



miro

Descrição Projeto Spotify

TABELAS								
NOME DA TABELA	USER	PLAYS	GENRE	SONGS	ARTIST	PLAYLIST	NEW_PLAYLIST	NEW_UPDATE_USER
DESCRIÇÃO	Tabela onde estão armazenadas as informações do cadastro dos nossos usuários.	Tabela onde serão armazenadas informações sobre a escolha do usuário	Tabela onde estão informações sobre o gênero da música	Tabela onde estão armazenadas as informações sobre as músicas	Tabela onde estão armazenadas as informações sobre os artistas	Tabela onde serão armazenadas informações sobre a playlist do usuário	Tabela onde serão armazenadas informações de atualização dos dados de playlists	Tabela onde serão armazenadas informações de atualização dos dados de usuário
CAMPO 1	USER_ID(INTEGER) NOT NULL AUTO_INCREMENT	User_ID(INTEGER)	GENRE_NAME(VARCHAR 50)	SONG_ID(INTEGER) NOT NULL AUTO_INCREMENT	ARTIST_NAME(VARCHAR 100)	User_ID(INTEGER)	PLAYLIST_ID(INTEGER)	User_ID(INTEGER)
CAMPO 2	USER_NAME(VARCHAR 30)	GENRE_ID (INTEGER)	GENRE_ID(INTEGER) NOT NULL AUTO_INCREMENT	SONG_NAME(VARCHAR 100)	GENRE_ID(INTEGER)	GENRE_ID(INTEGER)	USER_ID(INTEGER)	USER_NAME(VARCHAR 30)
CAMPO 3	USER_EMAIL(VARCHAR 50)	SONG_ID(INTEGER)		DURATION (TIME)	ARTIST_ID(INTEGER) NOT NULL AUTO_INCREMENT	SONGS_ID(INTEGER)	ARTIST_ID(INTEGER)	USER_EMAIL(VARCHAR 50)
CAMPO 4	USER_TYPE(VARCHAR 30)	ARTIST_ID (INTEGER)		GENRE_ID(INTEGER)	SONG_ID(INTEGER)	ARTIST_ID(INTEGER)	SONG_ID(INTEGER)	USER_TYPE(VARCHAR 30)
CAMPO 5	DATE_BIRTH(DATE)	PLAYLIST_ID (INTEGER)		ARTIST_ID (INTEGER)	COUNTRY(VARCHAR 30)	PLAYLIST_NAME(VARCHAR 40)	DATE_BIRTH(DATE)	
CAMPO 6		PLAYS_ID(INTEGER) NOT NULL AUTO_INCREMENT		RECORD_LABEL(VARCHAR 50)		PLAYLIST_ID(INTEGER) NOT NULL AUTO_INCREMENT	PLAYLIST_ID(VARCHAR 40)	REFERENCE_DATE (DATE)
CAMPO 7		Date_play(Date)					REFERENCE_DATE (DATE)	REFERENCE_TIME (TIME)
CAMPO 8		Time_play(Time)					REFERENCE_TIME (TIME)	USER (VARCHAR50)
CAMPO 9							USER (VARCHAR50)	ACTION (VARCHAR50)

CHAVES								
NOME DA TABELA	USER	PLAYS	GENRE	SONGS	ARTIST	PLAYLIST	NEW_PLAYLIST	NEW_UPDATE_USER
CHAVES PRIMARIAS	USER_ID(INTEGER)	Play_ID(INTEGER)	GENRE_ID(INTEGER)	SONG_ID(INTEGER)	ARTIST_ID(INTEGER)	PLAYLIST_ID(INTEGER)		
CHAVES ESTRANGEIRAS		USER_ID, GENRE_ID, SONG_ID, ARTIST_ID, PLAYLIST_ID		GENRE_ID, ARTIST_ID	SONG_ID, GENRE_ID	USER_ID, ARTIST_ID, SONG_ID, GENRE_ID		

VIEWS								
NOME DA VIEW	MUSICAS	GRAVADORAS	GENEROS	CADASTRO_USERS	PLAYLIST_USERS			
OBJETIVO	View que permite visualizar todas as músicas disponíveis e suas informações	View que permite visualizar dados de gravadoras por artistas	View que permite visualizar dados de generos de todas as músicas disponíveis	View que permite visualizar todos os dados dos usuarios cadastrados	View que permite visualizar todas as playlists existentes por usuários			
TABELAS	Tabelas SONGS e ARTISTS	tabelas SONGS e ARTISTS	Tabelas SONGS e GENRE	Tabelas USER	Tabelas USER e PLAYLIST			

FUNCTIONS								
NOME DA FUNÇÃO	GET_GENRE	GET_PLAYED_SONG_NAMES						
OBJETIVO	Esta função permite identificar o nome dos generos através de um genre_ID. O objetivo é permitir que o genre_ID, mais utilizado nas demais tabelas do schema consiga referenciar o nome do genero com mais facilidade	Função permite identificar as músicas ouvidas pelos usuarios.						
COMPOSIÇÃO	Tabela GENRE	Tabelas SONGS e PLAYS						
COMO FUNCIONA?	Basta inserir o GENRE_ID durante o select desta função. Desta forma, o resultado será o nome deste genero.	Basta inserir o USER_ID durante o select desta função. Desta forma, o resultado será a lista de todas as músicas ouvidas por este usuário.						

STORED PROCEDURES								
NOME DA SP	SP_ORDER_TABLE	SP_NEW_USER						
OBJETIVO	Esta procedure permite ordenar qualquer informações das tabelas desde esquema de acordo com a vontade do usuário. O maior ganho aqui é facilidade na gestão das informações por parte desse usuário.	Esta procedure permite inserir novos dados de usuário na tabela de USER. O principal ganho com essa SP é fazer mais facilidade e agilidade na criação de novos usuários para o nosso banco.						
COMPOSIÇÃO	Tabelas USER, PLAYS, GENRE, SONGS, ARTIST, PLAYLIST, NEW_PLAYLIST, NEW_UPDATE_USER	Tabela USER						
COMO FUNCIONA?	Basta inserir o nome da tabela, a coluna a ser ordenada e como ordená-la -- EX ordenando a tabela artists, pelo nome do artista de forma crescente call sp_order_table('artist', 'name' 'ASC').	Basta inserir o novo ID, nome do usuário, tipo de conta e data de nascimento -- EX inserindo um novo usuário Joao						