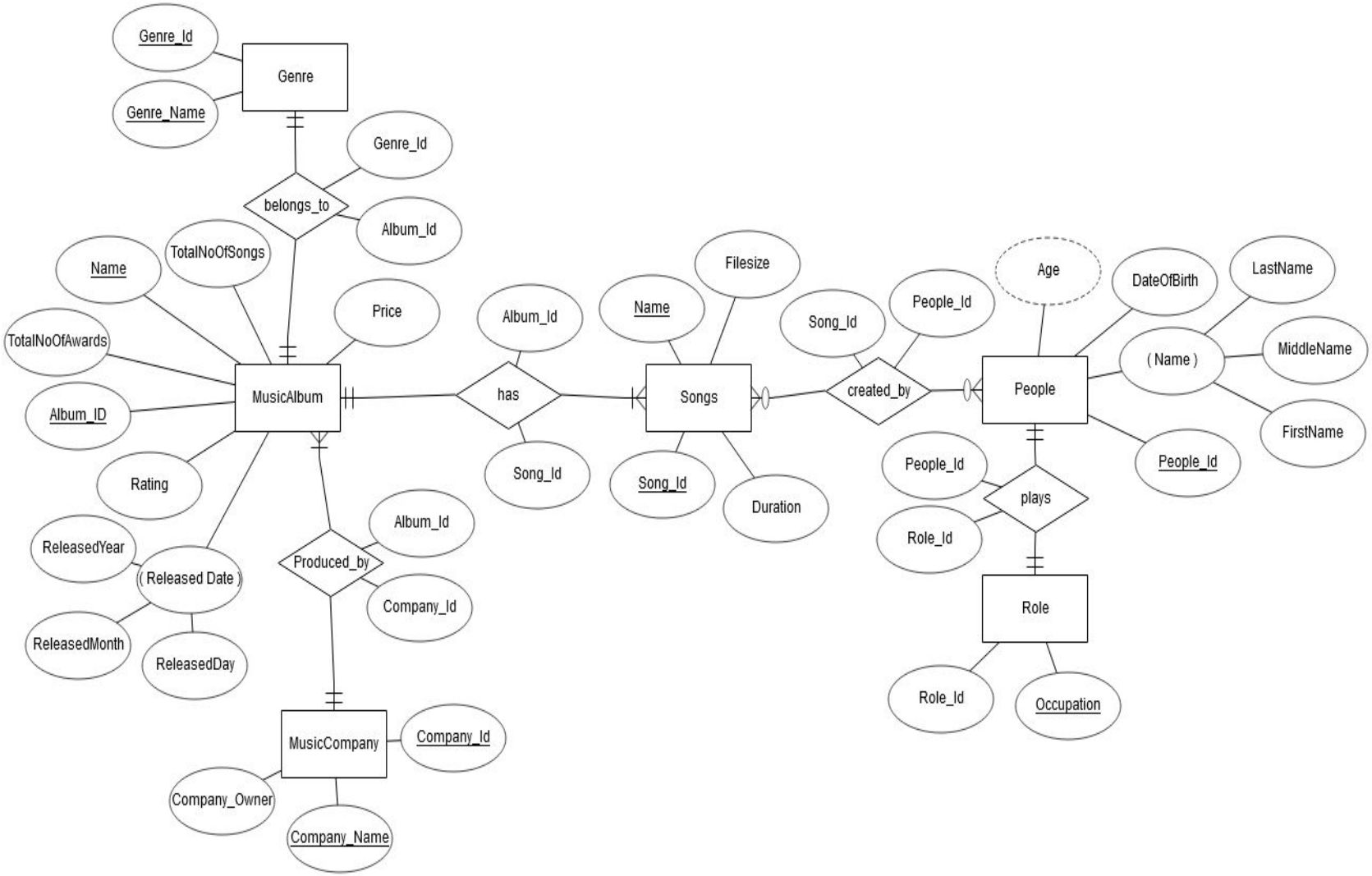
ER Diagram:


Genre_Id
Genre
Genre_Name
Genre_Id
belongs_to
Album_Id
Name
TotalNoOfSongs
Price
TotalNoOfAwards
Album_ID
MusicAlbum
Rating
ReleasedYear
(Released Date)
ReleasedMonth
ReleasedDay
Produced_by
Album_Id
Company_Id
MusicCompany
Company_Id
Company_Owner
Company_Name
Album_Id
has
Song_Id
Name
Filesize
Songs
Song_Id
Duration
Song_Id
created_by
People_Id
Age
DateOfBirth
LastName
People
(Name)
MiddleName
FirstName
People_Id
People_Id
plays
Role_Id
Role
Role_Id
Occupation

Description:

~MusicAlbum

The MusicAlbum table has 9 attributes of its own known as AlbumID ,AlbumName,ReleaseDay ,ReleaseMonth,ReleaseYear ,TotalNoOfSongs ,TotalNoOfAdwards Price ,Company_id and genre. Among which the two unique keys are AlbumID and AlbumName, the Album ID is a system-generated number known as the surrogate key which helps organize the database, and AlbumName is used as the unique candidate key. Table MusicAlbum has two relationships: [1] one to one with genre Table and [2] many to one with MusicCompany Table. Because of the cardinality of the relationships, the MusicAlbum table includes 2 more attributes or columns as the foreign key. The genre_id column is a foreign key in MusicAlbum Table is a foreign key referencing the primary key of the genre Table. The Company_id is also a foreign key in the MusicAlbum Table referencing the primary key in the MusicCompany Table.

~MusicCompany

The MusicCompany Table has 3 attributes where CompanyID is chosen as the primary key and the Company_Name is a candidate key which is unique as any two companies cannot have the same name.

~Genre

The genre Table keeps track of all the possible genres of movies with the GenreID as the primary key and the Type as a candidate unique attribute.

~Songs

The Songs Table has 4 attributes with SongID as the primary key of the table. The SongName is a unique candidate key. The Songs Table has two relationships: many to one with the MusicAlbum Table and many to many with people Table. Therefore the Song has 1 extra column as a foreign key to reference the album_id from MusicAlbum Table.

~People

The people table has its own primary key and a candidate key and has a one on one relationship with role which is established by a foreign key.

~Role

The Role Table has only 2 attributes with a primary key RoleID and a candidate key occupation. The Occupation column, therefore, is unique.

~CreatedBy

The CreatedBy Table has two attributes containing the primary keys of both Songs and people Tables. This is because the people and Songs Tables have a many to many relationship.