**SOFTWARE DOCUMENTATION**

EER DIAGRAM:

TABLES

#### The Artist Table

The Artist table lists information for all artists.

The Artist table is referred to by a foreign key in the Event, Album and Track tables.

**Columns**

* artist\_id: A surrogate primary key used to uniquely identify each artist in the table.
* name: The artist's name.
* genre: A foreign key identifying the artist's genre.
* playcount: The artist's play count given by LastFM.
* listeners: The artist's listeners given by LastFM.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Columns | | | | | | |
| Column name | Datatype | NN | UQ | UN | AI | Default |
| artist\_id | SMALLINT(5) | V |  | V | V |  |
| name | VARCHAR(45) | V |  |  |  |  |
| genre | VARCHAR(45) |  |  |  |  | NULL |
| playcount | INT(11) |  |  | V |  | NULL |
| listeners | INT(11) |  |  | V |  | NULL |

|  |  |  |
| --- | --- | --- |
| Indexes | | |
| Index name | Type | Index Column |
| PRIMARY | PRIMARY | artist\_id |
| listeners\_idx | INDEX | listeners |

#### The Event Table

The Event table lists all the artist's events, including information on sale date, event date, and where the event is taking place.

The event table refers to the Artist, Country and City table using a foreign key.

##### Columns

* event\_id: A surrogate primary key used to uniquely identify each event in the table.
* artist\_id: A foreign key identifying the artist which performs the event.
* description: Details of the event
* sale\_date: Starting sale date.
* date: The event date.
* venue: The name of the venue where the event takes place.
* city\_id: A foreign key identifying the city where the event takes place.
* country\_id: A foreign key identifying the country where the event takes place.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Columns | | | | | | |
| Column name | Datatype | NN | UQ | UN | AI | Default |
| event\_id | SMALLINT(5) | V |  | V | V |  |
| artist\_id | SMALLINT(5) | V |  | V |  |  |
| description | TEXT |  |  |  |  | NULL |
| sale\_date | DATE |  |  |  |  | NULL |
| date | DATE | V |  |  |  |  |
| venue | VARCHAR(150) | V |  |  |  |  |
| city\_id | SMALLINT(5) | V |  | V |  |  |
| county\_id | SMALLINT(5) | V |  | V |  |  |

|  |  |  |
| --- | --- | --- |
| Indexes | | |
| Index Name | Type | Index Column |
| PRIMARY | PRIMARY | event\_id |
| fk\_event\_artist\_idx | INDEX | artist\_id |
| fk\_event\_city\_idx | INDEX | city\_id |
| fk\_event\_country\_idx | INDEX | country\_id |
| event\_date\_idx | INDEX | date |

|  |  |  |  |
| --- | --- | --- | --- |
| Foreign Keys | | | |
| Foreign Key Name | Refers to | On Update | On Delete |
| fk\_event\_artist | Artist.artist\_id | CASCADE | CASCADE |
| fk\_event\_city | City.city\_id | CASCADE | RESTRICT |
| fk\_event\_country | Country.country\_id | CASCADE | RESTRICT |

|  |  |  |
| --- | --- | --- |
| Triggers | | |
| Trigger Name | Type | Checks |
| event\_BEFORE\_INSERT | BEFORE INSERT | event date > current date  sale date <= event date  city\_id corresponded to country\_id |
| check\_dates | BEFORE UPDATE | sale date <= event date |

#### The Country Table

The Country table contains a list of countries.

The Country table is referred to by a foreign key in the City and Event tables.

##### Columns

* country\_id: A surrogate primary key used to uniquely identify each country in the table.
* country: The name of the country.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Columns | | | | | | |
| Column name | Datatype | NN | UQ | UN | AI | Default |
| country\_id | SMALLINT(5) | V |  | V | V |  |
| country | VARCHAR(45) | V | V |  |  |  |

|  |  |  |
| --- | --- | --- |
| Indexes | | |
| Index Name | Type | Index Column |
| PRIMARY | PRIMARY | country\_id |
| country\_UNIQUE | UNIQUE | country |

#### The City Table

The City table contains a list of cities.

The City table refers to the Country table using a foreign key, and is referred to by a foreign key in the Event table.

##### Columns

* city\_id: A surrogate primary key used to uniquely identify each city in the table.
* city: The name of the city.
* country\_id: A foreign key identifying the country that the city belongs to.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Columns | | | | | | |
| Column name | Datatype | NN | UQ | UN | AI | Default |
| city\_id | SMALLINT(5) | V |  | V | V |  |
| city | VARCHAR(45) | V |  |  |  |  |
| country\_id | SMALLINT(5) | V |  | V |  |  |

|  |  |  |
| --- | --- | --- |
| Indexes | | |
| Index Name | Type | Index Column |
| PRIMARY | PRIMARY | city\_id |
| city\_idx | INDEX | city |
| fk\_city\_country\_idx | INDEX | county\_id |

|  |  |  |  |
| --- | --- | --- | --- |
| Foreign Keys | | | |
| Foreign Key Name | Refers to | On Update | On Delete |
| fk\_city\_country | Country.country\_id | CASCADE | RESTRICT |

#### The Album Table

The Album table lists all the artist's albums.

The Album table refers to the Artist table using a foreign key, and is referred to by a foreign key in the AlbumTracks table.

##### Columns

* album\_id: A surrogate primary key used to uniquely identify each album in the table.
* title: The album title.
* artist\_id: A foreign key identifying the artist which released the album.
* release\_year: The release year of the album.
* num\_of\_tracks: The number of tracks in the album.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Columns | | | | | | |
| Column name | Datatype | NN | UQ | UN | AI | Default |
| album\_id | SMALLINT(5) | V |  | V | V |  |
| title | VARCHAR(45) | V |  |  |  |  |
| artist\_id | SMALLINT(5) | V |  | V |  |  |
| release\_year | YEAR(4) |  |  |  |  | NULL |
| num\_of\_tracks | TINYINT(5) |  |  | V |  | NULL |

|  |  |  |
| --- | --- | --- |
| Indexes | | |
| Index Name | Type | Index Column |
| PRIMARY | PRIMARY | album\_id |
| fk\_artist\_album\_idx | INDEX | artist\_id |
| album\_release\_year\_idx | INDEX | release\_year |

|  |  |  |  |
| --- | --- | --- | --- |
| Foreign Keys | | | |
| Foreign Key Name | Refers to | On Update | On Delete |
| fk\_artist\_album | Artist.artist\_id | CASCADE | CASCADE |

|  |  |  |
| --- | --- | --- |
| Triggers | | |
| Trigger Name | Type | Checks |
| check\_release\_year | BEFORE INSERT | release year <= current year |

#### The Track Table

The Track table lists all the artist's tracks.

The Track table refers to the Artist table using a foreign key, and is referred to by a foreign key in the AlbumTracks table.

##### Columns

* track\_id: A surrogate primary key used to uniquely identify each track in the table.
* title: The track title.
* artist\_id: A foreign key identifying the artist which released the track.
* duration: The track's duration time on seconds.
* listeners: The track's listeners given by LastFM.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Columns | | | | | | |
| Column name | Datatype | NN | UQ | UN | AI | Default |
| track\_id | INT(10) | V |  | V | V |  |
| title | VARCHAR(45) | V |  |  |  |  |
| artist\_id | SMALLINT(5) | V |  | V |  |  |
| duration | SMALLINT(5) |  |  | V |  | NULL |
| listeners | INT(11) |  |  | V |  | NULL |

|  |  |  |
| --- | --- | --- |
| Indexes | | |
| Index Name | Type | Index Column |
| PRIMARY | PRIMARY | track\_id |
| fk\_track\_artist\_idx | INDEX | artist\_id |

|  |  |  |  |
| --- | --- | --- | --- |
| Foreign Keys | | | |
| Foreign Key Name | Refers to | On Update | On Delete |
| fk\_track\_artist | Artist.artist\_id | CASCADE | CASCADE |

#### The Lyrics Table

The Lyrics table lists all the track's lyrics. This tables uses MyISAM engine in order to use FULLTEXT index on lyrics column. The track\_id IS NOT refers to Track table, because MyISAM engine does not support foreign keys. Although, the track\_id appears in Track table.

##### Columns

* track\_id: A surrogate primary key used to uniquely identify each lyrics in the table.
* lyrics: The track's lyrics.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Columns | | | | | | |
| Column name | Datatype | NN | UQ | UN | AI | Default |
| track\_id | INT(10) | V |  | V |  |  |
| lyrics | TEXT |  |  |  |  | NULL |

|  |  |  |
| --- | --- | --- |
| Indexes | | |
| Index name | Type | Index Column |
| PRIMARY | PRIMARY | track\_id |
| lyrics\_idx | FULLTEXT | lyrics |

#### The AlbumTracks Table

The AlbumTracks table lists all the tracks of the album.

The AlbumTracks table refers to the album and Track tables using a foreign key.

##### Columns

* album\_id: A foreign key identifying the album.
* track\_id: A foreign key identifying the track.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Columns | | | | | | |
| Column name | Datatype | NN | UQ | UN | AI | Default |
| album\_id | SMALLINT(5) | V |  | V | V |  |
| track\_id | INT(10) | V |  | V |  |  |

|  |  |  |
| --- | --- | --- |
| Indexes | | |
| Index Name | Type | Index Column |
| PRIMARY | PRIMARY | album\_id, track\_id |
| fk\_track\_idx | INDEX | track\_id |

|  |  |  |  |
| --- | --- | --- | --- |
| Foreign Keys | | | |
| Foreign Key Name | Refers to | On Update | On Delete |
| fk\_album | Album.album\_id | CASCADE | CASCADE |
| fk\_track | Track.track\_id | CASCADE | CASCADE |

VIEWS

STORED PROCEDURES

#### The sp\_insertAlbum Stored Procedure

##### Description

The sp\_insertAlbum stored procedure is used to insert a new album to Album table. If the album is already exists (i.e. the artist already had an album with the same title), then error is returned.

##### Parameters

* p\_title: The title of album.
* p\_artist\_id: The ID of the artist. Must be a valid ID (i.e. exists in Artist table).
* p\_release\_year: The release year of the album. Cannot be older than current year.
* p\_num\_of\_tracks: The number of tracks in the album.

#### The sp\_insertEvent Stored Procedure

##### Description

The sp\_insertEvent stored procedure is used to insert a new event to Event table.

##### Parameters

* p\_artist\_id: The ID of the artist. Must be a valid ID (i.e. exists in Artist table).
* p\_description: Details of the event.
* p\_sale\_date: Starting sale date. Cannot be older than the event date.
* p\_date: The event date. Cannot be earlier than current date.
* p\_venue: The name of the venue where the event takes place.
* p\_city\_id: The ID of the artist. Must be a valid ID (i.e. exists in City table).

#### The sp\_updateEventDate Stored Procedure

##### Description

The sp\_updateEventDate stored procedure is used to update the event date.

##### Parameters

* p\_event\_id: The ID of the event. Must be a valid ID (i.e. exists in Event table).
* p\_event\_date: The event date. Cannot be earlier than sale date.