Aula 3

Data Definition Language
DDL - Continuação

Faculdade IMPACTA

Continuação de Data Definition Language DDL

Data Definition Language (DDL)

UNIQUE: Chave única é semelhante a chave primária, fazendo com que o campo envolvido seja único na tabela. Podemos ter várias chaves únicas em uma tabela, diferentemente da chave primária (que só podemos ter uma).

Por exemplo, numa tabela de Cliente, podemos ter um campo que é o número do cliente, CPF e RG. Todos os três campos não permitem repetição na tabela. Podemos eleger qualquer um desses campos como chave primária, por exemplo número do cliente.

CONSTRAINT <nome da unique key> UNIQUE (coluna1, coluna2, ...)

Data Definition Language (DDL)

Exemplo:

```
CREATE TABLE Cliente
    (
    NumCliente int not null IDENTITY (1, 1)
    , CPF int NOT NULL
    , RG int NOT NULL
    , CONSTRAINT pkCliente PRIMARY KEY (NumCliente)
    , CONSTRAINT uqClienteCPF UNIQUE (CPF)
    , CONSTRAINT uqClienteRG UNIQUE (RG)
    );
```

Data Definition Language (DDL)

Faculdade IMPACTA

Podemos alterar tabelas depois de criadas, adicionando novos campos, removendo campos existentes ou alterando tipos de dados e configurações de colunas existentes.

O comando básico para isso é o **ALTER TABLE**. Para adicionar uma coluna, utilizamos a claúsula **ADD**. Para eliminar, **DROP COLUMN**. Para alterar, **ALTER COLUMN**. Esses comandos só mexem em uma coluna por vez, ou seja, se quisermos adicionar três colunas, teremos que emitir um comando para cada coluna.

ALTER TABLE <nome da tabela> ADD <nome da coluna> <tipo de dados>

ALTER TABLE <nome da tabela> ADD CONSTRAINT <nome da constraint> ...

ALTER TABLE <nome da tabela> ALTER COLUMN <nome da coluna> <tipo de dados>

ALTER TABLE <nome da tabela> DROP COLUMN <nome da coluna>

ALTER TABLE <nome da tabela> DROP CONSTRAINT <nome da constraint>

Data Definition Language (DDL)

Exemplos:

```
ALTER TABLE Cliente ADD Nome varchar(30) NOT NULL
ALTER TABLE Cliente ADD SobreNome varchar(30) NOT NULL
```

```
ALTER TABLE Cliente ALTER COLUMN Nome varchar(50) NULL
```

```
ALTER TABLE Cliente DROP COLUMN SobreNome
```

Data Definition Language (DDL)

Para eliminar colunas existentes em uma tabela, usamos o comando **DROP TABLE**.

DROP TABLE <nome da tabela1>, <nome da tabela2>, ..., <nome da tabela n>

Note que, se uma tabela possui uma chave primária e esta chave participa de relacionamentos com outras tabelas como chave estrangeira, não será permitida a eliminação da tabela que contém a chave primária, ou seja, neste caso, precisamos eliminar as tabelas que mencionam essa chave primária,
para, depois, conseguirmos eliminar a tabela de chave primária.

Data Definition Language (DDL)

Como exemplo, temos a tabela **Aluno** (chave primária) relacionada com a tabela **Prova** (chave estrangeira). Se quisermos eliminar a tabela **Aluno**, precisamos, primeiro, eliminar a tabela **Prova**, ou retirar as constraints desses objetos, liberando a “amarração” entre eles.

```
DROP TABLE Prova, Aluno
```

Data Definition Language (DDL)

Não podem fazer referência a uma outra coluna da tabela, ou a outras tabelas, exibições ou procedimentos armazenados. As definições **DEFAULT** serão removidas quando a tabela for descartada.

<nome da coluna> <tipo de dados> CONSTRAINT <nome do default>
DEFAULT (<valor, texto, data,
função escalar>)

Exemplos:

MBAExterior VARCHAR(100) CONSTRAINT dfTextoNA DEFAULT 'Não'
Desconto DECIMAL(9, 2) CONSTRAINT dfDesconto DEFAULT 0

Data Definition Language (DDL)

Exemplo:

```
CREATE TABLE Venda
    (
    DataVenda date not null CONSTRAINT dfDataVenda DEFAULT (getdate())
    , Quantidade smallint not null CONSTRAINT dfQtd DEFAULT (1)
    , NumeroCliente int not null
    , CONSTRAINT pkVenda PRIMARY KEY (DataVenda)
    , CONSTRAINT fkVenda FOREIGN KEY (NumeroCliente)
      REFERENCES Cliente(idCliente)
    );
```

Data Definition Language (DDL)

Os tipos de dados incluem uma restrição ao preenchimento dos dados em uma coluna, assim, quando definimos uma coluna como **TINYINT**, sabemos que os valores permitidos irão de 0 a 255.

Uma coluna pode ter qualquer número de restrições **CHECK** e os critérios podem incluir diversas expressões lógicas combinadas com **AND** e **OR**. Várias restrições **CHECK** são validadas na ordem de criação.

Data Definition Language (DDL)

A avaliação do critério de pesquisa deve usar uma expressão Booleana (true/false) como base e não pode fazer referência a outra tabela.

A restrição **CHECK** no nível de coluna pode fazer referência somente à coluna restrita.

Restrições **CHECK** oferecem a mesma função de validação dos dados durante instruções **INSERT** e **UPDATE**.

Se existirem uma ou mais restrições **CHECK** para uma coluna, todas as restrições serão avaliadas.

CONSTRAINT <nome da regra> CHECK (<coluna com expressão booleana>)

Data Definition Language (DDL)

Exemplo:

```
CONSTRAINT ckIdade CHECK (Idade <= 100)

CONSTRAINT ckTaxa CHECK (Taxa >= 1 and Taxa <= 5)

CONSTRAINT CK_emp_id CHECK (emp_id LIKE
'[A-Z][A-Z][A-Z][1-9][0-9][0-9][0-9][0-9][FM]' OR emp_id LIKE
'[A-Z]-[A-Z][1-9][0-9][0-9][0-9][0-9][FM]')
```

Data Definition Language (DDL)

Exemplo:

```
create table Cliente
(
  idCliente smallint identity(-32767, 1)
  , Telefone VARCHAR(14)
  , DataEntrada datetime
  , Idade tinyint not null,
  , constraint ckIdade CHECK (Idade between 18 and 90)
  , constraint ckTelefone CHECK
  (
  Telefone LIKE '[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]' OR
  Telefone LIKE '([0-9][0-9][0-9]) [0-9][0-9][0-9]-[0-9][0-9]...)‘
  )
)
```

Data Definition Language (DDL)

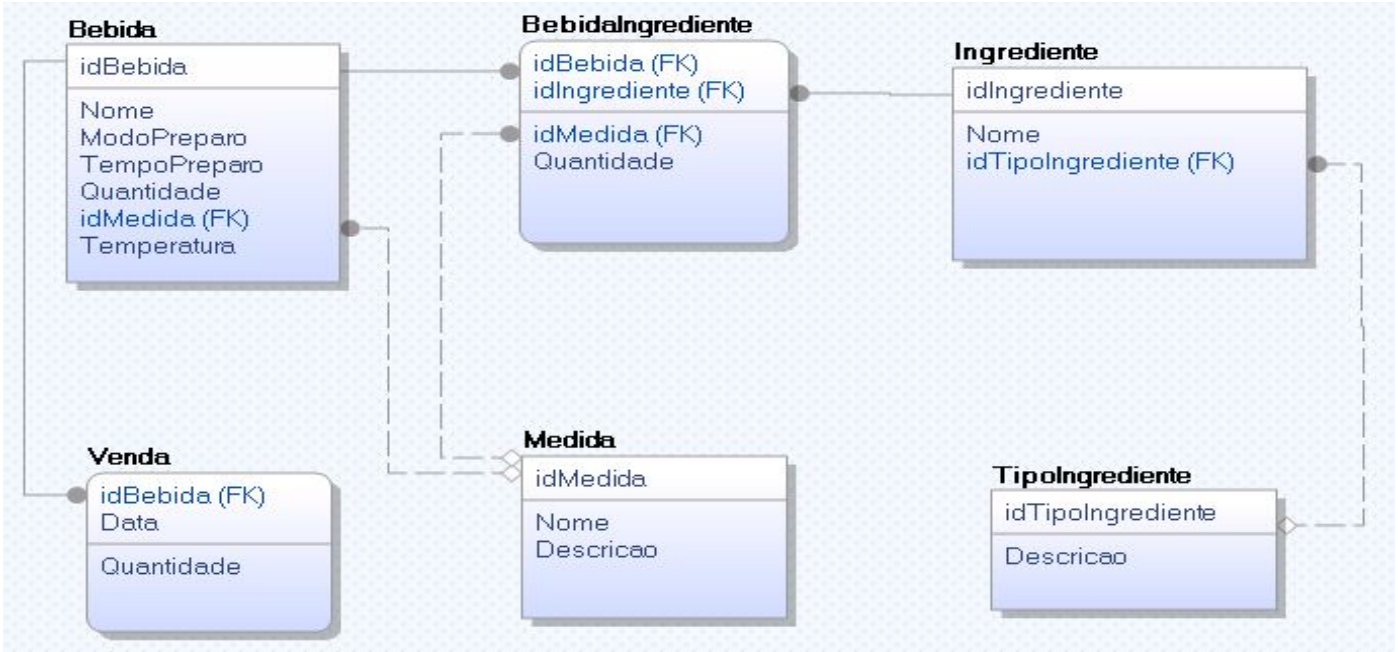
As Constraints Primary key, Foreing Key, Unique, Default e Check, podem ser incluídas ou retiradas de uma tabela utilizando os mesmos comandos mencionados anteriormente. Exemplo:

```
ALTER TABLE Cliente Add Constraint ckIdade CHECK (idade between 18 and 90)
```

```
ALTER TABLE Venda Alter Column DataVenda date Constraint dfDtVenda DEFAULT (getdate())
```

```
ALTER TABLE Cliente Drop Constraint ckIdade
```

Modelo para Restaurante
Venda de Drinks



Atividades extraclasse

- Leitura do arquivo PDF disponibilizado na plataforma