



WBA0178_v2.0

Banco de Dados Relacional e Big Data





A SQL e os Sistemas de Gestão de Banco de Dados

Bloco 1

Cassio Rodolfo Aveiro da Silva



A importância do DBMS

- Atualmente, a grande dificuldade das empresas é saber lidar com seus bancos de dados cada vez maiores.

Figura 1 - Exemplificação de gestão de banco de dados



Fonte: <https://pxhere.com/pt/photo/1432569>. Acesso em: 12 nov. 2020.

A importância do DBMS

- A solução está no **Data Base Management System** (DBMS, em português, Sistema de Gestão de Banco de Dados):
 - ✓ Software único ou conjunto de softwares.
 - ✓ Criação, armazenamento, organização e acesso aos dados.
 - ✓ Sistemas modulares.
 - ✓ Relação cliente/servidor.

(LAUDON; LAUDON, 2014)

A importância do DBMS

- Os DBMS podem ser desenvolvidos em: PCs; *Workstation* Remotas; Laptops e Notebooks e dispositivos móveis (ELMASRI; NAVATHE, 2009).

Figura 2 - Uso de DBMS por meio de PCs



Fonte: <https://pxhere.com/pt/photo/1437805>. Acesso em: 12 nov. 2020.

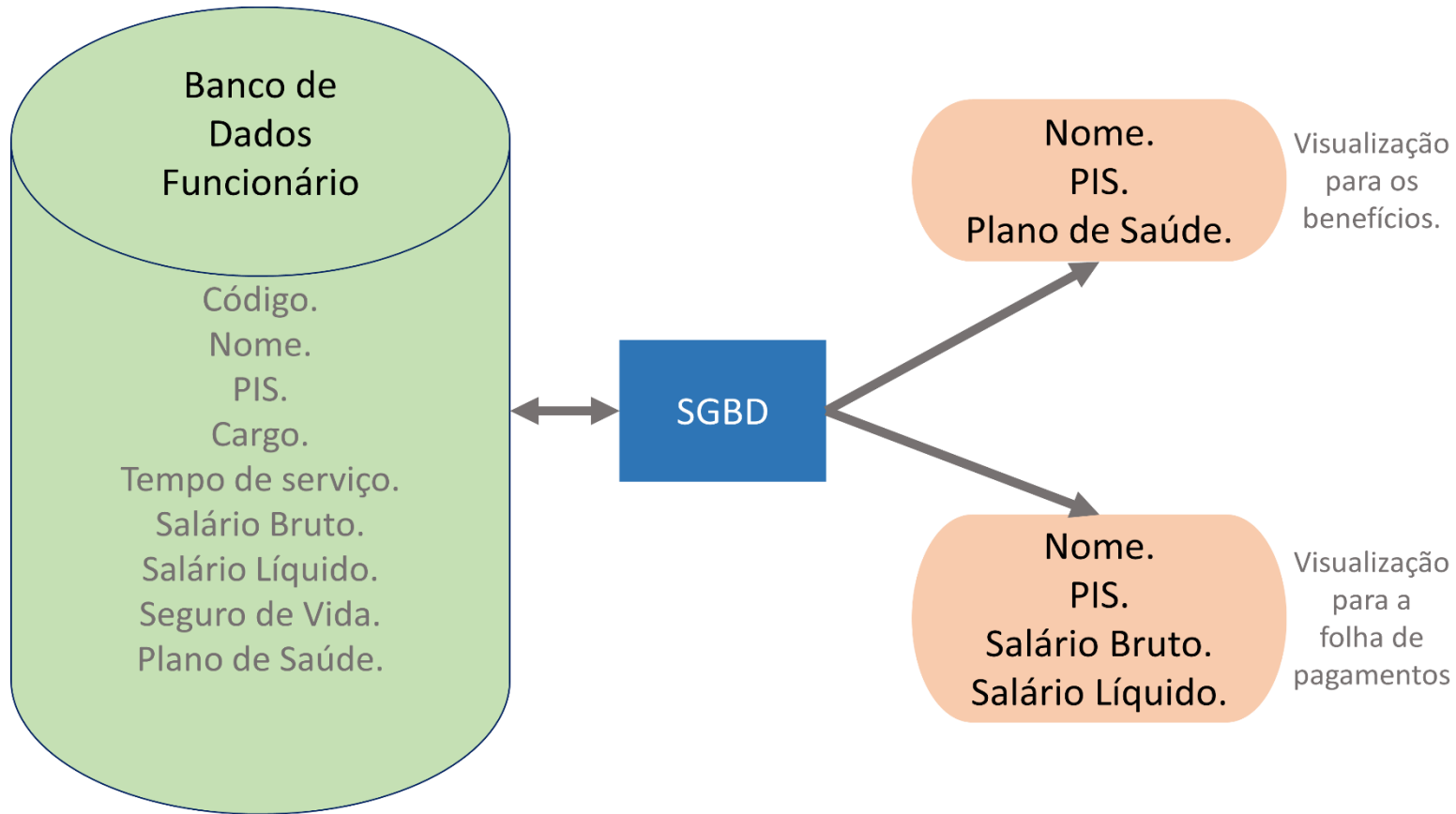
A importância do DBMS

- Voltado à relação **cliente/servidor**, os DBMSs têm a responsabilidade de relacionar duas concepções ou modelos:
 - ✓ **Lógica** dos dados.
 - ✓ **Física** dos dados.
- O objetivo é identificar **onde e como os dados** estão **armazenados** em um banco de dados.

(LAUDON; LAUDON, 2014)

A importância do DBMS

Figura 3 - Ação do DBMS no banco de dados de Recursos Humanos



Fonte: adaptada de Laudon e Laudon (2014).

Estrutura do DBMS

- O DBMS apresenta uma **estrutura** constituída de dois principais **módulos** (MACHADO, 2020):

Módulo Cliente: interfaces de relacionamento com o cliente.

Módulo Servidor: operações técnicas do armazenamento de dados.

Interfaces do DBMS

- Há seis interfaces predominantes na utilização do DBMS:
 - ✓ **Interface Baseada em Menus (MBI - *Menu Based Interface*):**
 - Cliente da web/navegação.
 - Ações selecionadas por botões.
 - Elimina a necessidade de guardar comandos.
 - Pesquisas exploratórias.

(ELMASRI; NAVATHE, 2009)

Interfaces do DBMS

- ✓ **Interface Baseada em Formulários (FBI - *Forms Based Interface*):**
 - Formulários de preenchimento integral.
 - Formulários de preenchimento parcial.
- ✓ **Interface Gráfica para Usuários (GUI):**
 - Manipulação de formulários e menus diagramáticos.

(ELMASRI; NAVATHE, 2009)

Interfaces do DBMS

- ✓ Interface de Linguagem Natural (NLI - *Natural Language Interface*):
 - Consultas em inglês ou em outros idiomas diretamente.
 - Palavras-padrão em diálogo interativo para refinamento.
- ✓ Interface para os Administradores dos Bancos de Dados (DBAI - *Data Base Administrator Interface*):
 - Comando utilizados apenas por DBAs. Com acesso restrito.

(ELMASRI; NAVATHE, 2009)

Interfaces do DBMS

- Por fim a:
 - ✓ Interface para Usuários Parametrizáveis (PUI - *Parameterizable User Interface*):
 - Para operações repetitivas.
 - Redução de comandos.
 - Redução de teclas e cliques.

(ELMASRI; NAVATHE, 2009)



A SQL e os Sistemas de Gestão de Banco de Dados

Bloco 2

Cassio Rodolfo Aveiro da Silva



DBMS relacional

- Para **Banco de Dados Relacionais (BDR)**, os sistemas de gestão de banco de dados (DBMS) devem trabalhar **de acordo com a estrutura** deste tipo de banco:
 - ✓ Combinações de tabelas.
 - ✓ Utilização de chaves estrangeiras.

(LAUDON; LAUDON, 2014)

DBMS Relacional

- O DBMS relacional utiliza três funções essenciais:
 - I. Select:* seleção de Tuplas.
 - II. Project:* escolha dos principais atributos para uma nova tabela.
 - III. Join:* combinação de tabelas para mais informações.

(LAUDON; LAUDON, 2014)

DBMS Relacional

- Além das funções essenciais, os DBMS devem disponibilizar **recursos** para:
 - I. Definir, manipular e organizar dados.
 - II. Criação de um dicionário de dados.
- Especificação da **estrutura do banco de dados**.
- Definições **registradas em arquivos** (registros).

(LAUDON; LAUDON, 2014)

DBMS e sua composição

- Todos os sistemas de gestão de banco de dados são um conjunto de **componentes essenciais**:
 - ✓ Sistema operacional.
 - ✓ Gerenciador de armazenamento e *buffers*.
 - ✓ Processador *runtime*.
 - ✓ Compiladores de consulta.
 - ✓ Computadores cliente e servidor.

(MACHADO, 2020)

DBMS e sua composição

- E também de utilitários:
 - ✓ *Loading.*
 - ✓ *Backup.*
 - ✓ *File Reorganization.*
 - ✓ *Performance Monitoration.*

(MACHADO, 2020)



A SQL e os Sistemas de Gestão de Banco de Dados

Bloco 3

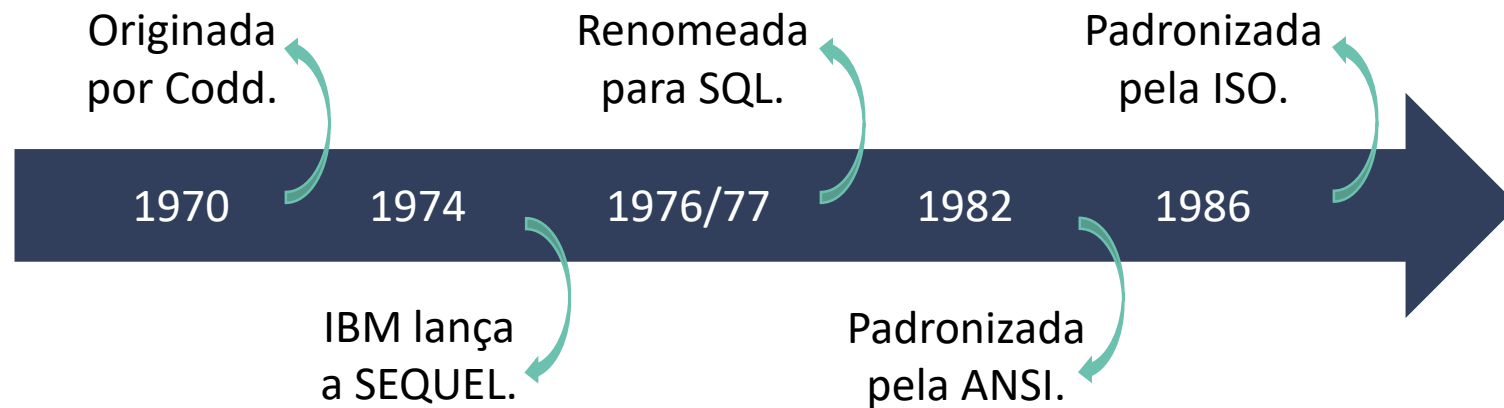
Cassio Rodolfo Aveiro da Silva



A *Structure Query Language* (SQL)

- O sucesso dos Bancos de Dados Relacionais e de seus sistemas de gestão é a ***Structure Query Language* (SQL - Linguagem Estruturada de Consulta)** (COSTA, 2006; MACHADO, 2020):

Figura 4 - Histórico da *Structure Query Language*



Fonte: elaborada pelo autor.

A Structure Query Language (SQL)

- Voltada especialmente aos **Banco de Dados Relacionais**.
- Seus objetivos gerais são:
 - ✓ **Segurança** dos dados.
 - ✓ **Controle** dos dados.
 - ✓ **Integridade** dos dados.

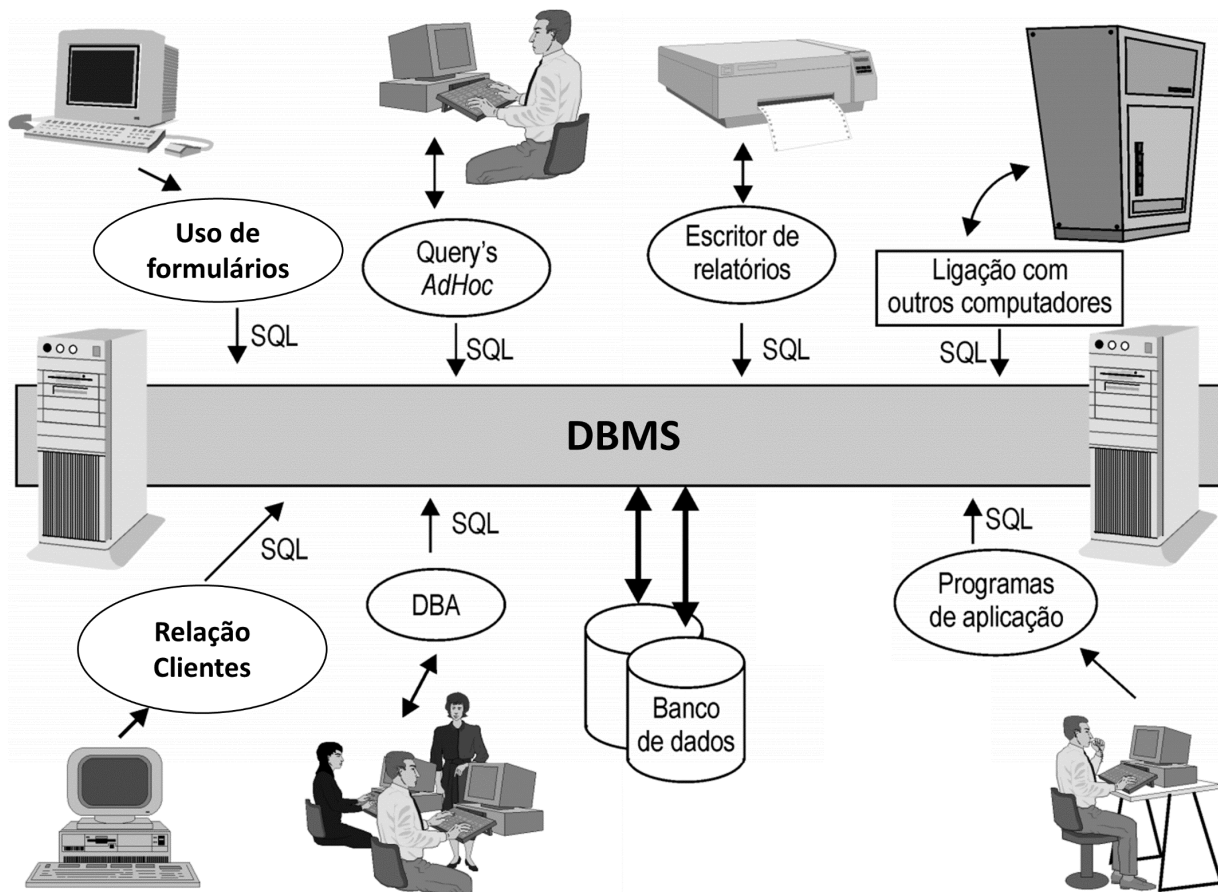
(LAUDON; LAUDON, 2014)

A *Structure Query Language* (SQL)

- Pode ser aplicada por meio de **linguagens hospedeiras** (C, C++, Fortran etc.).
- Elimina a preocupação de programadores e usuários em:
 - ✓ **Converter** dados.
 - ✓ **Migrar** dados de um DBMS para outro.
 - ✓ Realizar **treinamentos** sobre linguagem de **programação específica**.

A Structure Query Language (SQL)

Figura 5 - Multifaces de utilização da SQL em DBMS



Fonte: adaptada de Machado (2020).

A Structure Query Language (SQL)

- Pode manipular diferentes classes entre as funções de um DBMS (MACHADO, 2020).

Figura 6 - Funções de um DBMS alteradas e controladas pela SQL



Fonte: adaptada de Machado (2020).

A *Structure Query Language* (SQL)

- Sub-linguagens da SQL (ELMASRI; NAVATHE, 2009):

DDL - *Data Definition Language* - definição de dados.

DML - *Data Manipulation Language* - manipulação de dados.

A *Structure Query Language* (SQL)

- A DML pode ser subdividida em:
 - ✓ **Alto Nível:** capacidade de entender o usuário. Pode ou não ser utilizada uma linguagem de programação geral. Processam um conjunto de dados.
 - ✓ **Baixo Nível:** exige que seus comandos estejam embutidos em uma linguagem de programação geral. Comandos de *loop*. Processam um dado por vez.

(ELMASRI; NAVATHE, 2009)

A *Structure Query Language* (SQL)

- Comandos básicos DDL (ELMASRI; NAVATHE, 2009):

```
CREATE TABLE <nome_tabela>
```

```
SHOW DATABASES;
```

```
DROP DATABASE <nome_BD>;
```

```
AFTER TABLE <nome_tabela> ADD  
<nome_campo> <tipo_campo>(tamanho);
```

```
TRUNCATE <nome_tabela>;
```

```
AFTER TABLE <nome_tabela> RENAME  
<novo_nome_tabela>;
```

A Structure Query Language (SQL)

- Comandos básicos DML (ELMASRI; NAVATHE, 2009):

```
INSERT <nome_tabela> VALUES  
(<Valor_campo_1>,<Valor_campo_2>,...);
```

```
SELECT*FROM <nome_tabela>;
```

```
UPDATE <nome_tabela> SET <coluna>  
<operador> <novo_valor>;
```

```
DELETE FROM <nome_tabela> WHERE <condição>;
```


A *Structure Query Language* (SQL)

- Benefícios extras:
 - ✓ Compartilhamento de dados entre usuários concorrentes.
 - ✓ Segurança e sigilos de informações importante.
 - ✓ Otimização da integridade contra falhas.

(LAUDON; LAUDON, 2014)



Teoria em Prática

Bloco 4

Cassio Rodolfo Aveiro da Silva



Refleta sobre a seguinte situação

Sua empresa, do ramo de venda de produtos importados, possui uma alta demanda de pedidos, o que exige um relacionamento estrito com fornecedores para garantir o atendimento aos clientes. Atualmente, a empresa conta com sistemas de gestão da cadeia de suprimento e de relacionamento com o cliente, sistemas de informação que já seriam suficientes para garantir o sucesso dos processos de negócios. Porém, o banco de dados que alimenta tais sistemas não possui uma gestão adequada, causando atrasos e informações desconexas. Você recebe a missão de resolver esse problema. Nesta situação, como você poderia utilizar um Sistema de Gestão de Banco de Dados estruturado em SQL? Quais utilitários e interfaces recomendadas para esse sistema?

Norte para a resolução...

- Claramente, está havendo uma utilização não eficiente dos sistemas de informação em função da má gestão do Banco de Dados (BD) da empresa. Isso ocorre porque esse BD dever ser relacional, permitindo a integração entre os sistemas.
- Assim, um Sistema de Gestão de Banco de Dados deve ser utilizado como um gerenciador de um BD Relacional estruturado em SQL, com foco nos módulos principais: clientes e servidor. Pois, é esse o ponto crítico do problema, transformar os dados físicos do BD em visualizações claras e objetivas aos interessados (pessoal do Marketing, de vendas, de estoque...).
- Pensando nisso, o DBMS pode ser aplicado com o uso de utilitários, principalmente o utilitário *Loading* para a localização e carregamento dos arquivos corretos e do *Performance Monitoration*, visando auxiliar as tomadas de decisões nos negócios.



Dica do Professor

Bloco 5

Cassio Rodolfo Aveiro da Silva



Orientações para o bloco 5

Filme: *Curso SQL Completo 2019 [Iniciantes] + Desafios + Muita Prática*. Canal: Dev Aprender, disponível no Youtube.

INTRODUÇÃO AO SGBD. Canal: Kleber Pereira de Almeida, disponível no Youtube.

Trabalho Científico:

MACIEL, P. R. M.; TEDESCO, P. C. A. R. **DBSitter-AS**: um framework orientado a agentes para construção de componentes de gerenciamento autônomo para SGBD. 2007. Dissertação (Mestrado) - Universidade Federal de Pernambuco, Recife, 2007.

Referências

COSTA, R. L. **SQL: guia prático**. 2. ed. Rio de Janeiro: Brasport, 2006.

DATE, C. J. **SQL e teoria relacional**: como escrever códigos SQL precisos. Tradução: Rafael Zanolli. São Paulo: Novatec, 2015.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. Tradução: Marília G. Pinheiro. 4. ed. São Paulo: Pearson, 2009.

LAUDON, K.; LAUDON, J. **Sistemas de informações gerenciais**. Tradução: Célia Taniwaki. 11. ed. São Paulo: Pearson, 2014.

MACHADO, F. N. R. **Projeto e implementação de banco de dados**. 4. ed. São Paulo: Érica, 2020.



Bons estudos!

