

Banco de Dados 1

SQL/DDL Básico - Comandos

SQL (Structured Query Language - Linguagem de Consulta Estruturada) é a linguagem declarativa padrão para banco de dados relacional (base de dados relacional).

DDL (Data Definition Language - Linguagem de Definição de Dados) pode ser caracterizado como um subconjunto de instruções SQL para definição da estrutura de um banco de dados (tabelas, colunas, constraints, etc).

Hoje iremos executar comandos SQL/DDL diretamente no terminal psql para:

- Criar tabelas e restrições;
- Remover tabelas e restrições;
- Alterar tabelas e restrições;

LEMBRE-SE:

Os comandos SQL devem sempre encerrar com “;” (ponto-e-vírgula).

Caso o comando seja aceito (executado pelo SGBD), será exibido o nome do comando no terminal, como resposta. Por exemplo: “CREATE TABLE”

Abaixo estão os comandos que você irá utilizar HOJE.

- CREATE TABLE - Criação de tabelas;
- DROP TABLE - Remoção de tabelas;
- ALTER TABLE - Alteração de tabelas;

A seguir estão vários exemplos de comandos SQL/DDL.

Exemplo de uso de CREATE TABLE:

```
CREATE TABLE nome_da_tabela (  
  nome_atributo1 tipo_atributo1,  
  nome_atributo2 tipo_atributo2,  
  nome_atributon tipo_atributon  
);
```

Para o roteiro de HOJE, vocês devem usar apenas 3 tipos de Constraints:

- PRIMARY KEY;
- FOREIGN KEY;
- NOT NULL;

Exemplo de uso de DROP TABLE:

```
DROP TABLE nome_da_tabela;
```

Exemplos de uso de CREATE TABLE com Constraint de Chave Primária:

Com apenas um atributo na chave:

```
CREATE TABLE nome_da_tabela (  
    nome_atributo1 tipo_atributo1 PRIMARY KEY,  
    nome_atributo2 tipo_atributo2,  
    nome_atributon tipo_atributon  
);
```

```
CREATE TABLE nome_da_tabela (  
    nome_atributo1 tipo_atributo1 CONSTRAINT nome_da_constraint PRIMARY KEY,  
    nome_atributo2 tipo_atributo2,  
    nome_atributon tipo_atributon  
);
```

```
CREATE TABLE nome_da_tabela (  
    nome_atributo1 tipo_atributo1,  
    nome_atributo2 tipo_atributo2,  
    nome_atributon tipo_atributon,  
    PRIMARY KEY (atributo1)  
);
```

Com mais de um atributo na chave:

```
CREATE TABLE nome_da_tabela (  
    nome_atributo1 tipo_atributo1,  
    nome_atributo2 tipo_atributo2,
```

```
nome_atributon tipo_atributon,  
CONSTRAINT nome_da_constraint PRIMARY KEY(atributo1,atributo2)  
);
```

```
CREATE TABLE nome_da_tabela (  
  nome_atributo1 tipo_atributo1,  
  nome_atributo2 tipo_atributo2,  
  nome_atributon tipo_atributon,  
  PRIMARY KEY(atributo1,atributo2)  
);
```

PADRONIZAÇÃO: No caso de chaves primárias, usaremos nome_da_constraint como sendo nome_da_tabela_pkey.

Exemplos de uso de CREATE TABLE com Constraint de Chave Estrangeira:

```
CREATE TABLE nome_da_tabela (  
  nome_atributo1 tipo_atributo1,  
  nome_atributo2 tipo_atributo2,  
  nome_atributo3 tipo_atributo3 REFERENCES nome_outra_tab (atributo_outra_tab),  
  nome_atributo4 tipo_atributo4,  
  nome_atributon tipo_atributon,  
);
```

```
CREATE TABLE nome_da_tabela (  
  nome_atributo1 tipo_atributo1,  
  nome_atributo2 tipo_atributo2,  
  nome_atributo3 tipo_atributo3,  
  nome_atributo4 tipo_atributo4,  
  nome_atributon tipo_atributon,  
  FOREIGN KEY (atributo1_dest_a_tab, atributo2_dest_a_tab) REFERENCES  
  nome_outra_tabela (atributo1_outrat_a_tab, atributo2_outra_tab)  
);
```

```
CREATE TABLE nome_da_tabela (  
  nome_atributo1 tipo_atributo1,  
  nome_atributo2 tipo_atributo2,
```

```
nome_atributo3 tipo_atributo3,  
nome_atributo4 tipo_atributo4,  
nome_atributon tipo_atributon,  
CONSTRAINT nome_da_constraint FOREIGN KEY (atributo1_desta_tab)  
REFERENCES nome_outra_tab (atributo1_outra_tab)  
);
```

PADRONIZAÇÃO: No caso de chaves estrangeiras, usaremos nome_da_constraint como sendo nomedestatabela_atributodestatabela_fkey (nome da tabela e nome do atributo da tabela que referencia).

Exemplos de uso de CREATE TABLE com Constraint de NOT NULL:

```
CREATE TABLE nome_da_tabela (  
  nome_atributo1 tipo_atributo1,  
  nome_atributo2 tipo_atributo2 NOT NULL,  
  nome_atributo3 tipo_atributo3,  
  nome_atributo4 tipo_atributo4 NOT NULL,  
  nome_atributon tipo_atributon,  
);
```

Exemplos de uso de ALTER TABLE:

```
ALTER TABLE nome_atual_tabela RENAME TO nome_novo_tabela;  
  
ALTER TABLE nome_tabela ADD COLUMN nome_atributo varchar(30);  
  
ALTER TABLE nome_tabela DROP COLUMN nome_atributo RESTRICT;  
  
ALTER TABLE nome_tabela RENAME COLUMN nome_atual_atributo TO  
nome_novo_atributo;  
  
ALTER TABLE nome_tabela ALTER COLUMN nome_atributo SET NOT NULL;  
  
ALTER TABLE nome_tabela ALTER COLUMN nome_atributo DROP NOT NULL;  
  
ALTER TABLE nome_tabela DROP CONSTRAINT nome_da_constraint;  
  
ALTER TABLE nome_da_tabela ADD CONSTRAINT nome_constraint FOREIGN KEY  
(atributo) REFERENCES outra_tabela (atributo);  
  
ALTER TABLE nome_tabela ADD PRIMARY KEY (atributo);
```

PADRONIZAÇÃO: observe que nos exemplos acima todos os comandos SQL estão em letras MAIÚSCULAS, enquanto os demais termos estão em letras minúsculas o usando o separador '_'. Adotaremos este padrão em nossa coficiação.

Definições dos comportamentos das FOREIGN KEYs para os comandos DELETE e UPDATE

Sintaxe:

... ON DELETE [RESTRICT | CASCADE | SET NULL]

... ON UPDATE [RESTRICT | CASCADE | SET NULL]

Exemplos:

```
ALTER TABLE nome_da_tabela ADD CONSTRAINT nome_constraint FOREIGN KEY  
(atributo) REFERENCES outra_tabela (atributo) ON DELETE CASCADE;
```

```
ALTER TABLE nome_da_tabela ADD CONSTRAINT nome_constraint FOREIGN KEY  
(atributo) REFERENCES outra_tabela (atributo) ON UPDATE SET NULL;
```

```
ALTER TABLE nome_da_tabela ADD CONSTRAINT nome_constraint FOREIGN KEY  
(atributo) REFERENCES outra_tabela (atributo) ON DELETE CASCADE ON UPDATE  
CASCADE;
```