



Linguagem SQL

Parte 1

Fundamentos de banco de dados 2

Bibliografia

- ELMASRI, Ramez.; NAVATHE, Shamkant B. *Sistema de banco de dados*. 3. ed. São Paulo: Pearson do Brasil, 2011.
- KORTH, Henry F.; SILBERSCHATZ, Abraham. *Sistema de banco de dados*. São Paulo: Pearson Makron Books, 2008.
- LIGHTSTONE, Sam; TEOREY, Toby; NADEAU, Tom. *Projeto e modelagem de bancos de dados*. Rio de Janeiro: Campus, 2006.

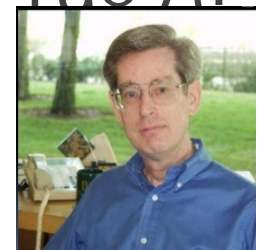


Introdução

- SQL
 - Usuário apenas especifica qual o resultado desejado através de uma consulta declarativa, deixando a otimização com o SGBD

Histórico

- SEQUEL - ***S**tructured **E**nglish **QUE**ry **L**anguage*
 - Projetada e implementada no Centro de Pesquisa da IBM como interface de um banco de dados relacional experimental chamado System R (no início dos anos 70);
 - ANSI (*American National Standards Institute*) e a ISO (*International Standards Organization*) juntaram esforços de pesquisa, originando a versão SQL96, ou SQL1 (em 1987).
 - SQL2, aprovada em 1992 (SQL92);
 - SQL3 aprovada em 1999 e SQL:2003 (xml) depois.



Donald D. Chamberlin

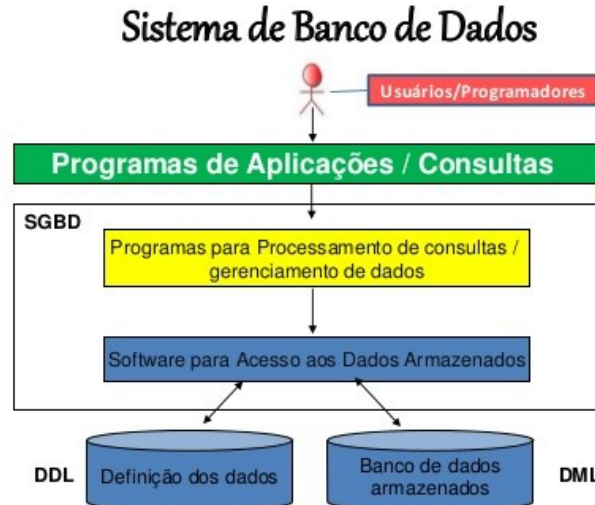
SQL – *Structured Query Language*

- É uma linguagem padrão de definição e manipulação para Banco de Dados Relacional.
 - Para armazenar e recuperar dados de/para um SGBD:
 - ORACLE, MySQL, Postgre, Sybase entre outros.
- **Definição:** Conjunto de comandos para definir o esquema do banco de dados, como tabelas, chaves, regras e restrições de integridade.
 - Conhecida como **DDL** (Data Definition Language).
- **Manipulação:** Conjunto de comandos para manipulação dos dados do esquema, como inserir, apagar e alterar linhas em tabelas.
 - Conhecida como **DML** (Data Manipulation Language).
- **Consultas:** Comando para manipulação de dados para consulta das tuplas existentes no esquema.
 - Conhecida como **DQL** (Data Query Language)

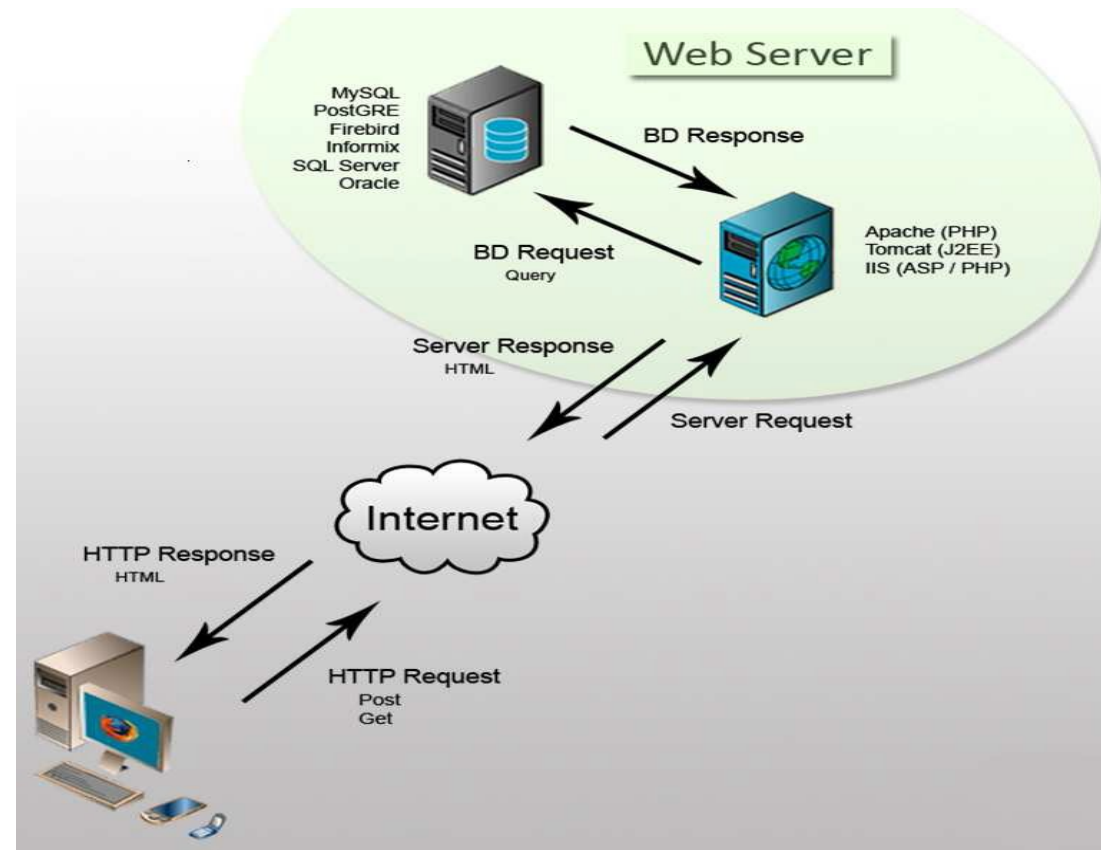
Obs.: A SQL é um padrão ANSI (American National Standard Institute – www.ansi.org) onde estão registrados a maioria dos padrões de mercado de diversas tecnologias.

Linguagem SQL

- Todo o SGBD relacional apresenta o SQL na sua forma padrão adicionando alguns recursos específicos do próprio SGBD, como por exemplo domínios de atributos.
 - A linguagem SQL necessita de uma linguagem hospedeira (Java, PHP, C, Python, etc)
- Os SGBD's apresentam além do SQL uma linguagem de programação proprietária que tem com chamados de **Stored Procedures**

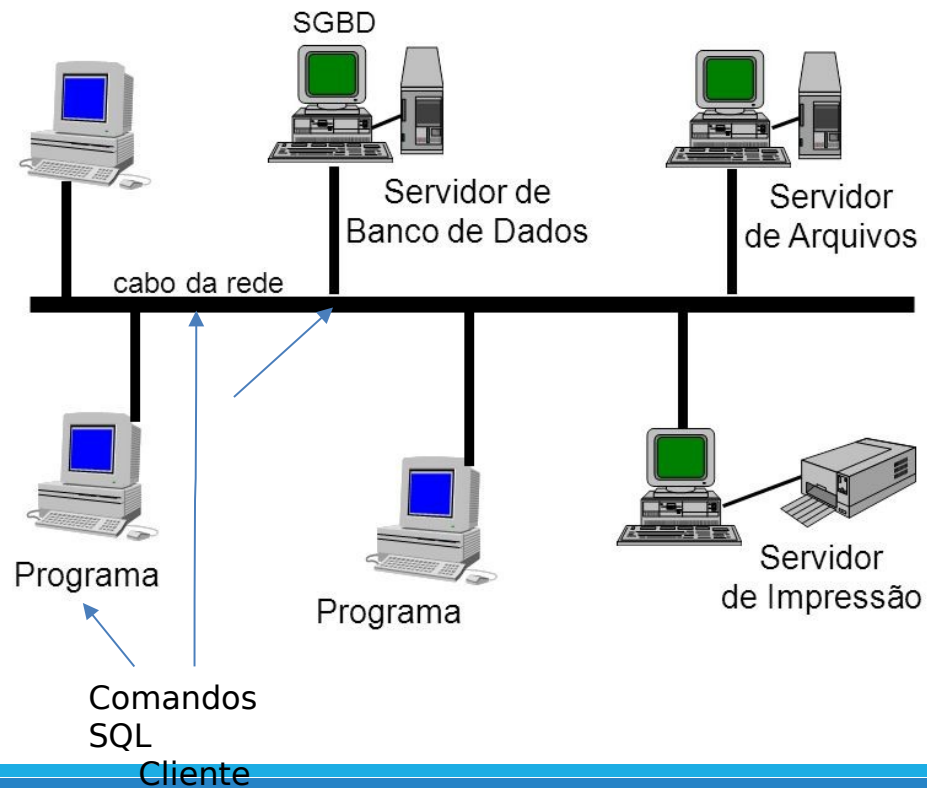


Arquitetura WEB (aplicação)



Arquitetura Cliente/Servidor

Arquitetura Cliente/Servidor com Servidor de Bancos de Dados



SQL - DDL

- Será verificado alguns comandos básicos de DDL para:
 - Criação/Remoção de um banco de dados;
 - Criação/Remoção de Objetos do Banco de dados;
 - (Tabelas/Visões/Índices)
 - DDL (Data Definition Language): Linguagem de Definição dos dados
 - Permite a definição da estrutura do banco de dados
 - CREATE
 - **CREATE** DATABASE bancoexemplo
 - ALTER
 - **ALTER** TABLE teste ADD nome VARCHAR(30)
 - DROP
 - **DROP** DATABASE bancoteste

SQL - DML

- DML (Data Manipulation Language): Linguagem de Manipulação dos dados e Recuperação de dados (DQL)
 - Possibilita a recuperação dos dados do banco
 - SELECT
 - **SELECT** * FROM tabelaexemplo
 - Possibilita atualização dos dados (tupla)
 - Permite alterações e exclusões de dados já existentes, bem como a adição de novos dados
 - INSERT
 - **INSERT** INTO tabelatestes VALUES (1, 'Fulano', 2000)
 - UPDATE
 - **UPDATE** tabelatestes SET nome = 'Cilano' WHERE código = 1
 - DELETE
 - **DELETE** FROM tabelatestes WHERE código=1

SQL – Outros Comandos

- Controle de acesso (DCL)

- Fornece meios para determinar o controle de acesso de usuários aos dados do banco
 - GRANT (Fornecer acesso)
 - REVOKE (Retirar acesso)

- Compartilhamento de dados (transações) (TC)

- Coordena o acesso concorrente de usuários, assegurando que um não interfira no trabalho do outro
 - COMMIT (Confirma ação)
 - ROLLBACK (Voltar/desfazer uma ação em ponto de referência)
 - SAVEPOINT (Ponto de referência criado)

- Integridade de dados

- Permite definir restrições de integridade aos dados do banco
 - CONSTRAINT (Criação de chaves: Primária e Estrangeira)

Vantagens

- Independência de fabricante;
- Portabilidade entre computadores;
- Redução dos custos com treinamento;
- Inglês estruturado de alto nível;
- Múltiplas visões dos dados.
- Declarações SQL podem ser embutidas em praticamente todas as linguagens de programação modernas
 - Java, C#, Python, etc.
 - Driver de Conexão (ODBC)

• CRIAR BANDO DE DADOS

- Banco de dados também é conhecido pelo nome: **Schema**
- O sistema do MySQL pode suportar vários bancos de dados diferentes.
- Geralmente será criado um banco de dados para cada aplicação.
- Para cria um banco de dados no MySQL será utilizado a palavra-chave **CREATE DATABASE**.
- Sintaxe

```
CREATE DATABASE nome_do_banco_de_dados;
```

```
CREATE SCHEMA nome_do_banco_de_dados;
```

• NOMENCLATURA

- Pode inserir no comando letras maiúsculas, mas o sistema interpreta e cria o banco com letras minúsculas.

Válido	Inválido
nome_banco	nome banco
3nome_banco	nome-banco
banco_pái	nome%banco
NomeBanco	

- Como boas práticas é preferível definir o nome todo em caixa baixa, sem acentos e se for necessário ter palavras compostas, separá-las com underline.

```
CREATE DATABASE nome_do_banco_de_dados;
```

• TIPOS DE DADOS SQL

- Números Inteiros

} N = comprimento máximo de exibição

Tipo	Faixa	Descrição
TINYINT(N)	-128...127 ou 0...255	Inteiro muito pequeno – 1 byte
BIT		Sinônimo de TINYINT
BOOL		Sinônimo de TINYINT
SMALLINT(N)	-32768... 32767 ou 0... 65535	Inteiro pequeno – 2 bytes
MEDIUMINT(N)	-8388608...8388607 ou 0...16777215	Inteiro de tamanho médio – 3 bytes
INT(N)	-2147483648... 2147483647 ou 0... 4294967295	Inteiro regular – 4 bytes
INTEGER(N)		Sinônimo de INT
BIGINT(N)	-9223372036854775808... 9223372036854775807 ou 0... 18446744073709551615	Inteiro grande – 8 bytes

• TIPOS DE DADOS SQL

- Números Flutuantes

Tipo	Faixa	Descrição
FLOAT		Utilizado para números de ponto flutuante duplo ou simples

• TIPOS DE DADOS SQL

- Strings

Tipo	Faixa	Descrição
VARCHAR(N)	1 a 255 caracteres	
TEXT	0 a 65.535 caracteres	Campo TEXT de tamanho normal

• ATRIBUTOS SQL

- Not null
- Auto-increment
- Primary key
- constraint

• CRIAR TABELA

- Após criar um banco de dados, é necessário criar tabelas para atender certa demanda.
- Não é possível criar mais de uma tabela com mesmo nome em um banco de dados.
- Para criar uma tabela em um banco de dados no MySQL será utilizado a palavra-chave **CREATE TABLE**.
- Sintaxe

```
CREATE TABLE nome_tabela  
(  
    <nome_da_coluna1> <tipo_da_coluna1> [<atributos_da_coluna1>],  
    ...  
    <nome_da_coluna> <tipo_da_coluna> [<atributos_da_coluna>]  
);
```

• APRENDA FAZENDO

- Crie a tabela carro no banco de dados banco_1

```
CREATE TABLE carro
(
  id_carro int not null auto_increment,
  modelo varchar(50),
  cor varchar(15),
  ano smallint(4),
  primary key (id_carro)
)
```

Convenção para Nomes

- Objetos: Tabelas, Views, Increments, Índices.
 - 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

SQL - DDL

- Instruções para definição do esquema da base de dados:

- CREATE TABLE

- Cria uma nova tabela na base de dados, especificando nome, atributos e restrições

- ALTER TABLE

- Altera definições de uma tabela

- DROP TABLE

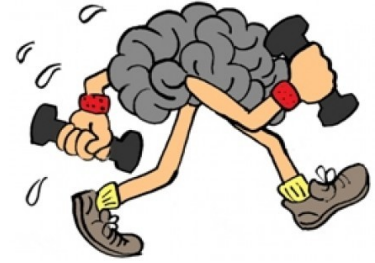
- Remove uma tabela, quando suas definições não são mais necessárias

- Criação de Banco de Dados:

- CREATE DATABASE

- Cria um novo banco de dados
 - Sintaxe pode mudar a depender do SGBD
 - CREATE DATABASE <nomedobanco>;
 - CREATE DATABASE BancoTeste;

Atividade Prática!



- Comandos – Referência MySQL
- Command Line Client do MySQL
 - Teste os comandos:
 - create database teste;
 - show databases;
 - drop database teste;
 - show databases;
 - create database locadora;
 - use locadora;
 - exit;

```
MySQL 5.6 Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.6.25-log MySQL Community Server <GPL>

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| agendamento |
| bancoweb |
| candidatos |
| db |
| mysql |
| ola |
| performance_schema |
| sakila |
| test |
| testandoweb |
| teste |
| venda |
| world |
+-----+
14 rows in set (0.02 sec)

mysql>
```

Criação de Tabelas

- CREATE TABLE

- Colunas são especificadas primeiro, sob a forma:

- <nomeCol> <domínio> <restrição>

- Depois Chaves, integridade referencial e restrições de integridade

- CREATE TABLE <nome_da_tabela>

- (<nome_da_col1> <tipo_da_col1> **NOT NULL**,
<nome_da_col2> <tipo_da_col2> **NOT NULL**,

- ...

- PRIMARY KEY** <lista_de_nomes_de_col>,

- FOREIGN KEY** <nomes_de_col>

- REFERENCES** <nome_tab_ref>(<nome_da_col_ref>)

-);

Criação de Tabelas

- Exemplo:
- ```
CREATE TABLE Fornecedor
(Fcod INTEGER NOT NULL,
 Fnome VARCHAR(20) NOT NULL,
 Status INTEGER,
 Cidade VARCHAR(20)
);
```

| Fcod | Fnome | Status | Cidade |
|------|-------|--------|--------|
|      |       |        |        |

# Restrições e valores DEFAULT

---

- Restrições:

- NOT NULL

- Restrição aplicadas a colunas cujos valores não podem ser nulos

- Valores Default

- Adiciona-se a cláusula DEFAULT <valor> logo após a restrição:

- CREATE TABLE Empregado( ...

- Sexo CHAR(1) NOT NULL DEFAULT "F", ... );

RESTRIÇÃO

DEFINIÇÃO DE VALOR  
DEFAULT



---

# Linguagem SQL

## Parte 1

---