	Elsa	Alex
1	Elsa	Mary Alice Tom Lou
2	Golda	George Fred
3	Viola	Ava
4	Iris	Kayla
5	Daisy	Harry



# Does this solve the problem?

No, but it does not stop people from designing databases this way

Still unnormalized

Mother Id	Mother Name	Child1	Child2	Child3	Child4
1	Elsa	Alex	NULL	NULL	NULL
1	Elsa	Mary	Alice	Tom	Lou
2	Golda	George	Fred	NULL	NULL
3	Viola	Ava	NULL	NULL	NULL
4	Iris	Kayla	NULL	NULL	NULL
5	Daisy	Harry	NULL	NULL	NULL

# First normal form

---

- Tuples in a relation must contain the same number of fields
- The domain of each attribute contains atomic values
- The value of each attribute contains only a single value
  - No attributes are sets or a repeating group.

Relational  
Model

1st normal  
form

# UNF to 1NF

---

- Nominate an attribute or group of attributes to act as the key for the unnormalized table.
- Identify the repeating group(s) in the unnormalized table which repeats for the key attribute(s).
  - Remove the set by creating a separate table for the set or if there is an upper limit to the set you can flatten it into fields

# 1st Normal Form

Mother Id	Mother Name
1	Elsa
2	Golda
3	Viola
4	Iris
5	Daisy

Decompose table, remove repeating attributes

Child Id	Name	Mother
11	Mary	1
12	Alice	1
13	George	2
14	Fred	2
15	Ava	3
16	Kayla	4
17	Harry	5
18	Alex	1
19	Tom	1
20	Lou	1

# Second normal form

---

- Requirement for tables that have a composite key
- Table must already be in first normal form
- Every non-primary key attribute is fully functionally dependent on the (entire) primary key
- A table in first normal form and having a primary key with only one field is also in 2nd normal form

# Second normal form

---

Based on the concept of full functional dependency.

Full functional dependency indicates that if

A and B are attributes of a relation,

B is fully dependent on A if B is functionally dependent on A but not on any proper subset of A.

# 1NF to 2NF

---

Identify the primary key for the 1NF relation.

Identify the functional dependencies in the relation.

If partial dependencies exist on the primary key  
remove them by placing them in a new relation  
along with a copy of their determinant.

# Example 2NF with a composite key

1<sup>st</sup> Normal Form but NOT 2nd NORMAL FORM

<u>Mother Id</u>	First Name	Last Name	<u>Hospital</u>	Hospital Address
1	Elsa	General	BIDMC	Boston
2	Golda	Major	MGH	Boston
3	Viola	Funt	TMC	Cambridge
4	Iris	Batter	1	1
5	Daisy	Mae	4	4

2<sup>nd</sup> NORMAL FORM

<u>Hospital ID</u>	Hospital	Hospital Address
1	BIDMC	Boston
2	MGH	Boston
3	TMC	Cambridge
4	Mayo	Allston

2<sup>nd</sup> NORMAL FORM

<u>Mother Id</u>	First Name	Last Name	Hospital Id
1	Elsa	General	1
2	Golda	Major	2
3	Viola	Funt	3
4	Iris	Batter	1
5	Daisy	Mae	4



# Third normal form

---

- Table is in first and second normal form
- No dependencies between 2 non-key attributes
- No non-primary-key attribute is transitively dependent on the primary key
- Solution: decompose the table so that the offending attribute is in a separate table
- Attribute is fully functionally dependent on the primary key

# Third normal form

---

Based on the concept of transitive dependency.

Transitive Dependency is a condition where

A, B and C are attributes of a relation such that if  $A \rightarrow B$   
and  $B \rightarrow C$ ,

then C is transitively dependent on A through B. (Provided that A is not functionally dependent on B or C).

# 2NF to 3NF

---

- Identify the primary key in the 2NF relation.
- Identify functional dependencies in the relation.
- If transitive dependencies exist on the primary key remove them by placing them in a new relation along with a copy of their dominant.

# Example: to 3rd normal form

## 2nd NORMAL FORM

<u>Mother Id</u>	First Name	Last Name	Hospital Id	Room	State
1	Elsa	General	1	36	MA

## 3rd Normal Form

<u>Register Id</u>	Hospital Id	Room Id
1	1	36
2	2	48
3	3	38
4	1	41
5	4	32

<u>Register Id</u>	Hospital Id	Room Id
3	48	MA
3	3	
4	<u>Hospital ID</u>	Hospital
3	1	BIDMC
	2	MGH
	3	TMC
	4	Mayo

## 3rd NORMAL FORM

<u>Mother Id</u>	First Name	Last Name	<u>Register Id</u>
1	Elsa	General	1
2	Golda	Major	2
3	Viola	Funt	3
		Batter	4
		Mae	5

## Bill Kent's quote:

---

Every non-key attribute must provide a fact :  
about the key,  
the whole key  
and nothing but the key

# Summary

---

To remove unnecessary redundancy, a table needs to be decomposed and the redundancy should be broken out into a separate table.