Database Foundations

3-9
Validating Data Using Normalization
Roadmap

Conceptual and Physical Data Models

Business Rules

Entities

Attributes

Unique Identifiers

Relationships

Validating Relationships

Tracking Data Changes over Time

Validating Data Using Normalization

You are here
Objectives

This lesson covers the following objective:

• Use normalization to validate data
Why Should You Normalize Data?

• Reduce redundant data in the existing design
• Increase the integrity of data and the design’s stability
• Eliminate other types of data inconsistencies and anomalies
• Identify missing tables, columns, and constraints
What Is Normalization?

• Normalization is a relational database concept, but its principles apply to data modeling.

• Rules:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Normal Form (1NF)</td>
<td>All attributes must be single-valued.</td>
</tr>
<tr>
<td>Second Normal Form (2NF)</td>
<td>An attribute must be dependent on its entity's entire UID.</td>
</tr>
<tr>
<td>Third Normal Form (3NF)</td>
<td>No non-UID attributes can be dependent on another non-UID attribute.</td>
</tr>
</tbody>
</table>
Types of Normal Forms

Main types of normal forms introduced by E.F. Codd:

• First Normal Form (1NF)
• Second Normal Form (2NF)
• Third Normal Form (3NF)
• Boyce Codd Normal Form (BCNF)
First Normal Form (1NF)

All attributes must be single-valued.

The Date Contacted and Contact Details attributes have multiple values. Therefore, this is not in 1NF.

Create an additional entity, CONTACT, with a 1:M relationship to CLIENT.
Second Normal Form (2NF)

An attribute must be dependent on its entity's entire UID.

The Bank Location attribute is dependent on BANK, not on ACCOUNT. Therefore, this is not in 2NF. Move the attribute to the BANK entity.
Third Normal Form (3NF)

Each attribute depends only on the UID of its entity.

The Quantity and Price attributes are dependent on the Order ID (UID) and Line Item ID (non-UID). Therefore, this is not in 3NF.

Create a new ORDER ITEM entity. Move the Line Item ID, Quantity, and Price attributes to the new entity, and then create an identifying relationship.
Boyece Codd Normal Form

• A table is in Boyce-Codd Normal Form (BCNF) if it is in 3NF and if every determinant is a candidate key.

• A candidate key can be defined as an attribute or a combination of attributes that can be uniquely used to identify a record in a table.
First Normal Form (1 NF): Example 1

Un-normalized Information pertaining to School Building and classrooms located in the building

<table>
<thead>
<tr>
<th>School Building Code</th>
<th>Building Name</th>
<th>Address</th>
<th>Floor Number</th>
<th>Number of Classrooms Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

School Building Code uniquely identifies a row and is called the primary key of the table.
Second Normal Form (2 NF): Example 1

<table>
<thead>
<tr>
<th>School Building Code</th>
<th>Building Name</th>
<th>Address</th>
<th>Floor Number</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table holds information about the school building and the departments.

2 NF

<table>
<thead>
<tr>
<th>School Building Code</th>
<th>Building Name</th>
<th>Address</th>
<th>Department Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data pertaining to the school building and department has been split into two tables.
## Third Normal Form (3 NF): Example 1

<table>
<thead>
<tr>
<th>School Building Code</th>
<th>Building Name</th>
<th>Address</th>
<th>Department Code</th>
<th>Department Code</th>
<th>Department Name</th>
<th>Floor Code</th>
<th>Floor Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3 NF

The transitive dependencies between the tables are removed.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.
### Normalization Example 2: Unnormalized Data

**Ordered by:**
- **Customer ID:**
- **Customer Name:**
- **Address Line 1:**
- **Address Line 2:**
- **Address Line 3:**
- **City, State ZIP:**

**Ship to:**
- **Ship Via:**
- **Name:**
- **Address Line 1:**
- **Address Line 2:**
- **Address Line 3:**
- **City, State ZIP:**

<table>
<thead>
<tr>
<th>Item ID</th>
<th>Color</th>
<th>Size</th>
<th>Quantity</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Order Total:**

---

Validating Data Using Normalization
Normalization Example 2: Transforming to First Normal Form

ORDER

# Order ID
  o Order Date
  o Order Total
  o Customer ID
  o Customer Name
  o Customer Address1
  o Customer Address2
  o Customer Address3
  o Customer City
  o Customer State
  o Customer Zip
  o Ship Via
  o Ship To Name
  o Ship To Address1
  o Ship To Address2
  o Ship To Address3
  o Ship To City
  o Ship To State
  o Ship To Zip
  o Item ID
  o Color
  o Size
  o Quantity
  o Description
  o Price
Normalization Example 2: Transforming to First Normal Form

```
ORDER
# * Order ID
0 Order Date
0 Order Total
0 Customer ID
0 Customer Name
0 Customer Address1
0 Customer Address2
0 Customer Address3
0 Customer City
0 Customer State
0 Customer Zip
0 Ship Via
0 Ship To Name
0 Ship To Address1
0 Ship To Address2
0 Ship To Address3
0 Ship To City
0 Ship To State
0 Ship To Zip

LINEITEM
# * Item ID
# * Color
# * Size
0 Quantity
0 Description
0 Price
```
Normalization Example 2: Transforming to Second Normal Form

[Diagram of database schema showing ORDER, LINE ITEM, and ITEM tables with attributes]
Normalization Example 2: Transforming to Third Normal Form
Summary

In this lesson, you should have learned how to use normalization to validate data.