

# Introdução ao Modelo Físico

- Trata-se da implementação do Modelo Lógico, através da linguagem de consulta estruturada - *Structured Query Language* (SQL);
- São usados os termos tabela, linha e coluna para os termos do modelo relacional formal: **relação**, **tupla** e **atributos**;
- Através desta tecnologia é possível realizar a Criação, Leitura, Atualização e Exclusão dos dados (CRUD)
  - ◆ *CREATE*;
  - ◆ *READ*;
  - ◆ *UPDATE*;
  - ◆ *DELETE*.

## Criando um novo Banco de Dados / Esquema

- É possível criar um novo banco de dados ou esquema, através do comando `CREATE SCHEMA [nome do banco];` ou `CREATE DATABASE [nome do banco];`

Exemplo:

**CREATE SCHEMA** empresa;

ou

**CREATE DATABASE** empresa;

## Deletando um Banco de Dados / Esquema

- É possível deletar um banco de dados ou esquema criado anteriormente, através do comando `DROP DATABASE [nome do banco];` ou `DROP SCHEMA [nome do banco];`

Exemplo:

**DROP DATABASE** empresa;

ou

**DROP SCHEMA** empresa;

## Criando uma nova Tabela

→ Após ter criado o banco de dados, caso ele ainda não esteja selecionado, utiliza-se o comando USE [nome do banco];

Exemplo:

**USE empresa;**

→ Para criar uma nova tabela, utiliza-se o comando

**CREATE TABLE** [nome da tabela] (  
    [descrição dos atributos],  
    [descrição das chaves],  
    [descrição das restrições]

);

Exemplo:

**CREATE TABLE Empregados** (  
    matricula **INT NOT NULL** AUTO\_INCREMENT,  
    nome **VARCHAR(100) NOT NULL**,  
    salario **DECIMAL(10, 2) NOT NULL**,  
    supervisor **INT**,  
    **PRIMARY KEY**(matricula),  
    **FOREIGN KEY**(supervisor) **REFERENCES**(matricula)  
);

## Deletando uma Tabela

→ É possível excluir uma tabela criada anteriormente, através do comando DROP TABLE [nome da tabela];

Exemplo:

**DROP TABLE Pessoas;**

## Tipos de Dados

Numérico	<ul style="list-style-type: none"><li>➤ TINYINT, SMALLINT, INT, MEDIUMINT, BIGINT;</li><li>➤ FLOAT, REAL, DOUBLE;</li></ul>
----------	---

	➤ DECIMAL(i, j).
Texto	➤ CHAR(n), CHARACTER(n); ➤ VARCHAR(n). ➤ TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT.
Lógico	➤ BIT; ➤ BOOLEAN.
Binário	➤ TINYBLOB; ➤ BLOB; ➤ MEDIUMBLOB; ➤ LONGBLOB.
Data e Hora	➤ DATE; ➤ TIME; ➤ YEAR; ➤ DATETIME; ➤ TIMESTAMP.

## Inserir Dados em uma Tabela

→ Para inserir dados em uma tabela, utiliza-se o comando INSERT INTO [nome da tabela] ([campos da tabela]) VALUES ([valores]);

Exemplo:

```
INSERT INTO Pessoas(id, nome, nascimento, sexo, peso, altura, nacionalidade) VALUES (DEFAULT, "Creuza", "1920-12-30", "F", "50.2", "1.65", "Brasil");
```

→ Não é preciso informar os campos antes do VALUES, basta apenas adicionar os valores na ordem dos campos, sendo possível também utilizar o comando desta forma: INSERT INTO [nome da tabela] VALUES ([valores]);

Exemplo:

```
INSERT INTO Pessoas VALUES (DEFAULT, "Adalgiza", "1930-11-2", "F", "63.2", "1.75", "Irlanda");
```

→ Também é possível inserir vários registros de uma só vez da seguinte forma: INSERT INTO [nome da tabela] VALUES ([valores]), ([valores]), ([valores]), ... , ([valores]);

Exemplo:

```
INSERT INTO Pessoas VALUES  
(DEFAULT, "Ana", "1975-12-22", "F", "52.3", "1.45", "EUA"),  
(DEFAULT, "Pedro", "2000-07-15", "M", "87", "2", "Brasil"),  
(DEFAULT, "Maria", "1999-05-30", "F", "75.9", "1.70",  
"Portugal");
```

## Alterando a Estrutura de uma Tabela

→ É possível alterar a estrutura de uma tabela através do comando ALTER TABLE [nome da tabela] e seguido da operação que deseja realizar, algumas são:

- ◆ Adicionar uma nova coluna à tabela;
- ◆ Remover uma coluna da tabela;
- ◆ Modificar o tipo de uma coluna da tabela;
- ◆ Modificar as *constraints* de uma coluna da tabela;
- ◆ Adicionar uma chave primária ou estrangeira à tabela;
- ◆ Renomear a tabela.

### Adicionando uma nova Coluna à Tabela

→ Para se adicionar uma nova coluna à tabela, utiliza-se o comando ALTER TABLE [nome da tabela] ADD COLUMN [nome da coluna] [tipo da coluna] [constraints];

Exemplo:

```
ALTER TABLE Pessoas  
ADD COLUMN profissao VARCHAR(10);
```

→ Também é possível adicionar uma nova coluna após alguma coluna específica da tabela, utilizando o comando ALTER TABLE [nome da tabela] ADD COLUMN [nome da coluna] [tipo da coluna] [constraints] AFTER [nome da coluna anterior];

Exemplo:

**ALTER TABLE** Pessoas

**ADD COLUMN** profissao **VARCHAR(10)** **AFTER** nome;

- Ademais, pode-se adicionar uma nova coluna como primeira coluna da tabela, através do comando **ALTER TABLE** [nome da tabela] **ADD COLUMN** [nome da coluna] [tipo da coluna] [constraints] **FIRST**;

Exemplo:

**ALTER TABLE** Pessoas

**ADD COLUMN** codigo **INT** **FIRST**;

## Removendo uma Coluna da Tabela

- Para remover alguma coluna da tabela, utiliza-se o comando **ALTER TABLE** [nome da tabela] **DROP COLUMN** [nome da coluna];

Exemplo:

**ALTER TABLE** Pessoas

**DROP COLUMN** profissao;

## Modificando Colunas da Tabela

- É possível modificar o tipo e as constraints de alguma coluna da tabela, através do comando **ALTER TABLE** [nome da tabela] **MODIFY COLUMN** [nome da coluna] [novo tipo da coluna] [novas constraints];

Exemplo:

**ALTER TABLE** Pessoas

**MODIFY COLUMN** profissao **VARCHAR(20)** **NOT NULL**  
**DEFAULT ""**;

- É possível modificar o nome de uma coluna, além do tipo e suas constraints, através do comando **ALTER TABLE** [nome da tabela] **CHANGE COLUMN** [nome atual da tabela] [novo nome da tabela] [novo tipo da coluna] [novas constraints];

Exemplo:

```
ALTER TABLE Pessoas  
CHANGE COLUMN profissao prof VARCHAR(20);
```

## Renomeando uma Tabela

→ Pode-se renomear uma tabela utilizando o comando ALTER TABLE [nome da tabela] RENAME TO [novo nome da tabela];

Exemplo:

```
ALTER TABLE Pessoas  
RENAME TO Gafanhotos;
```

## Adicionando Chaves a uma Tabela já Criada

→ Para adicionar uma chave primária a uma tabela já existente, após ter adicionado a coluna que será chave primária a esta tabela, utiliza-se o comando ALTER TABLE [nome da tabela] ADD PRIMARY KEY ([nome da coluna]);

Exemplo:

```
ALTER TABLE Cursos  
ADD COLUMN idcurso INT FIRST;
```

```
ALTER TABLE Cursos  
ADD PRIMARY KEY(idcurso);
```

→ Também é possível adicionar uma chave estrangeira a uma tabela já criada, após ter adicionado a coluna que será chave estrangeira a esta tabela, através do comando ALTER TABLE [nome da tabela] ADD FOREIGN KEY ([nome da coluna]) REFERENCES [nome da tabela de origem da coluna] ([nome da coluna]);

Exemplo:

```
ALTER TABLE Empregados  
ADD COLUMN codigo_departamento INT;
```

```
ALTER TABLE Empregados
```

**ADD FOREIGN KEY**(codigo\_departamento) **REFERENCES**  
**Departamentos**(codigo);

## Manipulando Linhas/Registros

- Além de alterar a estrutura da tabela, com o SQL também é possível modificar as linhas/registros.

### Modificando Linhas/Registros com o UPDATE

- É possível modificar uma linha/registro através do comando  
UPDATE [nome da tabela] SET [nome da coluna] = “[novo valor]”  
WHERE [chave primária] = “[valor da chave primária]”;

Exemplo:

**UPDATE** **Cursos**  
**SET** nome = “HTML5”  
**WHERE** idcurso = “1”;

- Também é possível modificar o valor de mais de uma coluna de uma linha/registro em apenas um comando UPDATE, separando cada campo com uma vírgula;

Exemplo:

**UPDATE** **Cursos**  
**SET** nome = “Java”, ano = “2015”, carga = “40”  
**WHERE** idcurso = “5”;

### Excluindo Linhas/Registros com o DELETE

- Para remover algum registro da tabela, utiliza-se o comando  
DELETE FROM [nome da tabela] WHERE [nome da coluna] = “[nome do valor da coluna]”;

Exemplo:

**DELETE FROM** **Cursos**  
**WHERE** ano = “2018”;

- O comando anterior irá apagar todos os registros da tabela cursos que possuem 2018 como ano; portanto, para evitar conflitos e a exclusão de um registro acidentalmente, o recomendado é que seja usado a chave primária no WHERE;

Exemplo:

```
DELETE FROM Cursos  
WHERE idcurso = "8";
```

- Caso seja necessário, existe um comando específico para excluir todos os registros de uma única tabela; para tal propósito, utiliza-se o comando TRUNCATE [nome da tabela];

Exemplo:

```
TRUNCATE Cursos;
```