First Normal Form

by Sophia

WHAT'S COVERED

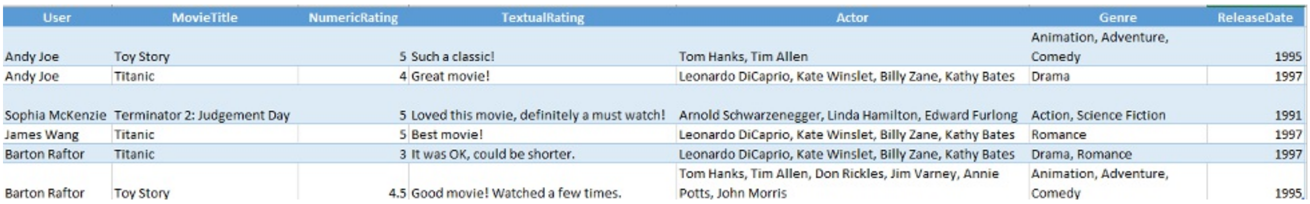
This tutorial explores the first normal form (1NF) on an unnormalized data set in two parts:

1. Adding Relationships
2. Splitting Column Data

1. Adding Relationships

Recall that in first normal form (1NF), you can only have single values at the intersection of each row and column, meaning there should be no repeating groups of data. In order to manage this appropriately, you need to remove any repeating group by creating two new relations. As a starting point, the primary key of the new relationship will be a combination of the primary key of the original relation plus the attribute from the newly created relation to ensure uniqueness.

Let us look at our movie rating table to see how this works:



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

This table has the following attributes:

- Movie (User, MovieTitle, NumericRating, TextualRating, Actor, Genre, ReleaseDate)

In reviewing the data, you do not have any attributes that uniquely identify each row. You will create a new primary key for the table called MovieId. Now our attributes include:

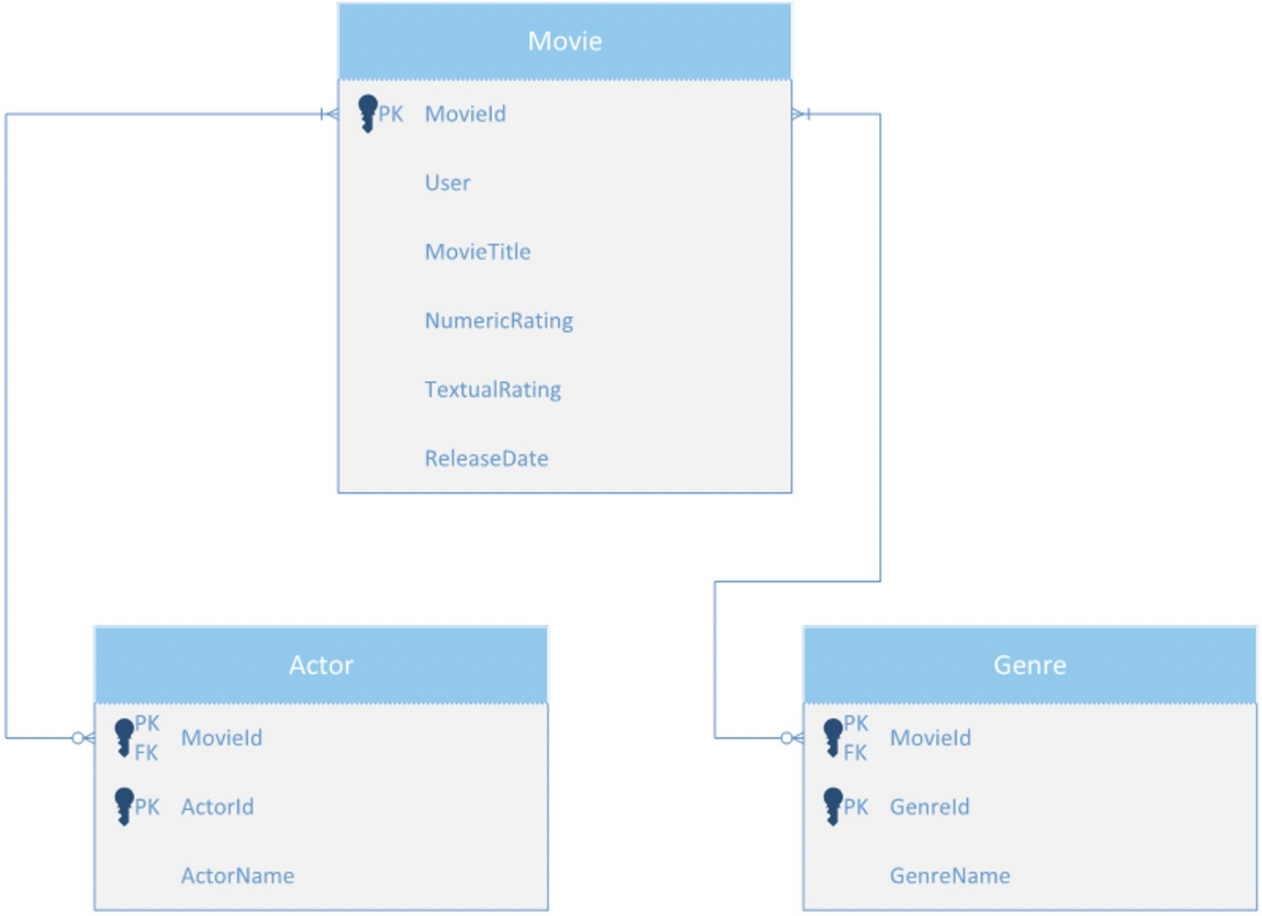
- Movie (MovieId, User, MovieTitle, NumericRating, TextualRating, Actor, Genre, ReleaseDate)

In this table, you also have a few repeating groups, including Actor and Genre. There can be many actors in a movie, and a movie can belong to more than one genre. To remove the repeating groups for Actor and Genre, you will create a new table for each. Now you have three tables:

- Movie (MovieId, User, MovieTitle, NumericRating, TextualRating, ReleaseDate)
- Actor (MovieId, ActorId, ActorName)

- Genre (MovieId, GenreId, GenreName)

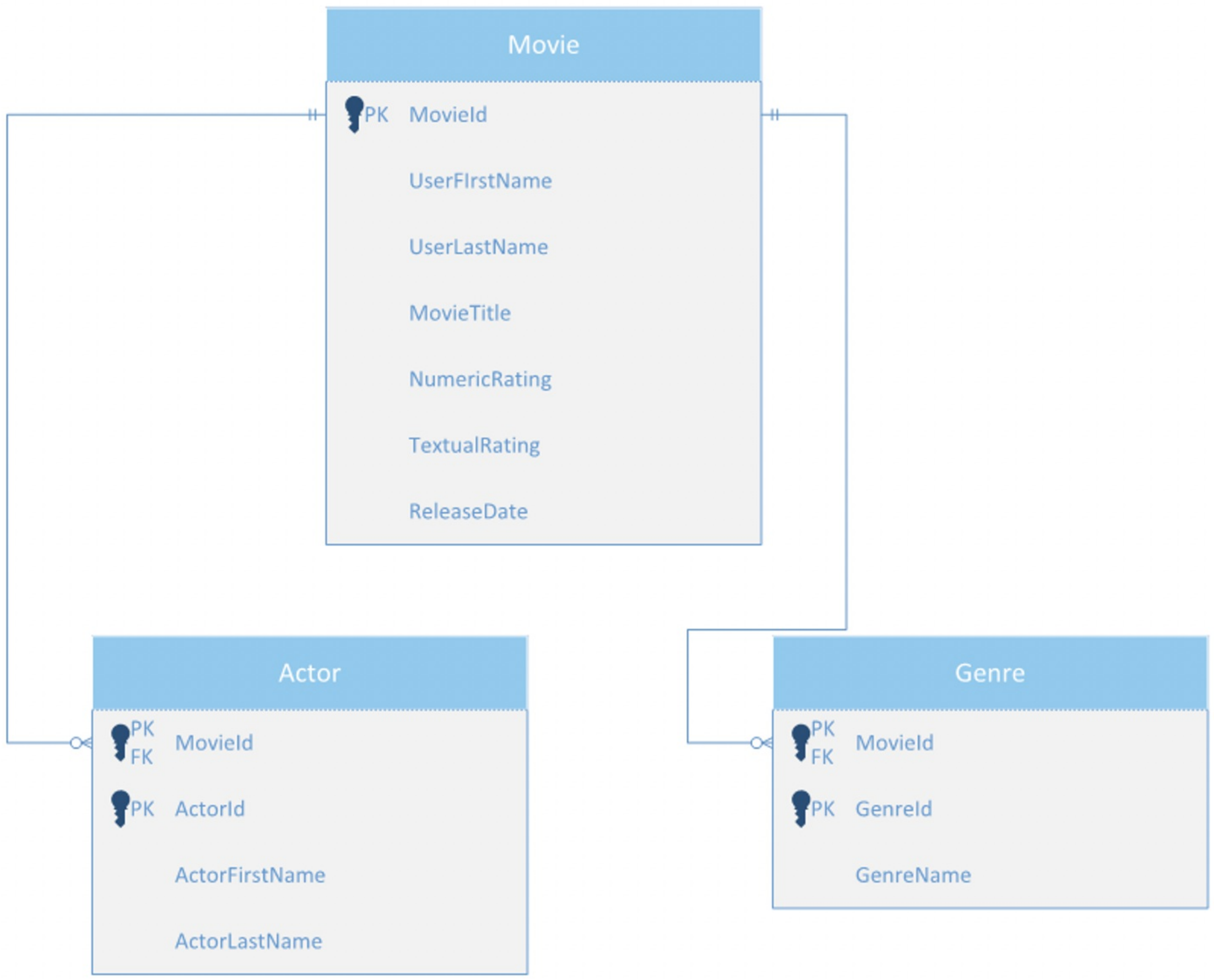
The resulting ERD should look like the following:



2. Splitting Column Data

By going through the first normal form (1NF), you have removed all of the repeating groups (multiple entries for a single record). However, there can still be scenarios where you may have multiple values stored in a single column. In our case, the ActorName and the User can be split into a first and last name, creating the following:

- Movie (MovieId, UserFirstName, UserLastName, MovieTitle, NumericRating, TextualRating, ReleaseDate)
- Actor (MovieId, ActorId, ActorFirstName, ActorLastName)
- Genre (MovieId, GenreId, GenreName)



This complete first normal form (1NF) is ready for us to move to the second normal form (2NF).

SUMMARY

In this tutorial, you learned that since you can only have single values at the intersection of each row and column, you need to **add relationships** for any repeating group. Then you learned that you need to **split column data** when you have scenarios where multiple values are stored in a single column. You saw an example of an ERD with relationships added and column split completing the first normal form (1NF).

Source: Authored by Vincent Tran