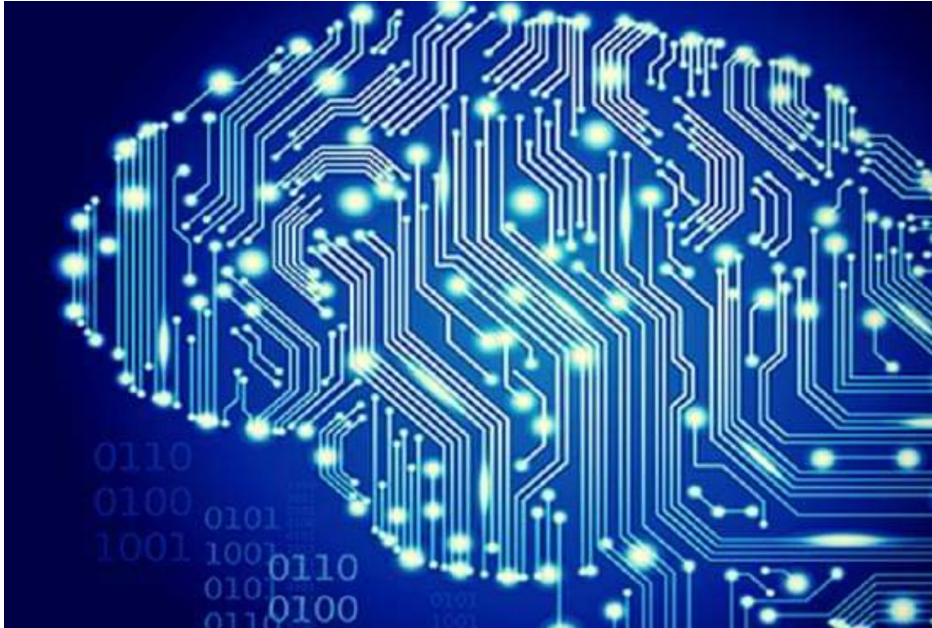




# Programação em Banco de dados

Wesley Neves

# Links da Turma



PROGRAMAÇÃO EM BANCO DE DADOS

Link :

<https://meet.google.com/qjc-xkhd-sbq>

Cód. turma: zzs6ep3

<https://meet.google.com/cob-sygp-kea>

wesley.neves@unidesc.edu.br

# Apresentação

Wesley Neves ,Graduação em  
Sistemas de informação na UDF,

Pós graduação em Banco de dados e  
Business Intelligence -SENAC-DF 2017

Pós graduação em Big data e Analytics  
SENAC-DF 2019

MBA Cloud computing -IGTI cursando.

DBA-SQL Server e gerenciamento de  
Cloud computing



<https://www.linkedin.com/in/wesleyenneves/>

Wesley Neves

05

## Revisão VA1

# Reflexão



L2 -Lista de comandos DDL

Data de entrega: Hoje

Item postado em 5 de set.

4

Entregues

20

Trabalhos atribuídos



L2.Exercicios DDL.docx  
Word

Ver atividade

## Reflexão

O tamanho do seu  
esforço determinará  
a qualidade da sua  
recompensa.

“ PENSADOR

Fontenele

Wesley Neves

# Reflexão

## SQL Developer - US\$7000 /month (remote)



Revelo Brasil · Fortaleza, Ceará, Brasil (Remoto) há 2 dias · **9 candidaturas**



Tempo integral · Pleno-sênior



201-500 funcionários · Recrutamento e seleção



Veja como você se compara a 9 candidatos. [Experimente o Premium grátis](#)



## Dados x Informações



### Dados x Informação

#### DADOS

**O dado não possui significado relevante e não conduz a nenhuma compreensão.** Representa algo que não tem sentido a princípio. Portanto, não tem valor algum para embasar conclusões, muito menos respaldar decisões.

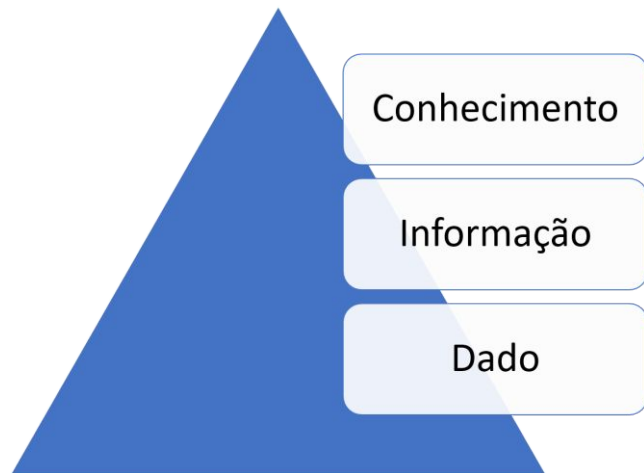
#### INFORMAÇÃO

A informação é a ordenação e organização dos dados de forma a **transmitir significado e compreensão dentro de um determinado contexto.**





# Dados x Informação x Conhecimento



## CONHECIMENTO

**Os dados geram informação, que por sua vez fornece o conhecimento.** O

conhecimento é sempre um saber prático, pois ele foi vivenciado, testado e experimentado, independente das informações ou dados que lhe compõem.



# Dados x Informação x Conhecimento



- Sabedoria: **o mais alto grau desta pirâmide**, refere-se não somente à interpretação do processo em si, mas também na criação de insights de valor e crenças, ou seja, como otimizar um processo, o local que essa otimização deverá ocorrer, as vantagens e ganhos dessa otimização, etc.



## AS CARACTERÍSTICAS DOS DADOS

Compartilhamento : **Os dados devem poder ser compartilhados pelos diversos interessados, usuários**, departamentos e sistemas de informação

Disponibilidade : **Os dados devem estar disponíveis para quem estiver interessado**, seja para uma simples consulta ou relatório ou para uma análise detalhada feita por um tomador de decisão.

Segurança : **Os dados precisam estar seguros**, protegidos contra destruição, alteração ou uso não autorizado.



## AS CARACTERÍSTICAS DOS DADOS

Precisão : **Os dados devem ser absolutamente confiáveis.** Não deve haver dúvida quanto à precisão e veracidade dos dados armazenados

Atualizados : **Os dados devem estar atualizados** para que seja possível à empresa aproveitar as oportunidades de negócio que possam surgir no mercado



# O Projeto de um Banco de Dados

O fato central nas três fases do projeto de bases de dados é a modelagem do dado e suas propriedades. Existe ainda uma ordem lógica nessas fases. A partir do momento que se levanta os requisitos ou requerimentos do software, pode-se iniciar pela primeira fase até chegar na última.

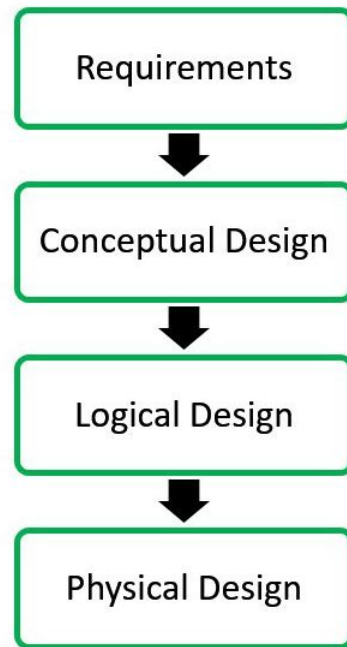


Figura 1.6 — Fases do projeto de base de dados — ordem lógica



# Modelagem de Dados

## I. Modelo Conceitual

Esta é a primeira etapa da modelagem, onde representaremos uma **visão macro dos dados e seus relacionamentos**.

Um modelo conceitual é uma descrição do banco de dados de forma independente de implementação em um SGBD

### Exemplo: Cadastro de Estoque

Dados necessários: Nome do produto, categoria de produto (limpeza, higiene, etc), código do fornecedor, tipo de embalagem, tamanho, quantidade.

\* Neste modelo o projeto é independente do SGBD.



## Modelo Entidade-Relacionamento

Também conhecido pela sigla MER, **trata-se de um modelo conceitual** usado para descrever

objetos envolvidos no domínio de um sistema a ser construído, incluindo seus atributos e relacionamentos.

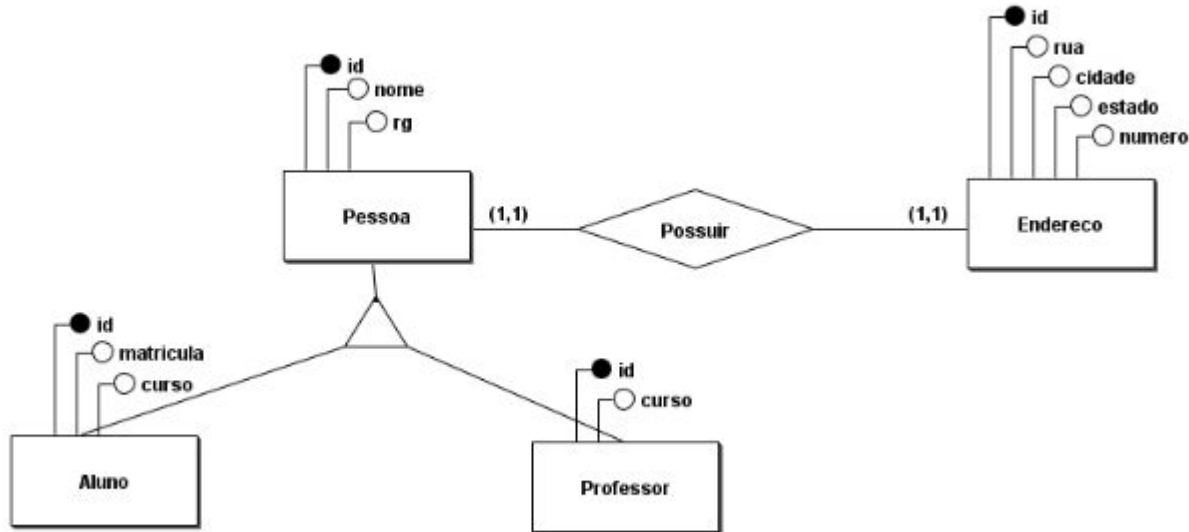
**O modelo de dados ER (MER) não possui ligação nenhuma com SGBD.**

O MER permite representar de forma abstrata a estrutura que irá constituir o banco de dados.

- **É composto pelos seguintes objetos:**
  - Entidades**
  - Atributos**
  - Relacionamentos**
-



# Modelagem de Dados







# Modelagem de Dados

## II. Modelo Lógico

Este modelo é uma visão refinada do modelo conceitual.

Pontua as ligações entre as tabelas de banco de dados, as chaves primárias e estrangeiras, e os componentes de cada uma.

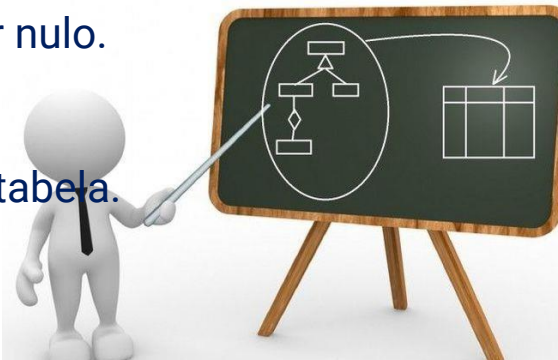
### Chaves primárias (Primary keys ou PK)

Uma PK nunca se repete na mesma tabela, não pode possuir valor nulo.

### Chave externas ou estrangeiras (Foreign Key ou FK)

Uma FK é um campo que faz referência a chave primária de outra tabela.

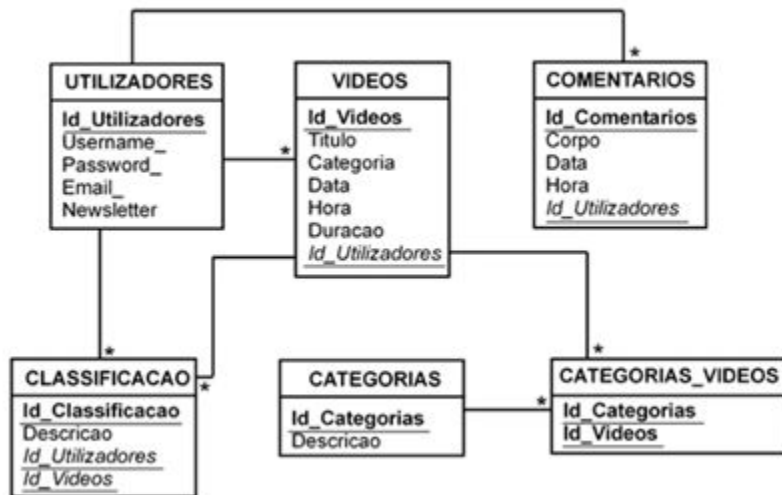
\* Neste modelo o projeto é independente do SGBD.





# Modelagem de Dados

## II. Modelo Lógico





# Modelagem de Dados

## III. Modelo Físico

Demonstra como os dados são fisicamente

```
CREATE TABLE Funcionarios (  
  Fun_Nome VARCHAR(255) NOT NULL PRIMARY KEY,  
  Fun_CPF VARCHAR(15) NOT NULL,  
  Fun_End VARCHAR(255) NOT NULL,  
  Fun_Tel VARCHAR(14) NOT NULL,  
  Fun_Email VARCHAR(255) NOT NULL,  
  Fun_Login VARCHAR(255) NOT NULL,  
  Fun_Senha VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Clientes (  
  Cli_Nome VARCHAR(255) NOT NULL PRIMARY KEY,  
  Cli_CPF VARCHAR(15) NOT NULL,  
  Cli_DtNasc VARCHAR(10) NOT NULL,  
  Cli_End VARCHAR(255) NOT NULL,  
  Cli_Tel VARCHAR(14) NOT NULL,  
  Cli_Email VARCHAR(255) NOT NULL,  
  Cli_Debito REAL NOT NULL  
);
```



# Modelo de Entidade Relacionamento (MER)



## Entidades

Representa tudo aquilo sobre o qual se deseja manter informações.

Ex. Empregado, consulta, fornecedor, departamento, etc.

São representadas graficamente por um retângulo!

Funcionário

Departamento

Fornecedor



# Modelo de Entidade Relacionamento (MER)

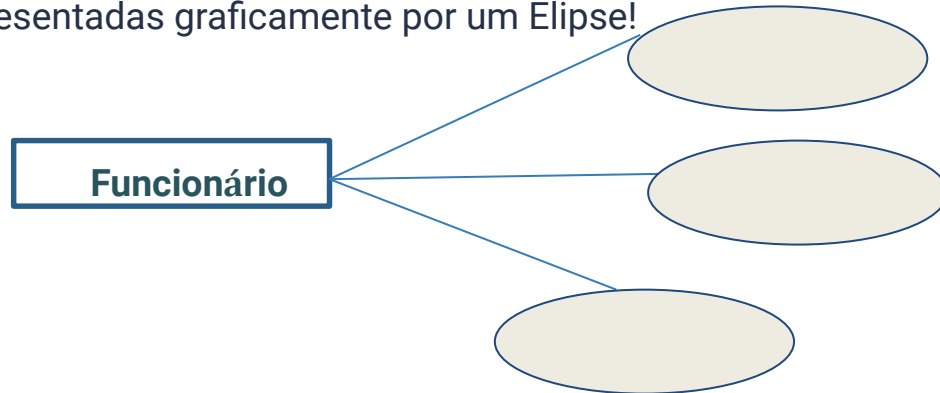


## Atributos

Propriedades que caracterizam e descrevem uma entidade ou um relacionamento.

Ex. A entidade empregado poderia ter os atributos matrícula, nome e função dentro outros.

São representadas graficamente por um Elipse!



\*Possuem um tipo de dados (domínio) nome e valor específico.



# Modelo de Entidade Relacionamento ( MER)



## Atributos

Os atributos podem ser vários tipos, tais como:

- Simples;
- Composto;
- Multivalorado;
- Determinante;
- Identificador;
- Entre outros atributos.



# Modelo de Entidade Relacionamento (MER)



## Relacionamentos

Uma associação **entre duas ou mais entidades cujo significado seja de interesse para a realidade analisada.**

EX: Um cliente compra produtos.

Um filme possui vários atores.

Um empregado trabalha em um departamento.





# Cardinalidade

## Restrições de Cardinalidade

- Especifica o número máximo de instâncias de relacionamentos em que uma entidade pode participar
- Ex: relacionamento Trabalha\_para  
Departamento:Funcionário  
Cardinalidade → 1:N
- Tipos de cardinalidade
  - 1:1, 1:N, N:1 e M:N

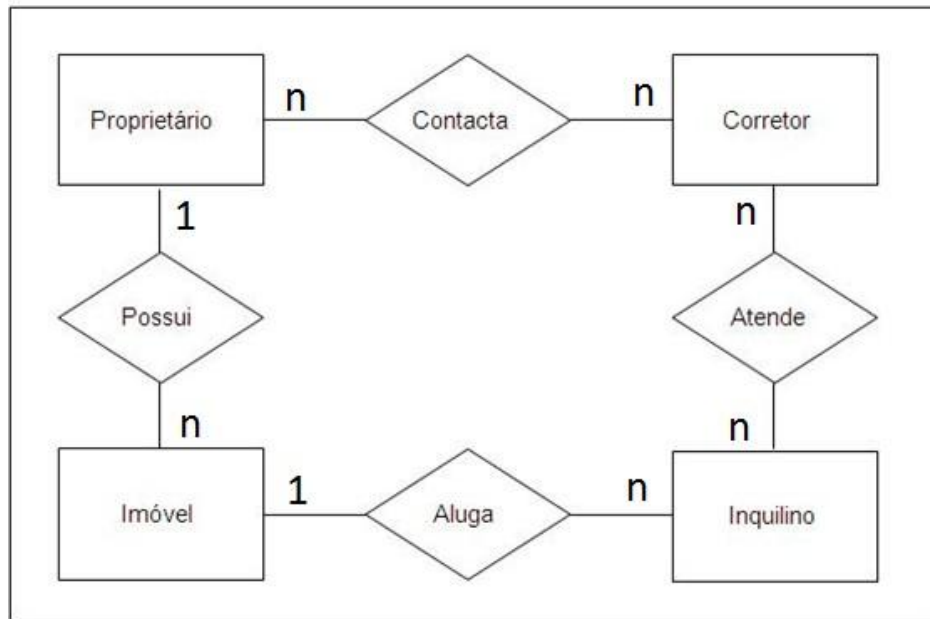
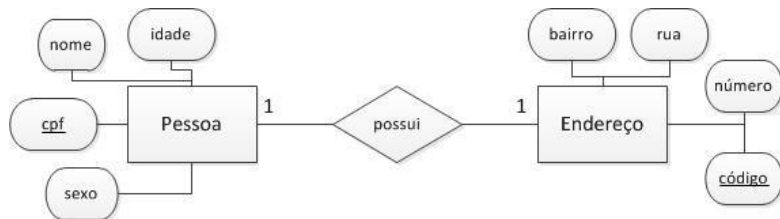




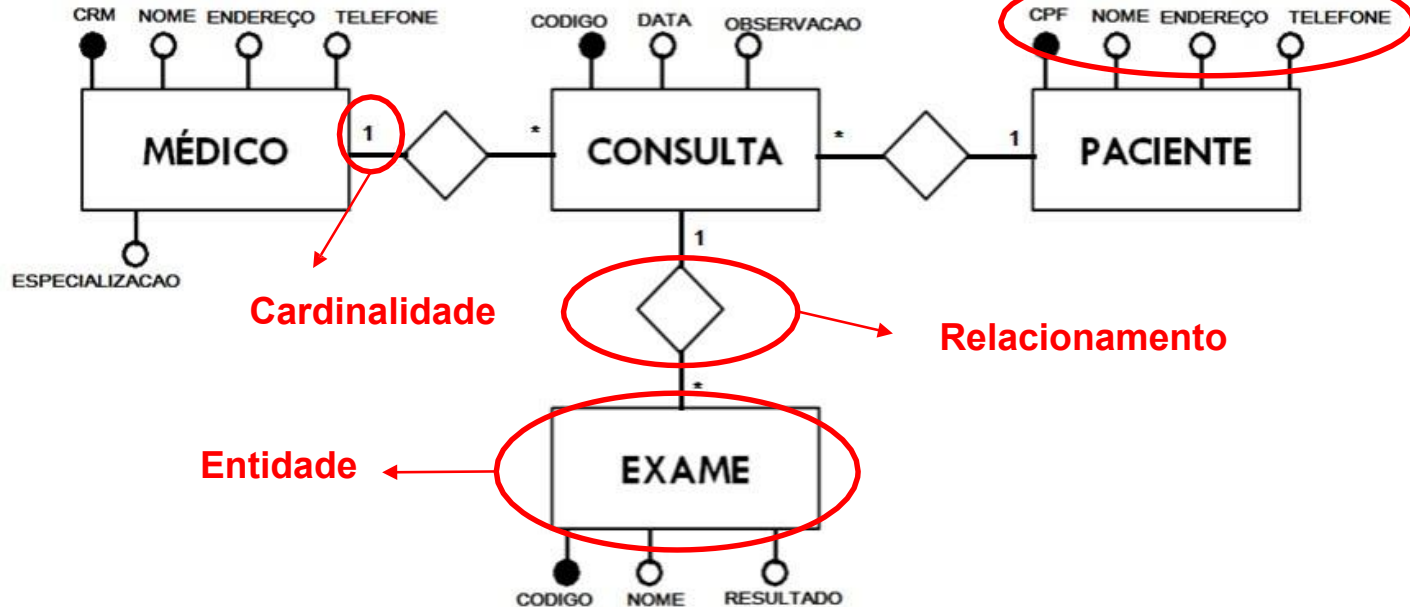
# Cardinalidade

## Cardinalidade

- Um para muitos (1:N)
- Um para um (1:1)
- Muitos para Muitos (N:N)



# Modelo de Entidade Relacionamento (MER)





# LINGUAGENS DE BANCO DE DADOS

## Definição

### DDL

Data Definition  
Language

CREATE

ALTER

DROP

RENAME

TRUNCATE

COMMENT

## Consulta

### DQL

Data Query  
Language

SELECT

## Manipulação

### DML

Data Manipulation  
Language

INSERT

UPDATE

DELETE

MERGE

## Controle

### DCL

Data Control  
Language

GRANT

REVOKE

## Transação

### TCL

Transaction Control  
Language

COMMIT

ROLLBACK

SAVEPOINT

SET TRANSACTION



# LINGUAGENS DE BANCO DE DADOS

## DDL: linguagem de definição de dados

O DDL (Data Definition Language) é o subconjunto SQL que apresenta **comandos usados para gerenciar as estruturas do banco de dados**. Com ele, podemos criar, atualizar e remover objetos da base, como tabelas e índices. Os comandos definidos pelo DDL são: CREATE, DROP e ALTER.



# LINGUAGENS DE BANCO DE DADOS

**DDL** - linguagem de definição de dados (Data Definition Language) permite a criação, eliminação e alteração da estrutura física da Base de Dados. Ela também define os índices (chaves), especifica as ligações entre as tabelas, e impõe restrições entre tabelas. As declarações mais importantes DDL em SQL são:

**CREATE DATABASE** - cria um novo banco de dados;

**CREATE TABLE** - criar uma nova tabela;

**CREATE INDEX** - cria um índice;

**ALTER DATABASE** - altera um banco de dados;

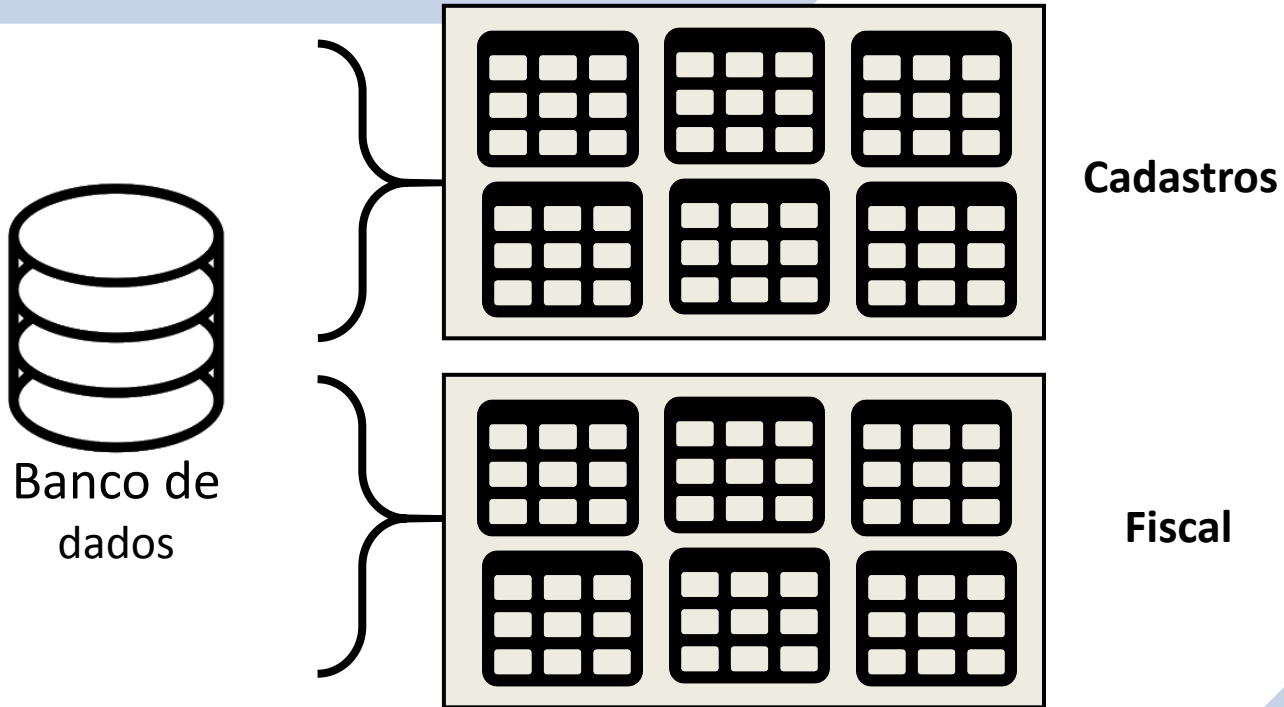
**ALTER TABLE** - modificar uma tabela;

**DROP TABLE** - apaga uma tabela;

**DROP INDEX** - excluir um índice.



# Schema (Esquema)





# Tabelas

A instrução CREATE TABLE a seguir cria a tabela `Employee`.

T-SQL: Criar uma nova tabela

```
CREATE TABLE Employee
(
    EmployeeID int,
    FirstName nvarchar(50) NOT NULL,
    LastName nvarchar(50) NOT NULL,
    EMail nvarchar(50),
    Phone varchar(15),
    HireDate date,
    Salary Money
);
```



# Manipulando a definição de tabelas

## ALTER TABLE

```
ALTER TABLE nome_tabela;
```

- Altera o esquema de uma tabela do BD

- adiciona
- remove
- altera

} colunas ou restrições  
de integridade

)





# ALTER TABLE ADD Colunas

O seguinte adiciona uma nova coluna `Address` de tipo `varchar` e tamanho 500 à `Employee` tabela.

```
ALTER TABLE dbo.Employee  
Add Address varchar(500) NOT NULL;
```

O seguinte adiciona três colunas à `Employee` tabela.

```
ALTER TABLE dbo.Employee  
Add Address varchar(500) NOT NULL,  
    Designation varchar(50) NOT NULL,  
    Qualification varchar(100);
```



## Manipulando a definição de tabela

```
ALTER TABLE [Nome da tabela] ALTER COLUMN [nome da coluna] {Nova definição}
```

**Esse comando altera o domínio do atributo da tabela , mas não altera o nome do atributo**

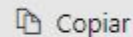


# alterando a definição de tabelas

## a. Renomeando uma tabela

O exemplo a seguir renomeia a tabela `SalesTerritory` como `SalesTerr` no esquema `Sales` .

SQL



```
USE AdventureWorks2012;  
GO  
EXEC sp_rename 'Sales.SalesTerritory', 'SalesTerr';  
GO
```

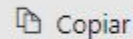


# alterando a definição de tabelas

## B. Renomeando uma coluna

O exemplo a seguir renomeia a `TerritoryID` coluna na `SalesTerritory` tabela como `TerrID`.

SQL



```
USE AdventureWorks2012;  
GO  
EXEC sp_rename 'Sales.SalesTerritory.TerritoryID', 'TerrID', 'COLUMN';  
GO
```