INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO

ALUNO: Douglas Enrico Loureiro (CJ3025721)

PROFESSOR: Paulo Giovani de Faria Zeferino

CURSO: Análise e Desenvolvimento de Sistemas

SISTEMA DE BANCO DE DADOS PARA REDE SOCIAL DE MÚSICAS

CAMPOS DO JORDÃO 2025

RESUMO

Com o avanço das plataformas digitais, redes sociais de música se tornaram populares por permitir que os usuários compartilhem, descubram e cataloguem suas experiências musicais. O presente trabalho propõe-se a desenvolver um modelo conceitual e físico do banco de dados de um sistema de gerenciamento para uma rede social de música, focado na coleta e análise de dados de reprodução musical, bem como na interação social entre os usuários. O sistema será desenvolvido baseado no estudo da interface de programação de aplicativos (API) de redes sociais/aplicativos com propósitos semelhantes ao tema do projeto, como Spotify e Last.Fm, sendo uma pesquisa bibliográfica, mediante a revisão de estudos e artigos científicos realizados sobre o tema.

Palavras-Chave: banco de dados; música; aplicativos; rede social.

ABSTRACT

With the advancement of digital platforms, social music networks have become popular for allowing users to share, discover and catalog their musical experiences. The present work proposes to develop a conceptual and physical database model of a management system for a social music network, focused on the collection and analysis of music reproduction data, as well as social interaction between users. The system will be developed based on the study of the application programming interface (API) of social networks/applications with purposes similar to the project theme, such as Spotify and Last.Fm, being a bibliographical research, through the review of studies and scientific articles carried out on the topic.

Keywords: database; music; applications; social media.

SUMÁRIO

1	**INTRODUÇÃO**	**3**
1.1	Objetivos	3
1.2	Justificativa	4
1.3	Aspectos Metodológicos	4
1.4	Aporte Teórico	4
2	**METODOLOGIA**	**5**
2.1	Ferramentas Utilizadas	6
2.2	Descrição do Projeto de Dados	6
2.3	Coleta das Regras de Negócio	6
3	**RESULTADOS OBTIDOS**	**7**
3.1	Modelo Conceitual (Diagrama de Heuser)	7
3.2	Regras de Negócio	7
3.3	Dicionário de Dados das Tabelas	8
3.4	Modelo Físico (código SQL)	9
3.5	Inserção de Dados	10
3.6	Consultas	11
4	**CONSIDERAÇÕES FINAIS**	**13**
5	**REFERÊNCIAS**	**13**

1 INTRODUÇÃO

Com a ascensão das plataformas digitais, redes sociais dedicadas à música

têm desempenhado um papel significativo na forma como os usuários consomem, compartilham e descobrem novas experiências musicais. Serviços como Last.fm e Spotify exemplificam essa tendência, oferecendo aos usuários não apenas um registro de suas preferências musicais, mas também um espaço interativo para trocas sociais e personalização de recomendações. Esses sistemas, alimentados por tecnologias de banco de dados robustas, se tornaram ferramentas essenciais no cenário da música digital, conectando milhões de pessoas a conteúdos personalizados e relevantes.

A proposta busca integrar funcionalidades como registro de músicas ouvidas, recomendação personalizada e a criação de um espaço para interação entre os membros da plataforma, permitindo o compartilhamento de playlists, comentários e rankings de popularidade musical.

1.2 Objetivos

Este trabalho tem por objetivo elaborar um projeto de banco de dados, com um modelo conceitual e físico, que suporte as operações do sistema, garantindo a eficiência no armazenamento e recuperação de dados, além de atender aos requisitos de escalabilidade e integração de dados externos, como players de música e serviços de streaming.

1.2 Justificativa

A organização eficiente de grandes volumes de informações é essencial para

oferecer uma experiência rica e interativa aos usuários, ao mesmo tempo que facilita o desenvolvimento de recursos avançados, como algoritmos de recomendação. Além disso, a compreensão de como esses sistemas operam contribui academicamente para os estudos em modelagem de banco de dados e desenvolvimento de aplicações orientadas a dados.

1.3 Aspectos Metodológicos

O presente estudo baseia-se em uma pesquisa bibliográfica, com revisão de estudos e artigos científicos sobre o tema, além de análises de APIs de redes sociais musicais populares, como Last.fm e Spotify. A partir desse levantamento, será realizado o levantamento dos requisitos funcionais e não funcionais do sistema, seguido pela construção do modelo conceitual utilizando diagramas ER (Entidade Relacionamento) e o desenvolvimento físico em comandos SQL.

1.4 Aporte Teórico

Este trabalho engloba conceitos de modelagem de dados, teoria de bancos de dados relacionais e aplicação prática de comandos SQL. O cenário proposto busca simular uma rede social que integra aspectos de registro musical automatizado, interação social e análise de dados, proporcionando insights sobre o comportamento do usuário e promovendo a descoberta musical personalizada.

2 METODOLOGIA

2.1 Ferramentas Utilizadas

O desenvolvimento do projeto conceitual foi conduzido com o uso da ferramenta

draw.io, amplamente empregada para criação de diagramas, permitindo a elaboração de um modelo de dados no padrão de Heuser. O projeto físico foi desenvolvido com o uso do SQL Management Studio 20, sendo fundamental para criar o banco de dados, as tabelas e as consultas utilizando o MSSQL.

2.2 Descrição do Projeto de Dados

O projeto de dados foi estruturado com base na análise das necessidades funcionais de uma rede social voltada ao compartilhamento de gostos musicais e estatísticas pessoais. Durante o planejamento, cinco entidades principais foram identificadas:

- **Usuário:** Representa os participantes da rede social, responsáveis por interagir com o sistema e compartilhar informações. Usuários podem interagir entre si e registrar músicas no sistema ao ouvir.
- **Músicas:** Refere-se às faixas musicais disponíveis, associadas a álbuns e artistas. As músicas são registradas com a interação dos usuários. Também é armazenado a duração, popularidade e se a música possui conteúdo explícito nas letras.
- **Álbuns:** Agrupam coleções de músicas relacionadas a um artista específico.
- **Artistas:** Representa os criadores das músicas, podendo ser associados a múltiplos gêneros musicais.
- **Gêneros:** Classificação estilística das músicas e artistas, permitindo categorizar as preferências musicais dos usuários.

2.3 Coleta das Regras de Negócio

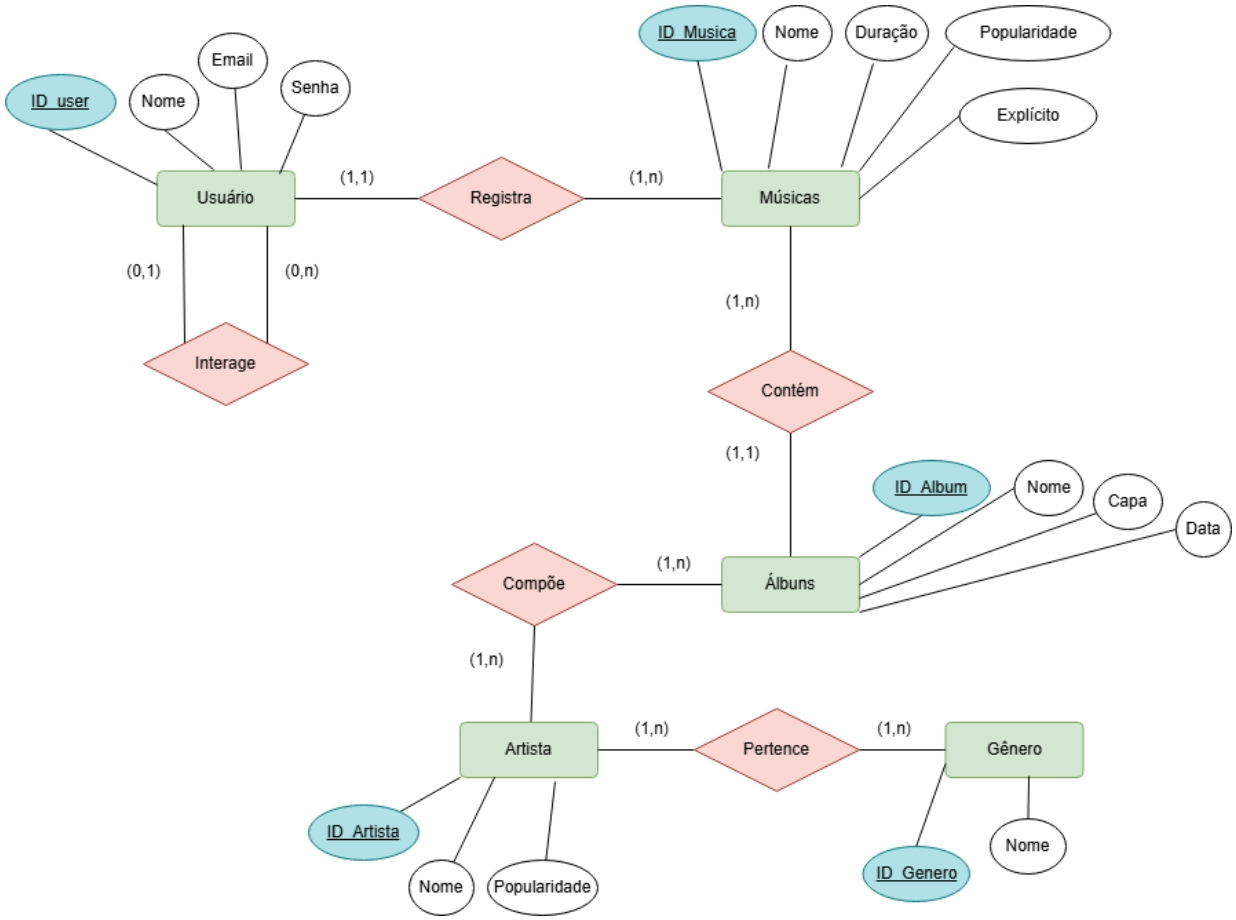
A coleta das regras de negócio foi realizada com base em uma pesquisa bibliográfica e em análise comparativa de sistemas semelhantes, como Last.fm e

Spotify, que possuem funcionalidades alinhadas ao escopo deste projeto. Além disso, foram consideradas as boas práticas de modelagem de sistemas de gerenciamento de informações.

- **Cadastro de Usuários:** Cada usuário deve possuir um identificador único, nome, email e senha. Um usuário pode associar-se a múltiplas músicas, álbuns e gêneros favoritos.
- **Relação entre Músicas, Álbuns e Artistas:** Uma música pertence a um álbum, que por sua vez é associado a um artista. Um artista pode criar múltiplos álbuns e músicas.
- **Interação Social:** Usuários podem compartilhar suas músicas, álbuns ou artistas favoritos com outros usuários, promovendo a interação na rede.

3 RESULTADOS OBTIDOS

3.1 Modelo Conceitual (Diagrama de Heuser):



3.2 Regras de Negócio:

• Cada usuário deve possuir um identificador único, juntamente com informações de autenticação, como nome, email e senha.

• Um usuário pode registrar múltiplas músicas como favoritas.

• Uma música pertence a exatamente um álbum, mas um álbum pode conter várias músicas.

• Músicas e artistas são associados a pelo menos um gênero musical.

• Usuários podem interagir entre si, compartilhando gostos e interesses musicais.

3.3 Dicionário de Dados das Tabelas:

Tabela	Coluna	Tipo	Descrição

Usuario	id_usuario	INT (PK)	Identificador do usuário
	nome	NVARCHAR(100)	Nome do usuário
	email	NVARCHAR(100)	E-mail único
	senha	NVARCHAR(100)	Senha da conta
Artista	id_artista	INT (PK)	Identificador do artista
	nome_artista	NVARCHAR(100)	Nome do artista
Album	id_album	INT (PK)	Identificador do álbum
	titulo_album	NVARCHAR(200)	Título do álbum
	id_artista	INT (FK)	Artista do álbum
	ano_lancamento	INT	Ano de lançamento
Musica	id_musica	INT (PK)	Identificador da música
	titulo	NVARCHAR(200)	Título da música
	duracao	INT	Duração em segundos
	popularidade	INT (0-100)	Escala de popularidade
	explicita	BIT	Indica se a música é explícita
	id_album	INT (FK)	Álbum ao qual pertence
	ano_lancamento	INT	Ano da música
Genero	id_genero	INT (PK)	Identificador do gênero
	nome_genero	NVARCHAR(50)	Nome do gênero
Musica_Genero	id_musica, id_genero	INT (PK, FK)	Associa música a gêneros
Usuario_Musica	id_usuario_musica	INT (PK)	Identificador da escuta
	id_usuario	INT (FK)	Quem escutou
	id_musica	INT (FK)	Música escutada
	data_registro	DATETIME	Data/hora da escuta
Usuario_Genero	id_usuario, id_genero	INT (PK, FK)	Preferência do usuário

3.4 Modelo Físico (MSSQL)

Código SQL para a criação das tabelas e do banco de dados do projeto:

```
CREATE TABLE Usuario (
    id_usuario INT PRIMARY KEY IDENTITY(1,1),
    nome NVARCHAR(100), email NVARCHAR(100) UNIQUE, senha NVARCHAR(100)
);

CREATE TABLE Artista (
    id_artista INT PRIMARY KEY IDENTITY(1,1),
    nome_artista NVARCHAR(100)
);

CREATE TABLE Album (
    id_album INT PRIMARY KEY IDENTITY(1,1),
    titulo_album NVARCHAR(200), id_artista INT, ano_lancamento INT,
    FOREIGN KEY (id_artista) REFERENCES Artista(id_artista)
);

CREATE TABLE Musica (
    id_musica INT PRIMARY KEY IDENTITY(1,1),
    titulo NVARCHAR(200), duracao INT, popularidade INT CHECK (popularidade BETWEEN 0 AND 100),
    explicita BIT, id_album INT, ano_lancamento INT,
    FOREIGN KEY (id_album) REFERENCES Album(id_album)
);

CREATE TABLE Genero (
    id_genero INT PRIMARY KEY IDENTITY(1,1),
    nome_genero NVARCHAR(50)
);

CREATE TABLE Musica_Genero (
    id_musica INT, id_genero INT,
    PRIMARY KEY (id_musica, id_genero),
    FOREIGN KEY (id_musica) REFERENCES Musica(id_musica),
    FOREIGN KEY (id_genero) REFERENCES Genero(id_genero)
);

CREATE TABLE Usuario_Musica (
    id_usuario_musica INT PRIMARY KEY IDENTITY(1,1),
    id_usuario INT, id_musica INT, data_registro DATETIME DEFAULT GETDATE(),
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario),
    FOREIGN KEY (id_musica) REFERENCES Musica(id_musica)
);

CREATE TABLE Usuario_Genero (
    id_usuario INT, id_genero INT,
    PRIMARY KEY (id_usuario, id_genero),
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario),
    FOREIGN KEY (id_genero) REFERENCES Genero(id_genero)
);
```

3.5 Inserção de dados em tabelas

```
-- Usuários
INSERT INTO Usuario (nome, email, senha) VALUES
('Fred Guedes', 'fred@ifsp.edu.br', 'senha123'),
('Heihachi Mishima', 'mishima@ifsp.edu.br', 'senha123'),
('Ana Julia', 'ana.julia@email.com', 'senha456'),
('Felipe Alves', 'felipe.alves@email.com', 'abc123'),

-- Artistas
INSERT INTO Artista (nome_artista) VALUES
('The Strokes'),
('Michael Jackson'),
('MF DOOM'),
('Tim Maia');

-- Álbuns (com id_artista correspondente)
INSERT INTO Album (titulo_album, id_artista, ano_lancamento) VALUES
('Is This It', 1, 2001),
('Thriller', 2, 1982),
('Madvillainy', 3, 2004),
('Tim Maia 1973', 4, 1973);

-- Músicas (relacionadas a álbuns e artistas)
INSERT INTO Musica (titulo, duracao, popularidade, explicita, id_album, ano_lancamento) VALUES
-- The Strokes (Rock)
('Last Nite', 180, 88, 0, 1, 2001),
('Hard To Explain', 224, 78, 0, 1, 2001),
('Soma', 169, 82, 0, 1, 2001),
('Barely Legal', 193, 80, 0, 1, 2001),
-- Michael Jackson (Pop)
('Billie Jean', 294, 100, 0, 2, 1982),
('Thriller', 294, 100, 0, 2, 1982),
('Beat It', 258, 95, 0, 2, 1982),
('Wanna Be Startin'' Somethin'", 362, 90, 0, 2, 1982),
-- MF DOOM (Rap)
('ALL CAPS', 130, 74, 1, 3, 2004),
('Accordion', 130, 75, 0, 3, 2004),
('Rhinestone Cowboy', 200, 70, 0, 3, 2004),
-- Tim Maia (MPB)
('Gostava Tanto de Você', 252, 73, 0, 4, 1973),
('Réu Confesso', 215, 76, 0, 4, 1973),
('Não Quero Dinheiro', 185, 80, 0, 4, 1973);

-- Gêneros musicais
INSERT INTO Genero (nome_genero) VALUES
('Rock'),
('Pop'),
('Rap'),
('MPB');

-- Associação entre músicas e gêneros
-- Rock (id_genero = 1)
INSERT INTO Musica_Genero VALUES (1,1), (2,1), (3,1), (4,1);
-- Pop (id_genero = 2)
INSERT INTO Musica_Genero VALUES (5,2), (6,2), (7,2), (8,2);
-- Rap (id_genero = 3)
INSERT INTO Musica_Genero VALUES (9,3), (10,3), (11,3);
-- MPB (id_genero = 4)
INSERT INTO Musica_Genero VALUES (12,4), (13,4), (14,4);

-- Músicas escutadas/favoritadas por usuários
-- Fred Guedes
INSERT INTO Usuario_Musica (id_usuario, id_musica) VALUES
(1, 1), (1, 1), (1, 2), (1, 5), (1, 9);
-- Heihachi
INSERT INTO Usuario_Musica (id_usuario, id_musica) VALUES
(2, 9), (2, 9), (2, 10), (2, 11);
-- Ana Julia
INSERT INTO Usuario_Musica (id_usuario, id_musica) VALUES
(3, 12), (3, 13), (3, 14), (3, 5), (3, 6);
-- Felipe
INSERT INTO Usuario_Musica (id_usuario, id_musica) VALUES
(4, 1), (4, 3), (4, 5), (4, 10);
-- Gêneros favoritos dos usuários
INSERT INTO Usuario_Genero VALUES
(1, 1), (1, 2),       -- Fred gosta de Rock e Pop
(2, 3),               -- Heihachi gosta de Rap
(3, 2), (3, 4),       -- Ana gosta de Pop e MPB
(4, 1), (4, 3),       -- Felipe gosta de Rock e Rap
```

3.6 Consultas no banco de dados

```
-- 1. Lista as músicas favoritadas por um usuário específico
SELECT m.titulo
FROM Usuario u
JOIN Usuario_Musica um ON u.id_usuario = um.id_usuario
JOIN Musica m ON um.id_musica = m.id_musica
WHERE u.nome = 'Fred Guedes';

-- 2. Mostra o álbum mais escutado por um usuário
SELECT TOP 1 a.titulo_album, COUNT(*) AS total
FROM Usuario u
JOIN Usuario_Musica um ON u.id_usuario = um.id_usuario
JOIN Musica m ON um.id_musica = m.id_musica
JOIN Album a ON m.id_album = a.id_album
WHERE u.nome = 'Fred Guedes'
GROUP BY a.titulo_album
ORDER BY total DESC;

-- 3. Gênero mais escutado por um usuário
SELECT TOP 1 g.nome_genero, COUNT(*) AS total
FROM Usuario u
JOIN Usuario_Musica um ON u.id_usuario = um.id_usuario
JOIN Musica m ON um.id_musica = m.id_musica
JOIN Musica_Genero mg ON m.id_musica = mg.id_musica
JOIN Genero g ON mg.id_genero = g.id_genero
WHERE u.nome = 'Fred Guedes'
GROUP BY g.nome_genero
ORDER BY total DESC;

-- 4. Encontra usuários com músicas favoritas semelhantes a um usuário
SELECT u2.nome, COUNT(*) AS em_comum
FROM Usuario u1
JOIN Usuario_Musica um1 ON u1.id_usuario = um1.id_usuario
JOIN Usuario_Musica um2 ON um1.id_musica = um2.id_musica
JOIN Usuario u2 ON um2.id_usuario = u2.id_usuario
WHERE u1.nome = 'Fred Guedes' AND u2.id_usuario <> u1.id_usuario
GROUP BY u2.nome
ORDER BY em_comum DESC;

-- 5. Mostra as 5 músicas mais ouvidas no sistema
SELECT TOP 5 m.titulo, COUNT(*) AS total
FROM Usuario_Musica um
JOIN Musica m ON um.id_musica = m.id_musica
GROUP BY m.titulo
ORDER BY total DESC;

-- 6. Lista os gêneros mais escutados por todos os usuários
SELECT g.nome_genero, COUNT(*) AS total
FROM Usuario_Musica um
JOIN Musica m ON um.id_musica = m.id_musica
JOIN Musica_Genero mg ON m.id_musica = mg.id_musica
JOIN Genero g ON mg.id_genero = g.id_genero
GROUP BY g.nome_genero
ORDER BY total DESC;

-- 7. Lista as músicas explícitas mais escutadas
SELECT m.titulo, COUNT(*) AS total
FROM Musica m
JOIN Usuario_Musica um ON m.id_musica = um.id_musica
WHERE m.explicita = 1
GROUP BY m.titulo
ORDER BY total DESC;

-- 8. Lista os artistas mais escutados na plataforma
SELECT ar.nome_artista, COUNT(*) AS total
FROM Usuario_Musica um
JOIN Musica m ON um.id_musica = m.id_musica
JOIN Album a ON m.id_album = a.id_album
JOIN Artista ar ON a.id_artista = ar.id_artista
GROUP BY ar.nome_artista
ORDER BY total DESC;

-- 9. Top 3 usuários que ouviram mais músicas
SELECT TOP 3 u.nome, COUNT(*) AS total
FROM Usuario u
JOIN Usuario_Musica um ON u.id_usuario = um.id_usuario
GROUP BY u.nome
ORDER BY total DESC;

-- 10. Mostra os gêneros preferidos registrados por cada usuário
SELECT u.nome, STRING_AGG(g.nome_genero, ', ') AS preferencias
FROM Usuario u
JOIN Usuario_Genero ug ON u.id_usuario = ug.id_usuario
JOIN Genero g ON ug.id_genero = g.id_genero
GROUP BY u.nome;
```

```
-- 11. Lista músicas e todos os seus gêneros
SELECT m.titulo, STRING_AGG(g.nome_genero, ', ') AS generos
FROM Musica m
JOIN Musica_Genero mg ON m.id_musica = mg.id_musica
JOIN Genero g ON mg.id_genero = g.id_genero
GROUP BY m.titulo;

-- 12. Mostra qual álbum tem mais músicas cadastradas
SELECT a.titulo_album, COUNT(m.id_musica) AS total
FROM Album a
JOIN Musica m ON a.id_album = m.id_album
GROUP BY a.titulo_album
ORDER BY total DESC;

-- 13. Média de popularidade das músicas por artista
SELECT ar.nome_artista, AVG(m.popularidade) AS media_pop
FROM Artista ar
JOIN Album a ON ar.id_artista = a.id_artista
JOIN Musica m ON m.id_album = a.id_album
GROUP BY ar.nome_artista;

-- 14. Lista músicas que pertencem a mais de um gênero
SELECT m.titulo, COUNT(*) AS qtd_generos
FROM Musica m
JOIN Musica_Genero mg ON m.id_musica = mg.id_musica
GROUP BY m.titulo
HAVING COUNT(*) > 1;

-- 15. Total de músicas favoritadas por dia
SELECT CAST(data_registro AS DATE) AS dia, COUNT(*) AS total
FROM Usuario_Musica
GROUP BY CAST(data_registro AS DATE)
ORDER BY dia DESC;

-- 16. Quantidade de músicas lançadas por ano
SELECT ano_lancamento, COUNT(*) AS total
FROM Musica
GROUP BY ano_lancamento
ORDER BY ano_lancamento;

-- 17. Artistas com mais de um álbum
SELECT ar.nome_artista, COUNT(*) AS qtd_albuns
FROM Artista ar
JOIN Album a ON ar.id_artista = a.id_artista
GROUP BY ar.nome_artista
HAVING COUNT(*) > 1;

-- 18. Músicas escutadas que são do gênero favorito do usuário
SELECT DISTINCT m.titulo
FROM Usuario u
JOIN Usuario_Genero ug ON u.id_usuario = ug.id_usuario
JOIN Musica_Genero mg ON ug.id_genero = mg.id_genero
JOIN Musica m ON mg.id_musica = m.id_musica
WHERE u.nome = 'Fred Guedes';

-- 19. Gêneros mais populares (com base nas execuções)
SELECT g.nome_genero, COUNT(*) AS total_execucoes
FROM Usuario_Musica um
JOIN Musica m ON um.id_musica = m.id_musica
JOIN Musica_Genero mg ON m.id_musica = mg.id_musica
JOIN Genero g ON mg.id_genero = g.id_genero
GROUP BY g.nome_genero
ORDER BY total_execucoes DESC;

-- 20. Artistas com músicas escutadas por usuários
SELECT DISTINCT ar.nome_artista
FROM Usuario_Musica um
JOIN Musica m ON um.id_musica = m.id_musica
JOIN Album a ON m.id_album = a.id_album
JOIN Artista ar ON a.id_artista = ar.id_artista;
```

4 CONSIDERAÇÕES FINAIS

O desenvolvimento do modelo conceitual e físico do banco de dados para a rede social de música inspirado no Last.fm permitiu atender aos objetivos propostos, como o mapeamento eficiente das entidades e suas relações, representando adequadamente os dados e funcionalidades essenciais do sistema. A estrutura proposta foi capaz de representar de forma adequada os principais elementos de uma rede social voltada à música, incluindo o cadastro de usuários, artistas, álbuns, músicas, gêneros e a relação entre esses elementos. Também foi possível implementar mecanismos para registrar interações dos usuários com músicas e seus gêneros favoritos.

As consultas desenvolvidas demonstram a capacidade do banco de fornecer informações relevantes, como identificação de usuários com gostos semelhantes, gêneros e álbuns mais escutados, bem como rankings de popularidade. A flexibilidade da estrutura permite expandir facilmente a aplicação para cenários mais complexos.

5 REFERÊNCIAS

API docs. Disponível em: <https://www.last.fm/api>. Acesso em: 23 nov. 2024.

SIMONYAN, H. System Design Interview Question: Design Spotify. Disponível em: <https://youtu.be/OYtYc98XBZw?si=FwM2LWAoS_YK_D8J>.

SUN, H. Case Study—Spotify. Em: Digital Revolution Tamed. Cham: Springer International Publishing, 2019. p. 135–170.

Web API. Disponível em: <https://developer.spotify.com/documentation/web api>.Acesso em: 23 nov. 2024.