

# My name is Base, Database

João Marcelo Borovina Josko

`marcelo.josko@ufabc.edu.br`

June 10, 2017

# Tópicos

- 1 Introdução
  - Dados, Banco de Dados e SGBD
- 2 A 1a. Onda
  - Os Primórdios
  - The World is ACID
- 3 A Nova Onda
  - O mundo muda
  - A Tempestade
- 4 New Players
  - NoSQL
  - NewSQL
  - In-memory DB
  - Visão Comparada

# Tópicos

- 1 **Introdução**
  - Dados, Banco de Dados e SGBD
- 2 A 1a. Onda
  - Os Primórdios
  - The World is ACID
- 3 A Nova Onda
  - O mundo muda
  - A Tempestade
- 4 New Players
  - NoSQL
  - NewSQL
  - In-memory DB
  - Visão Comparada

# Nice to meet you, I am data!

- Remetem as propriedades que caracterizam fatos de um Universo de Discurso
- Crescente percepção de que dados organizacionais são ativos
- Volume de dados capturados geram desafios....
  - **Organizar** os dados para uso posterior
  - Procedimentos de anotação e publicação dos dados (*data curation*)

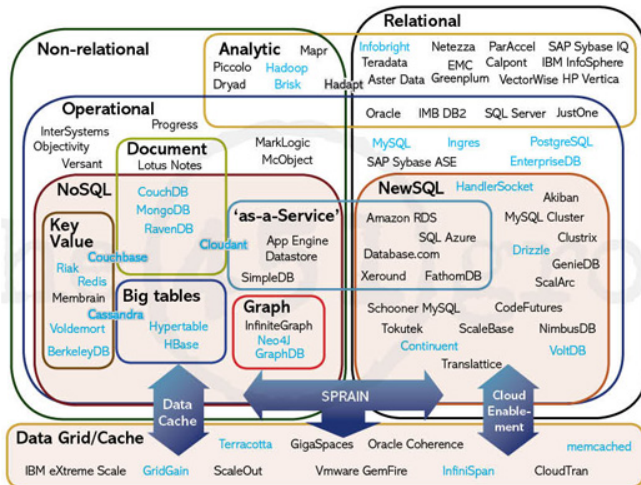
# O que caracteriza um Banco de Dados?

- Representam fatos de **interesse** sobre objetos do **Universo de Discurso**
- Coleção de fatos **relacionados**
- Somos todos um banco de dados....
  - Livro de Receitas (*Manual*)
  - Notas Fiscais de um E-Commerce (*Automatizado*)

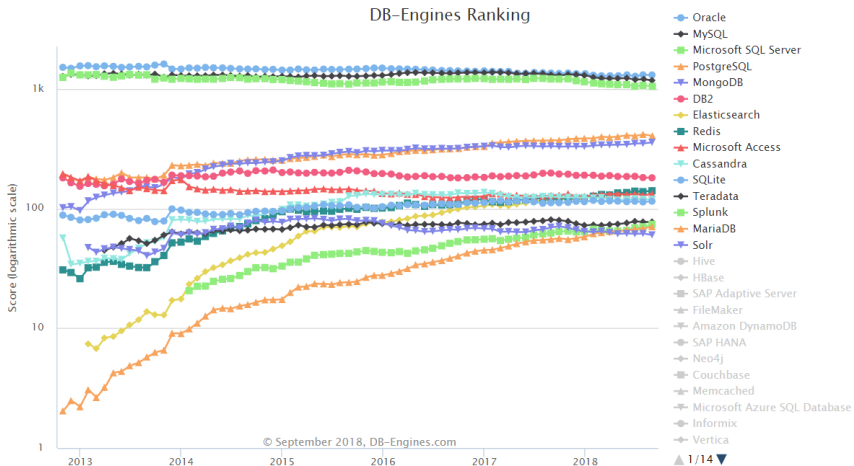
# O que caracteriza um **SGBD**?

- **Sistema Gerenciador de Banco de Dados**
- Software de propósito geral que permite....
  - Definir, modificar e gerenciar a estrutura de um Banco de Dados
  - Manipular as instâncias de um Banco de Dados
  - Gerir a segurança das instâncias de um Banco de Dados
  - Construir objetos programados estáticos e dinâmicos
  - Aplicar estratégias de Recuperação de um Banco de Dados
  - Acessar itens de dados de modo concorrente
  - Recuperação automática de vários tipos de falhas
- **Pero.....**
  - Nem todos os SGBDs oferecem todo esse conjunto de recursos.

# Highway to Hell



# Database Market Share



Fonte: [https://db-engines.com/en/ranking\\_trend](https://db-engines.com/en/ranking_trend)



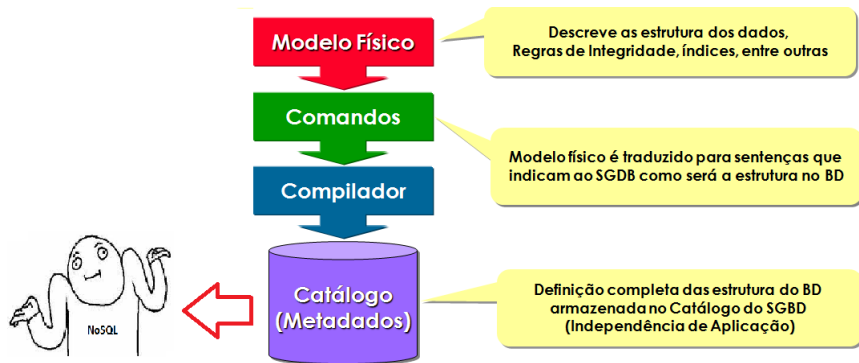
# Why so many?

- SGBDs variam em propriedades e capacidades....
  - Arquiteturas de Processamento (Centralizada, Distribuída, etc.)
  - Algoritmos internos (Índexação, operadores de consultas, etc.)
  - *Features* (Administração, Programação, etc.)
  - Modelo de Dados
    - Conjunto de construtores para descrever os dados
    - E.g., Relacional, Documento, Objeto-Orientado, Grafo, etc.

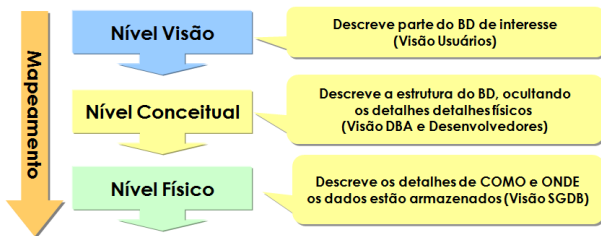
# Propriedades do SGBD

- Natureza Autodescritiva
- Abstração de Dados
- Suporte a múltiplas visões do mesmo dado
- Compartilhamento dos mesmos dados de modo Concorrente
- **However....** os SGBDs oferecem esses recursos de modo desigual

# Propriedades - Natureza Autodescritiva



# Propriedades - Abstração de Dados



- Possibilita o isolamento entre dados e programas
- Capacidade de modificar um nível sem afetar o outro
- Exemplos de Modificações....
  - Ajuste de regras de integridade (Conceitual  $\nRightarrow$  Visão)
  - Criar um índice (Físico  $\nRightarrow$  Conceitual)

# Propriedades - Demais

- Os **mesmos dados** podem ser observados segundo perspectivas diferentes
  - Notas Fiscais Sintéticas ou de SP
- Os mesmos dados serão utilizados (transacionados) de modo simultâneo
  - SGBD garante uso concorrente do mesmo dado
  - Uma transação  $\mathcal{A}$  não interfere em outras transações  $\mathcal{B}, \mathcal{C}, \dots, \mathcal{Z}$

# Tópicos

- 1 Introdução
  - Dados, Banco de Dados e SGBD
- 2 A 1a. Onda
  - Os Primórdios
  - The World is ACID
- 3 A Nova Onda
  - O mundo muda
  - A Tempestade
- 4 New Players
  - NoSQL
  - NewSQL
  - In-memory DB
  - Visão Comparada

# No início tudo era arquivo!

- Dados armazenados em arquivos do Sistema Operacional
  - Manipulação requeria vários programas
  - Não há independência entre a estrutura e as regras de negócio
  - Vários problemas...
    - Redundância de dados (Utiliza uma única perspectiva)
    - redundância de regras de negócio
    - Impossibilidade de acesso concorrente

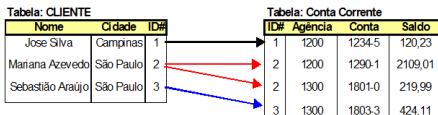
# Depois viram os primeiros modelos de dados

- Dados representados por coleções de ponteiros
  - Modelo em **Rede**: Listas Encadeadas
  - Modelo **Hierárquico**: Organização em Árvore
  - Vários problemas...
    - Representação N:N
    - Enorme dificuldade de manutenção
    - Dependência de Implementação



# Here I am

- A abordagem **Relacional** marcou o início de uma era
  - Relacionamentos estabelecidos a partir dos próprios valores dos dados
  - Proporcionou a independência entre a camada física e lógica dos dados
  - Disponibilizou uma sofisticada linguagem declarativa *english-like*
  - Facilitou a administração e gestão de segurança
  - Formalização de execução consistente e com garantia de recuperação a falha



# Here I am (Cont.)



Sua concepção bem feita propiciou...

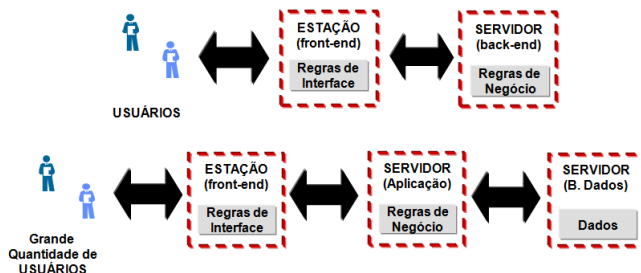
- Enfocar nas necessidades da aplicação e não em detalhes técnicos
- Dividir com o SGBD a responsabilidade de consistência
- Sinônimo de confiança e consistência

# “Sua força é também sua fraqueza”

- Nos anos 70, 80 até meados dos 90 os pressupostos eram [Stonebraker *et al.* 2007]...
  - Estruturas de dados altamente estruturados
    - Disponível antes de qualquer carga de dados
  - Tipos de dados baseados em números e texto
  - Garantia estrita de consistência
  - Transações curtas e de manutenção
    - Características do OLTP
  - Garantia de Recuperação baseada em *LOG*
  - Mundo Centralizado e baseado em disco
    - Capacidades de Processadores e Memória diferentes das atuais

# Arquiteturas

- Centralizada
  - Sistemas de Informação e o SGBD compartilham a mesma máquina
- Arquitetura Cliente-Servidor (*N-Tier*)
  - Cliente assume operações de interface gráfica Servidor assume responsabilidade pelas regras de negócio e controle de transações



# Tópicos

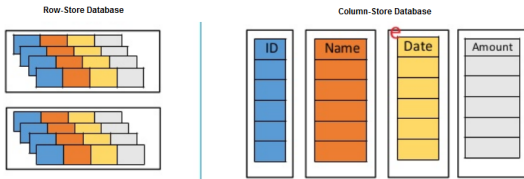
- 1 Introdução
  - Dados, Banco de Dados e SGBD
- 2 A 1a. Onda
  - Os Primórdios
  - The World is ACID
- 3 A Nova Onda
  - O mundo muda
  - A Tempestade
- 4 New Players
  - NoSQL
  - NewSQL
  - In-memory DB
  - Visão Comparada

# Surge uma brisa do mar

- Surgem certas necessidades acirradas pela WEB.....
  - Manipular outros tipos de dados
    - *XML, BLOB, Espacial*
  - Quero fazer mais análises do que transações
    - *MOLAP, HOLAP, ROLAP*
  - E-commerce não utiliza transações curtas
    - Transações longas (*Long – lived Transactions*)
  - Maior *escalabilidade*, menor centralização
    - Gerenciadores de Banco de Dados Distribuídos e Paralelos

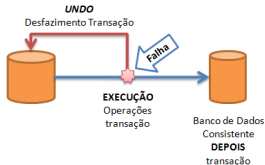
# Surge uma brisa do mar (Cont.)

- Análise Tradicional
  - Modelos Multidimensionais
    - Permitem **correlação de variáveis** (Causa  $\rightarrow$  Efeito)
    - Diferentes operações: *Pivot*, *Roll-up*, *Drill-down*, *Slice*, ...
  - Estrutura
    - *Dimensões* denotam as variáveis de análise
    - *Fato* denota os dados correlacionados pelas variáveis
  - Necessidade deu origem ao **C-Store Databases** (e.g., Vertica, MonetDB)



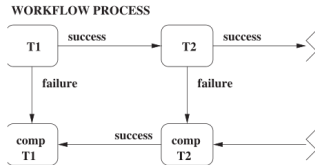
## Surge uma brisa do mar (Cont.)

- Transações Curtas
  - Regidas pelas propriedades ACID [Borovina Josko 2012]
- Transações Longas
  - Atomicidade é garantida por operações de compensação [Alonso 2005]

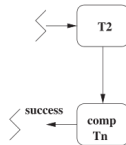


Fonte: BOROVIĆA JOSKO 2012.

**ACID**



### SAGAS - Compensação

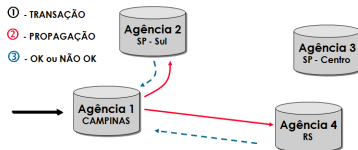


Fonte: Alonso 2005.



# Surge uma brisa do mar (Cont.)

## ● Arquitetura Distribuída



Dados fragmentados (Vertical, Horizontal e Mista)

Compartilham um esquema global comum

Elementos de processamento

- Homogêneos com níveis de autonomia
- Heterogêneos autônomos (Federado)

Transparência de Replicação, Fragmentação e Execução

# E a brisa era uma tempestade

- O Dilúvio de dados [Strauch *et al.* 2011, Leavitt 2010]
  - *WEB* 2.0, Sensores Remotos, Celulares, IoT geram enorme volume de dados
  - Novos dispositivos permite persistir o volume de dados
  - Dados agora em tipos e *shapes* variados
  - Relacionamentos mais complexos
  - Ter dados não significa nada (Desafios e Oportunidades)
    - Como organizá-los?
    - Como integrá-los?
    - Que padrões esses apresentam que podem ser úteis para mim?
    - Descompasso entre PRODUÇÃO e CAPTURA de dados e ANÁLISE

# E a brisa era uma tempestade (Cont.)

- E o que impacta o mundo de Banco de Dados?
  - Aumento do volume de dados capturados e armazenados
    - Necessidade de maior *throughput*
  - Tratamento de diferentes tipos de dados (Documentos, imagem, vídeo)
  - Aplicações que não requerem consistência estrita em todas as situações
  - Dados menos estruturados e com relacionamentos mais complexos
  - Maior enfoque no **Analítico** e não em OLTP e OLAP
  - Latência próxima de zero (Rede com alta capacidade)

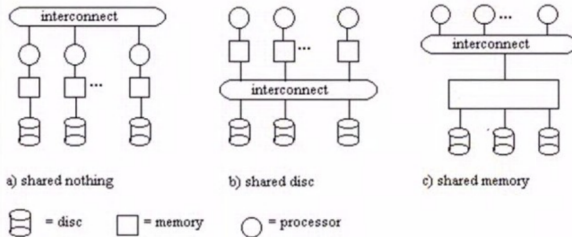
# E a brisa era uma tempestade (Cont.)

- Novas necessidades determinaram limites do modelo de dados Relacional
  - Escalabilidade limitada e cara [Stonebraker *et al.* 2007]
    - Crescimento horizontal e baseado em máquinas baratas
  - Consistência estrita é um *overhead* em certos contextos [Leavitt 2010]
    - Troca de consistência por desempenho
  - Impendância Objeto → Relacional é um *overhead*
  - Dificuldade de acolher dados com relacionamentos mais complexos e estruturas dinâmicas
  - Gestão do desempenho é cara e baseada em pessoas
    - *Self-tuning* disponível não é abrangente

# Qual seria a arquitetura apropriada?

- Disponibilidade de memória permite dados OLTP em memória  
[Stonebraker *et al.* 2007]
- Complexidade requer capacidades de *self-tuning*
- Modelo capaz de atender a dinâmica do negócio
- Persistência Poliglota (diferentes tipos de dados)
- Distribuir o processamento de dados de modo barato
  - Arquitetura *Shared-nothing* com máquinas baratas [Stonebraker *et al.* 2007]
  - Capacidade de Particionamento Horizontal do Processamento
  - Facilidade de Expansão
  - Alta disponibilidade com replicação dos dados

# Principais Arquiteturas



Shared-nothing	Shared-Disk	Shared-Memory
Nada é compartilhado	Proc. compartilham Discos	Proc. compartilham Discos e Memórias
↓ <i>Load Balancing</i> Dependente (Estr. Particionamento)	Certa capacidade <i>Load Balancing</i>	↑ <i>Load Balancing</i> em tempo de execução
↑ Baixo custo	↑ Fácil migração do Centralizado	↑ Simplicidade
↑ Alta disponibilidade	Certa disponibilidade	Certa disponibilidade
↑ Fácil Expansão	Disco e capacidade de rede	↓ Limite no número de processadores
↑ Nós independentes	↓ Requer controle de Lock (Coerência de cache)	Controle dos proc. por único SO
		↓ Requer conexões de rede especiais

Table: Comparação das Arquiteturas

# Thread-off!!!

- Conjectura CAP - propriedades para um mundo distribuído [Brewer 2000; 2012]
  - **Consistency**
    - Todos os nós observam o mesmo dado ao mesmo tempo
  - **Availability**
    - Cada requisição recebe uma resposta de sucesso ou falha
  - **Partition Tolerance**
    - Funcionamento contínuo com perda de mensagens ou queda parcial da *network*
- Não podemos perfeitamente ter todas as três propriedades!

# Thread-off!!! (Cont.)

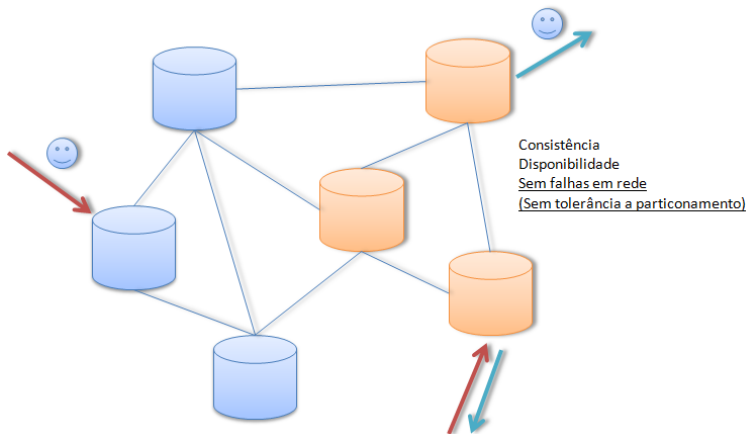


Figure: Exemplo propriedades CAP (Fonte: BOROVINA JOSKO, J.M.)



# Thread-off!!! (Cont.)

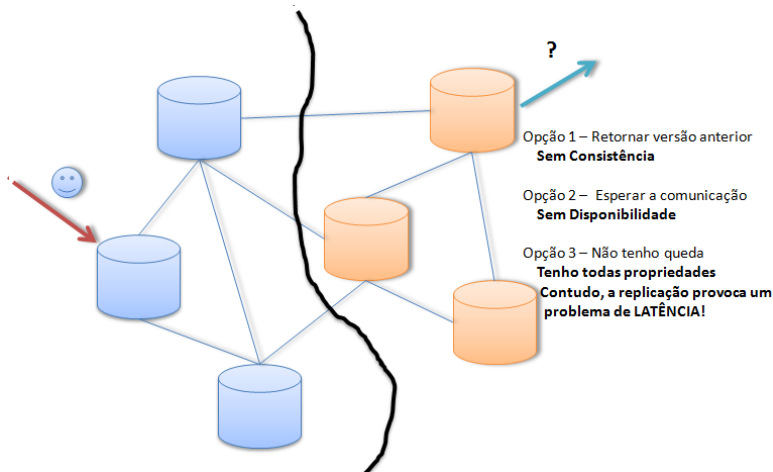


Figure: Exemplo propriedades CAP (Fonte: BOROVINA JOSKO, J.M.)

# So, Let's relax ACID!

- Proposto o modelo BASE [Brewer 2000; 2012]
  - **B**asically available
  - **S**oft-state
  - **E**ventual consistency
- “Estou disponível a qualquer momento, mas não tenho consistência estrita. Eventualmente, ao longo do tempo, serei consistente”
- Consistência e Isolamento são preteridas em favor da disponibilidade

# So, Let's relax ACID! (Cont.)

ACID	BASE
Consistência estrita	Consistência Eventual (Não falha)
Isolamento	Disponibilidade (quero ver agora)
Abordagem Pessimista	Abordagem Otimista (Be happy!)
Foco no commit	Utiliza uma aproximação
Esquema Rígido	<i>Schemaless</i>
Garantia pelo SGBD	Garantia pelo programador

Table: ACID *versus* BASE

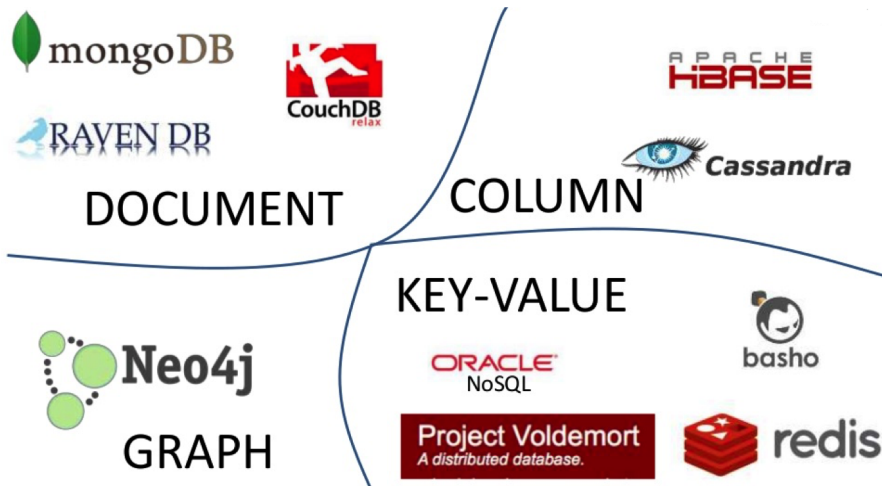
# Tópicos

- 1 Introdução
  - Dados, Banco de Dados e SGBD
- 2 A 1a. Onda
  - Os Primórdios
  - The World is ACID
- 3 A Nova Onda
  - O mundo muda
  - A Tempestade
- 4 **New Players**
  - NoSQL
  - NewSQL
  - In-memory DB
  - Visão Comparada

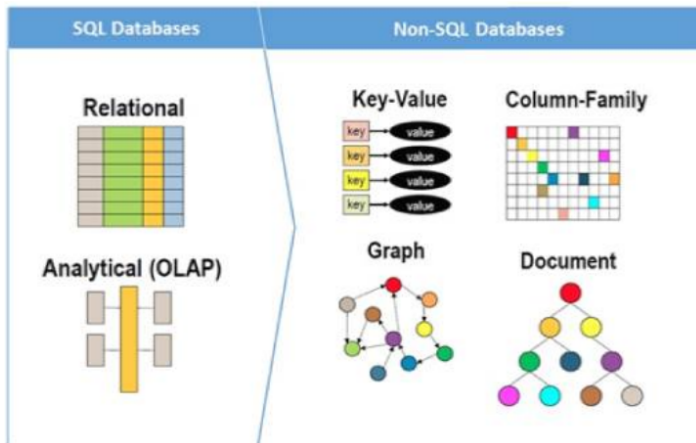
# Comment ça va? Je suis NoSQL

- Baseados nas propriedades do BASE
  - Resposta contrária ao ACID e ao mundo Relacional
- Orientados a distribuição massiva
  - Mecanismos de particionamento dos dados
  - Recursos de *load-balancing*
  - Uso intensivo de replicação de dados
  - It's a kind of magic?
- Estruturas flexíveis. Relacional NO MORE!
  - Novos modelos Grafo, Key-Value, Documento (extensão do key-value), etc.

# Principais NoSQL



# Principais Modelos de Dados do NoSQL



# Não mexa no meu queijo!



## Não mexa no meu ACID!

- Consigo ter desempenho compatível às necessidades atuais?
- Não dá para particionar horizontalmente?
- Não dá para ser mais tolerante a falhas na rede?



# Comment ça va? Je suis NewSQL

- Novo OLTP (escalabilidade do NoSQL + ACID) [Stonebraker 2012, Pavlo e Aslett 2016]
  - Escritos para arquitetura distribuída baseada em recursos *shared-nothing*
  - Dados em memória (na maioria das opções)
  - Respostas analíticas em tempo-real
  - Controle de concorrência baseada em timestamp (MVCC) em não 2PL
- Asta la vista, baby!!!
  - Esquema OLTP + ETL + DW/BI está com os dias contados

# ¿Cómo estás? Me llamo In-memory Database

- Dados armazenados totalmente em memória RAM [Faerber *et al.* 2017]
  - Acesso mais rápido (Memória  $\simeq 100\text{ns}$ , SDD  $\simeq 150000\text{ ns}$ )
  - Algoritmos não enfatizam o uso de disco
  - Compressão dos dados em memória
  - Mecanismos otimistas de controle de concorrência

# Caracterização Comparada

<i>Características</i>	Relational Trad.	NoSQL	NewSQL
Relacional	<i>Sim</i>	<i>Não</i>	<i>Sim</i>
SQL	<i>Sim</i>	<i>Não</i>	<i>Sim</i>
Consistência Estrita	<i>Sim</i>	<i>Não</i>	<i>Sim</i>
Consistência Eventual	<i>Não</i>	<i>Sim</i>	<i>Não</i>
Suporte a transações	<i>Sim</i>	<i>Não</i>	<i>Sim</i>
Escalabilidade Horizontal	<i>Não</i>	<i>Sim</i>	<i>Sim</i>
Escalabilidade Vertical	<i>Sim</i>	<i>Sim</i>	<i>Sim</i>
Esquema Estático	<i>Sim</i>	<i>Não</i>	<i>Sim</i>
Alto Desempenho (Big Data)	<i>Não</i>	<i>Sim</i>	<i>Sim</i>
Dificuldades em atualizações não simétricas	<i>Não</i>	<i>Sim</i>	<i>Não</i>
Dificuldades em acessos não simétricos	<i>Não</i>	<i>Sim</i> <sup>1</sup>	<i>Não</i>

**Table:** Comparativo das Características

*1 - Exceto baseados em Grafo*

# Hora de acordar!!!!



*Calvin and Hobbes by William B. Watterson II*

Um momento de análise...

- Quais problemas e desconfortos surgem nesse cenário?
- Quais oportunidades surgem nesse cenário?
- Como fica a integração?

# Referências I

- Alonso (2005)** Gustavo Alonso. Transactional business processes. *Process-Aware Information Systems: Bridging People and Software through Process Technology*, Wiley-Interscience, New York, páginas 257–278.
- Borovina Josko (2012)** J.M. Borovina Josko. Transações Aninhadas e Savepoints: Extensões ao Modelo Transacional Clássico. *SQL magazine*, 102:50 – 55.
- Brewer (2012)** Eric Brewer. CAP twelve years later: How the" rules" have changed. *Computer*, 45(2):23–29.
- Brewer (2000)** Eric A Brewer. Towards robust distributed systems. Em *PODC*, volume 7.
- Faerber et al. (2017)** Franz Faerber, Alfons Kemper, Per-Åke Larson, Justin Levandoski, Thomas Neumann, Andrew Pavlo et al. Main memory database systems. *Foundations and Trends® in Databases*, 8(1-2):1–130.
- Leavitt (2010)** Neal Leavitt. Will NoSQL databases live up to their promise? *Computer*, 43(2).
- Pavlo e Aslett (2016)** Andrew Pavlo e Matthew Aslett. What's Really New with NewSQL? *ACM Sigmod Record*, 45(2): 45–55.
- Stonebraker (2012)** Michael Stonebraker. Newsql: An alternative to nosql and old sql for new oltp apps.
- Stonebraker et al. (2007)** Michael Stonebraker, Samuel Madden, Daniel J Abadi, Stavros Harizopoulos, Nabil Hachem e Pat Helland. The end of an architectural era:(it's time for a complete rewrite). Em *Proceedings of the 33rd international conference on Very large data bases*, páginas 1150–1160. VLDB Endowment.
- Strauch et al. (2011)** Christof Strauch, Ultra-Large Scale Sites e Walter Kriha. NoSQL databases. *Lecture Notes*, Stuttgart Media University, 20.