

**UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC**

**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**FÁBIO ALANO DUARTE**

**INDEXAÇÃO TRANSACIONAL POR BITMAP: UMA ANÁLISE DE  
DESEMPENHO NO BANCO DE DADOS PÓS-RELACIONAL CACHE**

**CRICIÚMA, DEZEMBRO DE 2007**

**FÁBIO ALANO DUARTE**

**INDEXAÇÃO TRANSACIONAL POR BITMAP: UMA ANÁLISE DE  
DESEMPENHO NO BANCO DE DADOS PÓS-RELACIONAL CACHE**

**Trabalho de Conclusão de Curso  
apresentado para obtenção do Grau de  
Bacharel em Ciência da Computação da  
Universidade do Extremo Sul  
Catarinense.**

**Orientador: Prof. MSc. Paulo João  
Martins**

**CRICIÚMA, DEZEMBRO DE 2007**

**FÁBIO ALANO DUARTE**

**INDEXAÇÃO TRANSACIONAL POR BITMAP: UMA ANÁLISE DE  
DESEMPENHO NO BANCO DE DADOS PÓS-RELACIONAL CACHE**

Submetido ao corpo docente do Departamento de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

---

**Prof<sup>a</sup> MSc. Ana Cláudia Garcia Barbosa**

Coordenadora do Curso de Ciência da Computação

Banca Examinadora:

---

**Prof. MSc. Paulo João Martins (UNESC)**

Orientador

---

**Prof. MSc. Paracelso de Oliveira Caldas (UNESC)**

---

**Prof. Msc. Alessandro Zanini (ESUCRI / UNISUL)**

*Dedico essa pesquisa aos meus pais  
Arlei Duarte e Russimary Alano Duarte  
por tudo que fizeram e ainda fazem por mim  
em todos os aspectos de minha vida.*

## **AGRADECIMENTOS**

Agradeço a Deus por ter me concedido os pais que tenho, Arlei e Russimary, e por ELE ter me concedido a permissão de existir, e a oportunidade de evoluir como ser humano.

Ao meu orientador Sr. Paulo João Martins pela sua disposição, atenção, compreensão e experiências transmitidas, e por ele ter me incentivado nos momentos em que pensei em desistir.

Agradeço a todos os professores que tive, pois acredito que houve a contribuição de cada um para que eu tivesse condições de chegar até aqui, e concluir essa pesquisa.

Aos meus amigos que me deram força, incentivo e apoio nos momentos mais complicados, e que também estavam ao meu lado nos bons momentos.

Ao bar amarelo, lugar aonde por muitas vezes deixamos nosso stress, e a todos que de uma forma ou de outra já fizeram parte de minha vida.

*"Pensar é o trabalho mais duro que há.  
O que é provavelmente o motivo por que  
tão pouca gente se dedica a fazê-lo." (Henry Ford)*

## RESUMO

Esse trabalho tem como objetivo principal comparar o desempenho de duas estruturas de indexação, índices de *bitmap* versus índices tradicionais, utilizando o banco de dados pós-relacional Caché.

Foram construídas duas classes (tabelas), e foram populadas com cinco milhões de registros, utilizou-se a população automática de dados do Cachê passando como parâmetro o tipo de registro que seria desejado para cada propriedade da classe, procurando com isso aproximar o máximo possível de um ambiente real, foi utilizado um computador com uma configuração simples, e o banco em uma versão de avaliação.

Os testes buscaram realizar consultas simples e complexas, procurando simular a realidade de um sistema que utilize esses tipos de indexação, todas as características tanto do hardware como dos parâmetros de configuração do banco foram mantidas em todas as consultas, e realizaram-se vinte e uma consultas para cada comando SQL.

Com base nos resultados das tomadas de tempos tivemos a oportunidade de verificar que a indexação de *bitmap* foi em todos os casos mais eficiente tanto em consultas simples, como em consultas complexas, ficando claro que o tipo de índice influencia no desempenho da consulta.

**Palavras-Chave:** Indexação de *Bitmap*; Banco de dados Caché; *Data Warehouse*;

*Benchmark.*

## ABSTRACT

This work has as main objective to compare the performance of two structures of indexation, indexes of bitmap versus traditional indexes, using the after-relational data base Cache. Two classrooms had been constructed (tables), and had been populated with five million registries, used automatic population of data of the Cache passing as parameter the type of registry that would be desired for each property of the classroom, looking for with this to approach as much as possible of a real environment, was used a computer with a simple configuration, and the bank in an evaluation version. The tests had searched to carry through simple and complex consultations, looking for to simulate the reality of a system that uses these types of indexation, all the characteristics in such a way of the hardware as of the parameters of configuration of the bank the consultations had been kept in all, and had been become fulfilled twenty one consultations for each command SQL. With base on the results of the taking of the times, we had the chance to verify that the indexation of bitmap was in all the cases most efficient in such a way in simple consultations, as in complex consultations, being clearly that the type of index influences in the performance of the consultation.

**key-words:** Indexation of Bitmap; Data base Caché; Data Warehouse; Benchmark.

## LISTA DE SIGLAS

|       |                                                                            |
|-------|----------------------------------------------------------------------------|
| ANSI  | <i>American National Standards Institute</i>                               |
| BD    | Banco de Dados                                                             |
| CSP   | <i>Caché Server Page</i>                                                   |
| COS   | <i>Caché Object Script</i>                                                 |
| DB2   | <i>Universal Data Base</i>                                                 |
| DDL   | <i>Data Definition Language</i>                                            |
| DW    | <i>Data Warehouse</i>                                                      |
| HTML  | <i>HyperText Markup Language</i>                                           |
| IBM   | <i>International Business Machines</i>                                     |
| MUMPS | <i>Massachusetts General Hospital Utility Multi-Programming<br/>System</i> |
| OLTP  | <i>On-Line Transaction Processing</i>                                      |
| SUB   | <i>System Under Benchmark</i>                                              |
| SGBD  | Sistema Gerenciador de Banco de Dados                                      |
| SGBDR | Sistema Gerenciador de Banco de Dados Relacional                           |
| SQL   | <i>Structured Query Language</i>                                           |
| SI    | Sistema de Informação                                                      |
| SIG   | Sistemas de Informações Gerenciais                                         |
| TI    | Tecnologia da Informação                                                   |
| UF    | Unidade da Federação                                                       |
| VB    | <i>Visual Basic</i>                                                        |

## LISTA DE FIGURAS

|                                                                                                                 |    |
|-----------------------------------------------------------------------------------------------------------------|----|
| Figura 1. Arquitetura do Caché. ....                                                                            | 30 |
| Figura 2. Índices Tradicionais no Caché.....                                                                    | 42 |
| Figura 3. Execução do comando SQL na tabela com índices de bitmap.....                                          | 45 |
| Figura 4. Definição da Classe PessoaBit.....                                                                    | 49 |
| Figura 5. Definição da Classe Pessoa. ....                                                                      | 50 |
| Figura 6. Total de linhas na tabela PessoaBit.....                                                              | 51 |
| Figura 7. Total de linhas na tabela Pessoa. ....                                                                | 52 |
| Figura 8. Dados que foram gerados automaticamente.....                                                          | 52 |
| Figura 9. Propriedade da tabela PessoaD.....                                                                    | 53 |
| Figura 10. Tempo gasto pela consulta 12 na tabela Pessoa. ....                                                  | 56 |
| Figura 11. Tempo gasto pela consulta 12 na tabela PessoaBit.....                                                | 57 |
| Figura 12. Tempo gasto na execução da primeira consulta após a inicialização do banco<br>tabela PessoaBit.....  | 59 |
| Figura 13. Tempo gasto na execução da segunda consulta após a inicialização do banco,<br>tabela PessoaBit.....  | 59 |
| Figura 14. Tempo gasto na execução da primeira consulta após a inicialização do banco,<br>tabela PessoaBit..... | 60 |
| Figura 15. Tempo gasto na execução da segunda consulta após a inicialização do banco,<br>tabela Pessoa.....     | 61 |
| Figura 16. Armazenamento dos índices de <i>bitmap</i> .....                                                     | 62 |
| Figura 17. Armazenamento dos índices tradicionais. ....                                                         | 62 |

## LISTA DE TABELAS

|                                                                              |    |
|------------------------------------------------------------------------------|----|
| Tabela 1. Características e versões do Caché.....                            | 28 |
| Tabela 2. Plataformas e sistemas operacionais que o Caché suporta. ....      | 31 |
| Tabela 3. Índice tradicional.....                                            | 42 |
| Tabela 4. Tabela Pessoa. ....                                                | 43 |
| Tabela 5. Índice de Bitmap na propriedade Estado. ....                       | 44 |
| Tabela 6. Índice de <i>bitmap</i> na propriedade idade.....                  | 45 |
| Tabela 7: Resultado da operação AND em índices de bitmap. ....               | 46 |
| Tabela 8. Configuração do computador utilizado nos testes.....               | 47 |
| Tabela 9. Tamanho das tabelas.....                                           | 53 |
| Tabela 10. Comparativo entre índice <i>bitmap</i> e índice tradicional. .... | 55 |
| Tabela 11. Resumo dos tempos das consultas. ....                             | 57 |
| Tabela 12 Tempo gasto na primeira consulta de cada classe. ....              | 58 |

## LISTA DE QUADROS

|                                                 |    |
|-------------------------------------------------|----|
| Quadro 1. Tempos das consultas 1, 2 e 3.....    | 67 |
| Quadro 2. Tempos das consultas 4 e 5.....       | 68 |
| Quadro 3. Tempos das consultas 6 e 7.....       | 69 |
| Quadro 4. Tempo das consultas 8 e 9. ....       | 70 |
| Quadro 5. Tempos das consultas 10, 11 e 12..... | 72 |

## SUMÁRIO

|                                                                                        |           |
|----------------------------------------------------------------------------------------|-----------|
| <b>1 INTRODUÇÃO .....</b>                                                              | <b>14</b> |
| 1.1 OBJETIVO GERAL .....                                                               | 15        |
| 1.2 OBJETIVOS ESPECÍFICOS .....                                                        | 15        |
| 1.3 JUSTIFICATIVA .....                                                                | 15        |
| 1.4 ESTRUTURA DO TRABALHO.....                                                         | 17        |
| <b>2 IMPORTÂNCIA DA INFORMAÇÃO COMO PLANEJAMENTO ESTRATÉGICO PARA AS EMPRESAS.....</b> | <b>18</b> |
| 2.1 GESTÃO DA INFORMAÇÃO .....                                                         | 20        |
| <b>3 BANCO DE DADOS.....</b>                                                           | <b>22</b> |
| 3.1 VISÃO GERAL DE UM SISTEMA GERENCIADOR DE BANCO DE DADOS .....                      | 22        |
| 3.2 ARQUITETURAS DE BANCO DE DADOS.....                                                | 23        |
| 3.3 ARQUITETURAS .....                                                                 | 24        |
| 3.4 BANCO DE DADOS RELACIONAL.....                                                     | 26        |
| 3.5 BANCO DE DADOS CACHÉ .....                                                         | 26        |
| 3.6 HISTÓRICO DO CACHÉ.....                                                            | 27        |
| 3.7 ARQUITETURA DO CACHÉ.....                                                          | 29        |
| 3.8 PLATAFORMA E SISTEMA OPERACIONAL .....                                             | 31        |
| 3.9 MECANISMO MULTIDIMENSIONAL DE DADOS.....                                           | 31        |
| 3.10 ESCALABILIDADE MULTIDIMENSIONAL .....                                             | 32        |
| 3.11 TRAVAMENTO LÓGICO .....                                                           | 33        |
| <b>4 BENCHMARK - AVALIAÇÃO DE DESEMPENHO EM BANCO DE DADOS.....</b>                    | <b>34</b> |
| <b>5 DATA WAREHOUSE .....</b>                                                          | <b>37</b> |
| 5.1 OTIMIZAÇÃO E DESEMPENHO .....                                                      | 38        |
| 5.1.1 <i>Cache</i> de dados .....                                                      | 38        |
| 5.1.2 Armazenamento de dados .....                                                     | 40        |
| 5.1.3 Índices.....                                                                     | 41        |
| 5.1.3.1 Índices Tradicionais .....                                                     | 42        |
| 5.1.3.2 Índices de <i>Bitmap</i> .....                                                 | 43        |

|                                                                   |           |
|-------------------------------------------------------------------|-----------|
| <b>6 BENCHMARK INDICE TRANSACIONAL DE BITMAP FRENTE INDEXAÇÃO</b> |           |
| <b>TRADICIONAL .....</b>                                          | <b>47</b> |
| 6.1 METODOLOGIA .....                                             | 47        |
| <b>7 CONCLUSÃO.....</b>                                           | <b>63</b> |
| <b>7.1 TRABALHOS FUTUROS .....</b>                                | <b>64</b> |
| <b>REFERÊNCIAS .....</b>                                          | <b>65</b> |
| <b>ANEXOS.....</b>                                                | <b>67</b> |

## 1 INTRODUÇÃO

Nas organizações, atualmente, existe uma grande necessidade de informações, pois estamos vivendo esta época, a revolução da informação, como podemos observar diariamente mais e mais cursos surgem oferecendo as pessoas cada vez mais elementos.

Dados brutos derivados dos sistemas corporativos existem em grande quantidade, porem existe a necessidade de transformá-los em informações que possam auxiliar administradores de maneira proveitosa na tomada de decisão.

Esses dados brutos diariamente vão sendo armazenados, através de notas fiscais de venda, de compra, orçamentos, enfim de todos os processamentos de transações da organização, porem necessitam de maneira ágil serem adequados a missão da empresa, e os que não alinharem-se devem ser descartados, e prontamente auxiliarem nas tomadas de decisões.

Existindo atualmente a necessidade do processamento de consultas e tratamento de dados, de maneira ágil, ou quanto mais disponível melhor, surge a necessidade de que os Bancos de Dados.

E com isso sejam cada vez mais rápidos na disponibilização destes, sendo que esses com o auxilio de Sistemas de Informações Gerencias (SIG), podem auxiliar na tomada de decisão. Muitos deles precisam ser tratados e disponibilizados de maneira *on-line*, sem que os administradores precisem perder tempo em preparar consultas, ou visões sobre o banco.

Nos Banco de Dados os índices são um dos principais responsáveis por agilizarem respostas as consultas, servem como atalho para informação requerida.

Tendo em vista essa necessidade, este trabalho tem por objetivo principal estudar a Indexação Transacional de *bitmap* no Banco de Dados Caché.

### 1.1 OBJETIVO GERAL

Analisar o desempenho em consultas na utilização de índices tradicionais versus indexação transacional por *bitmap* no banco de dados pós-relacional Caché.

### 1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa são:

- a) enunciar a necessidade da informação para a empresa diante do fator desempenho;
- b) descrever a arquitetura do Banco de Dados Caché;
- c) compreender e analisar a Indexação Transacional por *bitmap*;
- d) verificar a utilização de Indexação Transacional por *bitmap* frente a indexação tradicional.

### 1.3 JUSTIFICATIVA

Atualmente, com a utilização de sistemas informatizados em diversos ramos de negócios, os bancos de dados são de fundamental importância. Molia, Ullman e Widdon (2001) ressaltaram que bancos de dados (BD) são usados para manter registros internos, apresentar dados a consumidores e clientes na *World Wide Web* e fornecer suporte a muitos outros processos comerciais. Da mesma forma, os bancos de dados são encontrados no núcleo de muitas investigações científicas.

Nos bancos de dados para tornar as consultas mais eficientes é comum a criação de tabelas auxiliares, ou seja, índice. Estas tem por função acelerar o tempo de acesso às linhas de uma tabela, sendo transparentes para os usuários e aplicações. Da Silva (2002) destaca que o índice é um caminho mais rápido para acesso aos dados, implicando na velocidade de execução, ou seja, se um determinado valor foi indexado o índice apontará diretamente para a linha correspondente ao valor procurado.

Tratando-se de índices, temos o índice de *bitmap*, consiste em um vetor de *bits* (zeros (0) e uns (1)), é um tipo de índice diferente onde, usa-se um mapa de bits para cada valor da chave, ao invés de utilizar uma lista de identificadores para as linhas. Com a utilização de índices de *bitmap* consultas complexas podem ser realizadas executando-se operações *booleanas* (E,OU), desse modo permite determinar exatamente, quais linhas se ajustam às condições da consulta, sem a necessidade de fazer buscas pelo banco de dados completo (INTERSYSTEMS, 2002). Justifica-se, portanto conhecer as reais características e vantagens da implementação com o objetivo de adquirir conhecimento específico sobre indexação de *bitmap*.

O Caché implementou uma nova tecnologia – Índices Transacionais de *Bitmap* – que utilizam estruturas de dados multidimensionais, tal banco é utilizado pelo acadêmico no seu dia-a-dia de trabalho, observando pela experiência adquirida nos diversos projetos da empresa, surge então a necessidade de comparar o índice tradicional com o índice de *bitmap*, permitindo uma futura escolha do tipo de índice que apresentar um melhor desempenho.

É neste contexto que se norteia o tema em questão e que justifica a motivação da presente pesquisa.

## 1.4 ESTRUTURA DO TRABALHO

A pesquisa esta está estruturada em oito capítulos distribuídos da seguinte maneira:

O capítulo 1, é a introdução, aonde descreve-se o contexto do trabalho, e também são apresentados os objetivos geral e específicos, justificativa e a estrutura do trabalho, o capítulo 2, descreve de uma forma sucinta a importância que as informações tem na atualidade para as organizações, o capítulo 3 explana o que é um banco de dados, sua estrutura, e um pouco do histórico, também nesse capítulo é analisado o banco de dados *cahcé*, seu histórico, plataformas, e algumas de suas tecnologias.

O Capítulo 4 aborda alguns dos conceitos sobre os itens que são necessários para uma análise de desempenho, em sistemas transacionais. O que é um *data warehouse*, suas características, índices de *bitmap*, índices tradicionais, e alguns itens para que esse sistema tenha uma melhora de *performance* são abordados no capítulo 5, o capítulo 6 trata da pesquisa em si, aonde demonstra como foram realizadas as análises de desempenho, os tempos das consultas, comparativos entre os dois tipos de índices com relação ao desempenho.

A conclusão e sugestões de trabalhos futuros são apresentados no capítulo 7. Nos anexos estão todas as tabelas aonde foram catalogados os tempos de cada uma das consultas.

## 2 IMPORTÂNCIA DA INFORMAÇÃO COMO PLANEJAMENTO ESTRATÉGICO PARA AS EMPRESAS

A palavra informação deriva do latim *informationem* (delinear, conceber idéia), ou seja, dar forma ou moldar a mente. Mas na realidade existem muitas e variadas definições, cada uma mais complexa que outra. Podemos também dizer que é um processo que visa o conhecimento, ou, mais simplesmente, é tudo o que reduz a incerteza. É uma ferramenta que ajuda na compreensão sobre o mundo (BRAGA, 1996).

Uma nova revolução da informação está em andamento. Ela começou nas empresas, com informação de negócios, mas seguramente ira engolfar todas as instituições da sociedade e mudará radicalmente o significado de informação tanto para organizações como para indivíduos. DRUCKER (2001, p. 81).

Atualmente tornou-se uma necessidade diária crescente para qualquer setor da atividade humana e é indispensável mesmo que a sua procura não seja ordenada ou sistemática, mas resultante apenas de decisões casuais e/ou intuitivas.

Uma organização é, por natureza, um sistema aberto aonde ocorre interação apoiada por uma gama de processos interligados, onde os meios de comunicação existentes dentro da empresa e fora da empresa são irrigados por informação (BRAGA, 1996).

Devido a grande quantidade de acontecimentos e a crescente velocidade com que as mudanças ocorrem, faz com quem as organizações enfrentem novas situações quase que diariamente, e essas novas situações podem constituir ameaças ou oportunidades, exigindo assim que o gestor empresarial esteja bem informado, referente a questões internas e externas a organização.

Com o aumento da concorrência e da complexidade, faz sentir no meio empresarial, a necessidade da obtenção de melhores recursos com o objetivo de agradar

cada vez mais os clientes de uma maneira singular, e aperfeiçoar a utilização dos mesmos.

Com o desenvolvimento do comércio eletrônico, a facilidade de atualmente realizar-se o comércio internacional, fruto da constante e crescente interligação entre nações, os investimentos no exterior e a tendência dos padrões de consumo tornarem-se cada vez mais semelhantes, faz com que o mundo seja encarado cada vez mais como um só mercado, devido a esses acontecimentos as organizações têm de conviver com a competição internacional dentro dos seus mercados e ao mesmo tempo tentarem penetrar nos mercados externos com o objetivo de aproveitar as novas oportunidades de negócio.

Conforme Drucker (1993 apud BRAGA, 1996, p.2) defende o primado da informação como a base e a razão para um novo tipo de gestão, em que a curto prazo se perspectiva a troca do binômio capital/trabalho pelo binômio informação/conhecimento como fatores determinantes no sucesso empresarial.

Caminha-se para uma sociedade do saber onde o valor da informação tende a ir além da importância do capital. A informação e o conhecimento são as chaves para a produtividade e a competitividade.

A atual gestão, a gestão moderna estabelece que a tomada de decisão seja feita com o máximo de informação, se em outras épocas e em ambientes mais estáveis a informação apenas reduzia a incerteza, cada vez mais na atualidade apresenta-se como fator crítico de sucesso.

Com base nessa realidade, somos inspirados a acreditar que as empresas em sua grande maioria deverão fazer uma reestruturação organizacional em torno da informação. E é nesse contexto que tem lugar a gestão de tecnologias, consideradas como uma nova e importante fonte de vantagem competitiva, se bem empregada.

## 2.1 GESTÃO DA INFORMAÇÃO

No entanto, tão logo uma organização dê os primeiros passos incipientes para converter “dados” em “informações”, os seus processos de decisão, a sua estrutura administrativa e a sua maneira de trabalhar começam a se transformar. (DRUCKER, 1993, p. 178).

A informação pode ser analisada como o elemento essencial na tomada de decisão, sendo um dos pilares das organizações, o fato de não possuí-las torna a tomada de decisão uma atividade mais complexa, mesmo de posse desse recurso a falta de conhecimento em como utilizá-la também acarretará dificuldades no gerenciamento das tomadas de decisão.

Tão importante quanto ter a informação é o saber como, quando e em que situação utilizá-la e aprender novos modos de ver e empregar o recurso tornando dessa maneira o funcionamento da organização mais eficiente. Quanto mais eficiente for o acesso mais rapidamente pode-se alcançar o objetivo da organização

Os dados de onde provem as informações são de fundamental importância para as organizações dessa maneira, portanto esse recurso merece ser gerido de uma maneira particular, é esse o objetivo da Gestão da Informação

De acordo com o autor para que esta gestão [de informação] seja eficaz, é necessário que se estabeleçam um conjunto de políticas coerentes que possibilitem o fornecimento de informação relevante, com qualidade suficiente, precisa, transmitida para o local certo, no tempo correto, com um custo apropriado e facilidades de acesso por parte dos utilizadores autorizados. (REIS, apud Braga, 1996, p. 3).

A gestão da informação tem como objetivo fundamental sustentar a política da organização, e tornar mais eficiente o conhecimento e a interação entre diversos setores que a constituem; dar subsídio aos administradores na tomada de decisão; apresentar com maior nitidez o meio ao qual esta envolvida; apoiar a evolução da estrutura empresarial, no qual está permanentemente buscando adequar-se as novas exigências mercadológicas e da concorrência; ajudar a formar uma boa imagem da

organização, dos produtos e projetos, por meio da implantação de comunicação interna e externa mais ágeis e eficientes.

A Gestão da Informação deve estar de acordo com o SI (Sistema de Informação) da organização, realizando o papel de apoio na articulação de vários subsistemas que o constituem. Conforme os mesmos forem efetuando transações e processamentos provenientes de fontes de dados, produzirem assim informações úteis e em tempo real, auxiliando de modo *on-line* nas tomadas de decisão da empresa, contribuindo para criar vantagens competitivas de mercado (BRAGA, 1996).

Por se tratar de uma atividade relativamente nova que tenta fazer uma ponte entre gestão estratégica e a aplicação das Tecnologias de Informação nas organizações, a Gestão da Informação tem como objetivo principal tentar perceber qual a informação é de maior interesse para a organização e em seguida, definir os processos, identificar fontes, modelar sistemas. As novas Tecnologias de Informação são as ferramentas que permitem gerir a informação em novos moldes, fazendo com que o fluxo de informações, a tomada de decisão e a transmissão sejam feitos de maneira ágil, consumindo assim menos tempo e recursos, tornando a tarefa de gerenciamento e tomada de decisão menos complexa (BRAGA, 1996).

Conforme foi abordado o gerenciamento das informações é de vital importância para o sucesso organizacional, é nesse contexto que a utilização de Sistemas Gerenciadores de Banco de Dados (SGBD) e sistemas de *Data Warehouse* incluem-se, para o gerenciamento de informações e agilidade na recuperação das mesmas.

### 3 BANCO DE DADOS

Gracia-Molina, Ullman e Window (2001, p.2) definem “Banco de dados: uma coleção de dados organizados para facilitar o acesso e a modificação, preservada durante um longo período”. E são vários conjuntos de dados com estrutura regular que organizam a informação. Normalmente armazenam informações que serão utilizadas com um mesmo objetivo. Para o gerenciamento dos bancos de dados existe uma coleção de softwares específicos que são denominados de Sistemas Gerenciadores de Banco de Dados (SGBD), esses aplicativos têm por objetivo concentrar no servidor a responsabilidade de gerenciar o acesso, manipulação e organização dos dados, disponibilizando para o administrador uma interface para que possa ser feita a inclusão, exclusão e alteração de dados, como também projeto e testes sob a base de dados.

#### 3.1 VISÃO GERAL DE UM SISTEMA GERENCIADOR DE BANCO DE DADOS

De acordo com Garcia-Molina, Ullman e Window (2001, p.1) os recursos que um SGBD oferece ao usuário são:

- a) **armazenamento persistente:** permite o armazenamento de grandes quantidades de dados sendo que estes podem ser vistos como se fossem independentes dos processos que possam utilizá-los, e proporciona flexibilidade, com as estruturas de dados e permitem eficientes acessos a muitos dados;
- b) **interface de programação:** realiza o acesso e também a modificação dos dados por meio de uma linguagem de consultas;

- c) **gerenciamento de transações:** um elemento do banco pode ser acessado por diferentes transações ao mesmo tempo, e para que não ocorram conseqüências indesejáveis durante o acesso simultâneo, o SGBD utiliza isolamento<sup>1</sup>, atomicidade<sup>2</sup> e resiliência<sup>3</sup>.

### 3.2 ARQUITETURAS DE BANCO DE DADOS

Deve-se levar em consideração alguns aspectos importantes para atingir a eficiência dos sistemas informatizados, com o objetivo de atender os usuários nos mais variados ramos de aplicações existentes, tais aspectos são:

- a) projeto lógico e funcional do banco de dados deve ser capaz de prever o aumento de informações a curto, médio e longo prazo e adaptar-se;
- b) deve-se ter generalidade e alto grau de abstração de dados, ser confiável, eficiente no armazenamento, permitir a utilização de diferentes SGBD, utilizar linguagens de consultas padronizadas;
- c) projeto de uma interface ágil para propiciar aprendizado suave ao usuário, no intuito de minimizar o esforço cognitivo;
- d) implementação de um projeto de interface compatível com múltiplas plataformas (UNIX, Windows NT, Windows Workgroup, entre outros);
- e) independência da interface em relação ao SGBD que dará condição às operações de armazenamento de dados;
- f) conversão e mapeamento de diferentes paradigmas utilizados no desenvolvimento de interfaces (procedural, Orientado a Objeto, Orientado a evento), servidores de dados (Relacional, Orientado a

---

<sup>1</sup> Aparenta que as transações são executadas isoladas. Garcia-Molina, Ullman e Window (2001, p.2)

<sup>2</sup> Uma transação executada completamente ou anulada. Garcia-Molina, Ullman e Window (2001, p.2)

<sup>3</sup> Capacidade de recuperar muitos tipos de falhas e erros. Garcia-Molina, Ullman e Window (2001, p.2)

Objetos) e programação dos aplicativos (Imperativo, Orientado a Objetos).

### 3.3 ARQUITETURAS

As primeiras arquiteturas usavam *mainframes* para executar o processamento principal e de todas as funções do sistema, incluindo os programas aplicativos, programas de interface com o usuário, bem como a funcionalidade dos SGBD. Esta é a razão pela qual a maioria dos usuários fazia acesso aos sistemas via terminais que não possuíam poder de processamento, apenas a capacidade de visualização. Todos os processamentos eram feitos remotamente, apenas as informações a serem visualizadas e os controles, eram enviados do *mainframe* para os terminais de visualização, conectados a ele por redes de comunicação.

Como os preços do hardware foram decrescendo, muitos usuários trocaram seus terminais por computadores pessoais e estações de trabalho. No começo os SGBD usavam esses computadores da mesma maneira que usavam os terminais, ou seja, o SGBD era centralizado e toda sua funcionalidade, execução de programas aplicativos e processamento da interface do usuário eram executados em apenas uma máquina. Gradualmente, os SGBD começaram a explorar a disponibilidade do poder de processamento no lado do usuário, o que levou à arquitetura cliente-servidor.

A arquitetura cliente-servidor foi desenvolvida para dividir ambientes de computação onde um grande número de computadores pessoais, estações de trabalho, servidores de arquivos, de impressoras, de banco de dados e outros equipamentos estão conectados em uma mesma rede. A idéia é definir servidores específicos para cada função, por ex: servidor de arquivos, que mantém os arquivos de máquinas clientes,

servidores de impressão que esta conectada as várias impressoras; assim, quando se desejar imprimir algo, todas as requisições de impressão vão para uma única máquina. As máquinas clientes disponibilizam para o usuário as interfaces apropriadas para a utilização desses sistemas, bem como poder de processamento para executar aplicações locais. Esta arquitetura se tornou muito popular por algumas razões:

- a) facilidade de implementação dada a clara separação das funcionalidades dos servidores;
- b) servidor é inteligentemente utilizado porque as tarefas mais simples são delegadas às máquinas clientes mais baratas;
- c) o usuário pode executar uma interface gráfica que lhe é familiar, ao invés de usar a interface do servidor.

Desta maneira, a arquitetura cliente-servidor foi incorporada aos SGBD comerciais. Diferentes técnicas foram propostas para a implementação dessa arquitetura, sendo que a mais adotada pelos Sistemas Gerenciadores de Banco de Dados Relacionais (SGBDR) comerciais é a inclusão de suas funcionalidades no lado do servidor. As consultas e as transações permanecem no servidor, sendo que este é chamado de servidor de consulta ou transação. É assim que um servidor SQL (*Structured Query Language*) é fornecido aos clientes. Cada cliente tem que formular suas consultas SQL, prover a interface do usuário e suas funções usando uma linguagem de programação. O cliente pode também se referir a um dicionário de dados o qual inclui informações sobre a distribuição dos dados em vários servidores SQL, bem como os módulos para a decomposição de uma consulta global em um número de consultas locais que podem ser executadas em vários sítios. Comumente o servidor SQL também é chamado de *back-end machine* e o cliente de *front-end machine*. Como SQL provê

uma linguagem padrão para o SGBDR, esta criou o ponto de divisão lógica entre o cliente e o servidor.

### 3.4 BANCO DE DADOS RELACIONAL

O modelo de dados relacional possui estruturas de dados simples, são relações formadas por tabelas bidimensionais e seus elementos são itens de dados.

A normalização é um processo que auxilia o banco de dados relacional a eliminar possíveis anomalias, e possui regras de integridades para que os dados sejam mantidos de maneira uniforme.

Esse modelo permite a manipulação de relações orientada a conjuntos. Essa característica levou ao desenvolvimento de linguagens não procedurais poderosas baseadas na teoria dos conjuntos (álgebra relacional) ou em lógica (cálculo relacional).

### 3.5 BANCO DE DADOS CACHE

O Banco de dados Caché é um produto comercial de marca registrada da *InterSystems Corporation*. É um banco de dados do tipo pós-relacional<sup>4</sup> oferece a possibilidade de acesso integrado aos dados simultaneamente de três maneiras: orientado a objetos, SQL e acesso multidimensional. O mapeamento não é necessário entre acesso de *views*<sup>5</sup> de dados objetos, relacionais e multidimensionais, desta forma temos uma economia tanto em tempo de desenvolvimento como de processamento.

---

<sup>4</sup> Utiliza a visão relacional e orientada a objetos dos mesmos dados, sem necessidade de mapeamentos ou redundância.

<sup>5</sup> É uma relação que não armazena dados, mas é definida dinamicamente por uma consulta que é previamente analisada e otimizada.

Atua tanto como servidor de dados multidimensional e/ou um servidor de aplicações. (INTERSYSTEMS, 2007).

### 3.6 HISTÓRICO DO CACHE

O desenvolvimento do banco iniciou-se em 1995 como um produto baseado na tecnologia ANSI (*American National Standards Institute*) - *Standard M*, (tecnologia tipo M, derivado do MUMPS - *Massachusetts General Hospital Utility Multi-Programming System*). O objetivo principal era criar um produto de alto desempenho e de desenvolvimento rápido de aplicações (INTERSYSTEMS, 2007). A Tabela 1 apresenta as características de cada umas das versões.

**Tabela 1. Características e versões do Caché.**

| <b>Versão</b> | <b>Ano</b> | <b>Características</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2.0           | 1997       | <ul style="list-style-type: none"> <li>- Distributed Caché Protocol</li> <li>- Suporte ao Windows</li> <li>- Suporte ao UNIX</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 2.1           | 1997       | <ul style="list-style-type: none"> <li>- Caché Objects</li> <li>- ActiveX binding</li> <li>- ODBC</li> <li>- Suporte ao OpenVMS</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 3.0           | 1998       | <ul style="list-style-type: none"> <li>- Suporte ao Native UNICODE</li> <li>- Suporte ao Localization</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 3.1           | 1999       | <ul style="list-style-type: none"> <li>- Unified Data Architecture</li> <li>- Suporte ao Native object</li> <li>- Caché Explorer</li> <li>- Caché Control Panel</li> <li>- Caché Configuration Manager</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                 |
| 3.2           | 2000       | <ul style="list-style-type: none"> <li>- JDBC</li> <li>- Java Binding</li> <li>- Suporte a Linux</li> <li>- Caché SQL Manager</li> <li>- Caché Studio</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 4.0           | 2000       | <ul style="list-style-type: none"> <li>- Caché Server Pages</li> <li>- Incremento do CachéObjectScript</li> <li>- Caché SQL Gateway</li> <li>- Class Packages</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 4.1           | 2001       | <ul style="list-style-type: none"> <li>- Novo Caché database engine</li> <li>- Novo Lock Manager</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 5.0           | 2002       | <ul style="list-style-type: none"> <li>- Integração do Caché Studio</li> <li>- Sistema de documentação Online</li> <li>- Suporte a ECP Distributed Database</li> <li>- Suporte a Bitmap Index</li> <li>- Suporte a XML</li> <li>- Suporte a SOAP e Web Services</li> <li>- Suporte a Java e EJB</li> <li>- Suporte a C++</li> <li>- Caché Activate</li> </ul>                                                                                                                                                                                                                                     |
| 5.1           | 2005       | <ul style="list-style-type: none"> <li>- Nested transaction and rollback support</li> <li>- New language bindings</li> <li>- Namespace mapping for class packages</li> <li>- Object synchronization</li> <li>- Enhanced shadowing</li> <li>- SNMP support</li> <li>- ECP enhancements</li> <li>- Mac OS X server support</li> <li>- Index on computed fields</li> <li>- Studio Enhancements</li> </ul>                                                                                                                                                                                            |
| 5.2           | 2006       | <ul style="list-style-type: none"> <li>- Jalapeño Java persistence API</li> <li>- Caché Managed Provider for .net</li> <li>- IEEE 8-byte Floating Point Support</li> <li>- Direct FileMan Dictionary Converter</li> <li>- Code Completion in Caché Studio</li> <li>- Process-Private Globals</li> <li>- Caché Journal File Encryption</li> <li>- Version Checking (and Optimistic Concurrency)</li> <li>- Dynamic Dispatch</li> <li>- Free Text Search</li> <li>- ODBC Multiple Result Sets</li> <li>- WMI Support</li> <li>- Enhanced Debugging Capabilities</li> <li>- T-SQL Support</li> </ul> |

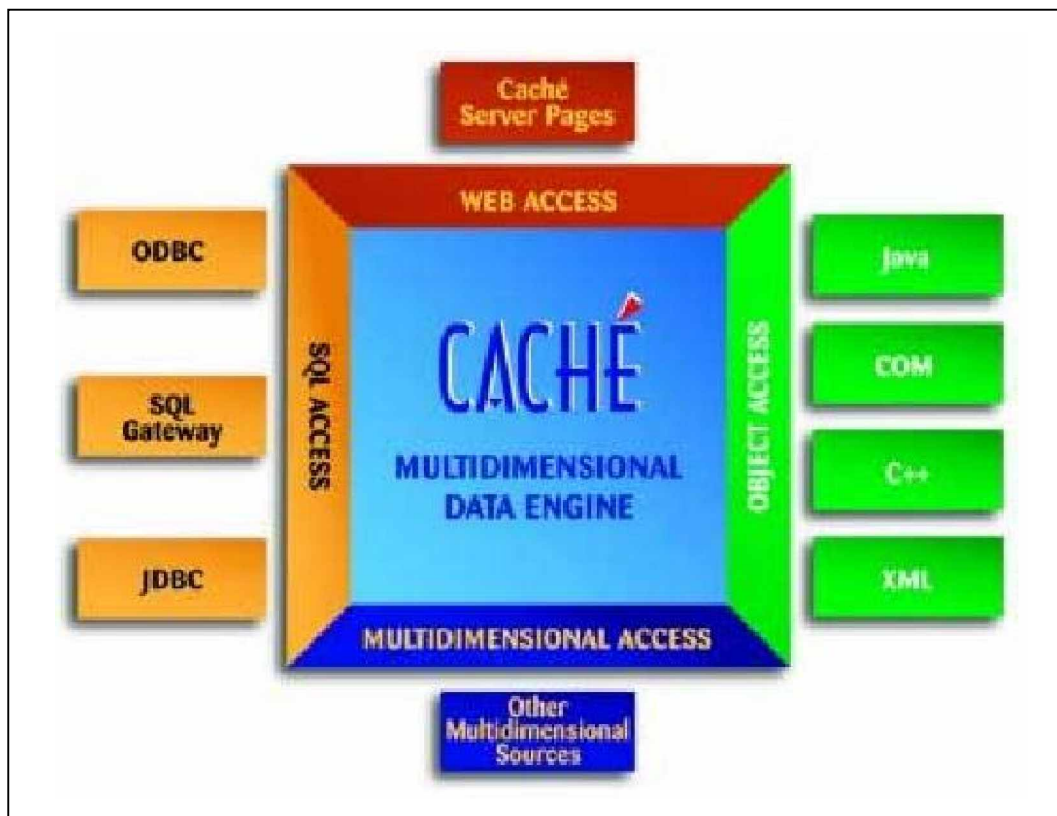
|         |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         |      | <ul style="list-style-type: none"> <li>- Device Level SSL and TLS support</li> <li>- Enhanced ECP Performance</li> <li>- Enhanced Windows Cluster Resource Management</li> <li>- Improved RPM Linux Installation</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 2007.01 | 2006 | <ul style="list-style-type: none"> <li>- Call In / Call Out Enhancements</li> <li>- New Error Handling Syntax</li> <li>- Long String Support</li> <li>- Security Enhancements</li> <li>- SQL Gateway via JDBC</li> <li>- Objective C Binding</li> <li>- New Distribution Mechanism for C++ Binding</li> <li>- Zen</li> <li>- SQL Enhancements</li> <li>- Enhanced T-SQL Support</li> <li>- Routine Performance Enhancements</li> <li>- Journal Enhancements</li> <li>- Light C++ Binding</li> <li>- CSP Gateway on OpenVMS</li> <li>- Support for GB18030 Character Set</li> <li>- Other Changes</li> <li>- Sun Solaris AMD</li> <li>- OpenVMS 8.3 for Alpha and Integrity</li> <li>- SUSE Linux Enterprise Server 10</li> </ul> |

Fonte: INTERSYSTEMS, (2007).

### 3.7 ARQUITETURA DO CACHÉ

Tem como características marcantes: alto desempenho, escalabilidade maciça, desenvolvimento rápido de aplicação (INTERSYSTEMS, 2007). O Caché trata o armazenamento de dados de forma multidimensional<sup>6</sup>, garantindo assim desempenho, os dados ficam disponíveis para uma gama de tecnologias como pode ser visto na Figura 1.

<sup>6</sup> Os dados são de-normalizados e estruturados em diversas dimensões.



**Figura 1. Arquitetura do Caché.**

Fonte: INTERSYSTEMS (2007).

O Caché possui também características de bancos de dados tradicionais, incorpora um ambiente de desenvolvimento de aplicações baseadas em *browser*<sup>7</sup>, aonde possibilita a criação de aplicação em *Caché Server Page* (CSP), possui também, para aplicações não baseadas em *browser* uma *interface* para integração com VB, Delphi, Java, ou C++. Sendo o Banco Caché desenvolvido em ANSI C e com otimizações em *Assembly*<sup>8</sup> específicas para cada plataforma suportada, torna o banco leve e pouco exigente com relação ao hardware (INTERSYSTEMS, 2007).

<sup>7</sup> Programa para visualização de paginas de Internet.

<sup>8</sup> Notação legível por humanos para o código de máquina.

### 3.8 PLATAFORMA E SISTEMA OPERACIONAL

Caché em sua versão 2007.01 tem suporte as seguintes plataformas e sistemas operacionais relacionados na Tabela 2.

**Tabela 2. Plataformas e sistemas operacionais que o Caché suporta.**

| Plataforma                                  | Sistema Operacional                                                     | SQL Gateway |
|---------------------------------------------|-------------------------------------------------------------------------|-------------|
| Apple                                       | Mac OS X 10.3, 10.4                                                     | Não         |
| HP Alpha                                    | OpenVMS 7.2-2, 7.3-1, 7.3-2                                             | Não         |
| HP Alpha                                    | Tru64 UNIX 5.1, 5.1 A, 5.1 B                                            | Não         |
| HP (PA-RISC)                                | HP-UX 11, 11i v1, 11i v2                                                | Sim         |
| HP (Itanium)                                | HP-UX 11i v2                                                            | Sim         |
| IBM pSeries                                 | AIX 4.3.3, 5.1, 5.2, 5.3                                                | Sim         |
| Red Hat Enterprise Linux AS (Intel 32-bit)  | 3, 4                                                                    | Sim         |
| Red Hat Enterprise Linux AS (64-bit)        | 3, 4                                                                    | Sim         |
| Sun Solaris (SPARC)                         | 8, 9, 10                                                                | Sim         |
| SuSE Linux (Intel 32-bit)                   | 7.3, 8.0, 9.0                                                           | Sim         |
| SuSE Linux Enterprise Server (Intel 32-bit) | 8, 9                                                                    | Sim         |
| SuSE Linux Enterprise Server (64-bit)       | 8, 9                                                                    | Sim         |
| Microsoft Windows (32-bit)                  | NT 4 (SP4, SP5, SP6), 2000 (SP3, SP4), XP (SP1, SP2), Server 2003 (SP1) | Sim         |
| Microsoft Windows (64-bit)                  | Server 2003 (SP1)                                                       | Sim         |

Fonte: INTERSYSTEMS (2007).

### 3.9 MECANISMO MULTIDIMENSIONAL DE DADOS

Diferente de bancos de dados relacionais que forçam os dados em tabelas dimensionais, o Caché os armazena em tabelas multidimensionais. Além de permitir modelagens realistas de dados, tabelas multidimensionais são mais rápidas de acessar, pois eliminam os custos fixos associados com os *table-hopping* e *joins* que exemplificam a tecnologia relacional. Outro fator de aumento de desempenho é o *Distributed Cache Protocol*, que reduz o tráfego de trabalho em sistemas distribuídos. Embora os dados estejam armazenados de forma multidimensional, dá ao desenvolvedor a liberdade para modelar seus dados da forma que escolher: como objeto, como tabela ou como matriz multidimensional. O Caché vem com uma *interface* gráfica

para criar *Caché Objects*. Ele também pode aceitar dados vindos do *Rational Rose* e arquivos de *Data Definition Language* (DDL).

Devido a Arquitetura Unificada de Dados, todos os dados são automaticamente acessados tanto como objetos como tabelas. Não existe a necessidade de mapear uma forma para a outra, e nenhum custo fixo é necessário para converter formas. A Arquitetura Unificada de Dados aumenta tanto a produtividade quanto ao desempenho. Também permite escolhas quando o banco de dados fica com o *script* da lógica de negócios. O *Caché Objects Script* (COS) suporta diversos métodos de acessos a dados: objetos, SQL, multidimensional e HTML (*HyperText Markup Language*) embutido. O *Caché Basic* é similar ao *Visual Basic*, com poucas modificações para aproveitar as vantagens do banco (INTERSYSTEMS, 2007).

### 3.10 ESCALABILIDADE MULTIDIMENSIONAL

O armazenamento dos dados é em vetores esparsos, o que torna seu servidor compacto, capaz de armazenar eficientemente, no mesmo espaço em disco, mais informações que outros SGBDR. Esta eficiência resulta não só em desempenho superior, como também em um menor investimento em recursos de hardware.

O descongestionamento do tráfego na rede é outra forma que se encontrou de melhorar o desempenho e a escalabilidade<sup>9</sup> em sistemas distribuídos por meio de sua abordagem única de cache de dados. Ao tirar vantagem da natureza multidimensional, o Caché reduz drasticamente o tráfego na rede. Menos tráfego significa que as aplicações ganham velocidade e podem atender mais usuários concorrentes (INTERSYSTEMS, 2007).

---

<sup>9</sup> Em uma rede ou em um processo e a capacidade de manipular porções crescentes de trabalho de forma uniforme, ou estar preparado para o crescimento do mesmo.

### 3.11 TRAVAMENTO LÓGICO

Em sistemas com milhares de usuários, a redução de conflitos entre processos concorrentes é crítica para conseguir alto fluxo de trabalho. Um dos maiores conflitos ocorre entre transações que desejam acessar os mesmos dados.

Os processos do Caché não travam páginas de dados inteiras enquanto fazem atualizações.

Ao invés disto, como as transações requerem acessos freqüentes ou mudanças de pequenas quantidades de dados, o travamento do banco de dados no Caché é feito ao nível lógico. Conflitos no banco de dados são ainda mais reduzidos usando operações de adição e de subtração atômicas que não requerem travamento. Estas operações são particularmente úteis no incremento de contadores usados para alocar números de ID ou modificar contadores de estatísticas.

O modelo de dados multidimensional transacional permite escalar aplicações baseadas no Caché até milhares de clientes sem sacrificar seu alto desempenho. Isso pode ser feito porque o acesso aos dados em um modelo multidimensional não é significativamente afetado pelo tamanho ou complexidade do banco de dados, diferentemente do modelo relacional. As transações podem acessar os dados de que necessitam sem executar complicadas junções ou saltando de tabela em tabela. O fato de o Caché usar travamento lógico para atualizações em vez de fechar páginas físicas é outra contribuição importante para a concorrência, assim como seu sofisticado cache de dados dentro da rede (INTERSYSTEMS, 2007).

#### 4 **BENCHMARK - AVALIAÇÃO DE DESEMPENHO EM BANCO DE DADOS**

Em sistemas informatizados é comum realizar-se medidas de desempenho, *benchmark*, com o objetivo de verificar capacidade, velocidade de acesso, respostas, confiança, segurança, e assim comparar diferentes tipos de hardware e software, das mais diversas áreas. Essas avaliações permitem a criação de padrões de desempenho. Vieira, Durães e Madeira (2005) apontam que devem possuir as seguintes características para serem considerados válidos:

- a) **representatividade:** devem representar sistemas reais, ou seja, não devem ser aplicados em sistemas simulados;
- b) **portabilidade:** devem ser portáveis para diferentes plataformas, proporcionando assim, comparativos de desempenhos em diferentes sistemas operacionais;
- c) **repetibilidade:** quando aplicado no mesmo ambiente, mais de uma vez, ele deve produzir resultados semelhantes;
- d) **escalabilidade:** essa característica prevê que os *benchmarks* realizem avaliações em ambientes com diferentes capacidades;
- e) **não intrusividade:** na necessidade de avaliar outro ambiente deve-se realizar o mínimo ou nenhuma alteração nesse novo ambiente;
- f) **simplicidade:** ser de fácil implementação e utilização pelos usuários que farão as avaliações.

Diversos *benchmarks* para realizar medidas de desempenho, nos mais variados sistemas, como servidores *WEB*, *SGBD*, aplicações gráficas, entre outros foram idealizados e implementados. Conforme Vieira, Durães e Madeira (2005) especificam e validam em sistemas transacionais. Entendem-se sistemas transacionais

como sistemas que possuem “motor transacional”, sendo seu maior representante os SGBD, o que faz deste o principal componente de um sistema OLTP (*On-Line Transaction Processing*). Na especificação de um *benchmark* de confiabilidade são apresentados os principais componentes que o constituem, os quais são:

- a) **medidas:** quantifica os resultados obtidos durante as avaliações. Podem ser subdivididas em duas classes: condicionais ou incondicionais. A primeira demonstra o funcionamento do sistema e são utilizadas para realizar comparações entre os ambientes, e são obtidas por meio das experiências realizadas. A segunda representa a confiabilidade global do sistema, como por exemplo, disponibilidade, integridade, entre outros;
- c) **workload:** é o responsável por medir o trabalho do sistema transacional, ou seja, é um sistema que trabalha junto com o SGBD para analisar seu desempenho;
- d) **faultload:** é utilizado quando o *benchmark* é voltado para verificar a confiabilidade de sistemas. Identifica e avalia as causas das falhas apresentadas no sistema, podendo ser considerado como um dos elementos mais complexos de serem identificados, devido às inúmeras possibilidades de elementos causadores e a complexidade das falhas;
- e) **procedimentos e regras:** são variáveis que dependem da finalidade do *benchmark*, algumas podem ser utilizadas de forma geral, como as regras para divulgação de resultados finais, construção e aplicação das análises de acordo com as especificações e igualdade de precisão das medidas nos testes realizados em diferentes ambientes;
- f) **ambiente experimental:** descreve completamente o ambiente e a plataforma necessária para execução das análises.

Vieira, Durães e Madeira (2005), também nos mostram quatro etapas para especificar um *benchmark* de confiabilidade, que são elas:

- a) identificação do domínio da aplicação;
- b) caracterização do SUB (*System Under Benchmark*);
- c) definição do componente de medida;
- d) definição dos outros componentes.

Identificar o domínio é necessário devido à grande quantidade de sistemas, com essa definição consegue-se mais facilmente identificar as aplicações que farão parte da análise e os componentes necessários, não se consegue identificar os componentes de avaliação sem conhecer o seu domínio, pois esses estão ligados diretamente aos métodos e funcionamentos do sistema.

A segunda etapa caracteriza-se pelos componentes do SUB, sendo eles as características e funcionalidades esperadas no domínio das aplicações definidas na primeira etapa.

A terceira e a quarta etapa podem ser reunidas em uma única, pois definem os componentes utilizados no *benchmark*. A diferença entre elas é que na terceira, são definidas as medidas que serão obtidas durante a avaliação. Na quarta serão definidas as outras partes dos componentes que podem depender diretamente das medidas definidas como, por exemplo, as falhas que ocorreram no sistema durante a avaliação, e devem ser expressas conforme as medidas definidas.

Pelo fato de a pesquisa de Vieira, Durães e Madeira (2005), ser voltada para avaliação de confiabilidade, as características descritas mostram genericamente, os componentes e etapas a serem seguidos no desenvolvimento de um *benchmark*.

## 5 DATA WAREHOUSE

Um *Data Warehouse* (DW) é utilizado para armazenar informações relativas ao funcionamento da organização em um BD, é um conjunto de dados que oferece apoio ao processo de tomada de decisão. Surgiram devido a necessidade de integração entre os ambientes organizacionais, aonde fossem armazenados e gerenciados de forma eficiente os dados produzidos pela organização.

Em circunstâncias normais o DW atende aos setores administrativos e alta gerencia, na criação e análise de relatórios, os dados ali presentes já foram previamente consolidados, sumarizados e depurados, para serem utilizados num ambiente orientado a consultas, possuem uma modelagem específica para essa finalidade.

As organizações possuem vários sistemas OLTP no ambiente operacional e seus dados fazem parte da estrutura empresarial, sendo detalhados, não redundantes, atualizáveis e refletem os valores de um determinado período atual. Contudo, os dados necessários para o processo de decisão são freqüentemente sumarizados, abrangendo um período extenso de tempo, não são atualizáveis e são redundantes para suportar diversas visões. Para fornecer um suporte a decisão, os dados relevantes são extraídos do meio operacional, filtrados, codificados e sumarizados, e aí sim são carregados no DW (SINGH, 2001, p.79).

Um *data warehousing* pode ser montado de acordo com o ambiente onde ele está inserido e conforme a sua utilização. Podem ser citadas como formas de implementação:

- a) *data warehouse* corporativo envolverá esforço, dinheiro e comprometimento por parte de todas as equipes envolvidas, e o coordenador

normalmente será o diretor de Tecnologia da Informação (TI) da organização;

b) *data warehouse* virtual constrói-se visões a partir das bases operacionais;

c) *data marts*, são pequenos *data warehouses*, geralmente podem ser por setores ou de outras formas, conforme a necessidade. *Data Marts* são menores, mais baratos e mais rápidos de serem implementados do que o um *data warehouse* corporativo. É considerada uma boa solução para empresas que necessitam deste tipo de sistema, mas que não possuem grandes recursos (ALMEIDA, E., 2004).

## 5.1 OTIMIZAÇÃO E DESEMPENHO

O desempenho do SGDB para um sistema de *data warehouse* é de fundamental importância, pois com base nisso é que a informação será ou não disponibilizada de maneira mais ágil tornando o processo de tomada de decisão útil em tempo hábil.

Para que uma melhora de desempenho seja alcançada, temos três pontos importantes que são: *cache* de dados, armazenamento e índices.

### 5.1.1 *Cache* de dados

Num sistema de *data warehouse*, grandes volumes de dados são processados durante todo o tempo gerando dessa maneira muita E/S. Processamento de E/S acarreta em uma baixa no desempenho e portando demora na recuperação de informações para o usuário final. Em todos os sistemas a comunicação entre o

processador e o armazenamento secundário (E/S) tem sido tradicionalmente reconhecido como o maior gargalo no desempenho. E em caso específico falando de SGBD esse gargalo ocorre porque na submissão de uma consulta é verificada, primeiramente, a possibilidade de resolvê-la em memória *cache*, se não for possível então o disco rígido é acessado (operação de E/S), após este acesso os dados são carregados em memória para então apresentar o resultado ao usuário (GARCIA-MOLLINA, ULLMAN e WINDOW).

Os *data warehouses* possuem como características consultas chamadas “*Adhoc*”<sup>10</sup>, que são flexíveis e criadas pelo usuário. Esta característica significa que poucas consultas serão repetidas, então a troca dos dados entre memória e disco rígido é muito intensa.

Diante dessa realidade é possível perceber que a otimização de acesso aos dados é de fundamental importância, e pode ser implementada através de *caches* de dados ou *buffers* que armazenam as informações recuperadas do disco por algum tempo.

Alguns parâmetros devem ser estudados para uma ótima configuração, como: tamanho deste *cache* e também como os objetos ficarão armazenados.

Os SGBD, assim como outros aplicativos utilizam a *cache* para armazenar os dados que mais serão utilizados ou que foram recuperados recentemente, dependendo da estratégia escolhida pelo administrador do banco de dados. Este tipo de área tem a finalidade de poupar operações de E/S, tornando todo o sistema mais rápido.

---

<sup>10</sup> Significa literalmente “para isto”, é quando uma consulta não é previamente preparada, ela é montada na hora de sua utilização.

### 5.1.2 Armazenamento de dados

A maneira como são alocados os dados em blocos ou paginas na disco rígido é outro ponto que também influencia no desempenho. Na criação do repositório de um BD OLTP é utilizado normalmente páginas de disco com tamanho variando entre 2Kb e 4Kb. Em um DW os valores podem ser maiores, sendo dimensionados a partir de 8Kb (ALMEIDA, E., 2004).

Em DW o tamanho é maior, pois não são utilizados dados “normalizados”, e assim sendo com a utilização de paginas maiores evita-se a leitura de mais de uma pagina, fenômeno chamado de encadeamento de linha.

Esse encadeamento acontece quando em uma recuperação de informação, o SGBD acessa mais de uma página de informação, para valores de uma mesma tupla, essa utilização desta forma podemos perceber que a utilização de paginas maiores poder ser indicada para leituras seqüenciais de linhas muito grandes (ALMEIDA, E., 2004).

O autor descreve que as páginas maiores melhoram o desempenho de leituras de índice. As páginas maiores podem manter mais entradas de índice em cada página, o que reduz o número de níveis dos índices grandes. Menos níveis de índice significam menos operações de E/S durante a passagem pelas ramificações do índice. Este aspecto se refere somente a bancos de dados de *data warehouse*, pois utilizar páginas grandes em bancos de dados OLTP pode gerar muitas disputas nas páginas de índice durante os processos de atualização. (JUELOT, apud ALMEIDA, E., 2004 p. 25).

Em um DW a normalização de dados irá interferir diretamente na eficiência de todo o sistema. Uma tabela de fato já é totalmente normalizada, mas a normalização das dimensões apenas irá aumentar o número de relacionamentos necessários para se obter uma resposta desejada de uma consulta requerida pelos usuários, prejudicando assim o tempo de resposta devido o aumento de processamento para executá-la.

Com isso diminui-se o espaço utilizado pelos dados, mas em contrapartida é necessário a criação de mais estruturas de indexação para auxiliar as consultas, o que acaba gerando perda de espaço físico. (ALMEIDA, A., 2006).

### 5.1.3 Índices

Índice é qualquer estrutura de dado que recebe como entrada uma propriedade de registros, em geral o valor de um ou mais campos, e encontra os registros com essa propriedade “rapidamente”. Em particular, um índice nos permite localizar um registro sem ter de examinar mais que uma pequena fração de todos os registros possíveis. O campo cujo valor se baseia no índice é chamado de chave de pesquisa, ou apenas chave (GARCIA-MOLINA; ULLMAN e WINDOM, 2001).

Índices são transparentes para aplicações e usuários, pois sua presença ou ausência não requer alteração de qualquer SQL, pode ser considerado como um atalho para a informação pesquisada (SILVA, 2002).

Como já foi referido se eles são como um atalho para as informações, consequentemente são os responsáveis pela melhora de desempenho dos sistemas que fazem uso dos mesmos, não só em projetos de BD mais também em *data warehouses*.

Quando em uma tabela defini-se os índices o SGBD vai percorrer a estrutura de índices, obtém o endereço da linha e apresenta o dado solicitado, caso não possua o índice a pesquisa terá que ser feita em toda a tabela utilizando assim mais tempo e maior processamento.

O segredo para aperfeiçoar o tempo de resposta na maior parte das consultas é o índice, e são usados num *data warehouse* justamente para esse fim (ALMEIDA, E., 2004).

### 5.1.3.1 Índices Tradicionais

Em um índice tradicional, uma propriedade da classe (coluna de uma tabela) possui uma lista de identidades dos registros, para cada valor da propriedade existe uma lista desses registros.

Como podemos observar na Tabela 3.

**Tabela 3. Índice tradicional**

| <b>Índice tradicional na propriedade “Tipo de Animal”</b> |                                     |
|-----------------------------------------------------------|-------------------------------------|
| Gato                                                      | 5,35,70,82,97,120,135,195,201...    |
| Cachorro                                                  | 1,3,7,92,125,197,205,305,401,502... |
| ...                                                       | ...                                 |

Fonte: INTERSYSTEMS (2002).

A Figura 2 mostra uma tabela de índices no cachê.

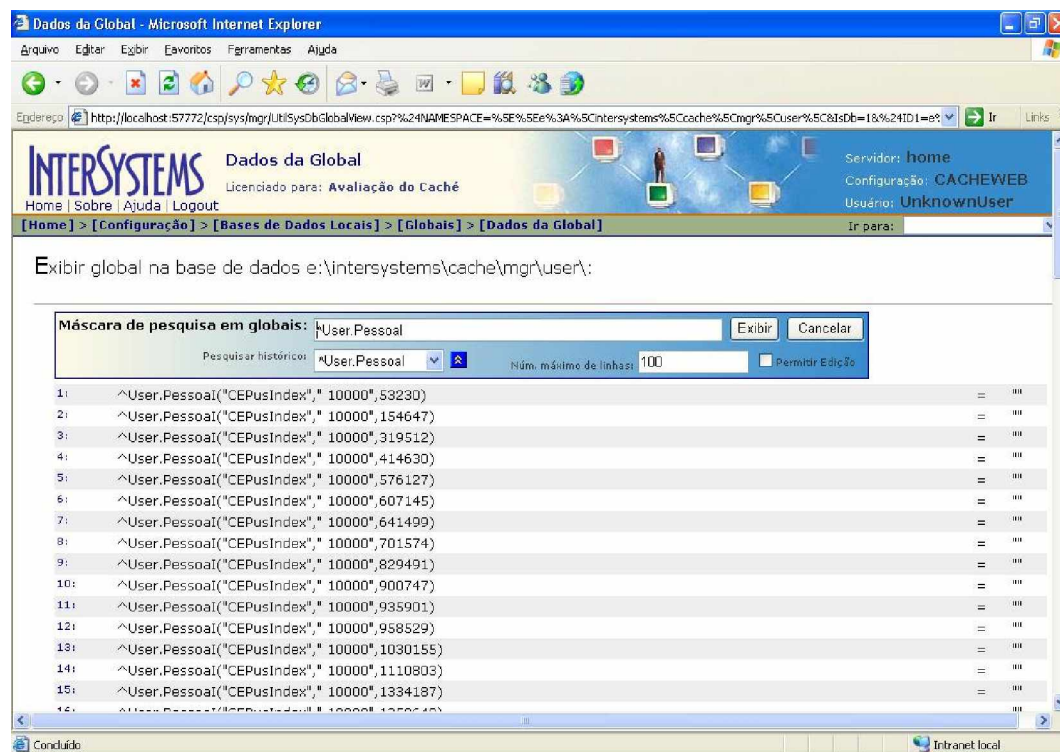


Figura 2. Índices Tradicionais no Cachê.

### 5.1.3.2 Índices de *Bitmap*

Os índices de bitmap também são chamados de índices de agrupamento, são frequentemente utilizados em *data warehouse*.

De acordo com o autor os índices de *bitmap* destinam-se a colunas de baixa cardinalidade, que contém um número limitado de valores. (JUEHOT, apud Almeida, E., 2004).

Alguns SGBD como: Oracle, IBM DB2 (*Universal Data Base*), Sybase e Informix, entre outros, implementam os índices de *bitmap* para uma cardinalidade de até 0,1% das linhas existentes na tabelas. Se tivermos numa tabela de 100.000 linhas o atributo de Unidade da Federação (UF) com 27 valores distintos, então, teremos uma cardinalidade de 27/100000 ou 0,027%, ou seja, para este atributo pode ser construído um índice de *bitmap*.

Um índice de *bitmap* (mapa de bits) é a utilização de um vetor de bits, aonde tem-se um (1) se o valor pertence a propriedade e zero (0) caso contrário (Garcia-Molina; Ulman e Window). A Tabela 4 mostra o exemplo de uma tabela Pessoa.

**Tabela 4. Tabela Pessoa.**

| Pessoa       |             |              |               |                  |     |
|--------------|-------------|--------------|---------------|------------------|-----|
| <i>RowID</i> | <i>Nome</i> | <i>Idade</i> | <i>Estado</i> | <i>Profissão</i> | ... |
| 1            | Smith       | 24           | NY            | Administrador    | ... |
| 2            | Jones       | 35           | NY            | Doutor           | ... |
| 3            | Presley     | 48           | CA            | Farmaceutico     | ... |
| 4            | Nixon       | 72           | NY            | Professor        | ... |
| ...          | ...         | ...          | ...           | ...              | ... |

**Fonte:** INTERSYSTEMS (2007).

No exemplo acima, a criação de um índice de *bitmap* simples para a propriedade estado, ficaria como apresentado na Tabela 5 a baixo.

**Tabela 5. Índice de Bitmap na propriedade Estado.**

|     | EstadoIndexBitmap |      |      |      |     |
|-----|-------------------|------|------|------|-----|
|     | Row1              | Row2 | Row3 | Row4 | ... |
| CA  | 0                 | 0    | 1    | 0    | ... |
| NY  | 1                 | 1    | 0    | 1    | ... |
| WY  | 0                 | 0    | 0    | 0    | ... |
| ... | ...               | ...  | ...  | ...  | ... |

Fonte: INTERSYSTEMS (2007).

Como pode ser observado na tabela 5, o índice de *bitmap* cria um vetor de zeros e uns, aonde cada posição desse vetor indica uma linha da tabela, e respectivamente se ele possui ou não o valor do índice.

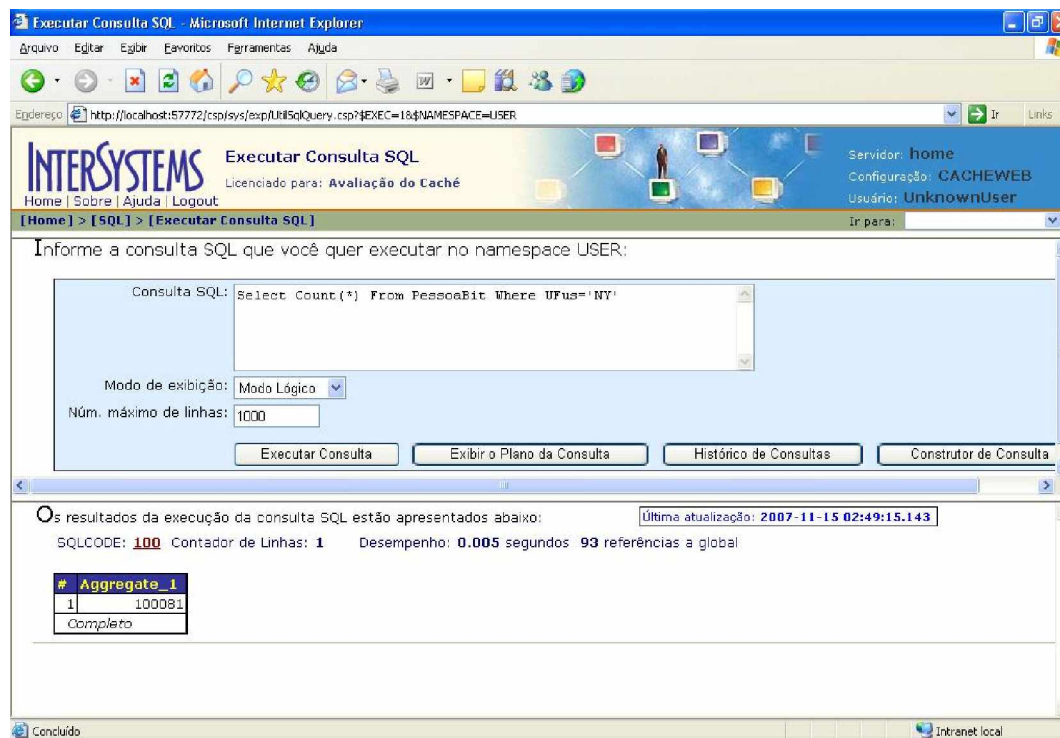
Se executássemos um comando SQL simples, por exemplo:

```
Select * From Pessoa Where Estado = 'NY'
```

O SGBD iria percorrer apenas a linha aonde está indexado a propriedade NY e nos mostraria os dados contidos nas colunas aonde for encontrado o *bit* de valor um.

Um comando SQL que conte os valores aonde qual o número de pessoas mora em *Nova York* (NY) seria: `Select Count(*) From Pessoa Where Estado = 'NY'`

Esse comando iria apenas percorrer o vetor NY, contando no número de uns e apresentando esse total. Na Figura 3, podemos observar que foi executado esse comando em uma tabela com indexação de *bitmap*.



**Figura 3. Execução do comando SQL na tabela com índices de bitmap.**

A Figura 3 nos mostra a execução do comando e o tempo que o SGBD levou para fazer a contagem de uns no vetor NY, sendo que o tempo gasto foi de 0,005 segundos, e encontrou 100.081 pessoas que residem em *Nova York*.

Na verdade como pode ser observado esse tempo foi apenas necessário para percorrer e contar os uns, nada mais.

Um índice de *bitmap* para a propriedade idade ficaria como mostra a Tabela 6.

**Tabela 6. Índice de *bitmap* na propriedade idade.**

|     | IdadeIndexBitmap |      |      |      |     |
|-----|------------------|------|------|------|-----|
|     | Row1             | Row2 | Row3 | Row4 | ... |
| 24  | 1                | 0    | 0    | 0    | ... |
| 35  | 0                | 1    | 0    | 0    | ... |
| 48  | 0                | 0    | 1    | 0    | ... |
| ... | ...              | ...  | ...  | ...  | ... |

**Fonte:** INTERSYSTEMS (2007)

Uma das características do *bitmap* é a utilização de operações *booleanas* (AND, OR) durante as consultas, no caso em consultas aonde mais de uma propriedade sendo pesquisada, por exemplo: `Select * From Pessoa Where UF='NY' AND Idade= 35`

A Tabela 7 mostra como seria essa realizada essa consulta pelo SGBD.

**Tabela 7: Resultado da operação AND em índices de bitmap.**

| EstadoIndexBitmap |   |   |   |   |     |
|-------------------|---|---|---|---|-----|
| NY                | 1 | 1 | 0 | 1 | ... |
| AND               |   |   |   |   |     |
| IdadeIndexBitmap  |   |   |   |   |     |
| 35                | 0 | 1 | 0 | 0 | ... |
| =                 |   |   |   |   |     |
| Resultado         | 0 | 1 | 0 | 0 | ... |

**Fonte:** INTERSYSTEMS (2007).

Nesse caso o mecanismo SQL utiliza as duas propriedades como pode ser visto na Tabela 7, e realiza a operação *booleana* (AND) resultando em um novo vetor, no qual contém o resultado para a consulta solicitada.

## 6 BENCHMARK INDICE TRANSACIONAL DE *BITMAP* FRENTE INDEXAÇÃO TRADICIONAL

Com o objetivo de verificar se a indexação de *bitmap* melhora o desempenho em consultas foi montando duas classes (tabelas), uma utilizando indexação de *bitmap* e a outra utilizando indexação tradicional.

### 6.1 METODOLOGIA

Todos os testes foram realizados em um único computador com a configuração conforme a Tabela 8.

**Tabela 8. Configuração do computador utilizado nos testes**

| <b>Item</b> | <b>Configuração</b>          |
|-------------|------------------------------|
| CPU         | AMD Athlon XP 1600+ 1.06 ghz |
| Memória     | 480 mb                       |
| HD 1        | 32 gb                        |
| HD 2        | 60 gb                        |
| SO          | Windows XP spk 2             |

O software que foi utilizado em todos os testes foi o Banco de Dados Pós-Relacional Caché na versão 2007.01, o banco requer a utilização de um servidor web, o servidor utilizado foi o Internet Information Server, que vem junto com Windows XP, porém poderia ter sido utilizado outro servidor web.

As classes foram desenvolvidas no Studio, uma ferramenta nativa do banco de dados Caché, que é utilizada tanto para a criação de classes como programação em COS, modelagem de consultas, criação de páginas CSP.

Nas classes foram criadas nove colunas e para sete delas foram definidos índices, tanto de *bitmap*, como tradicional – todas menos ID que é única e Nome que é

quase única. (Os índices de *bitmap* geralmente não são recomendados para colunas que têm um número muito grande de valores únicos.) A Figura 4 mostra a classe PessoaBit que utiliza indexação de *bitmap*.

```

/// Classe implementada com índice de bitmap
Class User.PessoaBit Extends (%Persistent, %Populate, %XML.Adaptor)
{

// profissão

Property Profissao As %String(POPSPEC = "Title()");

Index ProfissaoIndex On Profissao [ Type = bitmap ];

Property Sexo As %String(MAXLEN = 1, POPSPEC =
"ValueList("",M,F)");

Index SexoIndex On Sexo [ Type = bitmap ];

// salario

Property Salario As %Currency(POPSPEC = "Currency(5,10)");

Index SalarioIndex On Salario [ Type = bitmap ];

Property Idade As %Integer(POPSPEC = "Integer(15,80)");

Index IdadeIndex On Idade [ Type = bitmap ];

// COD POSTAL

Property CEPus As %String(MAXLEN = 10, POPSPEC = "USZip()");

Index CEPusIndex On CEPus [ Type = bitmap ];

// cor do cabelo

Property CorCabelo As %String(MAXLEN = 10, POPSPEC =
"ValueList("",Loiro,Preto,Branco,Castanho,Vermelho)");

Index CorCabeloIndex On CorCabelo [ Type = bitmap ];

//nome
Property Nome As %String(POPSPEC = "Name("" As %String = "")");
// estado
Property UFus As %String(MAXLEN = 2, POPSPEC = "USState()");

Index UFusIndex On UFus [ Type = bitmap ];
}

```

Figura 4. Definição da Classe PessoaBit.

A Figura 5 mostra como foi implementada a classe Pessoa, que utiliza indexação tradicional.

```

///
Class User.Pessoa Extends (%Persistent, %Populate, %XML.Adaptor)
{

// profissão
Property Profissao As %String(POPSPEC = "Title()");

Index ProfissaoIndex On Profissao;

Property Sexo As %String(MAXLEN = 1, POPSPEC = "ValueList('','M,F')");

Index SexoIndex On Sexo;

// salario
Property Salario As %Currency(POPSPEC = "Currency(5,10)");

Index SalarioIndex On Salario;

Property Idade As %Integer(POPSPEC = "Integer(15,80)");

Index IdadeIndex On Idade;

// COD POSTAL
Property CEPus As %String(MAXLEN = 10, POPSPEC = "USZip()");
Index CEPusIndex On CEPus;

// cor do cabelo
Property CorCabelo As %String(MAXLEN = 10, POPSPEC =
"ValueList('','Loiro,Preto,Branco,Castanho,Vermelho')");
Index CorCabeloIndex On CorCabelo;

//nome
Property Nome As %String(POPSPEC = "Name('' As %String = '')");

// estado
Property UFus As %String(MAXLEN = 2, POPSPEC = "USState()");

Index UFusIndex On UFus;

}

```

**Figura 5. Definição da Classe Pessoa.**

As duas classes após montadas foram preenchidas com dados utilizando a classe %Populate (essa classe permite a população automática da base de dados), em cada propriedade foi passado como parâmetro que tipo de população de dados deveria ser utilizado. A população de dados foi feita utilizando o seguinte comando:

- a) Classe PessoaBit: Do ##Class(PessoaBit).Populate(5000000);
- b) Classe Pessoa: Do ##Class(Pessoa).Populate(5000000);

Como pode ser observado no comando de população de dados as tabelas foram preenchidas com cinco milhões de linhas. A Figura 6 mostra uma consulta contando o número de linhas da tabela PessoaBit.

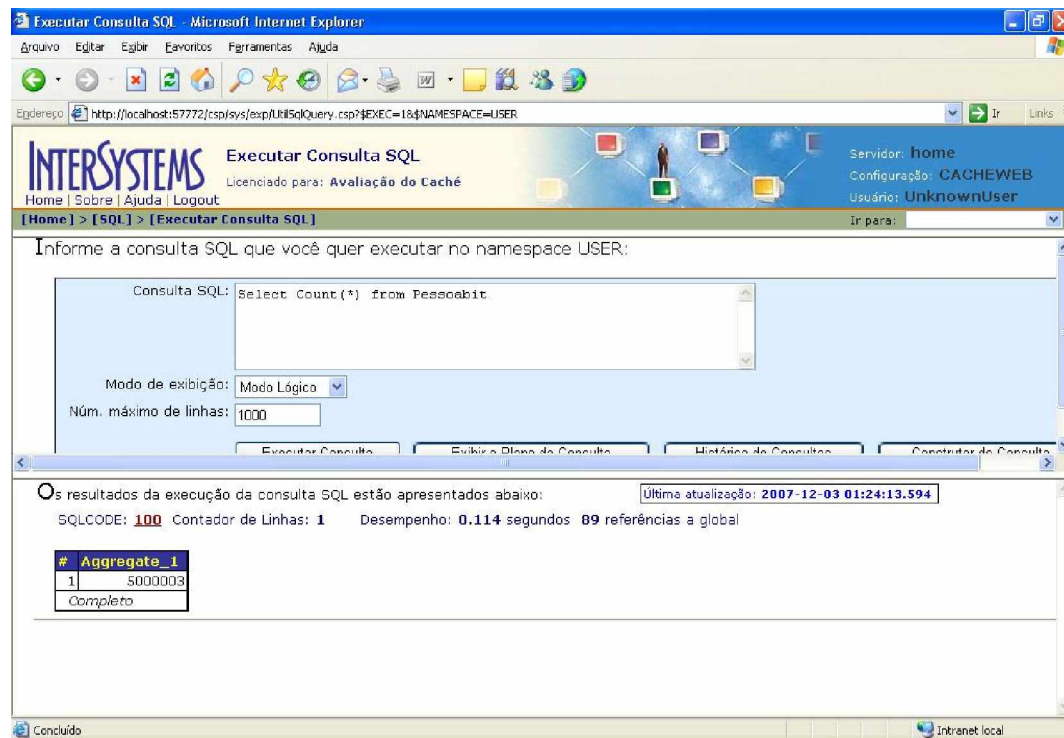


Figura 6. Total de linhas na tabela PessoaBit.

A Figura 7 uma consulta contando o numero de linhas da tabela Pessoa.

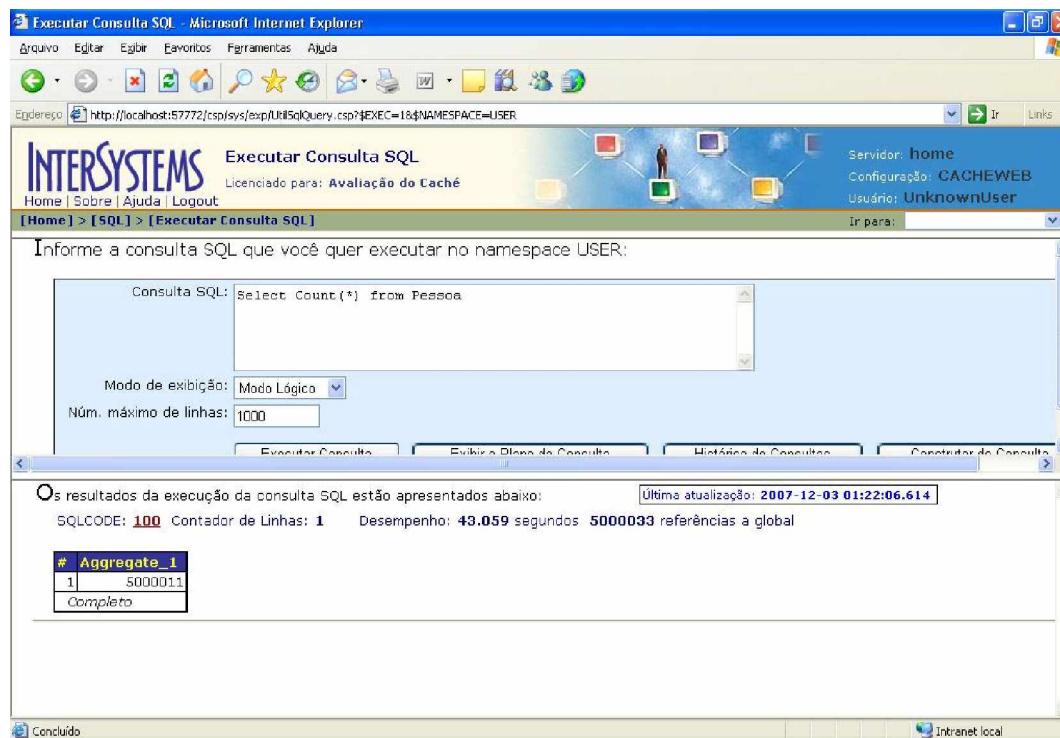


Figura 7. Total de linhas na tabela Pessoa.

A Figura 8 mostra alguns dados que a população automática gerou.

The screenshot shows a web browser window titled 'Abrir Tabela - Microsoft Internet Explorer'. The address bar displays a URL: `http://localhost:57772/csp/sys/exp/UHISqlQuery.csp?EXEC=18&NAMESPACE=USER`. The page header includes the 'INTERSYSTEMS' logo and the title 'Abrir Tabela'. Below the header, there is a navigation bar with links: 'Atualizar' and 'Fechar Janela'. The main content area is titled 'SQLUser.Pessoa no namespace USER'. It displays a table with 27 rows and 9 columns. The columns are: '#', 'ID', 'CEPus', 'CorCabelo', 'Idade', 'Nome', 'Profissao', 'Salario', 'Sexo', and 'UFus'. The status bar at the bottom indicates 'Concluído' and 'Intranet local'.

| #  | ID | CEPus | CorCabelo | Idade | Nome                 | Profissao                    | Salario | Sexo | UFus |
|----|----|-------|-----------|-------|----------------------|------------------------------|---------|------|------|
| 1  | 1  | 13942 | Castanho  | 80    | Davis, Emily L.      | Laboratory Marketing Manager | 7.0450  | F    | GA   |
| 2  | 2  | 51254 | Loiro     | 49    | Quixote, Julie Y.    | Strategic Resources Director | 9.0993  | M    | NV   |
| 3  | 3  | 32782 | Loiro     | 75    | Zevon, Valery A.     | Senior Marketing Manager     | 9.1719  | M    | AZ   |
| 4  | 4  | 98791 | Loiro     | 16    | Love, Sophia H.      | Strategic Developer          | 6.6486  | M    | KS   |
| 5  | 5  | 63866 | Castanho  | 54    | Uberoth, Usha O.     | Strategic Support Engineer   | 6.5438  | F    | IA   |
| 6  | 6  | 96442 | Castanho  | 42    | Dunlap, Barbara X.   | Laboratory Developer         | 6.6367  | F    | GA   |
| 7  | 7  | 49859 | Preto     | 64    | Quilty, Hannah W.    | Engineer                     | 9.7176  | M    | KY   |
| 8  | 8  | 19902 | Castanho  | 22    | Ingleman, Wilma X.   | Global Product Manager       | 5.0426  | M    | MO   |
| 9  | 9  | 34260 | Preto     | 54    | Nichols, Lola M.     | Executive Sales Rep.         | 5.7276  | M    | MT   |
| 10 | 10 | 28162 | Preto     | 80    | Gold, Nataliya U.    | Senior Administrator         | 9.9853  | F    | MA   |
| 11 | 11 | 62202 | Branco    | 43    | Adam, Laura G.       | Executive Developer          | 6.4932  | M    | DE   |
| 12 | 12 | 92219 | Loiro     | 79    | Jaynes, Barb G.      | Associate Engineer           | 7.0744  | F    | MI   |
| 13 | 13 | 96101 | Branco    | 17    | Solomon, Pat F.      | Executive Sales Rep.         | 5.8970  | F    | LA   |
| 14 | 14 | 49814 | Loiro     | 53    | Gibbs, Christen G.   | Global Developer             | 6.5352  | M    | AZ   |
| 15 | 15 | 49864 | Loiro     | 27    | Avery, Filomena N.   | Accounts Rep.                | 8.4850  | F    | WA   |
| 16 | 16 | 88593 | Vermelho  | 70    | Grabscheid, Olga Y.  | Assistant Accountant         | 7.0170  | F    | CO   |
| 17 | 17 | 37755 | Preto     | 60    | O'Rielly, Barbara F. | Executive Director           | 7.8251  | M    | VT   |
| 18 | 18 | 63886 | Branco    | 27    | Clay, Lola F.        | Systems Engineer             | 9.8577  | M    | NY   |
| 19 | 19 | 98879 | Vermelho  | 40    | Woo, Laura S.        | Global WebMaster             | 7.9211  | M    | VA   |
| 20 | 20 | 15985 | Branco    | 79    | Goncharuk, Liza M.   | Strategic Administrator      | 9.5853  | M    | NE   |
| 21 | 21 | 73830 | Castanho  | 68    | Macrakis, Lola I.    | Laboratory Accountant        | 6.3060  | M    | SC   |
| 22 | 22 | 23077 | Vermelho  | 30    | Faust, Emma R.       | Associate Resources Director | 5.3734  | M    | KS   |
| 23 | 23 | 38878 | Branco    | 36    | Hills, Gertrude C.   | Executive Systems Engineer   | 9.5416  | F    | MO   |
| 24 | 24 | 89554 | Castanho  | 39    | Hammel, Terry D.     | Global Systems Engineer      | 9.9102  | F    | WA   |
| 25 | 25 | 17864 | Castanho  | 62    | Isaacs, Van B.       | Executive Marketing Manager  | 6.1684  | F    | GA   |
| 26 | 26 | 84955 | Preto     | 67    | Cooper, Belinda U.   | Associate Accounts Rep.      | 6.4142  | F    | AR   |
| 27 | 27 | 94859 | Branco    | 24    | Javnes, Michelle R.  | Associate Technician         | 8.3104  | F    | KS   |

Figura 8. Dados que foram gerados automaticamente.

Após a população de dados foi feita uma análise to tamanho de cada uma das tabelas incluindo também as de índices, a Tabela 9 mostra o tamanho em *megabytes*.

**Tabela 9. Tamanho das tabelas**

| <b>Tabela</b> | <b>Tamanhos</b> |
|---------------|-----------------|
| PessoaBitD    | 373 mb          |
| PessoaD       | 373 mb          |
| PessoaBitI    | 130 mb          |
| PessoaI       | 270 mb          |
| Total         | 1146 mb         |

A Figura 9 mostra algumas propriedades da tabela PessoaD, assim como o tamanho em *megabytes*.



**Figura 9. Propriedades da tabela PessoaD**

Após a montagem das tabelas, e os dados inseridos iniciou-se a bateria de testes, cada comando SQL foi repetido vinte e uma vezes, todos os resultados foram catalogados em planilhas que estão anexadas ao final do trabalho, após essa coleta de tempos foi feito a média de cada consulta, os tempos médios das consultas podem ser analisados na Tabela 10.

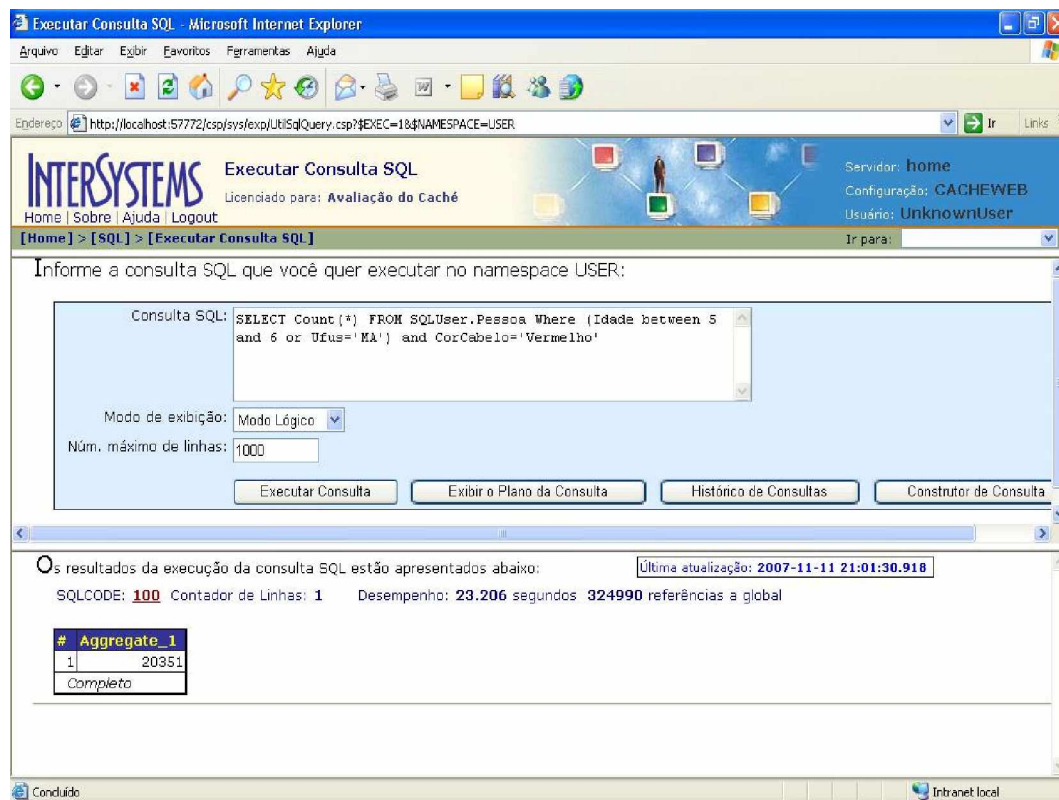
Tabela 10. Comparativo entre índice *bitmap* e índice tradicional.

| Nr. Consulta | Bitmap                                                                                                     |           |                | Tradicional                                                                                                  |           |                |
|--------------|------------------------------------------------------------------------------------------------------------|-----------|----------------|--------------------------------------------------------------------------------------------------------------|-----------|----------------|
|              | Comando                                                                                                    | Registros | Tempo segundos | Comando                                                                                                      | Registros | Tempo segundos |
| 1            | SELECT * FROM SQLUser.PessoaBit                                                                            | 1.000     | 0,553          | SELECT * FROM SQLUser.Pessoa                                                                                 | 1.000     | 0,544          |
| 2            | SELECT * FROM SQLUser.PessoaBit                                                                            | 5.000     | 2,086          | SELECT * FROM SQLUser.Pessoa                                                                                 | 5.000     | 2,180          |
| 3            | SELECT * FROM SQLUser.PessoaBit                                                                            | 10.000    | 4,268          | SELECT * FROM SQLUser.Pessoa                                                                                 | 10.000    | 4,277          |
| 4            | SELECT count(*) FROM SQLUser.PessoaBit Where Ufus='MA'                                                     | 100.221   | 0,168          | SELECT count(*) FROM SQLUser.Pessoa Where Ufus='MA'                                                          | 100.365   | 1,116          |
| 5            | SELECT count(*) FROM SQLUser.PessoaBit Where Ufus='MA' Order By 1 DESC                                     | 100.221   | 0,023          | SELECT count(*) FROM SQLUser.Pessoa Where Ufus='MA' Order By 1 DESC                                          | 100.365   | 0,997          |
| 6            | SELECT count(*) FROM SQLUser.PessoaBit Where Sexo='F'                                                      | 2.500.376 | 0,175          | SELECT count(*) FROM SQLUser.Pessoa Where Sexo='F'                                                           | 2.501.460 | 19,840         |
| 7            | SELECT count(*) FROM SQLUser.PessoaBit Where Sexo='M'                                                      | 2.499.624 | 0,048          | SELECT count(*) FROM SQLUser.Pessoa Where Sexo='M'                                                           | 2.498.540 | 19,927         |
| 8            | SELECT count(*) FROM SQLUser.PessoaBit Where Idade < 40                                                    | 1.893.274 | 0,343          | SELECT count(*) FROM SQLUser.Pessoa Where Idade < 40                                                         | 1.893.883 | 14,106         |
| 9            | SELECT count(*) FROM SQLUser.PessoaBit Where Idade between 10 and 40                                       | 1.968.955 | 0,344          | SELECT count(*) FROM SQLUser.Pessoa Where Idade between 10 and 40                                            | 1.969.284 | 14,748         |
| 10           | SELECT Nome FROM SQLUser.PessoaBit Where Ufus='MA' and Idade = 50                                          | 1.524     | 0,515          | SELECT Nome FROM SQLUser.Pessoa Where Ufus='MA' and Idade = 50                                               | 1.542     | 2,265          |
| 11           | SELECT Nome FROM SQLUser.PessoaBit Where (Idade between 5 and 6 or Ufus='MA') and CorCabelo='Vermelho'     | 19.865    | 22,572         | SELECT Nome FROM SQLUser.Pessoa Where (Idade between 5 and 6 or Ufus='MA') and CorCabelo='Vermelho'          | 20.351    | 32,457         |
| 12           | SELECT Count(*) FROM SQLUser.PessoaBit Where (Idade between 5 and 6 or Ufus='MA') and CorCabelo='Vermelho' | 19.865    | 14,563         | SELECT Nome Count(*) FROM SQLUser.Pessoa Where (Idade between 5 and 6 or Ufus='MA') and CorCabelo='Vermelho' | 20.351    | 22,209         |
|              | Totais                                                                                                     | 9.119.925 | 45,658         |                                                                                                              | 9.122.141 | 134,666        |

Nas três primeiras consultas, como pode ser observado, não existe diferença significativa nos tempos, não foi feita referencia a indexação e foi colocado um numero máximo de linhas a serem pesquisados, isso ocorreu devido a todas essas consultas

utilizarem as mesmas estruturas de dados, a única modificação aplicada foi a tabela na qual seria realizada a consulta.

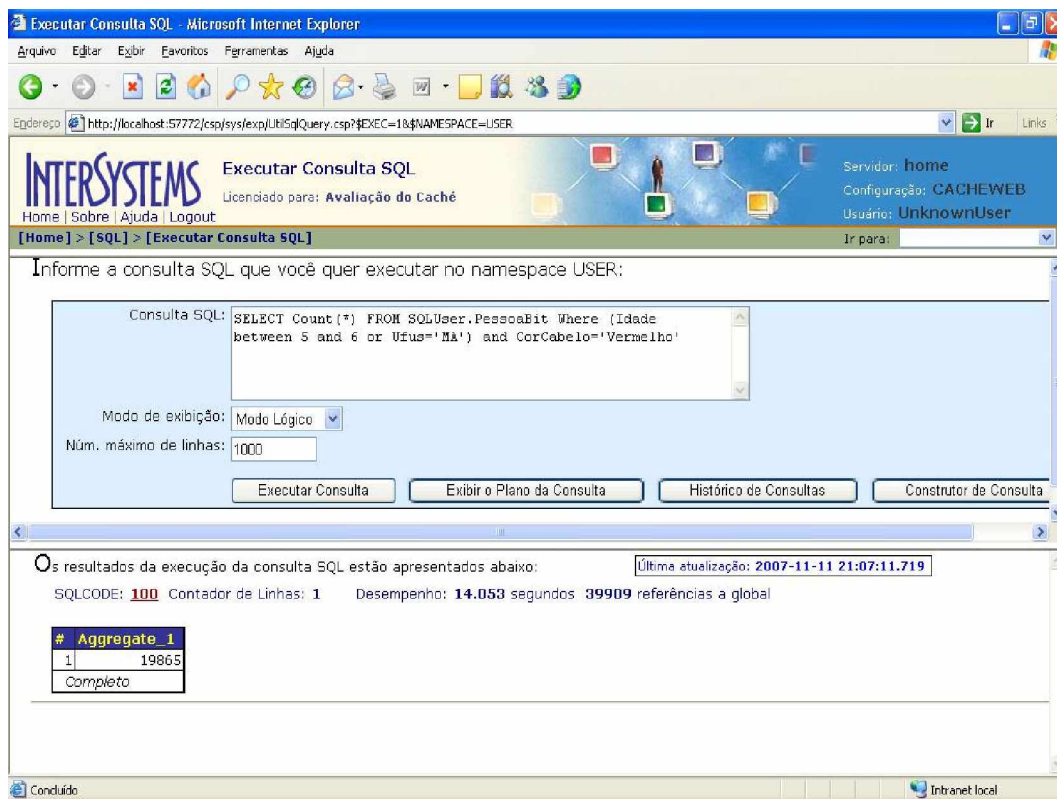
O tempo gasto em cada consulta é retornado pelo BD após a conclusão da mesma, como pode ser observado na Figura 10.



**Figura 10. Tempo gasto pela consulta 12 na tabela Pessoa.**

Na Figura 10, podemos observar o número de linhas que foram contadas, e o tempo que foi gasto para que o banco percorresse toda a tabela, e trouxesse o número de linhas que satisfazem o comando.

Na Figura 11, executamos a mesma consulta porem na tabela PessoaBit, utilizando a indexação de *bitmap*.



**Figura 11. Tempo gasto pela consulta 12 na tabela PessoaBit.**

Como pode ser observado através da Figura 9 e da Figura 10, aonde foi realizada as consultas mais complexas dos testes a indexação de *bitmap* foi 1,66 vezes mais rápida que a indexação tradicional.

Temos na Tabela 11 um resumo dos tempos das consultas.

**Tabela 11. Resumo dos tempos das consultas.**

| <b>Consulta</b> | <b>Bitmap</b> | <b>Tradicional</b> | <b>Bitmap X Tradicional</b> |
|-----------------|---------------|--------------------|-----------------------------|
| 1               | 0,533         | 0,544              | 1,021                       |
| 2               | 2,086         | 2,180              | 1,045                       |
| 3               | 4,268         | 4,277              | 1,002                       |
| 4               | 0,168         | 1,116              | 6,643                       |
| 5               | 0,023         | 0,997              | 43,348                      |
| 6               | 0,175         | 19,840             | 113,371                     |
| 7               | 0,048         | 19,927             | 415,146                     |
| 8               | 0,343         | 14,106             | 41,125                      |
| 9               | 0,344         | 14,748             | 42,872                      |
| 10              | 0,515         | 2,265              | 4,398                       |
| 11              | 22,572        | 32,457             | 1,438                       |
| 12              | 14,563        | 22,209             | 1,525                       |
| <b>Totais</b>   | <b>45,638</b> | <b>134,666</b>     |                             |

Podemos verificar na Tabela 11, com o somatório dos tempos de cada índice, que a indexação de *bitmap* chegou a ser 2,95 vezes mais rápida que a indexação tradicional, mostrando-se claramente mais eficiente em alguns casos, e menos em outros, porém em todos com superioridade, isso nos deixa claro que esse tipo de índice pode e deve ser utilizado sempre que possível, pois melhora a *performance* das aplicações de uma maneira significativa e justifica o porque de sua utilização em sistemas de *data warehouse*.

No decorrer de todos os testes observamos que a primeira consulta executada, após a inicialização do banco, retornava um tempo maior, tanto na tabela com indexação de *bitmap*, como na de índices tradicionais, porém, após a primeira consulta na tabela com índices de *bitmap*, o tempo de resposta das consultas diminuía drasticamente, e na tabela com a indexação tradicional isso não ocorria.

Podemos atribuir essa característica a memória cachê do banco, e foi observado que a indexação de *bitmap* utilizava essa área de memória de uma maneira mais eficientemente que a indexação tradicional.

Na Tabela 12 temos os tempos gastos na primeira execução de cada comando em ambas as classes.

**Tabela 12 Tempo gasto na primeira consulta de cada classe.**

| <b>Consulta</b> | <b>Bitmap</b> | <b>Tradicional</b> | <b>Bitmap X Tradicional</b> |
|-----------------|---------------|--------------------|-----------------------------|
| 1               | 0,535         | 0,581              | 1,086                       |
| 2               | 2,851         | 2,444              | 0,857                       |
| 3               | 4,234         | 4,581              | 1,082                       |
| 4               | 3,491         | 3,850              | 1,103                       |
| 5               | 0,363         | 0,977              | 2,691                       |
| 6               | 3,473         | 19,672             | 5,664                       |
| 7               | 0,698         | 19,753             | 28,299                      |
| 8               | 5,387         | 14,164             | 2,629                       |
| 9               | 5,547         | 14,548             | 2,623                       |
| 10              | 2,922         | 9,066              | 3,103                       |
| 11              | 21,225        | 31,567             | 1,487                       |
| 12              | 15,518        | 22,635             | 1,459                       |
| Total           | 66,244        | 143,838            |                             |

A Figura 12 mostra a execução de uma consulta na tabela PessoaBit, após a inicialização do banco, sendo essa a primeira consulta.

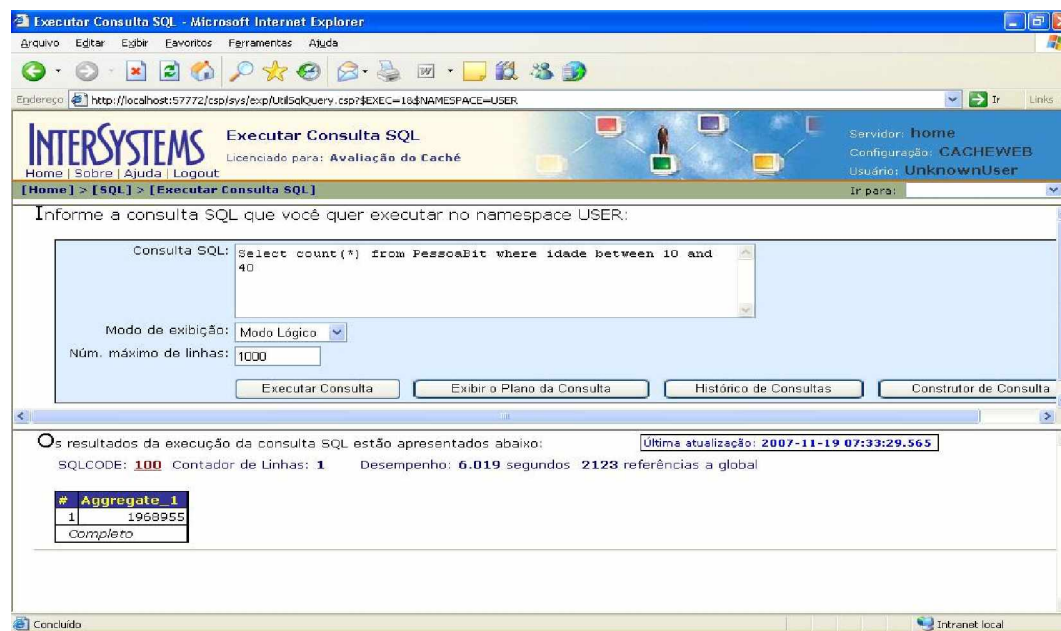


Figura 12. Tempo gasto na execução da primeira consulta após a inicialização do banco tabela PessoaBit.

A Figura 13 mostra o tempo que levou a segunda consulta.

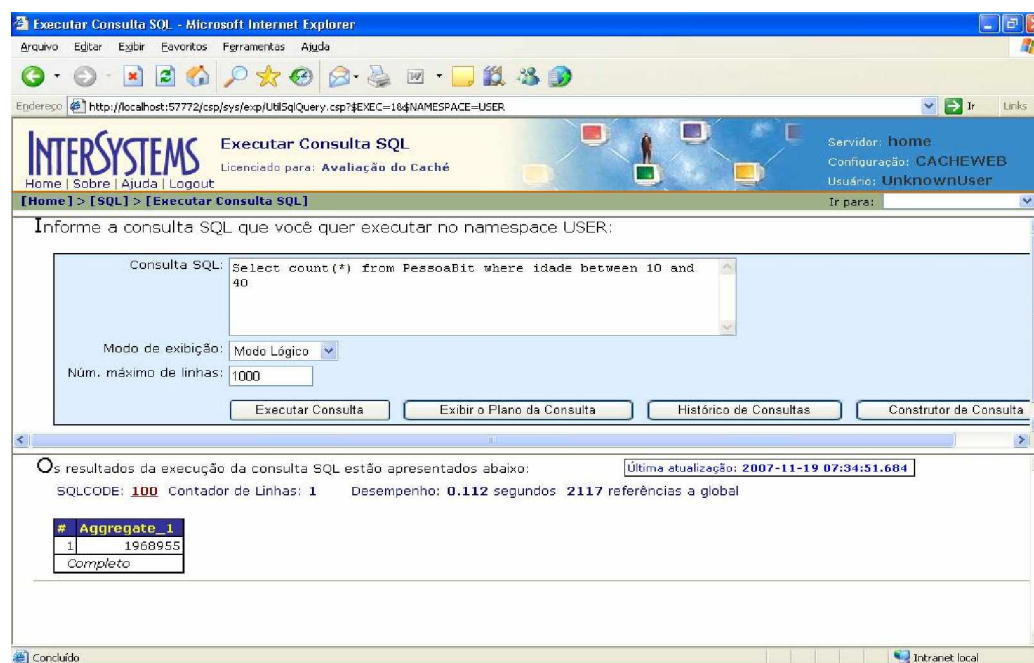
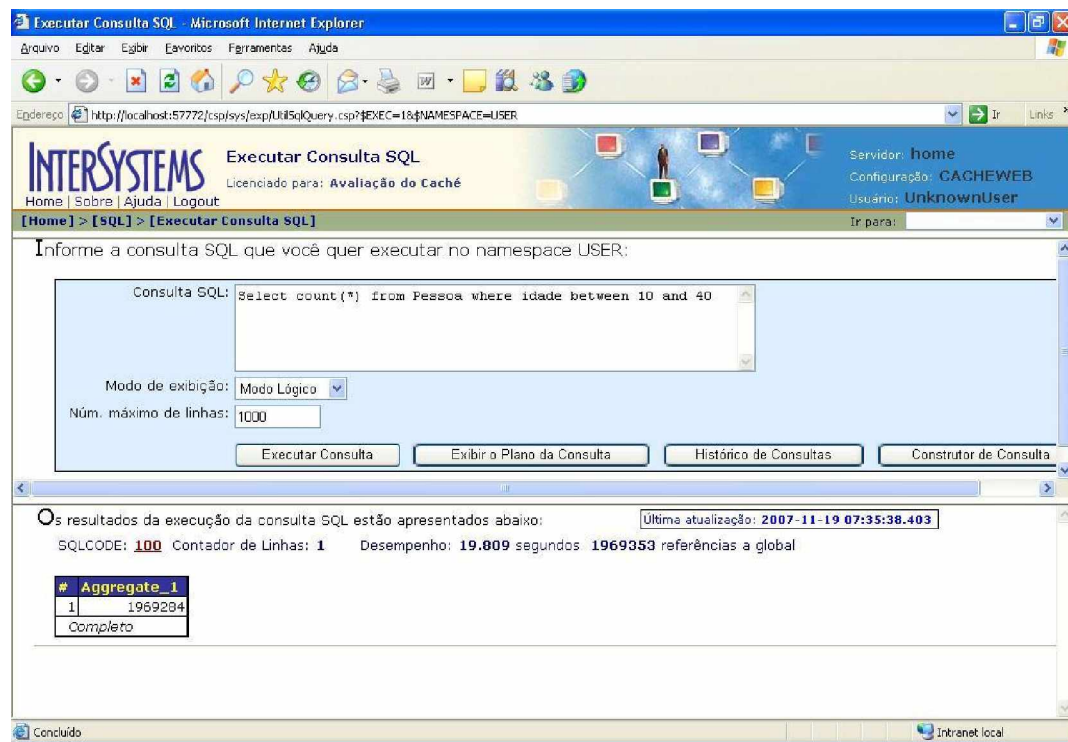


Figura 13. Tempo gasto na execução da segunda consulta após a inicialização do banco, tabela PessoaBit.

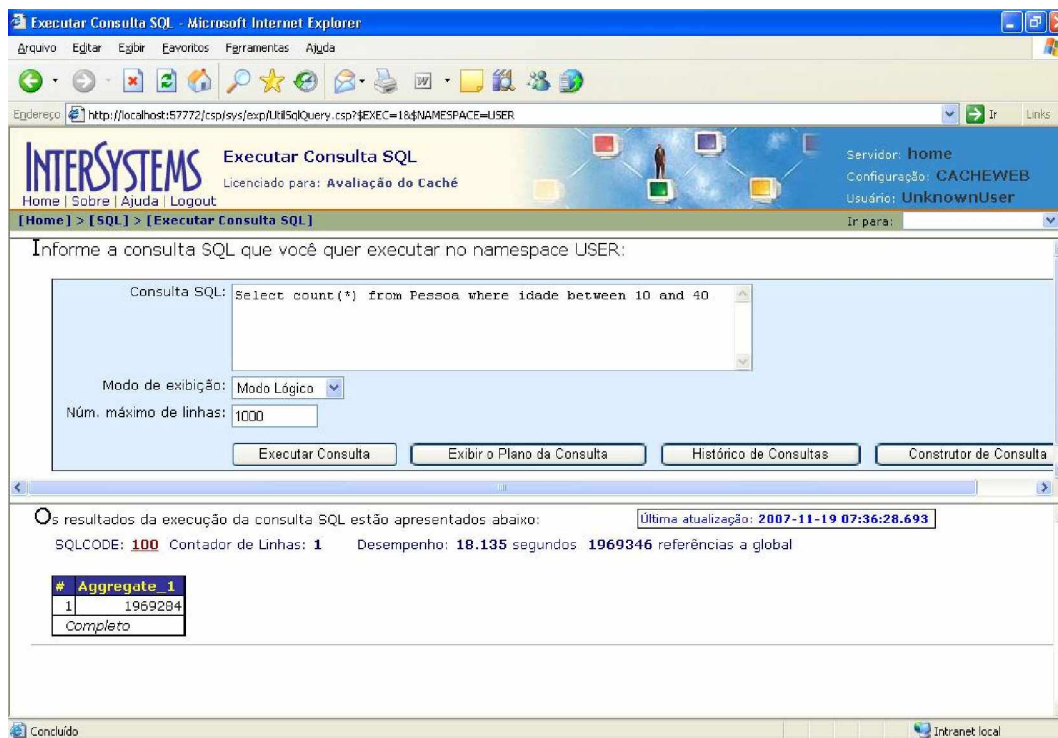
Como pode ser observado pelas Figuras 12 e 13, o tempo da segunda consulta reduziu extremamente, com relação à primeira, utilizando a indexação de *bitmap*.

Na Figura 14 temos a mesma consulta sendo executada após a inicialização do banco na tabela Pessoa, que utiliza indexação tradicional.



**Figura 14.** Tempo gasto na execução da primeira consulta após a inicialização do banco, tabela PessoaBit.

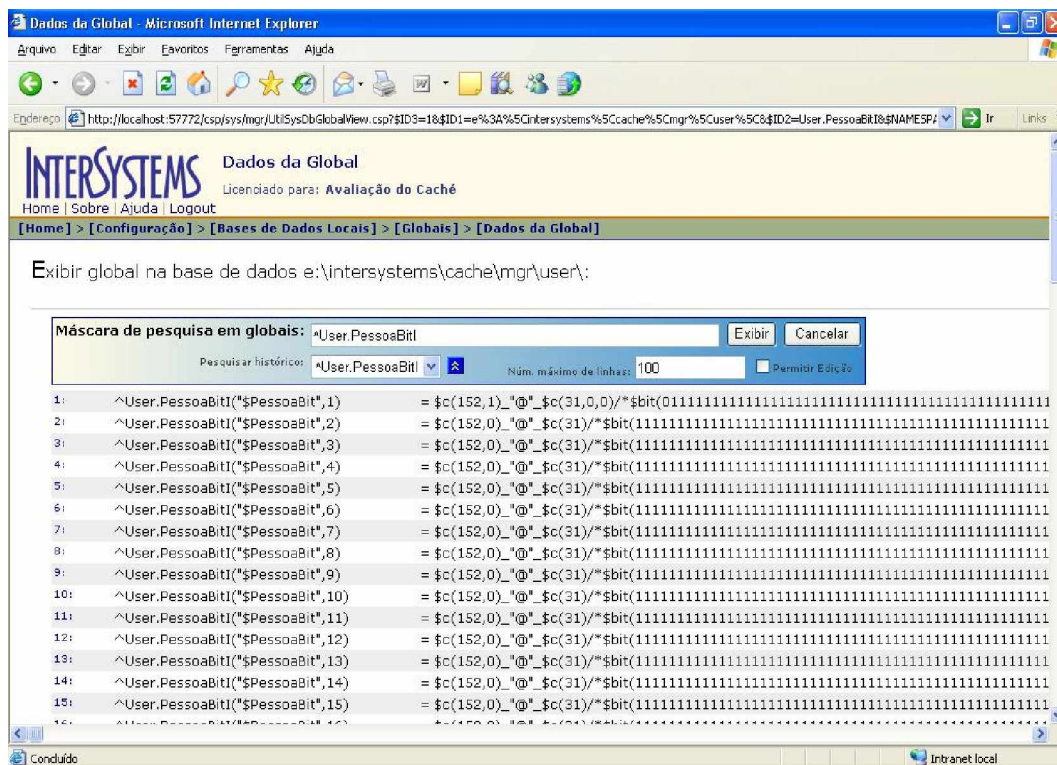
Na Figura 15 temos a execução da segunda consulta utilizando a indexação tradicional.



**Figura 15.** Tempo gasto na execução da segunda consulta após a inicialização do banco, tabela Pessoa.

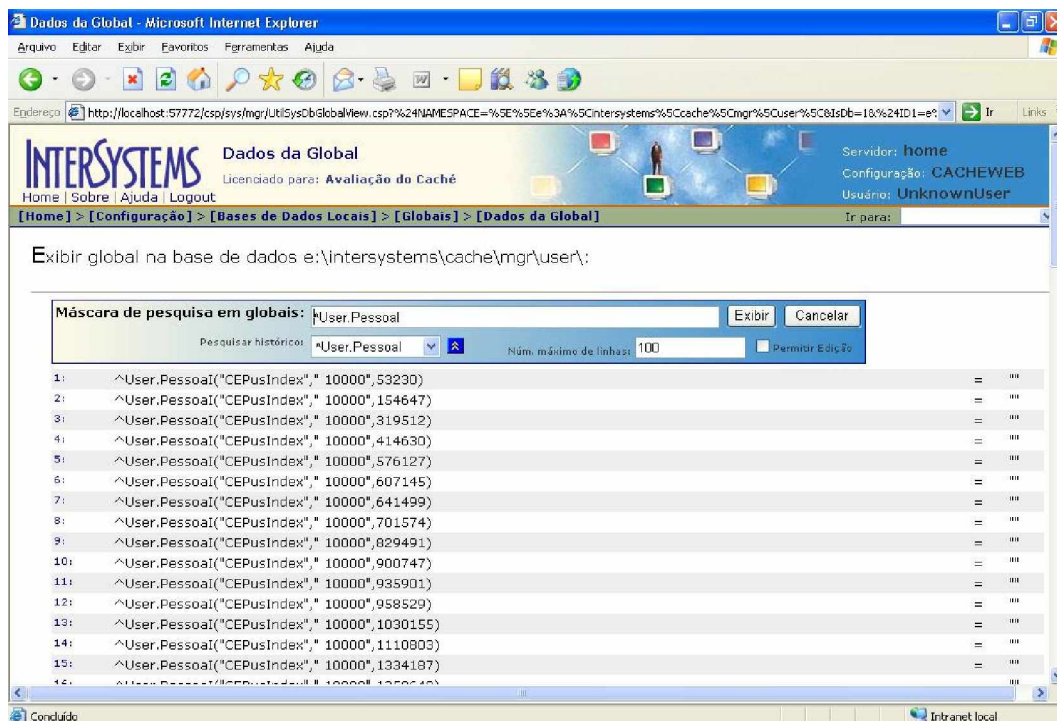
Como pode ser observado nas Figuras 14 e 15, não houve uma redução de tempo considerável na consulta, sendo que em alguns casos pode-se observar um aumento de tempo nas consultas que foram realizadas utilizando a indexação tradicional. Todos os tempos gastos em todas as consultas encontram-se em anexo a pesquisa.

A Figura 16 mostra como são armazenados na tabela PessoaBitI os índices de *bitmap*.



**Figura 16. Armazenamento dos índices de *bitmap*.**

A Figura 17 mostra como é o armazenamento dos índices tradicionais na tabela **PessoalI**.



**Figura 17. Armazenamento dos índices tradicionais.**

## 7 CONCLUSÃO

Tanto os bancos de dados, quanto os sistemas de *Data Warehouse* estão justamente no mercado corporativo para auxiliarem nas coletas de dados e na geração de informações, justamente para que cada vez mais decisões de qualquer tamanho possam ser acertadas. Juntamente com os bancos de dados e sabendo-se que grandes quantidades de informações ali são depositadas, surge a indexação, mecanismo responsável por agilizar essa busca.

Um DW, é utilizado frequentemente para consultas, e os índices como foi visto tem a função de agiliza-las, índices de *bitmap* são recomendados para esse tipo de aplicação.

Como objeto de estudo dessa pesquisa comparamos o desempenho da indexação de *bitmap* versus indexação tradicional, como podemos observar nos testes realizados esse tipo de indexação foi superior a tradicional, não queremos aqui dizer que seria necessário que fossem substituídos todos os índices por *bitmap*, porem com base nos resultados, percebemos que na medida do possível se fossem implementados trariam apenas benefícios para as aplicações.

Podemos observar que na medida do possível, a utilização de índices de *bitmap* em determinadas aplicações é vantajosa, e não podemos descartar também os índices tradicionais que sem duvida são de grande valia em diversos sistemas, e dessa maneira cabe ao desenvolvedor uma análise minuciosa da aplicação a qual ele esta trabalhando, e sugerimos que o mesmo seja aplicado no desenvolvimento e projeto de DW.

Ressaltamos que todos os testes foram realizados no servidor, devido a limitação imposta pela versão do banco ser de avaliação, não tem tempo de expiração porem não aceita conexões em rede.

## 7.1 TRABALHOS FUTUROS

Como tivemos a oportunidade de analisar as diversas características do banco, sua tecnologia recente, integração entre objetos e relacional, seu mapeamento de dados transparente, armazenamento multidimensional e o desempenho atingido nas consultas, podemos perceber que é um produto que no qual seria interessante que mais trabalhos surgissem com o intuito de avaliar outras características, e assim contribuir cada vez mais para a evolução desses tipos de sistemas, como sugestão de trabalhos futuros, temos:

- a) comparar a indexação de *bitmap* do caché com outros bancos do tipo: relacionais, pós-relacionais e orientado a objetos, que possuam esse mesmo tipo de indexação;
- b) utilização do caché em sistemas de *data minig*, implementando a indexação de *bitmap*;
- c) implementação de um *data warehouse* usando o caché como banco e uma interface específica para aplicações de *data warehouse*;
- d) analisar a indexação de *bitmap*, em bancos de dados, com inserções e remoções de registros.

## REFERÊNCIAS

ALMEIDA, Alexandre Marques de: **Proposição de Indicadores para avaliação técnica de projetos de data warehouse**: Um estudo de caso no data warehouse da plataforma Lattes. 2006. 83f. Tese de mestrado (Pós-Graduação em Engenharia de Produção) - Universidade Federal de Santa Catarina. Florianópolis.

ALMEIDA, Eduardo Cunha de: **Estudo de viabilidade de uma plataforma de baixo custo para data warehouse**. 2004. 90f. Tese de mestrado (Pós-Graduação em Informática) - Universidade Federal do Paraná. Curitiba.

BORBA, Sueli de Fátima.: **Metodologia para implantação de modelos multidimensionais em bancos de dados orientado a objetos**. 2006. 228 f. Tese de doutorado (Pós-Graduação em Engenharia de Produção) – Universidade Federal do Estado de Santa Catarina. Florianópolis.

BRAGA, Ascensão. **Trabalho realizado a partir da Tese de Mestrado em Gestão** – Universidade da Beira Interior (1996).

COSTA, André Fernandes da; ANCIÃES, Felipe Curvello.: **Aspectos de criação e carga de um ambiente de Data Warehouse** - Trabalho de Conclusão de Curso – Curso de Ciências da computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2001.

DRUCKER, Peter F.: **As novas realidades no governo, na economia e nas empresas, na sociedade e na visão do mundo**. São Paulo: Pioneira, 1993.

DRUCKER, Peter F.: **O advento da nova organização**. Rio de Janeiro: Campus, 2001.

DRUCKER, Peter F.: **Desafios Gerenciais para o Século XXI**. São Paulo: Pioneira, 2001.

GARCIA-MOLINA, Hector; ULLMAN, Jeffrey D; WIDDON, Jeniffer: **Implementação de Sistemas de bancos de dados**. Rio de Janeiro: Campus, 2001.

INTERSYSTEMS. Documentação técnica. 2007. Disponível em < <http://www.intersystems.com.br/isc/CacheDownDoc.csp>>. Acesso em: 01 mar. 2007.

MACHADO, F. N. R.; **Tecnologia e Projeto de Data Warehouse**. São Paulo:

Érica, 2004.

O' BRIEN, James A.: **Sistemas de Informação e as decisões gerenciais na era da internet**. São Paulo: Saraiva, 2003.

OKIDA, Gustavo Tadao, **Índices para consultas por conteúdo em bancos de dados heterogêneos** – Dissertação de Mestrado - Instituto de Matemática e Estatística – Curso de Ciências da Computação, Universidade de São Paulo, São Paulo, 2004.

ÔZSU, M. Tamer; VALDURIEZ, Patrick.: **Princípios de sistemas de Banco de Dados distribuídos**. Rio de Janeiro: Campus, 2001.

PINHEIRO, Edenir Monteiro. **Tópicos de Pesquisa em Data Warehouse**. 2002. 69 f. Projeto Final de Curso (Graduação em Informática – Especialização em Banco de Dados) – Universidade Federal de Goiás, Goiânia.

SILBERSCHATZ, Abraham; KORTH, Henry. F.; SUDARSHAN, S.: **Sistemas de Bancos de Dados**. São Paulo: Makron Books, 1999.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S.: **Sistema de Banco de Dados**. Rio de Janeiro: Elsevier, 2006.

SILVA, Alexsandra Cassiano Alves da: **Análise de Formas de Otimização de Consultas**. Trabalho de Conclusão de Curso – Curso de Ciência da computação, Universidade Federal de Pernambuco, Recife, Pernambuco, 2002.

SILVA, Ricardo Czelusniak da. **Benchmark em banco de dados multimídia: Análise de desempenho em recuperação de objetos multimídia**. 2006. 64 f. Tese de Mestrado (Pós-Graduação em Informática) – Universidade Federal do Paraná, Curitiba.

SINGH, Harry. **Data warehouse**. São Paulo: Makron Books, 2001.

STAIR, Ralph M.: **Princípios de sistemas de informação, uma abordagem gerencial**. São Paulo: LTC, 1998.

VIEIRA, M.; DURÃES, J.; MADEIRA, H. **Especificação e Validação de Benchmarks de Confiabilidade para Sistemas Transaccionais**. IEEE Latin America Transactions, 2005.

## ANEXOS

Em anexo as tabelas com as tomadas de tempo de cada consulta realizada na pesquisa.

| <b>Comando:</b> SELECT * FROM SQLUser.PessoaBit |                     |                  |              |                  |            |                  |            |
|-------------------------------------------------|---------------------|------------------|--------------|------------------|------------|------------------|------------|
| <b>Tabela</b>                                   | <b>Nr consultas</b> | <b>Registros</b> | <b>Seq</b>   | <b>Registros</b> | <b>Seq</b> | <b>Registros</b> | <b>Seq</b> |
| PessoaBit                                       | 1                   | 1.000            | 0,535        | 5.000            | 2,851      | 10.000           | 4,234      |
|                                                 | 2                   | 1.000            | 0,555        | 5.000            | 2,216      | 10.000           | 4,028      |
|                                                 | 3                   | 1.000            | 0,549        | 5.000            | 2,179      | 10.000           | 4,299      |
|                                                 | 4                   | 1.000            | 0,512        | 5.000            | 1,999      | 10.000           | 4,694      |
|                                                 | 5                   | 1.000            | 0,415        | 5.000            | 2,085      | 10.000           | 4,158      |
|                                                 | 6                   | 1.000            | 0,508        | 5.000            | 2,096      | 10.000           | 4,579      |
|                                                 | 7                   | 1.000            | 0,562        | 5.000            | 2,077      | 10.000           | 4,310      |
|                                                 | 8                   | 1.000            | 0,485        | 5.000            | 1,944      | 10.000           | 4,156      |
|                                                 | 9                   | 1.000            | 0,694        | 5.000            | 2,104      | 10.000           | 4,125      |
|                                                 | 10                  | 1.000            | 0,512        | 5.000            | 1,948      | 10.000           | 4,300      |
|                                                 | 11                  | 1.000            | 0,443        | 5.000            | 1,936      | 10.000           | 4,367      |
|                                                 | 12                  | 1.000            | 0,579        | 5.000            | 2,181      | 10.000           | 4,062      |
|                                                 | 13                  | 1.000            | 0,624        | 5.000            | 2,075      | 10.000           | 4,554      |
|                                                 | 14                  | 1.000            | 0,563        | 5.000            | 2,022      | 10.000           | 4,067      |
|                                                 | 15                  | 1.000            | 0,559        | 5.000            | 1,962      | 10.000           | 4,507      |
|                                                 | 16                  | 1.000            | 0,466        | 5.000            | 1,931      | 10.000           | 4,141      |
|                                                 | 17                  | 1.000            | 0,547        | 5.000            | 1,990      | 10.000           | 4,117      |
|                                                 | 18                  | 1.000            | 0,454        | 5.000            | 2,138      | 10.000           | 4,242      |
|                                                 | 19                  | 1.000            | 0,584        | 5.000            | 2,011      | 10.000           | 4,289      |
|                                                 | 20                  | 1.000            | 0,609        | 5.000            | 2,059      | 10.000           | 4,118      |
|                                                 | 21                  | 1.000            | 0,442        | 5.000            | 1,999      | 10.000           | 4,274      |
| <b>Média</b>                                    |                     |                  | <b>0,533</b> | <b>2,086</b>     |            | <b>4,268</b>     |            |
| Pessoa                                          | 1                   | 1.000            | 0,581        | 5.000            | 2,444      | 10.000           | 4,581      |
|                                                 | 2                   | 1.000            | 0,578        | 5.000            | 2,147      | 10.000           | 4,329      |
|                                                 | 3                   | 1.000            | 0,592        | 5.000            | 2,099      | 10.000           | 4,253      |
|                                                 | 4                   | 1.000            | 0,542        | 5.000            | 2,073      | 10.000           | 4,142      |
|                                                 | 5                   | 1.000            | 0,553        | 5.000            | 2,031      | 10.000           | 4,166      |
|                                                 | 6                   | 1.000            | 0,537        | 5.000            | 2,179      | 10.000           | 4,035      |
|                                                 | 7                   | 1.000            | 0,651        | 5.000            | 2,057      | 10.000           | 4,168      |
|                                                 | 8                   | 1.000            | 0,441        | 5.000            | 2,567      | 10.000           | 4,535      |
|                                                 | 9                   | 1.000            | 0,489        | 5.000            | 2,089      | 10.000           | 4,508      |
|                                                 | 10                  | 1.000            | 0,463        | 5.000            | 2,082      | 10.000           | 4,074      |
|                                                 | 11                  | 1.000            | 0,492        | 5.000            | 2,044      | 10.000           | 4,585      |
|                                                 | 12                  | 1.000            | 0,951        | 5.000            | 2,084      | 10.000           | 4,035      |
|                                                 | 13                  | 1.000            | 0,435        | 5.000            | 2,012      | 10.000           | 4,121      |
|                                                 | 14                  | 1.000            | 0,619        | 5.000            | 1,948      | 10.000           | 4,168      |
|                                                 | 15                  | 1.000            | 0,457        | 5.000            | 2,130      | 10.000           | 4,531      |
|                                                 | 16                  | 1.000            | 0,543        | 5.000            | 1,944      | 10.000           | 4,217      |
|                                                 | 17                  | 1.000            | 0,461        | 5.000            | 1,972      | 10.000           | 4,246      |
|                                                 | 18                  | 1.000            | 0,592        | 5.000            | 2,038      | 10.000           | 4,442      |
|                                                 | 19                  | 1.000            | 0,433        | 5.000            | 2,166      | 10.000           | 4,238      |
|                                                 | 20                  | 1.000            | 0,491        | 5.000            | 2,023      | 10.000           | 4,248      |
|                                                 | 21                  | 1.000            | 0,530        | 5.000            | 2,138      | 10.000           | 4,197      |
| <b>Média</b>                                    |                     |                  | <b>0,544</b> | <b>2,108</b>     |            | <b>4,277</b>     |            |

**Quadro 1. Tempos das consultas 1, 2 e 3.**

| <b>Comando:</b>                                                           |                  |                 |                                                                        |                 |
|---------------------------------------------------------------------------|------------------|-----------------|------------------------------------------------------------------------|-----------------|
| SELECT count(*) FROM SQLUser.PessoaBit<br>Where Ufus='MA'                 |                  |                 | SELECT count(*) FROM SQLUser.Pessoa<br>Where Ufus='MA'                 |                 |
| <i>Nr. Consultas</i>                                                      | <i>Registros</i> | <i>segundos</i> | <i>Registros</i>                                                       | <i>segundos</i> |
| 1                                                                         | 100.221          | 3,419           | 100.365                                                                | 3,850           |
| 2                                                                         | 100.221          | 0,005           | 100.365                                                                | 0,997           |
| 3                                                                         | 100.221          | 0,004           | 100.365                                                                | 1,024           |
| 4                                                                         | 100.221          | 0,006           | 100.365                                                                | 0,885           |
| 5                                                                         | 100.221          | 0,004           | 100.365                                                                | 0,925           |
| 6                                                                         | 100.221          | 0,006           | 100.365                                                                | 0,939           |
| 7                                                                         | 100.221          | 0,004           | 100.365                                                                | 1,029           |
| 8                                                                         | 100.221          | 0,004           | 100.365                                                                | 1,015           |
| 9                                                                         | 100.221          | 0,004           | 100.365                                                                | 0,955           |
| 10                                                                        | 100.221          | 0,010           | 100.365                                                                | 1,040           |
| 11                                                                        | 100.221          | 0,004           | 100.365                                                                | 1,024           |
| 12                                                                        | 100.221          | 0,018           | 100.365                                                                | 0,969           |
| 13                                                                        | 100.221          | 0,004           | 100.365                                                                | 0,966           |
| 14                                                                        | 100.221          | 0,004           | 100.365                                                                | 0,981           |
| 15                                                                        | 100.221          | 0,004           | 100.365                                                                | 0,962           |
| 16                                                                        | 100.221          | 0,004           | 100.365                                                                | 0,968           |
| 17                                                                        | 100.221          | 0,004           | 100.365                                                                | 1,014           |
| 18                                                                        | 100.221          | 0,004           | 100.365                                                                | 0,978           |
| 19                                                                        | 100.221          | 0,004           | 100.365                                                                | 0,978           |
| 20                                                                        | 100.221          | 0,009           | 100.365                                                                | 0,969           |
| 21                                                                        | 100.221          | 0,004           | 100.365                                                                | 0,977           |
| <b>Média</b>                                                              |                  | <b>0,168</b>    | <b>1,116</b>                                                           |                 |
| SELECT count(*) FROM SQLUser.PessoaBit<br>Where Ufus='MA' Order By 1 DESC |                  |                 | SELECT count(*) FROM SQLUser.Pessoa<br>Where Ufus='MA' Order By 1 DESC |                 |
| <i>Nr. Consultas</i>                                                      | <i>Registros</i> | <i>segundos</i> | <i>Registros</i>                                                       | <i>segundos</i> |
| 1                                                                         | 100.221          | 0,363           | 100.365                                                                | 0,977           |
| 2                                                                         | 100.221          | 0,016           | 100.365                                                                | 1,097           |
| 3                                                                         | 100.221          | 0,004           | 100.365                                                                | 1,004           |
| 4                                                                         | 100.221          | 0,005           | 100.365                                                                | 0,995           |
| 5                                                                         | 100.221          | 0,008           | 100.365                                                                | 0,985           |
| 6                                                                         | 100.221          | 0,023           | 100.365                                                                | 1,040           |
| 7                                                                         | 100.221          | 0,004           | 100.365                                                                | 1,038           |
| 8                                                                         | 100.221          | 0,004           | 100.365                                                                | 0,992           |
| 9                                                                         | 100.221          | 0,004           | 100.365                                                                | 0,989           |
| 10                                                                        | 100.221          | 0,004           | 100.365                                                                | 0,978           |
| 11                                                                        | 100.221          | 0,004           | 100.365                                                                | 0,961           |
| 12                                                                        | 100.221          | 0,004           | 100.365                                                                | 0,963           |
| 13                                                                        | 100.221          | 0,006           | 100.365                                                                | 0,974           |
| 14                                                                        | 100.221          | 0,004           | 100.365                                                                | 1,044           |
| 15                                                                        | 100.221          | 0,004           | 100.365                                                                | 1,003           |
| 16                                                                        | 100.221          | 0,004           | 100.365                                                                | 0,986           |
| 17                                                                        | 100.221          | 0,004           | 100.365                                                                | 0,975           |
| 18                                                                        | 100.221          | 0,004           | 100.365                                                                | 0,953           |
| 19                                                                        | 100.221          | 0,005           | 100.365                                                                | 1,019           |
| 20                                                                        | 100.221          | 0,007           | 100.365                                                                | 1,000           |
| 21                                                                        | 100.221          | 0,004           | 100.365                                                                | 0,964           |
| <b>Média</b>                                                              |                  | <b>0,023</b>    | <b>0,997</b>                                                           |                 |

Quadro 2. Tempos das consultas 4 e 5.

| <b>Comando:</b><br>SELECT count(*) FROM SQLUser.PessoaBit Where<br>Sexo='F' |                              |                 |  |  | SELECT count(*) FROM SQLUser.Pessoa<br>Where Sexo='F' |                 |
|-----------------------------------------------------------------------------|------------------------------|-----------------|--|--|-------------------------------------------------------|-----------------|
| <i>Nr. consultas</i>                                                        | <i>Registros Encontrados</i> | <i>Segundos</i> |  |  | <i>Registros Encontrados</i>                          | <i>Segundos</i> |
| 1                                                                           | 2.500.376                    | 3,473           |  |  | 2.501.460                                             | 19,672          |
| 2                                                                           | 2.500.376                    | 0,008           |  |  | 2.501.460                                             | 19,854          |
| 3                                                                           | 2.500.376                    | 0,020           |  |  | 2.501.460                                             | 19,830          |
| 4                                                                           | 2.500.376                    | 0,010           |  |  | 2.501.460                                             | 19,802          |
| 5                                                                           | 2.500.376                    | 0,012           |  |  | 2.501.460                                             | 19,890          |
| 6                                                                           | 2.500.376                    | 0,007           |  |  | 2.501.460                                             | 19,796          |
| 7                                                                           | 2.500.376                    | 0,008           |  |  | 2.501.460                                             | 19,841          |
| 8                                                                           | 2.500.376                    | 0,007           |  |  | 2.501.460                                             | 19,830          |
| 9                                                                           | 2.500.376                    | 0,008           |  |  | 2.501.460                                             | 19,781          |
| 10                                                                          | 2.500.376                    | 0,007           |  |  | 2.501.460                                             | 19,853          |
| 11                                                                          | 2.500.376                    | 0,007           |  |  | 2.501.460                                             | 20,054          |
| 12                                                                          | 2.500.376                    | 0,007           |  |  | 2.501.460                                             | 19,865          |
| 13                                                                          | 2.500.376                    | 0,015           |  |  | 2.501.460                                             | 19,832          |
| 14                                                                          | 2.500.376                    | 0,007           |  |  | 2.501.460                                             | 19,803          |
| 15                                                                          | 2.500.376                    | 0,008           |  |  | 2.501.460                                             | 19,894          |
| 16                                                                          | 2.500.376                    | 0,022           |  |  | 2.501.460                                             | 19,933          |
| 17                                                                          | 2.500.376                    | 0,011           |  |  | 2.501.460                                             | 19,776          |
| 18                                                                          | 2.500.376                    | 0,008           |  |  | 2.501.460                                             | 19,854          |
| 19                                                                          | 2.500.376                    | 0,013           |  |  | 2.501.460                                             | 19,806          |
| 20                                                                          | 2.500.376                    | 0,008           |  |  | 2.501.460                                             | 19,840          |
| 21                                                                          | 2.500.376                    | 0,007           |  |  | 2.501.460                                             | 19,825          |
|                                                                             | <b>Média</b>                 | <b>0,175</b>    |  |  |                                                       | <b>19,840</b>   |
| SELECT count(*) FROM SQLUser.PessoaBit Where<br>Sexo='M'                    |                              |                 |  |  | SELECT count(*) FROM SQLUser.Pessoa<br>Where Sexo='M' |                 |
| <i>Nr consultas</i>                                                         | <i>Registros Encontrados</i> | <i>Segundos</i> |  |  | <i>Registros Encontrados</i>                          | <i>Segundos</i> |
| 1                                                                           | 2.499.624                    | 0,698           |  |  | 2.498.540                                             | 19,753          |
| 2                                                                           | 2.499.624                    | 0,008           |  |  | 2.498.540                                             | 19,797          |
| 3                                                                           | 2.499.624                    | 0,007           |  |  | 2.498.540                                             | 20,059          |
| 4                                                                           | 2.499.624                    | 0,008           |  |  | 2.498.540                                             | 19,824          |
| 5                                                                           | 2.499.624                    | 0,008           |  |  | 2.498.540                                             | 19,965          |
| 6                                                                           | 2.499.624                    | 0,007           |  |  | 2.498.540                                             | 19,739          |
| 7                                                                           | 2.499.624                    | 0,007           |  |  | 2.498.540                                             | 19,702          |
| 8                                                                           | 2.499.624                    | 0,007           |  |  | 2.498.540                                             | 19,770          |
| 9                                                                           | 2.499.624                    | 0,027           |  |  | 2.498.540                                             | 20,497          |
| 10                                                                          | 2.499.624                    | 0,110           |  |  | 2.498.540                                             | 19,781          |
| 11                                                                          | 2.499.624                    | 0,007           |  |  | 2.498.540                                             | 19,794          |
| 12                                                                          | 2.499.624                    | 0,007           |  |  | 2.498.540                                             | 19,759          |
| 13                                                                          | 2.499.624                    | 0,008           |  |  | 2.498.540                                             | 19,771          |
| 14                                                                          | 2.499.624                    | 0,008           |  |  | 2.498.540                                             | 19,729          |
| 15                                                                          | 2.499.624                    | 0,034           |  |  | 2.498.540                                             | 19,739          |
| 16                                                                          | 2.499.624                    | 0,008           |  |  | 2.498.540                                             | 19,792          |
| 17                                                                          | 2.499.624                    | 0,011           |  |  | 2.498.540                                             | 19,710          |
| 18                                                                          | 2.499.624                    | 0,011           |  |  | 2.498.540                                             | 19,786          |
| 19                                                                          | 2.499.624                    | 0,008           |  |  | 2.498.540                                             | 19,787          |
| 20                                                                          | 2.499.624                    | 0,008           |  |  | 2.498.540                                             | 21,944          |
| 21                                                                          | 2.499.624                    | 0,007           |  |  | 2.498.540                                             | 19,764          |
|                                                                             | <b>Média</b>                 | <b>0,048</b>    |  |  |                                                       | <b>19,927</b>   |

Quadro 3. Tempos das consultas 6 e 7.

| <b>Comando:</b><br>SELECT count(*) FROM SQLUser.PessoaBit<br>Where Idade < 40 |                              |                 | SELECT count(*) FROM SQLUser.Pessoa<br>Where Idade < 40              |                 |
|-------------------------------------------------------------------------------|------------------------------|-----------------|----------------------------------------------------------------------|-----------------|
| <b>Nr consultas</b>                                                           | <b>Registros Encontrados</b> | <b>Segundos</b> | <b>Registros Encontrados</b>                                         | <b>Segundos</b> |
| 1                                                                             | 1.893.274                    | 5,387           | 1.893.883                                                            | 14,164          |
| 2                                                                             | 1.893.274                    | 0,069           | 1.893.883                                                            | 14,271          |
| 3                                                                             | 1.893.274                    | 0,080           | 1.893.883                                                            | 14,109          |
| 4                                                                             | 1.893.274                    | 0,063           | 1.893.883                                                            | 14,028          |
| 5                                                                             | 1.893.274                    | 0,069           | 1.893.883                                                            | 14,065          |
| 6                                                                             | 1.893.274                    | 0,156           | 1.893.883                                                            | 14,120          |
| 7                                                                             | 1.893.274                    | 0,094           | 1.893.883                                                            | 14,094          |
| 8                                                                             | 1.893.274                    | 0,087           | 1.893.883                                                            | 14,094          |
| 9                                                                             | 1.893.274                    | 0,076           | 1.893.883                                                            | 14,135          |
| 10                                                                            | 1.893.274                    | 0,087           | 1.893.883                                                            | 14,224          |
| 11                                                                            | 1.893.274                    | 0,075           | 1.893.883                                                            | 14,098          |
| 12                                                                            | 1.893.274                    | 0,093           | 1.893.883                                                            | 14,112          |
| 13                                                                            | 1.893.274                    | 0,100           | 1.893.883                                                            | 14,108          |
| 14                                                                            | 1.893.274                    | 0,127           | 1.893.883                                                            | 14,064          |
| 15                                                                            | 1.893.274                    | 0,096           | 1.893.883                                                            | 14,086          |
| 16                                                                            | 1.893.274                    | 0,113           | 1.893.883                                                            | 14,034          |
| 17                                                                            | 1.893.274                    | 0,082           | 1.893.883                                                            | 14,115          |
| 18                                                                            | 1.893.274                    | 0,077           | 1.893.883                                                            | 14,084          |
| 19                                                                            | 1.893.274                    | 0,093           | 1.893.883                                                            | 14,091          |
| 20                                                                            | 1.893.274                    | 0,102           | 1.893.883                                                            | 14,076          |
| 21                                                                            | 1.893.274                    | 0,082           | 1.893.883                                                            | 14,055          |
|                                                                               | <b>Média</b>                 | <b>0,343</b>    |                                                                      | <b>14,106</b>   |
| SELECT count(*) FROM SQLUser.PessoaBit<br>Where Idade between 10 and 40       |                              |                 | SELECT count(*) FROM SQLUser.Pessoa<br>Where Idade between 10 and 40 |                 |
| <b>Nr consultas</b>                                                           | <b>Registros Encontrados</b> | <b>Segundos</b> | <b>Registros Encontrados</b>                                         | <b>Segundos</b> |
| 1                                                                             | 1.968.955                    | 5,547           | 1.969.284                                                            | 14,548          |
| 2                                                                             | 1.968.955                    | 0,082           | 1.969.284                                                            | 14,675          |
| 3                                                                             | 1.968.955                    | 0,101           | 1.969.284                                                            | 14,822          |
| 4                                                                             | 1.968.955                    | 0,087           | 1.969.284                                                            | 14,668          |
| 5                                                                             | 1.968.955                    | 0,070           | 1.969.284                                                            | 14,678          |
| 6                                                                             | 1.968.955                    | 0,079           | 1.969.284                                                            | 14,668          |
| 7                                                                             | 1.968.955                    | 0,101           | 1.969.284                                                            | 14,715          |
| 8                                                                             | 1.968.955                    | 0,076           | 1.969.284                                                            | 14,692          |
| 9                                                                             | 1.968.955                    | 0,076           | 1.969.284                                                            | 14,661          |
| 10                                                                            | 1.968.955                    | 0,063           | 1.969.284                                                            | 14,671          |
| 11                                                                            | 1.968.955                    | 0,081           | 1.969.284                                                            | 14,601          |
| 12                                                                            | 1.968.955                    | 0,075           | 1.969.284                                                            | 14,703          |
| 13                                                                            | 1.968.955                    | 0,090           | 1.969.284                                                            | 14,655          |
| 14                                                                            | 1.968.955                    | 0,097           | 1.969.284                                                            | 14,589          |
| 15                                                                            | 1.968.955                    | 0,081           | 1.969.284                                                            | 14,651          |
| 16                                                                            | 1.968.955                    | 0,090           | 1.969.284                                                            | 14,680          |
| 17                                                                            | 1.968.955                    | 0,073           | 1.969.284                                                            | 14,641          |
| 18                                                                            | 1.968.955                    | 0,070           | 1.969.284                                                            | 14,656          |
| 19                                                                            | 1.968.955                    | 0,088           | 1.969.284                                                            | 16,423          |
| 20                                                                            | 1.968.955                    | 0,097           | 1.969.284                                                            | 14,664          |
| 21                                                                            | 1.968.955                    | 0,098           | 1.969.284                                                            | 14,638          |
|                                                                               | <b>Média</b>                 | <b>0,344</b>    |                                                                      | <b>14,748</b>   |

Quadro 4. Tempo das consultas 8 e 9.

| <b>Comando:</b><br>SELECT Nome<br>FROM SQLUser.PessoaBit<br>Where Ufus='MA' and Idade = 50                      |                              |                 | SELECT Nome<br>FROM SQLUser.Pessoa<br>Where Ufus='MA' and Idade = 50                                         |                 |
|-----------------------------------------------------------------------------------------------------------------|------------------------------|-----------------|--------------------------------------------------------------------------------------------------------------|-----------------|
| <i>Nr<br/>consultas</i>                                                                                         | <i>Registros Encontrados</i> | <i>Segundos</i> | <i>Registros Encontrados</i>                                                                                 | <i>Segundos</i> |
| 1                                                                                                               | 1524                         | 2,922           | 1542                                                                                                         | 9,066           |
| 2                                                                                                               | 1524                         | 0,346           | 1542                                                                                                         | 1,947           |
| 3                                                                                                               | 1524                         | 0,402           | 1542                                                                                                         | 1,893           |
| 4                                                                                                               | 1524                         | 0,363           | 1542                                                                                                         | 1,966           |
| 5                                                                                                               | 1524                         | 0,390           | 1542                                                                                                         | 1,963           |
| 6                                                                                                               | 1524                         | 0,343           | 1542                                                                                                         | 1,897           |
| 7                                                                                                               | 1524                         | 0,394           | 1542                                                                                                         | 1,908           |
| 8                                                                                                               | 1524                         | 0,314           | 1542                                                                                                         | 1,941           |
| 9                                                                                                               | 1524                         | 0,375           | 1542                                                                                                         | 1,928           |
| 10                                                                                                              | 1524                         | 0,483           | 1542                                                                                                         | 1,919           |
| 11                                                                                                              | 1524                         | 0,323           | 1542                                                                                                         | 1,912           |
| 12                                                                                                              | 1524                         | 0,383           | 1542                                                                                                         | 1,894           |
| 13                                                                                                              | 1524                         | 0,474           | 1542                                                                                                         | 1,909           |
| 14                                                                                                              | 1524                         | 0,343           | 1542                                                                                                         | 1,919           |
| 15                                                                                                              | 1524                         | 0,413           | 1542                                                                                                         | 1,942           |
| 16                                                                                                              | 1524                         | 0,347           | 1542                                                                                                         | 1,901           |
| 17                                                                                                              | 1524                         | 0,351           | 1542                                                                                                         | 1,858           |
| 18                                                                                                              | 1524                         | 0,378           | 1542                                                                                                         | 1,909           |
| 19                                                                                                              | 1524                         | 0,367           | 1542                                                                                                         | 1,942           |
| 20                                                                                                              | 1524                         | 0,477           | 1542                                                                                                         | 1,962           |
| 21                                                                                                              | 1524                         | 0,621           | 1542                                                                                                         | 1,993           |
|                                                                                                                 | Média                        | 0,515           |                                                                                                              | 2,265           |
| SELECT Nome<br>FROM SQLUser.PessoaBit<br>Where (Idade between 5 and 6<br>or Ufus='MA') and CorCabelo='Vermelho' |                              |                 | SELECT Nome<br>FROM SQLUser.Pessoa<br>Where (Idade between 5 and 6<br>or Ufus='MA') and CorCabelo='Vermelho' |                 |
| <i>Nr<br/>consultas</i>                                                                                         | <i>Registros Encontrados</i> | <i>Segundos</i> | <i>Registros Encontrados</i>                                                                                 | <i>Segundos</i> |
| 1                                                                                                               | 19865                        | 21,225          | 20351                                                                                                        | 31,567          |
| 2                                                                                                               | 19865                        | 23,242          | 20351                                                                                                        | 31,732          |
| 3                                                                                                               | 19865                        | 22,850          | 20351                                                                                                        | 32,624          |
| 4                                                                                                               | 19865                        | 22,649          | 20351                                                                                                        | 32,439          |
| 5                                                                                                               | 19865                        | 22,909          | 20351                                                                                                        | 32,612          |
| 6                                                                                                               | 19865                        | 22,243          | 20351                                                                                                        | 33,017          |
| 7                                                                                                               | 19865                        | 22,063          | 20351                                                                                                        | 32,486          |
| 8                                                                                                               | 19865                        | 22,548          | 20351                                                                                                        | 32,809          |
| 9                                                                                                               | 19865                        | 21,990          | 20351                                                                                                        | 32,919          |
| 10                                                                                                              | 19865                        | 22,711          | 20351                                                                                                        | 32,760          |
| 11                                                                                                              | 19865                        | 22,606          | 20351                                                                                                        | 32,443          |
| 12                                                                                                              | 19865                        | 22,622          | 20351                                                                                                        | 32,533          |
| 13                                                                                                              | 19865                        | 22,557          | 20351                                                                                                        | 32,549          |
| 14                                                                                                              | 19865                        | 22,928          | 20351                                                                                                        | 32,541          |
| 15                                                                                                              | 19865                        | 22,881          | 20351                                                                                                        | 32,647          |
| 16                                                                                                              | 19865                        | 22,627          | 20351                                                                                                        | 32,379          |
| 17                                                                                                              | 19865                        | 22,981          | 20351                                                                                                        | 32,466          |
| 18                                                                                                              | 19865                        | 22,899          | 20351                                                                                                        | 32,430          |
| 19                                                                                                              | 19865                        | 22,457          | 20351                                                                                                        | 32,312          |

|                                                                                                                     |                              |                 |                                                                                                                  |                 |
|---------------------------------------------------------------------------------------------------------------------|------------------------------|-----------------|------------------------------------------------------------------------------------------------------------------|-----------------|
| 20                                                                                                                  | 19865                        | 22,622          | 20351                                                                                                            | 32,140          |
| 21                                                                                                                  | 19865                        | 22,399          | 20351                                                                                                            | 32,191          |
|                                                                                                                     | <b>Média</b>                 | <b>22,572</b>   |                                                                                                                  | <b>32,457</b>   |
| SELECT Count(*)<br>FROM SQLUser.PessoaBit<br>Where (Idade between 5 and 6 or<br>Ufus='MA') and CorCabelo='Vermelho' |                              |                 | SELECT Count(*)<br>FROM SQLUser.Pessoa<br>Where (Idade between 5 and 6<br>or Ufus='MA') and CorCabelo='Vermelho' |                 |
| <b>Nr<br/>consultas</b>                                                                                             | <b>Registros Encontrados</b> | <b>Segundos</b> | <b>Registros Encontrados</b>                                                                                     | <b>Segundos</b> |
| 1                                                                                                                   | 19865                        | 15,518          | 20351                                                                                                            | 22,635          |
| 2                                                                                                                   | 19865                        | 16,575          | 20351                                                                                                            | 22,593          |
| 3                                                                                                                   | 19865                        | 13,742          | 20351                                                                                                            | 21,322          |
| 4                                                                                                                   | 19865                        | 16,582          | 20351                                                                                                            | 22,129          |
| 5                                                                                                                   | 19865                        | 17,041          | 20351                                                                                                            | 21,752          |
| 6                                                                                                                   | 19865                        | 16,168          | 20351                                                                                                            | 21,824          |
| 7                                                                                                                   | 19865                        | 13,943          | 20351                                                                                                            | 21,483          |
| 8                                                                                                                   | 19865                        | 14,022          | 20351                                                                                                            | 21,583          |
| 9                                                                                                                   | 19865                        | 14,512          | 20351                                                                                                            | 22,306          |
| 10                                                                                                                  | 19865                        | 13,819          | 20351                                                                                                            | 26,114          |
| 11                                                                                                                  | 19865                        | 13,871          | 20351                                                                                                            | 21,531          |
| 12                                                                                                                  | 19865                        | 14,261          | 20351                                                                                                            | 22,190          |
| 13                                                                                                                  | 19865                        | 13,880          | 20351                                                                                                            | 21,333          |
| 14                                                                                                                  | 19865                        | 14,022          | 20351                                                                                                            | 22,050          |
| 15                                                                                                                  | 19865                        | 14,005          | 20351                                                                                                            | 21,332          |
| 16                                                                                                                  | 19865                        | 13,902          | 20351                                                                                                            | 21,636          |
| 17                                                                                                                  | 19865                        | 14,528          | 20351                                                                                                            | 21,514          |
| 18                                                                                                                  | 19865                        | 13,788          | 20351                                                                                                            | 23,836          |
| 19                                                                                                                  | 19865                        | 13,881          | 20351                                                                                                            | 22,135          |
| 20                                                                                                                  | 19865                        | 14,013          | 20351                                                                                                            | 22,494          |
| 21                                                                                                                  | 19865                        | 13,746          | 20351                                                                                                            | 22,607          |
|                                                                                                                     | <b>Média</b>                 | <b>14,563</b>   |                                                                                                                  | <b>22,209</b>   |

Quadro 5. Tempos das consultas 10, 11 e 12.