



Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"

**MARCELL GUERREIRO SILVA**

## **BANCO DE DADOS EM MEMÓRIA (*IN-MEMORY*)**

Assis

2013

**MARCELL GUERREIRO SILVA**

**BANCO DE DADOS EM MEMÓRIA (*IN-MEMORY*)**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e Fundação Educacional do Município de Assis – FEMA, como requisito do Curso de Graduação.

**Orientador:** Prof. Dr. Alex Sandro Romeo de Souza Poletto.

**Área de Concentração:** Informática

Assis

2013

## FICHA CATALOGRÁFICA

SILVA, Marcell Guerreiro

Banco de dados em memória (*In-Memory*) / Marcell Guerreiro Silva. Fundação Educacional do Município de Assis – FEMA – Assis, 2013.

50 p.

Orientador: Prof. Dr. Alex Sandro Romeo de Souza Poletto

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA

1. *In-Memory* 2. Banco de Dados 3. Agilidade 4. Disponibilidade 5. Volume

CDD: 001.6

Biblioteca da FEMA

# **BANCO DE DADOS EM MEMÓRIA (*IN-MEMORY*)**

**MARCELL GUERREIRO SILVA**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e Fundação Educacional do Município de Assis – FEMA, como requisito do Curso de Graduação.

Orientador: Prof. Dr. Alex Sandro Romeo de Souza Poletto

Analizador: Prof. Esp. Guilherme de Cleve Farto

Assis  
2013

## **AGRADECIMENTOS**

Agradeço, primeiramente, a Deus por me dar condições para a realização de um sonho em meio a muitas dificuldades, sem Ele não poderia chegar até onde cheguei.

Aos meus pais Marcos e Ligia e minha irmã Déborah, que me incentivaram e me deram força para sempre continuar, sem desistir dos sonhos que eles também sonharam comigo.

Especialmente a Ana Elisa pela paciência, companheirismo, amizade e amor. Uma pessoa muito importante e que tem feito muita diferença em minha vida, sempre me apoiando em todas as coisas que faço.

Aos meus amigos que sempre em vários momentos ajudaram nos estudos compartilhando seus conhecimentos, sempre com muita disposição e humor. Esses anos de convívio não serão esquecidos.

E não também ao meu orientador, Professor Dr. Alex Sandro Romeo de Souza Poletto dando suporte nas horas das dúvidas e nos direcionando em cada passo na conclusão deste trabalho.

Que os vossos esforços desafiem as  
impossibilidades, lembrai-vos de que as grandes coisas do homem  
foram conquistadas do que parecia impossível.

Charles Chaplin

## RESUMO

Este trabalho tem por objetivo apresentar um estudo exploratório sobre Banco de Dados em Memória (*In-Memory*), mostrando como são ordenadas as estruturas de dados na memória volátil, bem como realizar um comparativo com relação às propriedades ACID entre os Sistemas de Gerenciamento de Banco de Dados convencionais e o *In-Memory*. Além disso, serão apresentados alguns estudos de caso de grandes empresas que recorreram a esta nova vertente de Banco de Dados para conseguir suprir as necessidades de análise e de processamento de grandes volumes de dados em tempo real e de forma consistente.

**Palavras-chave:** *In-Memory*; Agilidade; Disponibilidade; Volume; Propriedades ACID.

## **ABSTRACT**

This work aims to present an exploratory study on Database in Memory (In-Memory), showing how the structures are ordered data in volatile memory, and perform a comparative relation between the ACID properties Database Management Systems conventional Data and in-Memory. In addition, some case studies of large companies that used this new strand Database to achieve meet the needs of analysis and processing of large volumes of data in real time and consistently will be presented.

**Keywords:** In-Memory; Agility; Availability; Volume; ACID Properties.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Garantia das Propriedades ACID .....	19
Figura 2 - Círculo de Desempenho Empresarial com In-Memory.....	22
Figura 3 - Comparação do modelo Orientado a Linhas x Modelo Orientado a Coluna .....	25
Figura 4 - Amostra das colunas em memória com dicionário e atributo vetor. ....	28
Figura 5 - Resultado da busca pelo nome e sobrenome dos dados a serem excluídos. ....	29
Figura 6 – Busca dos dados para exclusão no atributo vetor.....	29
Figura 7 – Realocação dos dados no atributo vetor .....	30
Figura 8 - Tabela de exemplo população mundial.....	31
Figura 9 – Exemplo da análise no dicionário antes de uma nova inserção.....	32
Figura 10 – Exemplo do valor encontrado no dicionário .....	32
Figura 11 – Exemplo do dado gravado no vetor de atributos. ....	33
Figura 12 – Exemplo da leitura do dicionário de nomes.....	33
Figura 13 – Exemplo de inserção no dicionário de nomes.....	34
Figura 14 – Exemplo ordenação do dicionário de nomes. ....	34
Figura 15 – Exemplo da ordenação do vetor de atributos. ....	35
Figura 16 – Exemplo da tabela com os dados inseridos e ordenados. ....	35
Figura 17 – Tabela população mundial. ....	36
Figura 18 – Representação do plano de execução. ....	37



# SUMÁRIO

1. INTRODUÇÃO .....	13
1.1. OBJETIVOS .....	14
1.2. JUSTIFICATIVAS.....	14
1.3. MOTIVAÇÃO.....	15
1.4 ESTRUTURA DO TRABALHO.....	15
2. BANCO DE DADOS RELACIONAL .....	17
2.1 BREVE HISTÓRICO SOBRE BANCOS DE DADOS RELACIONAIS .....	17
2.2 ESTRUTURA DOS BANCOS DE DADOS RELACIONAIS .....	18
2.3 PROPRIEDADES ACID .....	18
2.4 ARMAZENAMENTO DE DADOS.....	20
3. BANCO DE DADOS EM MEMÓRIA .....	21
3.1 ARMAZENAMENTO DOS DADOS E AS PROPRIEDADES ACID .....	23
3.2 ORGANIZAÇÃO DOS DADOS .....	25
3.3 COMPRESSÃO DOS DADOS .....	26
3.3.1 <i>Cluster encoding</i> .....	26
3.3.2 Run lenght encoding .....	26
3.3.3 <i>Prefix encoding</i> .....	27
3.4 OPERAÇÕES ESSENCIAIS REALIZADAS NO BANCO DE DADOS <i>IN-MEMORY</i> .....	27

3.4.1 Função <i>delete</i> .....	27
3.4.2 Função <i>insert</i> .....	30
3.4.3 Função <i>select</i> .....	35
3.5 EXEMPLOS DO USO DO <i>IN-MEMORY</i> .....	38
3.5.1 Sensores de dados .....	38
3.5.2 Análise de eventos de jogos .....	39
3.6 PROPIEDADES ACID: BANCO DE DADOS CONVENCIONAIS X BANCO DE DADOS EM MEMÓRIA .....	40
3.6.1 PROPRIEDADES ACID EM BANCO DE DADOS CONVENCIONAIS .....	40
3.6.2 PROPRIEDADES ACID EM BANCO DE DADOS EM MEMÓRIA.....	41
4. ESTUDOS DE CASOS: EXEMPLOS PRÁTICOS.....	43
4.1. SAMSUNG APPS.....	43
4.2. SK TELECOM .....	44
4.3 LEADING JAPANESE INVESTMENT BANK - THE FIRM .....	45
5. CONCLUSÃO.....	47
5.1 TRABALHOS FUTUROS .....	47
REFERÊNCIAS.....	49

## 1. INTRODUÇÃO

Com a evolução das tecnologias, as empresas precisam se adequar rapidamente para a incorporação das novas formas de informações, pois logo que as informações entram em uma organização, já se inicia um processo de tempo útil, ficando ultrapassada muito rapidamente e conseqüentemente o seu valor diminui podendo até se tornar obsoleta e irrelevante. Por isso as informações em tempo real são valiosas demais para ficarem esperando alguma tarefa para sua utilização. Com os dados armazenados em memória, o tempo de processamento e sua própria alocação podem ser executados em tempo real podendo ser utilizados imediatamente (FRENKIEL, 2013).

Uma solução da qual, grandes empresas e organizações estão optando, devido sua agilidade, praticidade e pronta resposta é o banco de dados em memória. Grandes empresas especializadas em soluções empresariais e na manipulação de banco de dados, como a *Oracle* e a *SAP*, estão investindo altos valores neste novo conceito, criando ferramentas que executam procedimentos de armazenagem na memória, como por exemplo, a *TimesTen In-Memory Database* e a *Sybase ASE*. Tais produtos, quando em interação com outras ferramentas já existentes, como o *Business Intelligence* (BI) auxiliam as organizações a tomarem decisões com mais rapidez e de forma mais precisa.

Os Bancos de Dados *In-Memory* permitem que o banco fique todo na memória principal do sistema, quebrando assim o maior gargalo dos sistemas de banco de dados comuns: a leitura e gravação dos dados no disco. O banco de dados em memória não há necessidade de alterações na aplicação para que o sistema se adeque a ele, pois a aplicação o vê como um banco de dados não importando se é um banco de dados convencional ou um banco *in-memory*, diferente de outras soluções similares que utilizam a tecnologia como, por exemplo, NoSQL (PRONI, 2013).

Estes bancos tem um sistema de gerenciamento de informações, que depende diretamente da memória principal para melhorar o tempo de resposta, os dados são carregados na memória do sistema e comprimido em um formato não relacional,

permitindo uma agilidade maior para a realização das consultas. Ele é um banco do tipo analítico, que armazena dados históricos sobre métricas para serem lidos por aplicações como *Business Intelligence* (BI) ou *Business Analytics* (BA) que normalmente fazem parte dos processos de um *Data Warehouse*, permitindo aos seus usuários executar consultas e gerar relatórios sobre as informações contidas no banco, sendo que sua atualização é realizada regularmente (ROUSE, 2012).

### 1.1. OBJETIVOS

O objetivo deste projeto é estudar uma nova ferramenta de banco de dados que ainda é pouco utilizada, mas que vem ganhando espaço no mercado pela sua agilidade em consultar bases de dados com grande volume de informações. Além disso, será feito um breve comparativo da garantia das propriedades ACID, que são as premissas básicas de um Sistema de Gerenciamento de Banco de Dados.

Principalmente em uma época em que a mineração de dados está em foco nas principais empresas do mercado e precisam de muita velocidade na busca para conseguir atender a demanda de serviços solicitados por seus clientes, parceiros e acionistas de forma eficiente e rápida.

Com esta pesquisa pretende-se contribuir para a área de sistemas de banco de dados no que se diz respeito geração de um material científico para novos trabalhos sobre banco de dados em memória.

### 1.2. JUSTIFICATIVAS

Como todas as empresas e organizações estão sempre em uma corrida por espaço no mercado para sair na frente dos concorrentes, é preciso que todas as tomadas de decisão sejam feitas em tempo real, para isso a tecnologia tem que se adequar às necessidades do mercado. Para isso cada vez mais surgem soluções para que os dados não fiquem ultrapassados e sejam utilizados de forma rápida e precisa. Isso

justifica a necessidade de estudos voltados para a área de banco de dados em memória e a confecção de materiais literários sobre o tema.

Outra justificativa para este trabalho é a busca por conhecimento e a vontade de aprender o que se tem de mais atual no mercado de trabalho, de forma a ajudar a comunidade com o conhecimento adquirido por meio desta pesquisa.

### 1.3. MOTIVAÇÃO

O conceito de banco de dados em memória vem sendo criado há quase três décadas, porém no passado era quase inviável o desenvolvimento de um sistema de gerenciamento de banco de dados em memória principal, pois o hardware não tinha uma capacidade de processamento como nos dias de atuais. Então, com melhoria no *hardware* e a queda dos custos sobre estes equipamentos a motivação para o início ao desenvolvimento de *softwares* e ferramentas baseadas nos conceitos *in-memory* começaram a surgir.

Considerando se tratar de um assunto um tanto quanto novo, a motivação acaba se tornando ainda maior, gerando a necessidade de obter conhecimentos sobre este tema, que possui um campo muito grande de vertentes e uma notória falta de mão de obra qualificada para preencher as vagas que estão disponíveis no mercado. Além disso, pretende-se confeccionar um bom material para apoiar demais pesquisas, de forma a contribuir com subsídios para projetos futuros.

### 1.4 ESTRUTURA DO TRABALHO

O trabalho está estruturado da seguinte forma:

Capítulo 1 – Introdução: Neste capítulo é apresentado uma breve introdução do tema apresentado, assim como os objetivos do trabalho, justificativas para a realização e a motivação e como estará estruturado o presente trabalho.

Capítulo 2 – Banco de Dados Relacional: Capítulo onde é relatado sobre o banco de dados relacional, com um breve histórico, sua arquitetura, propriedades ACID.

Capítulo 3 - Banco de Dados em Memória: São apresentadas algumas das principais funcionalidades deste banco, como compressão dos dados na memória, organização e as funções básicas como *INSERT*, *DELETE*, *SELECT* assim como alguns exemplo do uso deste banco e um comparativo entre a sua propriedade ACID com o modelo de dados relacional.

Capítulo 4 – Estudos de Casos: exemplos práticos: Aborda empresas que tiveram problemas com sua arquitetura de dados e necessitaram de novas ferramentas para solucioná-los.

Conclusão: Encerramento do estudo após o levantamento de informações sobre o comparativo entre o modelo de dados relacional e o banco de dados em memória.

Referências: Materiais base deste estudo para futuras consultas relacionadas ao presente tema.



## 2. BANCO DE DADOS RELACIONAL

Neste capítulo serão explanados os conceitos sobre Sistemas de Banco de Dados Relacionais (SDBR).

Os sistemas de gerenciamento de dados com o passar dos anos se tornaram muito utilizados pela sua facilidade em acesso aos dados, flexibilidade para modelagem de dados e uma linguagem de fácil entendimento para manipulação dos dados.

### 2.1 BREVE HISTÓRICO SOBRE BANCOS DE DADOS RELACIONAIS

Atualmente estamos acostumados a lidar com os benefícios trazidos pelos bancos de dados relacionais, com sua capacidade de armazenamento e o acesso rápido aos dados em computadores de baixo custo. Mas até a criação destes modelos relacionais os dados eram armazenados em uma estrutura hierárquica inflexível e de difícil navegação (ROB, CORONEL, 2011, p.12).

Segundo Silberschatz (2006, p.61), “o modelo relacional estabeleceu-se como o primeiro modelo de dados para aplicações comerciais”

Em 1970, um matemático da IBM chamado Edgar Codd, usando uma notação matemática conseguiu resumir várias páginas de códigos complexos em uma única linha de programação. Por meio de tal habilidade e domínio em programação, ele motivou a IBM a criar um novo projeto, intitulado Sistema R, que se destinava a construir um protótipo de banco de dados relacional, vindo a se tornar a SQL e o banco de dados DB2 da IBM (ROB, CORONEL, 2011, p.12).

O modelo de dados relacional veio para dinamizar os sistemas que usavam os primeiros modelos de dados que eram de sistemas hierárquicos, sistemas baseados em modelo de rede e sistemas de arquivos invertidos, esses modelos tinham um grande gargalo quanto à flexibilidade na identificação de novas consultas e transações. Um dos principais problemas destes bancos era a mistura de

relacionamentos conceituais com o armazenamento e posicionamento físico dos registros nos discos. Estes tipos de sistemas não tinham capacidade para abstração de dados e independência entre dados e programas (ELMASRI; NAVATHE, 2011, p.15).

## 2.2 ESTRUTURA DOS BANCOS DE DADOS RELACIONAIS

O modelo relacional utiliza um conjunto de tabelas tanto para representar dados como para representar a relação entre eles, cada uma com um único nome (ELMASRI; NAVATHE, 2011, p.17).

Ainda segundo Elmasri; Navathe (2011, p.17), cada uma das linhas que formam a tabela também podem ser vistas como um registro, que representam um relacionamento entre um conjunto de valores. Quando uma tabela dessas se torna um conjunto de relacionamentos, se estreita o conceito de tabelas com o conceito matemático de relação. Do qual se origina o nome deste modelo de dados

## 2.3 PROPRIEDADES ACID

Os gerenciadores de banco de dados têm uma estrutura completa para atender as todas as propriedades de Atomicidade, Consistência, Isolamento e Durabilidade (ACID), tudo para proteger os dados e certificar que eles chegaram corretamente ao banco de dados, garantindo assim sua integridade.

De acordo com Issa (2011, p.14), pode-se descrever as propriedades ACID da seguinte forma:

- Atomicidade – é a capacidade de realizar satisfatoriamente as tarefas do banco de dados.
- Consistência – tarefa que permite ao banco de dados manter sua integridade quando as transações isoladas são executadas.

- Isolamento – é a característica que permite que execuções independentes sejam concluídas de forma integral sem que haja perda de rendimento devido a outras transações que estiverem em execução.
- Durabilidade – garante a consolidação das alterações realizadas pelas tarefas que foram executadas com sucesso, mesmo que após a conclusão do processo ocorra falhas no sistema do banco de dados

Segundo Silberschatz et al. (2006), as propriedades ACID tornam possíveis as avaliações da consistência lógica dos dados e das transações, isso a torna indispensável para que os dados cheguem a banco de dados da forma correta garantindo completamente a execução das transações.

A figura abaixo exemplifica as transações do banco de dados executando as propriedades do ACID.

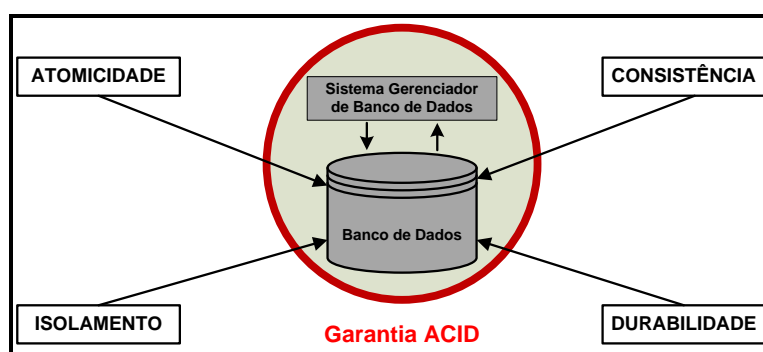


Figura 1 – Garantia das Propriedades ACID

Outros fatores favorecem para a popularização deste modelo de dados, como sua facilidade de utilização da linguagem SQL, tornando possível o suporte para gerenciar as estruturas de dados, linguem de programação PL/SQL para implantações de regras de negócio, rotinas automáticas através de *procedures*, *triggers* entre outros (SILBERSCHARZ et al, 2006).

## 2.4 ARMAZENAMENTO DE DADOS

Os armazenamentos dos dados no SGBDR são todos residentes permanentemente em disco físico, que são divididos em unidades de tamanho fixo chamados de blocos. Estes blocos são unidades para transferência de dados do disco ou para o disco contendo vários itens de dados (tuplas, registros), e são considerados armazenamentos temporários (*buffering*) que permite partes do banco carregadas na memória principal. (ELMASRI et al, 2005).

O processo de entrada e saída são todos executados em blocos, estes blocos gerados nas transações são enviados para o *buffer* que é a memória principal e depois para o disco. O movimento entre a memória (bloco de *buffer*) e o *hard disk* (bloco físico) são feitos por duas operações, o *input* que faz o movimento de transferir um bloco do disco físico para a memória *ram* do computador e o *output* que faz o movimento contrario, transferindo os blocos utilizados de volta ao disco físico.

Na realização do processo de transação do sistema, são incluídos uma ou mais operações para o acesso ao banco de dados, nestes modelos de banco de Dados todas as operações de inserção, exclusão, alteração e recuperação, realizadas são garantidas pelas Propriedades ACID do Sistema de Gerenciamento de Banco de Dados.

### 3. BANCO DE DADOS EM MEMÓRIA

As empresas estão em uma era na qual necessitam de informações instantâneas para poder auxiliar rapidamente em como serão tomadas as novas decisões para garantir mais agilidade e produtividade tanto para os negócios quanto para os usuários (PLATTNER; ZEIER, 2011, p.1).

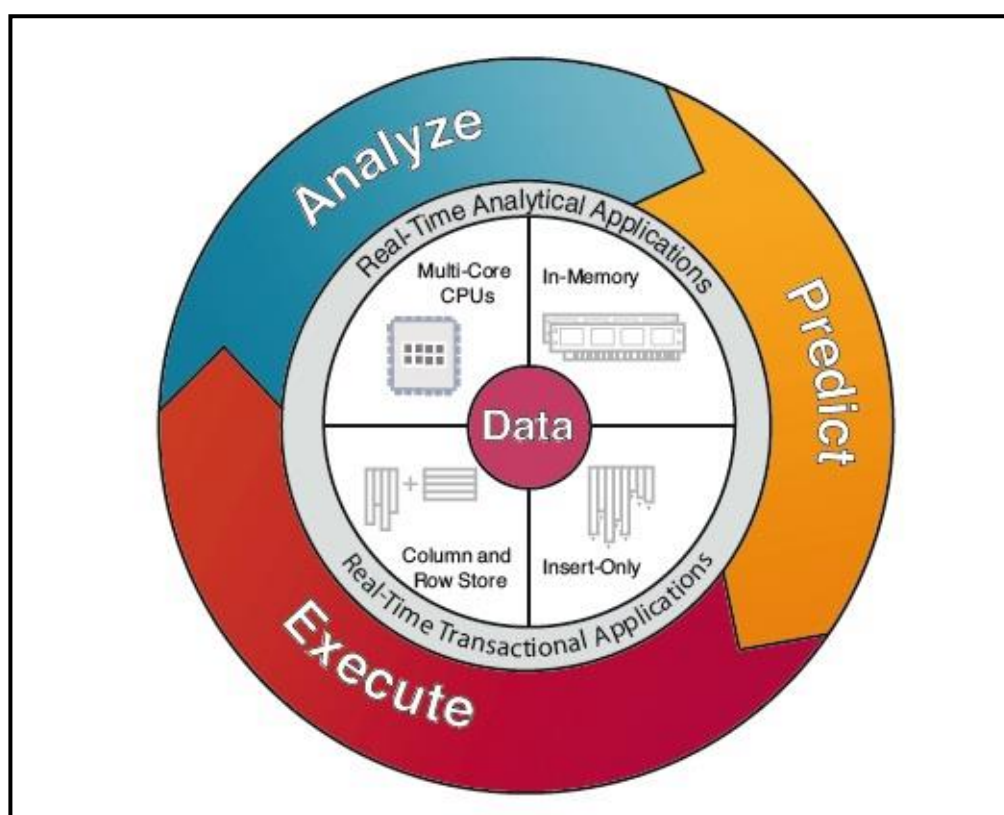
O banco de dados em memória chega como uma promessa de computação em tempo real, desempenhando um importante papel na condução dos valores de negócios sustentáveis, pois tal tecnologia ajuda a reduzir custos, acelerar os processos e mostrar novas possibilidades de como consumir as próprias aplicações. (PLATTNER; ZEIER, 2011, p.1).

A tecnologia *in-memory* ganhou espaço após a queda do custo das memórias de computador. Isso fez com que se tornasse viável um investimento nesta tecnologia. A proposta é tirar a trava na busca de informações em meio as grandes quantidades de dados existentes nos bancos. A trava nos Sistemas de Gerenciamento de Banco de Dados (SGBD) atuais são as entradas e saídas de dados no disco rígido (HD) dos servidores. Por mais rápido que seja a velocidade dos HD's ainda sim o processo é prolongado pela demora do acesso entre o dispositivo de armazenamento e o servidor de aplicação. (PLATTNER, ZEIER, 2011, p.1).

Não somente as tecnologias em torno das memórias sofreram uma evolução como fora dito por Plattner (2013, p.20), mas também os desenvolvimentos de processadores com um pensamento de paralelismo criaram um avanço considerável na arquitetura dos seus componentes e obtiveram resultados satisfatórios na criação de novos processadores como por exemplo, os *multi-core's* que são dotados de até oito núcleos de processamento em um único chip. Já as fabricantes de *Hard Disk's* (HDs) não tiveram grandes avanços que fizessem com que eles acompanhassem a evolução dos outros *hardwares*, o que resultou em um grande problema para trabalhar bases de dados e volumes grandes de informações (PLATTNER, 2013, p.20).

Os usuários dos negócios necessitam das informações dentro de curtos prazos para reagir rapidamente aos eventos de impacto nos negócios, assim se torna necessário obter informações cada vez mais precisa para reações imediatas nas situações de crise, por isso o sistema deve prover uma visão de grandes volumes de dados permitindo mudanças constantes e em tempo real. Os dados devem fornecer informações precisas e atualizadas a fim de se descobrir novas tendências e padrões (SAP AG. 2010, p.4).

A Figura 2 exibe como a junção das tecnologias permite uma análise instantânea e iterativa.



**Figura 2 - Círculo de Desempenho Empresarial com In-Memory (In: PLATTNER; ZEIER, 2011, p.22).**

Os princípios do banco de dados em memória são: velocidade, flexibilidade, volume e disponibilidade. Velocidade de consultas e nas tomadas de decisões, flexibilidade dos dados para geração de relatórios, volume de dados para resultados mais concretos e disponibilidade imediata da informação para toda a organização (SAP AG. 2010, p.5).

Segundo o *Overview* da SAP AG (2010, p.7), os softwares por ela produzidos realizam uma tarefa única no campo da computação *in-memory*, que é a combinação da análise das informações em conjunto com o emprego ativo das aplicações transicionais.

Com isso é possível observar um aprimoramento nos planejamentos, previsões e simulações de negócios através de análises de dados em tempo real e processamento de grandes massas de dados, permitindo melhorias nas pesquisas e nos armazenamentos de informação, além da centralização do gerenciamento e das operações. (SAP AG, 2010, p.7).

### 3.1 ARMAZENAMENTO DOS DADOS E AS PROPRIEDADES ACID

Em comparação com a maioria dos bancos de dados tradicionais, os bancos em memória têm seus dados armazenados permanentemente em memória principal. Estes dispositivos podem perder todas as suas informações armazenadas em caso de falha, faltas de energia ou desligamentos devido a alguma causa qualquer. Com isso o banco de IMDB se torna pouco eficiente na propriedade durabilidade dos dados, porém, atende todas as outras três propriedades que são atomicidade, consistência e isolamento (Sybase Inc. 2010).

A persistência primária dos dados é feita na memória principal, no entanto os IMDB's podem oferecer a durabilidade dos dados através de registros do estado do banco de dados em determinado momento guardando estes registros em discos de armazenagem não voláteis para a segurança dos dados, pois caso haja algum problema, não se corre o risco de perda das informações alocadas em memória (PLATTER, 2013, p.28).

Estes registros são gerados entre alguns períodos de tempo, caso ocorra um desligamento entre este espaço de tempo, os dados em memória serão perdidos e o sistema oferecerá uma durabilidade parcial dos dados recentes (SYBASE, 2010).

Ainda de acordo com a Sybase Inc. (2010), como forma de acrescentar garantias na durabilidade para os IMBD's, são necessárias execuções de algumas operações:

- Log de transação: a fim de registrar todas as alterações geradas no banco de dados, facilitando assim a recuperação automática do banco.
- A memória não volátil de acesso aleatório: comumente modelada com as características de uma memória RAM (*Random Access Memory*) estática, com a utilização de baterias de alimentação, ou uma memória ROM (*Read-Only Memory*) eletricamente apagável e programável (EEPROM). Restaurando assim o armazenamento de dados a partir de seu último estado consistente.
- Alta disponibilidade: Os dados são transferidos de servidor até outro através da replicação de dados, e este servidor salva os dados em um banco de dados convencional. Caso o servidor principal dê inesperadamente um problema, seja de *hardware* ou *software*, haverá sempre outro servidor recebendo os dados, dessa forma não existirá perda de dados.

Porém, a busca, a agregação ou a junção dos dados são realizadas em memória livrando o sistema de dificuldades que possam vir do processo de otimização do acesso ao disco (PLATTNER, 2013, p.29).

A fim de ser eficiente na utilização da memória principal como persistência primária, a organização dos dados é feita de forma diferente dos bancos tradicionais. A estrutura é arquitetada de forma colunar que mantém o armazenamento de cada coluna separada dentro do banco de dados, guardando sempre próximos os atributos pertencentes à mesma coluna. (SOARES, 2012. p.4).

O modelo colunar facilita a leitura dos dados onde há muitas entradas consecutivas de consulta em uma mesma coluna, o que é útil na agregação e varredura das colunas. (PLATTNER, 2013. P.29). A Figura 3 é uma comparação do modelo orientado a linhas com o modelo colunar.



Orientado a Linhas	Orientado a Colunas
Joao   2432.00   1988   Rio de Janeiro	Joao   Maria   Pedro   Jorge
Maria   2511.00   1986   São Paulo	2432.00   2511.00   3500.00   4200.00
Pedro   3500.00   1976   Mato Grosso	1988   1986   1976   1930
Jorge   4200.00   1930   Paraná	Rio de Janeiro   São Paulo   Mato Grosso   Paraná
(a)	(b)

**Figura 3 - Comparação do modelo Orientado a Linhas x Modelo Orientado a Coluna (In: SOARES, 2012. p. 4).**

As propriedades ACID no banco de dados em memória se assemelham em muito com as propriedades encontradas no banco de dados relacional, porém, existe uma diferença considerável na propriedade de durabilidade, pois como se trata de uma memória volátil o armazenamento possui um limite de tempo. Grandes empresas que fazem utilização dessa forma de armazenamento, mantem alocada a informação por um período de aproximadamente três meses, prazo esse geralmente pautado nas necessidades que a empresa requer (SYBASE INC, 2010).

### 3.2 ORGANIZAÇÃO DOS DADOS

A organização dos dados na memória é alocada de uma forma diferente dos bancos tradicionais, com o intuito de otimizar consultas, inclusões e qualquer outro tipo de alterações.

Um banco de dados relacional tem uma estrutura bidimensional para organizar seus dados, porém na memória principal os dados são organizados de forma unidimensional, criando endereços de memória começados em zero aumentando até seu limite final. (PLATTNER, 2013. p.55).

### 3.3 COMPRESSÃO DOS DADOS

Os bancos de dados em memória estão cada vez mais acessíveis por causa da redução dos custos da memória. Porém, ainda assim, os servidores de banco de dados totalmente em memória precisam de muito investimento em hardware, então como forma de melhorar a capacidade do armazenamento sem precisar de mais investimentos, os desenvolvedores criaram uma forma de comprimir os dados.

Os mais modernos mecanismos de armazenamento em memória faz a utilização de técnicas de compressão dos dados no início do dicionário de codificação, reduzindo assim os requisitos de memória. (PLATTNER. 2013. p.43).

#### 3.3.1 *Cluster encoding*

O grupo de codificação funciona com colunas de blocos de tamanho igual. O vetor é dividido em blocos de tamanho  $N$  fixo, normalmente com tamanhos de 1024. Se um grupo tem apenas um único valor ele é trocado por uma ocorrência deste valor. Caso contrário se não houver nenhuma ocorrência o grupo permanecerá descompactado. Um vetor adicional de bits de comprimento  $N$  cria um índice dos blocos que tenha sido substituído por um único valor, 1 para os que foram substituídos e 0 para os que permanecerem sem ocorrências. (PLATTNER, 2013. p.46).

#### 3.3.2 *Run lenght encoding*

A técnica de compressão *Run Lenght* é mais eficaz quando o atributo vetor tem poucos valores distintos, porém com grande número de ocorrências. A fim de obter a máxima compressão a coluna tem de ser resolvida, de modo que todos os valores situem o mesmo conjunto. Nesta codificação as sequências de valores iguais são substituídas por uma única instância do valor, ou por seu número de ocorrências ou

sua posição como compensação. Armazenando sua posição acelera o acesso, pois o endereço de um valor específico pode ser lido diretamente na coluna, ao invés de realizar cálculos a partir do endereço inicial. Proporcionando um acesso direto ao dado. (PLATTNER, 2013, p.45)

### **3.3.3 Prefix encoding**

Segundo o PLATTNER (2013, p.43), é com muita frequência que nos bancos de dados se encontra situações em que em uma coluna tem um valor predominante e os demais valores têm pouca redundância. Isso é como fazer o armazenamento do mesmo valor muitas vezes de forma não comprimida. Para este caso a codificação de prefixo é a forma mais simples e eficiente. A aplicação desta codificação consiste em classificar os conjuntos de dados pela coluna com o valor predominante. O atributo vetor tem que começar com este mesmo valor predominante. A compressão da coluna não armazena o valor predominante a cada vez que ocorre, apenas guarda a quantidade de suas ocorrências e uma única vez o valor no atributo vetor. Com isso o prefixo codificado guarda três informações essenciais para sua busca, que são:

- Número de Ocorrências do valor que predomina;
- Valor do ID do valor predominante;
- Valor do ID dos valores restantes.

## **3.4 OPERAÇÕES ESSENCIAIS REALIZADAS NO BANCO DE DADOS *IN-MEMORY***

Esta seção descreverá as principais funções SQL (*Structured Query Language*) executadas no Banco de Dados *In-memory*.

### **3.4.1 Função *delete***

Segundo Coronel e Rob (2011, p.241) a função *delete* é responsável pela exclusão de um registro em uma tabela de banco de dados.

Plattner (2013, p.71) escreve que tal função em um banco de dados em memória pode ser classificada em duas tipagens: física e lógica.

Quando se opta por realizar uma exclusão do tipo física, a informação que está armazenada no banco é completamente removida do sistema, ficando inacessível. Já a exclusão do tipo lógica é caracterizada pela remoção da visibilidade do arquivo, a tarefa remove a referência, mas mantém o dado no banco de informações, onde, por meio de consultas específicas, pode-se obter seu valor (PLATTNER, 2013, p. 71).

Plattner (2013, p.71) elucida a exclusão física da seguinte forma: todas as pessoas com o nome 'Jane Doe' devem ser removidas das tabelas onde contém o nome e sobrenome, utilizando o dicionário de codificações aplicadas. As tabelas a seguir evidenciam as duas tabelas de dicionário e dois vetores de valores.

Dicionário "fnome"		Atributo Vetor "fnome"		Dicionário "Inome"		Atributo Vetor "Inome"	
ValorID	Valor	RecID	ValorID	ValorID	Valor	RecID	ValorID
...	...	...	...	...	...	...	...
22	Andrew	38	22	17	Brown	38	19
23	Jane	39	24	18	Doe	39	21
24	John	40	25	19	Miller	40	17
25	Mary	41	23	20	Schmidt	41	18
26	Peter	42	24	21	Smith	42	18
...	...	43	26	...	...	43	20
		...	...			...	...

**Figura 4 - Amostra das colunas em memória com dicionário e atributo vetor (PLATTNER, 2013, p.71).**

Primeiro é feita a busca pelo dicionário onde o valor correspondente a 'Jane' no dicionário de nomes retorna um ValorID igual a 23, e no segundo dicionário de

sobrenomes o valorID 18 corresponde ao sobrenome 'Doe' (PLATTNER, 2013, p.71).

Dicionário "fnome"		Atributo Vetor "fnome"		Dicionário "Inome"		Atributo Vetor "Inome"	
ValorID	Valor	RecID	ValorID	ValorID	Valor	RecID	ValorID
...	...	...	...	...	...	...	...
22	Andrew	38	22	17	Brown	38	19
23	Jane	39	24	18	Doe	39	21
24	John	40	25	19	Miller	40	17
25	Mary	41	23	20	Schmidt	41	18
26	Peter	42	24	21	Smith	42	18
...	...	43	26	...	...	43	20
		...	...			...	...

**Figura 5 - Resultado da busca pelo nome e sobrenome dos dados a serem excluídos (PLATTNER, 2013, p.72).**

O próximo passo é fazer a busca no Vetor de Atributos buscando a onde estes dados estão gravados pelo ValorID encontrado no dicionário. Neste caso só existe uma *tupla* que corresponde à combinação do nome ao sobrenome. (PLATTNER, 2013, p.72).

Dicionário "fnome"		Atributo Vetor "fnome"		Dicionário "Inome"		Atributo Vetor "Inome"	
ValorID	Valor	RecID	ValorID	ValorID	Valor	RecID	ValorID
...	...	...	...	...	...	...	...
22	Andrew	38	22	17	Brown	38	19
23	Jane	39	24	18	Doe	39	21
24	John	40	25	19	Miller	40	17
25	Mary	41	23	20	Schmidt	41	18
26	Peter	42	24	21	Smith	42	18
...	...	43	26	...	...	43	20
		...	...			...	...

**Figura 6 – Busca dos dados para exclusão no atributo vetor (PLATTNER, 2013, p.72).**

Após a exclusão dos dados, são feitos os ajustes para que todos os espaços em memória sejam preenchidos a fim de não ter espaços desperdiçados.

Dicionário "fnome"				Dicionário "Inome"			
ValorID	Valor	ReclID	ValorID	ValorID	Valor	ReclID	ValorID
...	...	...	...	...	...	...	...
22	Andrew	38	22	17	Brown	38	19
23	Jane	39	24	18	Doe	39	21
24	John	40	25	19	Miller	40	17
25	Mary	<del>41</del>	<del>23</del>	20	Schmidt	<del>41</del>	<del>18</del>
26	Peter	41	24	21	Smith	41	18
...	...	42	26	...	...	42	20
...	...	...	...	...	...	...	...

Figura 7 – Realocação dos dados no atributo vetor (PLATTNER, 2013, p.72).

### 3.4.2 Função *insert*

De acordo com Coronel e Rob (2011, p.241) a função *insert* é a operação que realiza a inclusão de registros em uma tabela de banco de dados.

A função de inserção em um banco de dados em memória é mais complexa em comparação aos modelos de dados relacionais, a complexidade está na sua estrutura de colunas (PLATTNER, 2013, p.75).

Em uma inserção no modelo baseado em linhas, os dados simplesmente são armazenados na última posição da tabela. Porém, ao ser inserida uma nova *tupla* no modelo colunar torna-se necessária à criação de uma nova entrada em cada coluna da tabela, onde cada coluna é composta por um dicionário e um vetor de atributos. A cada nova entrada é feita uma verificação da necessidade de inserção do novo valor

na coluna do dicionário, caso não exista o valor, o mesmo é acrescido tanto no dicionário quanto no vetor de atributos (PLATTNER, 2013, p.75).

A inserção segundo Plattner (2013, p. 76) é exemplificada conforme a ilustração da Figura 8. A figura representa um cadastro da população mundial, sendo a coluna “lname” a representação do sobrenome e a coluna “fname” o nome.

Example Table: world\_population

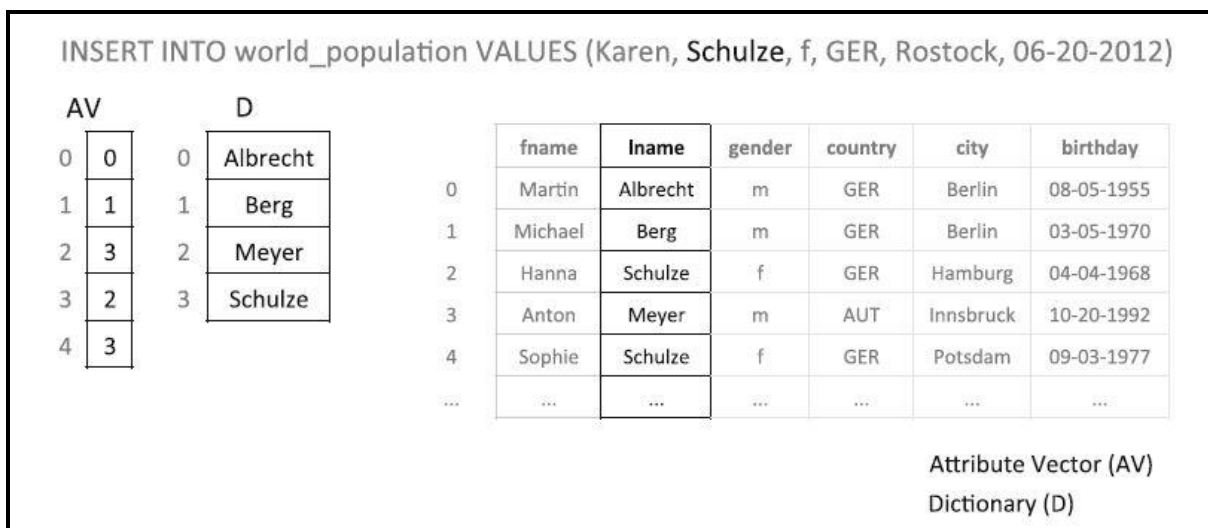
recID	fname	lname	gender	country	city	birthday
0	Martin	Albrecht	m	GER	Berlin	08-05-1955
1	Michael	Berg	m	GER	Berlin	03-05-1970
2	Hanna	Schulze	f	GER	Hamburg	04-04-1968
3	Anton	Meyer	m	AUT	Innsbruck	10-20-1992
4	Sophie	Schulze	f	GER	Potsdam	09-03-1977
...	...	...	...	...	...	...

INSERT INTO world\_population  
VALUES (Karen, Schulze, f, GER, Rostock, 06-20-2012)

**Figura 8 - Tabela de exemplo população mundial (PLATTNER, 2013, p. 76).**

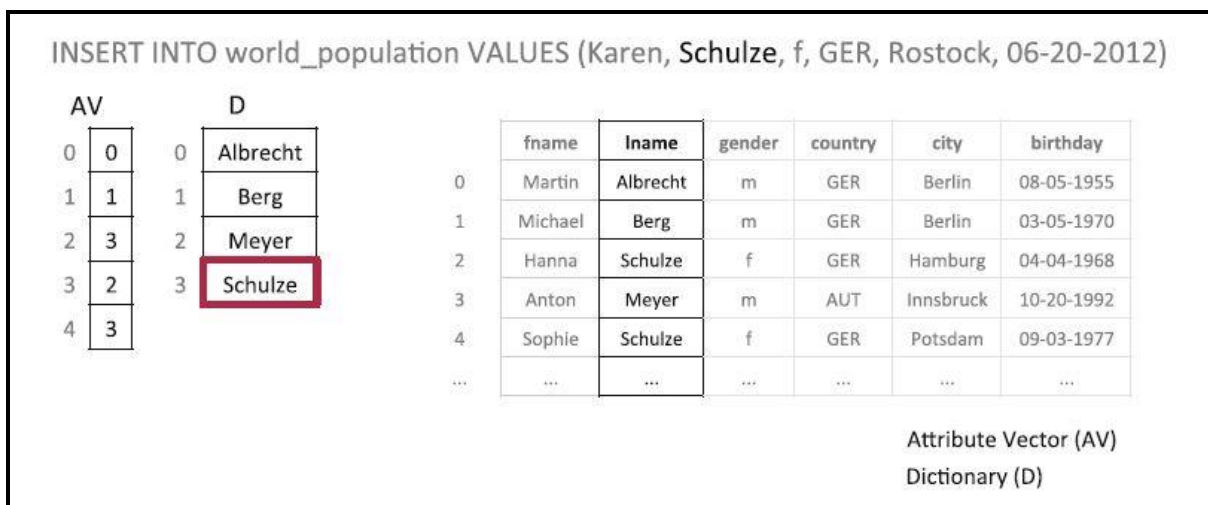
As figuras 8, 9 e 10 nos trazem uma amostragem de como são feitas inserções sem que sejam necessárias alterações no vetor de atributo e no dicionário de dados.

A primeira verificação feita é da coluna “lname”, nesta coluna é feita a verificação do valor Schulze.



**Figura 9 – Exemplo da análise no dicionário antes de uma nova inserção (PLATTNER, 2013, p. 77).**

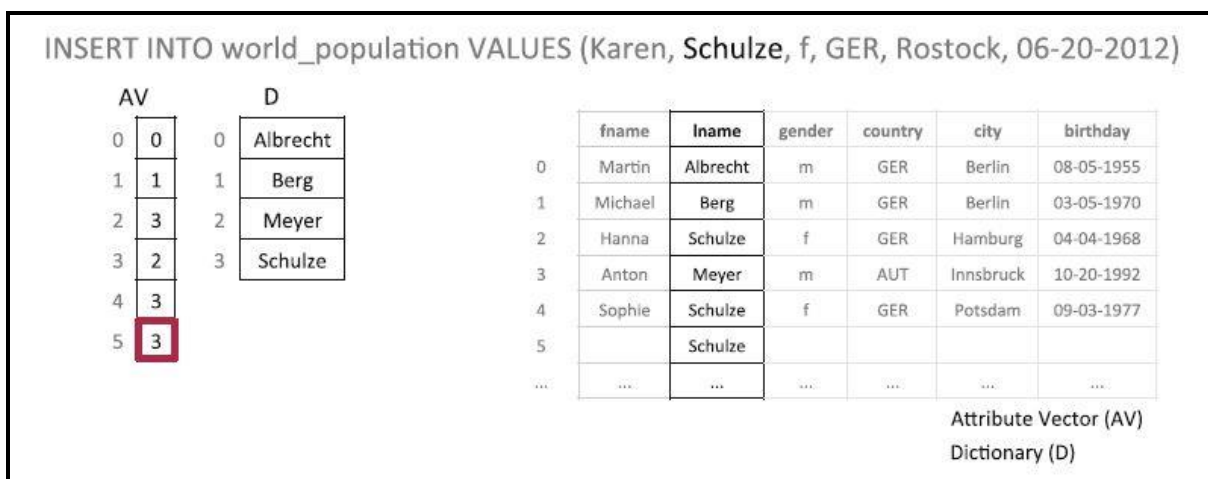
Como já existe esse dado na tabela não será necessária uma entrada de dados no dicionário de dados.



**Figura 10 – Exemplo do valor encontrado no dicionário (PLATTNER, 2013, p. 77).**

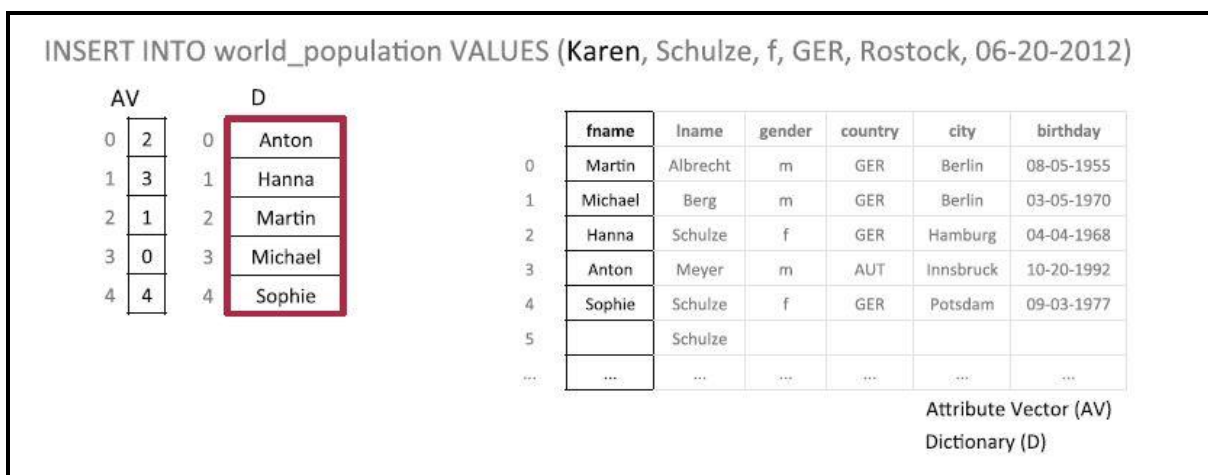
O dado encontrado no dicionário está na terceira posição, sendo assim, o vetor de atributos recebe em sua última posição apenas o número da posição onde já existe um dado igual.





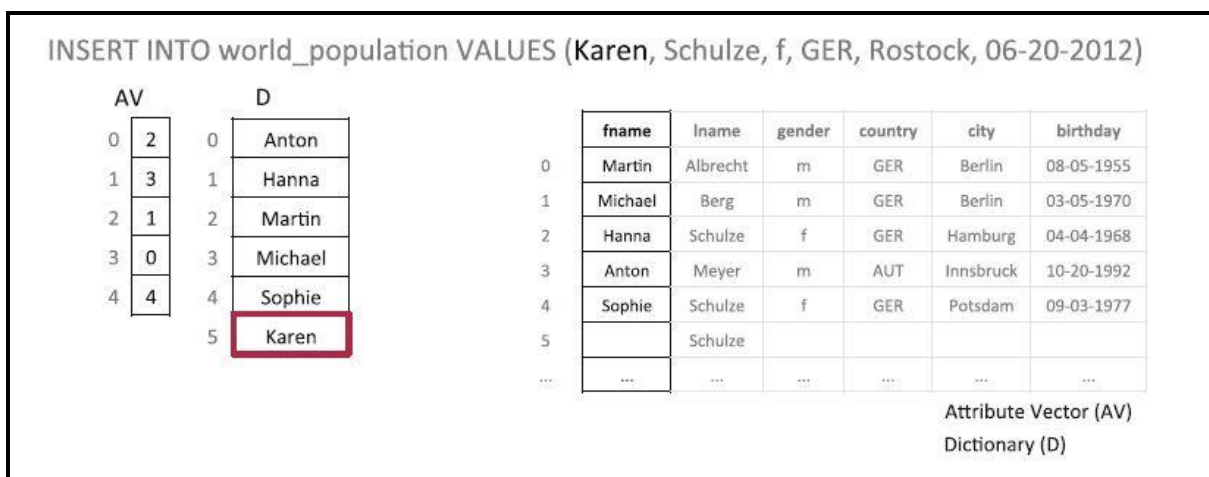
**Figura 11 – Exemplo do dado gravado no vetor de atributos (PLATTNER, 2013, p. 77).**

Na Figura 12, como exemplificada por Plattner (2013, p.78), a segunda parte da inserção é feita na coluna nome.



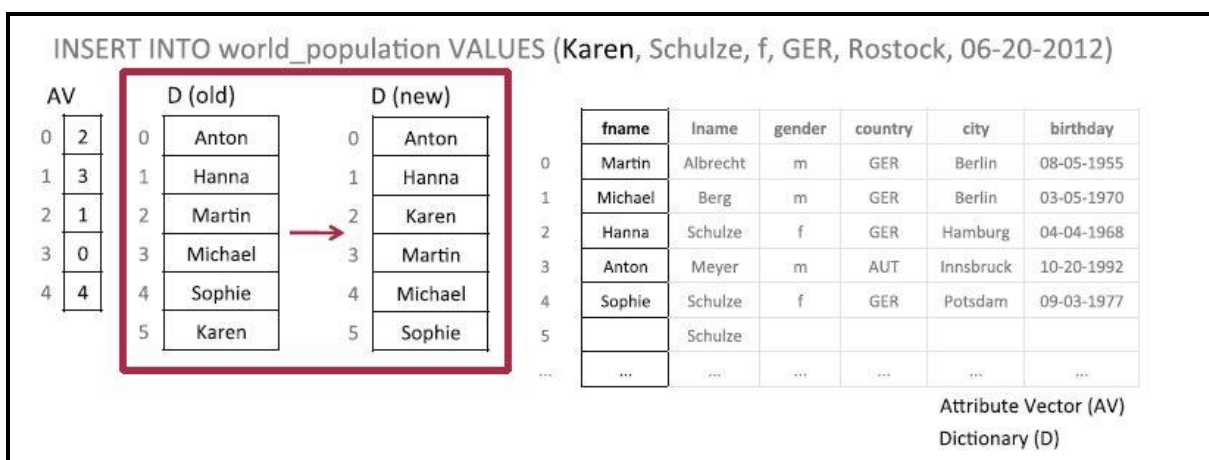
**Figura 12 – Exemplo da leitura do dicionário de nomes (PLATTNER, 2013, p. 78).**

Como mostrado abaixo não existe nenhum valor com o dado Karen, neste momento é inserido no dicionário em sua primeira posição disponível.



**Figura 13 – Exemplo de inserção no dicionário de nomes (PLATTNER, 2013, p. 78).**

O dicionário de dados precisa continuar ordenado, para isso é reordenado cada campo. Como mostrado abaixo:



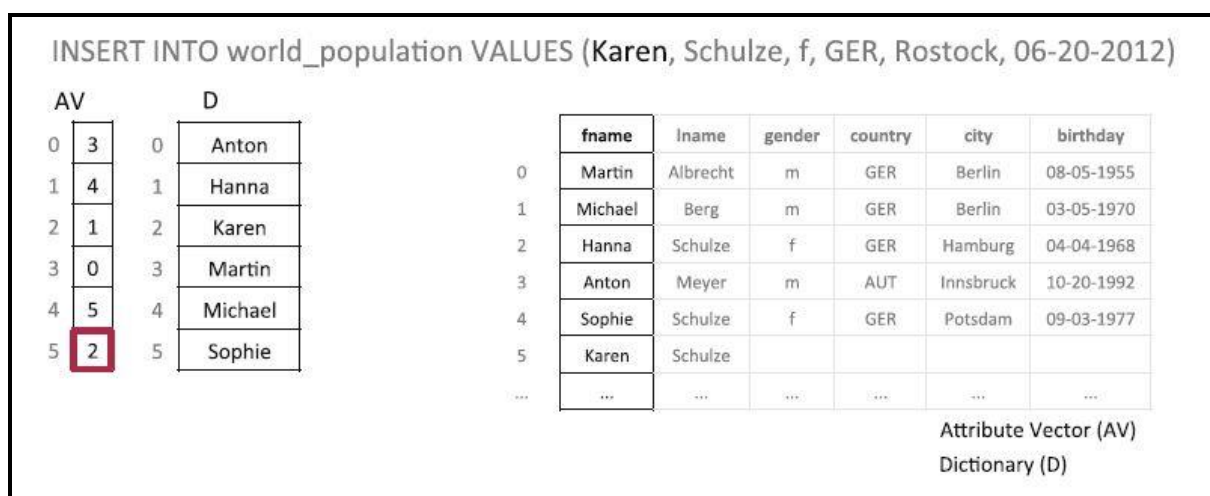
**Figura 14 – Exemplo ordenação do dicionário de nomes (PLATTNER, 2013, p. 78).**

Após a reordenação do dicionário de nomes, o novo valor é atribuído ao vetor de atributos, este também será reordenado a fim de que seus valores correspondam exatamente ao dicionário.



**Figura 15 – Exemplo da ordenação do vetor de atributos (PLATTNER, 2013, p. 79).**

E por fim os dados são inseridos e ordenados no dicionário de nomes e no vetor de atributos.



**Figura 16 – Exemplo da tabela com os dados inseridos e ordenados (PLATTNER, 2013, p. 77).**

### 3.4.3 Função *select*

Segundo Coronel e Rob (2011, p.241) a função *select* realiza a seleção de registros de uma ou mais tabelas em um banco de dados.

A função *select* em um banco de dados em memória apresenta uma declaração de *Structured Query Language* (SQL) específica do resultado que deseja receber do banco de dados, contudo para executar a SQL é necessário executar vários processos a fim de extrair os dados do banco, estas etapas são chamadas de plano de execução (PLATTNER, 2013, p. 100).

A cada consulta SQL são desencadeadas várias tarefas de execução para obter um mesmo resultado, porém com comportamentos diferentes. Para realizar o cálculo do custo de diferentes planos são utilizados otimizadores. Usando diferentes modelos de custo e heurística consegue-se alcançar um melhor plano de execução. Este processo é necessário para reduzir os conjuntos de resultados (PLATTNER, 2013, p. 100).

Usando a tabela população mundial Plattner (2013, p.101) exemplifica a forma de como se processa uma consulta SQL usando *select*, tal tarefa fica elucidada na Figura17.

id	fname	lname	country	gender
2394	Gianluigi	Buffon	Italy	m
3010	Lena	Gercke	Germany	f
3040	Mario	Balotelli	Italy	m
3949	Manuel	Neuer	Germany	m
4902	Lukas	Podolski	Germany	m
20102	Klaas-Jan	Huntelaar	Netherlands	m

**Figura 17 – Tabela população mundial (PLATTNER, 2013, p.101).**

Utilizado a seguinte instrução:

```
SELECT fname,lname
FROM word_population
WHERE country = 'Italy' AND gender = 'M';
```

Representando figurativamente o plano de execução pela Figura 18,

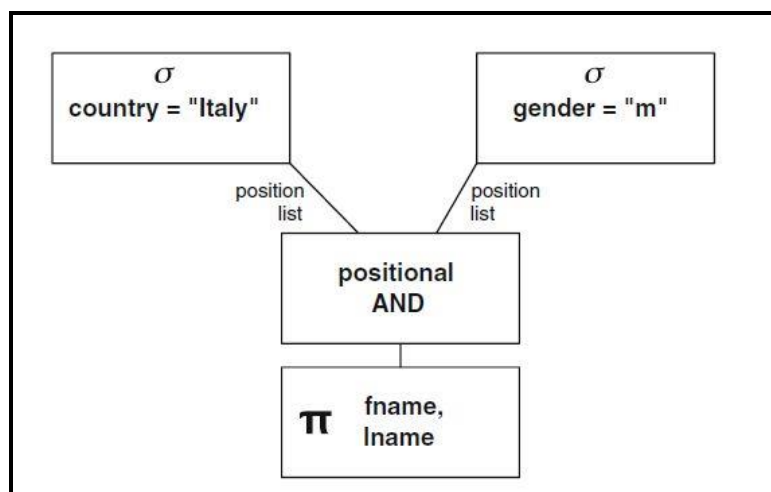


Figura 18 – Representação do plano de execução (PLATTNER, 2013, p.101).

O *script* da SQL tem por finalidade fazer a busca no banco de dados de todas as informações que são de origem italiana e do sexo masculino. No caso desta consulta são utilizados apenas estes dois parâmetros como especificação, a nacionalidade e o sexo. O *In-Memory* executa tais instruções de forma simultânea, otimizando a velocidade da consulta (PLATTNER, 2013, p. 101).

A Figura 19 explica os resultados da consulta citada acima.

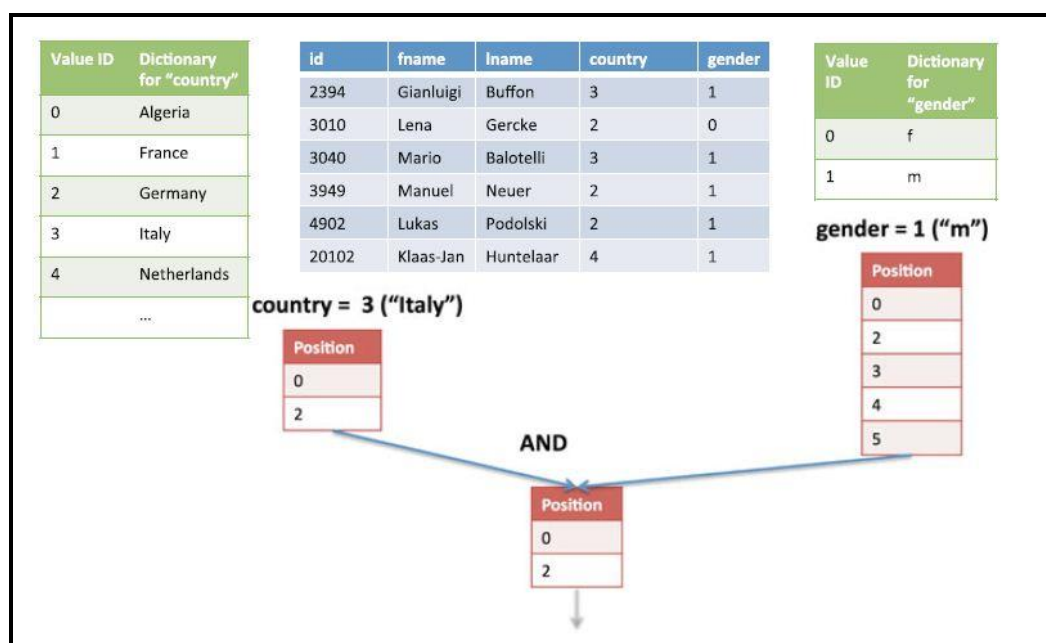


Figura 19 – Amostra do resultado de *select* (PLATTNER, 2013, p.102).

### 3.5 EXEMPLOS DO USO DO *IN-MEMORY*

O *In-Memory* está presente em muitos ambientes que possuem a necessidade de manipular grandes volumes de dados de forma simultaneamente e em tempo real.

Comumente os dados gerados por um evento são pequenos e aparentemente sem importância, vistos de forma individual. Como por exemplo, uma ordem de venda de um único produto em uma grande rede de supermercados. Porém, se a quantidade de vendas de um mesmo produto para indivíduos distintos se torna frequente, o processo fica demorado e se torna extenso (PLATTNER, 2013, p. 8).

#### 3.5.1 Sensores de dados

Os sensores atualmente são usados para fazer a supervisão de vários sistemas. Um exemplo de sua utilização no banco de dados em memória são os produtos farmacêuticos, roupas ou peças de reposição. Estes itens são identificados por radiofrequência ou códigos de barras onde cada produto tem um código único. Esta identificação carrega consigo dados como, o fabricante do produto, número da série, dentre outros. Como um produto pode ser envolvido em vários processos, ele acaba gerando várias leituras (nos processos de descompactação, reembalagem, recebimento ou envio). Todos os dados são enviados de eventos descentralizados, mas referentes ao mesmo produto gerando assim, uma grande série de informações processadas pelo sistema de banco de dados (PLATTNER, 2013, p. 8).

Outro exemplo é o rastreamento de produtos farmacêuticos, na Europa são expedidos por volta de 15 bilhões de produtos neste seguimento. Este acompanhamento resulta em aproximadamente 8000 (oito mil) leituras de eventos por segundo. As leituras desses eventos são utilizadas para impedir a falsificação do medicamento. Através dos dados é possível reconstituir todo o trajeto realizado pelo item. Com o *In-Memory* são rastreados 10 milhões de eventos em menos de 100ms (PLATTNER, 2013, p. 8).

Outro emprego da tecnologia *In-Memory* é realizado pela Fórmula 1, onde os carros são equipados com vários sensores espalhados em quase todas as peças do veículo, chegando em, aproximadamente, 600 (seiscentos) sensores por carro. Com isso são gerados *Gigabytes* ou até mesmo *terabytes* de informações a serem processadas e analisadas em tempo real durante toda a corrida para melhorar o desempenho dos carros, evitar falhas e até mesmo reduzir o consumo do combustível (PLATTNER, 2013, p. 8).

### 3.5.2 Análise de eventos de jogos

Os jogos *online* se tornaram populares em todo o mundo. Este fato fez com que aumentasse as empresas de construção de games. A Bigpoint, empresa alemã é fornecedora de jogos *online* e possuem mais de 200 milhões de usuários ativos. Em sua página web, os jogos *online* chegam a gerar um fluxo constante de mais de 10000 (dez mil) eventos por segundo. Os eventos mais comuns *em games* online são os dados do nível atual do jogador, os bens adquiridos, tempo gasto, dentre outros. Uma empresa como a Bigpoint acompanha cerca de 800 (oitocentos) milhões de eventos ao dia. Um banco de dados transacional não corresponde a um nível de dados grande como esse de forma iterativa, pois necessita de varreduras nas tabelas que precisam de índices complexos e armazenamento otimizado a fim de retornar dados de forma muito rápida (PLATTNER, 2013, p.9).

Em um determinado momento do jogo o usuário se vê na condição de desembolsar um determinado valor para avançar de nível ou até mesmo dar continuidade na utilização do produto. Este processo engloba várias atividades que necessitam de execução simultânea, como por exemplo, o acompanhamento do tempo que o jogador está conectado; a atualização dos valores de cada produto ofertado durante o jogo; o saldo que o cliente possui em sua conta; dentre outros. Dessa forma, tornando-se necessária a utilização de um banco de dados em memória, devido sua capacidade de manipulação simultânea e seu rápido tempo de resposta. Os testes são feitos com pequenos grupos de usuários, caso a análise seja positiva os descontos são aplicados a um grupo maior de usuário, porém se não for aprovado

são feitos novos descontos e testados da mesma forma das promoções antigas (PLATTNER, 2013, p.9).

### 3.6 PROPRIEDADES ACID: BANCO DE DADOS CONVENCIONAIS X BANCO DE DADOS EM MEMÓRIA

#### 3.6.1 PROPRIEDADES ACID EM BANCO DE DADOS CONVENCIONAIS

Após o relato do tipo de armazenamento, propriedades ACID e estrutura dos bancos de dados tanto relacionais como em memória, esta seção confronta os pontos que podem fazer a diferença entre o uso destes sistemas de banco de dados.

Os modelos de dados relacionais por toda sua evolução com o passar dos anos, ainda será utilizado por muito tempo. A sua fama se dá pelo benefício trazido pelas suas propriedades ACID, garantido a integridade e consistência dos dados, a sua força de mercado vem da utilização em muitas corporações que fazem o uso de sistemas de ERP, esses sistemas em sua grande maioria fazem o acesso a grandes bases de dados centralizadas em “Data Centers” gerando dezenas e/ou centenas de *Terabytes* de armazenamento.

Para as grandes corporações que tem a segurança de uma consagrada tecnologia em pleno funcionamento se torna inviável investimento em novas tecnologias emergentes no mercado, pois a casa já tem novos recursos de utilização e melhoria de desempenho são lançados visando os acessos aos dados e tendo grandes possibilidades de recuperação de dados apagados ou sobrescritos.

Como exemplo de empresa que trás soluções incorporadas aos SGBD tradicionais está a *Oracle*, com um nível de maturidade muito grande pelo tempo em que se encontra na área de Sistemas de Gerenciamento de Banco de Dados. A *Oracle* tem uma solução de banco de dados em memória, o *Oracle TimesTen In-Memory*. Através da compra de uma empresa pioneira no segmento de banco de dados em memória a *TimesTen*, a *Oracle* começou a desenvolver novas soluções incorporadas ao seu SGBD Relacional.



O *Oracle TimesTen In-Memory* é um Banco de Dados Relacional que pode processar tanto OLTP e cargas de trabalho analíticas, o que beneficia aos usuários a utilizar toda capacidade de memória RAM disponível nos seus servidores. Com isso o tempo de resposta é extremamente rápido pela ausência de entrada e saída de dados a partir do disco rígido, transações analíticas complexas são processadas em milissegundos.

Mesmo estando em memória o servidor é compatível com instruções de transações SQL, PL/SQL entre outras para compatibilidade com o Banco de Dados *Oracle*. O servidor tem serviços para sincronização automática dos dados com o sistema *Oracle Database* principal, em caso de perda de conexão o *Oracle Cache In-Memory* irá continuar a servir a aplicativo e sincronizar os assim que a conexão for reestabelecida. Além de trabalhar com aplicações existentes que possibilita alterações mínimas para acelerar as aplicações.

### 3.6.2 PROPRIEDADES ACID EM BANCO DE DADOS EM MEMÓRIA

A utilização dos Banco de Dados *In-Memory*, para substituir os Sistemas de Gerenciamento de Banco de Dados convencionais está muito longe de acontecer, pois essa tecnologia é nova e está em crescimento, há poucos estudos voltados para as empresas neste segmento de Banco de Dados. Os setores de Tecnologia da Informação (TI) não sentem ainda confiança nesta ferramenta devido a falta de avaliações e por se tratar de uma nova tecnologia.

O que ainda deixa a desejar é com relação a garantia da propriedade ACID “Durabilidade”, uma das propriedades do ACID, requisito básico para a confiabilidade e integridade nos Sistema de Bancos de Dados. Este requisito “Durabilidade” tem a funcionalidade de garantir a realização das gravações de em disco não voláteis reduzindo o risco de perda dos dados nos bancos de dados.

Como os dados nos são armazenados em memória volátil e ficam disponíveis por um longo período de tempo aumenta o risco da perda destes dados em caso de

falha ou desligamentos repentinos se comparado aos Bancos de Dados Convencionais.

Mas como a memória RAM perde seu conteúdo em caso de falhas, para evitar as perdas de dados são necessários envios constantes de dos dados na memória para os discos físicos, assim o sistema pode ficar lento indo contra a sua funcionalidade de reduzir as entradas e saídas entre disco e memória principal, tornando-o muito mais rápido.

Outro fator que dificulta o avanço dos Bancos de Dados em Memória é que todo o banco esta alocado em memória principal, os Sistemas ERP geram muitos dados, um servidor de banco de dados em memória necessitaria de um investimento muito alto com o hardware e principalmente com memórias RAM para subir todo o banco de dados.

De acordo com Guimarães (2012), os desafios da expansão dos dados têm como base as seguintes perspectivas:

- Velocidade: o conteúdo digital em todo o mundo duplicará em 18 meses, e a cada 18 meses depois disso.
- Volume: em 2005, a humanidade criou 150 Exabytes de informação. Em 2011/2012, serão criados 1200 Exabytes.
- Variedade: 80% dos dados corporativos serão não estruturados abrangendo fontes tradicionais e não tradicionais.

Em suma, essa não garantia da Propriedade “Durabilidade” faz com que o uso do Banco de Dados em Memória não desponte, porém, essa “falha” deverá ser resolvida muito em breve.

## 4. ESTUDOS DE CASOS: EXEMPLOS PRÁTICOS

O presente capítulo apresenta algumas grandes empresas que utilizam o *In-Memory*, como forma de atenderem ao mercado de forma eficiente evitando a perda de recursos e otimizando seus serviços.

### 4.1. SAMSUNG APPS

A *Samsung* é uma das líderes na atualidade na área de *Smartphones*, *tablets* e televisões inteligentes, dentre outros produtos nas mais diversas linhas tecnológicas (*ALTIBASE*).

Seu problema surge ao fornecer as autenticações de seus serviços no *Samsung Apps*, tanto para smartphones que para as TV's. Os serviços usados na nuvem através da *Amazon Elastic Computing 2* e *Amazon Web Services* não recebiam as autenticações para os usuários de maneira eficiente causando a falta de estabilidade necessária para suprir toda a sua demanda, este problema impossibilitou a *Samsung* de entrar por um tempo em novos mercados. Isso prejudicou a aquisição de novos clientes e até mesmo a manutenção dos atuais clientes (*ALTIBASE*).

A sobrecarga no serviço de autenticação global travava o sistema por várias horas, isso prejudicava em uma média de 100 milhões de usuários. Os desgastes causados pelo problema fez com que a empresa perdesse dados críticos, e um dano irreparável (*ALTIBASE*).

Com esse problema a *Samsung Eletronics* foi atrás de um serviço que lhe desse alto desempenho para o serviço de autenticação global de forma confiável e rápida. Essa necessidade levou a empresa a utilizar uma ferramenta de alta performance (em memória) para os dados e armazenar grandes volumes de dados em bases históricas (discos rígidos) (*ALTIBASE*).

A solução usada pela *Samsung Eletronics* foi uma arquitetura híbrida, neste caso o *Altibase HDB*, permitindo assim, que os dados recentes ficassem todo em memória principal e os dados já utilizados armazenados em uma base no disco. Mantendo

sua segurança em caso de eventuais problemas evitando a perda de dados críticos (*ALTIBASE*).

## 4.2. SK TELECOM

A *SK Telecom* é uma operadora sul-coreana de telefonia móvel. Ela possui mais da metade dos usuários de telefonia no mercado asiático, cerca de 50,6% isso representa mais de 50 milhões de assinantes de seus serviços (*ALTIBASE*).

Usando uma infraestrutura legada, baseada nos mainframes de Classificação Real (RTRS), permitia apenas processos de faturamento e classificação em modo *batch* (*windows*) o que resultava em números de transações incontroláveis no banco de dados, com o aumento no número de assinantes em novos serviços, como o pré-pago, a empresa sentiu limitações devido aos custos de manutenções e atualizações nos serviços de faturamento e outras áreas, tornando o custo total de propriedade (TCO) incontrolável (*ALTIBASE*).

Isso acarretou em falta de desligamento nos serviços pré-pagos depois de esgotado os saldos carregados pelos clientes. Com a limitação na precisão da identificação dos saldos de seus usuários, a empresa se viu com saldos negativos e fundos não cobrados (*ALTIBASE*).

A solução para a *SK Telecom* foi utilizar um banco de dados em memória, proporcionando o máximo desempenho e funcionalidade reduzindo significamente o TCO. Usando uma combinação do *Altibase HDB* e instâncias do *ORACLE DBMS* a *SK Telecom* projetou um sistema de RTRS. O *Altibase* implantado em modo DBMS em memória hospedando dados recentes de até três meses e o SGBD *Oracle* como repositório para os dados históricos. Com este sistema a taxa de transferência passou a ser mais de 450 milhões por dia utilizando apenas 50% dos recursos do sistema (*ALTIBASE*).

Com o sistema de banco de dados em memória a *SK Telecom* conseguiu um aumento de desempenho de 5,3 vezes sobre o sistema legado, hoje ela acompanha com precisão os saldos de seus clientes. Tendo também capacidade de processar o

uso de seus usuários com o intuito de melhorar a comercialização de novos serviços, essa capacidade de processamento gera uma média de 14,5 milhões de dólares por ano (*ALTIBASE*).

#### 4.3 LEADING JAPANESE INVESTMENT BANK - THE FIRM

A empresa *The Firm* como é conhecida é um banco de investimentos, voltado para fazer investigação de novos comércios, tecnologia de ponta e soluções financeiras para seus clientes (*ALTIBASE*).

No final do ano de 2010 a *The Firm* teve problemas com seu sistema de *Foreing Exchange* (FX) que é o serviço de mercado de câmbio entre moedas. Este é um mercado descentralizado, neste mercado fazem parte os maiores centros financeiros do mundo com a função de realizar negociações entre diferentes tipos compradores e vendedores ao redor do mundo. Este serviço do FX que determina os valores relativos entre diferentes tipos de moedas tornando possível que o banco consiga gerar margens de lucro para seus investimentos (*ALTIBASE*).

Porém, a *The Firm* não foi bem estruturada para realizar negociações eficientes com o serviço FX, o que causou um mau planejamento nas suas margens de lucros (*ALTIBASE*).

A fim de maximizar suas negociações a *The Firm* viu a necessidade de novas tecnologias na utilização do serviço do FX (*ALTIBASE*).

A solução mais eficiente, que não causaria nenhum atraso com os dados levantados de maneira confiável, com alto desempenho e que pudesse ser implantado de forma rápida foi o banco de dados em memória. No começo de 2011 o projeto já havia sido concluído com uma solução personalizada da forma que solucionasse suas necessidades e a empresa começou a ter uma rápida evolução em relação aos serviços utilizados, o serviço do FX da *The Firm* ficou sendo o mais avançado de todo o mundo (*ALTIBASE*).

Como resultado da implantação deste banco de dados consegue capitalizar as margens de negociação e aumentar sua competitividade.

Também conseguindo processar mais 1000 operações de câmbio por segundo, lidando com dispositivos *mobile*, *Web* e aplicativos *rich client* que são soluções de processamento de pedidos otimizados.

## 5. CONCLUSÃO

O tema apresentado neste trabalho já é conhecido por grandes empresas, porém falta ainda material literário além de estudos deste novo seguimento. Muitas informações são encontradas espalhadas, mas não são profundas o suficiente para a realização de trabalhos.

O presente estudo sobre a tecnologia *In-Memory* apresentou a estrutura de um *In-Memory Database* (IMBD), exemplificando suas principais instruções de utilização, sua forma de manipular e expor os dados na memória principal além de sua estruturação, visando realizar um desempenho significativo em relação aos bancos convencionais que alocam suas informações em discos.

Foi descrito também sobre os sistemas de banco de dados relacional fazendo uma breve comparação entre suas estruturas de armazenamento de dados e a garantia das propriedades ACID.

Como prova deste fato, os estudos de casos demonstrados comprovam a eficiência do banco de dados em memória ao necessitar de análises de dados com gigantescos volumes de dados sem que haja travamentos causados pelos sistemas de dados antigos. Porém ainda os sistemas de banco de dados em memória não substituirá tão cedo o banco de dados relacional, porém, pela sua capacidade de respostas em tempo real com grandes volumes dados sendo carregados em seu sistema poderá no futuro ser o banco de dados mais utilizado. Isto pode ser viabilizado também por conta dos *Hardwares* que tem se avançado tecnologicamente e diminuído seu custo, facilitando o acesso a servidores mais potentes.

### 5.1 TRABALHOS FUTUROS

Como possíveis trabalhos futuros, pode ser realizado o processo de instalação com a finalidade de testar o desempenho dos bancos de dados em memória traçando um comparativo com o desempenho apresentado nos Sistemas de Bancos de Dados Convencionais..

Também se pode fazer uma amostragem do processo de implantação de um IMDB, relatando assim suas configurações principais e até mesmo confrontando as informações com outros tipos de implantações de banco de dados.



## REFERÊNCIAS

SAP AG. **Overview of SAP's In-Memory Computing Strategy and Applications** 2010.

ALTIBASE, Disponível em <<http://www.altibase.com/case-studies>>. Acesso em 4 de nov de 2013.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. 5. ed. Tradução de Marília G. Pinheiro; Cláudio C. Canhette; Glenda C. V. Melo; Claudia V. Amadeu; Rinaldo M. de Moraes. São Paulo: Editora PEARSON. 2005.

FRENKIEL, Eric. Na Interview MemSQL CEO Eric Frenkiel: Contexto n some amazing third-party benchmark results. Disponível em: <<http://ctoivision.com/2013/08/interview-memsql-ceo-eric-frenkiel-context-amazing-third-party-benchmark-results/>>. Acesso em 14 de mar, 2013.

GUIMARÃES, Carlos. *SAP HANA - Empresas e Negócios em Tempo Real*. Setor de Database & Technology, março de 2012.

ISSA, Felipe Gustavo de Sousa. **Estudo comparativo entre Banco de Dados Relacionais e Banco de Dados NoSQL na utilização por aplicações de Business Intelligence**. 2011. 62p.. Trabalho de Conclusão de Curso (Técnico em Processamento de Dados) – Faculdade de Tecnologia de São José dos Campos – FATEC SJC.

PLATTNER, Hasso. **A Course in In-Memory Data Management – The Inner Mechanics of In-Memory Databases**. 1. Ed. Brandenburg: Editora SPRINGER. 2013.

PLATTNER, Hasso; ZEIER, Alexander; **In-Memory Data Management: An Inflection Point for Enterprise Application**. ISBN 978-3-642-19362-0 e-ISBN 978-3-642-19363-7 – DOI 10.1007/978-3-642-19363-7. Springer Heidelberg Dordrecht London New York. Berlin. Springer-Verlag, 2011.

PREIMESBERGER, Chris. **In-Memory Databases Driving Big Data Efficiency: 10 Reasons Why**. EWEEK. Disponível em <

<http://www.eweek.com/database/slideshows/in-memory-databases-driving-big-data-efficiency-10-reasons-why/>>. Acesso em: 14 de mar, 2013.

PRONI, Ricardo Portilho. **Trabalhando com Sybase ASE – In-Memory Database**. Revista SQL Magazine, DEVMEDIA. Disponível em: <<http://www.devmedia.com.br/trabalhando-com-o-sybase-ase-in-memory-database-revista-sql-magazine-93/22801>>. Acesso em: 14 de mar, 2013.

ROB, Peter; CORONEL, Carlos. **Sistemas de Banco de Dados**. 8. Ed. 2011

ROUSE, Margaret. **In-Memory Database**. TECHTARGET. Disponível em <<http://whatistechtarget.com/definition/in-memory-database>>. Acesso em: 14 de mar, 2013.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. *Sistemas de Bancos de Dados*. 4. ed. Rio de Janeiro: Editora Elsevier, 2006.

SOARES, Bruno Eduardo; BOSCARIOLI, Clodis. **Modelo de Banco de Dados Colunar: Características, Aplicações e Exemplos de Sistemas**. 2012. p.10. Artigo - Centro de Ciências Exatas e Tecnológicas – Universidade Estadual do Oeste do Paraná (UNIOESTE). Paraná, Cascavel, 2012.

SYBASE Inc. **Performance & Tuning for In-Memory Databases Adaptive Server Enterprise 15.5**. SYBASE, Inc. Worldwide Headquarters one Sybase Drive. Dublin. CA 94568-7902 - USA, 2010.