

# DB-SQL-SERVER-1 - Introduction to Relational Databases

---

1. Banco de dados
2. Conceitos SQL
3. MER
4. DER

## 1. Banco de dados

---

Segundo Korth, um banco de dados é "uma coleção de dados inter-relacionado, representando informações sobre um domínio específico", ou seja, sempre que for possível agrupar informações que se relacionam e tratam de um mesmo assunto, posso dizer que tenho um banco de dados. Por exemplo, as informações armazenadas de uma lista telefônica, um registros de usuários em um sistema de e-commerce, um sistema de controle de RH de uma empresa.

Já um Sistema de Gerenciamento de Banco de Dados (SGBD) é um software que possui recursos capazes de manipular as informações do banco de dados e interagir com o usuário. Exemplos de SGBDs são: Oracle, SQL Server, PostgreSQL, MySQL, Mongo DB, Cassandra e etc. O objetivo de um Sistema de Banco de Dados é isolar o usuário dos detalhes internos do banco de dados (promover a abstração de dados) e promover a independência dos dados em relação às aplicações, ou seja, tornar independente da aplicação, a estratégia de acesso e a forma de armazenamento.

## Tipos de Banco de dados

Há dois tipos Bancos de dados no mercado, que são classificados de acordo com a maneira como organizam as informações: **Relacional** e **Não Relacional**.

### Relacionais

Esta forma de organização dos Bancos de dados se caracteriza por organizar as informações em **tabelas** e depende da integração entre **colunas** e **linhas**. A linguagem usada para manipular essas informações é o **SQL** (Structured Query Language). Na imagem abaixo temos alguns exemplos de Banco de dados Relacionais.



## Não Relacionais

O formato Não Relacional atende uma parcela de dados que não podem ser inseridos e acessados através de tabela, como por exemplo, imagens. É muito valorizado pela sua alta performance e por manter todos os registros em um único lugar. Outro uso muito comum do Banco de dados Não Relacional são as áreas de conhecimento como Data Science, devido ao grande volume de dados. Observe que nos Bancos de dados Não Relacionais, não é necessário criar um sistema de relacionamento entre informações, como nos Bancos de Dados relacionais. Como as informações não estão associadas entre si, é mais fácil fazer alterações e exclusões no conteúdo. A linguagem utilizada é o NoSQL (Not Only SQL). Na imagem abaixo temos alguns exemplos de Banco de dados Não Relacionais:



## Diferenças SQL vs NoSQL

SQL	NOSQL
Armazenamento de Dados Estruturados por Tabela	Armazenamento de Dados estruturados e não-estruturados por colunas, grafos, chave-valor e documentos.
Esquema estático	Esquema dinâmico
Maturidade de suporte maior (geralmente pago)	Suporte por comunidade independente (open source)
Escalabilidade vertical	Escalabilidade horizontal
Pago e Gratuito	Gratuito
O desempenho não é alto em todas as consultas. Não suporta pesquisas e cruzamentos muito complexos.	Alto desempenho em consultas
Necessidade de predefinição de um esquema de tabela antes da adição de qualquer dado	Altamente flexível (fácil adição de colunas e campos de dados não estruturados)

## 2. Conceitos SQL

**SQL** significa **Structured Query Language** e é a linguagem padrão utilizada pelos banco de dados relacionais. Ela está dividida em 2 categorias principais:

- **DDL** - Linguagem de Definição de Dados: instruções DDL são responsáveis pela criação, alteração e exclusão dos objetos no banco de dados, ou seja, pela estrutura do Banco e suas Tabelas. Seus principais comandos são: CREATE DATABASE, CREATE TABLE, ALTER TABLE e DROP TABLE;
- **DML** - Linguagem de Manipulação de Dados: instruções DML indicam uma ação para o SGBD executar. Utilizados para recuperar, inserir e modificar um registro no banco de dados. Seus principais comandos são: INSERT, DELETE, UPDATE e SELECT;

## Tipos de Dados

No SQL Server, cada coluna, variável local, expressão e parâmetro tem um tipo de dados relacionado. O tipo de dados é um atributo que especifica o tipo de dados que o objeto pode manter: dados inteiros, dados de caractere, dados monetários, data e hora, cadeiasx de caracteres binárias etc.

### Categorias de tipo de dados:

- **Numéricos exatos**
  - Bigint
  - Numeric
  - Bit
  - Smallint
  - Decimal
  - Smallmoney
  - Int
  - Tinyint
  - Money
- **Cadeias de caracteres Unicode**
  - Nchar
  - Nvarchar
  - Ntext
- **Numéricos aproximados**
  - Float
  - Real
- **Cadeia de caracteres binária**
  - Binary
  - Varbinary
  - Image
- **Data e hora**
  - Date
  - Datetimeoffset
  - Datetime2
  - Smalldatetime
  - Datetime
  - Time
- **Outros tipos de dados**
  - Cursor
  - Rowversion

- Hierarchyid
- Uniqueidentifier
- Sql\_variant
- xml
- Spatial Geometry Types
- Spatial Geography Types
- Table
- **Cadeias de caracteres**
  - Char
  - Varchar
  - Text

Para melhor detalhamento de cada um dos tipos de dados acessar a documentação no [Link](#).

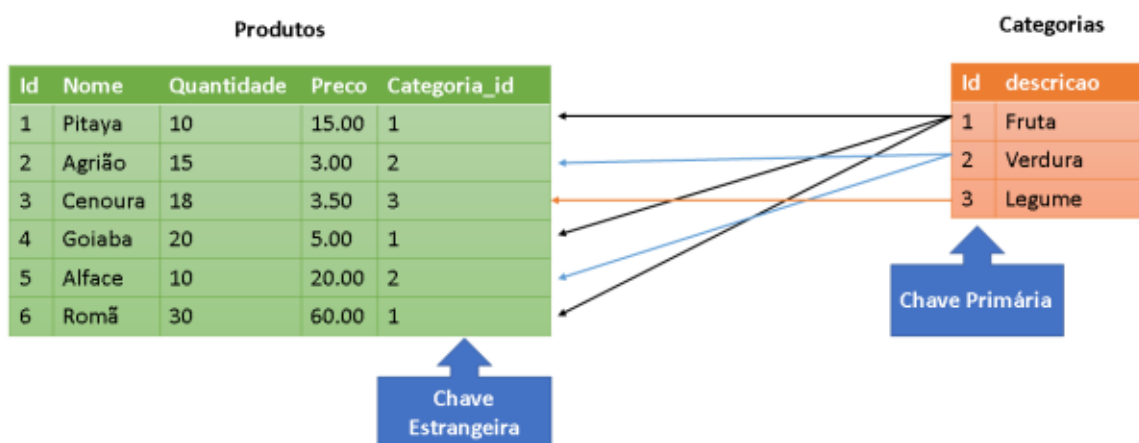
## Chave Primária (PK)

A **chave primária** é o que torna a linha ou o registro de uma tabela únicos. Geralmente, é utilizada uma sequência numérica automática para a geração dessa chave para que ela não venha a se repetir. Nenhuma linha possuirá o mesmo valor na coluna que é chave primária, ou seja, um identificador único do registro.

## Chave Estrangeira (FK)

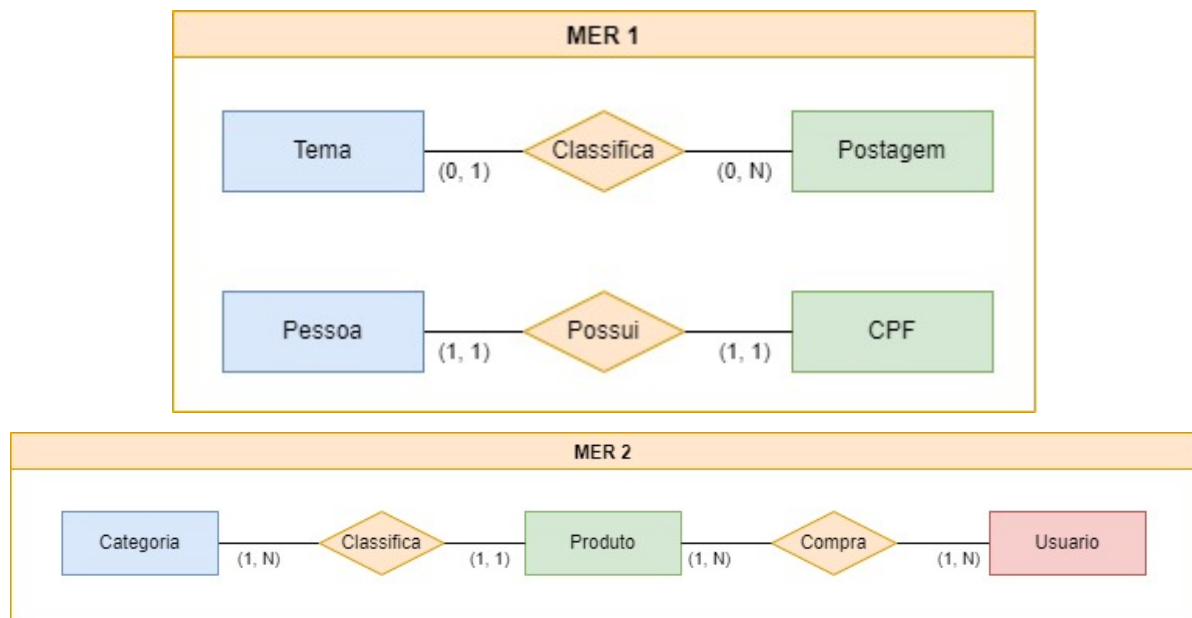
A **chave estrangeira** define um relacionamento entre tabelas, comumente chamado de integridade referencial. Esta regra baseia-se no fato de que uma chave estrangeira em uma tabela é a chave primária em outra. Na imagem abaixo, a tabela produtos tem o campo categoria\_id (Chave Estrangeira), isto é, ela pode se repetir na tabela de produtos. No entanto, deve ser único na tabela de categorias (Chave Primária), pois assim terá uma referência exclusiva.

### Exemplo chave PK e FK em tabelas



## 3. MER

Assim como todos os projetos, devemos iniciar por um **“esboço”** do que de fato será nossa base de dados, quando desenhamos esta estrutura, chamamos de **MER - Modelo Entidade Relacionamento** onde com apenas um papel e uma caneta ou uma tabela no Excel podemos desenhar o modelo do nosso banco de dados expondo todos os relacionamentos entre as tabelas. É possível esboçar um MER utilizando ferramentas também, o conceito é realizar uma pequena definição da tabela, suas características (atributos) e relacionamentos de forma amadora e simples:



É possível demonstrar de diferentes maneiras um sistema, no exemplo de **MER 1**, é possível ver o relacionamento entre 2 entidades de um sistema, tais entidades possuem **cardinalidades** que exemplifica a quantidades de elementos que podem possuir um do outro.

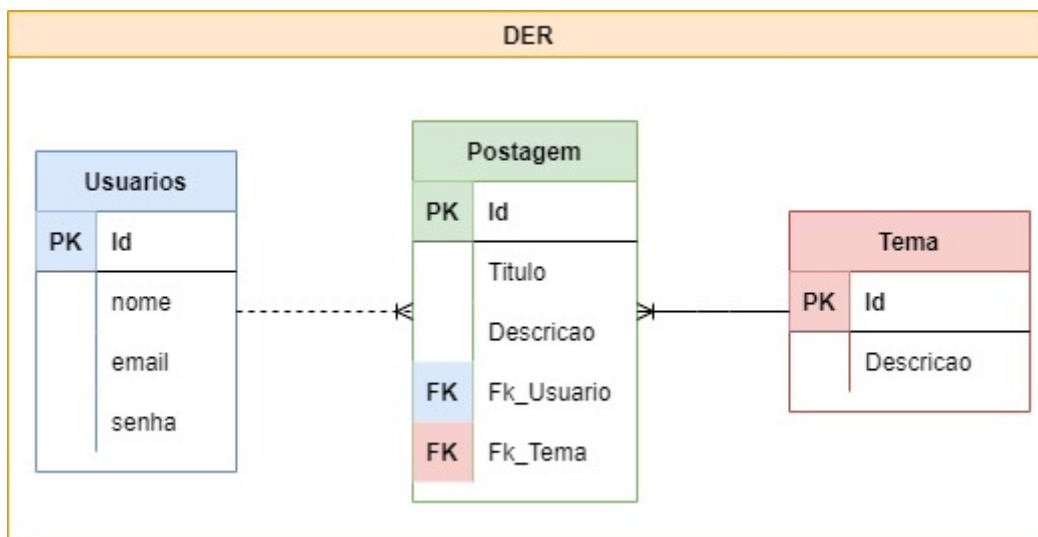
No exemplo **MER 2** é possível ver interações entre as entidades onde uma de suas interações é de **classificação** e outra de **compra**.

## Cardinalidades

- **Relacionamento 1...1 (um para um):** cada uma das duas entidades envolvidas referenciam obrigatoriamente apenas uma unidade da outra. Por exemplo, em um banco de dados de currículos, cada usuário cadastrado pode possuir apenas um currículo na base e cada currículo só pode pertencer a um único usuário cadastrado.
- **Relacionamento 1...n (um para muitos):** uma das entidades envolvidas pode referenciar várias unidades da outra, porém, do outro lado cada uma das várias unidades referenciadas só pode estar ligada a uma unidade da outra entidade. Por exemplo, em um sistema de plano de saúde, um usuário pode ter vários dependentes, mas cada dependente só pode estar ligado a um usuário principal.
- **Relacionamento n...n ou ... (muitos para muitos):** neste tipo de relacionamento cada entidade, de ambos os lados, podem referenciar múltiplas unidades da outra. Por exemplo, em um sistema de biblioteca, um título pode ser escrito por vários autores, ao mesmo tempo em que um autor pode escrever vários títulos. Assim, um objeto do tipo autor pode referenciar múltiplos objetos do tipo título, e vice versa.

## 4. DER

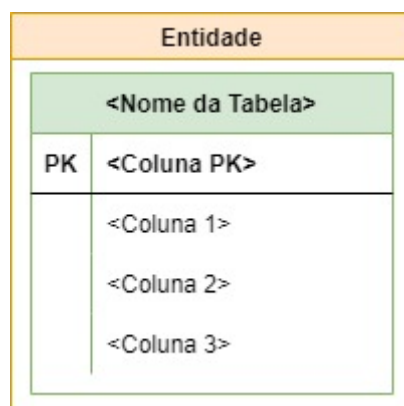
O **DER - Diagrama Entidade Relacional** também pode ser visto como um **“esboço”** de uma forma mais detalhada em comparação ao MER, porém ele é feito através de uma ferramenta específica, onde podemos compartilhar de uma forma segura com os demais membros de nossa equipe, e também extrair do nosso DER o código SQL usado para criar o Banco de dados.



O diagrama acima representa um esboço simples de uma rede social aonde pode ser representada em 3 tabelas com atributos definidos e devidamente relacionadas.

## Entidade

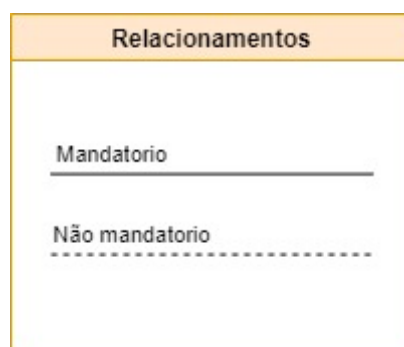
Uma entidade no DER pode ser representada da seguinte maneira:



Uma entidade define uma **tabela**, é possível associar um nome e colocar colunas para exemplificar atributos da tabela. Este conceito é muito importante pois é possível associar o conceito de **Entidade** a uma **Tabela** e em programação á uma **Classe**.

## Relacionamento

Existem diversas maneiras de exemplificar um relacionamento, no entanto deve se entender a ferramenta que se esta utilizando para compreender suas formas. Abaixo segue um exemplo de relacionamento:



Os relacionamentos são expressados de duas maneiras em um DER, uma linha continua para **Mandatório**, onde é necessariamente exista uma dependência entre as entidades e uma linha tracejada para **Não Mandatório**, onde não é necessário que exista uma dependência entre as entidades.

## Cardinalidades no relacionamento

Uma cardinalidade exemplifica uma quantidade para uma expressão. Com ela é possível validar se o relacionamento pode ter vários agentes para uma determinada entidade. É possível expressar as cardinalidades da seguinte maneira:



Para elaborar um diagrama é possível acessar a ferramenta gratuita draw.io no [Link](#).