

## **Unidade 5: Linguagem SQL**

### **5.1 Linguagem de Definição de Dados (DDL) – Criação de Tabelas (CREATE TABLE)**

**Prof. Rodrigo Baroni**

# SQL (*Structured Query Language*)

- Linguagem padrão dos SGBDs que seguem o Modelo Relacional.
- Há pequenas variações e extensões entre os fornecedores de SGBD, mas a essência é preservada.
- DDL (*Data Definition Language*): parte da linguagem do BD destinada à manutenção das estruturas (tabelas, índices, visões)
- Os comandos para criar, alterar e remover estruturas do BD são:
  - CREATE, ALTER, DROP

# Criação de Tabela (CREATE TABLE)

- Metadados são inseridos no dicionário de dados (catálogo)
- São especificados o nome da tabela e a lista de colunas com os respectivos tipos.
- Há restrições (*constraint*) particulares para algumas colunas e também para a tabela.
- *Constraints* garantem que regras sejam aplicadas aos dados de uma tabela quando linhas são inseridas, apagadas ou modificadas.

```
CREATE TABLE nome_tabela  
  
(nome_coluna tipo_coluna [DEFAULT valor]  
  
  CONSTRAINT nome_constraint tipo_constraint,  
  ...
```

# Tipos de Campos no Microsoft SQL Server



- Lógicos: bit (aceita 0 ou 1)
- Inteiros: tinyint (0 a 255), smallint (até 32.767), int (até 2 bilhões), bigint ( $2^{63}$ )
- Reais: numeric (p,s) ou decimal (p,s), onde p é a precisão (no.total de dígitos) e s é a escala (no. dígitos à direita do ponto decimal)
- Financeiros (2 casas decimais): smallmoney (até \$ 215 mil), money
- Alfanuméricos: **char** (n), **varchar** (n), onde n vai até 800 caracteres
  - Vale a pena usar varchar para  $n \geq 50$ , pois o ganho em armazenamento começa a compensar a ligeira perda em desempenho
- Texto: text  $2^{31}$  caracteres
- Tempo: smalldatetime (precisão de 1 minuto), datetime (precisão de milisegundos)
- Numéricos sequenciais: Identity (no.inicial, incremento)
  - Ex: Create Table Funcionario (Matricula smallint IDENTITY (1,1), ....

# Tipos de dados do Oracle

## SGBD Oracle

<b>VARCHAR2(size)</b>	Cadeia de caracteres com tamanho variável tendo no máximo <i>size</i> bytes. Deve ter especificado o tamanho
<b>CHAR(size)</b>	Dados do tipo character com tamanho fixo, com <i>size</i> bytes. Deve ser usado para dados de tamanho fixo, como por exemplo códigos "J300", "A102", etc.
<b>NUMBER(p,s)</b>	Número tendo, sendo precision p = comprimento do número em dígitos e scale s = posições após o ponto decimal  Inteiro: number(3) Real: number(5,2)
<b>DATE</b>	Tipo data.
<b>Tipo lógico</b>	Usar NUMBER ou CHAR

The Oracle logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

# Tipos de dados no MySQL

## MySQL

<b>Varchar(size)</b>	Cadeia de caracteres de comprimento variável. A cadeia poderá conter desde 0 até 255 caracteres.
<b>Char(size)</b>	Cadeia de caracteres de comprimento fixo.
<b>Int(p)</b>	Inteiro p = comprimento do número em dígitos
<b>Float(p,s)</b>	Real precision p = comprimento do número em dígitos e scale s = posições após o ponto decimal
<b>Date</b>	Armazenamento de ano-mes-dia
<b>Time</b>	Armazenamento de 'HH:MM:SS'
<b>Boolean</b>	0: falso 1: verdadeiro



# Restrição de Chave Primária

- **PRIMARY KEY *constraint*:** garante a integridade de entidade.
- Todas as colunas participantes de uma chave primária devem ser NOT NULL.
- Alguns SGBDs criam automaticamente um índice único (UNIQUE) para a chave primária.



# Exemplo de SQL-DDL

```
CREATE TABLE Funcionario (  
    Matric INT NOT NULL, Nome CHAR(30) NOT NULL,  
    Salario MONEY NOT NULL, Cargo CHAR(15) DEFAULT 'Analista',  
    Estado CHAR(2) NOT NULL, Cod_Depto SMALLINT NULL,  
    CONSTRAINT PK_Funcionario PRIMARY KEY (Matric),  
    CONSTRAINT UQ_Nome UNIQUE (Nome),  
    CONSTRAINT FK_Func_Depto FOREIGN KEY (Cod_Depto) REFERENCES  
    Departamento (Cod_Depto),  
    CONSTRAINT checkestado CHECK (Estado IN ('MG', 'RJ', 'SP'))
```



# Restrição de Unicidade

- **UNIQUE *constraint*:** como uma tabela possui somente uma chave primária, as chaves alternativas ou candidatas que sejam únicas são implementadas por meio desta restrição.
- Alguns SGBDs criam automaticamente um índice único (UNIQUE) para a restrição UNIQUE.

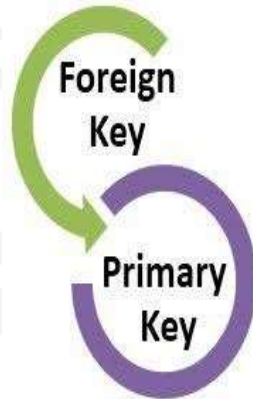


# Exemplo de SQL-DDL

```
CREATE TABLE Funcionario (  
    Matric INT NOT NULL, Nome CHAR(30) NOT NULL,  
    Salario MONEY NOT NULL, Cargo CHAR(15) DEFAULT 'Analista',  
    Estado CHAR(2) NOT NULL, Cod_Depto SMALLINT NULL,  
    CONSTRAINT PK_Funcionario PRIMARY KEY (Matric),  
    CONSTRAINT UQ_Nome UNIQUE (Nome),  
    CONSTRAINT FK_Func_Depto FOREIGN KEY (Cod_Depto) REFERENCES  
    Departamento (Cod_Depto),  
    CONSTRAINT checkestado CHECK (Estado IN ('MG', 'RJ', 'SP'))
```

# Restrição de Chave Estrangeira (FK)

- **FOREIGN KEY *constraint*:** a chave estrangeira pode admitir valores nulos ou valores válidos de uma chave primária da tabela referenciada.
- A restrição de chave estrangeira nunca permitirá o cadastramento de um valor inexistente de chave primária
- A tabela referenciada deve ter a chave primária já criada
- Esta restrição de chave estrangeira usualmente não cria índices automaticamente.



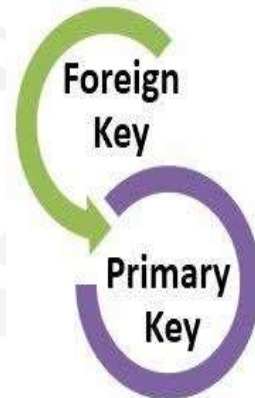
# Integridade Referencial Declarativa

```
[ON DELETE reference_option]
```

```
[ON UPDATE reference_option]
```

*reference\_option*:

```
RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT
```



# Exemplo de SQL-DDL

```
CREATE TABLE Funcionario (  
    Matric INT NOT NULL, Nome CHAR(30) NOT NULL,  
    Salario MONEY NOT NULL, Cargo CHAR(15) DEFAULT 'Analista',  
    Estado CHAR(2) NOT NULL, Cod_Depto SMALLINT NULL,  
    CONSTRAINT PK_Funcionario PRIMARY KEY (Matric),  
    CONSTRAINT UQ_Nome UNIQUE (Nome),  
    CONSTRAINT FK_Func_Depto FOREIGN KEY (Cod_Depto)  
    REFERENCES Departamento (Cod_Depto) ON DELETE RESTRICT,  
    CONSTRAINT checkestado CHECK (Estado IN ('MG', 'RJ', 'SP'))
```

# FOREIGN KEY (FK)

## Tabela pai

```
CREATE TABLE categoria
(id number(3) CONSTRAINT categoria_id_pk PRIMARY KEY,
 nome varchar2(25))
```

## Tabela filha

```
CREATE TABLE produto
(id number(3) CONSTRAINT produto_id_pk PRIMARY KEY,
 nome varchar2(25) CONSTRAINT produto_nome_nn NOT NULL),
 descricao varchar2(25) CONSTRAINT produto_descricao_uk UNIQUE,
 preco number(8,2) CONSTRAINT produto_preco_ck CHECK (preco>50),
 id_categoria number(3) CONSTRAINT produto_id_categoria_fk
REFERENCES categoria(id))
```

# Restrição de *Default*



- **DEFAULT *constraint*:** especifica o valor default que será gravado em uma coluna quando o valor do campo não for informado no momento do INSERT.
- Pode ser feito na frente do campo ou através de Constraint

```
[CONSTRAINT constraint_name]  
    DEFAULT {constant_expression | NULL}  
    [FOR col_name]
```

# Exemplo de SQL-DDL

```
CREATE TABLE Funcionario (  
    Matric INT NOT NULL, Nome CHAR(30) NOT NULL,  
    Salario MONEY NOT NULL, Cargo CHAR(15) DEFAULT 'Analista',  
    Estado CHAR(2) NOT NULL, Cod_Depto SMALLINT NULL,  
    CONSTRAINT PK_Funcionario PRIMARY KEY (Matric),  
    CONSTRAINT UQ_Nome UNIQUE (Nome),  
    CONSTRAINT FK_Func_Depto FOREIGN KEY (Cod_Depto) REFERENCES  
    Departamento (Cod_Depto) ON DELETE RESTRICT,  
    CONSTRAINT checkestado CHECK (Estado IN ('MG', 'RJ', 'SP'))
```



# Restrição de Verificação

- **CHECK *constraint(s)***: define a condição que cada coluna deve satisfazer

[CONSTRAINT *constraint\_name*]  
CHECK (*expression*)



# Exemplo de SQL-DDL

```
CREATE TABLE Funcionario (  
    Matric INT NOT NULL, Nome CHAR(30) NOT NULL,  
    Salario MONEY NOT NULL, Cargo CHAR(15) DEFAULT 'Analista',  
    Estado CHAR(2) NOT NULL, Cod_Depto SMALLINT NULL,  
    CONSTRAINT PK_Funcionario PRIMARY KEY (Matric),  
    CONSTRAINT UQ_Nome UNIQUE (Nome),  
    CONSTRAINT FK_Func_Depto FOREIGN KEY (Cod_Depto) REFERENCES  
    Departamento (Cod_Depto) ON DELETE RESTRICT,  
    CONSTRAINT checkestado CHECK (Estado IN ('MG', 'RJ', 'SP')))
```

# Definição de *Constraint* em Nível de Tabela

- *Constraint* é definida após definição de todas as colunas
- Pode referir a mais de uma coluna (*constraint* que é composta por mais de uma coluna juntas)

```
CREATE TABLE produto
(id number(3),
 nome varchar2(25) CONSTRAINT produto_nome_nn CONSTRAINT NOT NULL,
 descricao varchar2(25),
 preco      number(8,2) ,
 desconto  number(8,2) CONSTRAINT produto_desconto_nn CONSTRAINT NOT NULL,
 id_categoria      number(3),
 CONSTRAINT produto_id_pk PRIMARY KEY (id),
 CONSTRAINT produto_nome_descricao_uk UNIQUE(nome, descricao),
 CONSTRAINT produto_preco_ck CHECK (preco > desconto),
 CONSTRAINT emp_id_categoria_fk FOREIGN KEY (id_categoria)
REFERENCES categoria(id))
```

# ALTER TABLE

Usado para fazer alteração da definição de uma tabela.

## Ações de alteração de tabela:

- Acrescentar coluna
  - A coluna é adicionada no final da tabela.
  - Quando a tabela já possui registros incluídos, essa nova coluna deve aceitar valores nulos ou ser preenchida com *default*
- Excluir coluna
- Alterar definição de coluna
- Acrescentar *constraint* (restrição)
- Excluir *constraint*



SQL  
ALTER  
TABLE

# ALTER TABLE

## Sintaxes

```
ALTER TABLE table  
ADD (column datatype [DEFAULT expr]  
[column constraint],  
[table constraint]);
```

```
ALTER TABLE table  
MODIFY (column datatype [DEFAULT expr]  
[column constraint],  
[table constraint]);
```

```
ALTER TABLE table  
DROP COLUMN column;
```

# ALTER TABLE - Exemplos

## Acrescentar coluna

```
ALTER TABLE empregado  
ADD email varchar2(40);
```

## Excluir coluna

```
ALTER TABLE  
empregado  
DROP COLUMN (email);
```

```
ALTER TABLE empregado  
DROP COLUMN (email,  
data_nasc);
```

# ALTER TABLE - Exemplos

## Alterar definição de coluna

```
ALTER TABLE empregado  
MODIFY email varchar2(50) ;
```

```
ALTER TABLE empregado  
MODIFY status DEFAULT 1 ;
```

```
ALTER TABLE empregado  
MODIFY status DEFAULT  
null ;
```



SQL  
ALTER  
TABLE

# ALTER TABLE - Exemplos

## **Acrescentar *constraint***

```
ALTER TABLE empregado  
ADD CONSTRAINT empregado_numdep_fk  
FOREIGN KEY(numdep) REFERENCES  
departamento(numero) ;
```



SQL  
ALTER  
TABLE



# DROP TABLE

- O comando DROP TABLE é bastante radical, pois elimina a estrutura da tabela e os registros da mesma.
- Em alguns SGBDs, ao se eliminar uma tabela, todas as estruturas relacionadas a mesma (visões, índices) são também excluídos automaticamente.
- Quais as diferenças entre DROP TABLE e DELETE ?
  - Delete é comando DML, Drop Table é DDL
  - Delete exclui o conteúdo da tabela
  - Drop Table exclui todo o conteúdo e também a estrutura
- Ex: DROP TABLE Funcionarios





**PUC Minas**  
**Virtual**