



# DESEMPENHO DE DADOS EM MODELOS DE RELACIONAMENTOS POR HERANÇA

- Revisão Bibliográfica -

ANDRÉ CARDEAL SANTANA<sup>1</sup>; SUELI DE FATIMA POPPI BORBA<sup>2</sup>

<sup>1</sup> Universidade Paranaense – Unipar. Instituto Superior de Ciências Exatas, Agrárias, Tecnológicas e Geociências. Av Brasil 1123 Cianorte PR, Brasil. E-mail: andrecardeal@hotmail.com

<sup>2</sup> Universidade Paranaense – Unipar. Instituto Superior de Ciências Exatas, Agrárias, Tecnológicas e Geociências. Av Brasil 1123 Cianorte PR, Brasil. E-mail: sueli@unipar.br

---

## RESUMO

*Este artigo tem por objetivo apresentar um benchmarking entre os modelos de dados semelhantes, mas em visão diferenciada, implementando herança e agrupando os atributos em uma única entidade (não implementação de herança). Para este teste foi escolhido um dos casos clássicos da modelagem relacional, Cadastro de Pessoa. Em um modelo convencional implementaríamos, Pessoa, Pessoa Física e Pessoa Jurídica em uma única entidade, já no modelo com a implementação de herança teremos três entidades distintas. Seguindo a determinação dos procedimentos relacionados em cada uma das etapas, o trabalho realiza uma simulação a partir de dados reais, testando medidas e performance para ambos os modelos. Para tais testes usaremos o banco de dados Oracle 10g.*

**Palavras-chave:** Banco de dados; herança; benchmark

## PERFORMANCE OF DATA IN MODELS RELATED TO INHERITANCE

### ABSTRACT

*This article has for objective to present one benchmarking between the models of similar data, implementing inheritance and grouping the attributes in an only entity (not implementation of inheritance), for this test one of the classic cases of the relationary modeling was chosen, Registers in cadastre of Person, in a conventional model we would implement., Person, Natural person and Legal entity in an only entity, no longer model with the implementation of inheritance we will have three distinct entities. Following the determination of the procedures related in each one of the stages, the work carries through a simulation from real data testing measures and performance for both models. For such tests we will use the data base Oracle 10g.*

**Key-Words:** Data base; inheritance; benchmark

## INTRODUÇÃO

A crescente demanda por sistemas de informação desenvolvidos sob o paradigma da orientação a objetos abre espaço para as questões de persistência de dados na forma de objetos. A tecnologia de banco de dados relacional - BDR, amplamente utilizada em sistemas comerciais, começa a implementar características da orientação a objetos, criando uma extensão para o modelo relacional – objeto-relacional. Porém, os bancos de dados, objeto relacional – BDOR, começam a ser utilizados em escala comercial, através de novos produtos que implementam efetivamente a persistência dos objetos, além das demais características da orientação a objetos.

Neste contexto, este trabalho tem por objetivo desenvolver uma aplicação de teste de performance para banco de dados, visando comparar modelos de dados relacional e objeto relacional. Para isso foi escolhido um modelo clássico da modelagem relacional, em que usamos as entidades: pessoa, pessoa física e pessoa jurídica, comparando assim performance entre os dois modelos.

O presente artigo está organizado em 7 sessões : A sessão 2 aborda os conceitos de benchmark. A sessão 3 aborda vantagens e na utilização de um banco de dados objeto relacional. A sessão 4 aborda conceitos de modelos utilizando herança. A sessão 5 aborda conceitos de utilização de índices em atributos de uma entidade do banco de dados. A sessão 6 é apresentada a aplicação do benchmark segundo as etapas previamente definidas. A sessão 7 apresenta a aplicação

dos testes de desempenhos que foram realizados e seus resultados. A sessão 8 apresenta a conclusão que foi obtida com os testes efetuados.

## BENCHMARK

Benchmarking é o "processo por meio do qual uma empresa adota e/ou aperfeiçoa os melhores desempenhos de outras empresas em determinada atividade" (AURÉLIO, 2004).

"O processo de comparação do desempenho entre dois ou mais sistemas é chamado de benchmarking, e as cargas usadas são chamadas de benchmark" (SOUZA, 2003). Portanto, na área de informática, considera-se benchmarks programas que têm por finalidade medir a performance de componentes de sistemas de computação.

A necessidade dos benchmarks está, principalmente, na comparação de diferentes sistemas computacionais, a fim de verificar a melhor relação custo/benefício. Além disso, é extremamente importante nos testes de eficiência entre hardware/compilador. A utilização dos benchmarks não se limita a estas aplicações computacionais, porém, podem ser consideradas as mais comuns.

Existe grande discussão sobre a padronização dos benchmarks e de suas respectivas unidades de medida. Podem ser citados o comitê PARKBENCH (PARAllel Kernels and BENCHmarks - <http://www.netlib.org/parkbench/>) para benchmarking de sistemas paralelos, SPEC (Standard Performance Evaluation Corporation - <http://www.spec.org/>), benchmarking de processadores, TPC (Transaction Processing Performance Council - <http://www.tpc.org>) para banco de dados e 3DMark (<http://www.futuremark.com>) para processamento gráfico.

Benchmarks específicos para banco de dados objeto relacional podem ser analisados nos benchmarks Object Operation 1 - OO1 (CATTELL, 1991) ou OO7 (CAREY, 1993). O trabalho de DARMONT (2002) faz um estudo comparativo desses benchmarks exclusivos para comparar as características de banco de dados orientados a objetos ou objeto relacional.

Trabalhos com outras propostas de benchmarks podem ser encontrados para

diversos domínios, como sistemas de bases de dados (VIEIRA, 2003), sistemas operacionais (KALAKECH, 2004) e hardware (ZHU, 2003).

Deste modo, as medidas de um benchmark devem ser entendidas como resultados úteis somente para caracterizar a confiabilidade de sistemas de uma forma relativa, comparando diferentes alternativas.

A TPC é uma organização que tem o objetivo de definir padrões de benchmark para a indústria da computação, através da medição do número de transações por minuto que um ambiente computacional é capaz de realizar. Nos benchmarks TPC, os fabricantes de hardware e software trabalham juntos com a equipe de realização dos testes para buscar seu diferencial competitivo dentro das regras definidas pelo padrão de benchmark. O teste permite comparar, de uma vez só, hardware, sistema operacional, banco de dados e os componentes de cada sistema. Dessa forma, pode-se comparar produtos diferentes em plataformas diferentes, obedecendo a um mesmo conjunto de regras. A TPC possui atualmente definidos 3 padrões de benchmark: TPC-C (processamento on-line de transações), TPC-H (para sistemas de tomada de decisão) e TPC-W (para e-commerce). O TPC-H utiliza esquemas na 3FN, baseados no modelo ER. O workload deste benchmark consiste da execução de vinte e duas consultas e duas funções de atualização a partir de uma base de dados, em modo mono e multi-usuário (POESS, 2000).

Um benchmark possui algumas características para atender as especificações e provar sua validade, como definido nos estudos de GRAY (1993) através da representatividade, portabilidade, confiabilidade, repetibilidade, escalabilidade e não intrusível. De modo a fornecer resultados relevantes, um benchmark deve ser representativo de sistemas reais dentro do seu domínio de aplicação. As medidas estão fortemente relacionadas com o domínio de aplicação do benchmark e devem caracterizar os sistemas alvo de uma forma adequada e útil (GRAY, 1993).

Um benchmark deve ser portátil de modo a permitir a comparação de diferentes sistemas ou componentes no seu domínio de aplicação. A melhor forma de avaliar a portabilidade consiste em implementar e executar o benchmark num conjunto de ambientes representativos.

De modo a fornecer resultados confiáveis, um benchmark deve reportar resultados similares quando executado mais de uma vez no mesmo ambiente. A reprodutibilidade de resultados deve ser encarada de uma forma estatística, uma vez que é virtualmente impossível reproduzir de uma forma exata as mesmas condições, em termos do estado do sistema alvo, durante a execução do benchmark.

A repetibilidade do benchmark pode ser verificada executando o benchmark repetidamente no mesmo sistema. Os resultados são analisados e as variações são identificadas. A variação máxima representa o intervalo no qual os sistemas são considerados como tendo resultados semelhantes.

Um benchmark deve ser capaz de avaliar sistemas de escalas diferentes e (quando possível) permitir a comparação de sistemas de tamanhos distintos. A melhor forma de avaliar a escalabilidade é através da implementação e utilização do benchmark em sistemas de dimensões distintas. Um benchmark deve requerer poucas (ou nenhuma) alterações no sistema alvo. Um benchmark é intrusivo quando, para a sua execução, é necessário realizar modificações no sistema em teste. Essas modificações são normalmente dispendiosas e reduzem a portabilidade do benchmark. A intrusividade poderá ser avaliada durante a fase de implementação.

## VANTAGENS BDOR

Esta sessão tem por objetivo abordar pontos positivos e negativos do uso de um banco de dados objeto relacional e relacional. O modelo relacional combina os benefícios do modelo Relacional com a tecnologia de Orientação a Objetos sendo o modelo de dados objeto relacional é uma extensão do modelo Relacional onde:

- permite estender o banco de dados com tipos e funções específicas da aplicação
- A linguagem de consulta OR (SQL3) é uma extensão da linguagem SQL para suportar o modelo de objetos
- As extensões incluem consultas envolvendo objetos, atributos multivalorados, TADs, métodos e funções como predicados de busca em uma consulta

- Utiliza referências para representar conexões inter-objetos tornando as consultas baseadas em caminhos de referência mais compactas do que as consultas feitas com junção

- Herança é implementada organizando os tipos em hierarquias

- Atendem aos requisitos das novas aplicações e da nova geração de aplicações de negócios

- As extensões incluem consultas envolvendo objetos, atributos multivalorados, métodos e funções como predicados de busca em uma consulta.

## HERANÇA

É um mecanismo de reusabilidade muito poderoso. Com herança, uma classe chamada uma subclasse pode ser definida com base na definição de outra classe chamada a superclasse. A subclasse herda os atributos, métodos e mensagens de sua superclasse e pode ter atributos específicos, métodos e mensagens adicionais.

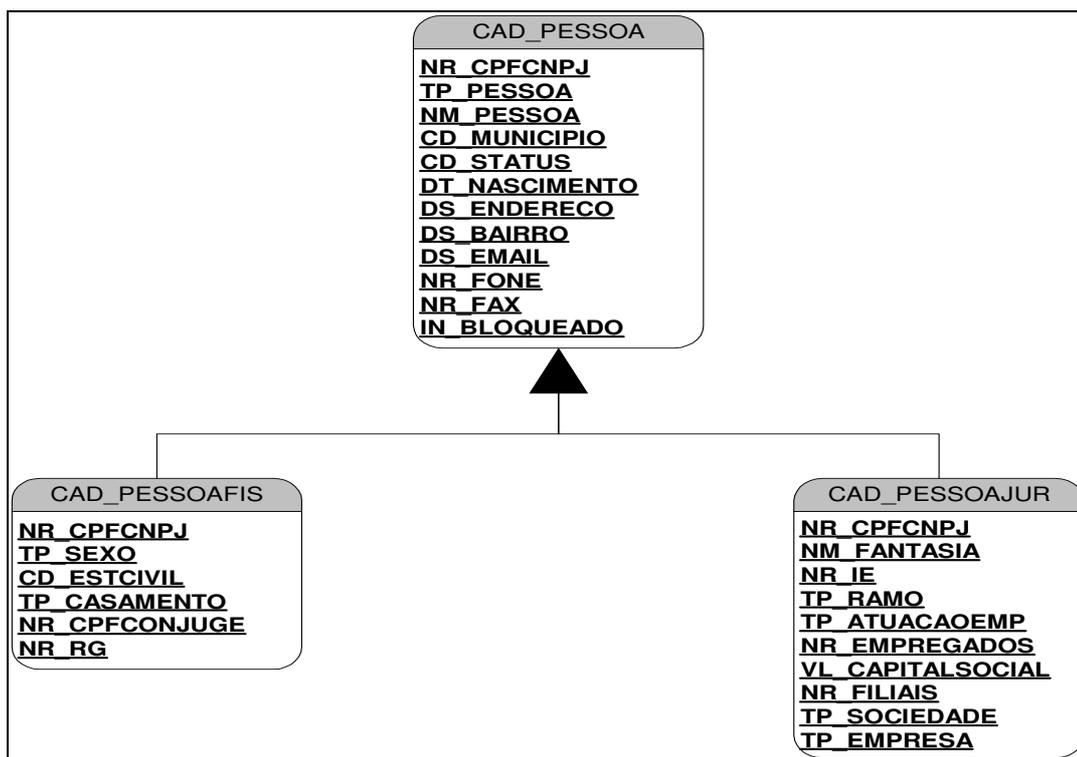
Vantagens da utilização de hierarquias de classe:

- diminui a quantidade de código a ser escrito;
- propicia uma descrição mais precisa e concisa da realidade.

Em certos sistemas, uma classe pode ter várias superclasses, em cujo caso diz-se que ocorre herança múltipla, enquanto outros impõem a restrição de uma única superclasse, dita herança simples.

A herança múltipla pode provocar problemas de conflitos, como por exemplo, duas ou mais superclasses podem ter um atributo com o mesmo nome, mas com diferentes domínios. Esses conflitos precisam ser tratados pelo sistema. Se existe uma relação de inclusão entre os domínios, então o domínio mais específico será escolhido como o domínio para a subclasse.

A Figura 1 representa um modelo de herança onde as entidade pessoajur e pessoafis herdam os dados da entidade pessoa.



**Figura 1.** Modelo representando herança.

## ÍNDICES

Um índice é uma estratégia de otimização de consulta para implementações desse modelo. Isso acontece porque o Modelo Relacional é um modelo teórico, porém quando transformado em uma implementação, as suas operações (tanto da álgebra quanto do cálculo) exigem um tempo de busca. Se os dados estiverem ordenados de uma forma ideal, esse tempo de busca pode ser bastante reduzido.

Os índices funcionam como catálogos organizados de forma a beneficiar certo tipo de consulta. É possível imaginar que uma tabela seria associada a quantidade máxima de índices, porém isso traria muitos gastos de inserção e alteração. Por isso, em uma implantação de um banco de dados, são escolhidos os índices mais adequados. Normalmente a chave primária é um índice.

O limitador é imposto exclusivamente pela ferramenta de software utilizada. Sendo assim, para a recuperação dos dados é necessário a existência de mecanismos que facilitem a consulta, proporcionando uma performance aceitável para a mesma. Para isso, os sistemas de bancos de dados relacionais criam índices das tabelas, sendo

que esses índices são atualizados constantemente.

Caso o índice se corrompa por algum motivo, é possível que pesquisas possam retornar resultados não desejados ou que inserções de chaves duplicadas aconteçam. Nesse caso o banco de dados será corrompido também. Os sistemas de bancos (bases) de dados possuem mecanismos de evitar que esses eventos ocorram como também possibilitam a recuperação dos índices e consistência da tabela caso eles ocorram.

## METODOLOGIA

Neste trabalho são propostas as seguintes etapas para a especificação de um teste de desempenho, baseadas no processo de benchmarking desenvolvido por ZEE (2002), que serão utilizadas na seção 7 desse artigo para a aplicação dos testes:

- Etapa 1 - determinar o escopo do problema;
- Etapa 2 - selecionar padrões e/ou banco de dados;
- Etapa 3 - identificar medidas;

- Etapa 4 - coletar ou gerar dados;
- Etapa 5 - analisar dados verificando os diferentes resultados.

As etapas definidas visam atender a um conjunto de propriedades (VIEIRA, 2005):

- Representatividade, pois o benchmark tem que representar sistemas reais;
- Repetibilidade, em que o benchmark deve produzir resultados semelhantes quando executada mais que uma vez no mesmo ambiente;
- Simplicidade de uso, considerando que o benchmark deve ser fácil de implementar e utilizar.
- Os principais componentes do teste de benchmark realizado são:
  - Medidas: caracterizam o desempenho das aplicações em avaliação;
  - Procedimentos e regras: descrição dos procedimentos e regras que devem ser seguidos durante a implementação do benchmark;
  - Ambiente experimental: descreve o ambiente de teste para a execução do benchmark.
  - A definição das regras e procedimentos baseia-se no trabalho proposto por Vieira et al. (2005), lembrando que tais regras dependem de cada benchmark, e que as regras e procedimentos são indicadores genéricos:
    - Condições uniformes para gerar o ambiente experimental, executando as tarefas de inicialização necessárias para a execução do benchmark;
    - Definição do nível de instrumentação do sistema, grau máximo de interferência

permitido e precisão para as medidas temporais;

- Definição de regras de escalabilidade para adaptar o benchmark a sistemas de tamanhos distintos, definindo a forma como deve ser alterada a carga de trabalho que o sistema deve executar.

## APLICAÇÃO

Etapa 1 – Determinando o escopo do problema

Nessa etapa é definido o domínio de aplicação através da geração do modelo de negócio a ser testado (ou módulo), além da definição da configuração do equipamento a ser utilizado no projeto, com suas respectivas características. Considerando o paradigma da orientação a objetos, o modelo será representado através do diagrama de classes. O presente artigo restringe-se especificamente ao modelo multidimensional de uma aplicação onde as entidades aplicadas fazem parte de um contexto para a movimentação de pessoa, pessoa física e pessoa jurídica, conforme mostra a figura 2. No banco de dados Oracle, as classes foram implementadas em duas etapas, uma usando o conceito objeto relacional e usando objetos e outra usando o conceito relacional usando apenas tabelas, segundo o modelo representado na figura 3. Essa etapa visa atender a propriedade de representatividade, caracterizada através da utilização de dados reais de uma aplicação.

Etapa 2 - Selecionar padrões e/ou banco de dados

O objetivo do presente trabalho é comparar o modelo de banco de dados relacional com o modelo objetos relacional. O quadro 1 mostra os componentes de sistema e hardware para a realização dos testes. Foi selecionado o banco de dados relacional Oracle 10g e também será utilizado o mesmo banco para os testes em objeto relacional.

**Tabela 1.** Componentes do Sistema e Hardware

<b>Processador</b>	<b>AMD Athlon XP64 3.0</b>
Memória	512 Mb
HD	80 Gb
Sistema Operacional	Windows XP Home
SGBD	Oracle 10g Express

Etapa 3 - Identificar medidas

O benchmark aqui proposto visa documentar medidas de tempo para a execução de uma determinada tarefa. Essas medidas são utilizadas para comparações entre os modelos implementados no mesmo banco de dados, Oracle, com as mesmas

condições e configurações de hardware e sistema. As medidas de desempenho baseiam-se no tempo de resposta (ou latência), definido como o intervalo de tempo entre uma requisição e a resposta do sistema.

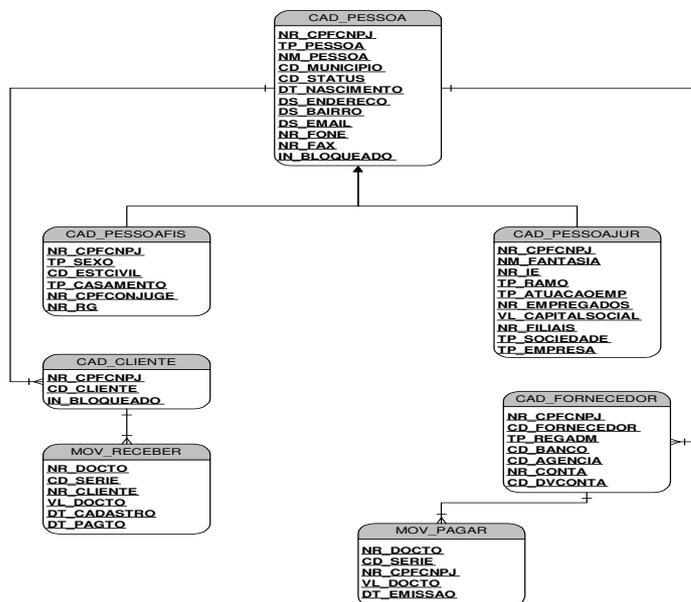


Figura 2. Modelo utilizando herança.

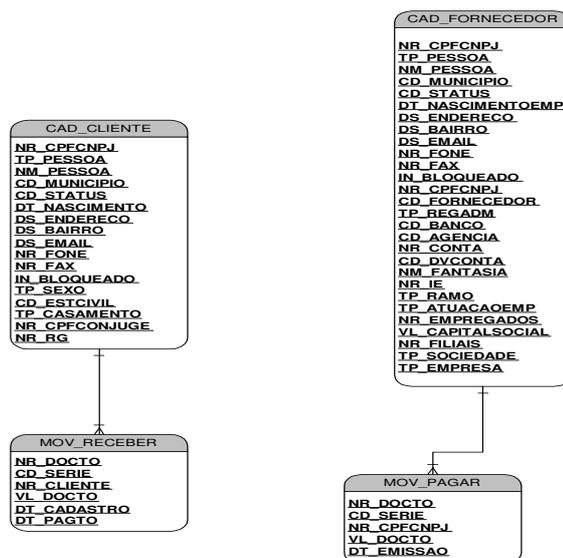


Figura 3. Modelo sem utilização de herança.

#### Etapa 4 - Coletar ou gerar dados

Nessa etapa o banco de dados é gerado no ambiente Oracle 10g, através da geração de dados efetuados através de programação em PL/SQL. A tabela 2 mostra o número de registros de cada classe / tabela utilizado para a realização do teste.

#### Etapa 5 - Analisar dados verificando os diferentes resultados.

Nessa etapa são analisadas as execuções dos experimentos obtendo uma amostra de dados, com o objetivo de fazer afirmações sobre as características dos modelos a serem avaliados. A propriedade de repetibilidade é alcançada através da execução da aplicação mais de uma vez no mesmo ambiente. A tabela 3 mostra a referência de inclusão de

dados "Insert", A tabela 4 mostra consultas "Select" de recuperação de dados executadas, com o número de registros encontrados em cada uma das consultas, e os respectivos tempos médios decorridos Oracle nos dois modelos testados, A tabela 5 mostra resultados referentes a teste de atualização "Update", A Tabela 6 mostra resultados referentes a exclusão de registros "Delete". A tabela 7 mostra resultados referentes a exclusão em cascata "Delete" ambos com tempo utilizando índices e não utilizando.

**Tabela 2.** Números de registros nas entidades

<b>Entidade</b>	<b>Nrº Registros</b>
Pessoa	400,000.00
Pessoa Física	200,000.00
Pessoa Jurídica	200,000.00
Cliente	220,000.00
Fornecedor	180,000.00

**Tabela 4.** Testes de consulta de recuperação de dados

**Tabelas abaixo de 150.000, 00 Registros**

Seleção	Registros	Banco.Tabela	Tempo (ms) Sem Índice	Tempo (ms) Com Índice *
Todos os clientes pessoa Jurídica.	5.000	Heranca.cliente	1.102	1.102
		Relacional.cliente	971	971
Todos os fornecedores pessoa física.	5.000	Heranca.cliente	931	931
		Relacional.cliente	852	852
Todas as contas a receber ordenadas pelo nome do cliente e data de vencimento.	10.000	Heranca.cliente	1.172	1.172
		Relacional.cliente	1.282	1.282
Todas as contas a pagar agrupadas por fornecedor	300	Heranca.cliente	854	854
		Relacional.cliente	792	792
Todas as contas e receber agrupadas por dia e com soma do total diário .	8	Heranca.cliente	30	30
		Relacional.cliente	30	30
Todas as contas e receber agrupadas por cliente unindo com todas as contas a pagar agrupadas por fornecedor, com vencimento maior ou igual a data atual.	8	Heranca.cliente	180	180
		Relacional.cliente	180	180
<b>Tabelas acima de 150.000,00 Registros</b>				
Todos os clientes pessoa Jurídica.	150.000	Heranca.cliente	48.109	41.199
		Relacional.cliente	51.915	40.568
Todos os fornecedores pessoa física.	150.000	Heranca.cliente	42.351	38.536
		Relacional.cliente	43.742	37.033
Todas as contas a receber ordenadas pelo nome do cliente e data de vencimento.	150.000	Heranca.cliente	28.910	25.887
		Relacional.cliente	30.393	25.477
Todas as contas a pagar agrupadas por fornecedor	2	Heranca.cliente	5.117	4.336
		Relacional.cliente	6.830	3.716
Todas as contas e receber agrupadas por dia e com soma do total diário .	8	Heranca.cliente	300	300
		Relacional.cliente	400	400
Todas as contas e receber agrupadas por cliente unindo com todas as contas a pagar agrupadas por fornecedor, com vencimento maior ou igual a data atual.	4	Heranca.cliente	8.712	10.947
		Relacional.cliente	11.726	11.146

Pagar	350,000.00
Receber	350,000.00

**Tabela 3.** Testes de Inserção de registros no banco de dados

<b>Banco.Tabela</b>	<b>Registros</b>	<b>Tempo (ms)</b>
Herança.cliente	10.000	3705
Relacional.cliente	10.000	2904
Herança.cliente	150.000	40659
Relacional.cliente	150.000	52626
Heranca.receber	10.000	1402
Relacional.receber	10.000	1201
Heranca.receber	150.000	20400
Relacional.receber	150.000	34339

**Tabela 5.** Testes de atualização de registros no banco de dados

<b>Seleção</b>	<b>Registros</b>	<b>Banco.Tabela</b>	<b>Tempo (ms) Sem Índice</b>
Alterar o tipo f/j de todos os clientes com id menor ou igual a 75000	75.000	Heranca.cliente	21.831
		Relacional.cliente	33.412

**Tabela 6.** Testes de exclusão de registros no banco de dados

<b>Seleção</b>	<b>Registros</b>	<b>Banco.Tabela</b>	<b>Tempo (ms) Sem Índice</b>
Remover todas as contas a pagar do fornecedor 2	75.000	Heranca.cliente	17.355
		Relacional.cliente	9.654

**Tabela 7.** Testes de exclusão de registros no banco de dados utilizando cascade

<b>Seleção</b>	<b>Registros</b>	<b>Banco.Tabela</b>	<b>Tempo (ms) Sem Índice</b>
Remover o cliente com id = 2	1 cliente e 75.000 Receber	Heranca.cliente	16.813
		Relacional.cliente	7.331

## CONCLUSÃO

Analisando os testes realizados, foi constatado que a utilização de um Banco de Dados Relacional(BDR) do tipo relacional, quando aplicada uma baixa quantidade de dados, tem um desempenho maior em relação ao Banco de Dados Objeto Relacional (BDOR), não havendo variações mesmo utilizando-se Índices. A partir da comparação feita com uma carga maior de dados (150.000), o BDOR saiu-se de forma mais satisfatória, sendo que no mínimo não houve diferença entre os resultados. O mesmo fato ocorreu quando aplicado Índices nas tabelas, alguns teste obtiveram uma grande diferença.

Nossos testes também mostraram que nas operações de update, o BDOR teve um desempenho melhor, em contrapartida, nas operações de delete e insert o BDR obteve melhores resultados.

## REFERÊNCIAS

CANTU, H. C., **Firebird Essencial**. São Paulo : Editora Ciência Moderna, 2005.

CAREY, M.J.; DEWITT, D.J.; NAUGHTON, J.F. The OO7 benchmark. ACM SIGMOD International Conference on Management of Data, USA, 1993.

CATTELL, R.G.G. **An engineering database benchmark**. San Francisco, CA: Morgan Kaufmann, 1991.

DARMONT, J.; SCHNEIDER, M. **Object-oriented database benchmarks**. Advanced topics in database research. Idea Group Publishing, 2002.

FERREIRA, B. O. A. **Dicionário Aurélio da Língua Portuguesa**. São Paulo : Editora Ibeq, 2004.

GRAY, J. **The Benchmark Handbook**. San Francisco, CA: Morgan Kaufmann Publishers, USA, 1993.

GUIMARÃES, C. C. **Fundamentos de banco de dados: modelagem, projeto e linguagem SQL**. Campinas: Editora da Unicamp, 2003.

KALAKECH, A.; KANOUN, K.; CROUZET, Y.; ARLAT, A. Benchmarking the Dependability of Windows NT, 2000 and XP. Intl Conf. on Dependable Systems and Networks. DSN 2004. Italy, 2004.

PENDER, T. **UML a Bíblia**. São Paulo: Editora Campus, 2004.

POESS, M.; FLOYD, C. **New TPC Benchmarks for Decision Support and Web Commerce**. ACM SIGMOD Record. December 2000.

SOUZA, M. BenchMark. Revista eletrônica Imasters, 2003. Disponível em <http://www.imasters.com.br/>. Acessado em: out 2006.

VIEIRA, M.; MADEIRA, H. A Dependability Benchmark for OLTP Application Environments. 29th Intl. Conf. Very Large Data Bases, VLDB2003, Germany, 2003.

ZEE, H. V. D. **Measuring the value of information technology**. Idea Group Publishing, 2002.

ZHU, J. J.; MAURO, J.; PRAMANICK, I. Robustness Benchmarking for Hardware Maintenance Events. In: Conference on Dependable Systems and Networks. DSN2003. USA, 2003.

NOVATHE, Elmarsri. **Sistemas de Bancos de Dados**. São Paulo : Editora Pearson , 2005.

---



Recebido 12 jul. 2007  
Aceito 05 ago. 2007