



BANCO DE DADOS I

Prof. Taciano Balardin
taciano@ulbra.edu.br

E-MAIL:

taciano@ulbra.edu.br

SITE DA DISCIPLINA:

<http://www.taciano.pro.br/>

SENHA:

[@bd1](#)





Data Definition Language (DDL)

AULA 16

Comandos SQL

SQL Commands

DDL

Create
Alter
Drop
Rename
Truncate

DML

Insert
Update
Delete
Select

DCL

Grant
Revoke

TCL

Savepoint
Rollback
Commit

Data Definition Language (DDL)

- Instruções utilizadas na definição da estrutura de um banco de dados;
- **Criação**, **modificação** e **exclusão** de estruturas de um banco de dados ou de uma tabela são feitas através das instruções DDL;

CREATE

ALTER

DROP

DDL – CREATE

- Instrução que permite a criação de um banco de dados e demais estruturas de um BD.
- As situações onde o **CREATE** é mais utilizado são:
 1. Criação de um **novo banco de dados**;

Exemplos:

```
mysql > CREATE DATABASE banco;
```

```
mysql > CREATE DATABASE IF NOT EXISTS banco;
```

SHOW DATABASES

- Podemos verificar rapidamente a existência do BD recém-criado, bem como a de todos os outros criados anteriormente, utilizando a instrução SHOW DATABASES (mostrar bancos de dados);

Exemplo:

```
mysql > SHOW DATABASES;
```

USE

- Como vimos, podemos criar vários bancos de dados, porém, podemos **manipular apenas um por vez**;
- Desta forma, **é preciso selecionar qual será o banco de dados** que queremos trabalhar;
- Isso é feito utilizando o comando **USE** (“usar” em inglês), seguido pelo nome do banco.

Exemplo:

```
mysql > USE banco;
```


DDL – CREATE



- Instrução que permite a criação de um banco de dados e demais estruturas de um BD.
- As situações onde o **CREATE** é mais utilizado são:
 2. Criação de uma **nova tabela**;

Exemplo:

```
mysql > CREATE TABLE agencia (  
    id integer not null primary key,  
    numero varchar(45) not null  
);
```

DDL – CREATE

- Comando para criar tabela, seguido do nome da tabela, os campos, seu tipo e seu tamanho.

Exemplo:

```
mysql > CREATE TABLE cadastro (  
    id integer not null primary key,  
    nome char(15),  
    sobrenome char(20)  
);
```

NOME DA
TABELA

DEFINIÇÃO
DO CAMPO

TAMANHO
DO CAMPO

CAMPO DA
TABELA

TIPO DE
DADOS

TIPOS DE DADOS

Há vários tipos possíveis de dados no SQL, os mais comuns são:

- **INT** ou **INTEGER**: para inteiros de tamanho normais;
- **FLOAT**: para valores reais;
- **TIMESTAMP**: para o timestamp , pode ser atribuído automaticamente;
- **DATE** e **DATETIME**: para data e data e hora, também pode ser atribuído automaticamente;
- **CHAR** e **VARCHAR**: para caracteres com no máximo 255 de tamanho;
- **TEXT** ou **LONGTEXT**: para textos longos;

DEFINIÇÕES DE CAMPOS

- Alguns campos podem ter particularidades:
 - Por exemplo, ser chave primária, não pode ser vazia e etc.
- O campo com a opção **NOT NULL**, significa que o campo não poderá ser nulo. Para utilizar isso, basta na criação do campo adicionar NOT NULL após o nome do campo.

Exemplo:

```
mysql > CREATE TABLE pessoas (  
    nome VARCHAR(255) NOT NULL  
);
```

DEFINIÇÕES DE CAMPOS

- Alguns campos podem ter particularidades:
 - Por exemplo, ser chave primária, não pode ser vazia e etc.
- Para definirmos que um campo é chave primária, utilizamos a opção **PRIMARY KEY**, após o nome do campo.

Exemplo:

```
mysql > CREATE TABLE pessoas (  
    id INT(5) NOT NULL PRIMARY KEY  
);
```

DEFINIÇÕES DE CAMPOS

- Alguns campos podem ter particularidades:
 - Por exemplo, ser chave primária, não pode ser vazia e etc.
- **AUTO INCREMENT**, significa que a cada registro de uma tabela, o valor será incrementado (aumentado). Geralmente, utilizamos para campos ID, CODIGO ou CHAVES PRIMARIAS;

Exemplo:

```
mysql > CREATE TABLE animais (  
    id INT(5) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(255) NOT NULL  
);
```

SHOW TABLES

- Para exibir a lista de tabelas do banco de dados que está usando atualmente, basta utilizar o comando **SHOW TABLES**.

Exemplo:

```
mysql > SHOW TABLES;
```

DESCRIBE

- Podemos também analisar a estrutura de uma tabela de maneira aprofundada usando o comando **DESCRIBE** (“descrever”, em inglês), seguido pelo nome da tabela.

Exemplo:

```
mysql > DESCRIBE tabela;
```


DDL – ALTER

- Para alterar uma tabela, basta utilizar **ALTER TABLE**, o nome da tabela o qual quer alterar e qual operação de alteração quer fazer.
- Operações: Adicionar novo campo, renomear nome da tabela e etc.

DDL – ALTER

Exemplo:

Renomeando uma tabela:

```
mysql > ALTER TABLE agencia RENAME TO cliente;
```

Adicionando um campo:

```
mysql > ALTER TABLE cliente ADD idade int(3);
```

Modificando um campo:

```
mysql > ALTER TABLE cliente MODIFY idade int(5);
```

DDL – ALTER

Exemplo:

Excluindo uma coluna:

```
mysql > ALTER TABLE cliente DROP idade;
```

Adicionando e posicionando um campo:

```
mysql > ALTER TABLE cliente ADD  
idade int(3) AFTER id;
```

Modificando a posição de um campo:

```
mysql > ALTER TABLE cliente MODIFY  
idade int(3) FIRST;
```

DDL – DROP

- Permite a exclusão de **TUDO** um banco de dados ou de uma única tabela **INTEIRA** de um BD.
- As situações onde o **DROP** é mais utilizado são:
 1. Exclusão de um **banco de dados**;

Exemplo:

```
mysql > DROP DATABASE banco;
```

```
mysql > DROP DATABASE IF EXISTS banco;
```



DDL – DROP

- Permite a exclusão de **TUDO** um banco de dados ou de uma única tabela **INTEIRA** de um BD.
- As situações onde o **DROP** é mais utilizado são:
 1. Exclusão de um banco de dados;
 2. Exclusão de uma única **tabela**;

Exemplo:

```
mysql > DROP TABLE agencia;
```

```
mysql > DROP TABLE IF EXISTS agencia;
```



Documentação do MySQL

