Emily Evans

Miguel Fernandez

Brenton Smelser

Shannon Smith

Group 1 – Data Dictionary

**Album Table**: This table stores the detailed information on an album.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary Key? | Name | Type | Size | Description |
| Primary | albumId | int |  | Number given to each album to uniquely identify it |
|  | title | varchar | 100 | Title of the album |
|  | recordLabel | varchar | 100 | Name of record label that produced the album. |
|  | releaseDate | date |  | Release date of album to the public. |

**Artist Table**: This table stored the detailed information on an artist/band/group.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary Key? | Name | Type | Size | Description |
| Primary | artistId | int |  | Number given to each artist to uniquely identify it. |
|  | name | varchar | 45 | artist name |
|  | formDate | date |  | date of formation |
|  | breakupDate | date |  | date of breakup |
|  | formationZipCode | int |  | zipcode of artist origin |

**AttendedPerformance Table**: This table stores links between users and the performances they attened.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary Key? | Name | Type | Size | Description |
| Primary | username | varchar | 16 | The username of the user who attended a performance. |
| Primary | performanceId | int |  | Unique Id that corresponds to the performanceId in the Performance Table that represents what performance the user attended. |

**Comment Table**: This table stores the detailed information on a comment entered by a user. This information can be updated and deleted by a moderator.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary Key? | Name | Type | Size | Description |
| Primary | commentId | int |  | Number given to each comment to uniquely identify it. |
|  | username | varchar | 16 | The username of the user who composed the comment. This can be null if the user deletes there account. |
|  | artistId | int |  | Number that matches a unique artist id to relate the comment to an artist. This can be null if it is related to a performance. |
|  | performanceId | int |  | number that matches a unique performanceId that tells this comment is related to a performance. This can be null if the comment is related to an artist. |
|  | comment | varchar | 300 | Content of the comment that is entered by the user. |
|  | postdate | date |  | Date when the comment was posted. |

**Favorite Table**: This table stores links between users and their favorite artists.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary Key? | Name | Type | Size | Description |
| Primary | username | varchar | 16 | Username of the user who is related to this favorite. |
| Primary | artistId | int |  | Number that corresponds to unique artist id for the artist that is a favorite of the user. |

**Member Table**: This table stores the detailed information on a member of a group/band/artist. This information is entered by a moderator.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary Key? | Name | Type | Size | Description |
| Primary | memberId | int |  | Number given to each member to uniquely identify them. |
| Primary | artistId | int |  | Number that corresponds to unique artist id linking the member to an artist/band/group. |
|  | joinDate | date |  | Date for when the member joined an artist/band/group |
|  | leaveDate | date |  | Date for when member left artist/band/group. This can be null if the member is still in with the artist/band/group. |
|  | name | varchar | 45 | The full name of the member. |

**Performance Table**: This table stores the detailed information on each performance. This information is entered by the moderator.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary Key? | Name | Type | Size | Description |
| Primary | performanceId | int |  | Number given to each performance to uniquely identify it |
| Primary | venueId | int |  | Number that corresponds to unique venue id of where the performance was performed. |
|  | duration | double |  | The duration/length of the concert in minutes. |
|  | date | date |  | The date when the concert was performed. |
|  | title | varchar | 60 | Title of the performance. |

**PerformancePlaylist Table**: This table stores what forms a list of songs that are performed at a given performance. There is an artist id here because often at concerts there could be a live performance of a song by an artist other than the creator.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary Key? | Name | Type | Size | Description |
| Primary | performanceId | int |  | Number that corresponds to unique performance id number |
| Primary | songId | int |  | Number that corresponds to unique song id linking the song to a given performance. |
| Primary | artistId | int |  | Number that corresponds to unique artist id that represents what artist performed a song. In the event the creator of the song is not who performed it. |

**Song Table**: This table stored the detailed information for each song in the system. The song is referenced by the song id throughout the program.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary Key? | Name | Type | Size | Description |
| Primary | song id | int |  | number given to each song to uniquely identify it |
|  | title | varchar | 100 | The title of the song. |
|  | duration | double |  | The duration of song (minutes) |

**User Table**: This table stores the detailed information for each user, this data is collected during registration. Users are referred to by their username in the rest of the database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary Key? | Name | Type | Size | Description |
| Primary | username | varchar | 16 | A unique username selected by the user upon registration. |
|  | firstName | varchar | 45 | The user’s first name. |
|  | lastName | Varchar | 45 | The user’s last name. |
|  | age | Int |  | The user’s age. Must be greater than 13. |
|  | email | Varchar | 100 | The user’s email address. |
|  | password | Varchar | 255 | The user’s encrypted password. |
|  | zipcode | Int |  | The user’s zipcode. |
|  | isModerator | Tinyint | 1 | A number (either 0 or 1) given to each user to show whether they have access to modify the site. 1 implies they are a moderator and 0 implies they are not. 0 is the default. |

**Venue Table**: This table stored the detailed information about each venue, this information is all entered by a moderator.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary Key? | Name | Type | Size | Description |
| Primary | venueId | Int |  | A number given to each venue to uniquely identify it |
|  | name | Varchar | 100 | The name of the venue. |
|  | streetAddress | Varchar | 200 | The street address of the venue. |
|  | city | Varchar | 100 | The city where the venue is located. |
|  | state | Varchar | 100 | The state where the venue is located. |
|  | zipcode | Int |  | The zipcode of where the venue is located. |

**Tracklist Table**: This table holds the information that forms links between an album and the songs on that album along with who performed that song. There will be many rows in this list specifically one row per song/track on an album.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Primary Key? | Name | Type | Size | Description |
| Primary | albumId | int |  | A number that corresponds to unique album id number. |
| Primary | songId | int |  | A number that corresponds to unique song id that related the song to an album. |
| Primary | artistId | int |  | A number that corresponds to unique artist id that is the artist who plays this song on this album. |
|  | track\_number | int |  | The track number for this song on the album. |