



Trabalhando Com:

Cláusula TOP e Restrições

Prof. Télvio Orrú

telvio.orrú@docente.unip.br

Material elaborado.:

Prof. Nathan Cirillo e Silva

Universidade Paulista UNIP

BD@2023

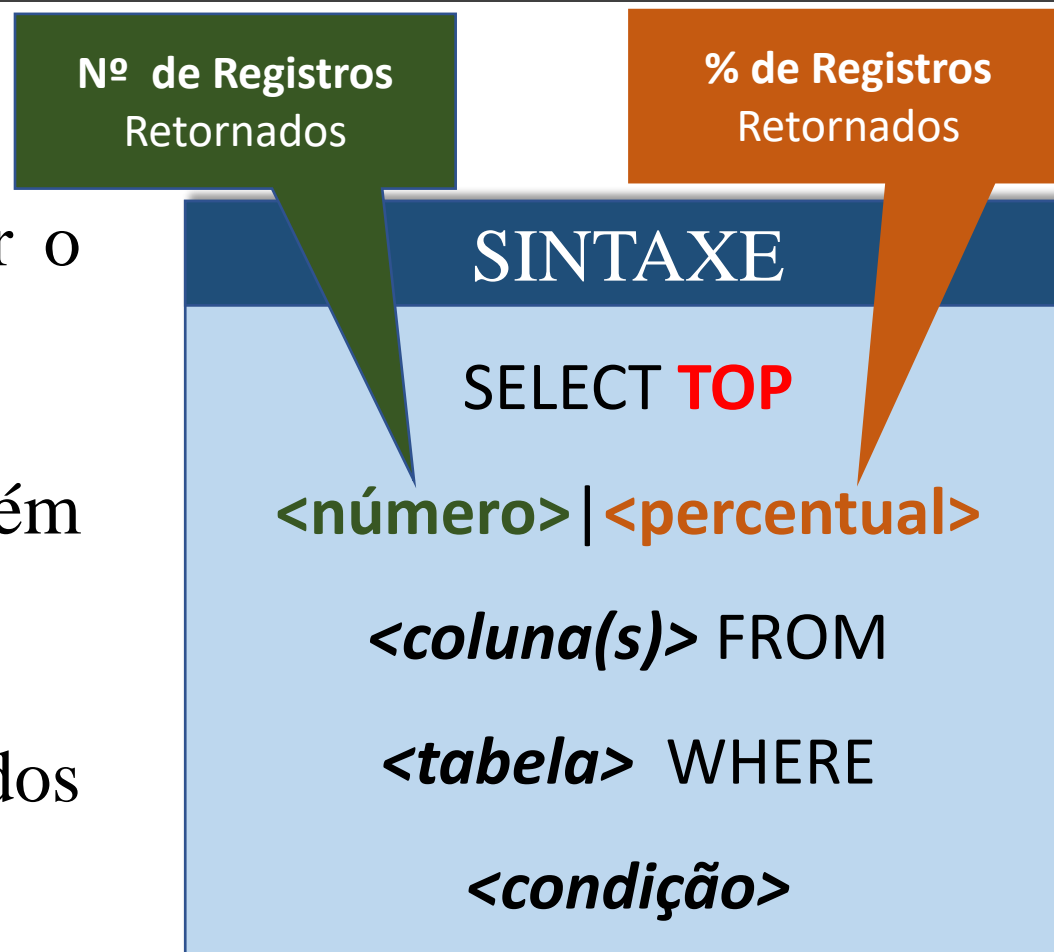
Aula 6

Restringindo Dados na Hora da Consulta



Cláusula TOP

- A cláusula TOP permite especificar o **nº de registros** a serem retornados;
- É útil em **tabelas grandes** que contém **milhares de registros**;
- Retornar grandes quantidades de dados pode *afetar o desempenho*.



Uso da Cláusula TOP

```
SELECT  
CategoryID, CategoryName,  
Description from  
Categories
```



	CategoryID	CategoryName	Description
1	1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	2	Condiments	Sweet and savory sauces, relishes, spreads, and ...
3	3	Confections	Desserts, candies, and sweet breads
4	4	Dairy Products	Cheeses
5	5	Grains/Cereals	Breads, crackers, pasta, and cereal
6	6	Meat/Poultry	Prepared meats
7	7	Produce	Dried fruit and bean curd
8	8	Seafood	Seaweed and fish


```
SELECT TOP 3  
CategoryID, CategoryName,  
Description from  
Categories
```



	CategoryID	CategoryName	Description
1	1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	2	Condiments	Sweet and savory sauces, relishes, spreads, and ...
3	3	Confections	Desserts, candies, and sweet breads


Uso da Cláusula TOP

```
SELECT  
CategoryID, CategoryName,  
Description from  
Categories
```



	CategoryID	CategoryName	Description
1	1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	2	Condiments	Sweet and savory sauces, relishes, spreads, and ...
3	3	Confections	Desserts, candies, and sweet breads
4	4	Dairy Products	Cheeses
5	5	Grains/Cereals	Breads, crackers, pasta, and cereal
6	6	Meat/Poultry	Prepared meats
7	7	Produce	Dried fruit and bean curd
8	8	Seafood	Seaweed and fish

```
SELECT TOP 50 PERCENT  
CategoryID, CategoryName,  
Description from  
Categories
```



	CategoryID	CategoryName	Description
1	1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	2	Condiments	Sweet and savory sauces, relishes, spreads, and ...
3	3	Confections	Desserts, candies, and sweet breads
4	4	Dairy Products	Cheeses

Delimitando a **Porção Afetada**

O **ORDER BY** pode ser usado para **primeiro ordenar** e depois **retornar os dados** necessários via cláusula TOP.

```
SELECT TOP 2  
  FirstName, LastName, BirthDate  
FROM Employees ORDER BY  
  BirthDate
```

Os dois funcionários
mais velhos!



	FirstName	LastName	BirthDate
1	Margaret	Peacock	1937-09-19 00:00:00.000
2	Nancy	Davolio	1948-12-08 00:00:00.000

```
SELECT TOP 2  
  FirstName, LastName, BirthDate  
FROM Employees ORDER BY  
  BirthDate DESC
```

Os dois funcionários
mais novos!



	FirstName	LastName	BirthDate
1	Anne	Dodsworth	1966-01-27 00:00:00.000
2	Janet	Leverling	1963-08-30 00:00:00.000

Utilizando TOP com WHERE

```
SELECT TOP 2  
FirstName, City FROM  
Employees  
WHERE City = 'London'  
ORDER BY FirstName
```



	FirstName	City
1	Anne	London
2	Michael	London

Ordena os Funcionários de Londres por ordem alfabética e retorna os dois Primeiros.



Retorna o valor do estoque dos três produtos mais caros e que estão disponíveis.



```
SELECT TOP 3  
ProductName, (UnitsinStock * UnitPrice) AS  
Valor FROM Products  
WHERE UnitsinStock > 0  
ORDER BY Valor DESC
```



	ProductName	Valor
1	Côte de Blaye	4479,50
2	Raclette Courdavault	4345,00
3	Queso Manchego La Pastora	3268,00

```
SELECT TOP 2  
ProductName, UnitPrice  
FROM Products  
WHERE CategoryID = 1  
ORDER BY UnitPrice DESC
```



	ProductName	UnitPrice
1	Côte de Blaye	263,50
2	Ipoh Coffee	46,00

Retorna os dois produtos mais caros da categoria 1.



Cláusulas **Offset** e **Fetch**

- As cláusulas **Offset** e **Fetch** também *limitam o n° de registros* em consultas;
- Obtém um **subconjunto de dados** da consulta realizada;
- Elas *obrigatoriamente* devem ser utilizadas em conjunto com o **Order By**.

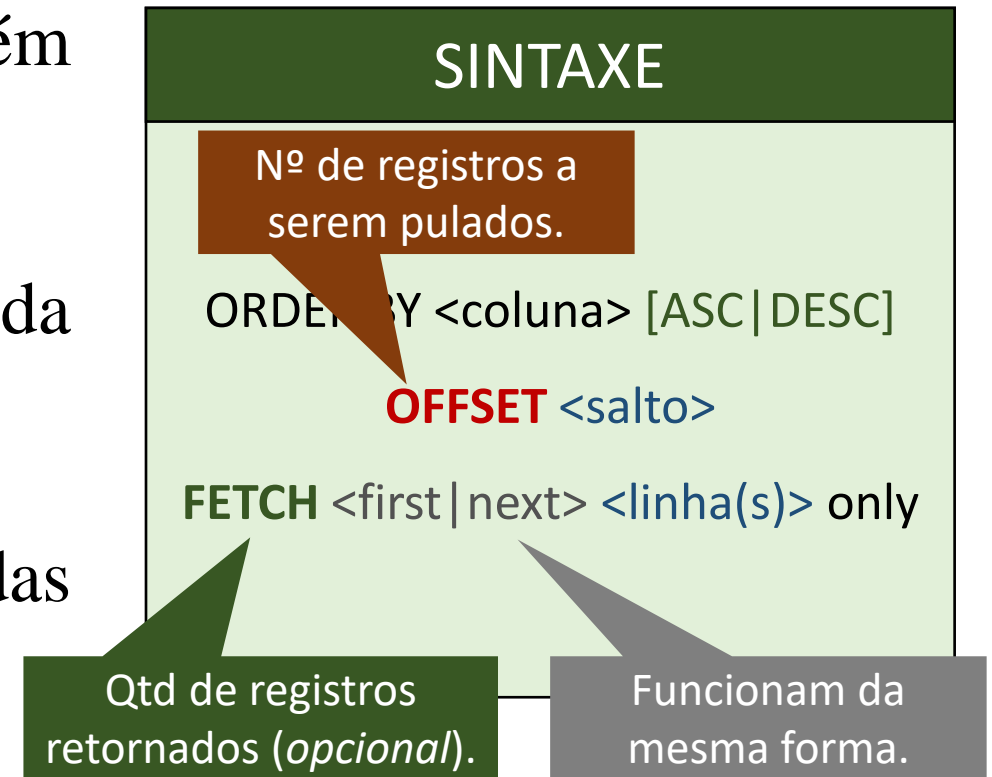
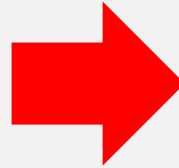


Ilustração do Offset e Fetch

OFFSET 5 ROWS



FETCH NEXT 10 ROWS ONLY



ID	Name
1	Item #1
2	Item #2
3	Item #3
4	Item #4
5	Item #5
6	Item #6
7	Item #7
8	Item #8
9	Item #9
10	Item #10
11	Item #11
12	Item #12
13	Item #13
14	Item #14
15	Item #15
16	Item #16
17	Item #17
18	Item #18
19	Item #19
20	Item #20

Será **pulado** (*ignorado*)
5 registros.

Será **retornado** os
próximos 10 registros.

Uso Somente do Offset

```
SELECT CategoryName,Description  
FROM Categories ORDER BY  
CategoryName
```



	CategoryName	Description
1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	Condiments	Sweet and savory sauces, relishes, spreads, and ...
3	Confections	Desserts, candies, and sweet breads
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and cereal
6	Meat/Poultry	Prepared meats
7	Produce	Dried fruit and bean curd
8	Seafood	Seaweed and fish

```
SELECT CategoryName,Description  
FROM Categories ORDER BY  
CategoryName OFFSET 3 ROWS
```



PULO 3 ROWS		
	CategoryName	Description
1	Dairy Products	Cheeses
2	Grains/Cereals	Breads, crackers, pasta, and cereal
3	Meat/Poultry	Prepared meats
4	Produce	Dried fruit and bean curd
5	Seafood	Seaweed and fish

Utilizando Offset com Fetch

```
SELECT CategoryName,Description  
FROM Categories ORDER BY  
CategoryName
```



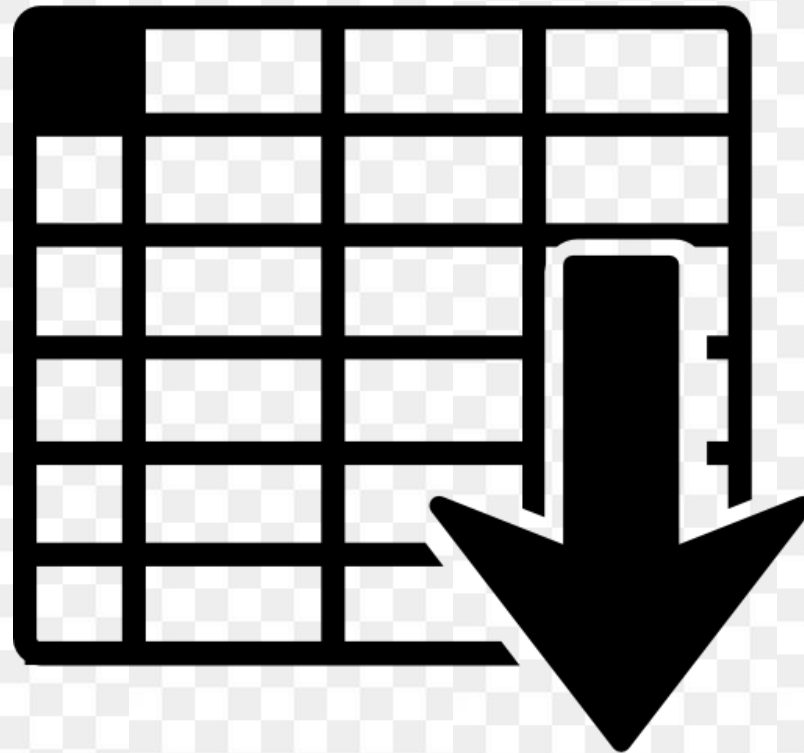
	CategoryName	Description
1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	Condiments	Sweet and savory sauces, relishes, spreads, and ...
3	Confections	Desserts, candies, and sweet breads
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and cereal
6	Meat/Poultry	Prepared meats
7	Produce	Dried fruit and bean curd
8	Seafood	Seaweed and fish

```
SELECT CategoryName,Description  
FROM Categories ORDER BY  
CategoryName OFFSET 3 ROWS  
FETCH NEXT 2 ROWS ONLY
```



PULOU 3 ROWS		
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and cereal
IGNORADO		

Restringindo Dados usados na Tabela



UNIQUE

- Sua função é **identificar unicamente** cada registro em uma tabela;
- Toda Primary Key, por padrão, *também é do tipo UNIQUE*;
- Ambas garantem a **unicidade dos elementos** armazenados;
- Você pode ter **vários campos** do tipo UNIQUE em uma tabela.

```
CREATE TABLE Pessoas
(
  id_pessoa int not null,
  nome varchar(50) not null,
  sobrenome varchar(50) not null,
  endereco varchar(100) not null,
  cidade varchar(50) not null,
  CONSTRAINT UQ_Pessoa UNIQUE(id_pessoa)
)
```

Criando a **restrição** para
coluna **id_pessoa**

CHECK

- **Limita a faixa de valores** que serão aceitos por uma coluna;
- Trata-se também de uma **validação antes da inserção**;
- O valor só é inserido se **atender ao critério** especificado.

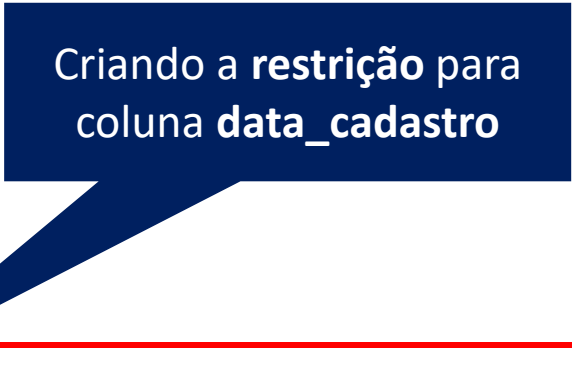
```
CREATE TABLE Pessoas
(
  id_pessoa int not null,
  nome varchar(50) not null,
  sobrenome varchar(50) not null,
  endereco varchar(100) not null,
  cidade varchar(50) not null,
  CONSTRAINT CHK_Pessoa CHECK (id_pessoa > 0)
)
```

Criando a **restrição** para
coluna **id_pessoa**

DEFAULT

Durante a inserção caso o valor **não seja informado**,
o valor **DEFAULT** será utilizado.

```
CREATE TABLE Pessoas
(
  id_pessoa int not null,
  nome varchar(50) not null,
  sobrenome varchar(50) not null,
  endereco varchar(100) not null,
  cidade varchar(50) not null,
  data_cadastro date CONSTRAINT DF_Pessoa DEFAULT GETDATE()
)
```



Criando a **restrição** para
coluna **data_cadastro**

Removendo e Adicionando Restrições

UNIQUE

```
ALTER TABLE Pessoas  
DROP CONSTRAINT  
UQ_Pessoa;
```



```
ALTER TABLE Pessoas  
ADD CONSTRAINT UQ_Pessoa UNIQUE  
(id_pessoa);
```



CHECK

```
ALTER TABLE Pessoas  
DROP CONSTRAINT  
CHK_Pessoa;
```



```
ALTER TABLE Pessoas  
ADD CONSTRAINT CHK_Pessoa CHECK  
(id_pessoa > 0);
```



DEFAULT

```
ALTER TABLE Pessoas  
DROP CONSTRAINT  
DF_Pessoa;
```

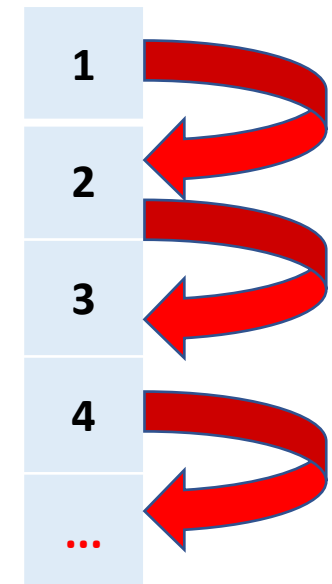


```
ALTER TABLE Pessoas  
ADD CONSTRAINT DF_Pessoa DEFAULT  
GETDATE() FOR data_cadastro;
```



Relembrando o **Atributo Identity**

- Como vimos antes, o SQL é capaz de **numerar automaticamente** colunas do **tipo inteiro**;
- É o *atributo identity* que define qual coluna receberá **numeração sequencial**;
- Essas colunas são conhecidas também como **colunas autonumeradas**;
- Pode haver apenas uma única coluna desse tipo por tabela.




Exemplo Prático de Utilização

```
CREATE TABLE Livro  
(  
  pagina int IDENTITY not null,  
  conteudo varchar(100) not null  
)
```

Inicia no 1 e
incrementa de 1 em 1

```
INSERT INTO Livro(conteudo) VALUES  
( 'Redes' ), ( 'Protocolo' ), ( 'Internet' )
```

```
SELECT * FROM Livro
```



	pagina	conteudo
1	1	Redes
2	2	Protocolo
3	3	Internet


Exemplo Prático de Utilização

Inicia no 0 e incrementa
de 10 em 10

```
CREATE TABLE Livro  
(  
  pagina int IDENTITY(0,10) not null,  
  conteudo varchar(100) not null  
)
```

```
INSERT INTO Livro(conteudo) VALUES  
( 'Redes' ), ( 'Protocolo' ), ( 'Internet' )
```

```
SELECT * FROM Livro
```



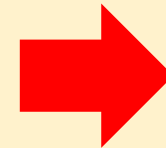
	pagina	conteudo
1	0	Redes
2	10	Protocolo
3	20	Internet

Valor Atual da Coluna Identity

```
SELECT IDENT_CURRENT('Livro');
```

OU

```
SELECT IDENT_CURRENT('dbo.Livro');
```



	UltimoValor
1	20

Desabilitar a Autonumeração

```
SET IDENTITY_INSERT Livro ON
```

Desabilita
Autonumeração

```
INSERT INTO Livro(pagina,conteudo)  
VALUES(6, 'Redes');
```

```
SET IDENTITY_INSERT Livro OFF
```

Habilita
Autonumeração

pagina	conteudo
0	Redes
10	Protocolo
20	Internet
6	Redes

Resete da Coluna Autonumerada

Mesmo apagando todos os registros da tabela, a autonumeração continua do último valor inserido.

Apagar todos os registros usando:

TRUNCATE TABLE

Executar o comando:

DBCC

CHECKIDENT(Livro, reseed, 0)

Dúvidas?

telvio.orrु@docente.unip.br



Mãos à Obra!!!



Envie por e-mail conforme o padrão apresentado na Aula 1