
B

Normalization

Introduction

Lesson aim

This lesson describes the steps involved in order to normalize table data to the third normal form for cases when there is no possibility of performing a full data analysis.

Overview

- Table Normalization
- Normal Forms of Tables

B-2

Topic	See Page
Introduction	2
Normalization and its Benefits	3
First Normal Form	7
Second Normal Form	9
Third Normal Form	11
Summary	13

Objectives

At the end of this lesson, you should be able to do the following:

- Define normalization and explain its benefits
- Place tables in Third Normal Form

Normalization and its Benefits

Why and When to Normalize Tables

Before we even talk about why you should normalize, first consider when you should normalize. If you are developing an application and use the techniques of entity relationship (ER) modeling, then you will not need to normalize. One of the advantages of entity relationship modeling is that the resulting table design is already normalized, provided there are no obvious errors in the ER model.

The only time you will need to normalize the data is if there has been no time to build an entity model and when a set of tables is already available. You can then employ the normalization techniques following the initial database design as a *last chance* to check for existing database integrity.

History of Normalization

Normalization is a technique established by the originator of the relational model, E.F. Codd. The complete set of normalization techniques, include twelve rules that databases need to follow in order to be described as truly normalized. It is a technique that was created in support of relational theory, years before entity relationship modeling was developed. The entity relationship modeling process has incorporated many of the normalization techniques to produce a normalized entity relationship diagram.

Two terms that have their origins in the normalization technique are still widely in use. One is *normalized data*, the other is *denormalization*.

Objective of Normalization

The major objective of normalization is to remove redundant data from an existing set of tables or table definitions, thereby increasing the integrity of the database design and to maximize flexibility of data storage. Removing redundant data helps to eliminate update anomalies. The first three normal forms progress in a systematic manner to achieve this objective.

There are many other normal forms in addition to the first three, and they deal with more subtle anomalies. In general, the IT industry considers normalization to the Third form an acceptable level to remove redundancy. With a few exceptions, higher normalization levels are not widely used.

The major subject of normalization is tables, not entities.

Why Normalize?

- **An Entity Model is not always available as a starting point for design**
- **To reduce redundant data in existing design**
- **To increase integrity of data, and stability of design**
- **To identify missing tables, columns and constraints**

Note: Third normal form is the generally-accepted goal for a database design that eliminates redundancy.

B-3

Normalization Compared to Normalized Data

Normalized data is data that contains no redundancies. This is important as data redundancy may cause integrity problems. Normalization is the activity, the process, that leads to a normalized data structure as does entity relationship modeling.

Benefits of Normalized Data

The major benefits of a correctly normalized database from an Information Systems perspective include:

- Refinement of the strategy for constructing tables and selecting keys.
- Improved communication with the end-users' application activities.
- Reduced problems associated with inserting and deleting data.
- Reduced enhancement and modification time associated with changing the data structure.
- Improved information for decisions relating to the physical database design.
- Identification of potential problems that may have been overlooked during analysis.

Recognize Unnormalized Data

USER _ID	USER _NAME	MSE _ID	REC _DATE	SUBJECT	TEXT	SRVR _ID	SERVER _NAME
2301	Smith	54101	05/07	Meeting Today	There is..	3786	IMAP05
2301	Smith	54098	07/12	Promotions	I like to.	3786	IMAP05
2301	Smith	54445	10/06	Next Assignment	Your next.	3786	IMAP05
5607	Jones	54101	05/07	Meeting Today	There is..	6001	IMAP08
5607	Jones	54512	06/07	Lunch?	Can you...	6001	IMAP08
5607	Jones	54660	12/01	Jogging Today?	Can you...	6001	IMAP08
7773	Walsh	54101	05/07	Meeting Today	There is..	9988	EMEA01
7773	Walsh	54554	03/17	Stock Quote	The latest	9988	EMEA01
0022	Patel	54101	05/07	Meeting Today	There is..	2201	EMEA09
0022	Patel	54512	06/07	Lunch?	Can we ...	2201	EMEA09

B-4

Unnormalized Data

Data that has not been “normalized” is considered to be “unnormalized” data or data in *zero-normal* form. This data is not to be confused with data that is *denormalized*. If no ER Model was created at the start of a database design project, you are likely to have unnormalized data, not denormalized data. If you want to add redundancy, for faster performance or other reasons, you follow the rules defined during the process of denormalization. But, to denormalize data you must start with normalized data. You cannot denormalize an unnormalized design, just as you cannot de-ice your car, if there is no ice on it.

Normalization

Normalization consists of a series of rules that must be applied to move from a supposedly unnormalized set of data to a normalized structure. The process is described in various steps which lead to a “higher” level of normalization. These levels are called normal forms.

Normalization Rules	
Normal Form Rule	Description
First Normal Form (1NF)	The table must express a set of unordered, two-dimensional tables. The table cannot contain repeating groups.
Second Normal Form (2NF)	The table must be in 1NF. Every non-key column must be dependent on all parts of the primary key.
Third Normal Form (3NF)	The table must be in 2NF. No non-key column may be functionally dependent on another non-key column.
“Each non-primary key value MUST be dependent on the key, the whole key, and nothing but the key.”	
B-5	

First Normal Form

Definition of First Normal Form (1NF)

The table must express a set of unordered, two-dimensional table structures. A table is considered in the first normal form if it contains no repeating groups.

Steps to Remove Repeating Groups

- 1 Remove the repeating columns from the original unnormalized table.
- 2 Create a new table with the primary key of the base table and the repeating columns.
- 3 Add another appropriate column to the primary key, which ensures uniqueness.
- 4 Create a foreign key in the new table to link back to the original unnormalized table.

Converting to First Normal Form

USER _ID	USER _NAME	MSE _ID	REC_ DATE	SUBJECT	TEXT	SRVR _ID	SERVER _NAME
2301	Smith	54101	05/07	Meeting Today	There is..	3786	IMAP05
2301	Smith	54098	07/12	Promotions	I like to.	3786	IMAP05
2301	Smith	54445	10/06	Next Assignment	Your next.	3786	IMAP05
5607	Jones	54512	06/07	Lunch?	Can you...	6001	IMAP08
5607	Jones	54101	05/07	Meeting Today	There is..	6001	IMAP08
5607	Jones	54660	12/01	Jogging Today?	Can you...	6001	IMAP08
7773	Walsh	54101	05/07	Meeting Today	There is..	9988	EMEA01
7773	Walsh	54554	03/17	Stock Quote	The latest	9988	EMEA01
0022	Patel	54101	05/07	Meeting Today	There is..	9988	EMEA01
0022	Patel	54512	06/07	Lunch?	Can we ...	9988	EMEA01

1. Remove repeating group from the base table.
2. Create a new table with the PK of the base table and the repeating group.

B-6

First Normal Form—Single Record

USERS

USER _ID	USER _NAME	MSE _ID	REC DATE	SUBJECT	TEXT	SRVR _ID	SERVER _NAME
2301	Smith	54101	05/07	Meeting Today	There is..	3786	IMAP05
5607	Jones	54512	06/07	Lunch?	Can you...	6001	IMAP08
7773	Walsh	54101	05/07	Meeting Today	There is..	9988	EMEA01
0022	Patel	54101	05/07	Meeting Today	There is..	9988	EMEA01



USERS

USER _ID	USER _NAME	SRVR _ID	SERVER _NAME
2301	Smith	3786	IMAP05
5607	Jones	6001	IMAP08
7773	Walsh	9988	EMEA01
0022	Patel	9988	EMEA01

B-7

First create a second table to contain the repeating group columns. Then create a primary key composed of the primary key from the unnormalized table and another column that is unique. Finally create a foreign key to link back to the first table.

First Normal Form—Repeating Groups

RECEIVED MESSAGES (1NF)

USER _ID	USER _NAME	MSE _ID	REC DATE	SUBJECT	TEXT
2301	54101	05/07	Meeting Today	There is..	
2301	54098	07/12	Promotions	I like to.	
2301	54445	10/06	Next Assignment	Your next.	
5607	54101	05/07	Meeting Today	There is..	
5607	54512	06/07	Lunch?	Can you...	
5607	54660	12/01	Jogging Today?	Can you...	
7773	54101	05/07	Meeting Today	There is..	
		/17	Stock Quote	The latest	
		/07	Meeting Today	There is..	
		/07	Lunch?	Can we ...	

USER _ID	USER _NAME	SRVR _ID	SERVER _NAME
2301	Smith	3786	IMAP05
5607	Jones	6001	IMAP08
7773	Walsh	9988	EMEA01
0022	Patel	9988	EMEA01

USERS (1NF)

B-8

Second Normal Form

Definition of Second Normal Form (2NF)

A table is in second normal form if the table is in the first normal form and every non-primary key column is functionally dependent upon the entire primary key. No non-primary key column can be functionally dependent on part of the primary key.

Depends on is defined as: a column B depends on column A means that B must be re-evaluated whenever A changes.

A table in the first normal form will be in second normal form if any one of the following apply:

- The primary key is composed of only one column.
- No nonkeyed columns exist in the table.
- Every nonkeyed attribute is dependent on all of the columns contained in the primary key.

Converting to Second Normal Form

1. Determine which non-key columns are not dependent upon the table's entire primary key.
2. Remove those columns from the base table.
3. Create a second table with those columns and the columns from the *PK* that they are dependent upon.

B-9

Steps to Remove Partial Dependencies

- 1 Determine which nonkey columns are dependent upon the table's entire primary key.
- 2 Remove those columns from the base table. Create a second table with those nonkeyed columns and a copy of the columns from the primary key that they are dependent upon.
- 3 Create a foreign key from the original base table to the new table, linking to the new primary key.

Tables Already in Second Normal Form

USERS

USER _ID	USER _NAME	SRVR _ID	SERVER _NAME
2301	Smith	3786	IMAP05
5607	Jones	6001	IMAP08
7773	Walsh	9988	EMEA01
0022	Patel	9988	EMEA01

Is the USERS table already in 2NF?

B-10

Convert to Second Normal Form

RECEIVED_ MESSAGES (1NF)	USER _ID	MSE _ID	REC DATE	SUBJECT	TEXT
	2301	54101	05/07	Meeting Today	There is.
	2301	54098	07/12	Promotions	I like to
	2301	54445	10/06	Next Assignmen	Your next
	5607	54101	05/07	Meeting Today	There is.
	5607	54512	06/07	Lunch?	Can you..
	7773	54660	12/01	Jogging Today?	Can you..
	3	54101	05/07	Meeting Today	There is.
	3	54554	03/17	Stock Quote	The lates
	2	54101	05/07	Meeting Today	There is.
	2	54512	06/07	Lunch?	Can we ..

RECEIVED_ MESSAGES (2NF)	USER _ID	MSE _ID	REC DATE
	2301	54101	05/07
	2301	54098	07/12
	2301	54445	10/06
	5607	54101	05/07
	5607	54512	06/07
	5607	54660	12/01
	7773	54101	05/07
	7773	54554	03/17
	0022	54101	05/07
	0022	54512	06/07

MESSAGES (2NF)	MSE _ID	SUBJECT	TEXT
	54101	Meeting Toda	There is.
	54098	Promotions	I like to
	54445	Next Assignm	Your next
	54512	Lunch?	Can you..
	54660	Jogging Toda	Can you..
	54554	Stock Quote	The lates

B-11

Third Normal Form

Definition of Third Normal Form (3NF)

A table is in third normal form if every nonkeyed column is directly dependent on the primary key, and not dependent on another nonkeyed column. If the table is in second normal form and all of the “transitive dependencies” are removed, then every non-keyed column is said to be “dependent upon the key, the whole key, and nothing but the key”.

Converting to Third Normal Form

Remove any columns that are dependent upon another non-key column:

- 1. Determine which columns are dependent upon another non-key column.**
- 2. Remove those columns from the base table.**
- 3. Create a second table with those columns and the non-key columns that they are dependent upon.**

B-12

Steps to Remove Transitive Dependencies

- 1** Determine which columns are dependent on another non-keyed column.
- 2** Remove those columns from the base table.
- 3** Create a second table with those columns and the non-key columns that they are dependent upon.
- 4** Create a foreign key in the original table linking to the primary key of the new table.

Tables Already in Third Normal Form

No non-key column can be functionally dependent upon another non-key column.

RECEIVED_
MESSAGES
(2NF)

USER _ID	MSE REC _ID DATE	
----	-----	-----
2301	54101	05/07
2301	54098	07/12
2301	54445	10/06
5607	54101	05/07
5607	54512	06/07
5607	54660	12/01
7773	54101	05/07
7773	54554	03/17
0022	54101	05/07
0022	54512	06/07

MESSAGES
(2NF)

ID	SUBJECT	TEXT
----	-----	-----
54101	Meeting Today	There is.
54098	Promotions	I like to
54445	Next Assignmen	Your next
54512	Lunch?	Can you..
54660	Jogging Today?	Can you..
54554	Stock Quote	The lates

Are these two tables in third normal form? Why?

B-13

Converting to Third Normal Form

USERS

USER _ID	USER _NAME	SRVR _ID	SERVER _NAME
----	-----	-----	-----
2301	Smith	3786	IMAP05
5607	Jones	6001	IMAP08
7773	Walsh	9988	EMEA01
0022	Patel	9988	EMEA01

USERS

ID	NAME	SRVR _ID
----	-----	-----
2301	Smith	3786
5607	Jones	6001
7773	Walsh	9988
0022	Patel	9988

MAIL_
SERVER

ID	NAME
----	-----
3786	IMAP05
6001	IMAP08
9988	EMEA01

B-14

The theory of normalization goes further than the third normal form to cater for several problematic constructions that may remain. Those normal forms are outside the scope of this lesson.

Summary

Summary

1NF The table must express a set of unordered, two-dimensional tables. The table cannot contain repeating groups.

2NF The table must be in 1NF. Every non-key column must be dependent on all parts of the primary key.

3NF The table must be in 2NF. No non-key column may be functionally dependent on another non-key column.

An entity relationship model transforms into normalized data design.

B-15

