**Unit 2 Movie Database Conceptualization**

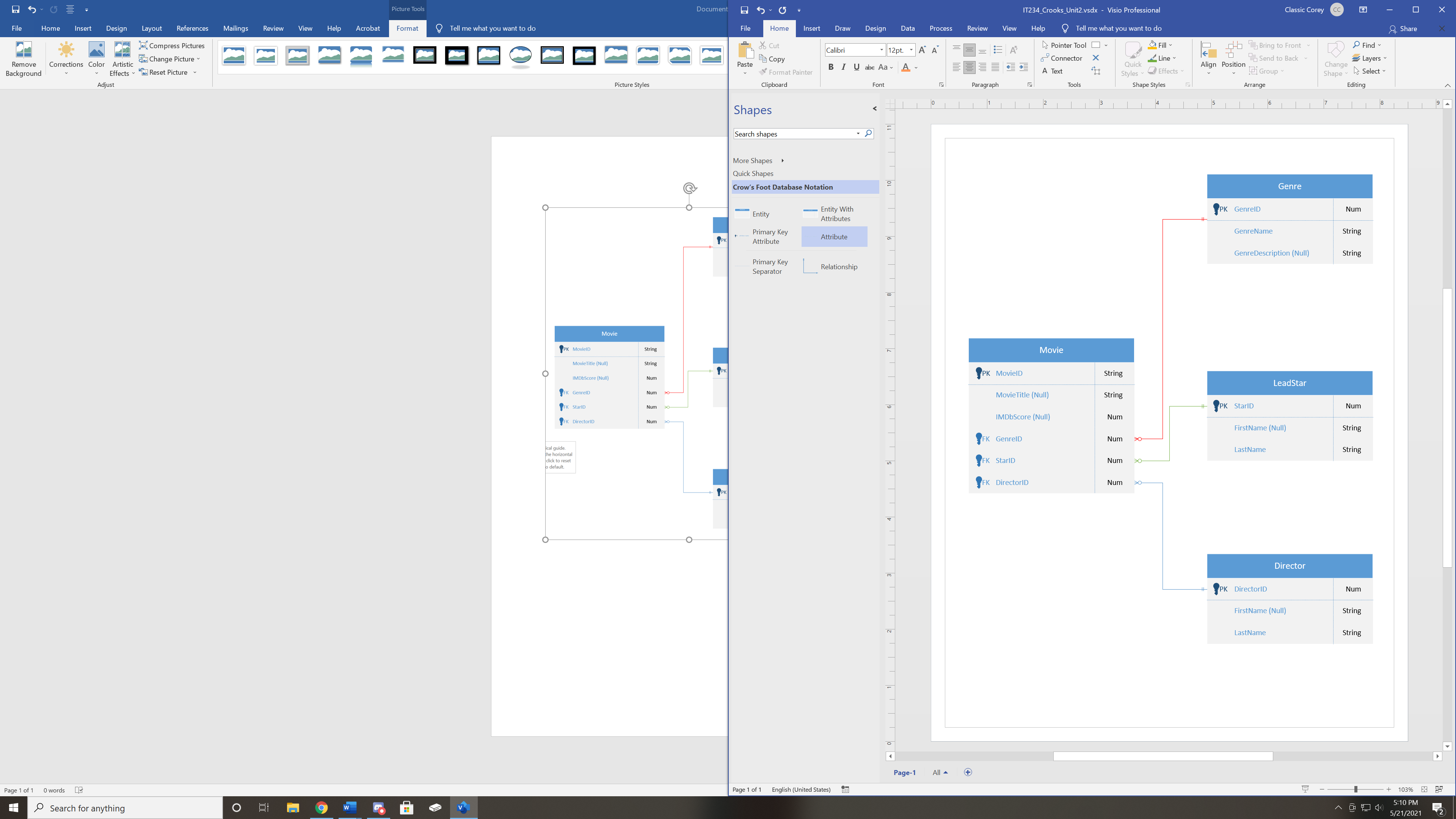
Corey Crooks

Purdue University Global

IT234 – Leon King

May 26, 2021

**Visio Database Conceptual Design**



**Movie Database Conceptualization**

Through further normalization of the Movies Spreadsheet, I have decided on 4 entities with varying amounts of attributes each. The entities are Movie, Genre, LeadStar, and Director. The Movie entity has the attributes MovieID, MovieTitle, IMDbScore, GenreID, StarID, and DirectorID. The Genre entity has the attributes GenreID, GenreName, and GenreDescription. The LeadStar entity has the attributes StarID, FirstName, and LastName. The Director entity has the attributes DirectorID, FirstName, and LastName. The ID attributes in each entity are used as foreign keys in the Movie entity except for MovieID, which serves as the table’s primary key. Nearly one-third of the attributes in the database as it stands can be left blank, or have nullability. These attributes are MovieTitle, IMDbScore, GenreDescription, and the two FirstName attributes of the LeadStar and Director entities. These can be left blank because they aren’t particularly needed as essential descriptors, as we already get the individuality of the records from the ID attributes (Wenzel, 2021). For instance, director John Doe and a director Mr. Doe would be distinguished by the required DirectorID attribute. Requiring input for at least the director’s last name makes sure to provide at least some context beyond just a generic number. A similar principle could be established for the GenreDescription attribute. If we did not require the GenreName attribute, then the Movie entity may be full of records that have “124298” and “90671324” as genre placements. This would not help anybody involved. Now, if we require the GenreName attribute, we could have records like “Thriller” and “Drama” as placements. This is a lot easier to understand rather than their numeric counterparts. We can then go further and describe the genre in-depth to add more context to how it gets assigned to a movie.

Each record per entity is ensured to be unique via the primary keys found at the top of each entity attribute section (Rabelo, 2020). Each ID tag serves as either a primary key or a foreign key depending on the table it is found in. That is to say, the Movie entity’s primary key is MovieID. The Genre entity’s primary key is GenreID. The LeadStar entity’s primary key is StarID. The Director entity’s primary key is DirectorID. GenreID, StarID, and DirectorID are also found in the Movie entity’s attribute field as foreign keys. Setting the entities up like this ensures that each primary key is guaranteed to be unique by a numeric value that can be used in the Movie entity to ensure a lack of duplicate entries, and to be able to refer to separate records with minimal information. Requiring an input for the primary key ensures that there are no ambiguous or duplicate records found in any one table.

# **References**

Rabelo, J. (2020, August 14). *Primary Key*. Retrieved from Techopedia: https://www.techopedia.com/definition/5547/primary-key

Wenzel, K. (2021, May 21). *What is a Database NULL Value?* Retrieved from EssentialSQL: https://www.essentialsql.com/get-ready-to-learn-sql-server-what-is-a-null-value/