



BANCO DE DADOS

SUMÁRIO

Introdução a Bancos de Dados.....	3
Tipos de Banco de Dados	4
Banco de Dados Relacional	4
Exemplos e prática sobre banco de dados SQL.....	6
Referências.....	19



INTRODUÇÃO A BANCOS DE DADOS

Um dos principais pilares a respeito de bancos de dados é a distinção clara entre dados e informações, portanto, para entendermos o que é um banco de dados, é muito importante compreendermos a diferença entre essas duas palavras.

- **Dados** são fatos brutos, em sua forma primária, e muitas vezes os dados podem não fazer sentido sozinhos. Exemplos: 2021, estudante, linguagem, programação, Java.
- **Informações** consiste no agrupamento de dados de forma organizada para fazer sentido, gerar conhecimento. Exemplo: No ano de 2021, os estudantes iniciaram o curso de programação em linguagem Java.



Um banco de dados é uma estrutura de dados organizada que permite a extração de informações, de forma prática e segura.

TIPOS DE BANCO DE DADOS

Existem dois tipos distintos de Banco de Dados, sendo eles:

• • • • • • •

1. **Banco de Dados relacional (SQL):** representa os dados em um banco de dados através de um conjunto de relações (tabelas). Essas relações contêm informações sobre entidades ou relacionamentos existentes no domínio da aplicação utilizada como alvo para a modelagem;
2. **Banco de Dados não relacional (noSQL):** como o próprio nome sugere, armazena dados que — em teoria — não têm forte relação entre si. Nesse caso, as tabelas dão lugar às coleções de dados, que podem ser armazenadas como:
 - a. pares chave/valor simples;
 - b. documentos *JSON*;
 - c. gráfico em bordas e vértices.

• • • • • • •

Neste e-book, vamos abordar sobre Banco de Dados relacional.

BANCO DE DADOS RELACIONAL

Um **banco de dados relacional** armazena dados em **tabelas**.

Tabelas são organizadas em colunas, e cada coluna armazena um tipo de dados (inteiro, números reais, *strings* de caracteres, data e outros tipos).

Os dados de uma tabela são armazenados como uma **linha**. Por exemplo, a tabela Cliente teria colunas como numeroCliente, primeiroNome e sobrenome, e uma linha na tabela teria algo como {123, "Arilo", "Dias"}.

Tabelas tipicamente possuem **chaves**, uma ou mais colunas que unicamente identificam uma linha na tabela. No caso da tabela Cliente, a chave seria a coluna numeroCliente.

Um banco de dados relacional é composto, de forma simplificada, pelos seguintes elementos:

- **Tabelas**

- Objetos onde são armazenados os dados em um banco de dados relacional.
- Uma tabela é uma coleção de entradas de dados relacionados e consiste em linhas e colunas.

- **Campos**

- São entidades que representam os atributos dos dados, como Nome, Data de Nascimento, Salário, Preço e outros.
- Um campo é uma coluna em uma tabela que mantém informações específicas sobre cada registro.

- **Registro**

- Linha ou Tupla.
- Cada entrada individual em uma tabela. Trata-se de um conjunto de campos relacionados que caracterizam os dados de uma entidade única.





EXEMPLOS E PRÁTICA SOBRE BANCO DE DADOS SQL

Agora é o momento de praticarmos os exemplos sobre banco de dados SQL, utilizando o **SQL Server Express** da **Microsoft**.

Para realizar o download do **SQL Server Express** da **Microsoft**, acesse o link a seguir ou escaneie o código QR:

<https://www.microsoft.com/pt-br/sql-server/sql-server-downloads>



1. **Acessar o Sistema de Gerenciamento do SQL Server Express:** digite na busca rápida o exemplo abaixo:

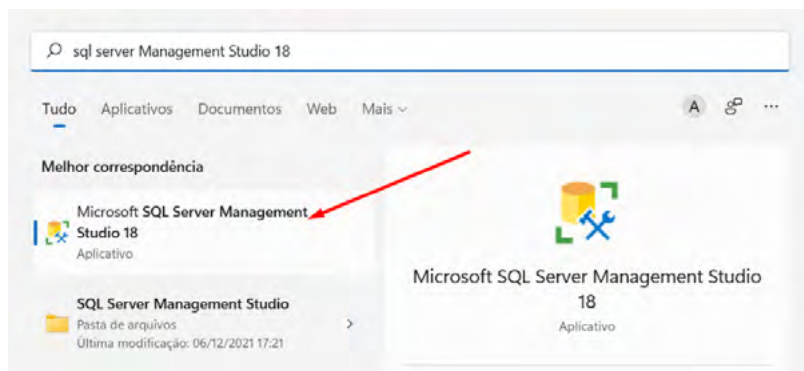


Figura 1 - Como acessar o Sistema de Gerenciamento do SQL Server Express

Fonte: Do autor (2022)

2. **Servidor:** no programa Microsoft SQL Server Management Studio, para conectar com o servidor, clique na opção "Server Name" e escolha algum servidor disponível, em seguida, clique no botão "Connect":

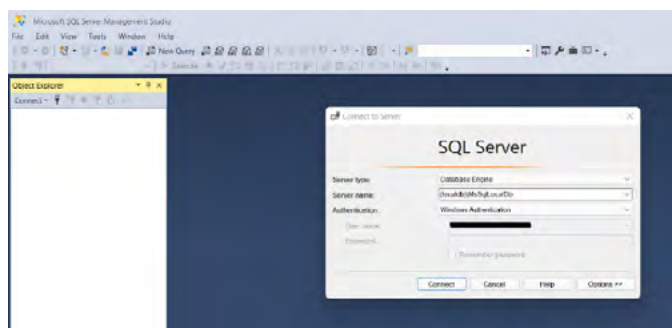


Figura 2 - Conexão com servidor de banco de dados

Fonte: Do autor (2022)

O servidor, também chamado de instância do mecanismo de banco de dados, gerencia vários bancos de dados do sistema de um ou mais bancos de dados de usuários.

- 3. Banco de dados:** com o servidor conectado no item 2, vamos criar um banco de dados no SQL Server para interagirmos com as nossas tabelas e registros. Mas, o que é um banco de dados?

Um banco de dados no SQL Server é composto por uma coleção de tabelas que armazenam um conjunto específico de dados estruturados.

- 3.1. Criar banco de dados:** a sintaxe para criar um banco de dados é:

```
CREATE DATABASE Faculdade;
```

- 3.2. Deletar banco de dados:** a sintaxe para deletar um banco de dados é:

```
DROP DATABASE Faculdade;
```

4. Tabelas

As tabelas contêm todos os dados referentes à um banco de dados, por exemplo: no banco de dados Faculdade, todas as tabelas criadas nele conterão informações registradas somente nesse banco, permitindo a inserção, consulta, atualização e exclusão de seus registros.

Na tabela, os dados são organizados de maneira lógica e dispostos em linha e coluna.

Cada linha representa um registro único e cada coluna representa um campo do registro.

Em um banco de dados, o nome de uma tabela é exclusivo; porém, o nome de uma coluna pode ser utilizado em várias tabelas.

- 4.1. Criar tabela:** na base de dados Faculdade, vamos aprender a sintaxe de criação de tabela:


```
USE [Faculdade]
```

```
GO
```

```
CREATE TABLE [dbo].[Aluno] (  
    [Id] [int] PRIMARY KEY IDENTITY(1,1) NOT NULL,  
    [CPF] [varchar](15) NULL,  
    [Nome] [varchar](150) NOT NULL,  
    [Email] [varchar](50) NOT NULL,  
    [Telefone] [varchar](14) NOT NULL,  
    [DataNascimento] [datetime] NULL  
)  
GO
```

4.2. Editar tabela

É permitido alterar a estrutura de uma tabela, incluindo colunas, chaves estrangeiras, tipo da coluna, entre outros.

Na tabela Aluno, vamos aprender a sintaxe para incluir uma coluna:

```
USE [Faculdade]
```

```
GO
```

```
ALTER TABLE dbo.Aluno add EnderecoCompleto [varchar](200) NULL;
```

4.3. Excluir tabela

Na exclusão de uma tabela, é importante verificar se ela possui relação (chave estrangeira) com outra tabela. Se for o caso, será necessário excluir a chave estrangeira ou a tabela primária desse relacionamento. Em uma tabela que contém apenas a chave primária, a sintaxe de exclusão é:

```
USE [Faculdade]

GO

DROP TABLE [dbo].[Aluno];
```

5. Colunas

As colunas são consideradas os campos de uma tabela. Elas caracterizam os tipos de dados e a quantidade de caracteres que deverão constar em cada campo de uma tabela.

5.1. Editar coluna

Na tabela Aluno, criamos o campo CPF como obrigatório; a sintaxe para alterá-lo para opcional é:

```
USE [Faculdade]

GO

ALTER TABLE dbo.Aluno alter column CPF [varchar](15) NULL;
```

5.2. Excluir coluna

Na tabela Aluno, a sintaxe para excluir a coluna EnderecoCompleto é:

```
USE [Faculdade]

GO

ALTER TABLE dbo.Aluno drop column EnderecoCompleto;
```

6. Tipos de dados nas tabelas

Uma tabela contém, no mínimo, uma ou mais colunas. Cada coluna tem um tipo de dados relacionado.

O tipo de dados é um atributo especificado para cada coluna; os tipos são agrupados da seguinte forma:

- 6.1. Numéricos exatos:** *bigint, decimal, int, money, numeric, smallint, smallmoney, bit e tinyint.*
- 6.2. Numéricos aproximados:** *float e real.*
- 6.3. Data e hora:** *date, datetime, time e smalldatetime.*
- 6.4. Cadeias de caracteres:** *char, text e varchar.*
- 6.5. Cadeias de caracteres Unicode:** *nchar, ntext e nvarchar.*
- 6.6. Cadeia de caracteres binária:** *binary, imagem e varbinary.*
- 6.7. Outros tipos de dados:** *xml, tipos de geografia espacial, uniqueidentifier, cursor, sql_variant, hierarchyid, table e rowversion.*

7. Chave estrangeira

A chave estrangeira (*foreign key*) é um conceito ligeiramente diferente. Não diz respeito, especificamente, a uma tabela, mas sim a um relacionamento entre tabelas. De forma sucinta, a chave estrangeira é uma referência em uma tabela a uma chave primária de outra tabela.

Diferentemente da chave primária, a chave estrangeira:

- › Pode ser nula (**NULL**);
- › É um campo em uma tabela que faz referência a um campo que é chave primária em outra tabela.

Para facilitar a compreensão, vamos criar a tabela Matricula, já adicionando uma chave estrangeira com a tabela Aluno.

```
USE Faculdade

GO

CREATE TABLE [dbo].[Matricula] (
    [Id] [int] PRIMARY KEY IDENTITY(1,1) NOT NULL,
    [DataMatricula] [datetime] NULL,
    [IdAluno] [int] NULL,
    CONSTRAINT fk_Matricula_Aluno FOREIGN KEY (IdAluno
REFERENCES Aluno(Id)
)
GO
```

8. Registros em tabelas

Os registros em uma tabela representam os dados correspondentes aos tipos definidos em cada coluna.

Vamos aprender como inserir, editar e excluir registros.

8.1. Inserir registro

A sintaxe para inserir registro é:

```
INSERT INTO table_name (column1, column2, column3, ...) VALUES
(value1, value2, value3, ...);
```

Vamos inserir um registro na tabela Aluno utilizando a sintaxe acima:

```
USE [Faculdade]
```

```
GO
```

```
INSERT INTO dbo.Aluno (CPF, Nome, Email, Telefone,  
DataNascimento) values '123.456.789-10', 'Maria da Silva',  
'maria.silva@gmail.com', '2020-9898', '1980-04-24');
```

8.2. Alterar registro

Para alterar um determinado registro, é importante informar o Id do registro que se pretende atualizar.

A sintaxe para alterar registro é:

```
UPDATE table_name SET column1 = value1, column2 =value2 WHERE  
condition;
```

Vamos alterar um registro na tabela Aluno utilizando a sintaxe acima:

```
USE [Faculdade]
```

```
GO
```

```
UPDATE dbo.Aluno SET Email = '123.456.789-10' WHERE Id =1;
```


8.3. Exclusão de registro

A exclusão de registro também segue a mesma premissa adotada na alteração de registro, ou seja, é importante informar o Id do registro que se pretende excluir.

A sintaxe para excluir registro é:

```
DELETE FROM table_name WHERE condition;
```

Vamos excluir um registro na tabela Aluno utilizando a sintaxe acima:

```
USE [Faculdade]
GO

DELETE FROM dbo.Aluno WHERE Id =1;
```

9. Seleção de registros

A seleção de registros permite a visualização dos dados inseridos na tabela, permitindo o uso de cláusulas específicas.

Para selecionar as colunas da tabela, a sintaxe é:

```
SELECT column1, column2, column3 FROM table_name;
```

Vamos utilizar a instrução acima na tabela Aluno:

```
SELECT [Id], [CPF], [Nome], [Email], [Telefone], [DataNascimento]
FROM [Faculdade].[dbo].[Aluno];
```

9.1. Cláusula **WHERE**

Utilizada para filtrar linhas da tabela. A sintaxe é:

```
SELECT column1, column2, column3 FROM table_name WHERE condition;
```

Vamos utilizar a instrução acima na tabela Aluno:

```
SELECT [Id], [CPF], [Nome], [Email], [Telefone], [DataNascimento]
FROM [Faculdade].[dbo].[Aluno] WHERE Id =1;
```

9.2. Cláusula **AND**

Utilizada para filtrar registros, caso todas as condicionais sejam verdadeiras.

A sintaxe é:

```
SELECT column1, column2, column3 FROM table_name WHERE
conditionA AND conditionB;
```



Vamos utilizar a instrução acima na tabela Aluno:

```
SELECT [Id], [CPF], [Nome], [Email], [Telefone], [DataNascimento]
FROM [Faculdade].[dbo].[Aluno] WHERE Nome like '%aluno%' AND
like '%9898%';
```

9.3. Cláusula **OR**

Utilizada para filtrar registros, caso uma das condicionais seja verdadeira.

A sintaxe é:

```
SELECT column1, column2, column3 FROM table_name WHERE
conditionA OR conditionB;
```

Vamos utilizar a instrução acima na tabela Aluno:

```
SELECT [Id], [CPF], [Nome], [Email], [Telefone], [DataNascimento]
FROM [Faculdade].[dbo].[Aluno] WHERE Nome like '%aluno%' OR like
'%9898%';
```

9.4. Cláusula **LIKE**

Utilizada para filtrar registros na tabela, utilizando uma cadeia de caracteres específica.

A sintaxe é:

```
SELECT column1, column2, column3 FROM table_name WHERE column1
LIKE condition;
```

Vamos utilizar a instrução acima na tabela Aluno:

```
SELECT [Id], [CPF], [Nome], [Email], [Telefone], [DataNascimento]
FROM [Faculdade].[dbo].[Aluno] WHERE Nome like '%aluno%';
```

9.5. Cláusula **NOT LIKE**

Nesta cláusula, a comparação se inverte; o resultado que será apresentado não corresponde à cadeia de caracteres especificada.

A sintaxe é:

```
SELECT column1, column2, column3 FROM table_name WHERE column1
NOT LIKE condition;
```

Vamos utilizar a instrução acima na tabela Aluno:

```
SELECT [Id], [CPF], [Nome], [Email], [Telefone], [DataNascimento]
FROM [Faculdade].[dbo].[Aluno] WHERE Nome NOT like '%aluno%';
```

9.6. Cláusula **BETWEEN**

Retorna a consulta dos dados de uma tabela, de acordo com o intervalo de dados especificados no filtro.

A sintaxe é:

```
SELECT column1, column2, column3 FROM table_name WHERE column1
BETWEEN conditionA WHERE conditionB;
```

Vamos utilizar a instrução acima na tabela Aluno:

```
SELECT [Id], [CPF], [Nome], [Email], [Telefone], [DataNascimento]
FROM [Faculdade].[dbo].[Aluno] WHERE DataNascimento BETWEEN like
'%aluno%';
```

9.7. Cláusula **ORDER BY**

Por padrão, os registros são exibidos pela ordem crescente de inserção.

Porém, é possível definir que a ordenação do resultado seja estabelecida por outra coluna da tabela, crescente (**ASC**) ou decrescente (**DESC**).

A sintaxe é:

```
SELECT column1, column2, column3 FROM table_name ORDER BY
column2 ASC;
```

Vamos utilizar a instrução acima na tabela Aluno:

```
SELECT [Id], [CPF], [Nome], [Email], [Telefone], [DataNascimento]
FROM [Faculdade].[dbo].[Aluno] ORDER BY Nome DESC;
```


REFERÊNCIAS

FUNDAÇÃO BRADESCO. **Modelagem de Dados**. Curso da Fundação Bradesco | Escola Virtual, São Paulo (SP), 2022. Disponível em: <https://www.ev.org.br/cursos/modelagem-de-dados>. Acesso em: 28 ago. 2022.

MICROSOFT. **Experimente o SQL Server na infraestrutura local ou na nuvem** [Página web: SQL Server > Downloads]. Microsoft. Redmond (WA, USA), 2022. Disponível em: <https://www.microsoft.com/pt-br/sql-server/sql-server-downloads>. Acesso em: 28 ago. 2022.

DIAS-NETO, A. C. **Bancos de Dados Relacionais**. Portal DevMedia, 2011. Disponível em: <https://www.devmedia.com.br/bancos-de-dados-relacionais/20401>. Acesso em: 28 ago. 2022.

MICROSOFT. **Instâncias de usuário do SQL Server Express** [Página web: Docs]. Microsoft. Redmond (WA, USA), 05 ago. 2022. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/framework/data/adonet/sql/sql-server-express-user-instances>. Acesso em: 28 ago.



Alessandra Santos

Experiência em análise e desenvolvimento de software, desenho de arquitetura de sistemas e palestrante sobre tecnologias Microsoft. Atualmente, é Arquiteta de Software no Banco Votorantim e Mentora Educacional no LAB365 do SENAI/SC.

SENAI <LAB365>