BANCO DE DADOS
SQL

Luciane Agliardi

INSTITUTO FEDERAL
Rio Grande do Sul
Campus Erechim

Sistema Gerenciador de Banco de Dados – SGBD

É uma coleção de programas que permitem ao usuário definir, construir e manipular bases de dados para as mais diversas finalidades.

Funções Básicas:

- Métodos de acesso

Permitir que seja possível manipular e editar um banco de dados

- Integridade Semântica

Garantir que o tipo de dado de um atributo no momento da criação, seja o mesmo na hora da inserção.

RDBMS

- RDBMS significa Sistema de Gerenciamento de Banco de Dados Relacional.
- O RDBMS é a base do SQL e de todos os sistemas de banco de dados modernos, como MS SQL Server, IBM DB2, Oracle, MySQL e Microsoft Access.
- Os dados no RDBMS são armazenados em objetos de banco de dados chamados tabelas.
- Uma tabela é uma coleção de entradas de dados relacionadas e consiste em colunas e linhas.

Linguagem SQL

Structured Query Language (Linguagem de Consulta Estruturada)

- SQL é uma linguagem padrão para armazenar, manipular e recuperar dados em bancos de dados.
- O padrão SQL define precisamente uma interface SQL para:
  - definição de um banco de dados e tabelas;
  - operações sobre as mesmas (seleção, projeção, junção e outras);
  - definição de regras de integridade de bancos de dados.
- A maioria das ações que você executa em um banco de dados são realizadas com instruções SQL.

Linguagem SQL

- Na década de 70 a IBM desenvolveu uma linguagem chamada **SEQUEL** (Structured English Query Language) para manipulação de bancos de dados relacionais.
- O SQL tornou-se um **padrão** do American National Standards Institute (ANSI) em 1986 e da International Organization for Standardization (ISO) em 1987.
- Embora SQL seja um padrão ANSI/ISO, existem diferentes versões da linguagem SQL.
- Para estar em conformidade com o padrão ANSI, todos eles suportam pelo menos os principais comandos (como SELECT, UPDATE, DELETE, INSERT, WHERE) de maneira semelhante.

Linguagem SQL

A linguagem SQL é subdividida em: DDL, DML e DCL

DDL (Data Definition Language) – Linguagem de Definição de Dados.

- Define a estrutura de banco de dados, cria, modifica e remove objetos do esquema (banco de dados).

DML (Data Manipulations Language) – Linguagem de Manipulação de Dados.

- Com ela é possível inserir dados em um BD;
- Modificar dados;
- Realizar consultas que retornem uma informação do banco de dados.

DCL (Data Control Language) – Linguagem de Controle dos Dados

- Controle de acesso, adiciona e remove permissões de usuários sobre tabelas do BD.

COMANDOS DDL

São responsáveis por manipular objetos em um BD. É possível criar, alterar e excluir toda a estrutura necessária para o BD funcionar, tais como tabelas, visões, etc.

- CREATE
- DELETE
- ALTER
- DROP

COMANDOS DML

Tem como função principal manipular os dados presentes no BD. Permitem inserir, alterar e excluir os registros de um banco de dados.

- INSERT
- DELETE
- UPDATE
- SELECT

COMANDOS DCL

Tem como função principal controlar a parte de segurança do banco de dados. (Dar e retirar permissões)

- GRANT
- REVOKE

TIPOS DE DADOS COMUNS

CHAR ou Character (tamanho exato, sexo CHAR(1))

INT ou Integer

DEC abreviação para Decimal (6,2)

DATETIME OU TIMESTAMP(captura a hora atual) E TIME

DATE

VARCHAR (até 255, adapta ao tamanho)

BLOB (grandes blocos de dados de texto, exemplo comentários)

RESTRIÇÕES EM SQL

- Restrições de chave e integridade referencial
  - PRIMARY KEY (chave primária)
  - UNIQUE (chave candidata)
  - FOREIGN KEY (chave estrangeira)
- Restrições sobre domínios de atributos
  - NOT NULL
  - DEFAULT (valor padrão)
  - CHECK (verifica limite de valores de atributos ou domínios)

Alguns dos comandos SQL mais importantes

- SELECT- extrai dados de um banco de dados
- UPDATE- atualiza dados em um banco de dados
- DELETE- exclui dados de um banco de dados
- INSERT INTO- insere novos dados em um banco de dados
- CREATE DATABASE- cria um novo banco de dados
- ALTER DATABASE- modifica um banco de dados
- CREATE TABLE- cria uma nova tabela
- ALTER TABLE- modifica uma tabela
- DROP TABLE- apaga uma tabela
- CREATE INDEX- cria um índice (chave de pesquisa)
- DROP INDEX- exclui um índice

COMANDO PARA CRIAR UM BD

CREATE DATABASE nome ;

CREATE DATABASE alunos;

COMANDO PARA USAR UM BD

USE nome ;

USE alunos;

COMANDO PARA CRIAR UMA TABELA

-
- CREATE TABLE nome_tabela (
  nome_campo1 INT,
  nome_campo2 VARCHAR(255)
  );

COMANDO PARA CRIAR UMA TABELA

```
CREATE TABLE alunos (
    nome VARCHAR (50),
    e-mail VARCHAR (50),
    aniversario DATE,
    profissao VARCHAR  (50),
    local VARCHAR (50),
    estado_civil VARCHAR (20)
    );
```

IMPORTANTE

- Use o comando CREATE DATABASE para criar o banco de dados que conterá todas as tuas tabelas;
- Use o comando USE DATABASE para entrar no banco de dados a fim de criar suas tabelas;
- Todas as tabelas são criadas com o comando CREATE TABLE, contendo os nomes da coluna e dos tipos de dados correspondentes;
- Alguns dos tipos de dados mais comuns são CHAR, VARCHAR, INT, DEC e DATE.

COMANDO DROP

- DROP DATABASE nomebd;

  Remover um banco de dados SQL existente.

- DROP TABLE alunos;

  Apaga sua tabela e os dados inseridos nela!

  Funcionará independentemente da existência de dados na tabela, então use o comando com atenção extrema. Uma vez apagada, a tabela já era, assim como os dados inseridos nela.

COMANDO DESC

Utilizado para ver a estrutura de sua tabela!

DESC nome_tabela;
(Sua tabela DESCrita)

COMANDO INSERT

Para inserir dados na tabela você usará o comando INSERT INTO

INSERT INTO your_table (column_name1,column_name2, …) VALUES (‘value1’, ‘value2’,…);

Onde:

INSERT INTO (inicia a declaração)

your_table (nome da sua tabela, meus_contatos)

(column_name1,column_name2, …) (nomes das colunas da tabela, separadas por vírgulas. Como nome, e-mail, telefone, etc)

VALUES (indica que os valores para as colunas a seguem)

(‘value1’, ‘value2’,…) (lista dos valores , separadas por vírgulas. Os valores precisam estar na mesma ordem que os nomes das colunas. As aspas simples quando estiver inserindo texto. Não há vírgula no último valor)

; (o ponto-e-vírgula encerram o código)

COMANDO INSERT INTO

Variações para o comando INSERT INTO

1) Mudando a ordem das colunas: você pode mudar a ordem das colunas, desde que os respectivos valores para cada coluna venham na mesma ordem!

2) Omitindo nomes das colunas: você pode omitir a lista de nomes das colunas, mas os valores devem estar todos ali, na mesma ordem que adicionou as colunas.

3) Deixando algumas colunas de fora: quando queremos iserir apenas parte dos nossos dados, neste caso, você deverá especificar as colunas. Nas colunas em que não foram inseridos dados aparecerá NULL.

4) Inserir várias linhas: inserir várias linhas de dados na mesma instrução, com vários valores. Separar cada conjunto de valores com uma vírgula.

NULL e NOT NULL

- Um Valor NULL é um valor indefinido. Um campo com valor NULL é um campo sem valor.
- Ele não se iguala a zero ou a valor vazio. Uma coluna com um valor NULL é NULL, mas nunca igual a outro NULL. Aparece em toda coluna sem valor inserido.

  Para não aceitar um valor NULL é necessário usar NOT NULL quando criar a tabela.

```
CREATE TABLE alunos
(
nome VARCHAR (30) NOT NULL,
endereço VARCHAR (30) NOT NULL
);
```

DEFAULT

Usar um valor DEFAULT preenche as colunas vazias com um valor específico.

Quando você tem uma coluna que sabe que comumente tem um valor específico, podemos atribuir um valor padrão DEFAULT.

O valor que segue a palavra-chave DEFAULT é automaticamente inserido na tabela cada vez que uma linha é inserida, caso nenhum outro valor seja atribuído.

O valor padrão deve ser do mesmo tipo de dados atribuído na coluna.

COMANDO SELECT

Seleciona dados de uma ou mais tabelas

SELECT * from meus_contatos;

SELECT *

É um pedido ao sistema SQL para selecionar todas as colunas da tabela

COMANDO SELECT

E se eu não quiser selecionar todas as colunas? Posso usar outra coisa ao invés do asterisco?

Sim. O asterisco seleciona tudo, mas posso selecionar apenas algumas colunas, fazendo seus resultados ainda mais fáceis de serem interpretados.

Cláusula Where

- Permite que o SQL algo específico para procurar
- Limita os resultados para nós e exibe somente as linhas que são compatíveis com a condição estabelecida.
- O sinal de igual testa se cada valor da coluna é igual ou compatível

Cláusula Where

SELECT * FROM meus_contatos WHERE nome= ‘Ana’;

* (diz ao SQL para retornar os valores de todas as colunas da tabela)

meus_contatos (nome da tabela)

WHERE (diz ao software que você quer procurar por algo específico)

nome (dirá que quer procurar apenas valores inseridos na coluna nome)

= (quer dizer é em SQL)

‘Ana’ (o valor para sua coluna, aspas simples para textos)

; (finaliza)

Cláusula Where

- Para escrever cláusulas WHERE válidas, é preciso certificar-se que cada tipo de dados está formatado corretamente, como:
- Sempre usa aspas simples: Char ou character, Varchar, Datetime, Time ou Timestamp, Blob e Date
- Nunca usa aspas simples: Dec ou Decimal e Int ou Integer

Cláusula Where

Selecionar dados específicos:

Seleciona apenas as colunas que desejo ver!

```
SELECT nome, e-mail, telefone
FROM  meus_contatos
WHERE cidade = ’Erechim’;
```

IMPORTANTE

- Use aspas simples nas suas cláusulas WHERE quando for selecionar campos de textos.
- Não use aspas simples quando for selecionar campos numéricos.
- Use o * no seu comando SELECT quando quiser selecionar todas as colunas.
- Quando puder, selecione colunas específicas na sua tabela ao invés de usar SELECT *.