

PROGRAMA DE ACESSO A BASE DE DADOS: MYANIMELIST

Leonardo Holtz de Oliveira

Luiza Schmidt

Linguagem e bibliotecas usadas:

A linguagem escolhida foi Python juntamente com a biblioteca psycopg2

Acesso ao banco de dados:

```
def connect():
    """ Conexão a base de dados PostgreSQL"""
    conn = None
    try:
        # conexão ao server PostgreSQL
        print("Conectando a base de dados PostgreSQL ...")
        conn = psycopg2.connect("dbname=MyAnimeList user=aluno password=aluno host=localhost")
        print("Conectado")
```

Foi usado o método connect da biblioteca psycopg2. A variável conn irá permitir o uso dos cursores em cada consulta e inserção no banco de dados. Caso não seja possível manter uma conexão com o banco de dados, uma exceção será gerada. No final do programa a conexão se encerrará com o método close.

```
        # encerra a comunicacao com o PostgreSQL
    except (Exception, psycopg2.DatabaseError) as error:
        print(error)
    finally:
        if conn is not None:
            conn.close()
            print('Conexao com a base de dados fechada.')
```

Consultas:

As consultas foram realizadas através da criação de um cursor por meio do método cursor a partir da conexão estabelecida. A execução dos procedimentos select é feito por meio do método execute. O método fetchall retornará o resultado das consultas através de uma lista de lista dos atributos selecionados.

```
def consultaN1(conn):
    cur = conn.cursor()
    cur.execute("select no
    rows = cur.fetchall()
```

As consultas que usam parâmetros fornecidos pelo usuário em tempo de execução são recebidos através do método input do python.

```
autor = input("Digite um autor existente: ")
genero = input("Digite um genero de manga existente: ")
..
```

Como a consulta não passa de uma string passada por parâmetro no método execute, para usar as variáveis nela, usamos a concatenação de strings. Em python é usado o símbolo '+' para fazer essa concatenação.

```
cur.execute("select nomemanga from manga where nomemanga IN  
  
(select nomemanga from caracterizacaoManga where nomeg = '"' + genero + "') AND  
nomemanga IN (select nomemanga from criacao where nomepessoa = '"' + autor + "')")
```

Para acessar e mostrar ao usuário esses atributos é feito 2 laços for aninhados com um print único, caso a consulta forneça somente um atributo em seus resultados.

```
for row in rows:  
    for item in row:  
        print(item)  
    print("\n\n")
```

Caso a consulta forneça uma série de atributos, é usado o método format do python para que os resultados sejam impressos de forma tabelada.

```
print("Usuario                codigo                Anime")  
for row in rows:  
    for item in row:  
        print('{:<24}'.format(item), end = '')  
    print()  
    print("\n\n")
```

Atualizações:

As atualizações fazem somente o uso das inserções que acionam um trigger de update. O trigger de update feito recalcula a média das notas de todas as listas sobre um determinado anime ou mangá e atribui essa nova média ao atributo medpontos na tabela referente àquele anime ou mangá. Foi usado o mesmo procedimento de receber inputs do usuário referente aos atributos da tabelas contencaoListaAnimes e contencaoListaMangas.

```
def insereAnime(conn):  
    anime = input("Digite o nome do anime (80 chars): ")  
    lista = input("Digite o codigo da lista (inteiro): ")  
    nota = input("Digite a nota de 0 a 10 (inteiro): ")  
    categoria = input("Digite a categoria (Assistindo, Completo, EmEspera, Desistido, ou PlanejadoParaAssistir): ")  
    progresso = input("Digite o numero de episodios vistos (inteiro): ")  
    inicio = input("Digite a data de inicio em formato DD/MM/AAAA (ou 0 caso não queria): ")  
    if inicio == "0":  
        inicio = "NULL"  
    else:  
        inicio = "'" + inicio + "'"  
    fim = input("Digite a data de finalização em formato DD/MM/AAAA (ou 0 caso não queria): ")  
    if fim == "0":  
        fim = "NULL"  
    else:  
        fim = "'" + fim + "'"
```

A partir disso, é usado novamente o método execute com o cursor juntamente com a concatenação de strings para inserir uma nova tupla nas tabelas.

```

cur = conn.cursor()
cur.execute("select medpontos from anime where nomeanime = '" + anime + "'")
rows = cur.fetchall()
rows
print()
print("Média de todas as notas do anime", anime, ":", end = '')
for row in rows:
    for item in row:
        print(item)
cur.execute("insert into contencaoListaAnimes values('" + anime + "', " + lista + ", " + nota + ", '" + categoria + "', " + progresso + ", " + inicio + ", " + fim + ");")
cur.execute("select medpontos from anime where nomeanime = '" + anime + "'")
rows = cur.fetchall()
rows
print()
print("Nova média de todas as notas do anime", anime, ":", end = '')
for row in rows:
    for item in row:
        print(item)

```

Para fins de comparação, a consulta do atributo medpontos na variável antes e depois da inserção e da ativação do trigger é feita.