Song Streaming Service

Student Name Ziad Alzarka

Student ID 1618120180100326

Level Two

Professor Dr Reda

Table of Contents

System Description

The system allows users to search through a music library, listen to songs, create playlists and add songs to them.

Data Dictionaries

User

Field Name	Data Type	Description	Example
id	int		
name	varchar		
email	varchar		
password	varchar	Encrypted hash	$2y$12$R87aZr6f8p8wpQLeLCCLqePpU1lMN.nedxodQbrG1sS9dVilTTxeC
is_admin	boolean	If the user has the right to manage the website	True
preferences	text (optional)	Encoded settings object	{“default_playlist”:3}
remember_token	varchar (optional)	Whether to remember user login or not	True
created_at	datetime		
updated_at	datetime		

Playlist

Field Name	Data Type	Description	Example

id	int		
user_id	int	Owner user id	
name	varchar		
rules	text (optional)	Playlist rules	{“public”: true}
created_at	datetime		
updated_at	datetime		

Interactions

The interactions table is used to store user interactions to collect later for analysis.

Field Name	Data Type	Description	Example
id	int		
user_id	int	The user interacting	
song_id	varchar	The song they are interacting with	
liked	boolean	Whether the song is liked by the user or not	True
play_count	int	How many times the song was played	
created_at	datetime		
updated_at	datetime		

Playlist Song

This table is a many-to-many relationship table between playlists and songs.

Field Name	Data Type	Description	Example
id	int		

playlist_id	int	The playlist the song belongs to	
song_id	varchar	The song they are interacting with	

Song

Field Name	Data Type	Description	Example
id	varchar		
album_id	int	The album the song belongs to	
artist_id	int	The artist who released the song	
title	boolean	Whether the song is liked by the user or not	
length	int	Song length	
track	int (optional)	Track number on the album	
disc	int	Disc number on the album	
lyrics	text		
path	text	Where the song is stored on the server	
mtime	int	Modification time	
created_at	datetime		
updated_at	datetime		

Artist

Field Name	Data Type	Description	Example
id	int		
name	varchar		
image	varchar (optional)	Artist image url	
created_at	datetime		
updated_at	datetime		

Album

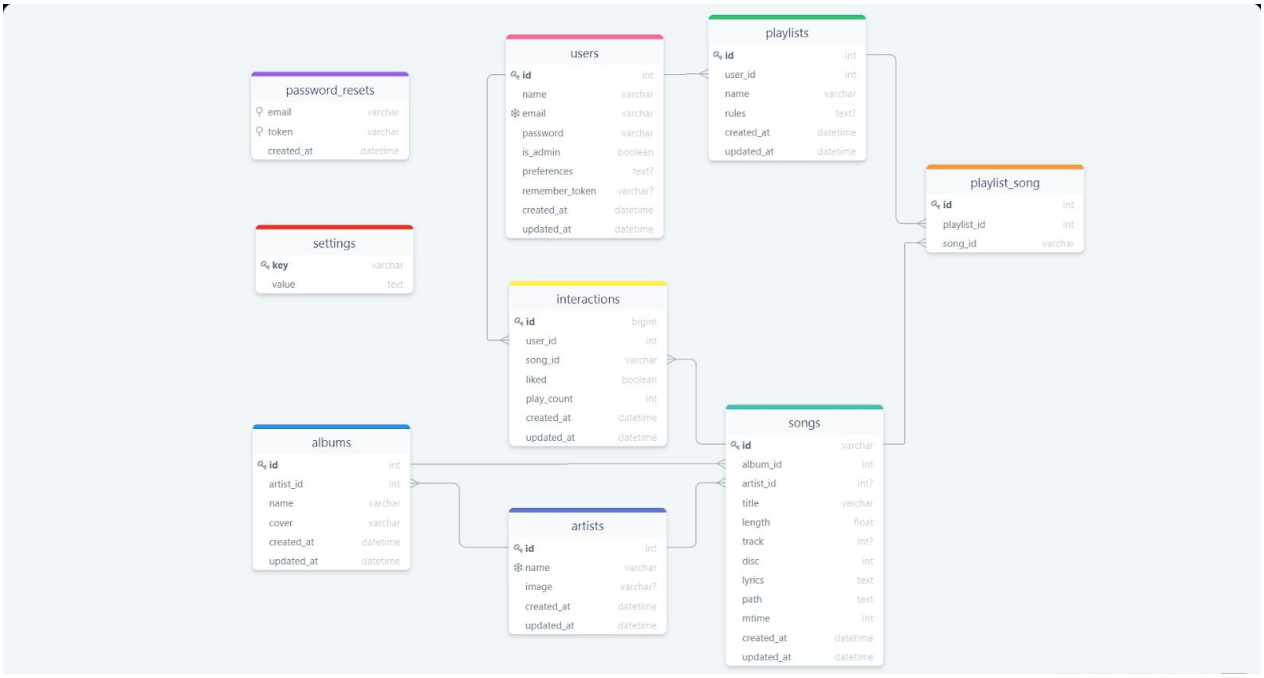
Field Name	Data Type	Description	Example
id	int		
artist_id	int		
name	varchar		
cover	varchar	Album cover image url	
created_at	datetime		
updated_at	datetime		

Settings

Field Name	Data Type	Description	Example
id	int		
key	varchar		
value	text		

Entity Relationship Diagram

This diagram explains the relationships between every field of a table in the database.



Select Statements using Different Functions

How many songs does this playlist have?

```
SELECT COUNT(*) FROM playlist_song WHERE playlist_id=<playlist_id>
```

What many times has this song been played?

```
SELECT SUM(play_count) FROM interactions WHERE user_id=<user_id> AND
song_id=<song_id>
```

What is the last album by this artist?

```
SELECT LAST(*) FROM albums WHERE artist_id=<artist_id>
```

What are my playlists?

```
SELECT * FROM playlists WHERE user_id=<user_id>
```

Select statements using Subquery

Who is the user whom this password reset token belongs to?

```
SELECT * FORM users WHERE email = (SELECT email FROM password_resets
WHERE token=<token>)
```

What artists have songs that are shorter than a minute?

```
SELECT * FROM artists WHERE artist_id IN (SELECT artist_id FROM songs
WHERE length <= 60)
```

How many times have our admins played songs?

```
SELECT SUM(play_count) FROM interactions WHERE user_id IN (SELECT id
FROM users WHERE is_admin = TRUE)
```

Select statements using Count and Group functions

How many times does each user play songs?

```
SELECT user_id, SUM(play_count) FROM interactions GROUP BY user_id
```

How many songs has this artist released?

```
SELECT COUNT(*) FROM songs WHERE artist_id=<artist_id>
```

Select statements using Different Joins

What are my favorite songs?

```
SELECT * FROM interactions, songs INNER JOIN songs ON
interactions.song_id = songs.id WHERE user_id=<user_id>
```

What songs does this playlist have?

```
SELECT * FROM playlist_song, songs INNER JOIN songs ON
playlist_song.song_id = songs.id WHERE playlist_id=<playlist_id>
```

List albums with who released them:

```
SELECT * FROM albums, artists LEFT JOIN artists ON albums.artist_id =
artists.id
```

Insert Statements

Create playlist

```
INSERT INTO playlists (user_id, name) VALUES (<user_id>, <name>)
```

Create artist

```
INSERT INTO artists (name, image) VALUES (<name>, <image>)
```

Create a new settings variable

```
INSERT INTO settings (key, value) VALUES (<key>, <value>)
```

Create new album

```
INSERT INTO albums (artist_id, name, cover) VALUES (<artist_id>, <name>, <cover>)
```

Create new password reset

```
INSERT INTO password_resets (email, token) VALUES (<email>, <token>)
```

Update Statements

Update existing password reset to null

```
UPDATE password_resets SET token = NULL WHERE email=<email>
```

Update song play count by user

```
UPDATE interactions SET play_count = (SELECT play_count FROM interactions WHERE song_id=<song_id> and user_id=<user_id>) + 1 WHERE song_id=<song_id> AND user_id=<user_id>
```

Update user name

```
UPDATE users SET name=<name> WHERE id=<user_id>
```

Update a settings key variable

```
UPDATE settings SET value=<value> WHERE key=<key>
```

Update album name

```
UPDATE albums SET name=<name> WHERE id=<album_id>
```

Delete Statements

Delete a user

```
DELETE FROM users WHERE user_id=<user_id>
```

Delete an album

```
DELETE FROM songs WHERE album_id=<album_id>
DELETE FROM albums WHERE id=<album_id>
```

Delete a reset token

```
DELETE FROM password_resets WHERE email=<email>
```

Delete a song from a playlist

```
DELETE FROM playlist_song WHERE playlist_id=<playlist_id> AND
song_id=<song_id>
```

Delete a song from album

```
DELETE FROM songs WHERE album_id=<album_id> AND song_id=<song_id>
```

References

- What is a data dictionary?
  https://www.tutorialspoint.com/What-is-Data-Dictionary
- What is an entity relationship diagram?
  https://www.smartdraw.com/entity-relationship-diagram/
- SQL Reference from W3Schools
  https://www.w3schools.com/sql/sql_ref_keywords.asp

GitHub Repository Link

GitHub Repository