



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
**PARAÍBA**



# *Linguagem SQL*

## *DDL – Linguagem de Definição de Dados*



- Definem os objetos do banco de dados, criando, alterando ou excluindo:
  - o banco de dados e tabelas;
  - os índices, procedures, triggers, views e outros objetos
- As instruções **DDL** são:
  - **CREATE** <nome do objeto> - criar objetos
  - **ALTER** <nome do objeto> - alterar objetos
  - **DROP** <nome do objeto> - excluir objetos



## Sintaxe:

```
CREATE DATABASE nome do banco de dados
```

**Exemplo 1:** Criar o banco de dados **Teste**.

```
CREATE DATABASE Teste;
```



## Sintaxe:

```
USE nome do banco de dados
```

**Exemplo 2:** Colocar em uso o banco de dados **Teste**.

```
USE Teste;
```



- Para sabermos quais tabelas estão contidas no banco de dados em uso, utilizamos o comando `SHOW TABLES`.

## Sintaxe:

```
SHOW TABLES;
```

**Exemplo 3:** Listar as tabelas do banco de dados **Teste**.

```
USE Teste;  
SHOW TABLES;
```



## Sintaxe:

**DROP DATABASE** nome do banco de dados

- A exclusão de um banco de dados remove do disco todos os arquivos utilizados pelo banco de dados.

**Exemplo 4:** Excluir o banco de dados **Teste**.

```
DROP DATABASE Teste;
```



- A criação das tabelas deverá ser feita de acordo com o dicionário de dados, definido no esquema lógico.
- Ao criar tabelas, deverão ser definidos previamente:
  - Nome, tamanho e tipo de dado de cada coluna;
  - Chaves primárias, estrangeiras e candidatas;
  - Colunas que aceitam valores nulos;
  - Restrições de domínio para cada coluna;
  - Restrições da tabela.



- Ao criar uma tabela deve-se observar as seguintes regras:
  - O nome de uma tabela deverá ser único em um banco de dados;
  - Cada coluna deverá ser criada através da especificação do seu nome, tipo e tamanho do dado que irá armazenar e, opcionalmente, a permissão de valor assumir valores nulos e restrições (constraints) impostas ao conteúdo;
  - O nome de uma coluna deverá ser único dentro de cada tabela podendo, entretanto, existir colunas com o mesmo nome em tabelas diferentes.



- Números inteiros:

Tipo de dado	Intervalo	Armazenamento
tinyint	de -128 a 127	1 byte
smallint	de -32768 a 32767	2 bytes
mediumint	de -8388608 a 8388607	3 bytes
int / integer	de -2147483648 a 2147483647	4 bytes
bigint	de $-2^{63}$ (-9223372036854775808) a $2^{63}-1$ (9223372036854775807)	8 bytes

- Pode-se aplicar **UNSIGNED** para impedir que valores negativos sejam armazenados. Nesse caso, a faixa de valores positivos é duplicada.

- Números decimais:
  - `numeric(m[,d])` | `decimal(m[,d])`
- **m** (total de dígitos) vai até 65 e **d** (parte fracionária) vai até 30.
- O armazenamento da parte inteira e da parte fracionária são determinados separadamente.
- Cada múltiplo de 9 dígitos são armazenados em 4 bytes e os dígitos restantes são armazenados de acordo com a tabela seguinte:

Dígitos restantes	Armazenamento
1 – 2	1 byte
3 – 4	2 bytes
5 – 6	3 bytes
7 – 9	4 bytes



- Data e hora:

Tipo de dado	Intervalo	Armazenamento
date	data de 01/01/1000 a 31/12/9999	3 bytes
time	de -838:59:59 a 838:59:59	3 bytes
datetime	data de 01/01/1970 a 31/12/9999 hora de 00:00:00 a 23:59:59	8 bytes



- Strings:

Tipo de dado	Intervalo	Armazenamento
char(n)	string com comprimento fixo de até 255 caracteres	n bytes
varchar(n)	string com comprimento variável de até 65535 caracteres	até n+1 bytes



## Sintaxe:

```
CREATE TABLE nome da tabela  
( nome_da_coluna  tipo_de_dado [ tamanho [, decimais]]  
  [ NULL | NOT NULL ] [ restrições de coluna ]  
  [AUTO INCREMENT]  
  [ restrições de tabela ]
```

### ■ NULL | NOT NULL

- Indica se o atributo é opcional (NULL) ou obrigatório (NOT NULL). Isso define se o valor do atributo deverá ser informado no momento de inserir o dado. Se omitido, o valor default é NULL.



- Também chamadas de ***constraints***, são condições que garantem que valores válidos sejam inseridos nas colunas e que os relacionamentos entre as tabelas sejam mantidos.
- Se dividem em:
  - **Restrição de coluna:** quando a restrição se aplica a uma única coluna da tabela.
  - **Restrição de tabela:** quando a restrição faz referência a várias colunas da tabela.



- As constraints são:
  - **Primary Key**
    - Define uma chave primária.
  - **Foreign Key**
    - Define uma chave estrangeira.
  - **Unique**
    - Define uma chave candidata.
  - **Check**
    - Define uma restrição de domínio.
  - **Default**
    - Define um valor default para o atributo.



- É utilizada para definir a chave primária da tabela;
- Impede a inclusão de valores duplicados para a coluna;
- Cria um índice na coluna para melhorar o desempenho;
- Uma tabela pode conter apenas uma restrição PRIMARY KEY;
- Em uma chave primária composta, os nomes das colunas devem ser separados por vírgulas e todas elas devem ser definidas como NOT NULL.





- É utilizada para definir uma chave estrangeira para a tabela;
- Todo valor não nulo introduzidos na coluna com uma restrição FOREIGN KEY deve existir na chave primária da tabela referenciada;
- O número de colunas e tipos de dados das colunas especificadas em uma restrição FOREIGN KEY devem coincidir com as colunas da tabela referenciada.
- O nome de uma coluna com restrição FOREIGN KEY não precisa, ter, necessariamente, o mesmo nome da coluna da chave primária da tabela referenciada, mas deve possuir o mesmo tipo de dado dessa tabela.



- É utilizada para definir a chave candidata da tabela;
- Impede a inclusão de valores duplicados para a coluna;
- Cria um índice na coluna para melhorar o desempenho.



- É utilizada para definir restrições de domínio.
- Uma coluna pode possuir várias restrições CHECK e os critérios podem incluir diversas expressões lógicas combinadas com AND e OR. As restrições são validadas pela ordem da criação.
- Uma restrição CHECK de coluna não pode fazer referência a outras colunas da tabela.
- Uma restrição CHECK de tabela pode fazer referência a outras colunas da tabela, mas não pode referenciar colunas de outras tabelas.



- É utilizada para definir um valor padrão para uma coluna quando esse valor não for especificado em um comando INSERT.
- Uma coluna contendo uma constraint DEFAULT deverá ser definida como NULL. O valor default possui prioridade.
- Uma coluna pode ter apenas uma restrição DEFAULT.
- Uma restrição DEFAULT não pode fazer referência a outra coluna.
- O valor atribuído em uma restrição DEFAULT deve estar de acordo com o tipo de dado da coluna.
- Uma restrição DEFAULT pode conter valores constantes ou funções do sistema.



## BANCO DE DADOS CONTPEDIDO

Esquema Conceitual

Esquema Lógico

Script SQL



## Sintaxe:

```
DESC nome_da_tabela
```

**Exemplo 5:** Listar as colunas da tabela **Cliente** .

```
DESC Cliente;
```



## Sintaxe:

```
ALTER TABLE nome_da_tabela  
  MODIFY [COLUMN] nome_da_coluna novo_tipo_de_dado  
  [ NULL | NOT NULL ]  
  | ADD definição_da_coluna  
  | ADD [ CONSTRAINT ] definição_da_constraint  
  | DROP COLUMN nome_da_coluna  
  | DROP [ CONSTRAINT ] nome_da_constraint
```



**Exemplo 6:** Adicionar a coluna **idade** do tipo INT na tabela **Cliente** .

```
ALTER TABLE Cliente  
ADD idade INT;
```

**Exemplo 7:** Modificar o tipo da coluna **idade** da tabela **Cliente** para SMALLINT.

```
ALTER TABLE Cliente  
MODIFY idade SMALLINT;
```

**Exemplo 8:** Adicionar uma constraint **CHECK** na coluna **idade** da tabela **Cliente**, de modo que a idade seja maior que 0 (zero).

```
ALTER TABLE Cliente  
ADD CONSTRAINT CK_Cliente_idade CHECK(idade>0);
```





**Exemplo 9:** Inserir a coluna **cpf** do tipo **char(11)** na tabela **Cliente** com uma constraint **CHECK** para validar o tamanho com 11 dígitos.

```
ALTER TABLE Cliente  
ADD cpf char(11)  
CONSTRAINT CK_Cliente_cpf CHECK (CHAR_LENGTH(cpf)=11)
```

**Exemplo 10:** Inserir na tabela **Cliente** uma constraint **UNIQUE** para definir a coluna **cpf** como sendo uma chave candidata.

```
ALTER TABLE Cliente  
ADD CONSTRAINT AK_Cliente_cpf UNIQUE(cpf);
```



**Exemplo 11:** Excluir a constraint **AK\_Cliente\_cpf** da tabela **Cliente**.

```
ALTER TABLE Cliente  
DROP CONSTRAINT AK_Cliente_cpf;
```

**Exemplo 12:** Excluir a coluna **cpf** da tabela **Cliente**.

```
ALTER TABLE Cliente  
DROP COLUMN cpf;
```



## Sintaxe:

```
DROP TABLE [ IF EXISTS ] nome_da_tabela
```

- Utilizado para excluir uma tabela e todos os dados, índices, gatilhos, restrições e especificações de permissão dessa tabela.
- DROP TABLE não pode ser usado para excluir uma tabela que é referenciada por uma restrição FOREIGN KEY. Nesse caso, a restrição FOREIGN KEY que faz referência ou a tabela de referência deve ser primeiramente excluída.



## Exemplo 13: Excluir a tabela **Cliente**.

```
DROP TABLE Cliente;
```

```
/* Erro: É necessário primeiro excluir a FK de  
Cliente na tabela Pedido para depois excluir a  
tabela Cliente.*/
```

```
ALTER TABLE Pedido
```

```
DROP CONSTRAINT FK_pedido_cliente;
```

```
DROP TABLE Cliente;
```

## Exemplo 14: Excluir a tabela **Cliente**, caso ela exista.

```
DROP TABLE IF EXISTS Cliente;
```