

FGA0137

# Sistemas de Banco de Dados 1

Prof. Maurício Serrano

Material original: Profa. Elaine Parros Machado de Sousa

Prof. Jose Fernando Rodrigues Junior

**2021/2**

# Linguagem SQL

## DDL

Módulo 4

# SQL - Introdução

- **SQL**

- linguagem declarativa  $\Rightarrow$  não procedural
- desenvolvida e implementada pelo laboratório de pesquisa da IBM em San Jose – década de 70
- inicialmente chamada SEQUEL (*Structured English QUery Language*)
- criada como interface entre usuários e o primeiro SGBDR – SYSTEM R

# SQL - Introdução

- Padrão industrial que atinge grande parte do mercado de SGBDs
  - atrativo: pequena quantidade de comandos para realizar uma grande quantidade de operações
    - simplicidade
    - grande poder de consulta
  - padrão facilita migração

# SQL - Introdução

- O padrão SQL
  - **SQL 2016**
  - **SQL 2011**
  - **SQL 2008**
  - **SQL 2006**
  - **SQL 2003**
  - **SQL99 (SQL3)**
  - **SQL92 – SQL2**
  - **SQL86**

# SQL

- Dois conjuntos principais de comandos:
  - **DDL – *Data Definition Language*** :  
especificação do esquema da base de dados
  - **DML – *Data Manipulation Language*** :  
inserção, remoção, alteração e consultas na instância da base de dados

# SQL

- Recursos:
  - DDL – *Data Definition Language*
  - DML – *Data Manipulation Language*
  - definição de restrições de integridade
  - criação de visões (*views*)
  - especificações de segurança e autorizações
  - controle de transação
  - regras para integração com linguagens de programação
  - ....

# DDL e DML

- DDL: CREATE, DROP, e ALTER
  - TABLE, DATABASE, DOMAIN, EXCEPTION, GENERATOR, INDEX, PROCEDURE, ROLE, SHADOW, TRIGGER, E VIEW
- DML: SELECT (FROM), DELETE (FROM), INSERT (INTO), UPDATE



# DDL e DML

- DDL: Criação, alteração e remoção de objetos de banco de dados.
  - TABLE, INDEX, VIEW, SEQUENCE, SYNONYM, TRIGGER, GRANT, REVOKE
- DML: SELECT (FROM), DELETE (FROM), INSERT (INTO), UPDATE

Existem variações entre fabricantes, tanto com relação ao conjunto de funcionalidades quanto a como estas funcionalidades operam.

# DDL - Introdução

- **Alguns comandos da DDL:**

- CREATE TABLE
- ALTER TABLE
- DROP TABLE
- CREATE DOMAIN
- ALTER DOMAIN
- DROP DOMAIN
- CREATE VIEW
- ALTER VIEW
- DROP VIEW
- CREATE INDEX
- ALTER INDEX
- DROP INDEX
- ...

# Comandos DDL

- **CREATE TABLE** - criar uma tabela, definir colunas e restrições

```
CREATE TABLE tabela (  
    atrib1 tipo [<restrições da coluna 1>],  
    atrib2 tipo [<restrições da coluna 2>],  
    ....  
    atribn tipo [<restrições da coluna n>],  
  
    <restrições da tabela>  
);
```

# SQL – Alguns tipos de dado

- **INTEGER | SMALLINT | NUMBER**
- **DECIMAL [(precision, scale)]**
  - *precision* - número total de dígitos
  - *scale* - número de dígitos depois do ponto
- **DOUBLE PRECISION | FLOAT | REAL**
- **CHAR(n)** - tamanho fixo - n caracteres
- **VARCHAR(n)** - tamanho variável
  - máximo de n caracteres
- **BLOB** – *Binary Large Object* –
- **DATE | TIME | TIMESTAMP**
- ...

# CREATE TABLE

- Restrições de colunas
  - NOT NULL
  - DEFAULT *valor*
  - CHECK(*condição*)

```
CREATE TABLE tabela (  
    atrib1 tipo [(tamanho)] [NOT NULL | DEFAULT valor]  
        [CHECK (condição)],  
    atrib2 tipo [(tamanho)] [NOT NULL | DEFAULT valor]  
        [CHECK (condição)],  
    . . .
```

# CREATE TABLE

- Restrições de tabela

- PRIMARY KEY ( *<atributos chave primária>* )
- UNIQUE ( *<atributos chave candidata>* )
- FOREIGN KEY ( *<atributos chave estrangeira>*  
REFERENCES *tabelaRef* [ (*<chave primária>*) ]

[*<ações>*]

- *<ações>*

- ON DELETE | ON UPDATE

- CASCADE | SET NULL | SET DEFAULT

- CHECK (*condição*)

# CREATE TABLE

Há grandes diferenças entre os diferentes fabricantes.

- Res

- I

Em Oracle, por exemplo, não são possíveis:

- U

-ON UPDATE

- I

-SET DEFAULT

- E

Em Borland InterBase, é necessário que um atributo que será chave seja explicitamente declarado como NOT NULL, o que não é necessário nem em Oracle nem em PostgreSQL.

- C

> )

ira>

) ]

## ■ CREATE TABLE

```
CREATE TABLE tabela (  
    atrib1 tipo [(tamanho)] [NOT NULL | DEFAULT valor]  
        [CHECK (condição)],  
    atrib2 tipo [(tamanho)] [NOT NULL | DEFAULT valor]  
        [CHECK (condição)],  
    ...  
    [CONSTRAINT nome da restrição  
        PRIMARY KEY (<atributos chave primária>),  
    [CONSTRAINT nome da restrição  
        UNIQUE (< atributos chave candidata>),  
    [CONSTRAINT nome da restrição  
        FOREIGN KEY (<atributos chave estrangeira>  
            REFERENCES tabelaRef [(<chave primária>)]  
            [ON DELETE CASCADE | SET NULL | SET DEFAULT]  
            [ON UPDATE CASCADE | SET NULL | SET DEFAULT],  
    [CONSTRAINT nome da restrição  
        CHECK (condição)  
);
```



# Exercício

- Criar as tabelas para o seguinte esquema

Aluno = {Nome, NMatr, Idade, DataNasc}

Professor = {Nome, NFunc, Idade, Titulação}

Disciplina = {Sigla, Nome, NCred, Professor, Livro}

Turma = {Sigla, Numero, NAlunos}

Matrícula = {Sigla, Numero, Aluno, Ano, Nota}

# Comandos DDL

- **ALTER TABLE** – incluir/alterar/remover definições de colunas e restrições

**ALTER TABLE *tabela* <ação>;**

■ **<ação>:**

- **ADD *novoStrib* *tipo* [<restrições de coluna>]**
- **ADD [CONSTRAINT *nome*] <restrição de tabela>**
- **DROP *atributo* [CASCADE | RESTRICT]**
- **DROP CONSTRAINT *nome***
- **ALTER *atributo* SET DEFAULT <valor>;**
- **ALTER *atributo* DROP DEFAULT;**

# ALTER TABLE

- **ADD *novoAtrib tipo* [<restrições de coluna>]**
- **DROP *atributo* [CASCADE | RESTRICT]**
  - **CASCADE** – todas as visões e restrições (*constraints*) que referenciam o atributo são removidas automaticamente
  - **RESTRICT** – atributo só é removido se não houver nenhuma visão ou restrição que o referencie

# Exercício

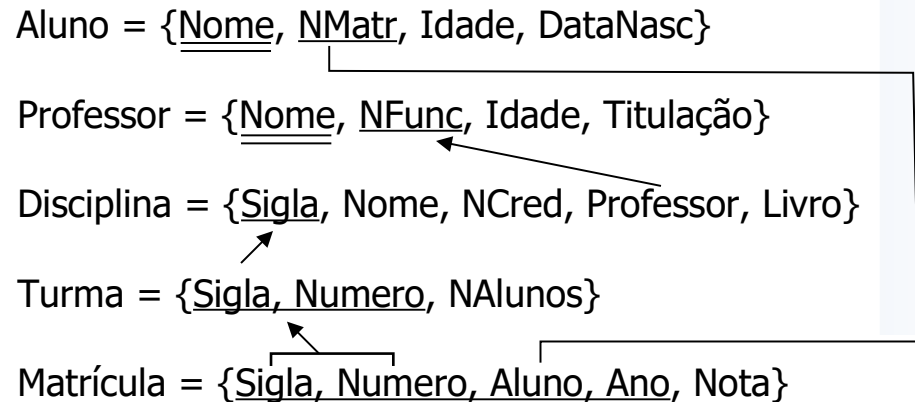
Aluno = {Nome, NMatr, Idade, DataNasc}

Professor = {Nome, NFunc, Idade, Titulação}

Disciplina = {Sigla, Nome, NCred, Professor, Livro}

Turma = {Sigla, Numero, NAlunos}

Matrícula = {Sigla, Numero, Aluno, Ano, Nota}



- Modificar as tabelas para:
  - acrescentar o atributo Cidade de Origem para a tabela Aluno
  - remover o atributo Numero da tabela Turma
    - como fica a tabela matrícula?
  - Alterar o atributo Numero da tabela Turma para um atributo Letra (turma A, B...)
  - acrescentar uma restrição para que a nota de um aluno seja sempre maior ou igual a 0 e menor ou igual a 10
  - remover uma restrição da tabela Disciplina

# Comandos DDL

- **DROP TABLE** - exclui uma tabela da base de dados

**DROP TABLE *tabela* [CASCADE | RESTRICT] ;**

- **CASCADE**: todas as visões e restrições que referenciam a tabela são removidas automaticamente
- **RESTRICT**: a tabela é removida somente se não for referenciada em nenhuma restrição ou visão

# Comandos DDL

- **CREATE DOMAIN** – cria um nome de domínio para especificar um tipo de dado (álías)

**CREATE DOMAIN *nome* AS *tipo***  
**[<restrições de coluna>]**

- facilita alterações de tipos de dados
- simplifica legibilidade

# CREATE DOMAIN - exemplo

- `CREATE DOMAIN CUSTNO  
AS INTEGER  
DEFAULT 9999  
CHECK (VALUE > 1000);`
- `CREATE DOMAIN PRODTYPE  
AS VARCHAR(12)  
CHECK (VALUE IN ('software', 'hardware', 'other', 'N/A'));`

# Exercício

- Modificar as tabelas para:
  - incluir em Matrícula um atributo de frequência do aluno, cujo tipo é um domínio Frequencia.

Aluno = {Nome, NMatr, Idade, DataNasc, CidadeOrigem}

Professor = {Nome, NFunc, Idade, Titulação}

Disciplina = {Sigla, Nome, NCred, Professor, Livro}

Turma = {Sigla, Letra, NAlunos}

Matrícula = {Sigla, Letra, Aluno, Ano, Nota}

```
graph TD; NMatr --> NFunc; NFunc --> Professor; Professor --> Letra; Letra --> Aluno;
```



# Leitura recomendada

- R. Elmasri, S. Navathe: *Fundamentals of Database Systems* – 4th Edition
  - Capítulo 8
- A. Silberschatz, H. F. Korth, s. Sudarshan: *Sistema de Banco de Dados*
  - Capítulo 4