Part 1: Выбор Сценария

Для данной работы выбран сценарий: **Сервис потоковой передачи музыки**: Эта система позволит управлять артистами, альбомами, песнями, пользователями и пользовательскими плейлистами.

Part 2: Проектирование Базы Данных и Документация

Идентификация Сущностей и Атрибутов:

1. **Артисты(artists)**
2. **Альбомы(albums)**
3. **Песни(songs)**
4. **Пользователи(users)**
5. **Плейлисты(playlists)**

Проектирование Таблиц:

1. **Table Name: users**

**Description: хранит информацию о пользователях**

**Attributes:**

**ID: INTEGER, PK, Not NULL, UNIQUE**

**username: VARCHAR(100), UNIQUE**

**password: VARCHAR(100),**

**email: VARCHAR(100), UNIQUE**

**Constraints:**

**PK\_ID: Primary key (ID)**

**NotNULL\_account: Not NULL (username, password)**

1. **Table Name: playlists**

**Description: хранит информацию о пользователе плейлиста**

**Attributes:**

**ID: INTEGER, PK, Not NULL, UNIQUE**

**user\_ID: INTEGER, FK(REFERENCES users), Not NULL**

**Constraints:**

**PK\_ID: Primary key (ID)**

**FK\_user\_ID: FOREIGN KEY (user\_ID) REFERENCES users(ID)**

1. **Table Name: songs**

**Description: хранит информацию о песнях**

**Attributes:**

**ID: INTEGER, PK, Not NULL, UNIQUE**

**album\_ID: INTEGER, FK(REFERENCES albums)**

**title: VARCHAR(100), Not NULL**

**text: TEXT, UNIQUE**

**Constraints:**

**PK\_ID: Primary key (ID)**

**FK\_album\_ID: FOREIGN KEY (album\_ID) REFERENCES album(ID)**

1. **Table Name: albums**

**Description: хранит информацию о названии альбома**

**Attributes:**

**ID: INTEGER, PK, UNIQUE**

**title: VARCHAR(100)**

**Constraints:**

**PK\_ID: Primary key (ID)**

**NotNULL\_albums: Not NULL (ID, title)**

1. **Table Name: history**

**Description: хранит историю прослушиваний**

**Attributes:**

**ID: INTEGER, PK, Not NULL, UNIQUE**

**user\_ID: INTEGER, FK(REFERENCES users), Not NULL**

**song\_ID: INTEGER, FK(REFERENCES songs), Not NULL**

**time: TIMESTAMP, Not NULL**

**Constraints:**

**PK\_ID: Primary key (ID)**

**FK\_user\_ID: FOREIGN KEY (user\_ID) REFERENCES users(ID)**

**FK\_song\_ID: FOREIGN KEY (song\_ID) REFERENCES songs(ID)**

1. **Table Name: artists**

**Description: хранит псевдонимы артистов**

**Attributes:**

**ID: INTEGER, PK, Not NULL, UNIQUE**

**FirstName: VARCHAR(100)**

**LastName: VARCHAR(100)**

**Constraints:**

**PK\_ID: Primary key (ID)**

**NotNULL\_artists: Not NULL (FirstName, LastName)**

1. **Table Name: playlists\_songs**

**Description: хранит информацию о песнях в плейлисте**

**Attributes:**

**ID: INTEGER, PK, Not NULL, UNIQUE**

**playlist\_ID: INTEGER, FK(REFERENCES playlists), Not NULL**

**song\_ID: INTEGER, FK(REFERENCES songs), Not NULL**

**Constraints:**

**PK\_ID: Primary key (ID)**

**FK\_user\_ID: FOREIGN KEY (playlist\_ID) REFERENCES playlists(ID)**

**FK\_song\_ID: FOREIGN KEY (song\_ID) REFERENCES songs(ID)**

1. **Table Name: albums\_artists**

**Description: хранит информацию о исполнителях альбома**

**Attributes:**

**ID: INTEGER, PK, Not NULL, UNIQUE**

**album\_ID: INTEGER, FK(REFERENCES albums), Not NULL**

**artist\_ID: INTEGER, FK(REFERENCES artists), Not NULL**

**Constraints:**

**PK\_ID: Primary key (ID)**

**FK\_album\_ID: FOREIGN KEY (album\_ID) REFERENCES albums(ID)**

**FK\_artist\_ID: FOREIGN KEY (artist\_ID) REFERENCES artists(ID)**

1. **Table Name: artists\_songs**

**Description: хранит информацию о исполнителях песни**

**Attributes:**

**song\_ID: INTEGER, FK(REFERENCES songs), Not NULL**

**artist\_ID: INTEGER, FK(REFERENCES artists), Not NULL**

**Constraints:**

**PK\_ID: Primary key (ID)**

**FK\_song\_ID: FOREIGN KEY (song\_ID) REFERENCES songs(ID)**

**FK\_artist\_ID: FOREIGN KEY (artist\_ID) REFERENCES artists(ID)**

**Взаимосвязи:**

* **users и playlists (один ко многим): Одному пользователю может соответствовать несколько плейлистов но плейлисту соответствует только один пользователь.**

**playlist.user\_ID является внешним ключом ссылающимся на users.ID**

* **songs и albums (Один ко многим): Одна песня может присутствовать только в одном альбоме, а с другой стороны в одном альбоме может присутствовать множество песен**

**songs.album\_ID является внешним ключом ссылающимся на albums.ID**

* **users и songs (многие ко многим): Один пользователь мог прослушать много песен как и одна песня могла быть прослушана многими пользователями.**

**history.song\_ID является внешним ключом ссылающимся на songs.ID**

**history.user\_ID является внешним ключом ссылающимся на user.ID**

* **playlists и songs (Многие ко многим): один плейлист может содержать множество песен так и одна песня может содержаться во множестве плейлистов**

**playlists\_songs.playlist\_ID является внешним ключом ссылающимся на playlist.ID**

**playlists\_songs.songs\_ID Является внешним ключом ссылающимся на songs.ID**

* **songs и artists (многие ко многим): как у одной песни может быть множество исполнителей так и у одного исполнителя может быть множество песен**

**songs\_artists.artists\_ID является внешним ключом ссылающимся на artists.ID**

**songs\_artists.songs\_ID является внешним ключом ссылающимся на songs.ID**

* **albums и artists (Многие ко многим): Как каждому альбому может соответствовать разное количество артистов так и артисту может соответствовать некоторое количество альбомов**

**albums\_artists.albums\_ID является внешним ключом ссылающимся на albums.ID**

**albums\_artists.artists\_ID является внешним ключом ссылающимся на artists.ID**

**Часть 3: ER-диаграмма**

**Изображение выглядит как снимок экрана, текст, диаграмма, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.**