

Aula 5 - Criando o BD

Docupedia Export

Author:Goncalves Donathan (SO/OPM-TS21-BR)

Date:16-Jul-2024 13:23

Table of Contents

1	O SGBD utilizado será o Microsoft SQL Server.	4
2	Microsoft SQL Server Management Studio é a interface para que o desenvolvedor tenha acesso ao banco de dados através do SQL.	5
3	Linguagem SQL	6
4	Modelo criado nas aulas de modelagem será a base para construção do banco.	8
5	USE - Seleciona o banco de dados que será utilizado para executar os próximos comandos.	9
6	CREATE - Usado para criar as características do Banco como tabela, trigger, view.	10
7	DROP - Apaga a entidade referenciada.	11
8	Schemas no SQL Server	12
9	Importante	14
10	Crie a entidade Sala	15
11	INSERT - Insere os dados em uma entidade.	16
12	SELECT - Consultas (queries) realizadas ao banco, retornando os dados solicitados.	17
13	Order By - Irá ordenar a busca, em relação à coluna especificada.	18
13.1	Ordem alfabética para String	18
13.2	Crescente para Números	18
13.3	Crescente para Datas	19
14	Order By	19
15	Crie a entidade Equipamento	21
16	DEFAULT - Valor padrão caso nada seja inserido.	22
17	CHECK - Restringe os dados que podem ser inseridos no banco.	23
18	Termine de criar as entidades ainda não criadas	24

19 Preencha com pelo menos três registros em cada tabela

25

1 O SGBD utilizado será o Microsoft SQL Server.



O SQL pode ser utilizado para realizar **consultas** nos dados, mas também cuidar da **manutenção** e **administração** dos bancos de dados relacionais.

Comandos SQL são divididos em três grupos

DDL (Data Definition Language)

Manutenção da **estrutura** do banco de dados.

Ex: Create, alter, truncate, drop e rename.

DML (Data Manipulation Language)

Gerenciar os **dados**/informações.

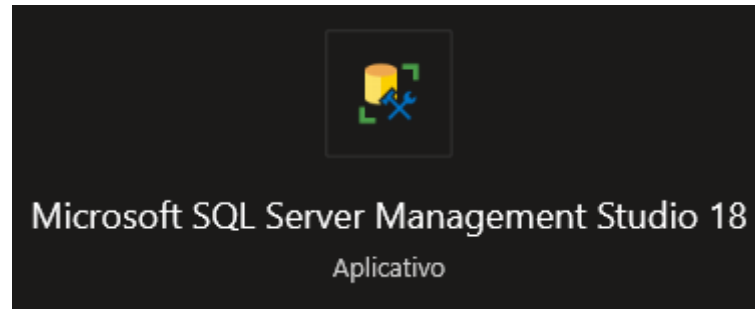
Ex: Insert, update, delete e lock.

DCL (Data Control Language)

Administrar as características do BD como **segurança**, **usuários**, como os dados são **armazenados** no banco.

Ex: Commit, rollback e savepoint.

2 **Microsoft SQL Server Management Studio é a interface para que o desenvolvedor tenha acesso ao banco de dados através do SQL.**



3 Linguagem SQL

Criando Banco de Dados

```
CREATE DATABASE VENDA_PRODUTO_01
```

```
CREATE DATABASE [VENDA_PRODUTO_01];
```

O SQL **não** é **Case Sensitive**, poder ser executado tanto em maiúsculo quanto em minúsculo.

É utilizado um **padrão** para criação desse Banco, em **propriedades** é possível ver essas configurações.

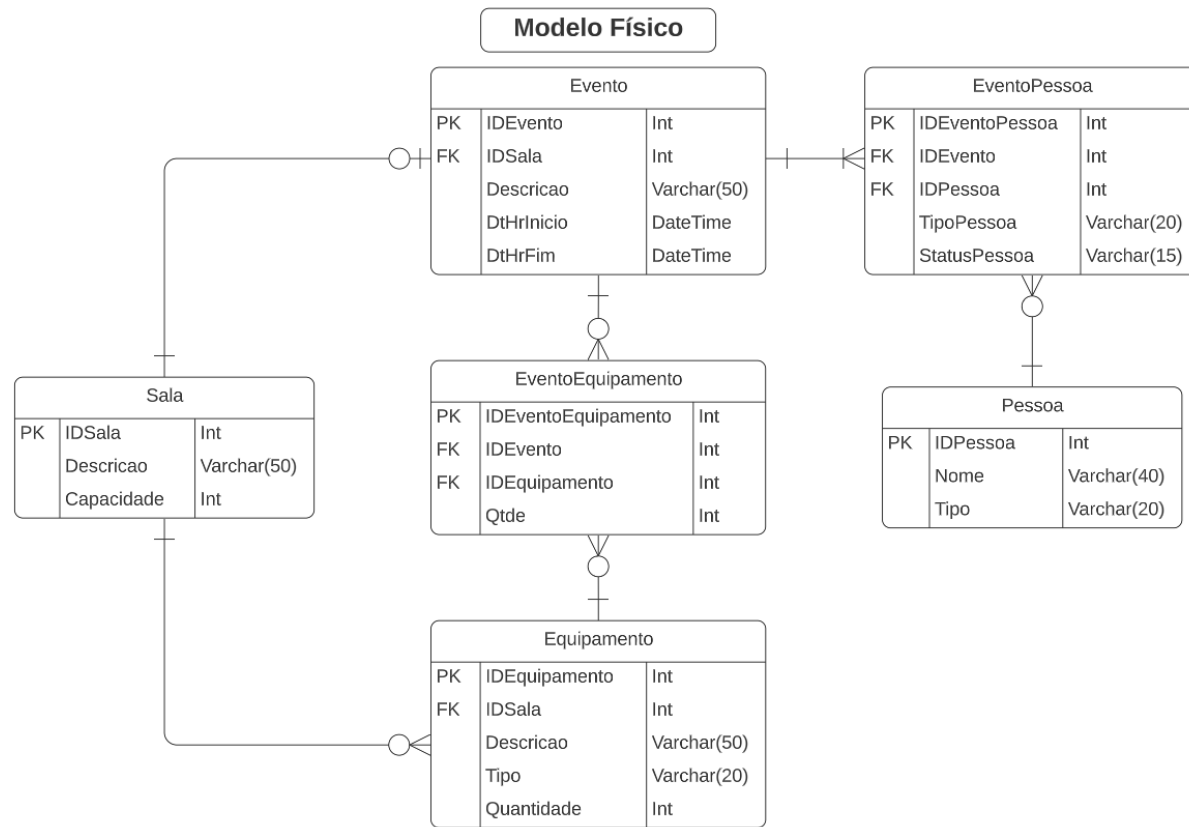
Script	Help
<div> <div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> <div></div> </div> </div>	
<div> <div>Backup</div> <div> <div>Last Database Backup</div> <div>None</div> </div> <div> <div>Last Database Log Backup</div> <div>None</div> </div> </div>	
<div> <div>Database</div> <div> <div>Name</div> <div>VENDA_PRODUTO_01</div> </div> <div> <div>Status</div> <div>Normal, Autoclosed</div> </div> <div> <div>Owner</div> <div>BR\LIQ1CT</div> </div> <div> <div>Date Created</div> <div>24/01/2023 08:40:59</div> </div> <div> <div>Size</div> <div>16,00 MB</div> </div> <div> <div>Space Available</div> <div>5,60 MB</div> </div> <div> <div>Number of Users</div> <div>4</div> </div> <div> <div>Memory Allocated To Memory Optimized Objec</div> <div>0,00 MB</div> </div> <div> <div>Memory Used By Memory Optimized Objects</div> <div>0,00 MB</div> </div> </div>	
<div> <div>Maintenance</div> <div> <div>Collation</div> <div>SQL_Latin1_General_CP1_CI_AS</div> </div> </div>	

-- Cria banco de dados substituindo o padrão

```
CREATE DATABASE VENDA_PRODUTO_02 ON
(NAME = 'VENDA_PRODUTO.DAT',
FILENAME = 'C:\TESTE\VENDA_PRODUTO_02.MDF', --Possui todos os dados do banco
SIZE = 10MB, --Tamanho inicial
MAXSIZE = 50MB,
FILEGROWTH = 5MB) --Taxa de crescimento do banco
```

```
LOG ON
(NAME = 'VENDA_PRODUTO.LOG',
FILENAME = 'C:\TESTE\VENDA_PRODUTO_02.LDF', --Possui todos os comandos realizados no banco
SIZE = 10MB,
MAXSIZE = 50MB,
FILEGROWTH = 5MB);
```

4 Modelo criado nas aulas de modelagem será a base para construção do banco.



5 USE - Seleciona o banco de dados que será utilizado para executar os próximos comandos.

USE

```
USE nome_do_database;
```

6 CREATE - Usado para criar as características do Banco como tabela, trigger, view.

CREATE

```
-- Criando a tabela Colaborador
CREATE TABLE Colaborador(
  EDV INT NOT NULL PRIMARY KEY,
  Nome VARCHAR(60) NOT NULL,
  Idade INT NOT NULL,
  Salario FLOAT);
```

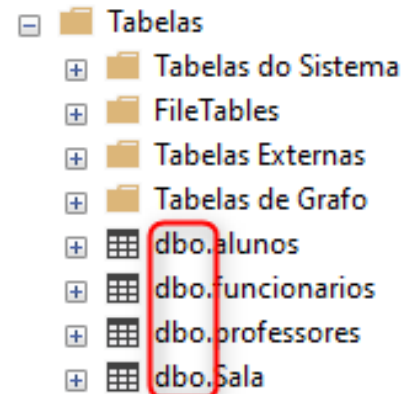
7 DROP - Apaga a entidade referenciada.

DROP

```
-- Excluir a tabela  
DROP TABLE Colaborador;
```

8 Schemas no SQL Server

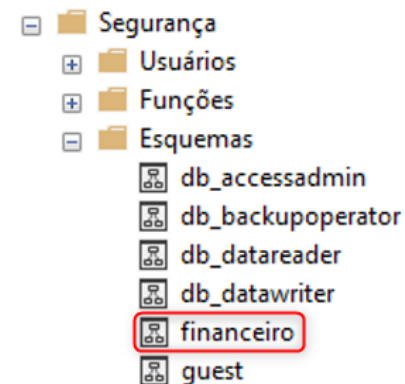
[DBO] é o esquema padrão do sistema.



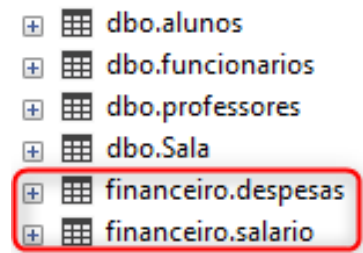
Utilizado para **classificar, agrupar e gerenciar permissões** das tabelas.

```
-- Criando Schema  
Create schema financeiro;
```

Ficará visível em **Banco criado > Segurança > Esquemas**



```
-- Utilizando o schema  
Create table financeiro.salario (...)
```



9 Importante

- 1- Sempre que for chamar essa tabela, deve **informar** o **schema** criado.
 - 2- É possível criar **tabelas** com o **mesmo nome** e schemas diferentes.
 - 3- Possibilidade de **restringir usuários** que tem acesso ao schema.
-

10 Crie a entidade Sala

Sala

Criando tabela Sala

```
-- Criando a tabela sala
CREATE TABLE Sala (
  IDSala INT NOT NULL PRIMARY KEY,
  Nome VARCHAR(50) NOT NULL,
  Capacidade INT NOT NULL
);
```

Sala		
PK	IDSala	Int
	Nome	Varchar(50)
	Capacidade	Int

11 INSERT - Insere os dados em uma entidade.

INSERT

-- Declarando os atributos

```
INSERT INTO Sala (IDSala, Nome, Capacidade) VALUES (1, 'Sala de TI', 20);
```

-- Não declarando atributos

```
INSERT INTO Sala VALUES (2, 'Sala de Reunião', 10);
```

-- Inserindo várias linhas

```
INSERT INTO Sala VALUES (3, 'Sala de palestras 1', 30), (4, 'Sala de aula', 15);
```


12 SELECT - Consultas (queries) realizadas ao banco, retornando os dados solicitados.

SELECT

-- Recebendo todos os valores

```
SELECT * FROM Sala;
```

-- Recebendo a Descrição e a Capacidade

```
SELECT Nome, Capacidade FROM Sala;
```

13 Order By - Irá ordenar a busca, em relação à coluna especificada.

- **13.1 Ordem alfabética para String**
- **13.2 Crescente para Números**

• 13.3 Crescente para Datas

SELECT ORDER BY

```
-- Maneira Crescente
SELECT Nome, Capacidade FROM Sala ORDER BY Capacidade

-- Maneira Decrescente na segunda coluna e Crescente na primeira
SELECT Nome, Capacidade FROM Sala ORDER BY Capacidade DESC, Nome ASC
```

14 Order By

	Nome	Capacidade
3	Sala Digital	20
1	Sala de Reunião 1	10
2	Sala de Reunião 2	10

Criando a Tabela Evento

```
CREATE TABLE Evento (
  IDEvento INT NOT NULL PRIMARY KEY,
  IDSala INT NOT NULL FOREIGN KEY REFERENCES Sala(IDSala),
  Descricao VARCHAR(100) NOT NULL,
  DtHrInicio DATETIME NOT NULL,
  DtHrFim DATETIME NOT NULL
);
```

Evento		
PK	IDEvento	Int
FK	IDSala	Int
	Descricao	Varchar(100)
	DtHrInicio	DateTime
	DtHrFim	DateTime

Inserindo registros em Eventos

```
INSERT INTO Evento VALUES (
  1,
  1,
  'Aula de IoT',
  '26/03/2022 19:00',
```

```
'26/03/2022 22:30'  
);
```

```
-- Inserindo valores sem saber o último índice  
INSERT INTO Evento VALUES (  
    (SELECT MAX(IDEvento) FROM Evento)+1,  
    1,  
    'Aula de Estrutura de Dados',  
    '28/03/2022 21:15',  
    '26/03/2022 22:30'  
);
```

OBS:

Data armazenada no banco é **YYYY-MM-DD HH:MM:SS**

Na inserção, irá variar a maneira de inserir esses dados.

15 Crie a entidade Equipamento

Equipamento

Criando tabela Equipamentos

```
CREATE TABLE Equipamento (  
  IDEquipamento INT NOT NULL PRIMARY KEY,  
  IDSala INT FOREIGN KEY REFERENCES Sala(IDSala),  
  Nome VARCHAR(50) NOT NULL,  
  Tipo VARCHAR(100)  
);
```

Equipamento		
PK	IDEquipamento	Int
FK	IDSala: Null	Int
	Nome	Varchar(50)
	Tipo	Varchar(100)

Insert Equipamento

```
INSERT INTO Equipamento VALUES (  
  (SELECT IsNull(MAX(IDEquipamento), 0) FROM Equipamento ) + 1,  
  Null,  
  'Notebook HP',  
  'Informática'  
);
```

Insira mais dois ou três registros na tabela Equipamento

IDENTITY - Realiza um auto incremento a cada inserção em um atributo(coluna).

Geralmente as **IDs** são **resultado** de algum **cálculo** ou utilizam **string**, por isso, Identity não é tão usado.

Nem sempre seguirá uma ordem exata dos valores.

16 DEFAULT - Valor padrão caso nada seja inserido.

Obs: Para deixar esse padrão, deve **selecionar** as outras colunas na hora de inserir.

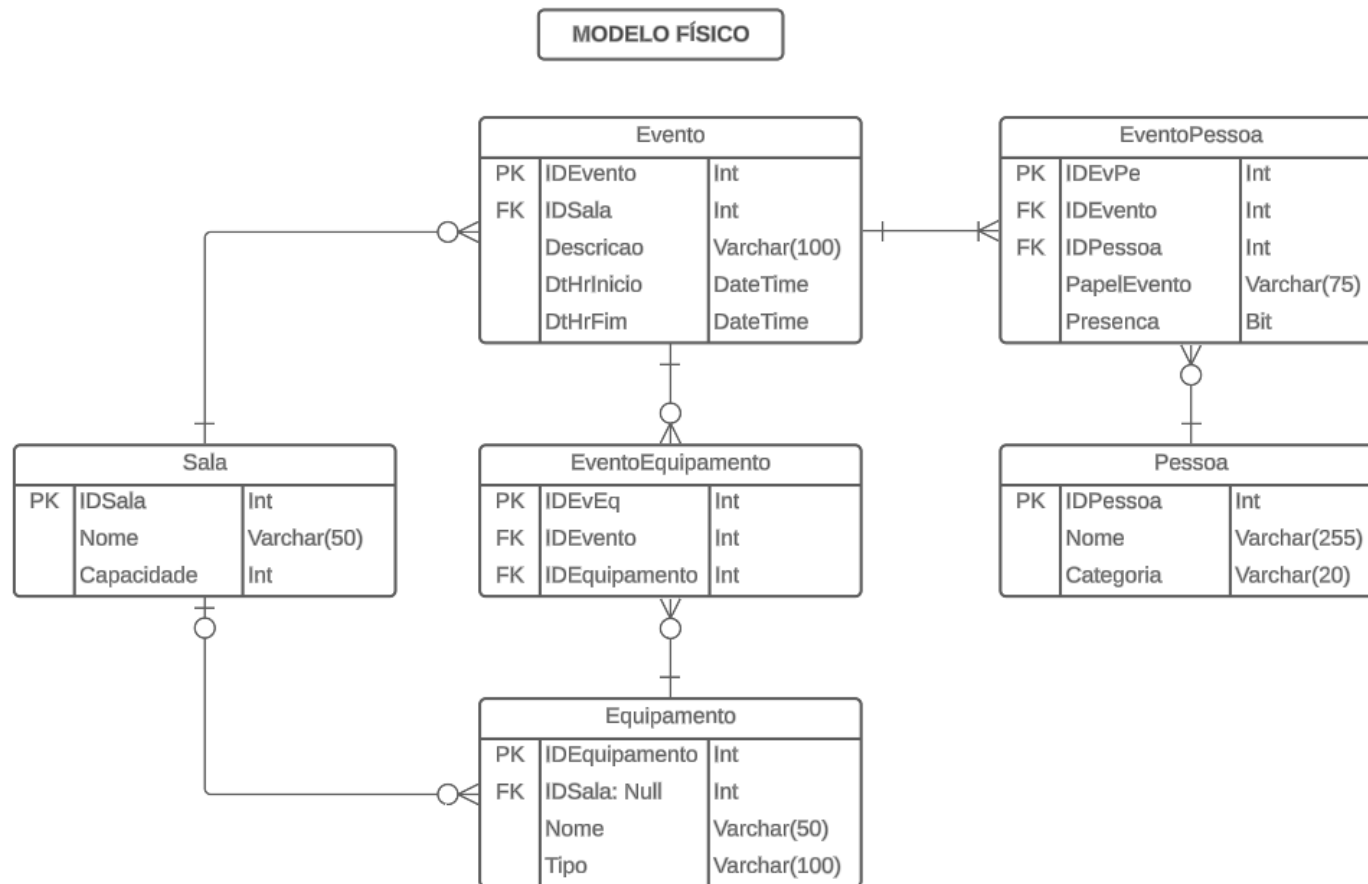
17 CHECK - Restringe os dados que podem ser inseridos no banco.

Identity

```
CREATE TABLE Aluno (  
    IDAluno INT IDENTITY(1,1) PRIMARY KEY NOT NULL,  
    Nome VARCHAR(255) NOT NULL,  
    Idade INT,  
    Nacionalidade VARCHAR(255) DEFAULT 'Brasil',  
    CHECK (Idade >= 18)  
);
```

18 Termine de criar as entidades ainda não criadas

19 Preencha com pelo menos três registros em cada tabela



Pessoa

```

CREATE TABLE Pessoa (
    IDPessoa INT NOT NULL PRIMARY KEY,
    Nome VARCHAR(40) NOT NULL,
    Categoria VARCHAR(20) NOT NULL
);
  
```

EventoPessoa

```
CREATE TABLE EventoPessoa (  
    IDEventoPessoa INT NOT NULL PRIMARY KEY,  
    IDEvento INT NOT NULL FOREIGN KEY REFERENCES Evento(IDEvento),  
    IDPessoa INT NOT NULL FOREIGN KEY REFERENCES Pessoa(IDPessoa),  
    PapelEvento VARCHAR(75) NOT NULL,  
    Presenca VARCHAR(15) NOT NULL  
);
```

EventoEquipamento

```
CREATE TABLE EventoEquipamento (  
    IDEventoEquipamento INT NOT NULL PRIMARY KEY,  
    IDEvento INT NOT NULL FOREIGN KEY REFERENCES Evento(IDEvento),  
    IDEquipamento INT NOT NULL FOREIGN KEY REFERENCES Equipamento(IDEquipamento)  
);
```