

Gabrielle Oliveira Santana / BP3022676

Ricardo Carriel de Oliveira Filho / BP3022366

**Projeto interdisciplinar**

Seminário: Projeto interdisciplinar

Banco de Dados – Professora Letícia

Bragança Paulista, Junho 2022

Ricardo Carriel -BP3022366

Gabrielle Oliveira Santana - BP3022676

=========================**Universo de discurso**==========================

O aplicativo é sobre uma loja de mídias de **Músicas**, chamado “DiscoDeMúsica” foi implementado visando atender as fases de desenvolvimento do software aplicando os conhecimentos adquiridos com Banco de Dados em sinergia com Engenharia de Software.

O aplicativo de DiscoDeMúsica a fim de gerenciar suas músicas, registra estilo musical,

cliente, preferência musical, cantor e a própria música que é escutada por cada usuário.

Algumas aplicações do usuário são implementadas por meio da tabela “clientes” se referem ao id, nome, e-mail e senha do mesmo, sendo o id exclusivo para cada um deles.

Cada estilo musical é descrito por um id e nome, cabe ressaltar que o id possui valor único para cada estilo musical.

Ademais, cada música é registrada com o id do cliente e id do estilo musical, e possui nome, tempo de duração e data de lançamento. O id da música possui um valor exclusivo para cada música.

Relacionado a música, o sistema registra o id e nome do tipo de cada estilo musical. Tanto o id como o nome possuem valores únicos para cada estilo musical.

Cada cantor é registrado com seu id e nome, sendo seu id exclusivo para cada cantor.

E por fim, cada playlist é criada com um id, nome da playlist, id do cantor e id da música. O valor do id da playlist é exclusivo para cada playlist.

===========================**Notação textual**==============================

Cliente(idCliente, nomeCliente, emailCliente, senha)

estiloMusical(idEstilo, nomeEstilo)

preferenciaMusical(idCliente, idEstilo)

idCliente referencia Cliente

idEstilo referencia estiloMusical

Cantor(idCantor, nomeCantor)

Musica(idMusica, idCantor, nomeMusica, idEstilo, tempoDuracao, dataLancamento)

idCantor referencia Cantor

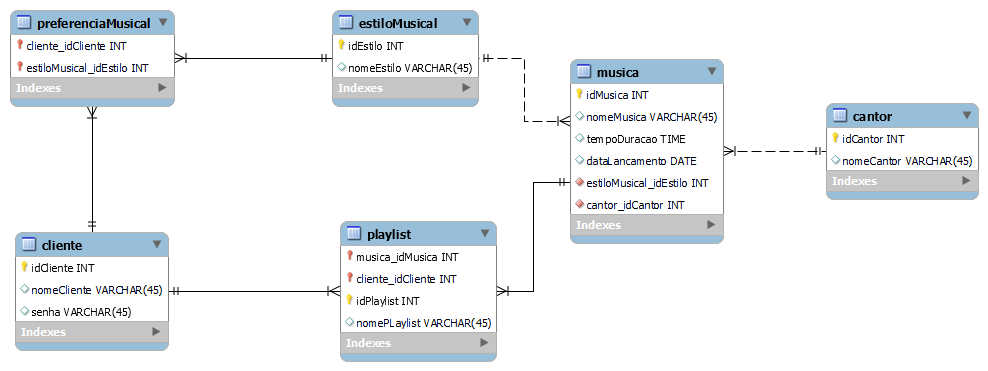
idEstilo referencia estiloMusical

playlist(idPlaylist, nomePlaylist, idCantor, idMusica)

idCantor referencia Cantor

idMusica referencia Musica

=====================**Diagrama Entidade Relacionamento**====================



===============================**SCRIPT**================================

drop database if exists DiscoDeMusica;

create database if not exists DiscoDeMusica;

use DiscoDeMusica;

create table if not exists estiloMusical

(idEstilo int,

nomeEstilo varchar(45),

primary key(idEstilo));

insert into estiloMusical values

(1, 'Rock'),

(2, 'Funk'),

(3, 'Rap'),

(4, 'Pop'),

(5, 'Eletronica'),

(6, 'Pagode'),

(7, 'Samba'),

(8, 'Hard Style'),

(9, 'Kpop');

create table if not exists cliente

(idCliente int,

nomeCliente varchar(45),

emailCLiente varchar(45),

senha varchar(45),

primary key(idCliente));

insert into cliente values

(1, 'Ricardo Carriel', 'ricardo.carriel@aluno,ifsp.edu.br', '12345678'),

(2, 'Kimberly Santis', 'kimberly.santis@aluno.ifsp.edu.br', '87654321'),

(3, 'Gabrielle Santana', 'santana.gabrielle@aluno.ifsp.edu.br', '11223344');

create table if not exists preferenciaMusical

(idCLiente int,

idEstilo int,

foreign key(idEstilo) references estiloMusical(idEstilo),

foreign key (idCLiente) references cliente(idCLiente));

insert into preferenciaMusical values

(1, 2),

(1, 3),

(1, 4),

(1, 5),

(1, 6),

(1, 8),

(3, 4),

(3, 9);

create table if not exists cantor

(idCantor int auto\_increment,

nomeCantor varchar(45),

primary key(idCantor));

insert into cantor values

(1, 'Tevvez'),

(2, 'Roughronny'),

(3, 'Bruno Mars'),

(4, 'Travis Scott'),

(5, 'Péricles'),

(6, 'Alok'),

(7, 'Post Malone'),

(8, 'BTS');

create table if not exists musica

(idMusica int,

idCantor int,

nomeMusica varchar(45),

idEstilo int,

tempoDuracao time,

dataLancamento date,

primary key (idMusica),

foreign key (idCantor) references cantor (idCantor),

foreign key (idEstilo) references estiloMusical(idEstilo));

insert into musica values

(32123, 3, 'Locked out of Heaven', 4, '00:03:52', '2012-10-01'),

(32124, 4, 'Highest in the room', 3, '00:02:55', '2020-10-04'),

(32125, 7, 'Sunflower', 3, '00:02:58', '2018-10-15'),

(32126, 2, 'Outside zyzz edit', 8, '00:02:19', '2022-02-28'),

(32127, 1, 'Legend', 8, '00:03:09', '2020-07-10'),

(33444, 8, 'IDOL' , 9, '00:03:52', '2018-08-24');

create table if not exists playlist(

idPlaylist int,

nomePlaylist varchar(45),

idCliente int,

idMusica int,

foreign key (idCliente) references cliente(idCliente),

foreign key (idMusica) references musica (idMusica)

);

insert into playlist values

(777, 'HARDRAVEE', 1, 32126),

(777, 'HARDRAVEE', 1, 32127),

(656, 'DIMIN', 3, 33444),

(656, 'DIMIN', 3, 32125),

(656, 'DIMIN', 3, 32124);

#1 Mostrar as preferências musicais de cada cliente

select c.nomecliente, g.idestilo, e.nomeestilo from cliente c inner join preferenciaMusical g on c.idcliente = g.idcliente inner join estiloMusical e on g.idestilo = e.idestilo;

#2 Mostrar clientes sem preferência musical registrado

select c.nomecliente as clientes\_sem\_preferencia from cliente c where c.idcliente not in (select idcliente from preferenciaMusical);

#3 Mostrar clientes que gostam de pagode

select c.nomecliente, g.idestilo, e.nomeestilo from cliente c inner join preferenciaMusical g on c.idcliente = g.idcliente inner join estiloMusical e on g.idestilo = e.idestilo where e.idestilo = 6;

#4 Mostrar as músicas de cada cantor

select c.nomecantor, m.nomemusica, tempoduracao from cantor c inner join musica m on c.idcantor = m.idcantor;

#5 Mostrar cantores do estilo Hard Style

select c.nomecantor as cantores\_HardStyle from cantor c inner join musica m on c.idcantor = m.idcantor inner join estiloMusical e on m.idestilo = e.idestilo where m.idestilo = 8;

#6 Mostrar cantores sem músicas cadastradas

select \* from cantor c where c.idcantor not in (select idcantor from musica);

#7 Mostrar músicas com mais de 3 minutos

select \* from musica where tempoduracao > '00:03:00';

#8 Mostrar músicas lancadas antes de 2015

select c.nomecantor, m.nomeMusica, m.datalancamento from musica m inner join cantor c on c.idcantor = m.idcantor where dataLancamento < '2015-01-01' group by nomeMusica;

#9 Mostrar a playlist com mais músicas e seu usuário

select count(p.nomeplaylist) as playlistMaisMusicas, c.nomeCliente from playlist p inner join cliente c on p.idcliente = c.idcliente group by p.nomeplaylist order by p.nomeplaylist limit 1;

#10 Mostrar a musica com o menor tempo de duraçao

select min(tempoDuracao) as menorMusica from musica;

#11 Mostrar as playlist dos usuário, junto com os nomes da música

select c.nomecliente, p.nomeplaylist, m.nomemusica from cliente c inner join playlist p on p.idcliente = c.idcliente inner join musica m on m.idmusica = p.idmusica;

#12 Mostrar a maior música das playlists do Ricardo

select p.idplaylist, p.nomeplaylist, m.nomemusica as maiorMusica, max(m.tempoduracao) from musica m inner join playlist p on m.idmusica = p.idmusica limit 1;