



# Banco de Dados

## Structured Query Language - SQL

Prof. Walteno Martins Parreira Jr

[www.waltenomartins.com.br](http://www.waltenomartins.com.br)

[waltenomartins@yahoo.com](mailto:waltenomartins@yahoo.com)

2015

# A Origem

Em 1970, Ted Codd (pesquisador da IBM) publicou o primeiro artigo sobre bancos de dados relacionais. Este artigo tratava sobre uma forma de consultar os dados (em tabelas) por meio de comandos.

A IBM criou em 1974 um grupo de pesquisa chamado System R, com o objetivo de desenvolver um sistema de BD para ser comercializado.

Em 1976 o System R introduziu uma linguagem chamada Structured Query Language (SQL).

# A Origem

A linguagem SQL deriva diretamente do SEQUEL (*Structured English Query Language*).

O SEQUEL tinha como objetivo se tornar uma linguagem universal (cujos comandos originaram do idioma inglês), usada para acessar os dados armazenados em um banco de dados, ignorando as complicações derivadas da estrutura interna de funcionamento de cada um.

A primeira versão usada foi distribuída em 1981 pela IBM e foi denominada *Structured Query Language / Data System (SQL/DS)*.

# A Origem

No caso do SQL, está-se falando de uma linguagem um pouco diferente das outras, porque só pode ser usada para acessar dados em um banco de dados, isto é, trata-se de uma linguagem específica para a manipulação de tabelas de dados.

O SQL não serve para criar rotinas de procedimentos a serem executados pelo computador, e sim, para informar quais dados (ou conjuntos de dados) se quer manipular.

A finalidade do SQL é acessar dados, independentemente do tipo de hardware ou software que está sendo utilizado.

# A Origem

O SQL é uma linguagem que possui seus comandos, seus parâmetros e sua sintaxe específicos.

A integridade dessas características é constantemente monitorada e mantida por um comitê, o *American National Standards Institute (ANSI)*.

O primeiro padrão para o SQL foi publicado em 1986, e em 1989 teve sua primeira atualização.

Em 1992, houve outra atualização, que se tornou conhecida como SQL92, ou simplesmente SQL2.

A última atualização é de 1999, mas foi publicada oficialmente em 2000, com o nome SQL99 (ou SQL3).

# A Origem

A definição de padrões para os comandos, os parâmetros e a sintaxe do SQL evita divergências entre as atualizações lançadas que devem respeitar as diretrizes estabelecidas pelo ANSI e ISO.

Mas os produtores de softwares lançam versões próprias desta linguagem, adicionando ao SQL novos comandos e funcionalidades.

Essas versões (denominadas de dialetos) que além dos comandos básicos do SQL (cerca de 50 instruções), trazem outros que em alguns casos fogem do propósito inicial do SQL, que é uma linguagem para a manipulação de dados em bancos de dados.

# Comandos SQL

Os comandos do SQL são classificados em três grupos, de acordo com suas principais funções:

- ✓ **DDL** (*Definition Data Language*) – Linguagem de Definição de Dados;
- ✓ **DML** (*Data Manipulation Language*) – Linguagem de Manipulação de Dados;
- ✓ **DCL** (*Data Control Language*) Linguagem de Controle de Dados → categoria não adotada oficialmente.

# Comandos SQL

**DDL:** são todos aqueles comandos usados para criar e alterar tabelas que compõem o banco de dados, ou seja, os comandos que definem a estrutura dos dados;

**DML:** pertencem a este grupo todos os comandos usados para extrair informações das tabelas, ou seja, para manipular os dados existentes;

**DCL:** conjunto de comandos usado em sistemas multiusuário para definir os privilégios de acesso aos dados a cada usuário. Os comandos de controle de acesso aos dados são usados para implementar segurança e privacidade em bancos de dados.

# Comandos SQL - DDL

Para criar de um banco de dados:

```
CREATE DATABASE nome_do_banco;
```

Exemplo:

```
mysql> CREATE DATABASE Teste; [Enter]
```

```
Query OK, 1 row affected (0.08 sec)
```

```
mysql> CREATE DATABASE IF NOT EXISTS Teste;
```

Para verificar a existência do BD recém-criado, e de todos os outros criados anteriormente (mostrar bancos de dados):

```
SHOW DATABASES;
```

# Comandos SQL - DDL

Para alterar um banco de dados:

```
ALTER DATABASE nome_do_banco;
```

Exemplo:

```
ALTER DATABASE database_name  
{ <add_or_modify_files> |  
<add_or_modify_filegroups> };
```

Modifica os arquivos e grupos de arquivos associados ao banco de dados no SQL Server. Adiciona ou remove arquivos e grupos de arquivos de um BD, altera os atributos de um BD ou seus arquivos e grupos de arquivos.

# Comandos SQL - DDL

Para excluir um banco de dados:

```
DROP DATABASE nome_do_banco;
```

Exemplos:

```
mysql> DROP DATABASE Teste; [Enter]  
Query OK, 1 row affected (0.05 sec)
```

```
mysql> DROP DATABASE Teste; [Enter]  
ERROR 1008 (HY000): Can't drop database 'Teste';  
database doesn't exist
```

```
mysql> DROP DATABASE IF EXISTS Teste; [Enter]
```

# Comandos SQL - DDL

Para ativar uma Banco de dados.

Sintaxe:

```
USE nome_do_banco;
```

Exemplo:

```
mysql> USE Teste; [Enter]  
Database changed
```

# Comandos SQL - DDL

Para criar de uma tabela no banco de dados.

Sintaxe:

```
CREATE TABLE nome_da_tabela;
```

```
CREATE TABLE [IF NOT EXISTS] nome_da_tabela;
```

Exemplo:

```
mysql> CREATE TABLE cadastro  
(  
  nome CHAR (15),  
  sobrenome CHAR (20)  
); [Enter]
```

# Comandos SQL - DDL

Existem vários tipos possíveis de dados no SQL, os mais comuns:

**INTEGER:** Para inteiros de tamanho normais (32 bits);

**SMALLINT:** Para inteiros (16 bits);

**DATE / TIME:** Para data e hora;

**TIMESTAMP:** Data hora e atribuídos automaticamente;

**CHAR** e **VARCHAR:** Para caracteres até no máximo 255 de tamanho (fixo e variável respectivamente);

**TEXT** ou **LONGTEXT:** Para textos longos;

**DECIMAL:** Para valor numérico fracionário;

**DOUBLE PRECISION:** Para fracionário com dupla precisão;

**FLOAT:** Para numérico de ponto flutuante com 7 dígitos.

# Comandos SQL - DDL

Para criar de uma tabela no banco de dados.

Sintaxe:

```
CREATE [GLOBAL TEMPORARY | LOCAL TEMPORARY]
TABLE nome_da_tabela
[ON COMMIT {PRESERVE ROWS | DELETE ROWS}]
(nome_da_coluna tipo_de_dados especificações,
[nome_da_coluna2, tipo_de_dados2 especificações2,...]) |
[LIKE nome_da_tabela] | [vínculo_tabela],...n ]
```

```
CREATE TABLE nome_da_tabela
(nome_coluna1 tipo_de_dados especificações, [nome_coluna2
tipo_de_dados especificações,...])
```

# Comandos SQL - DDL

## Sintaxe:

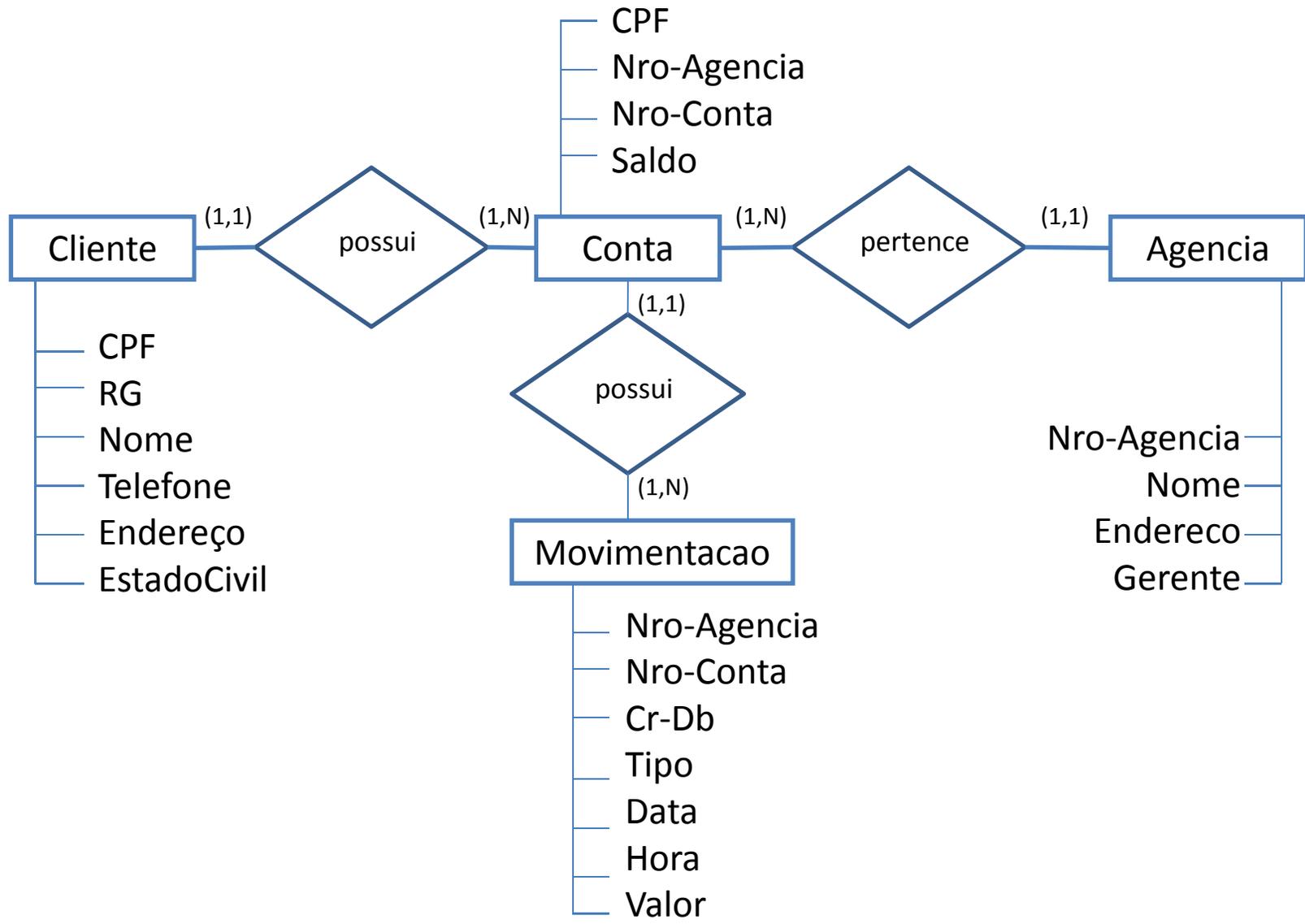
```
CREATE TABLE < nome_tabela > (  
nome_atributo1 < tipo > [ NOT NULL ],  
nome_atributo2 < tipo > [ NOT NULL ],  
nome_atributoN < tipo > [ NOT NULL ]  
PRIMARY KEY(nome_atributo)  
);
```

## Exemplo:

```
CREATE table alunos(  
codigo int NOT NULL AUTO_INCREMENT,  
nome VARCHAR(20) NOT NULL ,  
telefone CHAR(8) NOT NULL,  
PRIMARY KEY(codigo)  
);
```

# Exercício 1

Considerando o Modelo Conceitual ao lado faça: a) Criar o BD; b) Criar as tabelas.



# Comandos SQL - DDL

SHOW - Para exibir a lista de tabelas do banco de dados que está usando atualmente.

Sintaxe:

**SHOW TABLES;**

# Comandos SQL - DDL

ALTER - Este comando permite inserir/eliminar atributos nas tabelas já existentes.

Sintaxe:

```
ALTER TABLE < nome_tabela > ADD / DROP (
nome_atributo1 < tipo > [ NOT NULL ],
nome_atributoN < tipo > [ NOT NULL ] ) ;
```

Exemplo:

```
ALTER TABLE alunos ADD (
filiacao VARCHAR(30) NOT NULL ,
endereco CHAR(40) NOT NULL );
```

# Comandos SQL - DDL

DROP - elimina a definição da tabela, seus dados e referências existente.

Sintaxe:

```
DROP TABLE < nome_tabela > ;
```

Exemplo:

```
DROP TABLE alunos;
```

```
DROP TABLE estudantes;
```

# Comandos SQL - DDL

CREATE INDEX – Comando que cria índices secundários em uma tabela.

Sintaxe:

```
CREATE UNIQUE INDEX index_name  
ON table_name ( column1, column2,...);
```

Exemplo:

```
CREATE INDEX idx_clientes_codigo  
ON CLIENTES(Codigo);
```

# Comandos SQL - DDL

Pode-se criar o índice junto com a tabela ou posteriormente.

Exemplo:

```
mysql>CREATE TABLE CLIENTES  
(Codigo INT,  
Nome VARCHAR(50),  
INDEX (Codigo) ); [Enter]
```

Ou

```
mysql>CREATE TABLE CLIENTES  
(Codigo INT,  
Nome VARCHAR(50) ); [Enter]
```

```
mysql>CREATE INDEX idx_CLIENTES_CODIGO ON  
CLIENTES(Codigo); [Enter]
```

# Comandos SQL - DDL

CREATE VIEW – Cria uma visão com base em uma ou mais tabelas do BD.

Sintaxe:

```
CREATE  
  [ALGORITHM = {MERGE | TEMPTABLE | UNDEFINED}]  
VIEW [database_name].[view_name]  
AS [SELECT statement]
```

Exemplo:

```
Mysql>CREATE VIEW teste AS SELECT nome, endereco, cep  
FROM tabela1
```

# Comandos SQL - DDL

## **Exercício:**

Em um Cinema tem-se a necessidade de registrar os Filmes e em que Sala eles serão apresentados. Considere as informações.: Filmes (Título, Gênero, Classificação, Duração, Ator principal, Atriz principal), Sala (Número, Quantidade de assentos). Fazer o DER e depois criar o BD e as tabelas. Neste Banco de dados devem ser definidas as tabelas com suas chaves primárias e índices secundários. Há também a necessidade de criar uma visão para apresentar ao cliente e facilitar a sua escolha.



# Banco de Dados

Prof. Walteno Martins Parreira Jr

[www.waltenomartins.com.br](http://www.waltenomartins.com.br)

[waltenomartins@yahoo.com](mailto:waltenomartins@yahoo.com)

2015