

# **Modelagem de Banco de Dados**

## **- SQL introdução**

Profa. Letícia T. M. Zoby

[leticia.zoby@udf.edu.br](mailto:leticia.zoby@udf.edu.br)

# Introdução - SQL

- O nome SQL é derivado de *Structured Query Language* ou Linguagem de Consulta Estruturada.
- Originalmente chamada de SEQUEL (*Structured English QUery Language*).
- Desenvolvida pela IBM nos anos 70.
- Padronizada pela ANSI/ISSO.
- Linguagem declarativa

# Conceitos Iniciais

- SQL oferece um conjunto de comandos para uma variedade de tarefas diferentes, incluindo:
  - pesquisar, inserir, alterar e deletar linhas de uma tabela;
  - criar, deletar e alterar objetos de banco de dados;
  - controlar o acesso aos dados e objetos;
  - garantir a consistência dos dados.
- SQL é utilizada por todos os tipos de usuários de um banco de dados relacional:
  - Administrador de sistemas;
  - Administrador de banco de dados;
  - Desenvolvedores de sistemas;
  - Usuários finais.

# SQL e Linguagens de Programação

- Os comandos SQL podem ser embutidas em outra linguagem e utilizados para o acesso ao banco de dados
  - Java, C#, Delphi
- Driver de conexão ao SGBD
  - JDBC (Java Database Connectivity)
  - ODBC (Open Database Connectivity)

# Regras Básicas

- Regras Básicas para escrever comandos SQL
  - Os comandos podem ser escritos em mais de uma linha;
  - Cláusulas diferentes são colocadas usualmente em linhas diferentes;
  - Podem ser usadas tabulações;
  - Comandos podem ser escritos em letras maiúsculas ou minúsculas.

```
select * from cliente;
```

```
select  
*  
from cliente;
```

```
SELECT *  
from cliente;
```

# Regras Básicas

- “Regras para Nomeação”
  - Deve começar com uma letra
  - Pode ter de 1 a 30 caracteres
  - Deve conter somente A–Z, a–z, 0–9, \_, \$ e #
  - Não deve duplicar o nome de outro objeto de propriedade do mesmo usuário
  - Não deve ser uma palavra reservada do SGBD

# Comunicando-se com um SGBD Relacional usando SQL

A instrução SQL é informada

```
SQL> SELECT loc  
2   FROM  dept;
```

A instrução é enviada para o banco de dados

Banco de Dados

Os dados são exibidos

LOC
-----
NEW YORK
DALLAS
CHICAGO
BOSTON

# Sublinguagens de SQL

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Query Language (DQL)
- Data Control Language (DCL)
- Data Transaction Language (DTL)

# Linguagem DDL

- Data Definition Language (DDL)
- Linguagem de definição de dados
- Contém comandos que criam, modificam e excluem objetos de banco de dados.
- São exemplos de comando:
  - CREATE, ALTER, DROP e TRUNCATE.

# Linguagem DML

- Data Manipulation Language (DML)
- A linguagem de manipulação de dados
- Fornece instruções para trabalhar com os dados armazenados como SELECT, INSERT, UPDATE e DELETE.

# Linguagem DQL

- Data Query Language (DQL)
- A linguagem de consulta de dados é um subconjunto da DML que possui apenas a instrução de SELECT.

# Linguagem DCL

- Data Control Language (DCL)
- A linguagem de controle de dados contém os comandos relacionados com as permissões de controle de acesso.
- Garante os privilégios aos usuários para acessar os objetos do banco.
- Os mais conhecidos comandos são o GRANT e o REVOKE.

# Linguagem DTL

- Data Transaction Language (DTL)
- Linguagem de transação de dados
- Inclui comandos de COMMIT, ROLLBACK e SAVEPOINT

# Extensão de SQL: Linguagem Procedural

- Motivos para incluir uma linguagem procedural no SQL
  - Permitir o desenvolvimento de definições mais complexas no banco de dados.
  - Desempenho.
- Dialetos
  - **PL/SQL** - Encontrado em SGBDs Oracle. PL/SQL, que é a sigla para Procedural Language/SQL, contém muitas semelhanças com a linguagem de programação geral.
  - **Transact-SQL** - Usado pelo Microsoft SQL Server e Sybase Adaptive Server. Como Microsoft e Sybase se afastaram, a plataforma comum que compartilhavam no início na década de 1990 foi dividida, suas implementações de agora também divergem, produzindo dois dialetos distintos de Transact-SQL.
  - **SQL PL** - extensão processual do IBM DB2 para SQL, introduzido na versão 7.0 do SGBD, fornece construções necessárias para a execução de controle de fluxo em torno de consultas SQL tradicionais e operações.
  - **PL/pgSQL** - O nome do dialeto SQL e das extensões implementadas no PostgreSQL. A sigla significa Procedural Language/PostgreSQL.
  - **MySQL** - O MySQL introduziu uma linguagem procedural em seu banco de dados na versão 5, mas não há nenhum nome oficial para ela. É concebível que, como a Oracle adquiriu MySQL, pode introduzir PL/SQL como parte do MySQL.

# Comandos Estudados

- CREATE TABLE: cria uma nova tabela com seus campos e define as restrições de campo.
- Domínio:
  - Restrições
    - NOT NULL: Restrição aplicada em colunas cujos valores não podem ser nulos
    - Valor Default: Usado para inicializar o valor de uma coluna
    - **DEFAULT <valor>** logo após a restrição
  - Tipos de Domínios em SQL
    - CHAR(n). Cadeia de caracteres de tamanho fixo.
    - VARCHAR(n). Cadeia de caracteres de tamanho variável com comprimento máximo especificado.
    - INT. Inteiros (normalmente de 4 bytes).

# Comandos Estudados

- Tipos de Domínios em SQL
  - CHAR(n). Cadeia de caracteres de tamanho fixo.
  - SMALLINT. Inteiros pequenos (um subconjunto do domínio dos inteiros).
  - NUMERIC(p, d). Número de ponto fixo, com precisão e número de casas decimais.
  - REAL, DOUBLE PRECISION. Números de ponto flutuante e ponto flutuante de precisão dupla.
  - FLOAT(n). Número de ponto flutuante com precisão de no mínimo n dígitos.
  - DATE. Datas, contendo ano, mês e dia.
  - TIME. Tempo, em horas, minutos e segundos.
- Observação: Cada fabricante de SGBD adiciona seus próprios domínios (exemplos: boolean, character varying, serial, timestamp).
- Documentação PostgreSQL: <https://www.postgresql.org/docs/>

# Comandos - Criação de Tabelas

- O comando DDL para criar uma tabela deve conter os nomes das colunas, os tipos dos seus dados e os tamanhos dos dados a serem introduzidos.
- Sintaxe:

```
CREATE TABLE nome-da-tabela(  
    nome-da-coluna1 tipo-do-dados [(tamanho)] [restrição-coluna]  
    nome-da-coluna2 tipo-do-dados [(tamanho)] [restrição-coluna]  
    (...)  
);
```

Restrição: é um mecanismo pelo qual você limita ou restringe o tipo de dado que uma coluna pode armazenar

# Comandos - Criação de Tabelas

- Na criação da tabela é possível especificar vários tipos de restrições:
  - Chave Primária: PRIMARY KEY;
  - Chave Estrangeira: FOREIGN KEY;
  - Chave Alternativa (ou alternada): UNIQUE;
  - Restrição de Domínio: CHECK.
- Pode-se atribuir nomes às restrições de integridade:
  - CONSTRAINT NOME\_RESTRIÇÃO TIPO RESTRIÇÃO

EX:

**CREATE TABLE nome-da-tabela(**

**nome-da-coluna1 tipo-do-dados [(tamanho)] [restrição-coluna]**

**PRIMARY KEY <lista\_de\_Colunas>,**

**FOREIGN KEY <nome\_da\_coluna> REFERENCES  
            <nome\_tab\_ref>(<nome\_da\_coluna\_ref>);**

**);**

**REFERENCES**

# Comandos - Inserção

- Usado para inserir linhas individuais de dados num banco de dados já existente.
- Sintaxe:

```
INSERT INTO nome-da-tabela(  
    nome-da-coluna1, nome-da-coluna2, ...)  
VALUES (valor1, valor2,...);
```

# Comandos - Atualização

- Usado para alterar valores em linhas já existentes.
- Sintaxe:

```
UPDATE nome-da-tabela(
```

```
    SET coluna1=novo_valor, coluna2=novo_valor, ...
```

```
    WHERE condição;
```

# Comandos - Exclusão

- Usado para remover linhas de uma tabela.
- Sintaxe:

**DELETE**

**FROM nome-da-tabela**

**WHERE condição;**

# Bibliografia

- ELMASRI, R.; NAVATHE, S. B. Sistemas de Banco de Dados. 7 ed. São Paulo: Pearson Education do Brasil, 2018.

