

Apresentação de Modelo Relacional |

**Linguagem de Manipulação de Dados
(DML)**



DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

Considerações Iniciais

- Nosso horário é das **19:00 às 22:00 (10 min tolerância de chegada)**
- Intervalo às **20:30 de 20 minutos**
- Interaja na aula!
- Se tiver dúvidas, levanta a mão no chat e eu explico novamente
- Me corrija!!!!

Erros são comuns no mundo do Software

Eu posso errar. Corrija-me quando acontecer

Aceito feedbacks :D

- Façam as atividades práticas, só se aprende praticando!
- Qualquer coisa me procurem no **slack**



Agenda


- Instalação do Banco de Dados SQL Server Express
- Versões do SQL Server
- Conhecendo o SQL Server Express
- Sistema de Gerenciamento de Banco de Dados Relacional (**SGBDR**)
- O que é uma instância de banco de dados

Comandos DML

SQL

Linguagem de Consulta Estruturada

Modelagem de Dados

- O que é SQL?
 - Structured Query Language ou Linguagem de Consulta Estruturada
 - Linguagem padrão para acessar e manipular SGBDs relacionais.
 - É baseada no inglês, e usa palavras como:
 - SELECT, INSERT, DELETE como parte de seu conjunto de comandos;
- A SQL é composta por:
 - DDL: Data Definition Language (Linguagem de Definição de dados) 
 - DML: Data Manipulation Language (Linguagem de Manipulação de Dados)
 - DCL: Data Control Language (Linguagem de Controle de Dados)
- **SQL** = *DML* + *DDL* + *DCL*

Modelagem de Dados

- DML
 - Data Manipulation Language (Linguagem de manipulação de dados)
 - Permite recuperar e alterar os dados do BD.
 - Define comandos para a seleção, inserção, remoção e alteração de linhas das tabelas, basicamente a nível de “Linhas”.
 - Repetindo:
 - seleção (SELECT)
 - inserção (INSERT)
 - remoção (DELETE)
 - alteração (UPDATE)

Modelagem de Dados

- DCL
 - Data Control Language (Linguagem de controle de dados)
 - Permite obter comandos de controle de acesso ao banco de dados.
 - grupo de comandos que permitem ao administrador de banco de dados controlar o acesso aos dados deste banco

Exemplos dos comandos DCL:

- 1) GRANT: Fornece a determinada pessoa o privilégio de acesso dentro do banco de dados em tabelas específicas. Exemplo: `GRANT SELECT ON [nome_tabela] TO Maria`
- 2) REVOKE: Esse comando retira os privilégios de acesso. Ou seja, faz a operação inversa ao GRANT, negando a permissão. `REVOKE SELECT ON [nome_tabela] FROM Luiz;`

Modelagem de Dados

IMPORTANTE

- Instruções Gerais:
 - Os comandos podem ser escritos em mais de uma linha;
 - Cláusulas diferentes são colocadas usualmente em linhas diferentes;
 - Comandos podem ser escritos em letras maiúsculas e/ou minúsculas;
 - Finaliza-se a query com o sinal de ponto e vírgula (;)
 - Existe um comando que modifica isso para qualquer outra letra ou sinal
 - Comumente é utilizado por padrão o ;

Comandos DDL

- DDL
 - Data Definition Language (Linguagem de definição de dados)
 - Descreve a estrutura do BD.
 - Define comandos para criação, alteração e remoção de objetos do banco de dados, basicamente a nível de “tabelas”.
 - Repetindo:
 - criação (CREATE)
 - alteração (ALTER)
 - remoção (DROP)

Comandos DDL - CREATE

DDL - CREATE

PostgreSQL

- Criando nosso Primeiro Banco de Dados:
- Comandos SQL são INSTRUÇÕES, portanto:

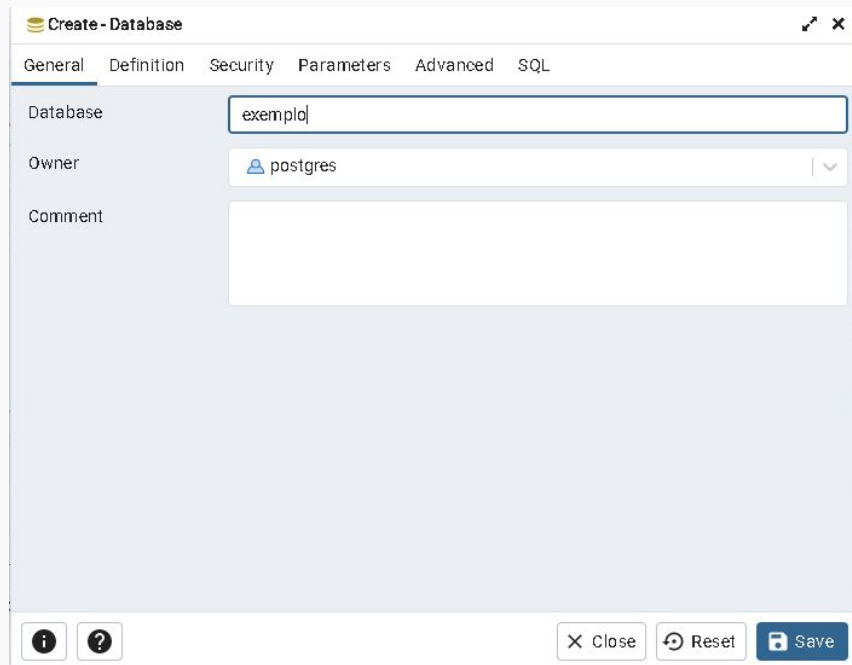
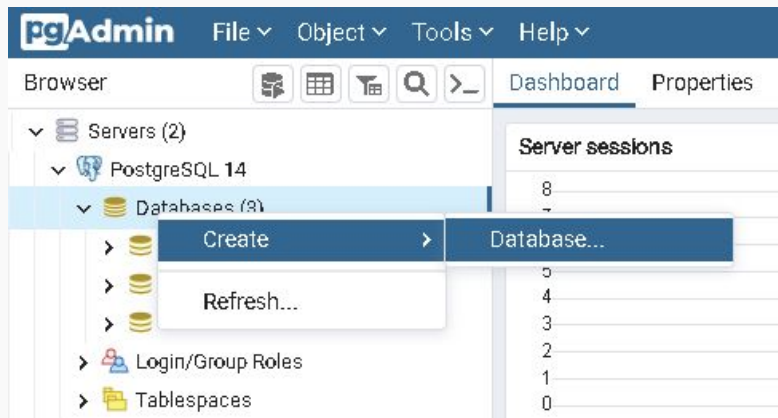
create database nomeBancoDeDados;

○

```
postgres=# create database exemplo;  
CREATE DATABASE
```

PostgreSQL

- Criando nosso Primeiro Banco de Dados:



PostgreSQL

- Como saber se o banco foi criado?
- Listando os bancos existentes no nosso serviço!
 - comando \l (L minúsculo)

```
postgres=# \l
```

List of databases					
Name	Owner	Encoding	Collate	Ctype	Access privileges
exemplo	postgres	UTF8	en_US.utf8	en_US.utf8	
postgres	postgres	UTF8	en_US.utf8	en_US.utf8	
rafa	postgres	UTF8	en_US.utf8	en_US.utf8	
template0	postgres	UTF8	en_US.utf8	en_US.utf8	=c/postgres +
					postgres=CTc/postgres
template1	postgres	UTF8	en_US.utf8	en_US.utf8	=c/postgres +
					postgres=CTc/postgres

(5 rows)

```
MariaDB [ucloud_clarobr]> show databases;
```

Database
billing
etice
information_schema
jobs
mysql
performance_schema
ucloud

PostgreSQL

- Trocando de banco
 - Quando startamos o serviço do banco, sempre temos que dizer para o Postgres qual banco vamos utilizar
 - PostgreSQL
 - `\c 'nome_database';`

```
rafa=# \c exemplo
You are now connected to database "exemplo" as user "postgres".
exemplo=#
```

Comandos DDL - CREATE

- **Criando uma nova tabela no nosso banco**
 - Vamos pegar uma das modelagens que fizemos nas atividades anteriores
 - Primeiro, uma mais Simples sem relacionamento
- **Comando para a criação de tabela**
 - **CREATE** TABLE 'table_name' ('column_name' TYPE column_constraint, 'column_name' TYPE);

Comandos DDL - CREATE

Criando tabelas em um banco de dados

- Para criar tabelas no banco de dados é necessário usar a declaração CREATE TABLE, como pode ser observado abaixo:

```
CREATE TABLE table_name (  
    column_name TYPE column_constraint,  
    table_constraint table_constraint  
) INHERITS existing_table_name;
```

- Onde:
 - **table_name**: é o nome da tabela que está sendo criada.
 - **column_name**: é o nome da coluna da tabela (para criar várias colunas basta separá-las por vírgulas).
 - **TYPE**: é o tipo de dados da coluna criada.
 - **column_constraint** (restrições das colunas): define os papéis das colunas (NOT NULL, PRIMARY KEY, etc.)
 - **INHERITS**: pode especificar uma tabela existente a partir da qual a tabela atual irá herdar características.



Comandos DDL

- Podemos colocar algumas restrições em nossas Colunas:
 - **NOT NULL** - o valor da coluna não pode ser nulo (em branco).
 - **UNIQUE** - o valor da coluna deve ser exclusivo em toda a tabela.
 - **PRIMARY KEY** - essa restrição é a combinação de restrições NOT NULL e UNIQUE.
 - Você pode definir uma coluna como PRIMARY KEY usando a restrição no nível da coluna. Caso a chave primária contenha várias colunas, você deve usar a restrição em nível de tabela.

Comandos DDL

- Podemos colocar algumas restrições em nossas Colunas:
 - **CHECK** - o valor da coluna deve ser validado no teste, exemplo:

```
CREATE TABLE products (  
    product_no integer,  
    name text,  
    price numeric CHECK (price > 0),  
    discounted_price numeric,  
    CHECK (discounted_price > 0 AND price > discounted_price)  
);
```

Comandos DDL

- Podemos colocar algumas restrições em nossas Colunas:
 - **INHERITS**: uma tabela herdar colunas de outra tabela, exemplo:

```
CREATE TABLE cities (  
    name      text,  
    population real,  
    elevation  int    -- (in ft)  
);
```

```
CREATE TABLE capitals (  
    state      char(2) UNIQUE NOT NULL  
) INHERITS (cities);
```

Comandos DDL - chave única

- A chave UNIQUE garante que todos os valores em uma coluna sejam diferentes
- Uma restrição PRIMARY KEY tem automaticamente uma restrição UNIQUE
- No entanto, você pode ter muitas restrições UNIQUE por tabela, mas apenas uma restrição

PRIMARY KEY por tabela

```
CREATE TABLE Funcionario (  
    Id int IDENTITY(1,1) PRIMARY KEY NOT NULL,  
    Matricula varchar(20) UNIQUE NOT NULL,  
    Nome varchar(200) NOT NULL,  
    Endereco varchar(200) NOT NULL,  
    Cidade varchar(200) NOT NULL,  
    Estado char(2) NOT NULL,  
    EstadoCivil varchar(30) NULL,  
    DataAdmissao Datetime NOT NULL  
)
```

Comandos DDL - CREATE

Exemplo:

```
create table pessoas (  
    id SERIAL PRIMARY KEY,  
    nome VARCHAR(80) NOT NULL,  
    sobrenome VARCHAR(80) NOT NULL,  
    endereco_id INT REFERENCES enderecos (id)  
);
```

Comandos DDL - CREATE

Vamos Praticar!

Comandos DDL - FK Chave estrangeira

Exemplo:

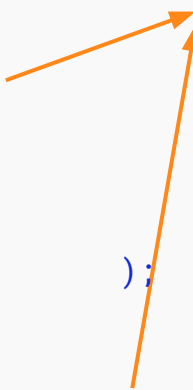
A)

```
create table pessoas (  
  id SERIAL PRIMARY KEY,  
  nome VARCHAR(80) NOT NULL,  
  sobrenome VARCHAR(80) NOT NULL,  
  endereco_id INT REFERENCES enderecos (id)  
);
```

B)

```
create table lojas (  
  id SERIAL PRIMARY KEY,  
  nome VARCHAR(80) NOT NULL,  
  descricao VARCHAR(255) NOT NULL,  
  endereco_id INT,  
  FOREIGN KEY (endereco_id) REFERENCES enderecos (id)  
);
```

```
create table enderecos (  
  id SERIAL PRIMARY KEY,  
  rua VARCHAR(255) NOT NULL,  
  numero INT,  
  complemento VARCHAR(40) NOT NULL,  
  bairro VARCHAR(80) NOT NULL,  
  cidade VARCHAR(80) NOT NULL,  
  uf CHAR(2) NOT NULL  
);
```



The diagram consists of two orange arrows. The first arrow originates from the 'endereco_id INT REFERENCES enderecos (id)' line in the 'pessoas' table definition and points to the 'id SERIAL PRIMARY KEY' line in the 'enderecos' table definition. The second arrow originates from the 'FOREIGN KEY (endereco_id) REFERENCES enderecos (id)' line in the 'lojas' table definition and also points to the 'id SERIAL PRIMARY KEY' line in the 'enderecos' table definition.

FK - On Delete e On Update

Nos exemplos anteriores, nossas entidades **Pessoas** e **Lojas** possuem uma chave estrangeira ***endereco_id*** para um endereço cadastrado na tabela **Enderecos**.

O que acontece se apagarmos um endereço da tabela **Enderecos** que tenha uma vinculação à um registro das outras tabelas?

FK - On Delete e On Update

Vamos imaginar uma outra situação.

Em um sistema de vendas, temos uma entidade **Pedidos** e uma entidade **Pedidoltems**.

Faz sentido existirem **Pedidoltem** sem um **Pedido**? E como fazemos para manter a **integridade de relacionamento** entre as tabelas?

FK - On Delete e On Update

Exemplo:

```
create table pedidos (  
  id SERIAL PRIMARY KEY,  
  data TIMESTAMP NOT NULL DEFAULT current_timestamp,  
  cliente VARCHAR(255) NOT NULL,  
  valor_total NUMERIC NOT NULL DEFAULT 0,  
);
```

```
create table pedido_itens (  
  id SERIAL PRIMARY KEY,  
  pedido_id INT NOT NULL,  
  produto_titulo VARCHAR(80) NOT NULL,  
  descricao VARCHAR(255) NOT NULL,  
  valor NUMERIC,  
  FOREIGN KEY (pedido_id) REFERENCES pedidos (id)  
);
```

FK - On Delete e On Update

Usamos as diretivas **ON DELETE** e **ON UPDATE** para determinar o que acontece com a entidade **atual** caso a entidade **referenciada** seja **modificada** ou **excluída**.

FK - On Delete e On Update

As opções são:

- **RESTRICT e NO ACTION:** impede a exclusão da entidade relacionada enquanto houver um relacionamento
- **CASCADE:** quando a entidade relacionada é removida, a entidade com a **FK** também é removida.
- **SET NULL** ou **SET DEFAULT:** quando a entidade relacionada é removida, o valor da coluna da entidade com **FK** passa a ser **NULL** ou o valor **DEFAULT**, respectivamente

FK - On Delete e On Update

Exemplo atualizado:

```
create table pedidos (  
    id SERIAL PRIMARY KEY,  
    data TIMESTAMP NOT NULL DEFAULT current_timestamp,  
    cliente VARCHAR(255) NOT NULL,  
    valor_total NUMERIC NOT NULL DEFAULT 0  
);  
  
create table pedido_itens (  
    id SERIAL PRIMARY KEY,  
    pedido_id int NOT NULL,  
    produto_titulo VARCHAR(80) NOT NULL,  
    descricao VARCHAR(255) NOT NULL,  
    valor NUMERIC,  
    FOREIGN KEY (pedido_id) REFERENCES pedidos (id) ON DELETE CASCADE  
);
```

Comandos DDL - chave estrangeira

ATENÇÃO

Podemos inserir relacionamentos de FK (chave estrangeira) em tabelas que já foram criadas.

Para isso, utilize o comando **ALTER TABLE**, veja o exemplo:

```
ALTER TABLE Carro  
ADD CONSTRAINT fk_Pessoa_Carro FOREIGN KEY (IdPessoa) REFERENCES Pessoa(Id)
```

Comandos DML

DML - INSERT

Comandos DML - INSERT INTO

- Inserindo dados numa Tabela:

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

- Os parâmetros de inserção são “parametrizáveis”, a depender do que queremos inserir

- Exemplo:

- `INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country) VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');`

INSERT - DML

- Inserindo dados numa Tabela:
 - `INSERT INTO table_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);`
- Os parâmetros de inserção são “parametrizáveis”, a depender do que queremos inserir
 - Exemplo:
 - `INSERT INTO Customers (column1, column4, column5) VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21');`
- Podemos inserir em mais linhas com um mesmo comando:
 - Exemplo:
 - `INSERT INTO table_name (column1) VALUES ('valor1'), ('valor2'), ('valor3')....);`

Comandos DML

Vamos Praticar!

Comandos DDL

DDL - ALTER

Comandos DDL - ALTER

- Alterando uma tabela no nosso banco:

- Adicionar coluna na tabela:

- `ALTER TABLE table_name ADD column_name datatype;`

- Remover uma coluna da tabela:

- `ALTER TABLE table_name DROP COLUMN column_name;`

- Alterar o tipo de dado de uma coluna:

- `ALTER TABLE table_name ALTER COLUMN column_name TYPE datatype;`

Alterando nome da coluna:

- `ALTER TABLE table_name RENAME column_name TO novo_column_name`

Alterando nome da tabela:

- `ALTER TABLE nome_atual RENAME TO novo_nome;`

Alterando nome do banco (não é possível fazer no banco atual e com conexões ativas):

- `ALTER DATABASE nome_database RENAME TO novo_nome;`

Comandos DDL - ALTER

- Outros exemplos de alter:

- **Adicionar restrição na tabela:**

- `ALTER TABLE produtos ADD CONSTRAINT unq_cod_prod UNIQUE (cod_prod);`

- **Remover restrição na tabela:**

- `ALTER TABLE table_name DROP constraint nome_constraint;`

- **Alterar comportamento / restrição de uma coluna da tabela:**

- `ALTER TABLE produtos ALTER COLUMN preco SET DEFAULT 7.77;`

Comandos DDL - ALTER

Vamos Praticar!

- Adicionar Coluna na tabela:

```
ALTER TABLE PESSOA ADD Estado char(2) NOT NULL
```

```
ALTER TABLE PESSOA ADD EstadoCivil varchar(20)
```

Comandos DDL - ALTER

Vamos Praticar!

- Remover uma Coluna da tabela:

```
ALTER TABLE PESSOA DROP COLUMN EstadoCivil
```

Comandos DDL - ALTER

Vamos Praticar!

- Alterar o tipo de uma coluna:

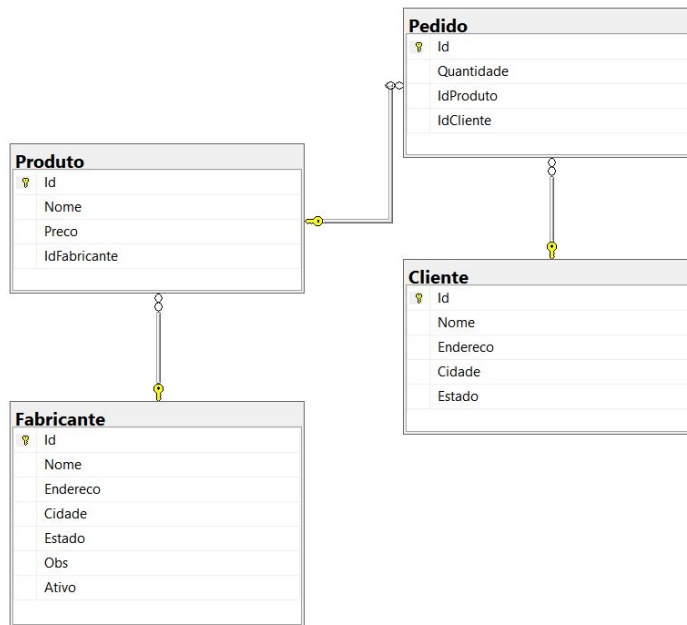
De: `ALTER TABLE PESSOA ADD EstadoCivil varchar(20)`

Para: `ALTER TABLE PESSOA alter column EstadoCivil nvarchar(30)`

Comandos DDL

Vamos Praticar!

- Vamos criar uma estrutura de pedidos, com as seguintes tabelas e os seus respectivos relacionamentos:



Comandos DDL

- DDL
 - Data Definition Language (Linguagem de definição de dados)
 - Descreve a estrutura do BD.
 - Define comandos para criação, alteração e remoção de objetos do banco de dados, basicamente a nível de “tabelas”.
 - Repetindo:
 - criação (CREATE)
 - alteração (ALTER)
 - **remoção (DROP)**

Comandos DDL

DDL - DROP

Comandos DDL -DROP

- Deletando uma Tabela do nosso banco

- `DROP TABLE table_name;`

- É o mesmo comando para também deletar um banco

- `DROP DATABASE database_name;`

Comandos DDL -DROP

Vamos Praticar!

Vamos criar um Banco de dados

```
CREATE DATABASE EscolaSenai;
```

Comandos DDL -DROP

Vamos Praticar!

Vamos criar e inserir dados na tabela Aluno

```
CREATE TABLE Aluno
```

```
(  
    Id SERIAL PRIMARY KEY,  
    Nome varchar(200) NOT NULL,  
    Endereco varchar(200) NOT NULL,  
    Cidade varchar(100) NOT NULL,  
    Estado char(2) NOT NULL,  
    Curso varchar(100) NOT NULL  
);
```

```
INSERT INTO Aluno values('Patricia', 'Rua Azul, 54', 'Bragança Paulista', 'SP', 'Administração de  
Empresa');
```

```
INSERT INTO Aluno values('João', 'Avenida Paulista, 1000', 'São Paulo', 'SP', 'Engenharia da Computação');
```

Comandos DDL -DROP

Vamos Praticar!

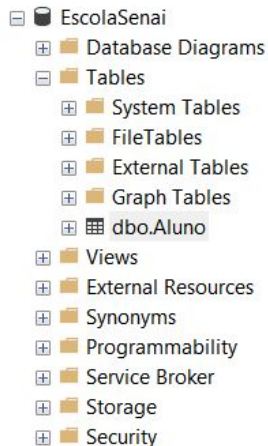
Vamos pesquisar se os registros
foram inseridos com sucesso.

```
SELECT  Id
        , Nome
        , Endereco
        , Cidade
        , Estado
        , Curso
FROM    EscolaSenai.dbo.Aluno
```

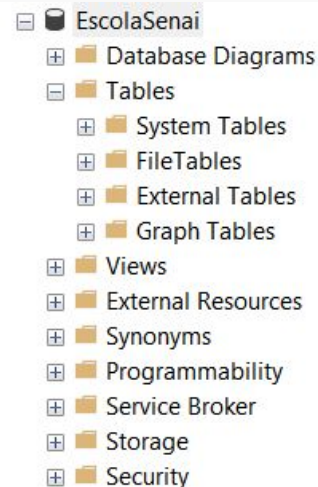
Comandos DDL -DROP

Vamos Praticar!

Agora chegou o momento de remover (drop) a tabela Aluno



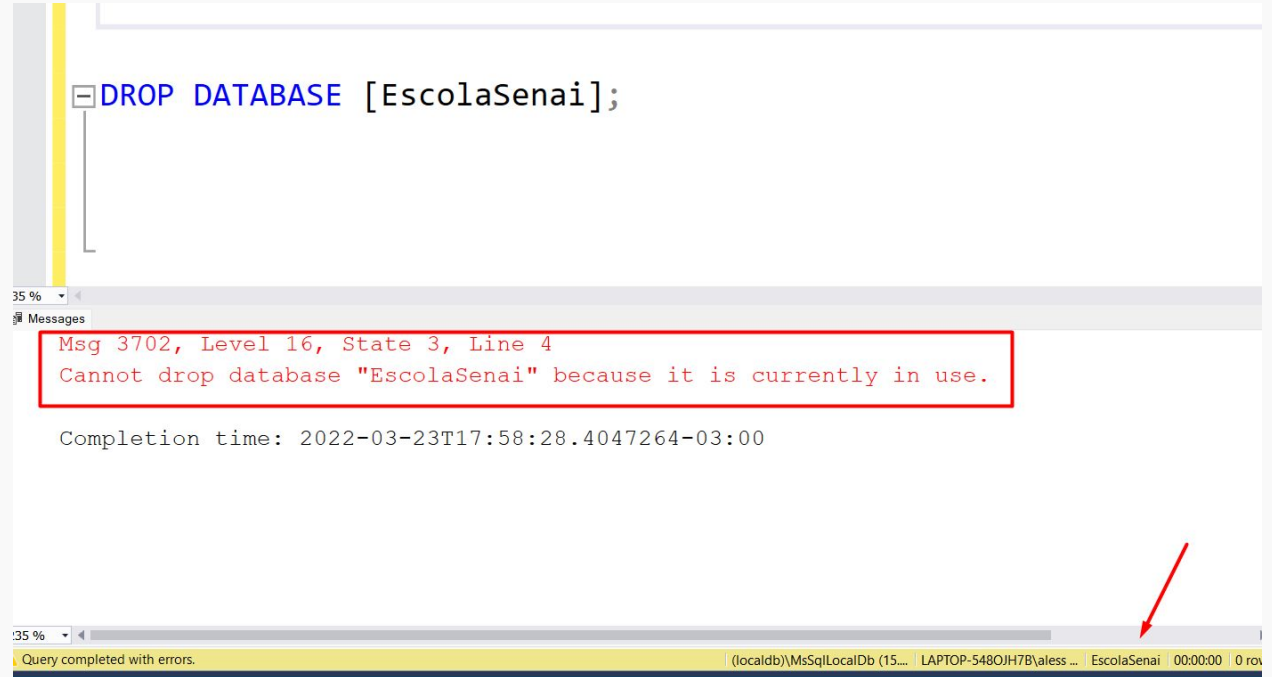
```
DROP TABLE Aluno;
```



Comandos DDL -DROP

Vamos Praticar!

Agora chegou o
momento de
remover (drop) o
banco de dados
EscolaSenai



The screenshot shows the SQL Server Enterprise Manager interface. In the top pane, the command `DROP DATABASE [EscolaSenai];` is entered. The bottom pane, titled 'Messages', displays an error message: 'Msg 3702, Level 16, State 3, Line 4: Cannot drop database "EscolaSenai" because it is currently in use.' Below the error message, the completion time is shown as '2022-03-23T17:58:28.4047264-03:00'. At the bottom of the window, a status bar indicates 'Query completed with errors.' and shows the current database as 'EscolaSenai'.

```
DROP DATABASE [EscolaSenai];
```

Msg 3702, Level 16, State 3, Line 4
Cannot drop database "EscolaSenai" because it is currently in use.

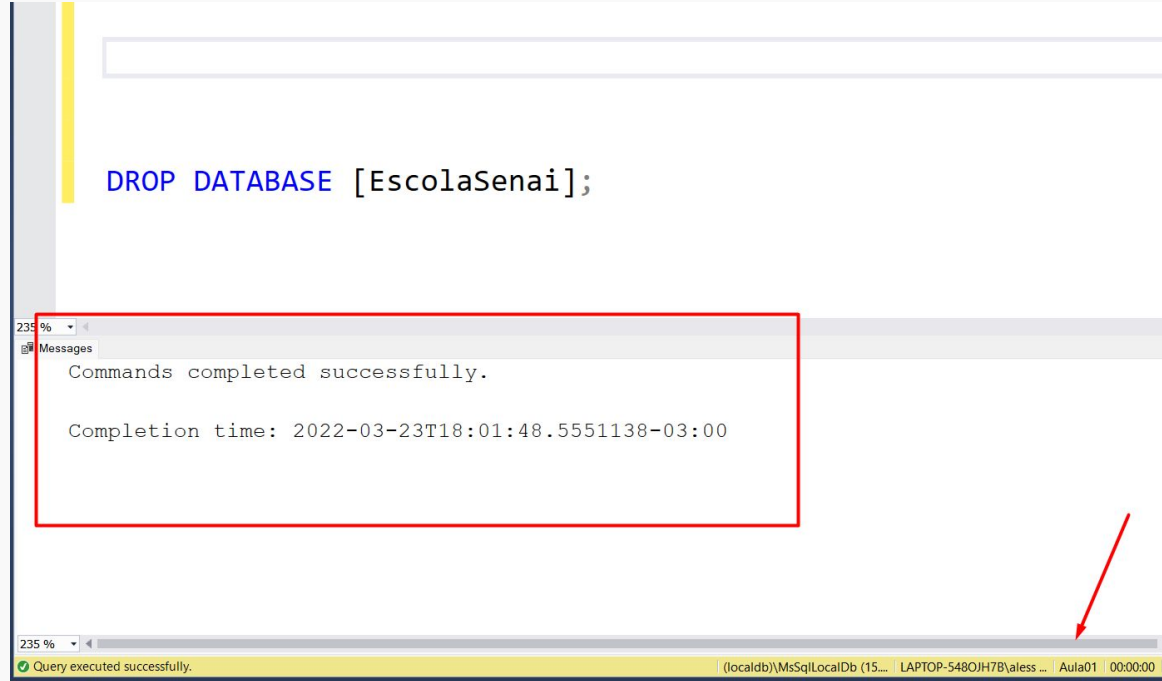
Completion time: 2022-03-23T17:58:28.4047264-03:00

Query completed with errors. | (localdb)\MsSqlLocalDb (15... | LAPTOP-548OJH7B\ale... | EscolaSenai | 00:00:00 | 0 rows

Comandos DDL -DROP

Vamos Praticar!


No slide anterior verificamos que **não foi possível remover o banco de dados EscolaSenai**, para isso, **clique em outro banco de dados da sua instância, clique em New Query e realize a instrução ao lado:**



Material adicional sobre Fundamentos de Banco de Dados

- Conceito de Banco de Dados Relacional
- Fundamentos de banco de dados
- Modelagem de dados sem segredos
- Banco de Dados
- Como utilizar a ferramenta de modelagem de dados
BrModelo

Modelagem de Dados

- O que é SQL?
 - Structured Query Language ou Linguagem de Consulta Estruturada
 - Linguagem padrão para acessar e manipular SGBDs relacionais.
 - É baseada no inglês, e usa palavras como:
 - SELECT, INSERT, DELETE como parte de seu conjunto de comandos;
- A SQL é composta por:
 - DDL: Data Definition Language (Linguagem de Definição de dados) 
 - DML: Data Manipulation Language (Linguagem de Manipulação de Dados)
 - DCL: Data Control Language (Linguagem de Controle de Dados)
- **SQL** = *DML* + *DDL* + *DCL*

Modelagem de Dados

- DML
 - Data Manipulation Language (Linguagem de manipulação de dados)
 - Permite recuperar e alterar os dados do BD.
 - Define comandos para a seleção, inserção, remoção e alteração de linhas das tabelas, basicamente a nível de “Linhas”.
 - Repetindo:
 - seleção (SELECT)
 - inserção (INSERT)
 - remoção (DELETE)
 - alteração (UPDATE)

Modelagem de Dados

IMPORTANTE

- Instruções Gerais:
 - Os comandos podem ser escritos em mais de uma linha;
 - Cláusulas diferentes são colocadas usualmente em linhas diferentes;
 - Comandos podem ser escritos em letras maiúsculas e/ou minúsculas;
 - Finaliza-se a query com o sinal de ponto e vírgula (;)

Material adicional sobre Modelagem de Dados

- [Modelagem de Dados](#)
- [Sete dicas sobre modelagem de dados](#)
- [Modelagem de dados - como ela é usada nas empresas](#)

AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.





DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>