Aula 4

Data Manipulation Language
DML

Faculdade IMPACTA

Introdução à Data Manipulation Language
DML

Revisão: Tipos de dados

Tipos de dados determinam quais tipos de valores serão permitidos no armazenamento e os principais tipos são agrupados em categorias conforme mostrado abaixo:

- Numéricos exatos
- Numéricos aproximados
- Data e Hora
- Strings de caractere
- Caractere Unicode
- Binários
- Outros tipos

Data Definition Language

- Fator de nulidade (**NULL** ou **NOT NULL**)
- Autopreenchimento (valores autoincrementais): **IDENTITY (1,1)**
- Criação da tabela

```
CREATE TABLE <nome da tabela>
    (
    <nome coluna 1> <tipo da coluna> (<tamanho da coluna>) [NOT NULL]  ,  ...
    );
```

Regras:

- **Primary Key**

CONSTRAINT <nome da primary key> PRIMARY KEY (coluna1, ...)

- **Foreign Key**

CONSTRAINT <nome da foreign key> FOREIGN KEY (coluna1, ...)

REFERENCES <tabela da primary key> (coluna1, coluna2, ...)

Data Definition Language

Regras:

- **Unique**
- CONSTRAINT <nome da unique key> UNIQUE (coluna1, coluna2, ...)
- **Check**
- CONSTRAINT <nome da regra> CHECK (<coluna com expressão booleana>)
- **Default**
- <nome da coluna> <tipo de dados> CONSTRAINT <nome do default> DEFAULT (<valor, texto, data, função escalar>)

Data Definition Language

```
CREATE TABLE Aluno
      (
      Matricula int not null IDENTITY (500, 1)
      , Nome varchar(20)
      , CONSTRAINT pkAluno
         PRIMARY KEY (Matricula)
      );
```

Matricula	Nome
500	José
501	Pedro
502	Mario

Data Manipulation Language

Faculdade IMPACTA

Após a definição de objetos que fazem a persistência de dados, precisamos de comandos SQL que manipulem informações dentro desses objetos.

As cláusulas a seguir tratam respectivamente de inserção, modificação e eliminação de registros dentro de tabelas:

INSERT

UPDATE

DELETE

Data Manipulation Language

Faculdade IMPACTA

INSERT

- A declaração **INSERT** adiciona uma ou mais linhas em uma tabela.
- **INSERT** insere um ou mais valores (**data_values**) dentro (**INTO**) da tabela especificada (**table_or_view**).
- **column_list** é a lista de nome das colunas usadas para especificar as colunas das quais os dados são fornecidos.

Data Manipulation Language

Faculdade IMPACTA

INSERT

Sintaxe do INSERT:

```
INSERT [INTO] table_or_view [(column_list)] data_values
```

Data Manipulation Language

Faculdade IMPACTA

Declaração simples com INSERT:

```
INSERT INTO MyTable (PriKey, Description)
VALUES (1, ‘TPX450’);
```

```
INSERT INTO Production.UnitMeasure
VALUES ('F2', 'Square Feet', GETDATE());
```

Data Manipulation Language

Faculdade IMPACTA

Inserindo múltiplas linhas de dados:

```
INSERT INTO Production.UnitMeasure
VALUES ('F2', 'Square Feet', GETDATE()),
       ('Y2', 'Square Yards', GETDATE());
```

```
INSERT INTO MyTable (PriKey, Description)
VALUES (1, ‘F200’), (2, ‘GTX’), (3, ‘CS’);
```

Data Manipulation Language

INSERT usando VALUES:

```
INSERT INTO MyTable (PriKey, Description)
VALUES (1, 'Texto 1')
```

```
INSERT INTO MyTable (PriKey, Description)
VALUES (1, 'F200'), (2, 'GTX'), (3, 'CS')
```

Data Manipulation Language

Faculdade IMPACTA

INSERT usando SELECT:

```
INSERT INTO MyTable (PriKey, Description)
      SELECT ForeignKey, Description
      FROM SomeView
```

INSERT usando TOP (número de inserts):

```
INSERT TOP (1) INTO SomeTableA
      SELECT SomeColumnX, SomeColumnY
      FROM SomeTableB
```

Data Manipulation Language

Devemos lembrar que colunas com **IDENTITY** não devem ser mencionadas no **INSERT**, isso porque estas colunas são “administradas” pelo banco de dados, e não pelos usuários.

Exemplo:

```
CREATE TABLE Veiculo
    (
    idVeiculo INT IDENTITY(1,1) NOT NULL
    , Placa AS char(8) NOT NULL
    , Marca AS varchar(20)  NOT NULL
    );
INSERT INTO Veiculo (Placa, Marca) VALUES ( ‘XPT-7654’, ‘Ford’);
INSERT INTO VeiculoVALUES (‘EXH-2566’, ‘Fiat’);
```

Atividades extraclasse

- Leitura do arquivo PDF disponibilizado na plataforma