

# Comandos DDL (Data Definition Language)

MySql

*Prof. Ricardo Satoshi*

# Comandos de Definição das Estruturas de Dados de um BD:

- Os comandos básicos são:
  - **CREATE** : usado para criar bancos de dados, tabelas, visões(views), procedimentos (stored procedures), etc..
  - **ALTER** : usado para alterar alguma característica de uma estrutura previamente criada
  - **DROP** : usado para excluir alguma estrutura criada anteriormente

# Comando CREATE

CREATE DATABASE <nome\_banco> [<especificações>\*]

*\* As especificações são opcionais, e referem-se à características físicas do banco, ou atributos relativos à sua manipulação, tais como taxa de crescimento do banco, caminhos de acesso, permissões, etc*

```
CREATE TABLE <nome_da_tabela>(
    <nome_da_coluna> tipo [tamanho] [<características>],
    ....
);
```

```
CREATE VIEW <nome_da_view> AS
    < comando_select> ....
```

# Create – Exemplo de um script completo

```
CREATE DATABASE Escola;  
USE escola;
```

```
CREATE TABLE TabAlunos(  
    matricula int auto_increment primary key,  
    nome varchar(40) not null,  
    sexo char(1) default 'M',  
    datanasc date,  
    media numeric (5,2) check (media>=0 and media<=10)  
);
```

```
CREATE VIEW AlunosView as  
    Select * from TabAlunos order by nome;
```

# Características que podem ser incluídas na definição do dados

- Auto\_increment  
*numeração automática. Apenas 1 coluna por tabela pode ter esta característica e ela é restrita à chave primária*
- Null / not null  
*permite ou não que a coluna seja nula. Not Null significa que a coluna é requerida*
- Primary key  
*atribuição de chave primária*
- Unique  
*indicador de valor unico*
- Default <valor>  
*define um valor padrão para a coluna que será usado se a coluna for omitida*
- [with] check / [with] nocheck  
*imposição ou anulação de restrições de preenchimento*

# Criando outros objetos

Outros objetos podem ser criados usando-se o comando CREATE, tais como:

- Views ( *tabelas lógicas* )
- Procedures
- Functions
- Triggers
- Index
- etc...

# Alterando uma tabela

Para alterar uma tabela, ou seja, modificar alguma de suas características originais, use o comando:

```
ALTER TABLE [nome_tabela] ....
```

Este comando pode ser usado para :

- **acrescentar** alguma nova coluna à tabela já existente,
- **alterar** características de alguma coluna,
- ou ainda para **eliminar** alguma coluna

# Alterando uma tabela – sintaxe completa

```
ALTER TABLE <nome_tabela>
{
  [{MODIFY <nome_coluna>
    {novo_tipo_de_dados [(precisão[, escala])]} }
  | {CHANGE <nomecoluna> <novonome>
    {novo_tipo_de_dados [(precisão[, escala])]} }
  | ALTER <coluna> SET <atributo> <valor>
  | ADD {nome_coluna_dados} {tipo_de_dado[{precisao[,
    escala])]}
  | [WITH CHECK | WITH NOCHECK]}
  | DROP {<nome_coluna>}
```

*\* \* MODIFY ou CHANGE devem ser usados no lugar do ALTER em alguns casos*



# Alterando uma tabela

Use **MODIFY** para alterar características relativas ao **tipo e/ou tamanho** da coluna

```
ALTER TABLE [nome_tabela]
{
[MODIFY nome_coluna
  {novo_tipo_de_dados [(precisão[, escala])]} ] }
```

Exemplo:

```
ALTER TABLE FUNCIONARIOS MODIFY NOMEFUNC
VARCHAR(40) NOT NULL
```

*Modifica o tamanho da coluna nomefunc para varchar(40) mantendo a condição de not null*

# Alterando uma tabela

Use **CHANGE** para **trocar** o nome de uma coluna por outro nome incluindo o tipo e tamanho da coluna

```
|  
ALTER TABLE [nome_tabela]  
{  
[CHANGE nome_coluna novo_nomecoluna  
  {tipo_de_dados [(precisão[, escala])]}  
|
```

Exemplo:

```
ALTER TABLE FUNCIONARIOS CHANGE NOMEFUNC NOME_F  
  VARCHAR(40) NOT NULL
```

*Troca o nome da coluna **nomefunc** para **nome\_f** mantendo o tipo **varchar(40)** e a condição de **not null***

# Alterando uma tabela – outros exemplos

```
ALTER TABLE TabAluno MODIFY nome varchar(30)
```

*-modificação do tamanho da coluna nome*

```
ALTER TABLE TabAluno ADD notaexame numeric(10,2)
```

*-inclusão de uma nova coluna (notaexame),*

```
ALTER TABLE TabAluno ADD check (media>=0 and  
media<=10)
```

*- acréscimo do atributo check à tabela impondo uma restrição (rule) para a coluna media (previamente criada)*

# Alterando uma tabela (cont)

## Exemplos:

ALTER TABLE TabAluno DROP media  
*-exclusão da coluna media*

ALTER TABLE TabAluno ALTER nome varchar(20) NOT NULL  
*-Alteração do nome incluindo cláusula NOT NULL que indica que a coluna nome não pode ser nula, sendo, portanto, requerida*

ALTER TABLE TabAluno CHANGE nome nomealuno  
varchar(20) NOT NULL  
*-indica que a coluna nome deve ser trocada por nomealuno mantendo suas características de tipo, tamanho e demais atributos*

ALTER TABLE TabAluno ALTER medias set DEFAULT 0  
*-altera a coluna media incluindo o valor default zero*

# Excluindo estruturas de dados

**Para excluir um banco de dados, usa-se:**

```
DROP DATABASE <nome_bancodedados> ;
```

Onde:

nome\_bancodedados é o *nome do banco de dados que se deseja excluir*.

**Exemplo:**

```
DROP DATABASE Exemplo;
```

Para excluir uma **tabela**, ou qualquer outra estrutura já criada por um comando CREATE, usa-se a sintaxe abaixo:

```
DROP <tipo_de_estrutura> <nome_da_estrutura>
```

Exemplos:

```
DROP TABLE Cliente1
```

```
DROP VIEW Visao_Vendas
```

etc...

# Outros comandos de apoio

- Show databases;

*(mostra os bancos de dados do seu servidor mysql)*

- Show tables;

*(mostra as tabelas ou views do banco ao qual se está conectado pelo comando USE)*

- Describe <tabela ou view>;

*(mostra a estrutura da tabela ou view cujo nome foi informado no comando)*

- Rename table <tabela> to <novonome>;

*(renomeia uma tabela )*

# Outros comandos de apoio

- `source <caminho>;`

*(carrega um script sql criado em um editor de textos  
passo a passo)*

Ex: `mysql> source c:/temp/bancox.txt;`

- `tee <caminho>;`

*(registra em um arquivo tipo texto todas as atividades  
de uma sessão do mysql no modo console  
Faz um logging das atividades)*

Ex: `mysql> tee c:/temp/log.txt;`