

Comandos SQL para manipulação de dados (DML)

Banco de Dados

Charles Tim Batista Garrocho

Instituto Federal de Minas Gerais – IFMG
Campus Ponte Nova

`garrocho.github.io/BDD`

`charles.garrocho@ifmg.edu.br`

Técnico em Informática



INSTITUTO FEDERAL

SQL apresenta uma série de comandos que permitem a **manipulação** dos dados, chamada de DML (*Data Manipulation Language*), é destinado a manipulação do Banco de Dados, das Tabelas que o compõe, além das relações existentes entre as tabelas.

Entre os comandos da classe DML, os principais são:

INSERT: permite a inclusão de novas linhas nas tabelas

UPDATE: altera os valores de dados já cadastrados

DELETE: remove dados já cadastrados

SELECT: usado para consultar o BD e retornar dados que satisfazem a determinada expressão em um comando



Inserindo informações na base de dados

O comando **INSERT** permite inserir uma linha de dados na tabela e possui a seguinte sintaxe abaixo:

```
INSERT INTO TABELA (COLUNA1, COLUNA2)  
VALUES (VALOR1, VALOR2)
```

No lugar de 'TABELA' você coloca o **nome da tabela**.

No lugar de 'coluna1', 'coluna2', você coloca os **nomes das colunas** da tabela.

No lugar de 'valor1', 'valor2', você coloca os **valores** (variáveis que contém o valor) para aquela coluna respectiva em 'coluna1' e 'coluna2'.



INSTITUTO FEDERAL

Recapitulando as informações da tabela Clientes

Campo	Tipo	Requerido
id	inteiro	sim
nome	texto	sim
idade	inteiro	não
cpf	texto (11)	sim
email	texto	sim
fone	texto	não
cidade	texto	não
uf	texto (2)	sim
criado_em	data	sim
bloqueado	booleano	não



INSTITUTO FEDERAL

Inserindo uma linha na tabela clientes

```
cursor = conn.cursor()
```

```
p_nome = raw_input('Nome: ')
```

```
p_idade = raw_input('Idade: ')
```

```
p_cpf = raw_input('CPF: ')
```

```
p_email = raw_input('Email: ')
```

```
p_fone = raw_input('Fone: ')
```

```
p_cidade = raw_input('Cidade: ')
```

```
p_uf = raw_input('UF: ')
```

```
p_criado_em = raw_input('Criado em (yyyy-mm-dd): ')
```

```
cursor.execute('INSERT INTO clientes (nome, idade, cpf, email,  
fone, cidade, uf, criado_em) VALUES (?, ?, ?, ?, ?, ?, ?, ?)', (p_nome,  
p_idade, p_cpf, p_email, p_fone, p_cidade, p_uf, p_criado_em))
```

```
conn.commit()
```



INSTITUTO FEDERAL

Alterando informações na base de dados

O comando **UPDATE** é usado para mudar valores de linhas de dados que já foram cadastrados anteriormente e que obedecem a determinados critérios, especificados em condições.

Este comando pode alterar **mais de uma linha ao mesmo tempo**, caso mais de uma linha obedeça a determinada **condição**. As condições podem também ser representadas utilizando os operadores: AND, OR e NOT.

O comando UPDATE, contém a cláusula **WHERE**, de forma a **restringir** o conjunto dos registros que serão processados pelo comando. Se não for colocada a cláusula WHERE no comando UPDATE, as alterações serão realizadas em **todos** os registros da tabela.



INSTITUTO FEDERAL

Alterando informações na base de dados

O comando **UPDATE** permite alterar uma ou mais linhas de dados na tabela e possui a seguinte sintaxe abaixo:

```
UPDATE TABELA SET COLUNA1 = VALOR1, COLUNA2 = VALOR2  
WHERE CONDIÇÕES
```

No lugar de 'TABELA' você coloca o **nome da tabela**. No lugar de 'coluna1', 'coluna2', você coloca os **nomes das colunas** da tabela.

No lugar de 'valor1', 'valor2', você coloca os **valores** (variáveis que contém o valor) para aquela coluna respectiva em 'coluna1' e 'coluna2'.

No lugar de **condições** (que é opcional) você deve especificar quais linhas específicas você quer alterar.



INSTITUTO FEDERAL

Recapitulando as informações da tabela clientes

Campo	Tipo	Requerido
id	inteiro	sim
nome	texto	sim
idade	inteiro	não
cpf	texto (11)	sim
email	texto	sim
fone	texto	não
cidade	texto	não
uf	texto (2)	sim
criado_em	data	sim
bloqueado	booleano	não



INSTITUTO FEDERAL

Alterando uma linha na tabela clientes

```
cursor = conn.cursor()
```

```
id_cliente = 1
```

```
novo_fone = '11-1000-2014'
```

```
novo_criado_em = '2014-06-11'
```

```
cursor.execute('UPDATE clientes SET fone = ?,  
criado_em = ? WHERE id = ?', (novo_fone, novo_criado_em,  
id_cliente))
```

```
conn.commit()
```



INSTITUTO FEDERAL

Excluindo informações na base de dados

O comando **DELETE** é usado para remover linhas de uma tabela. Este comando pode remover mais de uma linha ao mesmo tempo, caso mais de uma linha obedeça a uma certa condição. As condições podem ser representadas utilizando os operadores AND, OR e NOT.

O comando DELETE, contém a cláusula **WHERE**, de forma a restringir o conjunto dos registos que serão processados pelo comando. Se não for colocada a cláusula WHERE no comando DELETE, serão apagados todos os registos de uma tabela.

Assim como no comando UPDATE, podemos utilizar operadores relacionais (maior, menor, igual) e os operadores lógicos (AND, OR) para especificar as **condições** de exclusão de dados.



INSTITUTO FEDERAL

Excluindo informações na base de dados

O comando **DELETE** permite excluir uma ou mais linhas de dados na tabela e possui a seguinte sintaxe abaixo:

```
DELETE FROM TABELA WHERE <CONDIÇÕES>
```

No lugar de 'TABELA' você coloca o **nome da tabela**.

No lugar de **condições** (que é opcional) você deve especificar quais linhas específicas você quer excluir.



INSTITUTO FEDERAL

Resumindo as informações da tabela clientes

Campo	Tipo	Requerido
id	inteiro	sim
nome	texto	sim
idade	inteiro	não
cpf	texto (11)	sim
email	texto	sim
fone	texto	não
cidade	texto	não
uf	texto (2)	sim
criado_em	data	sim
bloqueado	booleano	não



INSTITUTO FEDERAL

Excluindo linhas na tabela clientes

```
cursor = conn.cursor()
```

```
# excluindo todas as linhas da tabela  
cursor.execute('DELETE FROM clientes')
```

```
conn.commit()
```

```
# excluindo a linha do cliente com id 8  
id_cliente = 8
```

```
cursor.execute('DELETE FROM clientes WHERE id = ?',  
(id_cliente,))
```

```
conn.commit()
```



INSTITUTO FEDERAL

O comando **SELECT** permite ler uma ou mais linhas de dados na tabela e possui a seguinte sintaxe abaixo:

```
SELECT <LISTA DE ATRIBUTOS> FROM TABELA WHERE  
<CONDIÇÕES>
```

No lugar de 'TABELA' você coloca o **nome da tabela**.

No lugar de **condições** (que é opcional) você deve especificar quais linhas específicas você quer ler.



Recapitulando as informações da tabela clientes

Campo	Tipo	Requerido
id	inteiro	sim
nome	texto	sim
idade	inteiro	não
cpf	texto (11)	sim
email	texto	sim
fone	texto	não
cidade	texto	não
uf	texto (2)	sim
criado_em	data	sim
bloqueado	booleano	não



INSTITUTO FEDERAL

Lendo linhas da tabela clientes

```
cursor = conn.cursor()
```

```
# lendo todas as linhas da tabela
```

```
cursor.execute('SELECT * FROM clientes;')
```

```
for linha in cursor.fetchall():  
    print(linha)
```

```
# lendo o nome e fone da linha do cliente com id 8
```

```
id_cliente = 8
```

```
cursor.execute('SELECT nome, fone FROM clientes WHERE id =  
?', (id_cliente,))
```



INSTITUTO FEDERAL

Utilizando o pycharm, implemente um sistema de controle e gerenciamento de empréstimos de livros de uma biblioteca.

Utilize a base de dados criada no exercícios práticos da última aula.

O sistema deve permitir as seguintes funcionalidades:

- Cadastrar, alterar e excluir livros, editoras, e autores;
- Permitir o empréstimo de livros para usuários;

Atividade extra: Pesquise sobre comandos DML para o banco de dados PostgreSQL.

