



Bancos de Dados I

Prof. Dr. Luis Mariano del Val Cura

Bancos de Dados I

Duas Provas

```
Prova 1: 15/04/ 4 pontos
```

Prova final : 01/ 07

Se média >=4,0 pode fazer PF.

Nota final aprovado
$$>= 6.0$$
 Presença $>= 75\%$

Bancos de Dados I

Bibliografia:

- ELMASRI, R.; NAVATHE, S. B. **Fundamentals of Database Systems**. 5th Ed. Redwood City: Addison-Wesley, 2007.
- KORTH, H. F.; SILBERSCHATZ, M.; SUDARSHAN, S. **Database Systems Concepts**. NewYork: McGraw Hill, 2006.
- GARCIA-MOLINA, H.; ULLMAN, J.; WIDOM, J. **Database Systems**: The Complete Book. 2nd Ed. London: Prentice Hall, 2008.
- HEUSER, C. A. **Projeto de Banco de Dados**. 6a. Ed. Porto Alegre: Bookman, 2009.
- GUIMARÃES, C.C. **Fundamentos de Bancos de Dados:** Modelagem, Projeto e Linguagem SQL. Campinas: Editora da Unicamp, 2003.
- RAMAKRISHNAN, R., GEHRKE, J. **Database Management Systems**. 3rd Ed. New York: McGraw-Hill, 2003.

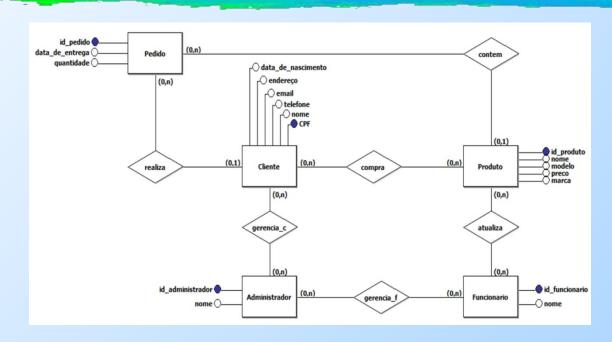
O que veremos neste curso?

- Modelagem conceitual de dados.
 - MER- Modelagem Entidade Relacionamento
- Modelo relacional.
- Álgebra e cálculo relacional.
- Linguagens de gerenciamento e consulta de Bancos de Dados Relacionais. SQL.
 - CRUD : Criação e Inserção (C), Consultas (R),
 Modificação(U) e Remoção(D)
 - Algum SGBD

Ou que veremos em Bancos de Dados II

- Projeto e Normalização de BD.
 - Otimização de Consultas
- Controle de transações e níveis de isolamento. ACID
- Visão real de Modelos de dados não convencionais. (XML)
- Visão geral de Modelos de dados NoSQL.

Modelagem conceitual de dados.



- Modelagem Entidade Relacionamento (MER) dos conceitos de Banco de dados.
- Este modelo conceitual independe do Modelo Lógico de Dados. (ex. Modelo Relacional)
 - Utilizaremos várias ferramentas

Modelo Relacional

- Criado por Edgar F. Codd em 1970.
- A partir dos anos 1980 começou a ser o modelo básico de bancos de dadoscontinua até hoje

Communications of the ACM, Vol. 13, No. 6, June 1970, pp. 377-387.

A Relational Model of Data for Large Shared Data Banks

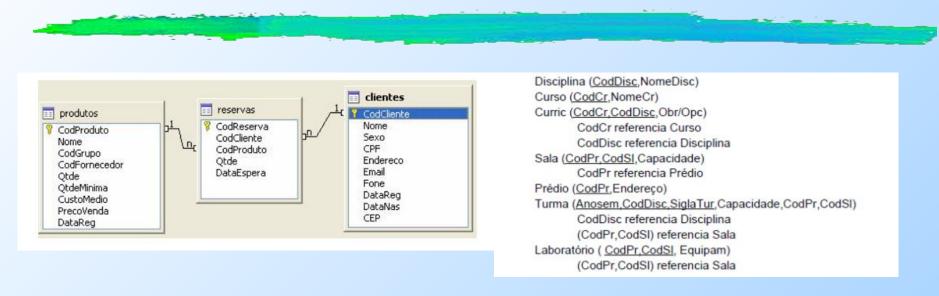
E. F. Codd

Abstract

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as' a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing non inferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on n-ary relations, a normal form for data base relations, and the concept of a universal data sub language are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

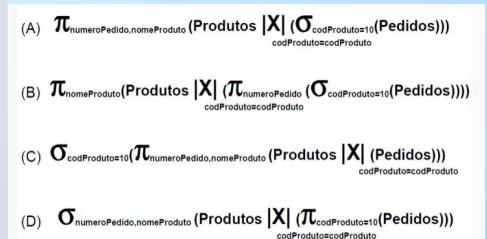
Modelo Relacional



- O modelo Conceitual (MER) é convertido em um Modelo Lógico Relacional.
- Baseado em Relações (Tabelas)
- Vamos ver o conjunto de restrições e de operações CRUD sobre esse modelo : inserção(c), atualização(U) e remoção(D).
- Na parte de consultas (C) veremos em Álgebra Relacional

Álgebra e Cálculo relacional

| OPERAÇÃO | SÍMBOLO | SINTAXE |
|--------------------|--------------|---|
| Projeção | π ("pi") | π sta de campos> (Tabela) |
| Seleção/ Restrição | ("sigma") | σ <condição de="" seleção=""> (Tabela)</condição> |
| União | U | (Tabela 1) ∪ (Tabela 2) |
| Interseção | \cap | (Tabela 1) ∩ (Tabela 2) |
| Diferença | - | (Tabela 1) – (Tabela 2) |
| Produto Cartesiano | X | (Tabela 1) X (Tabela 2) |
| Junção | X | (Tabela 1) X < condição de junção> (Tabela 2) |
| Divisão | ÷ | (Tabela 1) ÷ (Tabela 2) |
| Renomeação | P ("rho") | ρ Nome(Tabela) |
| Atribuição | ← | Variável ← Tabela |



- As consultas para o Modelo Relacional utilizamos um modelo matemático.
- Qualquer consulta pode ser escrita na álgebra

Linguagens de gerenciamento de Bancos de Dados Relacionais. SQL.

```
select lista_atributos
from tabela1 [inner] join tabela2 on
   condição_junção [join tabela3 on ...]
[where condição]
```

Mapeamento para a álgebra relacional

```
select a_1, ..., a_n
from t_1 join t_2
on t_1.x > t_2.x
where c
\pi_{a1, ..., an} \left( \sigma_c \left( t_1 \theta \times t_2 \right) \right)
```

 SQL é uma das implementações (definitiva!!.) da Álgebra Relacional.

Linguagens de gerenciamento de Bancos de Dados Relacionais. SQL.

- SQL Já é a Linguagem de Gerenciamento de BD.
- Veremos a parte de consulta as operações CRUD e outras operações da linguagem.
- Veremos o SGDB MySQL. Vamos a conectar ao server que existe na
- Para o pessoal que pode utilizar Celular
 - Utilizar a ferramenta MySQL Viewer no celular

Nesta aula...

- Introdução a conceitos básicos de bancos de dados
- Abordagem de Sistemas de Arquivos vs Bancos de Dados
- Histórico

Introdução

Bancos de dados

- Coleção de "dados" relacionados.
- Coleção lógica coerente de dados com um significado inerente
- Mini-mundo : universo de discurso (UoD)
- Projetado e construído com um propósito específico.
- MODELO DO MUNDO REAL
 - Amazon : Exemplo de grande BD.

Introdução

Sistemas de Gerenciamento de Bancos de Dados (SGBD)

- Conjunto de programas .
- Permite a criação, manutenção e manipulação de um Banco de Dados.
- Conjunto formado por um banco de dados mais as aplicações que manipulam e controlam estes dados [Elmasri...]
- Software que manipula todos os acessos ao Banco de Dados.[Date...]

Introdução

Transação

Operação que provoca leitura /escrita de dados no banco de dados.

Consulta

Transação de leitura com o objetivo de recuperar dados

Meta-Dados

- Definição da estrutura do banco de dados : tipos, relações e restrições dos dados a serem armazenados.
- Armazenado no SGDB como catálogo ou dicionário de dados.

Programa de Aplicação.

Programa que acessa ao banco de dados enviando consultas ao SGBD

Banco de Dados Compartilhado

 Múltiplos usuários e programas acessando o banco de dados simultaneamente.

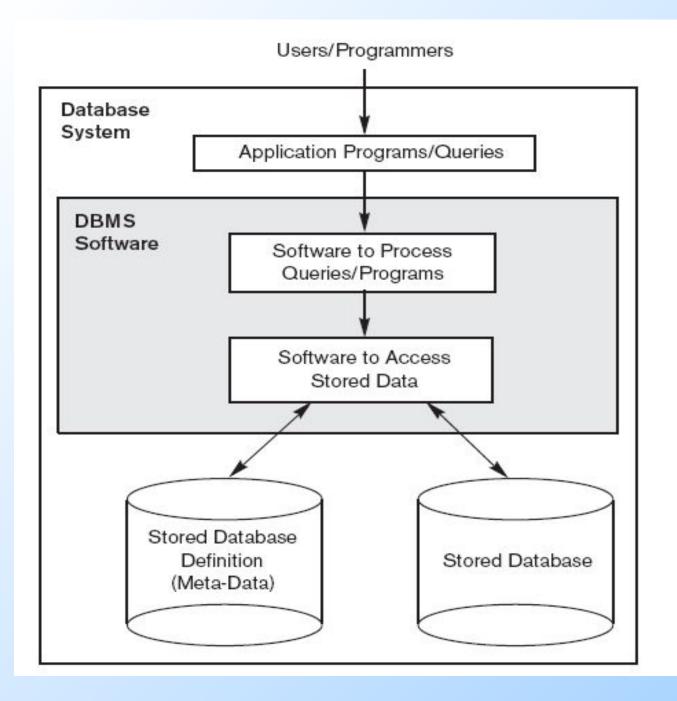


Figure 1.1
A simplified database system environment.

Aplicações de Bancos de Dados

Processamento Transacional em Tempo Real (On-Line Transaction Processing -OLPT)

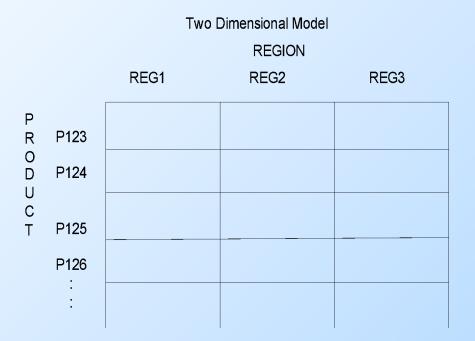
- Aplicações tradicionais de Bancos de Dados.
 - Armazenam informação textual ou numérica.
- Bancos de Dados Multimidia
 - Armazenam imagens, clips de áudio e streams de video.
- Sistemas de informação geográfica (GIS)
 - Armazenam e analisam mapas, imagens de satélite, etc.

Aplicações de Bancos de Dados

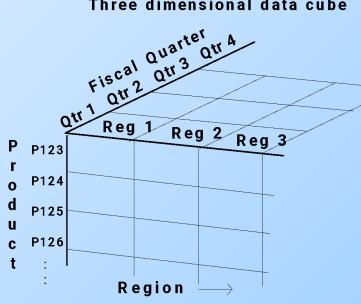
Processamento analítico. (On-Line Analitic Processing - OLAP)

- Extrai um estado temporal do banco de dados.
 Warehouse.
- Analisa informação de negócio de grandes bancos de dados.
 - Modelo de organização e operações diferentes ao modelo transacional. Cubos.
- Sistemas de suporte a decisão. Data warehouses, Data Mining. (BI)

Data Warehouse







Sistemas de arquivos convencionais vs Sistemas de Bancos de Dados

Abordagem com Sistemas de Arquivos Um exemplo (C)

- Aplicação de acesso a dados persistentes utilizando sistema de arquivos.
- Registros de alunos armazenados em um arquivo.
- Deseja-se determinar qual a maior idade de todos os alunos

ABORDAGEM COM SISTEMAS DE ARQUIVOS UM EXEMPLO (C)

Consulta: mostrar a quantidade dos alunos no "banco de dados" com mais de 15 anos

```
typedef {
    char RA[12];
    char nome[30];
    char sexo;
    int idade;
} struct aluno;
FILE *fp;
```



ABORDAGEM COM SISTEMAS DE ARQUIVOS UM EXEMPLO (C)

Consulta: mostrar a quantidade dos alunos no "banco de dados" com mais de 15 anos

```
aluno meuRegistro;
int Idade, maxIdade = 15;

fp = fopen("BD.dat", "wb");
Idade =0;
while (fread(&meuRegistro, sizeof(aluno), 1, fp) != NULL) {
  if (meuRegistro.idade >= maxIdade) {
    Idade++;
  }
printf("Alunos de mais de 15 anos = %d", Idade);
```

Gerenciamento de Dados. **Abordagem com Sistemas de Arquivos**

- Cada usuário define e implementa arquivos específicos para sua aplicação.
- A estrutura dos dados (esquema) é definida e armazenada na aplicação.
- Manipulação do banco (consultas e atualizações) envolve a escrita de programas para processamento do arquivo.
- Compartilhamento de dados : inconsistências, trancas do arquivo.
- Permissões e acesso aos dados: Tudo ou nada.

Abordagem com Bancos de Dados

Um exemplo (Java)

- Aplicação de acesso a dados persistentes utilizando bancos de dados.
- Registros de alunos armazenados em um banco de dados.
- Deseja-se determinar qual a maior idade de todos os alunos

Um exemplo (Java) Abordagem com Bancos de Dados

Gerenciamento de Dados. **Abordagem de Bancos de Dados**

- Um repositório armazena os dados que são definidos uma única vez e são acessados por múltiplos usuários.
- O estrutura dos dados (esquema) é armazenado no banco de dados. O Banco é auto-contido.
- Linguagem declarativa para atualizações e consultas.
 SQL
- Compartilhamento seguro dos dados. Processamento multi-usuário de transações.
- Tolerância a falhas. Controle de transações.
- Suporte a múltiplas visões de dados e de permissões sobre estes.

Banco de Dados auto-contido

- O sistema de banco de dados contem a definição da sua estrutura e suas restrições.
- Meta-dados
- O catálogo ou dicionário pode ser usado por:
 - Programas do SGBD
 - Usuários do Banco de Dados que precisam de informação sobre a estrutura deste.

Compartilhamento seguro dos dados.

Processamento multi-usuário de transações

Controle de Concorrência

- Múltiplos usuários acessando ao Banco de Dados simultaneamente
- Várias transações executado concorrentemente. Escalonamento
- Garante que múltiplos usuários modifiquem os mesmos dados de forma controlada. i.e, que o resultado das modificações seja correto

Propriedade de Isolamento

 Cada transação deve parecer que executa isolada de outras transações.

Exemplo de duas transações concorrentes

Contas (NoConta, NomeCorrentista, Saldo)

Transação 1:

Transferir R\$ 50.00 da conta 094567 para 462364

- 1. read(A)
- 2. A := A 50
- 3. write(A)
- 4. read(B)
- 5. B := B + 50
- 6. **write**(*B*)

Transação 2:

Depositar R\$ 300.00 na conta 094567

- 1. read(A)
- 2. A := A + 300
- 3. **write**(*A*)

Exemplo de duas transações concorrentes Um escalonamento incorreto

Contas (NoConta, NomeCorrentista, Saldo)

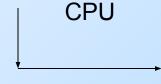
Transação 1:

Transferir R\$ 50.00 da conta 094567 para 462364

Transação 2:

Depositar R\$ 300.00 na conta 094567

- 1. read(A)
- 2. A := A 50



- 1. read(A)
- 2. A := A + 300
- 3. write(A)
- 4. commit

- 3. write(A)
- 4. read(B)
- 5. B := B + 50
- **6.** write(*B*)
- 7. commit

Tolerância a falhas.

- Propriedade de Atomicidade
 - Ou todas as operações de uma transação são executadas ou nenhuma é executada. (Recuperação de dados)
- Propriedade de Durabilidade.
 - Uma vez que uma transação termina o SGBD deve garantir a persistência da mudança realizada pela transação

Exemplo de duas transações concorrentes Um escalonamento correto com conflitos

Contas (NoConta, NomeCorrentista, Saldo)

CPU

Transação 1:

Transferir R\$ 50.00 da conta 094567

para 462364

- 1. read(A)
- 2. A := A 50
- 3. write(A)

- 4. read(B)
- 5. B := B + 50
- 6. **write**(*B*)
- 3. rollback

Transação 2:

Depositar R\$ 300.00 na conta 094567

- 1. read(A)
- 2. A := A + 300
- 3. write(A)
- 4. commit

Garantia de Durabilidade

Garantia de Atomicidade

Propriedades ACID das transações

- Atomicidade
- Consistência
- Isolamento
- Durabilidade

Histórico

Histórico

Década dos 60's

Surgimento dos sistemas de gerência de bancos de dados (SGBDs) pioneiros hierárquicos:

- IMS
- TOTAL

Início da década dos 70's

Padronização do modelo de Bancos de Dados em Redes

CODASYL DBTG Report [1971] e ANSI/SPARC Report [1975]

Início da década dos 70's

1970: O modelo relacional é apresentado por Edgar F. Codd

Communications of the ACM, Vol. 13, No. 6, June 1970, pp. 377-387.

A Relational Model of Data for Large Shared Data Banks

E. F. Codd

Abstract

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as' a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing non inferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on n-ary relations, a normal form for data base relations, and the concept of a universal data sub language are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

Década dos 70's

1072: *B-tree* para indexação de grandes volumes de dados.

1974: Linguagem SQL para BD Relacionais

1976: Modelo Entidade-Relacionamento (Peter Chen)

SGBDs relacionais pioneiros (SGBD-Rs):

- System R [1976]
- Ingres [1976]

Década dos 80's

SGBD-Rs distribuídos (SGBDD-R):

- Extensões da tecnologia relacional para tratar BD Distribuidos.
- Otimização de consultas distribuídas.
- Protocolo 2PC para transações distribuídas

SGBDD-R pioneiros

- System R*
- Ingres Distribuído

Final da Década dos 80's:

- SGBDs Orientados a Objeto (SGBD-OO)
 - Baseiam-se no paradigma de orientação a objeto (OO), como as linguagens de programação OO (LP-OO)

SGDB-OOs pioneiros:

- O2 [1988]
- Exodus [1986]
- ORION [1986]
- Padronização do SQL (1986 1989)
- Maturidade da tecnologia de SGBDs:
 - Vários SGBD-Rs com desempenho aceitável DB2, Ingres, Oracle, Sybase, Informix

Início da Década dos 90's

- O debate OO-RO
 - Projeto de software: Orientado a objetos, Metodologias Ágeis,
 - Projeto de dados: Relacional (Tabelas)
 - IMPEDÂNCIA OBJETO RELACIONAL
 - Orion , O2.
- SGBD-RO (Relacional- Objeto):

Combina características de OO com o modelo relacional POSTGRES [1986] , STARBURST , ORACLE

Atkinson et.al.. **The Object-Oriented Database System Manifesto.** 1989 Stonebraker et. Al. **Third-Generation Database System Manifesto** 1990

Década dos 90 - Maturidade

Maturidade da tecnologia de SGBDs Relacionais:

- Primeiros SGBD comerciais para grande volume de dados.
- SGDBs paralelos / dedutivos / ativos / temporais / ...
- Novas classes de aplicações:
 - Data Mining Data Warehouse
 - Bibliotecas Digitais
 - Vídeo-sob-demanda
 - Animação
 - Hipermídia e Multimídia em geral
 - GIS
 - Meteorologia

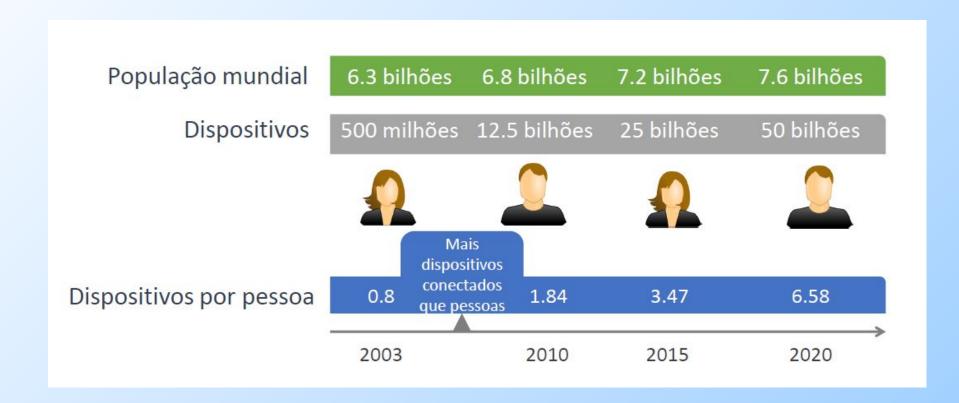
Século 21 : BIG Data + WEB + Distribuição

- Aplicações BIG DATA.
 - Facebook, Amazon, Google, Twiter, Whats.,
- Federações de bancos de dados em larga escala
 - A Web é um grande banco de dados federados.
 - Bilhões de clientes Web acessam milhões de bancos
 - Web semântica, RDF.
- Integração de dados estruturados e semi-estruturados
 - XML = dados organizados hierarquicamente
 - Unificação das tecnologias de banco de dados e da Web

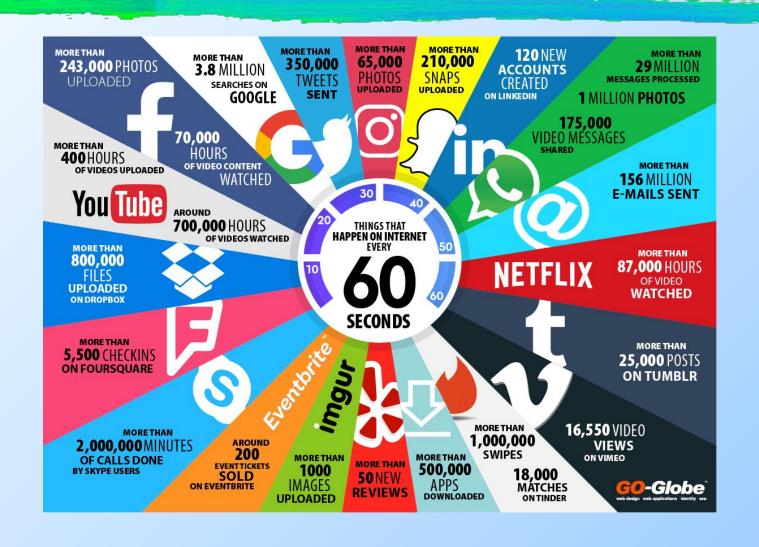
Big Data

- Volume : Enorme volume de dados gerados.
- Velocidade: Velocidade com que os dados são gerados. Capacidade para processar esses dados
- Variedade : Diversidade das fontes, formatos, estruturas dos dados a serem integrados
 (Stonebraker - três V´s)
- Valor : Valor e utilidade dos dados.
- Veracidade : Certeza sobre o dado processado.

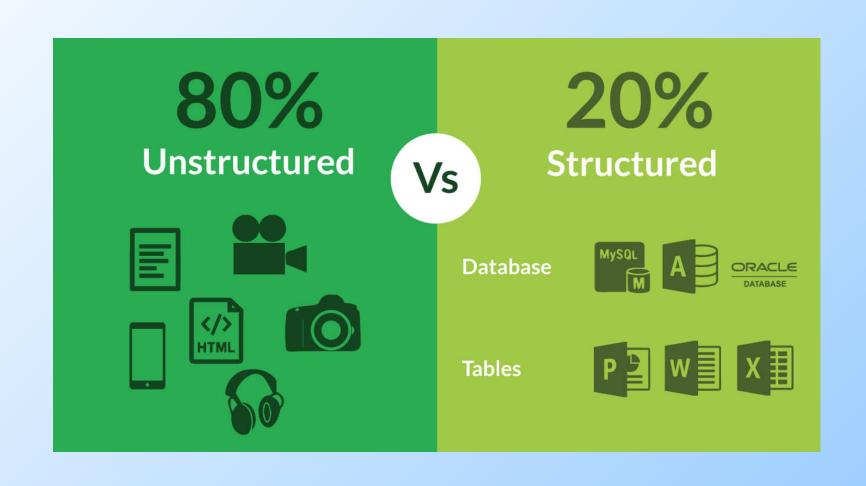
Big Data. Volume



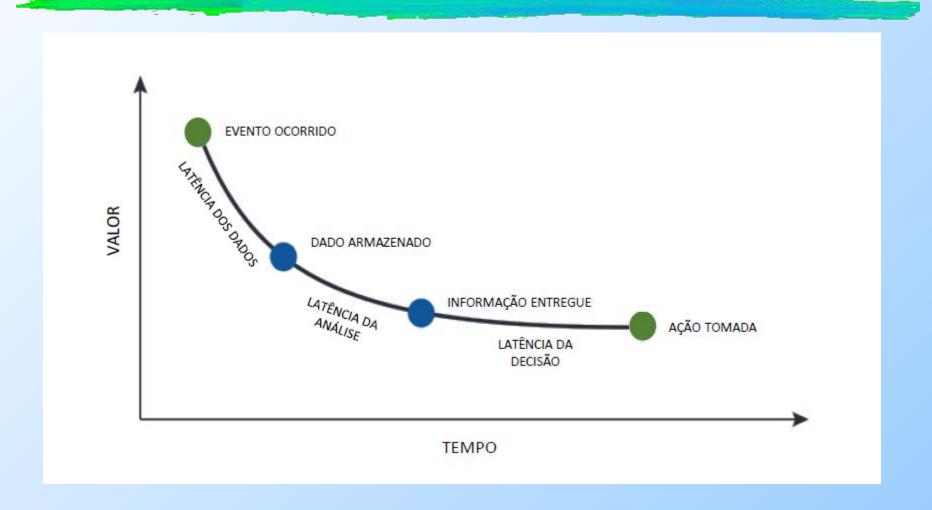
Big Data. Velocidade



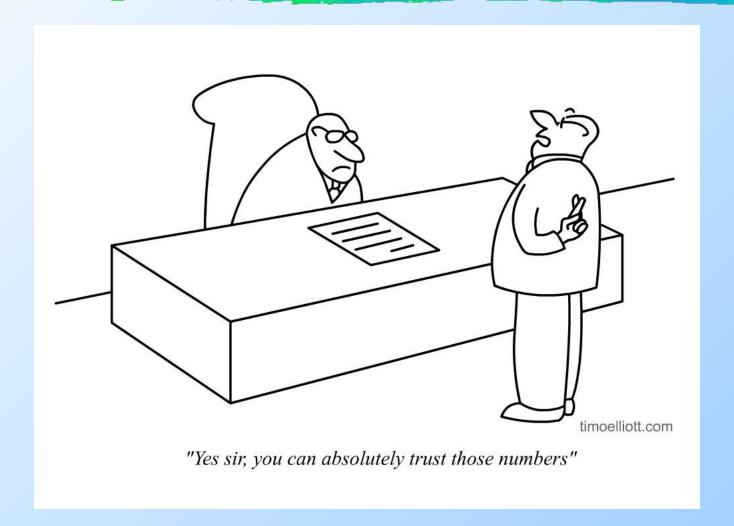
Big Data. Variedade



Big Data. Valor



Big Data. Veracidade



Big Data. Processamento Analítico

BIG ANALYTICS. Ciência de dados.

- Mineração de dados
- Mineração de textos
- Machine Learning
- Deep Learning
- Visualização de dados.
- TOMADA DE DECISÔES

Big Data (Old SQL)

Sistemas de Gerenciamento convencionais (Old SQL)

- Não resolvem as necessidades de desempenho e armazenamento das novas aplicações.(Big Elephants).
- Baseados, fundamentalmente, no modelo Relacional
- Necessidades.
 - Processamento distribuído. Escalabilidade.
 - Fragmentação. Replicação.
 - Modelos de dados mais eficientes.

Problema do modelo Relacional ou das suas implementações??

Big Data (NoSQL)

Movimento No SQL (No SQL -> Not Only SQL)

- Modelos de dados alternativos e simples.
 - Document stores : MongoDB, CouchDB
 - Key-value stores: Amazon Dynamo,
 - Column stores: Google BigTable, Cassandra
 - Graphs stores: Neo4J, Infinite Graph, OrientDB, etc
- Distribuição do processamento. Escalabilidade
- Flexibilidade de esquema.
- Flexibilidade no suporte a ACID.
- Soluções específicas para problemas específicos
- Aplicações Poliglotas

Big Data (NewSQL)

Movimento New SQL:

- Linguagens de consulta mais sofisticados baseados no modelo relacional.
- Bancos de Dados baseados em memória.
- Transações leves. No ACID
- Novos modelos de dados:
 - Modelos baseados em array : VoltDB, SciDB
 - Gerenciadores multimodelos NoSQL.: OrientDB
- Modelo Relacional com implementações distribuídas ?

Consenso

"One size does not fit all."

Soluções específicas de Bancos de Dados para aplicações específicas