

Projecto Base de Dados

Relatório Meta 2 (28 Novembro 2018)

Contactos

Nome	Email	Número de Estudante
Alexandre Faria	afaria@student.dei.uc.pt	2014226180
Carlos Poiares	cpoiares@student.dei.uc.pt	2014226236

Table of Contents

1. [Contactos](#)
2. [Instalação](#)
3. [Estrutura do código](#)
4. [Funcionalidades](#)
 - i. [Registo](#)
 - ii. [Login](#)
 - iii. [Músicas](#)
5. [Diagramas](#)
 - i. [Diagrama Conceptual](#)
 - ii. [Diagrama Físico](#)

Instalação

O projeto foi feito para python 3.

A configuração da ligação com a base de dados está definida em `config.py`

```
$ python3 -m venv venv
$ source venv/bin/activate
$ pip install -r requirements.txt
$ python3 -m bdcringe.main
```

Estrutura

O código da aplicação foi separado em 2 módulos python principais `database` e `menus`.

O módulo `database` contém vários submódulos que servem camada de abstração entre a interface do utilizador e o sql utilizado para as operações com a base de dados.

O módulo `menus` contém os menus para a interface de linha de comandos

Esta estrutura torna a transição para uma interface gráfica mais fácil.

```
|— __init__.py
|— config.py
|— main.py
|— database
|   |— __init__.py
|   |— albums.py
|   |— artists.py
|   |— groups.py
|   |— labels.py
|   |— playlists.py
|   |— songs.py
|   |— user.py
|— menus
|   |— __init__.py
|   |— albums.py
|   |— groups.py
```

Funcionalidades

Registo

Insere na base de dados um novo utilizador com o username e password fornecidos. Dá erro de inserção caso já exista algum utilizador com o mesmo username devido ao campo `username` servir de chave primária (unique).

```
bdcringe.database.user.register(username, password)
```

```
INSERT
INTO
  utilizador
(username, password)
VALUES
  (%s, %s)
returning
  username, editor
```

Login

```
bdcringe.database.user.login(username, password)
```

Dado um nome e password de um utilizador, verifica se existe na base de dados.

```
SELECT
  username, editor
FROM
  utilizador
WHERE
  username LIKE %s and
  password LIKE %s
```

Promover utilizador a editor

Para promover outro utilizador a editor são necessárias duas operações, uma de pesquisa e uma de atualização.

Procurar utilizador

É feita uma listagem de utilizadores sem permissões de editor com base no nome a procurar para tornar a procura mais fácil para o utilizador.

```
bdcringe.database.user.search(username)
```

```
SELECT
  nome
FROM
  utilizador
WHERE
  username LIKE %s% and
  editor = false
```

Tornar editor

A partir do username introduzido pelo utilizador altera os seus privilégios de editor.

```
bdcringe.database.user.make_editor(username)
```

```
UPDATE
  utilizador
SET
  editor = true
WHERE
  nome LIKE %s'
```

Procurar Artista

Dado um nome de um artista é retornada toda a informação relativa ao artista em questão, retorna erro no caso do artista não existir.

```
bdcringe.database.artists.search(nome)
```

```
SELECT
*
FROM
artista
WHERE
nome LIKE '%s'
```

Listar as músicas de um Artista

Dado um nome de um artista, devolve todas as músicas dele

```
bdcringe.database.artists.get_songs(nome)
```

```
SELECT
nome,
data,
historia
FROM
musica,
musica_artista
WHERE
artista_id = (
SELECT
id
FROM
artista
WHERE
nome LIKE '%s'
) and
musica_id = id
```

Inserir um Album

Dado um nome, uma data de lançamento válida, um nome de grupo musical e o nome da editora correspondente, insere um novo album. O id da editora é obtido através de uma subconsulta.

```
bdcringe.database.albums.insert(nome, lancamento, grupo_musical, editora)
```

```
INSERT
INTO
  album
  (nome, lancamento, editora_id, grupo_musical_nome)
SELECT
  %s,
  %s,
  id,
  %s
FROM
  editora
WHERE
  nome LIKE %s
```

Procura de um Album

Dado um nome de um album devolve toda a sua informação.

```
bdcringe.database.albums.search(nome)
```

```
SELECT
  *
FROM
  album
WHERE
  nome LIKE %(like)s ESCAPE '='
```

Alterar o nome de um Album

Dado um nome original e um novo nome, procura e altera o album.

```
bdcringe.database.albums.update_nome(antigo, novo)
```

```
UPDATE
  album
SET
  nome = %s
WHERE
  nome LIKE %s
```

Apagar um album

Dado o nome de um album, remove-o da base de dados.

```
bdcringe.database.albums.delete(nome)
```

```
DELETE
FROM
  album
WHERE
  nome LIKE %s
```

Listar músicas de um album

Dado um nome de um album, devolve todas as músicas a que ele pertencem.

```
bdcringe.database.albums.songs(nome)
```

```
SELECT
  *
FROM
  musica
WHERE
  album_nome LIKE %s
```

Criticas

Escrever uma crítica a um album

É pedido ao utilizador uma pontuação e uma justificação textual para essa pontuação e é registada na tabela de críticas.

```
bdcringe.database.albums.review(nome, pontuacao, justificacao, username)
```

```
INSERT
INTO
  critica
  (album_nome, utilizador_username, pontuacao, justificacao)
VALUES
  (%s, %s, %s, %s)
```

Mostrar todas as críticas de um album

Dado um nome de um album, mostra todas as suas críticas.

```
bdcringe.database.albums.reviews(nome)
```

```
select
  pontuacao, justificacao, utilizador_username
from
  critica
where
  album_nome like %s
```


Inserir Editora

Dado um nome, insere uma nova editora na base de dados.

```
bdcringe.database.labels.insert(nome)
```

```
INSERT  
INTO  
  editora  
  (nome)  
VALUES  
  (%s)
```

Procurar Editora

Dado um nome devolve toda a informação relativa a essa editora.

```
bdcringe.database.labels.search(nome)
```

```
SELECT  
  *  
FROM  
  editora  
WHERE  
  nome LIKE %s%
```

Alterar o nome de uma editora

Dado um nome original e um novo nome, procura e altera a editora.

```
bdcringe.database.labels.update(antigo, novo)
```

```
UPDATE  
  editora  
SET  
  nome = %s  
WHERE  
  nome LIKE %s
```

Remover Editora

Dado o nome de um album, remove-o da base de dados.

```
bdcringe.database.labels.delete(nome)
```

```
DELETE  
FROM  
  editora  
WHERE  
  nome LIKE %s
```

Listar editoras

Lista todas a editoras existentes.

```
bdcringe.database.labels.get_all()
```

```
SELECT
*
FROM
editora
```

Inserir Grupo Musical

Dado um nome, data de inicio e data de fim, insere um novo grupo musical.

```
INSERT
INTO
grupo_musical
(nome, inicio, fim)
VALUES
(%, %, %)
```

Adiciona Artista ao Grupo Musical

Dado um nome de artista e o nome do grupo musical, insere o artista um novo artista_grupo_musical.

```
INSERT
INTO
  artista_grupo_musical
  (artista_id, grupo_musical_nome)
SELECT
  a.id,
  g.nome
FROM
  artista a,
  grupo_musical g
WHERE
  a.nome LIKE %s and
  g.nome LIKE %s
```

Listar membros de um Grupo Musical

Dado um nome do grupo musical, lista todos os artista presentes no grupo. A pesquisa é feita através do id do grupo_musical(nome) na tabela de artista_grupo_musical, listando toda informação dos artistas associados ao id do grupo_musical(nome).

```
SELECT
  *
FROM
  artista a,
  artista_grupo_musical agm
WHERE
  agm.grupo_musical_nome LIKE %s and
  a.id = agm.artista_id
```

Inserir Música

Este processo é feito em dois passos, criando uma entrada na tabela musica e criando posteriormente uma entrada na tabela musica_artista.

Criar Musica e inserir na base de dados

Dado um nome da música, uma data de criação válida, um resumo breve da história, gênero, e nome do album cria uma nova música.

```
INSERT
INTO
  musica
  (nome, data, historia, genero, album_nome)
VALUES
  ('%s', '%s', '%s', '%s', '%s')
```

Associar a música a um artista

Dado o nome do artista a quem está associada a música, associa a música ao artista através da tabela musica_artista. A associação é feita através dos identificadores, o que é devolvido pela subconsulta.

```
INSERT
INTO
  musica_artista
  (artista_id, musica_id)
SELECT
  a.id,
  m.id
FROM
  artista a,
  musica m
WHERE
  m.nome LIKE '%s' and
  a.nome LIKE '%s'
```

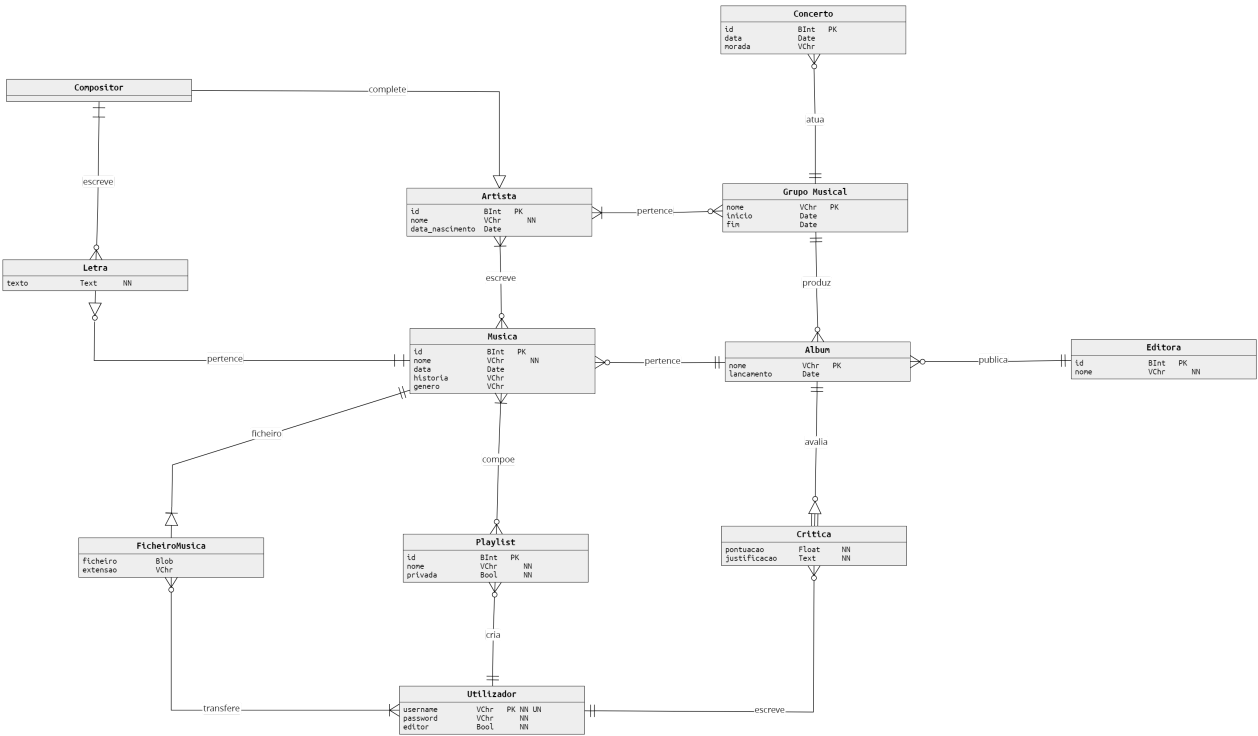
Procurar Música

Dado um nome é retornada toda a informação geral e a lista de artistas de cada música encontrada.

```
SELECT
  m.nome,
  m.data,
  m.historia,
  m.genero,
  a.nome,
  a.data_nascimento
FROM
  artista a,
  musica_artista ma,
  musica m
WHERE
  a.id = ma.artista_id
  and m.id = ma.musica_id
  and m.nome LIKE '%s%'
```

Diagramas

Conceptual



Físico

