

# Comandos SQL para definição de dados (DDL)

## 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

# Introdução

Para a criação de banco de dados, tabelas e atributos, utilizaremos a linguagem SQL que é composta de comandos de **manipulação**, **definição** e **controle** de dados.

A SQL estabeleceu-se como **linguagem padrão** de Banco de Dados Relacional.

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

Entre os comandos da classe DDL, os principais são:

CREATE (Criação da tabela)

ALTER (Alteração da tabela)

DROP (Exclusão da tabela)



INSTITUTO FEDERAL

# Definindo as informações

Para os exemplos considere a tabela clientes e seus campos:

<b>Campo</b>	<b>Tipo</b>	<b>Requerido</b>
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

Obs: O campo bloqueado nós vamos inserir depois com o comando **ALTER TABLE**.



INSTITUTO FEDERAL

# Conectando e desconectando do banco

Antes de tudo é necessário **importar** a biblioteca SQLite3.

```
import sqlite3
```

Após importar a biblioteca, podemos **criar e conectar** ao banco de dados. Isso pode ser realizado de duas formas. Através da Memória RAM:

```
conn = sqlite3.connect(':memory:')
```

ou persistindo em um banco de dados, *vamos usar sempre este caso.*

```
conn = sqlite3.connect('clientes.db')
```

Caso o arquivo não exista, o SQLite3 cria o arquivo, seja na memória RAM ou no disco. Após a utilização do banco de dados, é necessário **finalizar** a conexão com o seguinte comando:

```
conn.close()
```



INSTITUTO FEDERAL

# Criando uma tabela

Antes de criar a tabela é necessário obter o **cursor** da conexão com o banco de dados:

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

O comando **CREATE TABLE** é o principal comando DDL da linguagem SQL. A criação de tabelas é realizada em SQL utilizando este comando. Sua sintaxe básica é a seguinte:

```
CREATE TABLE NOME_DA_TABELA(  
TIPO CAMPO1,  
TIPO CAMPO2,  
TIPO CAMPO3)
```

Em SQL, cada campo (coluna) de uma tabela deverá possuir um **nome** e um **tipo** de dados, quem em geral são agrupados em 3 categorias: Caracteres (Strings), Numéricos, e Tempo e Data.



INSTITUTO FEDERAL

# Tipos de dados

Os tipos de dados de um campo (coluna) que podem ser escolhidos são os seguintes:

**INTEGER:** inteiro com sinal, armazenado em 1, 2, 3, 4, 6 ou 8 bytes dependendo da grandeza do valor.

**REAL:** valor de ponto flutuante armazenado em 8 bytes.

**TEXT:** uma string armazenada usando UTF-8, UTF-16BE ou UTF-16LE.

**BLOB:** para armazenar um grande conjunto de dados (em geral, imagens e arquivos).

**VARCHAR(N):** tipo de dados string com comprimento variável.

**BOOLEAN:** tipo de dados booleano sendo true ou false.



# Restrições na tabela

Na criação de tabelas, também podem ser definidas o uso de **RESTRIÇÕES**. Restrições são regras a que os valores de uma ou mais colunas devem obedecer. As principais regras são:

**NOT NULL**: para que uma coluna não admita valores nulos. Caso um valor não seja inserido em uma coluna o valor padrão (default) armazenado nela é NULL.

**PRIMARY KEY**: utilizado sempre que uma coluna é definida como chave primária.

**AUTO INCREMENT**: define que um campo irá inserir um valor inteiro e que os valores serão automaticamente incrementados.



INSTITUTO FEDERAL

# Recapitulando as informações

<b>Campo</b>	<b>Tipo</b>	<b>Requerido</b>
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



# Criando a tabela de clientes

```
import sqlite3
```

```
conn = sqlite3.connect('clientes.db')
```

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

```
cursor.execute( 'CREATE TABLE clientes
```

```
id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
```

```
nome TEXT NOT NULL,
```

```
idade INTEGER,
```

```
cpf VARCHAR(11) NOT NULL,
```

```
email TEXT NOT NULL,
```

```
fone TEXT,
```

```
cidade TEXT,
```

```
uf VARCHAR(2) NOT NULL,
```

```
criado_em DATE NOT NULL);')
```

```
conn.close()
```



INSTITUTO FEDERAL

# Excluindo e alterando uma tabela

O comando **DROP TABLE** permite remover uma determinada tabela de um Banco de Dados, e conseqüentemente, todos os dados existentes nela.

A sintaxe do comando é:

```
DROP TABLE NOME_DA_TABELA
```

O comando **ALTER TABLE** permite alterar a estrutura de uma tabela.

Com o comando é possível adicionar uma nova coluna, modificar uma coluna já existente ou eliminar uma coluna. Veja a sintaxe dos comandos:

```
ALTER TABLE NOME_DA_TABELA ADD NOME_COLUNA  
TIPO_COLUNA
```

```
ALTER TABLE NOME_DA_TABELA MODIFY NOME_COLUNA  
TIPO_COLUNA
```

```
ALTER TABLE NOME_DA_TABELA DROP NOME_COLUNA  
TIPO_COLUNA
```



INSTITUTO FEDERAL

# Recapitulando as informações

<b>Campo</b>	<b>Tipo</b>	<b>Requerido</b>
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

# Adicionando uma coluna na tabela de clientes

```
import sqlite3
```

```
conn = sqlite3.connect('clientes.db')
```

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

```
cursor.execute( 'ALTER TABLE clientes ADD COLUMN bloqueado  
BOOLEAN;')
```

```
conn.commit()
```

```
conn.close()
```

A novidade neste código é a função **commit**. Ela é necessária para refletir as informações na base de dados.



INSTITUTO FEDERAL

# Renomeando e excluindo a tabela clientes

No SQLite3, além da criação da tabela, é permitido apenas renomear ou excluir uma tabela.

Renomeando:

```
cursor.execute( 'ALTER TABLE clientes RENAME TO  
clientes_novo;')
```

Excluindo:


```
cursor.execute( 'DROP TABLE clientes_novo;')
```

Obs: no SQLite3 não é possível excluir uma coluna ou alterar o tipo da coluna.



INSTITUTO FEDERAL

Utilizando o pycharm, crie uma base de dados com o SQLite3. A base de dados deverá ser implementada para um sistema de controle e gerenciamento de empréstimos de livros de uma biblioteca.

- A biblioteca dispõe de livros, também denominados títulos. Estes possuem nome, autores e editoras.
  - Cada título possui vários exemplares. Cada exemplar possui um código único de identificação.
  - Cada título pode ter vários autores e um mesmo autor pode ter escrito vários títulos. Um autor possui código, nome, telefone e endereço.
  - As editoras possuem código, nome, telefone e endereço.
  - Usuários, que podem ser alunos, professores ou funcionários, tomam livros emprestados. Cada usuário possui um código, nome, telefone e endereço.
- 



INSTITUTO FEDERALE

Após a criação do banco de dados, novas informações foram solicitadas para serem adicionadas no base de dados. Adicione essas informações considerando apenas a modificação da base.

- A tabela livros (títulos) terá um novo campo chamado ano.
- Renomear a tabela usuários para clientes.

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

