

# Part 1: Выбор Сценария

Для данной работы выбран сценарий: Сервис потоковой передачи музыки. Система управляет артистами, альбомами, песнями, пользователями, плейлистами, жанрами и избранными песнями пользователей.

## Part 2: Проектирование Базы Данных и Документация

### Идентификация Сущностей и Атрибутов:

1. **Артисты (Artists):** исполнители музыкальных композиций.
2. **Альбомы (Albums):** сборники песен, выпускаемые артистами.
3. **Песни (Songs):** музыкальные треки, которые могут входить в разные альбомы и иметь нескольких артистов и жанров.
4. **Пользователи (Users):** зарегистрированные пользователи сервиса.
5. **Плейлисты (Playlists):** коллекции песен, созданные пользователями.
6. **Песни в плейлистах (PlaylistSongs):** связь многие-ко-многим между плейлистами и песнями.
7. **Жанры (Genres):** музыкальные жанры, к которым принадлежат песни.
8. **Избранное (Favorites):** связь многие-ко-многим между пользователями и песнями (избранные треки).
9. **Песни в альбомах (AlbumSongs):** связь многие-ко-многим между альбомами и песнями.
10. **Песни и артисты (SongArtists):** связь многие-ко-многим между песнями и артистами.
11. **Песни и жанры (SongGenres):** связь многие-ко-многим между песнями и жанрами.

### Проектирование Таблиц

#### 1. Table Name: Artists

- **Description:** Хранит информацию об исполнителях.
- **Attributes:**
  - ArtistID: INTEGER, PK, NOT NULL, UNIQUE
  - Name: VARCHAR(100), NOT NULL, UNIQUE
  - Country: VARCHAR(50)
- **Constraints:**
  - PK\_Artists: PRIMARY KEY (ArtistID)
  - UQ\_ArtistName: UNIQUE (Name)

#### 2. Table Name: Albums

- **Description:** Содержит информацию об альбомах.
- **Attributes:**
  - AlbumID: INTEGER, PK, NOT NULL, UNIQUE
  - Title: VARCHAR(100), NOT NULL
  - ReleaseDate: DATE
  - ArtistID: INTEGER, FK (REFERENCES Artists), NOT NULL
- **Constraints:**
  - PK\_Albums: PRIMARY KEY (AlbumID)
  - FK\_Albums\_Artists: FOREIGN KEY (ArtistID) REFERENCES Artists(ArtistID)

### 3. Table Name: Songs

- **Description:** Содержит информацию о песнях.
- **Attributes:**
  - SongID: INTEGER, PK, NOT NULL, UNIQUE
  - Title: VARCHAR(100), NOT NULL
  - DurationSec: NUMERIC(6,2), NOT NULL (*длительность песни в секундах с точностью*)
- **Constraints:**
  - PK\_Songs: PRIMARY KEY (SongID)

### 4. Table Name: Users

- **Description:** Хранит данные пользователей.
- **Attributes:**
  - UserID: INTEGER, PK, NOT NULL, UNIQUE
  - Username: VARCHAR(50), NOT NULL, UNIQUE
  - Email: VARCHAR(100), NOT NULL, UNIQUE
  - RegistrationDate: TIMESTAMPTZ, DEFAULT CURRENT\_TIMESTAMP
- **Constraints:**
  - PK\_Users: PRIMARY KEY (UserID)
  - UQ\_Username: UNIQUE (Username)
  - UQ\_Email: UNIQUE (Email)

### 5. Table Name: Playlists

- **Description:** Плейлисты пользователей.
- **Attributes:**
  - PlaylistID: INTEGER, PK, NOT NULL, UNIQUE
  - Name: VARCHAR(100), NOT NULL
  - UserID: INTEGER, FK (REFERENCES Users), NOT NULL
- **Constraints:**
  - PK\_Playlists: PRIMARY KEY (PlaylistID)
  - FK\_Playlists\_Users: FOREIGN KEY (UserID) REFERENCES Users(UserID)

### 6. Table Name: PlaylistSongs

- **Description:** Связующая таблица для связи многие-ко-многим между плейлистами и песнями.
- **Attributes:**
  - PlaylistID: INTEGER, PK, FK (REFERENCES Playlists), NOT NULL
  - SongID: INTEGER, PK, FK (REFERENCES Songs), NOT NULL
- **Constraints:**
  - PK\_PlaylistSongs: PRIMARY KEY (PlaylistID, SongID)
  - FK\_PlaylistSongs\_Playlists: FOREIGN KEY (PlaylistID) REFERENCES Playlists(PlaylistID)
  - FK\_PlaylistSongs\_Songs: FOREIGN KEY (SongID) REFERENCES Songs(SongID)

### 7. Table Name: Genres

- **Description:** Музыкальные жанры.
- **Attributes:**
  - GenreID: INTEGER, PK, NOT NULL, UNIQUE

- Name: VARCHAR(50), NOT NULL, UNIQUE
- **Constraints:**
  - PK\_Genres: PRIMARY KEY (GenreID)
  - UQ\_GenreName: UNIQUE (Name)

## 8. Table Name: Favorites

- **Description:** Избранные треки. Связующая таблица для связи многие-ко-многим между пользователями и песнями.
- **Attributes:**
  - UserID: INTEGER, PK, FK (REFERENCES Users), NOT NULL
  - SongID: INTEGER, PK, FK (REFERENCES Songs), NOT NULL
  - AddedDate: TIMESTAMPTZ, DEFAULT CURRENT\_TIMESTAMP
- **Constraints:**
  - PK\_Favorites: PRIMARY KEY (UserID, SongID)
  - FK\_Favorites\_Users: FOREIGN KEY (UserID) REFERENCES Users(UserID)
  - FK\_Favorites\_Songs: FOREIGN KEY (SongID) REFERENCES Songs(SongID)

## 9. Table Name: AlbumSongs

- **Description:** Связующая таблица многие-ко-многим между альбомами и песнями.
- **Attributes:**
  - AlbumID: INTEGER, PK, FK (REFERENCES Albums), NOT NULL
  - SongID: INTEGER, PK, FK (REFERENCES Songs), NOT NULL
- **Constraints:**
  - PK\_AlbumSongs: PRIMARY KEY (AlbumID, SongID)
  - FK\_AlbumSongs\_Albums: FOREIGN KEY (AlbumID) REFERENCES Albums(AlbumID)
  - FK\_AlbumSongs\_Songs: FOREIGN KEY (SongID) REFERENCES Songs(SongID)

## 10. Table Name: SongGenres

- **Description:** Связующая таблица многие-ко-многим между песнями и жанрами.
- **Attributes:**
  - SongID: INTEGER, PK, FK (REFERENCES Songs), NOT NULL
  - GenreID: INTEGER, PK, FK (REFERENCES Genres), NOT NULL
- **Constraints:**
  - PK\_SongGenres: PRIMARY KEY (SongID, GenreID)
  - FK\_SongGenres\_Songs: FOREIGN KEY (SongID) REFERENCES Songs(SongID)
  - FK\_SongGenres\_Genres: FOREIGN KEY (GenreID) REFERENCES Genres(GenreID)

## 11. Table Name: SongArtists

- **Description:** Связующая таблица многие-ко-многим между песнями и артистами.
- **Attributes:**
  - SongID: INTEGER, PK, FK (REFERENCES Songs), NOT NULL
  - ArtistID: INTEGER, PK, FK (REFERENCES Artists), NOT NULL
- **Constraints:**
  - PK\_SongArtists: PRIMARY KEY (SongID, ArtistID)
  - FK\_SongArtists\_Songs: FOREIGN KEY (SongID) REFERENCES Songs(SongID)
  - FK\_SongArtists\_Artists: FOREIGN KEY (ArtistID) REFERENCES Artists(ArtistID)

## Взаимосвязи:

- **Artists — Albums (1:N)**: один артист может выпускать много альбомов, каждый альбом связан с одним артистом.
- **Albums — Songs (M:N)**: одна песня может входить в несколько альбомов (сборники песен, the best итд.), а один альбом содержит много песен. Связь реализована через AlbumSongs.
- **Songs — Genres (M:N)**: одна песня может иметь несколько жанров, один жанр может включать много песен. Связь через SongGenres.
- **Songs — Artists (M:N)**: одна песня может иметь несколько артистов (коллаборации), один артист может иметь много песен. Связь через SongArtists.
- **Users — Playlists (1:N)**: один пользователь может создавать много плейлистов.
- **Playlists — Songs (M:N)**: песни могут входить в много плейлистов, плейлисты содержат много песен. Связь через PlaylistSongs.
- **Users — Songs (M:N)**: пользователи могут добавлять песни в избранное, связь через Favorites.

## Part 3: ER-Диаграмма

