

Normal Forms

by Sophia



WHAT'S COVERED

This tutorial explores the three main normal forms that are used to normalize a database in two parts:

1. Introduction
2. The Normalization Process

1. Introduction



BEFORE YOU START

Recall that normalization helps database designers focus on evaluating and correcting issues with table structures to help minimize data redundancy. The normalization process is a set of steps to work towards refining the tables that are generated.

Each table typically should represent a single subject. For example, in looking at our database in PostgreSQL, the customer table only contains the data about the customer, the employee table only contains data about the employee, and so forth. In addition, each intersection of a row and column should only contain one value and not multiple values.

⇒ **EXAMPLE** If you look at the example from the movie rating database, see that actor and genre have multiple values:

User	MovieTitle	NumericRating	TextualRating	Actor	Genre	ReleaseDate
Andy Joe	Toy Story	5	Such a classic!	Tom Hanks, Tim Allen	Animation, Adventure, Comedy	1995
Andy Joe	Titanic	4	Great movie!	Leonardo DiCaprio, Kate Winslet, Billy Zane, Kathy Bates	Drama	1997
Sophia McKenzie	Terminator 2: Judgement Day	5	Loved this movie, definitely a must watch!	Arnold Schwarzenegger, Linda Hamilton, Edward Furlong	Action, Science Fiction	1991
James Wang	Titanic	5	Best movie!	Leonardo DiCaprio, Kate Winslet, Billy Zane, Kathy Bates	Romance	1997
Barton Raftor	Titanic	3	It was OK, could be shorter.	Leonardo DiCaprio, Kate Winslet, Billy Zane, Kathy Bates	Drama, Romance	1997
Barton Raftor	Toy Story	4.5	Good movie! Watched a few times.	Tom Hanks, Tim Allen, Don Rickles, Jim Varney, Annie Potts, John Morris	Animation, Adventure, Comedy	1995

You also want to ensure that there is no data item that is unnecessarily stored in more than one table, to help minimize redundancy. This way, if data needs to be added, updated, or deleted, it only needs to be done in one place. All of the attributes in a table should be dependent on the primary key so that the data can be uniquely identifiable by the primary key values. Each of the tables should have no insertion, update, or deletion anomalies so that you can ensure that the data is consistent.

2. The Normalization Process

To help reach these objectives, database designers go through the normalization process one stage at a time. For the **first normal form (1NF)**, you ensure that each column in our tables is single-valued, meaning it does not contain multiple values. By doing so, you eliminate any repeating groups of columns. You also ensure that each of the attributes or columns in a table has a unique name. The first normal form (1NF) is also when you identify the primary key of each table, with the attributes being dependent on the primary key.



KEY CONCEPT

A common-sense rule with the first normal form (1NF) is to ensure that each of the column values are of the same data type and purpose. This means that if you set a phone number column in the customer table, you should only store the phone number, rather than putting an address in the field.

For the **second normal form (2NF)**, our tables should be in the first normal form (1NF) to begin, but you also ensure there is no partial dependency. It can help to look at a table like the employee table to understand dependency:

employee	
postal_code	VARCHAR (10)
employee_id	INTEGER
last_name	VARCHAR (20)
first_name	VARCHAR (20)
title	VARCHAR (30)
reports_to	INTEGER
birth_date	TIMESTAMP
hire_date	TIMESTAMP
address	VARCHAR (70)
city	VARCHAR (40)
state	VARCHAR (40)
country	VARCHAR (40)
phone	VARCHAR (24)
fax	VARCHAR (24)
email	VARCHAR (60)

The primary key in this table is the employee_id. This allows us to uniquely identify each row, even if you happen to have two employees with the same name. The employee_id is unique for every single row, and using it, you can reference a specific record.

⇒ **EXAMPLE** For example, if you need to find out the address of a specific employee, you can use the employee_id to reference it. That is also the case for every other column in this table. This means that every column depends on the primary key, and its respective data can be fetched by using it.

Now that you have an understanding of dependency, you can now define what partial dependency is. Partial dependency can occur when an attribute in a table depends on only a part of the primary key, and not the whole key. This can be the case even if the primary key is uniquely generated as you have to consider candidate keys. If this occurs, you need to remove the attribute that is causing partial dependency and move it to another table where it is fully dependent on the primary key.

For the **third normal form (3NF)**, the tables should be in the second normal form (2NF), and additionally have no transitive dependencies. A transitive dependency exists when there are functional dependencies.

⇒ **EXAMPLE** For example, you may have a column X dependent on the primary key of a table, but then you have column Y being dependent on column X. With this, you have a transitive dependency, with column Y being dependent on the primary key but indirectly. The solution ends up being quite simple: remove the columns that are creating a transitive dependency into another table.

You will learn about higher normal forms in a later tutorial, such as 4NF, in which there are no independent multivalued dependencies. But remember that the third normal form (3NF) is typically where the normalization process ends.



TERMS TO KNOW

First Normal Form (1NF)

The first normal form (1NF) ensures that each column in the tables is single-valued, meaning it does not contain multiple values.

Second Normal Form (2NF)

In the second normal form (2NF), tables should be in the first normal form (1NF) to begin, but you also ensure there is no partial dependency.

Third Normal Form (3NF)

In the third normal form (3NF), tables should be in the second normal form (2NF), and additionally have no transitive dependencies. A transitive dependency exists when there are functional dependencies.



SUMMARY

In this tutorial, you were **introduced** to some normalization standards, namely that each table should be a single subject, each intersection of a row and column should only contain one value and not multiple values, and there is no data item that is unnecessarily stored in more than one table. Then you learned about the three stages of the **normalization process** with specific rules to adhere to at each stage.

Source: Authored by Vincent Tran



TERMS TO KNOW

First Normal Form (1NF)

The first normal form (1NF) ensures that each column in the tables is single-valued, meaning it does not contain multiple values.

Second Normal Form (2NF)

In the second normal form (2NF), tables should be in the first normal form (1NF) to begin, but you also ensure there is no partial dependency.

Third Normal Form (3NF)

In the third normal form (3NF), tables should be in the second normal form (2NF) and additionally have no transitive dependencies. A transitive dependency exists when there are functional dependencies.