### Comandos INSERT, UPDATE e DELETE no SQL Server

Os comandos **INSERT, UPDATE e DELETE** são instruções SQL fundamentais para manipulação de dados em bancos de dados relacionais. Eles fazem parte do **DML (Data Manipulation Language)**, que permite a inserção, modificação e exclusão de registros em tabelas.

# 1. Comando INSERT

O comando **INSERT** é utilizado para adicionar novos registros em uma tabela.

### Sintaxe:

```
INSERT INTO NomeDaTabela (Coluna1, Coluna2, Coluna3, ...)

VALUES (Valor1, Valor2, Valor3, ...);
```

### Exemplo de aplicação:

Imagine que temos uma tabela chamada **Clientes**, com a seguinte estrutura:

```
CREATE TABLE Clientes (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    Nome VARCHAR(100),
    Email VARCHAR(100),
    DataNascimento DATE
);
```

Agora, vamos inserir três registros na tabela.

### Comando INSERT

```
INSERT INTO Clientes (Nome, Email, DataNascimento)
VALUES ('João Silva', 'joao.silva@email.com', '1990-05-10');

INSERT INTO Clientes (Nome, Email, DataNascimento)
VALUES ('Maria Souza', 'maria.souza@email.com', '1985-08-22');

INSERT INTO Clientes (Nome, Email, DataNascimento)
VALUES ('Carlos Pereira', 'carlos.pereira@email.com', '1992-11-30');
```

### Resultado após o INSERT

| ID | Nome | Email | DataNascimento |
|---|---|---|---|
| 1 | João Silva | joao.silva@email.com | 1990-05-10 |
| 2 | Maria Souza | maria.souza@email.com | 1985-08-22 |
| 3 | Carlos Pereira | carlos.pereira@email.com | 1992-11-30 |

# 2. Comando UPDATE

O comando **UPDATE** é usado para modificar registros existentes em uma tabela.

### Sintaxe:

```
UPDATE NomeDaTabela

SET Coluna1 = NovoValor1, Coluna2 = NovoValor2, ...

WHERE Condicao;
```

**Atenção!** Sempre use a cláusula **WHERE** para evitar atualizar todos os registros da tabela acidentalmente.

### Exemplo de aplicação:

Digamos que precisamos alterar o e-mail do cliente **João Silva**.

### Antes do UPDATE:

| ID | Nome | Email | DataNascimento |
|---|---|---|---|
| 1 | João Silva | joao.silva@email.com | 1990-05-10 |
| 2 | Maria Souza | maria.souza@email.com | 1985-08-22 |
| 3 | Carlos Pereira | carlos.pereira@email.com | 1992-11-30 |

### Executando o UPDATE

```
UPDATE Clientes
SET Email = 'joao.novo@email.com'
WHERE Nome = 'João Silva';
```

### Após o UPDATE:

| ID | Nome | Email | DataNascimento |
|---|---|---|---|
| 1 | João Silva | joao.novo@email.com | 1990-05-10 |
| 2 | Maria Souza | maria.souza@email.com | 1985-08-22 |
| 3 | Carlos Pereira | carlos.pereira@email.com | 1992-11-30 |

Agora, o e-mail do **João Silva** foi atualizado com sucesso.

# 3. Comando DELETE

O comando **DELETE** remove registros da tabela.

### Sintaxe:

```
DELETE FROM NomeDaTabela

WHERE Condicao;
```

**Atenção!** Se a cláusula **WHERE** não for usada, todos os registros da tabela serão excluídos.

### Exemplo de aplicação:

Removendo o cliente **Carlos Pereira**

```
DELETE FROM Clientes
WHERE Nome = 'Carlos Pereira';
```

### Antes do DELETE:

| ID | Nome | Email | DataNascimento |
|---|---|---|---|
| 1 | João Silva | joao.novo@email.com | 1990-05-10 |
| 2 | Maria Souza | maria.souza@email.com | 1985-08-22 |
| 3 | Carlos Pereira | carlos.pereira@email.com | 1992-11-30 |

### Após o DELETE:

| ID | Nome | Email | DataNascimento |
|---|---|---|---|
| 1 | João Silva | joao.novo@email.com | 1990-05-10 |
| 2 | Maria Souza | maria.souza@email.com | 1985-08-22 |

O cliente **Carlos Pereira** foi removido da tabela.

**Dica:** Se desejar excluir todos os registros e **zerar o contador de identidade (ID)**, utilize o comando **TRUNCATE TABLE** em vez de **DELETE**:

```
TRUNCATE TABLE Clientes;
```

**Diferença entre DELETE e TRUNCATE**:

- **DELETE** permite a exclusão de registros específicos e pode ser revertido com **ROLLBACK** (se estiver dentro de uma transação).
- **TRUNCATE** remove **todos os registros de uma vez** e **não pode ser revertido**.