

# Banco de Dados

**O que é?** Coleção de tabelas relacionadas (ou não) que são geralmente integradas, vinculadas ou referenciadas. Em outras palavras, é onde podemos armazenar dados de maneira organizada, o que além de facilitar o acesso aos dados também torna possível gerar informação através de relacionamentos.

**Banco de Dados Relacional:** Um banco de dados relacional é aquele onde a modelagem dos dados é feita de forma que estes sejam percebidos pelo usuário como tabelas com relações entre si. (Oracle, SQL Server e MySQL).



**Banco de dados não relacional:** Um banco de dados não relacional é um modelo onde não se usa a estrutura de tabela, isto é, de linhas e colunas encontrado na maioria dos sistemas de banco de dados tradicionais.

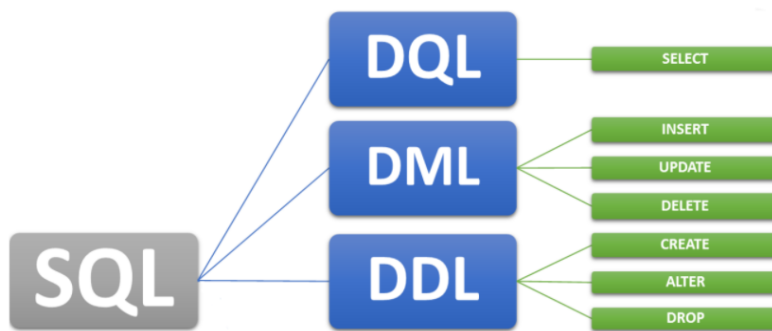
**Modelo Entidade Relacional:** Um modelo entidade relacionamento (modelo ER) é um modelo de dados para descrever os dados ou aspectos de informação de um domínio de negócio ou seus requisitos de processo, de uma maneira abstrata que em última análise se presta a ser implementada em um banco de dados, como um banco de dados relacional.

Modelo	Representação																																				
Modelagem Conceitual	<pre>graph LR; G[Generos] -- 1 --&gt; P{Possui}; P -- N --&gt; F[Filmes]</pre>																																				
Modelagem Lógica	<table border="1"><thead><tr><th colspan="2">Generos</th></tr></thead><tbody><tr><td>PK</td><td>IdGenero</td></tr><tr><td></td><td>Nome</td></tr></tbody></table> <table border="1"><thead><tr><th colspan="2">Filmes</th></tr></thead><tbody><tr><td>PK</td><td>IdFilme</td></tr><tr><td>FK</td><td>IdGenero</td></tr><tr><td></td><td>Titulo</td></tr></tbody></table>	Generos		PK	IdGenero		Nome	Filmes		PK	IdFilme	FK	IdGenero		Titulo																						
Generos																																					
PK	IdGenero																																				
	Nome																																				
Filmes																																					
PK	IdFilme																																				
FK	IdGenero																																				
	Titulo																																				
Modelagem Físico	<table><tr><th colspan="2">Generos</th><th></th><th colspan="3">Filmes</th></tr><tr><th>IdGenero</th><th>Nome</th><th></th><th>IdFilme</th><th>Titulo</th><th>IdGenero</th></tr><tr><td>1</td><td>Ação</td><td></td><td>1</td><td>Rambo</td><td>1</td></tr><tr><td>2</td><td>Romance</td><td></td><td>2</td><td>Vingadores</td><td>1</td></tr><tr><td></td><td></td><td></td><td>3</td><td>Ghost</td><td>2</td></tr><tr><td></td><td></td><td></td><td>4</td><td>Diário de uma paixão</td><td>2</td></tr></table>	Generos			Filmes			IdGenero	Nome		IdFilme	Titulo	IdGenero	1	Ação		1	Rambo	1	2	Romance		2	Vingadores	1				3	Ghost	2				4	Diário de uma paixão	2
Generos			Filmes																																		
IdGenero	Nome		IdFilme	Titulo	IdGenero																																
1	Ação		1	Rambo	1																																
2	Romance		2	Vingadores	1																																
			3	Ghost	2																																
			4	Diário de uma paixão	2																																

**Sistema Gerenciador de banco de dados (SGBD):** É um sistema de manipulação de registros por computador, ou seja, um sistema cujo objetivo global é manter as informações e torná-las disponíveis quando solicitado. Possibilitam o acesso, edição, consulta e inserção de dados no banco. (não são banco de dados, são ferramentas que ajudam na gestão destes).

<i><b>Tipos de Dados</b></i>	
<b>TINYINT</b>	Armazena valores numéricos inteiros, variando de 0 a 256.
<b>SMALLINT</b>	Armazena valores numéricos inteiros, variando de -32.768 a 32.767.
<b>INT</b>	Armazena valores numéricos inteiros, variando de -2.147.483.648 a 2.147.483.647.
<b>BIGINT</b>	Armazena valores numéricos inteiros, variando de -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807.
<b>SMALLMONEY</b>	Valores numéricos decimais variando de -214,748.3648 a 214,748.3647
<b>MONEY</b>	Valores numéricos decimais variando de -922,337,203,685,477.5808 a 922,337,203,685,477.5807
<b>NUMERIC(18,0)</b>	Armazena valores numéricos com casas decimais, utilizando precisão. O primeiro número entre os parênteses, representa a quantidade de inteiros a serem armazenados, o segundo número, indica a quantidade de casas decimais do número.
<b>DECIMAL(18,0)</b>	Tem as mesmas funcionalidades do tipo NUMERIC, a diferença é que o DECIMAL faz parte do padrão ANSI e NUMERIC é mantido por compatibilidade.
<b>FLOAT</b>	Armazena valores numéricos aproximados com precisão de ponto flutuante, variando de -1.79E + 308 a 1.79E + 308
<b>REAL</b>	Armazena valores numéricos aproximados com precisão de ponto flutuante, variando de -3.40E + 38 a 3.40E + 38
<b>BIT</b>	Armazena bits ou seja somente poderá conter os valores lógicos 0 ou 1.
<b>SMALLDATETIME</b>	Armazena data e hora, com precisão de minutos.
<b>DATETIME</b>	Armazena data e hora, com precisão de centésimos de segundos.
<b>TIME</b>	Armazena somente hora. Pode armazenar segundos até a fração de 9999999.
<b>DATE</b>	Armazena somente data.
<b>DATETIME2</b>	É uma combinação dos tipos de dados DATE e TIME. A diferença para o tipo DATETIME é a precisão ao armazenar as horas.
<b>DATETIMEOFFSET</b>	Armazena valores data e hora com a combinação da hora do dia com o fuso horário. O intervalo de deslocamento do fuso horário é de -14:00 a +14:00.

<b>CHAR(N)</b>	Armazena N caracteres fixos (até 8.000) no formato não Unicode. Independente da quantidade de caracteres utilizados, irá sempre armazenar o tamanho de caracteres do campo, sendo preenchido o restante com espaços em branco.
<b>VARCHAR(N)</b>	Armazena N caracteres (até 8.000) no formato não Unicode.
<b>VARCHAR(MAX)</b>	Armazena caracteres no formato não Unicode. MAX indica que o máximo a ser armazenado pode chegar a 2 <sup>31</sup> -1 bytes.
<b>TEXT</b>	Armazena caracteres no formato não Unicode. Esse tipo de dado suporta até 2.147.483.647 caracteres e existem funções específicas para trabalhar com esse tipo de dado.
<b>NCHAR(N)</b>	Armazena N caracteres fixos (até 4.000) no formato Unicode. Independente da quantidade de caracteres utilizados, irá sempre armazenar o tamanho de caracteres do campo, sendo preenchido o restante com espaços em branco.
<b>NVARCHAR(N)</b>	Armazena N caracteres (até 4.000) no formato Unicode.
<b>NVARCHAR(MAX)</b>	Armazena caracteres no formato Unicode. MAX indica que o máximo a ser armazenado pode chegar a 2 <sup>31</sup> -1 bytes.
<b>NTEXT</b>	Armazena caracteres no formato Unicode. Esse tipo de dado suporta até 1.073.741.823 caracteres e existem funções específicas para trabalhar com esse tipo de dado.
<b>BINARY (N)</b>	Armazena dados no formato binário, podendo chegar até 8.000 bytes. Independente da quantidade de dados armazenados, será preenchido com espaços em brancos até completar o tamanho do campo.
<b>VARBINARY(N)</b>	Armazena dados no formato binário, podendo chegar até 8.000 bytes.
<b>VARBINARY(MAX)</b>	Armazena dados no formato binário, podendo chegar até 2 <sup>31</sup> -1 bytes.
<b>IMAGE</b>	Armazena dados no formato binário, podendo chegar até 2,147,483,647 bytes.
<b>SQL_VARIANT</b>	Armazena todos os tipos de dados em um mesmo campo de uma tabela, com exceção dos tipos TEXT, NTEXT, TIMESTAMP e SQL_VARIANT.
<b>TIMESTAMP</b>	Este tipo de dados permite a geração automática de um valor binário para um campo de uma tabela.
<b>UNIQUEIDENTIFIER</b>	Esse tipo de dados é utilizado para a criação de um identificador global e único para uma tabela do SQL Server.
<b>GEOMETRY</b>	Armazena dados espaciais utilizando representação plana.
<b>GEOGRAPHY</b>	Armazena dados espaciais utilizando representação redonda.
<b>HIERARCHYID</b>	É usado para representar uma posição em uma hierarquia. Uma coluna desse tipo não representa automaticamente uma árvore. É até a aplicação para gerar e atribuir valores hierarchyid de tal forma que a relação desejada entre as linhas é refletida nos valores.
<b>XML</b>	Armazena dados no formato XML, não podendo exceder a 2Gb



**DDL:** usada para definir estruturas de dados e modificar dados.

**DML:** é utilizada para a recuperação, inclusão, remoção e modificação de informações em bancos de dados.

**DQL:** realiza consultas aos dados que pertencem a uma determinada tabela.

**DDL (Data Definition Language):** É um conjunto de instruções e comandos para definição de dados. É usada para definir estruturas de dados e modificar dados. Por exemplo, os comandos DDL podem ser usados para adicionar, remover ou modificar tabelas dentro de um banco de dados.

Comando	Representação
<b>CREATE DATABASE</b> (Para criar novos bancos de dados, no caso, está sendo criado o banco de dados "CATALOGO_M")	<pre>CREATE DATABASE CATALOGO_M; GO</pre>
<b>USE</b> (O comando USE define qual banco de dados será utilizado, no caso, o banco usado será o "CATALOGO_M")	<pre>USE CATALOGO_M; GO</pre>
<b>CREATE TABLE</b> (Para adicionar uma nova tabela em um banco de dados)	<pre>CREATE TABLE GENERO ( ); GO</pre>
<b>CHAVE PRIMÁRIA</b> (Original, não depende de informações de outras tabelas, no caso, sempre que houver um registro na GENERO, o registro receberá um idGenero, esse número será uma chave primária, ou seja, todo registro será único)	<pre>CREATE TABLE GENERO (   idGenero TINYINT PRIMARY KEY IDENTITY(1,1), ); GO</pre> <p><small>NOME COLUNA TIPO COLUNA O NÚMERO INSERIDO SERÁ SEMPRE A CHAVE PRIMÁRIA QUANDO HOUVER UM NOVO REGISTRO ACRESCENTAR 1</small></p>
<b>CHAVE ESTRANGEIRA</b> (Depende de informações de outras tabelas, no caso, um filme sempre terá um genero, por isso idGenero na tabela FILME se torna uma chave estrangeira )	<pre>CREATE TABLE FILME (   idFilme SMALLINT PRIMARY KEY IDENTITY(1,1),   idGenero TINYINT FOREIGN KEY REFERENCES GENERO(idGenero), ); GO</pre> <p><small>CHAVE PRIMÁRIA NOME TIPO INFORMAÇÕES DA TABELA GENERO REFERÊNCIA A TABELA E DEPOIS A COLUNA</small></p>
<b>COLONAS DA TABELA</b> (Adicionar as respectivas colunas da tabela)	<pre>CREATE TABLE GENERO (   nomeGenero VARCHAR(30) ); GO</pre> <p><small>NOME COLUNA TIPO COLUNA VALOR</small></p>

**DML (Data Manipulation Language):** Linguagem de manipulação de dados (ou DML, de Data Manipulation Language) é o grupo de comandos dentro da linguagem SQL utilizado para a recuperação, inclusão, remoção e modificação de informações em bancos de dados. Os principais comandos DML são Insert, Update e Delete.

Comando	Representação
<b>INSERT</b> (Para criar INSERIR valores nas tabelas, devemos utilizar o comando INSERT. Nesse caso, queremos inserir uma informação na tabela gênero, na coluna nome. Logo em seguida colocamos os valores que queremos inserir)	<p>Diagrama do comando SQL INSERT INTO GENERO (nomeGenero) VALUES ('AÇÃO'), ('ROMANCE').</p> <ul style="list-style-type: none"> <li>INSERIR UMA INFORMAÇÃO: INSERT</li> <li>NOME DA TABELA: GENERO</li> <li>LINHA DA TABELA: (nomeGenero)</li> <li>VALORES A SEREM INSERIDOS: ('AÇÃO'), ('ROMANCE')</li> </ul>
<b>UPDATE</b> (Para ATUALIZAR dados de uma ou mais tabelas no banco de dados, devemos utilizar o comando UPDATE)	<p>Diagrama do comando SQL UPDATE GENERO SET idGenero = 2 WHERE idGenero = 3.</p> <ul style="list-style-type: none"> <li>ATUALIZAR: UPDATE</li> <li>NOME DA TABELA: GENERO</li> <li>NOME DA COLUNA: idGenero</li> <li>NOVA INFORMAÇÃO: = 2</li> <li>ESPECIFICAR: WHERE</li> <li>A TABELA E O ID: idGenero = 3</li> </ul>
<b>DELETE</b> (Para EXCLUIR dados de uma ou mais tabelas no banco de dados, devemos utilizar o comando DELETE. Nesse caso, queremos deletar na tabela gênero o IDGenero 2.)	<p>Diagrama do comando SQL DELETE FROM GENERO WHERE idGenero = 2.</p> <ul style="list-style-type: none"> <li>DELETAR: DELETE</li> <li>NOME DA TABELA: GENERO</li> <li>ESPECIFICAR: WHERE</li> <li>A TABELA E O VALOR: idGenero = 2</li> </ul>

**DQL (Data Query Language):** Linguagem de consulta de dados.

Comando	Representação
<b>SELECT</b> (Para visualizar as tabelas que construímos, devemos utilizar o comando SELECT)	<p>Diagrama do comando SQL SELECT * FROM PESSOA.</p>