**Дахненко Михаил**

**Part 1:** Выбор Сценария

Для данной работы выбран сценарий: Сервис потоковой передачи музыки. Эта система будет 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):

5. Пользовательские плейлисты (PlaylistSongs):

**Проектирование Таблиц:**

1. Table Name: Artists

○ Description: Таблица хранит расширенную информацию об исполнителях и музыкальных группах, представленных в сервисе.  
Содержит как базовые сведения (имя, страна происхождения), так и дополнительные поля для описания творческого пути — краткая биография и год начала карьеры.

○ Attributes:

■ ArtistID: INTEGER, PK, NOT NULL, UNIQUE

■ FirstName: VARCHAR(100), NOT NULL, UNIQUE

■ LastName: VARCHAR(100), NOT NULL, UNIQUE

■ Country: VARCHAR(100)

■ Biography: TEXT

■ CareerStartYear: SMALLINT

○ Constraints:

■ PK\_Artists: PRIMARY KEY (ArtistID)

■ UQ\_ArtistFullName: UNIQUE (FirstName, LastName)

■ CHK\_CareerStartYear: CHECK (CareerStartYear >= 1900 AND CareerStartYear <= EXTRACT(YEAR FROM CURRENT\_DATE))

2. Table Name: Albums

○ Description: Таблица содержит сведения о музыкальных альбомах, выпущенных артистами.  
Включает название альбома, дату его релиза и связь с артистом через внешний ключ.

○ Attributes:

■ AlbumID: INTEGER, PK, NOT NULL, UNIQUE

■ Title: VARCHAR(150), NOT NULL

■ ReleaseDate: DATE, NOT NULL

■ ArtistID: INTEGER, FK (REFERENCES Artists), NOT NULL

○ Constraints:

■ PK\_Albums: PRIMARY KEY (AlbumID)

■ UQ\_AlbumTitleArtist: UNIQUE (Title, ArtistID)

■ CHK\_ReleaseDate: CHECK (ReleaseDate >= ‘1900-01-01’ AND

ReleaseDate <= CURRENT\_DATE)

■ FK\_Albums\_Artists: FOREIGN KEY (AuthorID) REFERENCES

Artists(ArtistID)

3. Table Name: Songs

○ Description: Таблица хранит список всех песен, доступных в сервисе.  
Каждая песня связана с конкретным альбомом, содержит название и длительность в удобном временном формате.

○ Attributes:

■ SongID: INTEGER, PK, NOT NULL, UNIQUE

■ Title: VARCHAR(150), NOT NULL

■ Duration: INTEGER, NOT NULL, CHECK (Duration > INTERVAL ‘0 seconds’)

■ AlbumID: INTEGER, FK (REFERENCES Albums), NOT NULL

○ Constraints:

■ PK\_SONGS: PRIMARY KEY (SongID)

■ UQ\_SongTitleAlbum: UNIQUE (Title, AlbumID)

■ FK\_Songs\_Albums: FOREIGN KEY (AlbumID) REFERENCES Albums (AlbumID)

4. Table Name: Users

○ Description: Содержит сведения о зарегистрированных пользователях сервиса. Включает имя, фамилию, уникальный email и дату регистрации.  
Email проходит проверку формата через регулярное выражение.

○ Attributes:

■ UserID: INTEGER, PK, NOT NULL, UNIQUE

■ FirstName: VARCHAR(100), NOT NULL

■ LastName: VARCHAR(100), NOT NULL

■ Email: VARCHAR(150), UNIQUE, NOT NULL

■ RegistrationDate: DATE NOT NULL DEFAULT CURRENT\_DATE

○ Constraints:

■ PK\_Users: PRIMARY KEY (UserID)

■ UQ\_Email: UNIQUE (Email)

5. Table Name: Playlists

○ Description: Хранит список плейлистов, созданных пользователями сервиса.  
Каждый плейлист связан с одним владельцем, имеет название и дату создания.  
Поддерживает флаг публичности, который позволяет делиться плейлистами с другими пользователями.

○ Attributes:

■ PlaylistID: INTEGER, PK, NOT NULL, UNIQUE

■ Title: VARCHAR(150), NOT NULL

■ UserID: INTEGER, FK (REFERENCES Users), NOT NULL

■ CreationDate: DATE NOT NULL DEFAULT CURRENT\_DATE

■ IsPublic: BOOLEAN NOT NULL DEFAULT FALSE

○ Constraints:

■ PK\_Playlists: PRIMARY KEY (PlaylistID)

■ UQ\_PlaylistTitleUser: UNIQUE (Title, UserID)

■FK\_Playlists\_Users: FOREIGN KEY (UserID) REFERENCES Users (UserID)

6. Table Name: PlaylistSongs

○ Description: Таблица связывает песни и плейлисты, реализуя отношение "многие ко многим". Позволяет хранить дату добавления каждой песни в плейлист.

○ Attributes:

■ PlaylistID: INTEGER, FK (REFERENCES Playlists), NOT NULL

■ SongID: INTEGER, FK (REFERENCES Songs), NOT NULL

■ AddedDate: DATE NOT NULL DEFAULT CURRENT\_DATE

○ Constraints:

■ PK\_PlaylistSongs: PRIMARY KEY (PlaylistSongID)

■ FK\_PlaylistSongs\_Playlists: FOREIGN KEY (PlaylistID) REFERENCES Playlists(PlaylistID)

■ FK\_PlaylistSongs\_Songs: FOREIGN KEY (SongID) REFERENCES Songs(SongID)

■ UQ\_Playlist\_Song: UNIQUE (PlaylistID, SongID) -- чтобы песня не повторялась в одном плейлисте

**Взаимосвязи:**

● Artists и Albums (Многие-ко-Многим):  Как у альбома может быть несколько артистов, так и у артиста может быть несколько альбомов.

● Albums и Songs (Один-ко-Многим): Один альбом содержит несколько песен.

● Users и Playlists (Многие-ко-Многим): У одного плейлиста может быть несколько пользователей

● Playlists и Songs (Многие-ко-Многим): Один плейлист может содержать много песен, а одна песня может входить в несколько плейлистов.

**Part 3: ER-Диаграмма**

