



Aula 16

- Cuidados ao alterar ou excluir tabelas ou dados
- Autoincremento
- Padrões
- Triggers

COMMIT E ROLLBACK

Transação no SQL Server: Uma unidade lógica de processamento que visa preservar a integridade e consistência dos dados

Sintaxe Básica

BEGIN TRANSACTION ➡ Cria um ponto de estado do banco de dados.

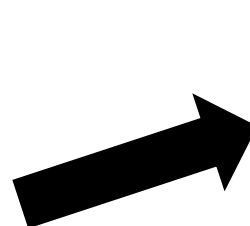
COMMIT ➡ Confirmando todas as operações entre o **BEGIN TRANSACTION** e o comando **COMMIT**. Todos os INSERTS, UPDATES ou DELETES irão ser confirmados e gravados na base de dados.

ROLLBACK ➡ Tudo que foi feito entre o **BEGIN TRANSACTION** e o **ROLLBACK** será desprezado e os dados voltarão ao status de quando o **BEGIN TRANSACTION** foi executado.

Campo auto-incremento

Tabela de Cliente

Campo Autoincremento



ID	Nome

Campo auto-incremento

Tabela de Cliente

```
INSERT INTO [TABELA DE CLIENTES] (NOME) VALUE  
( 'ROBERTO SILVA' )
```

```
INSERT INTO [TABELA DE CLIENTES] (NOME) VALUE ( 'ANDRÉ  
LUIZ' )
```

```
INSERT INTO [TABELA DE CLIENTES] (NOME) VALUE  
( 'JÉSSICA SANTOS' )
```

ID	Nome
1	ROBERTO SILVA
2	ANDRÉ LUIZ
3	JÉSSICA SANTOS

Campo auto-incremento

```
DELETE FROM [TABELA DE CLIENTES]  
WHERE ID = 1
```

Tabela de Cliente

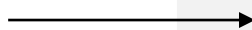
ID	Nome
2	ANDRÉ LUIZ
3	JÉSSICA SANTOS

Campo auto-incremento

Tabela de Cliente

ID	Nome
2	ANDRÉ LUIZ
3	JÉSSICA SANTOS
4	LEANDRO SILVA

```
INSERT INTO [TABELA DE CLIENTES] (NOME) VALUE  
( 'LEANDRO SILVA' )
```



Campo auto-incremento

IDENTITY [(início, incremento)]

- Quando executamos o comando **INSERT** não é necessário especificar o campo;
- Podemos ter apenas 1 campo **IDENTITY** por tabela;
- Propriedade **IDENTITY** será especificada na criação da tabela

Definindo padrões para os campos

```
[NOME DO CAMPO] VARCHAR(50) DEFAULT 'o padrão a se definir'
```


TRIGGER

O que é uma **TRIGGER**?

- Em português, significa gatilho;
- É uma ação que será disparada no momento em que um dado numa tabela é alterado;
- Pode ser uma atualização, inclusão ou exclusão.

TRIGGER

```
CREATE [ OR ALTER ] TRIGGER [ schema_name . ]trigger_name
ON { table | view }
[ WITH <dml_trigger_option> [ ,...n ] ]
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
[ WITH APPEND ]
[ NOT FOR REPLICATION ]
AS { sql_statement [ ; ] [ ,...n ] | EXTERNAL NAME <method specifier [ ; ] > }

<dml_trigger_option> ::=
    [ ENCRYPTION ]
    [ EXECUTE AS Clause ]

<method_specifier> ::=
    assembly_name.class_name.method_name
```

TRIGGER (Gatilho)

Um gatilho é um tipo especial de procedimento armazenado e executado automaticamente quando um evento ocorre no servidor de banco de dados.

Ex.: Ao incluir dados em uma tabela, atualize um log com data e hora.

```
INSERT INTO TAB1 (ID, NOME, VALOR)  
VALUES ('1', 'Pedro Santos', 10)
```



ID	NOME	VALOR
1	Pedro Santos	10
2	Jessé Maciel	20



ID	LOG
1	CLIENTE INCLUIDO
2	CLIENTE INCLUIDO

TRIGGER (Gatilho)

Um gatilho é um tipo especial de procedimento armazenado e executado automaticamente quando um evento ocorre no servidor de banco de dados.

Ex.: Ao incluir dados em uma tabela, atualize um log com data e hora.

```
DELETE FROM TAB1 WHERE ID = 1
```



ID	NOME	VALOR
2	Jessé Maciel	20



ID	LOG
1	CLIENTE INCLUIDO
2	CLIENTE INCLUIDO
1	CLIENTE EXCLUIDO

TRIGGER (Gatilho)

```
CREATE TRIGGER <NOME DA TRIGGER>  
ON [NOME DA TABELA]  
[FOR/AFTER/INSTEAD OF] [INSERT/UPDATE/DELETE]
```

<CORPO DA TRIGGER>

FOR = Executa ao mesmo tempo que o comando

AFTER = Executa depois do comando

INSTEAD OF = Executa no lugar do comando

TRIGGER (Gatilho)

```
CREATE TRIGGER <NOME DA TRIGGER>  
ON [NOME DA TABELA]  
[FOR/AFTER/INSTEAD OF] [INSERT/UPDATE/DELETE]
```

```
<CORPO DA TRIGGER>
```

FOR = Executa ao mesmo tempo que o comando

AFTER = Executa depois do comando

INSTEAD OF = Executa no lugar do comando

DESAFIO



DESAFIO

O SQL abaixo calcula a idade em anos baseado na data atual:

```
SELECT [CPF], [IDADE], [DATA DE NASCIMENTO],  
DATEDIFF(YEAR, [DATA DE NASCIMENTO], GETDATE())  
AS 'IDADE'  
FROM [TABELA DE CLIENTES]
```

	CPF	IDADE	DATA DE NASCIMENTO	IDADE
1	1471156710	27	1990-09-01	31
2	19290992743	18	2000-02-12	21
3	2600586709	18	2000-03-12	21
4	3623344710	23	1995-01-13	26
5	492472718	23	1994-07-19	27
6	50534475787	22	1995-09-11	26
7	5576228758	22	1995-11-14	26
8	5648641702	26	1991-08-30	30
9	5840119709	32	1985-03-16	36
10	7771579779	25	1992-03-25	29
11	8502682733	22	1995-10-07	26
12	8719655770	34	1983-12-20	38
13	9283760794	22	1995-10-07	26
14	94387575700	28	1989-06-20	32
15	95939180787	16	1992-01-05	29

Construa uma TRIGGER, de nome TG_CLIENTES_IDADE, que atualize as idades dos clientes, na tabela de clientes, toda vez que a tabela sofrer uma inclusão, alteração ou exclusão.

O que aprendemos nesta aula

- Como trabalham os campos de autoincremento;

- Como podemos definir padrões para os campos;



- Como desenhar os relacionamentos