

# Análise de desempenho de Bancos de Dados

Erick Rodrigues Ferreira<sup>1</sup>, Sergio M. Trad Júnior<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação – Universidade Presidente Antônio Carlos (UNIPAC)  
Rua Palma Bageto Viol S/N – Barbacena – MG – Brasil

erick.r.ferreira@hotmail.com, smtrad@gmail.com

**Abstract.** *There is concern about which system database is better than the other in face of practical situations. This causes the interest in developing analyzes between different systems on the market. In the scope of performance of database systems, this paper presents an analysis between databases Firebird 2.5, PostgreSQL 9.2.1, MySQL 5.5, SQL Server 2008, to assist users of databases about the choice of the most showing the way they behave in a series of tests.*

**Resumo.** *Existe uma preocupação com qual sistema de banco de dados é melhor do que o outro diante de situações práticas. Isto provoca o interesse em desenvolver trabalhos de análise entre os diferentes sistemas existentes no mercado. No escopo de desempenho de sistemas de bancos de dados, este trabalho apresenta uma análise entre os bancos Firebird 2.5, PostgreSQL 9.2.1, MySQL 5.5, SQL Server 2008, para auxiliar os usuários de bancos de dados quanto à escolha do mais adequado, apresentando como os mesmos se comportam perante uma bateria de testes.*

## 1. Introdução

A demanda por Sistemas de Gerenciamento de Bancos de Dados cresce continuamente a cada dia, e juntamente com essa demanda, cresce também o volume de dados que estes sistemas devem gerenciar assim como a complexidade de suas aplicações. Segundo [CARNEIRO 2011], efetuar operações sobre estas grandes coleções de dados é uma questão pontual, pois a avaliação de desempenho de um SGBD (Sistema Gerenciador de Banco de Dados) é medida, usando como base sua eficiência diante de consultas e alterações.

Atualmente, como cita [GESTAO, 2008], encontram-se no mercado diversos sistemas gerenciadores de banco de dados, os quais, na maioria dos casos, são instalados, configurados e utilizados com todos os seus parâmetros em valores padrões, sem levar em consideração o tipo de aplicação para o qual serão utilizados, o *hardware*, e até mesmo o sistema operacional. Com isso, o melhor desempenho do sistema de banco de dados, pode não ser obtido, visto que diversos parâmetros podem ser considerados e ajustados.

Ainda segundo [GESTÃO, 2008], o primeiro caminho para conseguir um desempenho adequado de um sistema de banco de dados e tomar decisões corentes durante seu projeto, seguindo assim um método de trabalho que irá lhe fornecer um melhor resultado.

De acordo com a definição do dicionário Websters II, um benchmark é um padrão para medida ou avaliação. Em ambientes computacionais, um *benchmark* é tipicamente um *software* que realiza um conjunto restrito e pré-definido de operações, uma carga de

trabalho e retorna um resultado em algum formato (métrica), que descreve o comportamento do sistema. Segundo [Gray 1993], executar o *benchmark*, torna possível realizar comparações de dois ou mais *softwares* em um mesmo computador. A escolha de um Sistema Gerenciador de Banco de Dados (SGBD), dentre a grande diversidade disponível no mercado é uma tarefa delicada, devido à importância e responsabilidade que essa ferramenta representa por gerenciar uma das maiores riquezas de uma organização, que são as informações. Nesta tarefa, vários fatores devem ser considerados, tais como confiabilidade, integridade e segurança dos dados, suporte à linguagem de programação, interoperabilidade, conjunto de ferramentas do produto, dentro outras. Contudo, com o crescimento constante do volume de dados que os SGBD's gerenciam, e que muitas vezes estão armazenados remotamente, um fator que vem se destacando como ponto decisivo para escolha de um sistema é o desempenho.

### 1.1. Objetivos

O objetivo geral deste trabalho é pesquisar alguns dos SGBD's mais utilizados atualmente, e medir o desempenho mesclando sistema abertos e proprietários. Ilustrar uma comparação entre eles, para que, através dos dados obtidos, contribua-se para o processo de determinação do sistema gerenciador de banco de dados a ser utilizado.

Para alcançar esse objetivo, será utilizado como ferramenta de comparação os *Benchmark's TPC-C (Transaction Process Council - Consórcio de Transações por Processo em Consultas)* e *AS<sup>3</sup>AP. de Banco de Dados (SGBDs)*. Nessas ferramentas, serão definidas cargas de processamento de transações que simula as atividades encontradas em um ambiente de aplicações complexas e verificada a amplitude na medição do desempenho.

Especificamente este trabalho realizará uma análise de desempenho de quatro sistemas Gerenciadores de Banco de Dados, sendo eles: Firebird 2.5, PostgreSQL 9.2.1, MySQL 5.5, SQL Server 2008. Para os testes e configuração dos bancos serão usadas, respectivamente, as seguintes ferramentas de administração: Ibxpert, Pgadmin II, MySQL Front e SQL Server Management Studio.

Essas ferramentas, segundo [LEITE,2007], fornecem um ambiente de desenvolvimento integrado para acessar, configurar, gerenciar e desenvolver todos os componentes dos SGBD's. Combina um amplo grupo de ferramentas gráficas com editores de scripts sofisticados que fornecem aos desenvolvedores e administradores acesso ao Banco. O método que será aplicado consiste em efetuar uma comparação entre eles, para que, através dos dados obtidos, contribuía-se para o processo de determinação do sistema de banco de dados a ser utilizado em cada tipo de aplicação. Desta forma o artigo visa:

- Realizar inserções de tipos e volumes diferentes e computar o tempo de resposta destas;
- Analisar o crescimento do volume de dados, verificando o aumento do espaço físico da base mediante as inserções;
- Apresentar gráficos comparativos mediante resultados obtidos;

### 1.2. Motivação

Segundo [Guimaraes, 2002], não existe um meio universal com o qual possamos avaliar o desempenho dos diversos sistemas computacionais, é necessário definir técnicas

e métricas para cada caso. Assim como a maioria dos sistemas de *software*, também os bancos de dados são complexos, pois há uma boa quantidade de variações pelas implementações dos diferentes fornecedores. Essas variações podem ser de performances da instância de dados, recuperação, disponibilidade, configuração e especificações de cada tecnologia. Como resultado, há significativas mudanças de desempenho em diferentes tarefas. Enquanto um sistema pode ser o mais eficiente para uma determinada tarefa, outro pode ser o mais eficiente para uma tarefa diferente [Silberschatz., Korth. e Sudarshan., 2006].

Comparação de SGBD's usando sistemas abertos ou proprietários, que mostram de uma forma objetiva o desempenho dos diversos sistemas de bancos de dados, podem ajudar a quem está ingressando nesta área. Desenvolvedores de aplicações se deparam com dificuldades grandes quando procuram comparações entre sistemas gerenciadores de bancos de dados, no intuito de determinarem qual deles é o mais adequado.

## 2. Sistemas de Bancos de Dados

Segundo [Silberschatz., Korth. e Sudarshan., 2006], sistema de bancos de dados são projetados para gerir grandes volumes de informações. O gerenciamento de informações implica a definição dos mecanismos para manipulação dessas informações. Ainda segundo os autores, um sistema de banco de dados deve garantir a segurança das informações armazenadas contra eventuais problemas como o sistema, além de impedir tentativas de acesso não autorizadas.

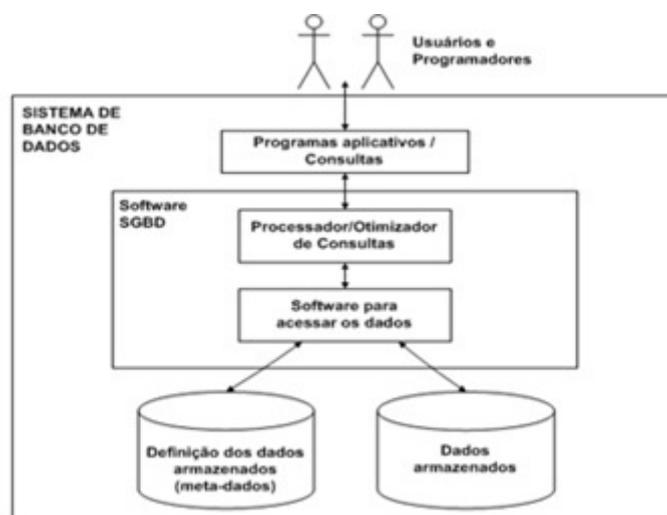


Figura 1. Sistema de banco de dados.[Silberschatz., Korth. e Sudarshan., 2006]

O banco de dados é um recurso que veio facilitar a busca de informações, eliminando os arquivos de pap'éis, integrando os dados de aplicações e fornecendo segurança. Com o objetivo de simplificar a criação e manutenção de um Banco de Dados surgiu os Sistemas Gerenciadores de Banco de Dados (SGBDs). Segundo [Date., 2000], os Sistemas de Banco de Dados tem a finalidade de gerenciar as informações de um banco de dados, que é um repositório para uma coleção de arquivos de dados computadorizados. Sendo assim, é uma camada que fica entre o banco de dados e os usuários, ou outros *softwares* que interagem com o SGBD. Um SGBD fornece aos usuários operações para

inserir, alterar, excluir, obter e atualizar dados em um sistema, assegurando a integridade dos dados. Sua utilização traz vantagens como: eficiência no acesso aos dados, redução do tempo de desenvolvimento de uma aplicação, acesso concorrente e rápido retorno, além da integridade e segurança dos dados. A (figura 1) apresenta os componentes de um sistema de banco de dados.

## **2.1. POSTGRESQL**

O PostgreSQL é definido como um Sistema Gerenciador de Banco de Dados objeto relacional, por apresentar características de um SGBD relacional e ainda, funcionalidades de orientação a objetos (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2005). Com 21 anos de desenvolvimento e uma combinação de esforços de uma grande comunidade de programadores espalhada pelo mundo, o PostgreSQL é um sistema de código aberto que está concorrendo com produtos comerciais como SQL Server e Oracle. Por se tratar de um código aberto, com licença BSD (*Berkeley Software Distribution*), o sistema possibilita o acesso, alterações e adaptações no código por seus próprios usuários, exigindo somente que o código licenciado mantenha suas informações de direitos autorais e licença. O PostgreSQL pode ser adquirido através de download do site: <http://www.postgresql.org/>.

### **2.1.1. Características**

O PostgreSQL segue a padronização SQL (*Structured Query Language*- Linguagem Estruturada de Consulta), uma linguagem de interface para SGBD. Possui uma ferramenta texto chamada psql, para receber os comandos SQL e uma interface gráfica chamada PGAdmin para tornar mais fácil e intuitivo o uso do SGBD. O sistema possibilita também a manipulação dos dados armazenados através de camadas ODBC e JDBC ou com acesso nativo na linguagem C. Dada estas-características, é possível desenvolver aplicações complexas para a manipulação de dados armazenados no PostgreSQL, usando qualquer linguagem de programação, como C/C++, Delphi, Java, Perl, PHP, Python, Visual Basic, e outras [LEITE, 2007].

## **2.2. MySQL**

MySQL é um banco de dados relacional, desenvolvido para plataformas Linux like, OS/2, Windows. Sendo um *softwares* de livre distribuição para plataformas não-Windows que o utilizam em um servidor Web. MySQL é um servidor multiusuário, multitarefa, compatível com o padrão SQL (Structured Query language Linguagem de Consulta estruturada), linguagem essa amplamente utilizada para manipulação de dados em RDBMS (Banco de dados Relacionais), sendo considerada um ferramenta de manipulação de base de dados de tamanho moderado. As principais características que destacam MySQL são: seu suporte as instruções SQL; sua natureza de distribuição gratuita; facilidade de integração com servidor Web e linguagens de programação de desenvolvimento de sites dinâmicos, especialmente a linguagem PHP.

### **2.2.1. Características**

MySQL é um banco de dados multiprocessado, significando que pode utilizar vários processadores ao mesmo tempo. Capacidade para manipular bancos com até 50 milhões

de registros. MySQL foi escrito em C e C++. Permite conexões via TCP/IP. Permite acesso via ODBC. Possui instruções para extração de informações relativas a tabelas, bancos, índices. Sua instalação através de RPM (*RedHat Package Manager* - Gerenciador de pacotes RedHat), é um processo simplificado, sendo criada toda a estrutura interna de arquivos no sistema operacional, bem como execução de scripts de inicialização e ajustes em arquivos de inicialização no servidor Web e módulo PHP.

### **2.3. SQL Server**

O SQL Server é um SGBD - sistema gerenciador de Banco de dados Relacional criado pela Microsoft em parceria com a Sybase em 1988 e inserido como produto complementar do Windows NT. Ao final da parceria, em 1994, a Microsoft continuou aperfeiçoando o produto. Com a nova versão, o Microsoft SQL Server 2008 fornece uma plataforma de dados confiável, produtiva e inteligente que permite que você execute suas aplicações de missão crítica mais exigentes, reduza o tempo e o custo com o desenvolvimento e o gerenciamento de aplicações e entregue percepção que se traduz em ações estratégica sem toda sua organização. O SQL é um SGBD robusto e usado por sistemas corporativos dos mais diversos portes.

#### **2.3.1. Características**

A plataforma proporciona segurança, confiabilidade, facilidade de gerenciamento e escalonável de data warehouse. Além disso, essa plataforma é adequada para aplicações de pequeno e grande porte. Confira outras facilidades do sistema:

- criação e compartilhamento de relatórios através do SQL Server Reporting Services;
- possibilidade de criptografia de informações valiosas em um banco de dados, arquivos de dados ou de log, sem necessidade de modificar as aplicações;
- visualização, integração e análise de qualquer tipo de arquivo, do relacional a documentos, informações geográficas e XML;

### **2.4. FIREBIRD**

O Firebird é um sistema de manutenção de bases de dados, sustentado pelo trabalho voluntário de um grupo de desenvolvedores e pela Fundação FireBird. O sistema Firebird está disponível no site <http://firebird.sourceforge.net/> para download. E por ser um projeto de código aberto, os usuários podem contribuir implementando novos recursos e corrigindo erros.

#### **2.4.1. Características**

Utiliza o padrão ANSI SQL-92. Está disponível para os principais sistemas operacionais (Windows, Linux, Solaris, MacOS), suporte a replicação, integridade referencial entre tabelas, backups incrementais, transações compatível com ACID (Atomicidade, Consistência, Isolamento, Durabilidade), UDF (User Defined Functions) e SubSelects. Além disso, o Firebird apresenta pouca ocupação de espaço em disco (10,33 Mb) e utiliza pouca memória em situações normais de uso. As opções mais conhecidas de ambiente gráfico do Firebird são o IBExpert, IBOConsole e Firebird Workbench.

### 3. BENCHMARKS

Um *benchmark* consiste em um padrão de testes para medir ou avaliar algo, sendo utilizado também para analisar o desempenho de um sistema. Pode ser implementado como uma instrução, um programa ou uma específicas de pedidos de interações com um componente de *software* [LORENA, 2007].

Segundo [Svobodova 1976], a finalidade da avaliação de desempenho de um Sistema é verificar a efetividade com que os recursos de *hardware* são utilizados para atender aos objetivos do *softwares*. Esta avaliação pode ser classificada como comparativa, quando o desempenho do sistema é avaliado mediante o desempenho de outro sistema, ou analítica, quando o sistema é avaliado com vários parâmetros de carga e de configuração do próprio sistema.

Conforme descrevem [PIRES., NASCIMENTO. e SALGADO, 2006], os programas de benchmark efetuam um número de operações pré-definidas (carga de trabalho) e apresenta um resultado em algum formato (métrica), que relata o comportamento do sistema. As métricas podem calcular a velocidade para completar a carga de trabalho ou vazão, que seriam quantas cargas de trabalho por unidade de tempo foram medidas.

Segundo Gray [1993], um *benchmark* deve atender os seguintes critérios:

- relevância: As medidas verificadas deverão descrever funcionalidade e expectativa de desempenho acerca do produto submetido ao benchmark. É necessário especificar com clareza o domínio da utilização da ferramenta e as operações a serem submetidas;
- portabilidade: Na elaboração de um *benchmark*, as definições e os termos utilizados devem estar em um nível de abstração que permita transportá-lo para as implementações de diferentes ferramentas. Portanto, não deve haver privilégio na utilização de um contexto que seja particular a uma determinada abordagem. O objetivo desta consideração é tornar o benchmark aplicável a uma gama maior de ferramentas a serem avaliadas;
- escalabilidade: O padrão das medições deve atender a pequenos ou grandes sistemas, independente do nível de complexidade dos mesmos. Fatores como monoprocessoamento ou multiprocessoamento, clock e tamanhos de memórias não devem tornar as medidas inconsistentes, nem influenciar na aplicabilidade do benchmark;
- simplicidade: Os critérios definidos no *benchmark* devem ser simples e de fácil compreensão. Porém, a relevância das métricas não deve ser negligenciada. Medidas muito simples podem não refletir boas características de julgamento de uma ferramenta. No entanto, se for definido um conjunto de métricas muito complexas e de difícil assimilação pela comunidade interessada nos resultados, poderá ocorrer uma perda de credibilidade do benchmark.

#### 3.1. Casos de Uso

Existem várias implementações de benchmarks para realizar medidas de desempenho, nos mais diversos sistemas, como servidores WEB, SGBDs, aplicações gráficas, entre outros. Como exemplo, pode-se citar: benchmarks para banco de dados (AS<sup>3</sup>AP, DBT-2, OSDB, TPCC-UVA, TPC-C, Wiscosin, 007) e benchmarks para arquiteturas de CPUs (SPEC 95, SPEC 2000).

### 3.2. Benchmarks AS<sup>3</sup>AP

O benchmark AS<sup>3</sup>AP, segundo [CARLOS, S. 2011], possui, como características principais a amplitude na medição do desempenho das principais funcionalidades de um banco de dados e a variação de tipos de carga de trabalho aplicados. O *benchmark* é dividido em módulos que podem ser executados separadamente ou em conjunto. Cada módulo é responsável por testar uma certa funcionalidade. Algumas organizações de público, como universidades e laboratórios de pesquisa, criaram implementações de código aberto do *benchmark* AS<sup>3</sup>AP, como o *Open Source Database Benchmark* [OSDB 2001], da *Compaq Computer Corporation*®

O *benchmark* AS<sup>3</sup>AP oferece os seguintes recursos:

- fornece um único relatório através de uma métrica: efetuar uma rotina de avaliação de igualde do banco AS<sup>3</sup>AP em menos de 12 horas;
- poder de processamento de bancos de dados;
- Fornece uma interpretação uniforme e métrica simples de resultados de *benchmark*;
- as definições e os termos utilizados devem estar em um nível de abstração que permita transportá-lo para as implementações de diferentes ferramentas. Portanto, não deve haver privilégio na utilização de um contexto que seja particular a uma determinada abordagem;
- critérios ser simples e de fácil compreensão;
- não utilizar um conjunto de métricas muito complexas e de difícil assimilação;

Sistemas testados com o *benchmark* AS<sup>3</sup>AP devem suportar tipos de dados comuns e fornecer uma interface completa relacional com integridade básica, consistência e mecanismos de recuperação. O *benchmark* AS<sup>3</sup>AP pode testar sistemas que variam de um único usuário do Sistema de Gerenciamento de Banco de Dados microcomputador (SGBD) para uma paralela de alto desempenho ou banco de dados distribuído. Porém, a relevância das métricas não deve ser negligenciada. Medidas muito simples podem não refletir boas características de julgamento de uma ferramenta. No entanto, se for definido um pela comunidade interessada nos resultados, poderá ocorrer uma perda de credibilidade do *benchmark*.

### 3.3. Benchmarks TPC

Com o objetivo de definir *benchmarks* para avaliar o processamento de transações e banco de dados, surgiu em 1988 o TPC. Se trata de um consórcio sem fins lucrativos, que realiza avaliações em ambientes diferentes e publica relatórios de *benchmarks* denominados: TPC-C, TPC-H, TPC-W e TPC-App, esse último é o substituto do TPC-App. O conjunto de transações que compõem os testes é o que diferencia os diversos relatórios TPCs [<http://www.tpc.org/information/benchmarks.asp>]

### 3.3.1. Benchmarks TPC-C

O Benchmark TPC-C é utilizado para avaliar Sistemas Gerenciadores de Banco de Dados (SGBDs). É definida uma carga de processamento de transações que simula as atividades encontradas em um ambiente de aplicações complexas, caracterizado por:

- execução simultânea de múltiplos tipos de transações;
- testes em ambientes ou mono-usuários;
- testes em ambiente online multi-usuários;
- composta por um conjunto de transações de leitura e escrita;
- entrada / Saída de disco significantes;
- a métrica de desempenho definida e uma taxa medindo o numero de transações efetivadas com sucesso por minuto (tpmC) ou por segundo (tmsC).

O propósito do TPC-C é prover dados relevantes de desempenho que auxiliem os usuários a comparar de maneira objetiva dois ou mais sistema quanto a capacidade de processar transações, não deve haver privilégio na utilização de um contexto que seja particular a uma determinada abordagem. O objetivo desta consideração é tornar o *benchmark* aplicável a uma gama maior de ferramentas a serem avaliadas [GUIMARAES 2007].

## 4. Ambiente de Simulação

As definições de carga e os testes terão como ambiente de execução um notebook Dell XPS com processador core I7 2,2GHz com 8GB de memória RAM. O sistema operacional utilizado será o Windows 7 Ultimate SP1.

Banco de Dados	Ferramenta de Administração
Firebird 2.5	IBExpert 2012
MySQL 5.5	MySQL Front 5.1
PostgreSQL 9.2.1	pgAdmin III
SQL Server 2008 R2	SQL Server Management Studio 2008

**Tabela 1. SGBD'S e ferramentas de administração.**

O sistema operacional foi configurado apenas com seus controladores, sem nenhum outro componente que pudesse comprometer o ambiente. As ferramentas de gerenciamento foram configuradas de forma padrão conforme (tabela 1), para que desta forma nenhuma delas pudesse obter um ganho de desempenho nos testes. Todos os testes foram efetuados em banco local e sem nenhum tipo de componente on-line, conforme os modelos de orientação dos *benchmarks* AS<sup>3</sup>AP e TPC-C.

#### 4.1. Aplicação do *Benchmark AS<sup>3</sup>AP*

Seguindo o modelo do *benchmark AS<sup>3</sup>AP*, a criação da tabela de bancos de dados conforme a (tabela 2) , foi efetuada de forma simples e sem criação de índices ou chaves. Segundo a norma, para os testes não deve haver privilégios para nenhum SGBD que o propicie algum ganho de desempenho em relação a qualquer outro na avaliação. O banco deve ser iniciado e, após a criação de uma base limpa, executados os scripts.

Tabela (RICADPAC )	Tipo de dados
NOME	VARCHAR(45),
NASC	DATE,
MAE	VARCHAR(45),
SEXO	VARCHAR(1),
ENDERECO	VARCHAR(40),
NUMERO	INTEGER,
BAIRRO	VARCHAR(30),
CIDADE	VARCHAR(30),
UF	VARCHAR(2)

**Tabela 2. Estrutura da tabela.**

Ainda respeitando o critério do *AS<sup>3</sup>AP*, os comandos devem ser simples e de fácil compreensão. Desta forma, os comando foram criados e salvos no formato (.sql), que pode ser acessado por todas as ferramentas gerenciadoras dos SGBD's, e separados da seguinte forma:

- Para a criação da tabela RICADPAC, como mostra a listagem 1, foi criado um script de criação de tabela.

```
CREATE TABLE RICADPAC (  
  NOME VARCHAR(45) ,  
  NASC DATE ,  
  MAE VARCHAR(45) ,  
  SEXO VARCHAR(1) ,  
  ENDERECO VARCHAR(40) ,  
  NUMERO INTEGER ,  
  BAIRRO VARCHAR(30) ,  
  CIDADE VARCHAR(30) ,  
  UF VARCHAR(2)  
);
```

**Listagem 1. Script de criação da tabela.**

- Conforme a listagem 2, para o comando de inserção, foi gerado um script que faz uma inclusão de dados em todos os campos da tabela RICADPAC.

```
INSERT VALUES INTO RICADPAC(  
  NOME, NASC, MAE, SEXO, ENDERECO, NUMERO, BAIRRO, CIDADE, UF  
);
```

**Listagem 2. Script de inserção.**

- Em seguida foi gerado o script de consulta, que conforme a listagem 3 percorre todos os campos da tabela RICADPAC.

```
SELECT * FROM RICADPAC;
```

#### Listagem 3. Script de consulta.

- Já na listagem 4, pode-se ver o comando de alteração. Nesse comando todos os campos da tabela forma alterados de uma só vez.

```
UPDATE RICADPAC SET NOME = 'ERICK RODRIGUES FERREIRA ',  
NASC = '1986-11-11', MAE = 'DENISE MARIA RODRIGUES FERREIRA ',  
SEXO = 'M', ENDERECO = 'AVENIDA DOM PEDRO II ', NUMERO = '1660 ',  
BAIRRO = 'PASSARINHOS', CIDADE = 'BARBACENA', UF = 'MG'  
WHERE = 'NOME DO REGISTRO UTILIZADO NOS TESTES ';
```

#### Listagem 4. Script de alteração.

- Por último foi gerado um script que apaga todos dados inseridos nos campos da tabela RICADPAC. Como pode ser visto na listagem 5.

```
DELETE FROM RICADPAC;
```

#### Listagem 5. Script de exclusão.

O *benchmark* AS<sup>3</sup>AP em conjunto com o TPC-C também será responsável por avaliar o tamanho máximo da base de dados após os testes.

## 4.2. Aplicação do *Benchmark* TPC-C

Conforme dito anteriormente, o TPC-C complementa AS<sup>3</sup>AP, pois utiliza a estrutura padrão do banco, criada para a aplicação dos volumes de cargas, e, desta maneira foram utilizados três volumes de cargas distintos:

- Os volumes de entrada de dados foram separados por 1000,10.000 e 100.000 registros.
- O computador era iniciado e logo em seguida a ferramenta de administração do SGBD escolhido.

Seguindo o *benchmark* TPC-C, a métrica utilizada foi a que mede o desempenho do número de transações de entrada e saída efetivadas com sucesso por segundo (tmsC). Cada volume testado três vezes em cada SGBD, e o resultado final foi obtido por meio da media aritmética desses comandos. Para entender melhor esse processo, pode-se descrevê-lo em etapas:

- Ao se iniciar o primeiro SGBD, aplica-se o comando de inserção do primeiro volume, de 1000 registros.
- Em seguida, ainda na mesma aplicação aberta, efetuam-se os comandos de consulta, alteração e exclusão, obtendo-se, assim, o primeiro resultado.

- O computador é reiniciado, pois, desta forma, pode-se assegurar que não há interferência na memória, no processamento do computador, ou, até mesmo, no cash de dados que poderiam ficar armazenados, proporcionado, assim, um ganho para a próxima consulta.
- Repete-se esse mesmo teste mais duas vezes nesse SGBD, obtendo-se, assim, a média aritmética.
- Ao final do teste com o primeiro volume, inicia-se o segundo, utilizando-se do mesmo critério da média.
- Em seguida, foi aplicada a mesma metodologia anterior para o terceiro volume de cargas.

## 5. Resultados dos testes comparativos

Os resultados foram separados por três etapas, levando-se em consideração os volumes de dados aplicados:

- A primeira etapa, foi efetuada com o um volume 1000 registros.
- Na segunda, foram utilizados 10.000.
- Por último, foram usados 100.000 registros.

A seguir podem-se ver os resultados obtidos nas três etapas.

### 5.1. Resultados do primeiro teste

O primeiro teste foi feito com volume de 1000 registros e conforme as tabelas (3, 4, 5, 6), resultados foram descritos em segundos, lavando-se em consideração três casas após a vírgula.

O primeiro teste aplicado foi o de inserção.

MÉDIA DE INSERÇÃO - 1.000 REGISTROS				
SGBD	TEMPO 1	TEMPO 2	TEMPO3	MÉDIA
MySQL	0,360	0,350	0,391	0,367
FIREBIRD	0,453	0,421	0,390	0,421
PostgreSQL	0,112	0,176	0,156	0,148
SQL Server 2008	0,452	0,422	0,394	0,423

**Tabela 3. Média Aritmética de Inserção - 1000 registros.**

O segundo foi o teste de consulta dos dados que retornava como resultado todos os registros inseridos na tabela.

MÉDIA DE CONSULTA - 1.000 REGISTROS				
SGBD	TEMPO 1	TEMPO 2	TEMPO3	MÉDIA
MySQL	0,021	0,023	0,022	0,022
FIREBIRD	0,078	0,062	0,094	0,078
PostgreSQL	0,094	0,112	0,158	0,121
SQL Server 2008	0,171	0,144	0,126	0,147

**Tabela 4. Média Aritmética de Consulta - 1000 registros.**

A terceira média foi a de Alteração; era executado em cada tempo juntamente com os outros comandos do teste.

MÉDIA DE ALTERAÇÃO - 1.000 REGISTROS				
SGBD	TEMPO 1	TEMPO 2	TEMPO3	MÉDIA
MySQL	0,094	0,082	0,072	0,083
FIREBIRD	0,074	0,063	0,071	0,069
PostgreSQL	0,080	0,111	0,084	0,092
SQL Server 2008	0,078	0,054	0,031	0,054

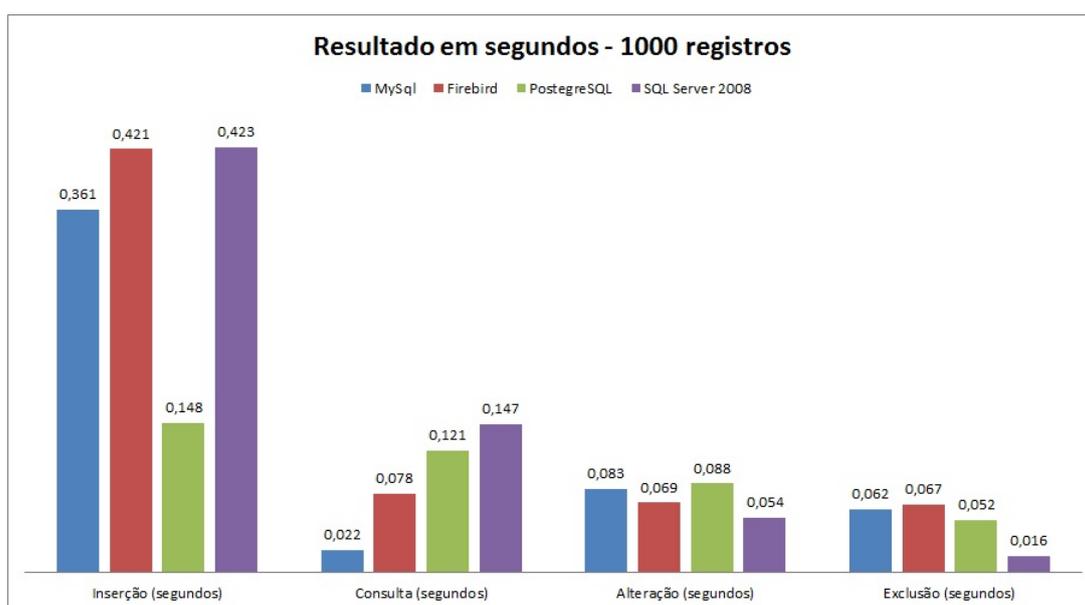
**Tabela 5. Média Aritmética de Alteração - 1000 registros.**

Por último, foi realizada a média dos tempos de exclusão de cada SGBD no volume de 1.000 registros.

MÉDIA DE EXCLUSÃO - 1.000 REGISTROS				
SGBD	TEMPO 1	TEMPO 2	TEMPO3	MÉDIA
MySQL	0,074	0,061	0,051	0,062
FIREBIRD	0,069	0,069	0,062	0,067
PostgreSQL	0,050	0,059	0,047	0,052
SQL Server 2008	0,015	0,016	0,017	0,016

**Tabela 6. Média Aritmética de Exclusão - 1000 registros.**

Como mostra a figura 2, após a primeira de testes com 1000 registros já foi possível verificar que o PostgreSQL apresentou o menor tempo de inserção de dados apresentando um resultado quase três vezes menor que a media dos outros três.



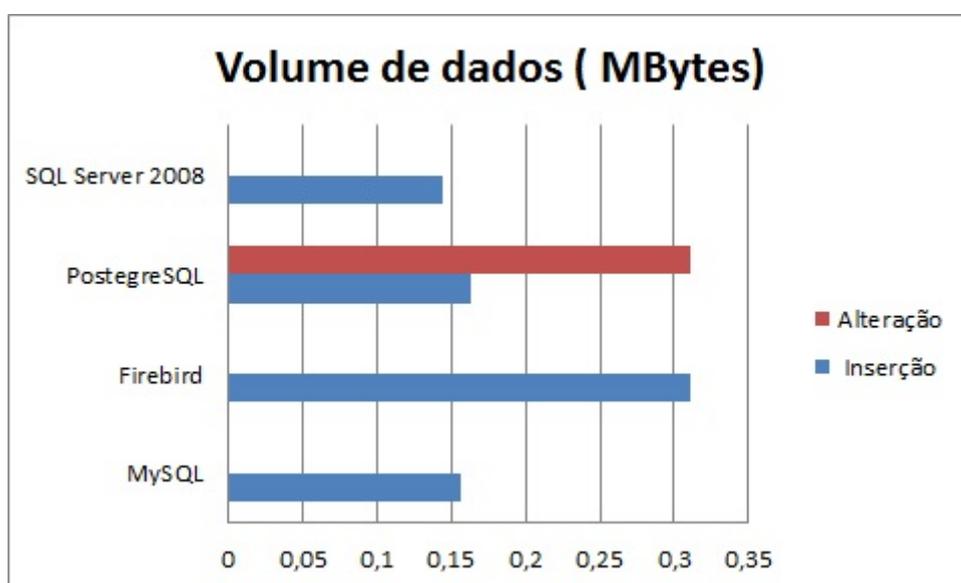
**Figura 2. Resultados de tempo (segundos) - 1000 registros**

Firebird foi o que apresentou o maior custo no volume de dados quanto à inserção. É possível ver na tabela 7 que ele obteve um aumento na base de 0,311 M Bytes.

Tamanho da Base de Dados - VOLUME DE 1000 REGISTROS	MySQL	Firebird	PostgreSQL	SQL Server 2008
Inserção (MB)	0,156	0,311	0,163	0,144
Alteração (MB)	-	-	0,311	-

**Tabela 7. Teste com volume de carga com 1000 registros - Crescimento da base de dados**

Já o PostgreSQL, conforme ilustrado na figura 4, foi o único SGBD que teve o tamanho da base de dados alterado com o comando UPDATE.



**Figura 3. Gráfico de Resultados do Primeiro Teste - Crescimento da base de dados.**

## 5.2. Resultados do segundo teste

O segundo teste foi feito com volume de 10.000 registros e, conforme as tabelas (8, 9, 10, 11) mostram, os resultados foram satisfatórios.

O primeiro teste aplicado foi o de inserção de 10.000 registros.

MÉDIA DE INSERÇÃO - 10.000 REGISTROS				
SGBD	TEMPO 1	TEMPO 2	TEMPO3	MÉDIA
MySQL	3,768	4,371	3,790	3,976
FIREBIRD	3,931	3,806	3,666	3,801
PostgreSQL	1,513	1,385	1,260	1,386
SQL Server 2008	3,192	3,265	3,373	3,277

**Tabela 8. Média Aritmética de Inserção - 10.000 registros.**

O segundo foi o teste de que utilizou o comando de seleção de todos os dados contidos na tabela.

MÉDIA DE CONSULTA - 10.000 REGISTROS				
SGBD	TEMPO 1	TEMPO 2	TEMPO3	MÉDIA
MySQL	0,086	0,070	0,061	0,072
FIREBIRD	0,172	0,153	0,172	0,166
PostgreSQL	0,982	0,968	0,999	0,983
SQL Server 2008	0,183	0,210	0,211	0,201

**Tabela 9. Média Aritmética de Consulta - 10.000 registros.**

A terceira média foi a de Alteração. Nesse comando pode ser notado uma grande igualdade entre os bancos.

MÉDIA DE ALTERAÇÃO - 10.000 REGISTROS				
SGBD	TEMPO 1	TEMPO 2	TEMPO3	MÉDIA
MySQL	0,484	0,560	0,406	0,483
FIREBIRD	0,265	0,234	0,249	0,249
PostgreSQL	0,266	0,166	0,390	0,274
SQL Server 2008	0,541	0,727	0,599	0,622

**Tabela 10. Média Aritmética de Alteração - 10.000 registros.**

Por último, foi realizada a exclusão de todos os registros da tabela.

MÉDIA DE EXCLUSÃO - 10.000 REGISTROS				
SGBD	TEMPO 1	TEMPO 2	TEMPO3	MÉDIA
MySQL	0,133	0,144	0,122	0,133
FIREBIRD	0,079	0,079	0,074	0,077
PostgreSQL	0,110	0,125	0,104	0,113
SQL Server 2008	0,146	0,082	0,150	0,126

**Tabela 11. Média Aritmética de Exclusão - 10.000 registros.**

Como pode-se notar na figura 5, mesmo com um volume de dados dez vezes maior o PostreSQL continuou apresentando o menor custo na inserção dos dados, mas a contrapartida passou a ter um desempenho muito ruim na consulta dos dados.

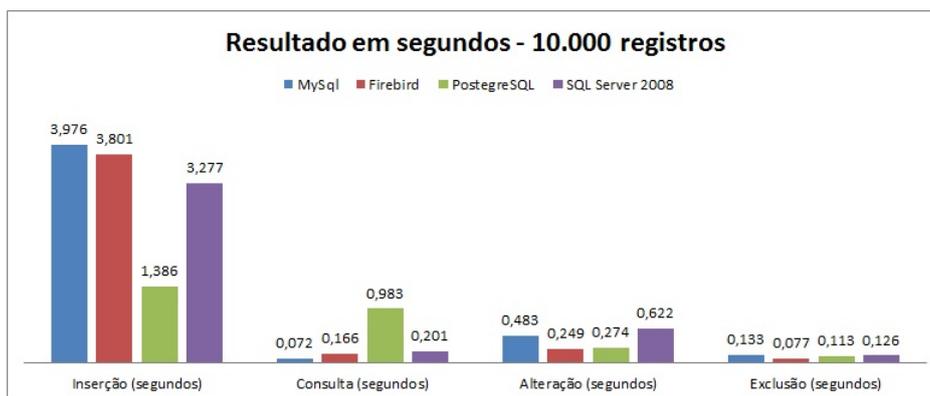


Figura 4. Resultados de tempo (segundos) - 10.000 registros.

Já na tabela 12, o Firebird continuou apresentando o maior aumento na base de dados, quando utilizado o comando de inserção.

Tamanho da Base de Dados - VOLUME DE 10.000 REGISTROS	MySQL	Firebird	PostgreSQL	SQL Server 2008
Inserção (MB)	1,5	1,998	1,499	1,361
Alteração (MB)	-	1,212	2,991	-

Tabela 12. Teste com volume de carga com 10000 registros - Medida de tempo.

Na soma das bases, após os comandos de alteração e de inserção, nota-se no figura 6 que o PostgreSQL teve o maior aumento na base de dados.

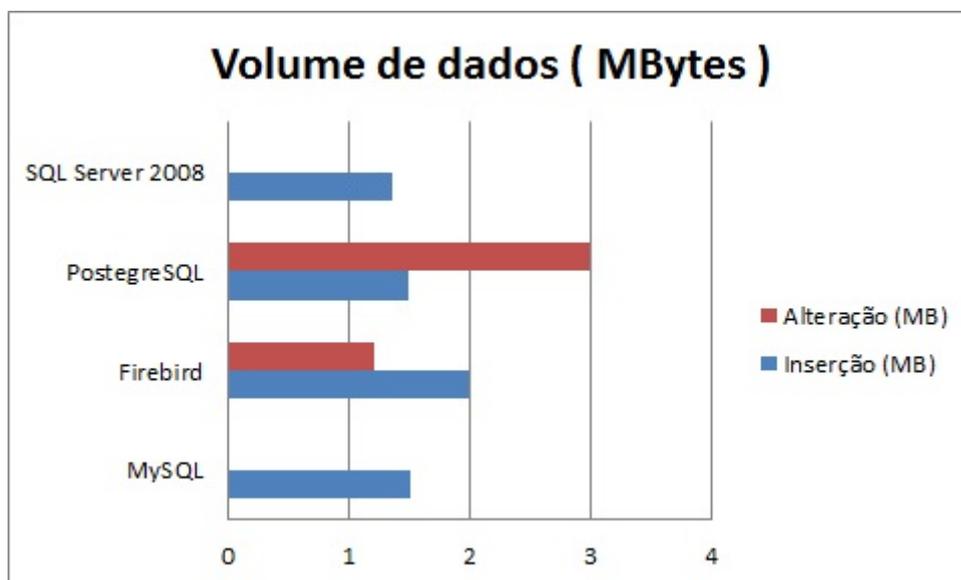


Figura 5. Gráfico com resultados do segundo teste - Crescimento da base de dados.

### 5.3. Resultados do terceiro teste

O terceiro teste foi feito com volume de 100.000 registros e, conforme as tabelas ( 13, 14, 15, 16), os resultados que foram obtidos na medida de tempo em segundos, sendo

consideradas três casa após a vírgula.

O primeiro teste aplicado foi o de inserção.

MÉDIA DE INSERÇÃO - 100.000 REGISTROS				
SGBD	TEMPO 1	TEMPO 2	TEMPO3	MÉDIA
MySQL	39,800	43,108	34,701	39,203
FIREBIRD	39,177	39,890	39,872	39,646
PostgreSQL	15,761	12,851	13,203	13,938
SQL Server 2008	37,445	35,826	36,904	36,725

**Tabela 13. Média Aritmética de Inserção - 100.000 registros.**

O segundo foi o teste de consulta dos dados; e retornava como resultado todos os registros inseridos na tabela.

MÉDIA DE CONSULTA - 100.000 REGISTROS				
SGBD	TEMPO 1	TEMPO 2	TEMPO3	MÉDIA
MySQL	0,117	0,125	0,093	0,112
FIREBIRD	0,731	0,731	0,737	0,733
PostgreSQL	6,374	6,766	6,672	6,604
SQL Server 2008	1,200	1,140	1,607	1,316

**Tabela 14. Média Aritmética de Consulta - 100.000 registros.**

A terceira média foi a de Alteração. Ele foi executado em cada tempo juntamente com os outros comandos estipulados por teste.

MÉDIA DE ALTERAÇÃO - 100.000 REGISTROS				
SGBD	TEMPO 1	TEMPO 2	TEMPO3	MÉDIA
MySQL	4,942	5,002	4,473	4,806
FIREBIRD	1,762	1,804	1,771	1,779
PostgreSQL	0,861	0,823	0,917	0,867
SQL Server 2008	4,973	5,169	5,413	5,185

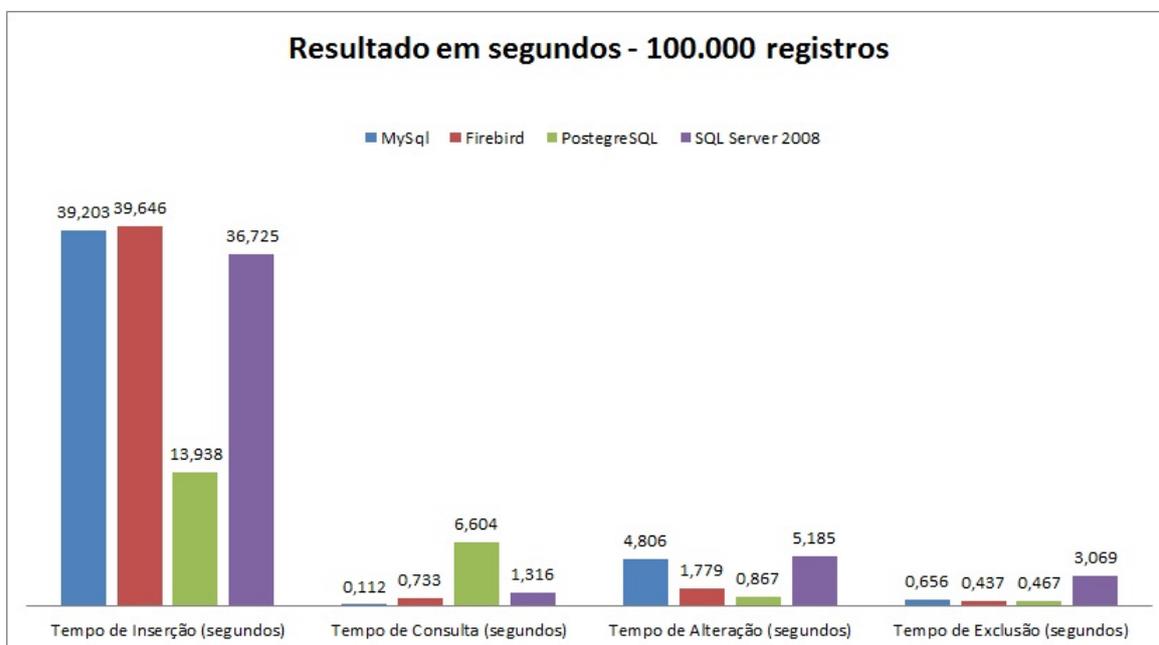
**Tabela 15. Média Aritmética de Alteração - 100.000 registros.**

Por último, foi realizada a média dos tempos de exclusão de cada SGBD no volume de 1.000 registros.

MÉDIA DE EXCLUSÃO - 100.000 REGISTROS				
SGBD	TEMPO 1	TEMPO 2	TEMPO3	MÉDIA
MySQL	0,532	0,713	0,722	0,656
FIREBIRD	0,468	0,407	0,437	0,437
PostgreSQL	0,795	0,263	0,343	0,467
SQL Server 2008	2,617	3,188	3,402	3,069

**Tabela 16. Média Aritmética de Exclusão - 100.000 registros.**

No teste a seguir pode-se chamar a atenção para diferença nos resultados, quando tratado o comando de inserção de dados da figura 6.



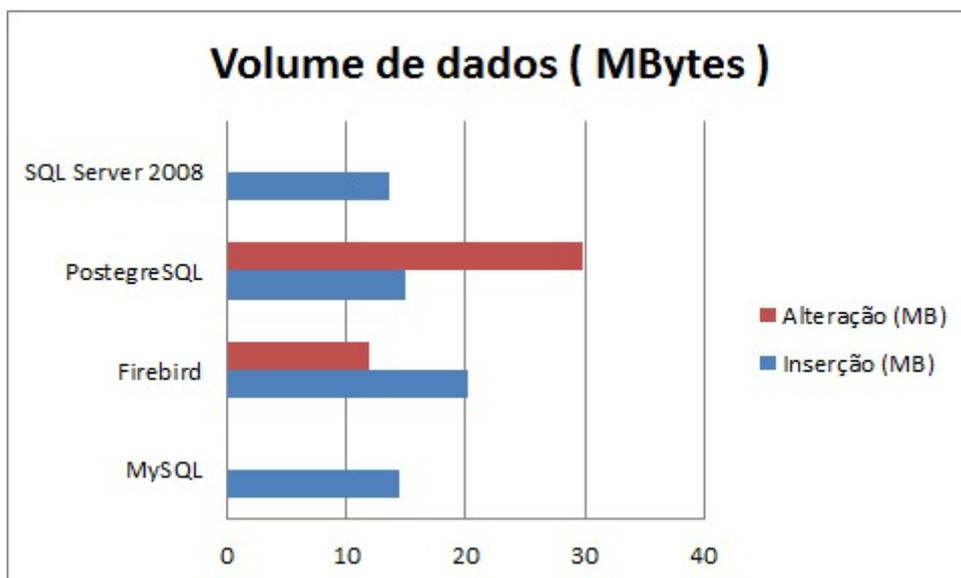
**Figura 6. Resultados de tempo (segundos) - 100.000 registros.**

No terceiro e último teste, o Firebird mostrou que é o banco de dados que mais exige espaço para o comando de inserção, 20,217 MB conforme tabela 17.

Tamanho da base de dados - Volume de 100.000 registros	MySQL	Firebird	PostgreSQL	SQL Server 2008
Inserção (MB)	14,5	20,217	14,909	13,562
Alteração (MB)	0	11,944	29,802	0

**Tabela 17. Teste com volume de carga com 10000 registros - Medida de tempo.**

Apesar de ser o SGBD que teve o melhor desempenho em relação ao tempo medido para os comandos de inserção e alteração, o PostgreSQL obteve um aumento total, em relação a sua base inicial, de 44,711 MBytes, ilustrado na figura 7.



**Figura 7. Gráfico com resultados do terceiro teste - Crescimento da base de dados.**

## 6. Conclusão

É importante ressaltar nesta conclusão que todos os resultados obtidos refletem o ambiente de teste no qual os bancos de dados foram colocados nesta pesquisa. Os resultados finais não podem ser levados em consideração como um padrão para qualquer outro tipo de ambiente ou aplicação. Os bancos de dados podem levar uma vantagem em relação a outro, quando considerados o ambiente no qual vão ser instalados, que plataforma que irão utilizar e principalmente as configurações que são efetuadas no início de sua implantação. Outro ponto de muita importância é que nesse artigo todos os comandos foram feitos de formas iguais para todos os SGBD's. Deve ser levado em consideração que no momento da escolha de um banco de dados é possível ganhar desempenho nessas mesmas consultas, pois cada banco de dados oferece recursos para aplicá-las de formas diferentes e mais eficientes.

Em todos os testes o banco que teve o melhor tempo para a inserção dos dados foi o Postgresql. Assim, pôde-se constatar que é um banco que, quanto maior o volume de dados inserido, melhor será o tempo de resposta em relação aos outros bancos de dados.

Ao fim do terceiro teste, pode-se concluir que o PostgreSQL é indicado para aplicações mais robustas, que recebem um grande número de inserções durante seu funcionamento. Esse banco de dados também foi o que apresentou o menor tempo, quando realizados os procedimentos de alteração de maior volume de dados.

Ao mesmo tempo em que é um banco ágil quando tratados comandos de INSERT E UPDATE, o PostgreSQL é um banco que ocupa muito espaço no disco físico, o que exige uma máquina com grande capacidade de armazenamento. Foi notado que a custo de ser mais rápido nas inserções, ele cobra essa parcela do HD.

O PostgreSQL também foi o banco que teve o pior desempenho, quando realizado o comando para consulta dos dados. Conforme (figura8), o custo para consultas de

100.000 foi muito alto, representando quase uma medida de tempo seis vezes maior que o MySQL.

Já o MySQL, foi o SGBD que obteve menor tempo de consulta em todos os testes, se mostrando um banco ágil quando a necessidade da aplicação for de uma leitura rápida para acesso aos dados.

O SQL Server 2008 levou uma pequena vantagem no tamanho da base de dados em todos os testes realizados. Em contra partida foi o que teve o pior desempenho na exclusão dos dados no último teste. O SGBD da Microsoft foi o único banco proprietário utilizado nos testes. Ele teve ganhos significativos apenas no primeiro teste com os comando de alteração e exclusão. É um banco de dados que possui inúmeros recursos que não foram levados em consideração na metodologia aplicada, portanto, não se pode dar uma definição conclusiva que é uma ferramenta que leva muita desvantagem em relação às outras.

Por fim, o firebird obteve o melhor desempenho no segundo e no terceiro testes, mas que ficaram muitos próximos de outros bancos de dados. Pode-se usar o mesmo comentário feito ao SQL Server para esse SGBD, pois é uma ferramenta que possui muitos recursos que não foram aplicados ao ambiente proposto.

Observa-se que este estudo poderá ser complementado nessa mesma linha de pesquisa, assim sugerindo as análises dos SGBD's, utilizando outros critérios como:

- análise pelas indexações do registros;
- análise da estrutura de gerenciamento do "buffer";
- outras cláusulas de operações da linguagem SQL;
- volumes de cargas maiores;
- processamento de dados em uma ambiente multi-usuário;

## Referências

- BARRIE, H., *Dominando Firebird: Uma Referência para Desenvolvedores de Bancos de Dados*. ED. Ciência Moderna , 2006.
- CARLOS, S., *Comparativo de desempenho de Bancos de dados de Código Aberto*. (UFPE), 2011 .
- CARNEIRO, ALESSANDRO., *Técnicas de Otimização de Bancos de Dados, Um estudo Comparativo: MySQL e PostgreSQL*. (FURG), 2011.
- FIREBIRD., <http://firebird.sourceforge.net> - Consulta Realizada 22/05/12.
- GESTÃO, REVISTA CIENTÍFICA DE ADMINISTRAÇÃO E SISTEMAS DE INFORMAÇÃO/UNIDADE DE ENSINO SUPERIOR, *Expoente*. V.11, n.11. Jul. /Dez. 2008.
- GRAY, J., *Database and Transaction Processing Performance Handbook (1993)*, Chapter 1. Morgan Kaufman Publishers.
- GUIMARÃES, LORENA., *Um Aplicativo para Teste de Performances de Bancos de Dados Através dos Benchmarks TPC-C e TPC-HL*.2007. <http://www.lia.ufc.br/site> - Consulta realizada 12/03/2012.

HOTEK, MIKE., *Microsoft SQL Server 2008, Implementação e Manutenção. Kit de Treinamento, 2010.*

LEITE, M., *Acessando Bancos de Dados com Ferramentas RAD. Braspor, Ed. 2007.*

MENDES, R., *Pernambuco (UFPE), 2006. [http://. cin.ufpe.br/ tg/ 2006 - 1/mrnm.pdf](http://cin.ufpe.br/tg/2006-1/mrnm.pdf) - Consulta realizada 19/03/2012 .*

POSTGRESQL., *<http://www.postgresql.org/> - Consulta Realizada 27/03/12.*

SILBERSCHATZ, ABRAHAM, *Sistemas de Bancos de Dados. Terceira ED. 1999.*

SILBERSCHATZ, ABRAHAM., KORT E SUDARSHAM. *Sistemas de Bancos de Dados. Quinta ED. 2006.*

SVOBODOVA, L., *Computer Performance Measurement and Evaluation Methods : Analysis and Applications .Vol. 2 .*

TPC., *<http://www.tpc.org/information/benchmarks.asp>.*