**Yet Another Normalization Example**

You’ve been hired by a new gaming company to design the database to handle online connections. Essentially, they need a table to hold the data of when each player connects to one of their games.

One of the developers tells you that having 1 table that holds the following info should be fine:

* Each player has an ID and a user name.
* The player can have multiple connections to multiple games.
* Each Game has an ID and description.
* Each connection will have the time the user logged in (logintime) and the server to which its connected to.
* Each server has an ID and a description

The developer tells you to use the following table and use the PlayerID as the primary key.

MultiPlayer (**PlayerId**, PlayerUserName,(GameId, GameDescription, LoginTime),ServerId, ServerName)

After reviewing the design, you feel that you can do a better design. To prove to the developer, you add some dummy data to prove the problems:

Multiplayer Table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PlayerId(PK) | PlayerUsername | GameID | GameDescription | LoginTime | ServerId | ServerName |
| 01 | MrReaper | 14,15 | Underwatch, PlanetCraft | 06/24/2017 800,06/24/2017 1000  | S01 | E01AppServer |
| 02 | WidowKiller | 14,15 | Underwatch, PlanetCraft | 06/24/2017 500, 06/24/2017 800 | S02 | W02AppServer |

**First Normal Form (1NF)**

You explain to the developer that this is not in 1NF (First Normal Form) because the values for GameID and GameDescription are not atomic (single valued). You can see this in the table design (denoted by the inner parenthesis)

MultiPlayer(**PlayerId**, PlayerUserName,(GameId, GameDescription, LoginTime), ,ServerId, ServerName)

So to change this design, you will have to remove the repeating groups. This is what 1NF is all about. To remove this repeating group, we might need to have single values for GameID, GameDescription and LoginTime. We can accomplish this if we had multiple rows for each game that the user plays. So the above table will now look like this:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PlayerId(PK) | PlayerUsername | GameID | GameDescription | LoginTime | ServerId | ServerName |
| 01 | MrReaper | 14 | Underwatch | 06/24/2017 800, | S01 | E01AppServer |
| 01 | MrReaper | 15 | PlanetCraft | 06/24/2017 1000 | S01 | E01AppServer |
| 02 | WidowKiller | 14 | Underwatch | 06/24/2017 500 | S02 | W02AppServer |
| 02 | WidowKiller | 15 | PlanetCraft | 06/24/2017 800 | S02 | W02AppServer |

This design is better because all values are single entry, but it has change our Primary Key. We can NO longer use Player ID as the Primary Key by itself. Based on the new design, the PK is now a composite PK composed of PlayerID AND GameID.

So, the design has removed the repeating groups, but the PK changed. This now denoted as:

MultiPlayer (**PlayerId**, PlayerUserName,**GameId**, GameDescription, LoginTime,ServerId, ServerName)

Now if we imagine loading data into this table, can see the values in each column are atomic (single valued) and we have a proper primary key

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **PlayerId(PK)** | PlayerUsername | **GameID(PK)** | GameDescription | LoginTime | ServerId | ServerName |
| 01 | MrReaper | 14 | Underwatch | 06/24/2017 800, | S01 | E01AppServer |
| 01 | MrReaper | 15 | PlanetCraft | 06/24/2017 1000 | S01 | E01AppServer |
| 02 | WidowKiller | 14 | Underwatch | 06/24/2017 500 | S02 | W02AppServer |
| 02 | WidowKiller | 15 | PlanetCraft | 06/24/2017 800 | S02 | W02AppServer |

This is now considered to be in 1NF (no repeating groups). So, we work on 2NF. To complete 2NF though, we need to understand the **functional dependencies** i.e. have a clear understand of which columns determine the other columns.

**Second Normal Form (2NF)**

After speaking with the gaming company, you write out the following dependencies:

PlayerId,GameId -> PlayerUsername,GameDescription,LoginTime,ServerId, ServerName

ServerId -> ServerName

GameId -> GameDescription

PlayerId -> PlayerUserName

With this information you can work on 2NF. 2NF asks that if any of columns that make up the primary key individually determine other columns, then new tables should be created to hold those columns and only dependent columns should be removed from the original table.

This is because even 2NF we can have data redundancy and other problems. For example, the data highlighted below is still duplicated:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **PlayerId(PK)** | PlayerUsername | **GameID(PK)** | GameDescription | LoginTime | ServerId | ServerName |
| 01 | MrReaper | 14 | Underwatch | 06/24/2017 800, | S01 | E01AppServer |
| 01 | MrReaper | 15 | PlanetCraft | 06/24/2017 1000 | S01 | E01AppServer |
| 02 | WidowKiller | 14 | Underwatch | 06/24/2017 500 | S02 | W02AppServer |
| 02 | WidowKiller | 15 | PlanetCraft | 06/24/2017 800 | S02 | W02AppServer |

Our Primary Key here is composed of PlayerID and GameID. So first we ask if PlayerID determines any other column. Based on this functional dependency, it does:

PlayerId -> PlayerUserName

So we will create a new table named “PlayerProfile” which will hold these 2 columns and we will remove PlayerUserName from the original table

PlayerProfile ( PlayerId, PlayerUserName)

MultiPlayer (**PlayerId**, ~~PlayerUserName~~,**GameId**, GameDescription, LoginTime,ServerId, ServerName)

Now, we will focus on the second column in the PK, GameId. GameID determines GameDescription so we will create a Game table:

Game (GameId, GameDescription)

MultiPlayer (**PlayerId**, ~~PlayerUserName~~,**GameId**, ~~GameDescription~~, LoginTime,ServerId, ServerName)

So,we are done with 2NF because there are no other columns in the PL. Our table design now looks like this:

PlayerProfile ( PlayerId, PlayerUserName)

Game (GameId, GameDescription)

MultiPlayer (**PlayerId**,**GameId**, LoginTime,ServerId, ServerName)

**Third Normal Form (3NF)**

Even though our new 3 table design removes some of the redundant values such as Game Description and Player UserName, we still have some problems. The Server Name column still repeats per connection.

MultiPlayer Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PlayerId(PK)** | **GameID(PK)** | LoginTime | ServerId | ServerName |
| 01 | 14 | 06/24/2017 800, | S01 | E01AppServer |
| 01 | 15 | 06/24/2017 1000 | S01 | E01AppServer |
| 02 | 14 | 06/24/2017 500 | S02 | W02AppServer |
| 02 | 15 | 06/24/2017 800 | S02 | W02AppServer |

We can test this for 3NF. 3Nf says that “determinants” (a column that determines the value of another column) can only exist in a table if it is a candidate key ( a column that determines ALL the other columns in a table).

Therefore, any determinants that are not candidate keys need to be moved to their own table.

So, we see based on this functional dependency:

ServerId -> ServerName

That serverId determines servername, and serverId is NOT a candidate key. Our move here is to create a new table to hold this info.

So we create the Server table:

Server (ServerID, ServerName)

And from the original table we remove servername

MultiPlayer (**PlayerId**,**GameId**, LoginTime, ServerId, ~~ServerName~~)

This completes our 3NF check because the Multiplayer table has no more determinants (since servername was remove).

**Final Tables**

Our Final table design is the following:

PlayerProfile ( **PlayerId**, PlayerUserName)

Game (**GameId**, GameDescription)

Server (**ServerId**, ServerName)

MultiPlayer (**PlayerId**,**GameId**, LoginTime,ServerId)

**Verification**

Why is this a better design?

We can use some sample data to see why this design is more efficient. Less duplication of data from our original table:

TABLE DESIGN ON NEXT PAGE….

Multiplayer Table

|  |  |  |  |
| --- | --- | --- | --- |
| **PlayerId(PK)** | **GameID(PK)** | LoginTime | ServerId |
| 01 | 14 | 06/24/2017 800, | S01 |
| 01 | 15 | 06/24/2017 1000 | S01 |
| 02 | 14 | 06/24/2017 500 | S02 |
| 02 | 15 | 06/24/2017 800 | S02 |

PlayerProfile Table Game Table

|  |  |
| --- | --- |
| **PlayerId(PK)** | PlayerUsername |
| 01 | MrReaper |
| 02 | WidowKiller |

|  |  |
| --- | --- |
| **GameID(PK)** | GameDescription |
| 14 | Underwatch |
| 15 | PlanetCraft |

Server Table

|  |  |
| --- | --- |
| **ServerId(PK)** | ServerName |
| S01 | E01AppServer |
| S02 | W02AppServer |