

# Banco de Dados

- **Aula 09: Linguagem SQL: Introdução e Construção de Tabelas**

**Prof.: Clayton Maciel Costa**  
clayton.maciel@ifrn.edu.br

# Agenda

- Linguagem SQL:
  - Introdução;
  - Construção/Definição de tabelas.

# Linguagem SQL

- SQL - *Structured Query Language*;
- Foi definida nos laboratórios de pesquisa da IBM em San Jose, California, em 1974;
- Teve seus fundamentos no modelo relacional;
- Sua primeira versão recebeu o nome de SEQUEL - *Structured English Query Language*.

# Linguagem SQL

- Órgãos como ANSI e ISO adotaram a SQL como o padrão oficial de linguagem em ambiente relacional;
- O ANSI publicou as padronizações SQL ANSI-89 e ANSI-92;
- Revisões da SQL: SQL99 (SQL 3).

# Linguagem SQL

- É uma linguagem usada em SGBDs para:
  - Definir estruturas de dados; (Ex: criar tabelas)
  - Modificar dados no BD; (Ex: inserir e alterar dados)
  - Especificar restrições de segurança; (Ex: privilégios de acesso)
  - Realizar consultas. (Ex: consultar o nome de uma pessoa)
  
- Não é uma linguagem *case-sensitive*.

# Linguagem SQL - Características

- **Definição de Dados:** através da DDL possibilita a definição da estrutura e organização dos dados;
- **Manipulação de Dados:** através da DML possibilita a manipulação dos dados armazenados, compreendendo inclusão, consulta, alteração e eliminação;
- **Controle de Acesso:** protege os dados de manipulações não autorizadas, através de comandos de autorização de acesso;
- **Integridade dos Dados:** define as regras de integridade dos dados contra corrupções, inconsistências e falhas do sistema;

# Linguagem SQL - Características

- **Independência de fabricante:** está incorporada em quase todos os SGBDs em seu padrão ANSI, com as extensões proprietárias de cada fabricante;
- **Portabilidade entre computadores:** pode ser usada desde em um PC como até em um celular;
- **Facilidade no entendimento:** oferece um rápido entendimento, com comandos escritos em um inglês estruturado de alto nível.

# Banco de Dados Exemplo

- Considere as tabelas abaixo de uma locadora de DVD:  
**CLIENTES** (codc, nome, cpf, data\_nasc, sexo, salario);  
**DVD** (codd, titulo, genero, duracao, situacao);  
**LOCACOES** (codc, codd, data).
- Importante: **os nomes das tabelas e dos atributos não podem conter acentos ou espaços em branco.**



# Criando Tabelas - Regras:

- O nome de uma tabela em um banco de dados deverá ser único para cada proprietário;
- Cada coluna deverá ser criada através da especificação do seu nome, tipo e tamanho do dado que irá armazenar;
- O nome de uma coluna deverá ser único dentro de cada tabela podendo, entretanto, pode existir colunas com o mesmo nome em tabelas diferentes.

# Criando Tabelas - Sintaxe

```
CREATE TABLE tabela(  
    atributo1 tipo1,  
    atributo2 tipo2,  
    ...,  
    restrições de integridade  
)
```

# Criando Tabelas - Exemplo

```
CREATE TABLE CLIENTES (  
    codc numeric(10,0) NOT NULL,  
    nome varchar2(80) NOT NULL,  
    cpf numeric(10,0) NOT NULL,  
    data_nasc date,  
    sexo char(1),  
    salario numeric(6,2),  
    PRIMARY KEY (codc)  
)
```

# Criando Tabelas

- **NOT NULL** – indica que o atributo deverá ser obrigatoriamente informado pelo usuário na hora de inserir dados na tabela. (Todos os campos, por *default*, aceitam valores nulos);
- **VARCHAR** (tamanho) – tipo de string com tamanho de armazenamento variável, de acordo com os valores inseridos pelo usuário (o tamanho indicado na criação da tabela é a quantidade máxima de caracteres que poderão ser armazenados no campo).

# Criando Tabelas

- **Char** (tamanho) – tipo de String com tamanho de armazenamento fixo, de acordo com a quantidade de caracteres definidos na criação da tabela;
- **Date** – tipo data no formato ano-mes-dia;
- **Real(n,d)** – tipo numérico que aceita valores reais (n indica o total de números e d indica a quantidade de decimais);
- **Primary Key** – restrição de integridade que define a chave primária da tabela (se a chave for composta, os nomes devem ser separados por vírgulas).

# Criando Tabelas

```
CREATE TABLE DVD (  
    codd numeric(10,0) NOT NULL,  
    titulo varchar2(40) NOT NULL,  
    genero varchar2(15),  
    duracao timestamp,  
    situacao varchar2(12),  
    PRIMARY KEY (codd)  
)
```

# Criando Tabelas

- **FOREIGN KEY** – restrição de integridade que define uma chave estrangeira para a tabela.
- SINTAXE: **FOREIGN KEY (atributo1) REFERENCES tabela (atributo2)**
  - **Por exemplo: FOREIGN KEY (codc) REFERENCES clientes (codc)**
- **Importante:**
  - Para que um atributo seja chave estrangeira de uma tabela, é **necessário que ele seja chave primária da tabela referenciada;**
  - O nome do campo na tabela que terá a chave estrangeira não precisa ser o mesmo nome do campo na tabela referenciada (Exemplo: o campo Codd em locacoes poderia ser substituído por Coddvd), **mas deve ser do mesmo tipo.**

# Criando Tabelas

```
CREATE TABLE LOCACOES (  
    codc numeric(10,0) NOT NULL,  
    codd numeric(10,0) NOT NULL,  
    data date,  
    PRIMARY KEY (codc, codd),  
    FOREIGN KEY (codc) REFERENCES clientes (codc),  
    FOREIGN KEY (codd) REFERENCES dvd (codd)  
)
```



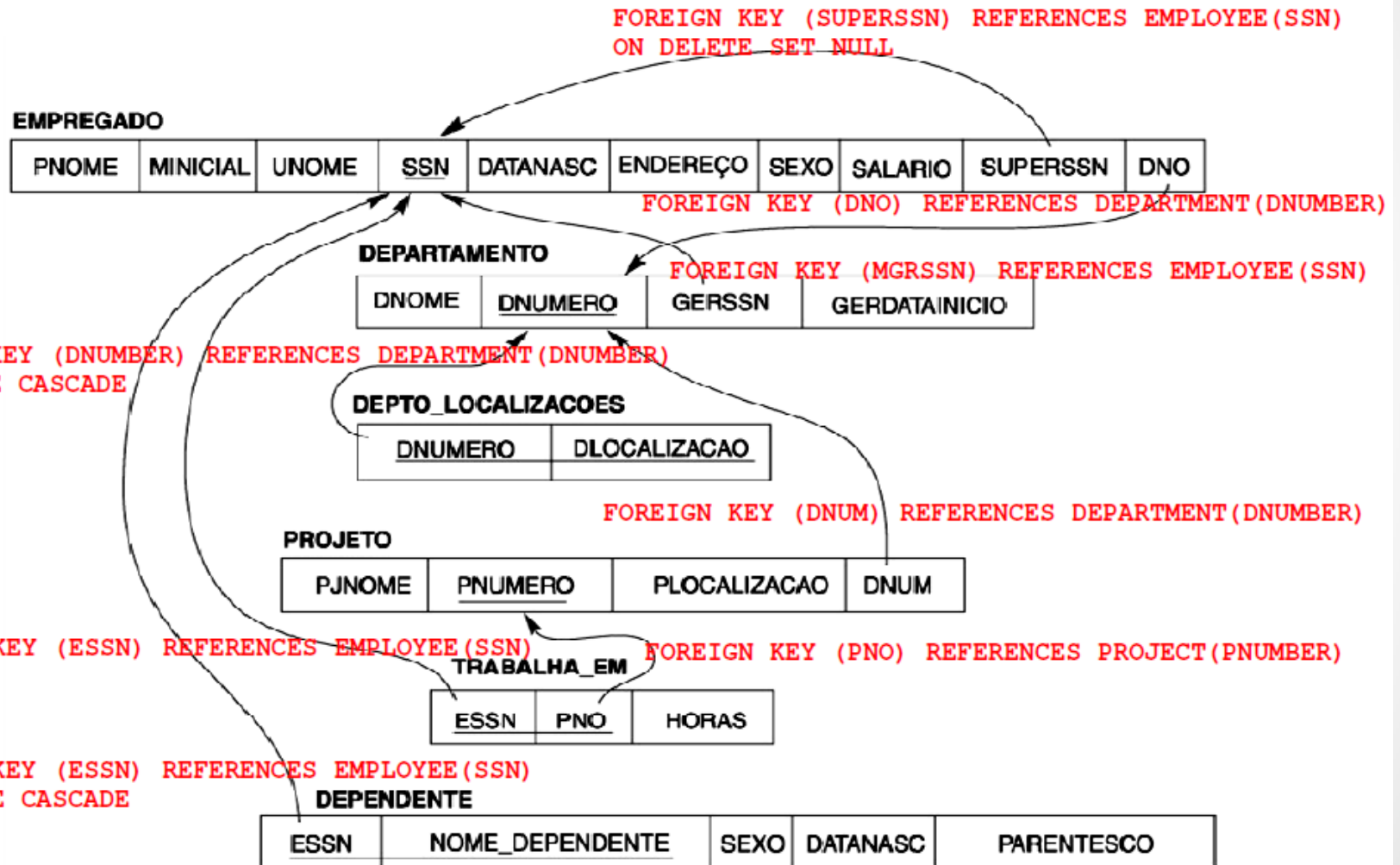
# Criando Tabelas - RIR

- Restrições de Integridade Referencial:
  - Remoção (cláusula **ON DELETE**):
    - CASCADE (propagação);
    - SET NULL (substituição por nulos);
    - Opção default: bloqueio (RESTRICT);

# Criando Tabelas

```
CREATE TABLE LOCACOES (  
    codc numeric(10,0) NOT NULL,  
    coddvd numeric(10,0) NOT NULL,  
    data date,  
    PRIMARY KEY (codc, coddvd),  
    FOREIGN KEY (codc) REFERENCES clientes (codc)  
    ON DELETE CASCADE,  
    FOREIGN KEY (coddvd) REFERENCES dvd (codd)  
    ON DELETE SET NULL  
)
```

# Criando Tabelas

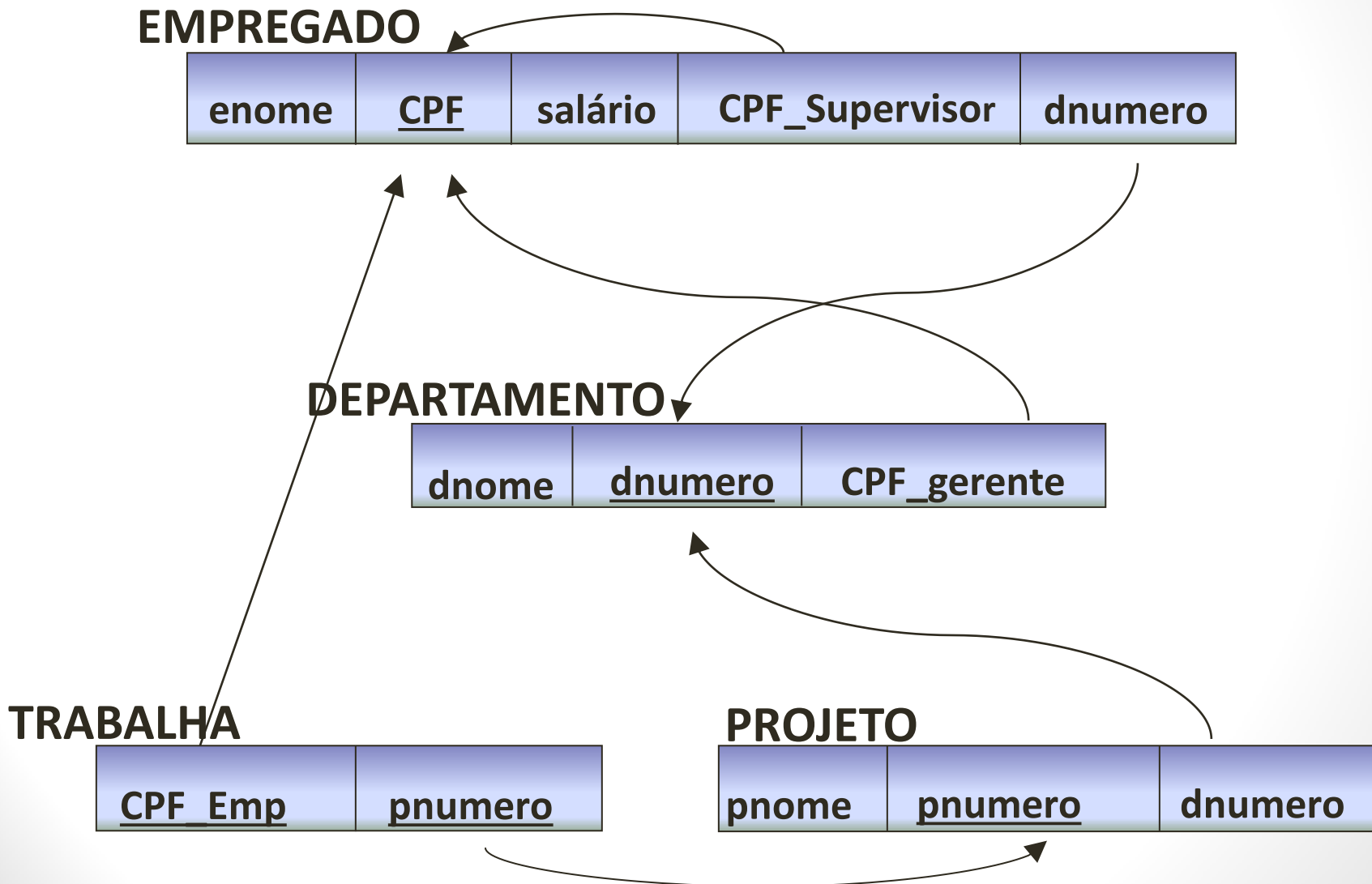


# Editando Tabelas

- **Eliminar** uma tabela:
  - Comando **DROP TABLE**:
    - **DROP TABLE nome\_tabela;**
  
- **Alterar** uma tabela:
  - Comando **ALTER TABLE**:
    - **ALTER TABLE EMPREGADO ADD GRATIFICACAO REAL(6,2);**
    - **ALTER TABLE EMPREGADO DROP GRATIFICACAO;**

# Exercício de Fixação

Fazer o esquema relacional no Oracle 10g



# FIM