Estudo Comparativo dos Sistemas Gerenciadores de Bancos de Dados: Oracle, SQL Server e PostgreSQL

Rodrigo de Carvalho Santos, Luís Augusto Mattos Mendes (Orientador)

Departamento de Ciências da Computação – Faculdade de Ciências da Computação e Comunicação Social (FACICS) – Universidade Presidente Antônio Carlos – Campus Magnus – Campolide – MG

rosjdr@gmail.com,luisaugustomendes@yahoo.com.br

Resumo. *As propriedades ACID[1] determinam parâmetros desejáveis em sistemas de banco de dados. No entanto, nem todo SGBD[2] implementa as 4 propriedades, o que pode ser catastrófico em algumas aplicações e imperceptível em outras. Resta então ao administrador do Banco de Dados conhecer as características e decidir qual é o melhor SGBD destinado à sua aplicação. Este artigo apresenta os resultados de um estudo comparativo entre três SGBD´s no que tange as propriedades ACID, desempenho e nível de isolamento das transações.*

Palavras chave: SGBD, Transação, *Benchmark*, Atomicidade, Consistência, Isolamento, Durabilidade, Desempenho.

1 - Introdução

Atualmente os sistemas de informação invadem as empresas tornando-as cada vez mais dependentes da tecnologia e, principalmente, da informática. Partindo desse quesito os sistemas de informação são constituídos por um software ou programa que gerencia todas as atividades para o qual ele foi desenvolvido e um banco de dados para armazenar os dados manipulados pelo sistema.

Antes, arquivos binários eram utilizados para armazenar informações de uma empresa. Conforme esse volume de informações cresce, torna-se necessário encontrar novas soluções que dão melhor estabilidade e maior segurança às informações armazenadas. Essas e outras características tornaram popular o uso dos chamados SGBD´s (Sistema Gerenciador de Banco de Dados)[1].

Um SGBD é um software de computador capaz de armazenar, organizar e oferecer alguma segurança às cada vez mais valiosas informações de uma empresa [1].

Ao desenvolver um sistema, o administrador do banco de dados precisa decidir qual será o SGBD utilizado para gerenciar os dados da aplicação. Existem várias opções de SGBD´s e estes são classificados por categorias. Para este estudo foram selecionados os SGBD´s Oracle, SQL Server e PostgreSQL, por serem classificados como SGBD´s da mesma categoria, serem SGBD´s para grandes bancos de dados e além disso, serem utilizados em grande fatia de mercado.

O objetivo geral deste artigo é mostrar a compatibilidade dos SGBD´s citados anteriormente no que tange as propriedades ACID (Atomicidade, Consistência,

[1] - Acrônimo utilizado para representar as propriedades Atomicidade, Consistência, Isolamento e Durabilidade.

[2] - Sigla utilizada para designar Sistema Gerenciador de Banco de Dados.

Isolamento e Durabilidade), desempenho e performance medidos através de *benchmark* e níveis de isolamento.

A seção 2 deste artigo descreve o que é um SGBD, os principais conceitos e as principais características desejáveis em um sistema de banco de dados. A seção 2.1 define o conceito de transação em um banco de dados. As propriedades ACID são apresentadas na seção 2.2 e explicadas em detalhes nas seções 2.3, 2.4, 2.5 e 2.6.

A seção 3.1 descreve o SGBD Oracle e suas principais características. Na seção 3.2 é apresentado o SQL Server, um SGBD desenvolvido pela Microsoft e utilizado em aplicações de grande porte. A seção 3.3 apresenta o PostgreSQL, um SGBD de código aberto desenvolvido e aperfeiçoado por vários programadores através da Internet.

Na seção 4 deste artigo, é realizado um estudo de caso com os SGBD's citados, um ambiente de trabalho é definido, as métricas utilizadas são apresentadas levando em conta o *Benchmark* TPC-C, um padrão de comparação entre os três SGBD´s é traçado e os resultados obtidos durante o estudo de caso são apresentados. A seção 5 apresenta as considerações finais sobre o levantamento teórico e resultados obtidos através de *benchmark*[3]. A seção 6 do artigo apresenta as referências bibliográficas.

2 – SGBD´s

Um sistema gerenciador de banco de dados é uma coleção de dados inter-relacionados e uma coleção de programas para acesso a estes dados. Sucintamente, um SGBD deve proporcionar um ambiente eficiente para recuperação e armazenamento das informações do banco de dados [1].

As funcionalidades de um bom SGBD vão muito além das quatro propriedades atomicidade, consistência, isolamento e durabilidade representadas pela sigla ACID[2].

Manter o controle de redundância é tarefa do projetista do banco de dados. Evitar que uma mesma informação seja gravada em diferentes tabelas, ajuda a manter a consistência dos dados e a economizar espaço em disco. No entanto, algumas vezes convém implementar a redundância controlada para agilizar o processamento das consultas ao banco de dados. Este tipo de redundância controlada pode ser especificado durante o projeto de banco de dados e forçado pelo SGBD sempre que um arquivo for atualizado para manter a consistência dos dados [2].

Segurança e um subsistema de autorização são outras características bem vistas em um bom SGBD. O controle de usuários deve ser feito de forma que garanta acesso restrito a determinadas tarefas para alguns usuários ou grupos de usuários. Essas restrições devem ser garantidas automaticamente pelo SGBD. Diferentes usuários podem ter acesso e controle a diferentes bancos de dados e áreas de configuração dentro de um SGBD.

O SGBD deve possuir estruturas de dados especializadas para aumentar a velocidade de pesquisa no disco dos registros desejados. O SGBD deve oferecer suporte a um tipo de arquivos auxiliares chamados *indexes*[4] que são utilizados com o objetivo de acelerar o desempenho de um SGBD ao processar uma consulta [2].

---

[3] - Benchmark – Carga de testes que simula o comportamento do mundo real para testar hardware ou software de computador.

[4] index – arquivo de índices do banco de dados

Um componente do SGBD chamado subsistema de backup e recuperação dos subsistemas é responsável por oferecer facilidades para restauração de falhas de hardware ou de software[2].

O SGBD deve fornecer múltiplas interfaces para os diferentes tipos de usuários que acessam os bancos de dados. Desde usuários casuais que desejam executar uma consulta aos dados à DBA's que definem regras e implementam projetos em um SGBD, cada um deve ter uma interface de comunicação específica com o SGBD.

Em um banco de dados, se encontram desde relacionamentos simples à relacionamentos complexos entre os dados. O SGBD deve ter a capacidade de representar a variedade de relacionamentos complexos entre os dados [2].

Durante o projeto do banco de dados o DBA deve especificar algumas restrições de integridade dos dados [2]. O SGBD deve oferecer mecanismos para forçar as restrições de integridade definidas no projeto.

O SGBD deve permitir inferências e ações usando as regras. Em alguns casos, é necessário realizar restrições ou cálculos para determinar algum valor implícito no banco de dados. Por exemplo, podem existir regras que calculem quando a quantidade de produtos de um estoque está abaixo do valor mínimo. Estas regras podem ser declaradas no SGBD e serem disparadas automaticamente por diferentes aplicações. Desta forma, determinar quais produtos estariam abaixo da quantidade mínima desejada seria uma função executada da mesma forma por várias aplicações diferentes [2].

2.1 - Transações

Ao executar uma ação que envolva leitura e escrita no banco de dados, toda a ação é uma única tarefa do ponto de vista do usuário. Assim, alguns respeitados livros de banco de dados definem transação como uma unidade lógica de execução de programas que acessa e, possivelmente, atualiza vários itens de dados[2].

Todos os dados armazenados no banco de dados devem estar íntegros após a execução de qualquer transação. Para uma melhor compreensão deste conceito pode-se imaginar uma operação simples de venda de um produto e atualização da quantidade de estoque do produto em um banco de dados. Suponha que uma operação A seja responsável por realizar a venda do produto e, que uma operação B seja responsável por atualizar a quantidade de produtos no estoque. Se uma operação C que calcula a quantidade de produtos no estoque for executada após a execução da operação A e antes da operação B, ela irá retornar um valor inconsistente dos dados, já que, a quantidade do estoque ainda não foi atualizada pela operação B. Ou ainda, se por algum fator externo a operação A não for executada a operação B poderá atualizar a quantidade de estoque do produto fazendo com que alguns produtos desapareçam do banco de dados.

Para solucionar este problema as operações A e B devem ser consideradas uma única transação. Ou as operações A e B são executadas corretamente ou nenhuma delas será executada.

Para garantir a integridade dos dados, um SGBD deve manter as 4 propriedades chamadas de propriedades ACID [2].

2.2 - Propriedades ACID

O modelo ACID é um dos mais antigos e mais importantes conceito da teoria de banco de dados. Este modelo é a sigla para as propriedades atomicidade, consistência, isolamento e durabilidade. Estas propriedades asseguram que um banco de dados continuará consistente após a execução de uma transação [2].

- Atomicidade é a propriedade que trata da transação de forma atômica, todas as instruções dentro de uma transação são tratadas de forma indivisível, como um átomo. A atomicidade implementa o "tudo ou nada" em um banco de dados, ou todas as modificações de uma transação são executadas com sucesso, ou nenhuma delas é executada.
- Consistência é a propriedade que assegura que todas as modificações serão efetuadas no banco de dados se forem consistentes, isto é, se os dados de uma transação forem válidos a transação é efetuada, senão toda a transação será cancelada.
- Isolamento é a propriedade responsável por garantir que cada transação no banco de dados seja executada de forma isolada. Caso duas transações A e B sejam efetuadas no banco de dados por usuários diferentes ao mesmo tempo, o isolamento garante que A será executada antes de B ou vice-versa. O isolamento não garante qual será a transação executada primeiro mas garante que uma transação não interfira em outra.
- Durabilidade consiste em que toda transação bem sucedida não seja perdida. Caso uma transação seja executada com sucesso até o fim todas as suas modificações são gravadas permanentemente no banco de dados e não serão perdidas mesmo que haja alguma falha física no equipamento.

É de suma importância que se defina claramente o que são operações no banco de dados e instruções de aplicação. Deve-se compreender como operações em banco de dados apenas as instruções de leitura e escrita. Entenda que um banco de dados é um arquivo armazenado em algum tipo de memória secundária, por exemplo um hd. Para realizar cálculos com as informações gravadas em um banco de dados o processador precisa que estas informações estejam armazenadas em memória principal. Para transferir então os dados da memória secundária para a memória principal, o sistema realiza uma operação de leitura no arquivo de banco de dados gravado no hd. Após realizar os cálculos com os dados que estavam na memória principal o sistema realiza uma operação de escrita para gravar os dados no hd do computador [2].

Para facilitar o entendimento da aplicação de cada uma das propriedades ACID pode-se imaginar um sistema simples de venda e atualização de estoque.

Considere uma transação (T) que deverá fazer a venda de 3 itens de um produto X em um sistema de banco de dados. A transação inicialmente realiza uma consulta à quantidade de produtos disponíveis no estoque através de uma leitura na tabela de produtos. Com a operação de leitura os dados são transferidos do disco para a memória principal do computador, então, a transação subtrai 3 da quantidade de produtos em estoque. Conforme se pode observar na Tabela 1, neste momento foi executada a linha 2 da transação T que é uma instrução de aplicação, portanto, a quantidade em estoque do produto na tabela X permanece inalterada. Na linha 3 da transação T, é realizada uma operação de escrita na tabela X. Conforme se observa na Tabela 1 esta é uma operação de banco de dados e após ser concluída, a quantidade em estoque do produto é alterada

para 97. Em seguida, uma instrução de leitura é realizada na venda Y para transferir os dados gravados em disco da venda para a memória principal. Através de uma instrução de aplicação, descrita na linha 5 da Tabela 1, o sistema adiciona 3 à quantidade de itens do produto na venda Y e só então, executa a gravação dos dados que estavam em memória principal no arquivo de banco de dados, como mostra a operação de escrita na linha 6 da Tabela 1.

Tabela 1: Transações com operações de leitura e escrita

Linha de tempo da Transação	Transação (T)	Operações no banco de dados	Instruções de aplicação	Produto (X) / Qtde em estoque	Venda (Y) / Qtde vendida
1	leitura(x);	X		100	0
2	x:= x – 3;		x	100	0
3	escrita(x);	X		97	0
4	leitura(y);	X		97	0
5	y:= y+3;		x	97	0
6	escrita(y)	X		97	3

2.3 - Atomicidade

A idéia da propriedade de atomicidade é executar todas as operações da transação ou não executar nenhuma delas.

Na transação descrita anteriormente, imagine se houvesse uma falha no sistema após a operação de escrita(X) na linha 3 e não fosse garantida a propriedade de atomicidade. A retirada de 3 itens seria feita no estoque X mas, a quantidade do item em Y não seria incluída. Como resultado, 3 itens teriam desaparecido do estoque, gerando um estado inconsistente dos dados.

A propriedade de atomicidade é tratada em um SGBD por um componente chamado de componente de gerenciamento de recuperação.

2.4 - Consistência

A preservação da consistência é geralmente considerada responsabilidade do programador que codifica os programas de banco de dados, ou do módulo do SGBD que garante as restrições de integridade [2].

Os conceitos de consistência e integridade em banco de dados estão intimamente relacionados. Uma vez que, o SGBD trabalha com um banco de dados em um estado previamente consistente e possui mecanismos para garantir a integridade dos dados, por exemplo a validação de atributos de um determinado tipo, os dados serão gravados de forma correta e isto leva o banco de dados a um novo estado consistente.

Para garantir então a consistência dos dados, o SGBD deve realizar uma prévia validação dos dados e só então inserir os dados em uma tabela. Considere uma tabela de empregados onde um de seus atributos é o campo data de admissão do tipo date. Se tentar cadastrar um empregado com a data de admissão “32/01/2008”, o SGBD irá retornar um erro e os dados não serão gravados na tabela de empregados, visto que se

tentou cadastrar uma data inexistente. Este tipo de restrição é imposta pelo SGBD devido a tipação dos dados. Em outro caso, suponha que a data atual seja 20/11/2008, se cadastrar um empregado com a data de admissão 20/11/2015 o SGBD aceitará o cadastro sem restrições já que a data de admissão cadastrada não fere a tipação dos dados. Mas, não é possível no mundo real cadastrar um empregado em 20/11/2008 com data de admissão em 20/11/2015. Este tipo de restrição é impercepitível ao SGBD mas deve ser tratado pelo programador do banco de dados através de *triggers*[5], *stored procedures*[6] ou outras formas de restrição para manter a consistência dos dados armazenados com a realidade.

2.5 - Isolamento

Em um banco de dados onde muitas transações são executadas, uma transação A não pode interferir no resultado de uma transação B ou vice-versa. Uma solução simples e de fácil implementação para este problema seria executar uma transação após a outra.

Facilmente percebe-se que esta solução comprometeria muito o desempenho das operações do banco de dados [2].

Executar transações de forma concorrente é uma tarefa difícil para o sistema de banco de dados. Proporcionar a execução de transações uma a uma, simplifica muito a implementação do sistema de banco de dados [2].

Por outro lado, alguns fatores tornam de suma importância a implementação de concorrência em um sistema de banco de dados. Sabe-se que o processador e o disco em um computador podem executar tarefas distintas simultaneamente, dessa forma, uma transação A pode utilizar recursos de leitura e escrita no disco, enquanto uma transação B pode realizar cálculos através do processador. Este fato aumenta consideravelmente o desempenho de um sistema de banco de dados [2]. Outro fato relevante é a possibilidade de haver várias transações no sistema sendo executadas de forma simultânea. Executar as transações seqüencialmente pode fazer com que uma transação rápida tenha que esperar que uma transação demorada termine sua execução. Para exemplificar esta idéia, imagine uma transação A que consulta os dados de um cliente e uma transação B que acresce 30% (trinta por cento) ao preço de todos os produtos em estoque. Caso a tabela de estoque possua muitos itens a transação A terá que esperar toda a transação B ser executada, o que geraria uma demora excessiva na resposta da transação A.

Quando o sistema executa várias transações concorrentes a consistência dos dados deve ser preservada [2].

Para controlar a execução das transações concorrentes o sistema de banco de dados dispõe de uma gama de mecanismos chamados de esquemas de controle de concorrência.

Em suma, os esquemas de controle de concorrência verificam as transações concorrentes e ordena-as, gerando escalas de execução. O objetivo destes esquemas é gerar escalas de execução que garantam a consistência dos dados e aproveitem da melhor forma possível os recursos computacionais.

---

[5] Triggers são gatilhos ou trechos de código em linguagem SQL que podem ser executados no SGBD para realizar uma função específica antes, durante ou após a execução de uma transação.

[6] Stored Procedures são um conjunto de instruções que são executadas dentro do banco de dados [3].

A SQL92 define ainda, quatro níveis de isolamento. Cada nível especifica os tipos de erros que não são permitidos. Os níveis mais altos incluem as restrições previstas nos níveis mais baixos[4].

Na Tabela 2 se especifica os quatro diferentes níveis de isolamento, seu nome e os possíveis problemas que podem ocorrer em cada nível.

Tabela 2 – Níveis de Isolamento e problemas tratados

Nível	Isolamento	Dirty Read	Nonrepeatable Read	Phantom Read
0	Read uncommited	Possível	Possível	Possível
1	Read commited	Impossível	Possível	Possível
2	Repeatable read	Impossível	Impossível	Impossível
3	Serializable	Impossível	Impossível	Impossível

O nível de isolamento zero *read uncommited* (leitura não efetivada) previne outras transações de modificar dados alterados por transações que ainda não foram efetivadas. As outras transações são bloqueadas de modificar os dados até que a transação seja efetivada ou cancelada. Porém, este nível de isolamento, permite que outras transações leiam dados que foram modificados por outras transações ainda não efetivadas [4].

O nível de isolamento um *read commited* (leitura efetivada) previne o problema de *dirty read*[7] (leitura suja)[4]. Essas leituras acontecem quando uma transação modifica um registro em uma tabela e uma segunda transação lê estes dados. Se a primeira transação não for efetivada, os dados voltarão ao seu estado original o que torna os dados lidos pela segunda transação inválidos.

No nível dois *repeatable read* (leituras repetitíveis) é tratado o problema de *Nonrepeateable read* (leituras não repetitíveis)[5]. Essas leituras acontecem quando uma transação lê um registro e então uma outra transação modifica aquele registro. Se a segunda transação for efetivada, as próximas leituras da primeira transação geram resultados diferentes do original.

O problema chamado *phantom read* (leitura fantasma) é tratado no nível três *serializable* (serializável)[4]. Os fantasmas ocorrem quando uma transação retorna um conjunto de registros que satisfaçam a uma condição. Então, uma transação modifica os dados de alguns registros. Se a primeira transação repetir a pesquisa com as mesmas condições de busca terá um conjunto diferente de dados.

2.6 – Durabilidade

Ao implementar a durabilidade em um banco de dados o componente de gerenciamento de recuperação também conhecido como esquema de recuperação, pode optar por trabalhar com um esquema chamado de cópias *shadow*[8].

Este esquema pressupõe que somente uma transação é executada por vez e que o banco de dados é um arquivo no disco. Ao atualizar o banco de dados uma cópia do

[7] - Expressão em inglês utilizada para representar o problema de leituras sujas em um banco de dados.
[8] Shadow – Técnica que consiste na criação de uma sombra do arquivo de banco de dados.

arquivo é gerada e as atualizações são feitas nesta nova cópia.Um ponteiro mantido no disco aponta para a cópia corrente do banco de dados. Se a transação for executada com sucesso, o ponteiro irá apontar para a nova cópia do arquivo e depois a cópia antiga é apagada. Caso a transação seja abortada devido a alguma falha, o ponteiro permanecerá na cópia antiga, que não sofreu alterações como pode se observar na Figura 1.

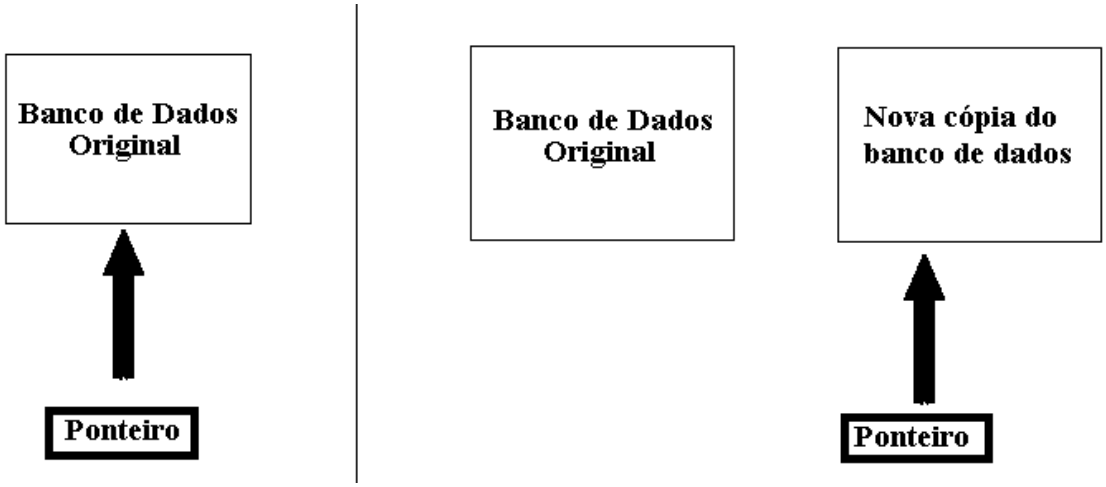

Figura 1 – Técnica Shadow implementada pelo gerenciamento de recuperação

Em casos reais, um banco de dados é um arquivo no disco com várias tabelas que podem ser acessadas por diversos usuários ao mesmo tempo. Como vários usuários realizam diferentes transações concorrentes ao mesmo tempo no banco de dados, falhas podem ocorrer durante a execução de uma ou mais operações. Para corrigir e evitar este problema, o SGBD dispõe de um sistema de recuperação.

O sistema de recuperação pode e deve trabalhar de uma forma mais eficiente realizando a classificação de cada tipo de falha como falha da transação, queda do sistema ou falha de disco. Dois tipos de erros podem levar o sistema a uma falha de transação, erros lógicos que impedem uma transação de continuar sua execução adequada devido a uma entrada inválida ou ainda um dado não encontrado. Uma queda do sistema pode ocorrer por diversos fatores como falta de energia elétrica, falha de hardware no computador e outros que podem levar o banco de dados a parar de funcionar temporariamente. Para recuperar-se de uma possível queda do sistema, o SGBD pode acessar o arquivo de log do sistema quando voltar a funcionar e verificar quais transações foram executadas com sucesso até o final mas, não foram gravadas no disco. Desta forma o SGBD efetiva as transações pendentes tornando-as duráveis [2].

Uma falha de disco pode ocorrer devido a um problema físico no disco onde os dados estão armazenados, este tipo de falha pode ser recuperado através da restauração de um backup executado previamente.[2]

Quando o sistema classifica suas possíveis falhas e sabe como cada uma afeta os dados, o SGBD pode utilizar os chamados algoritmos de recuperação para recuperar os dados a um estado consistente.

3 – Sistemas de mercado analisados

3.1 – Oracle

O SGBD Oracle começou a ser desenvolvido à partir do final da década de 70 e encontra-se atualmente na versão 11g. O Oracle oferece ferramentas que facilitam a vida do programador ao implementar bancos de dados[5].

A Oracle disponibiliza para programadores e DBA´s diferentes versões do SGBD. Cada versão se adapta melhor à realidade da empresa e possui limitações de software, como limite para o tamanho da base de dados, e limitações de hardware como por exemplo a quantidade de processadores ou memória do servidor [6].

Para instalar o SGBD Oracle 10g Express Edition é necessário no mínimo 32MB (trinta e dois megabytes), de espaço no mesmo disco onde o sistema operacional está instalado, para instalar o *jre*[9] e *oracle universal installer*[10]. É recomendado 512MB (quinhentos e doze megabytes) de memória RAM e um processador superior a 1200Mhz(mil e duzentos megahertz). O SGBD usa ainda cerca de 100MB para os arquivos de sistema e é aconselhável instalar em sistemas de arquivos NTFS. Caso se opte por fazer a instalação em sistemas de arquivos FAT32 as exigências são um pouco maiores [6].

Um dos mais recentes e importantes casos de sucesso na implantação da base de dados Oracle é o da editora Gailivro [7]. No mercado há mais de dezoito anos o objetivo da editora era conhecer cada vez mais seus clientes e implantar uma base de dados confiável para estudar o histórico de vendas. Segundo o diretor da empresa Agostinho Lisboa, após a implantação da base de dados Oracle os funcionários da empresa têm um acesso mais rápido às informações que necessitam e os diretores têm uma maior e melhor visão de seus clientes.

3.2 – SQL SERVER

De um modo geral o SQL Server é um SGBD poderoso e confiável, que oferece muitos recursos desejáveis em um bom SGBD. Dentre as principais características do SQL Server destacam-se a alta disponibilidade, desempenho e escalabilidade, segurança, gerenciamento e produtividade [8].

Para funcionar corretamente o SQL Server requer um processador superior a 600Mhz, no mínimo 192 megabytes de memória RAM e acima de 350 megabytes de espaço disponível em disco. O SGBD pode ser instalado em diversas versões do Windows XP, Windows 2000 e Windows 2003 mas, não é compatível com outros sistemas operacionais [8].

Dos casos de sucesso de implantação do SQL Server observa-se com sucesso a atualização do sistema do banco Nossa Caixa no estado de São Paulo. O banco teve um aumento de cento e dezenove mil transações eletrônicas em 2002 para um milhão e setecentos mil transações em 2007 [9]. Para atender a este crescimento, uma operação de atualização do sistema foi iniciada em 2007 e como principais benefícios do novo sistema observa-se o aumento do número de transações por segundo, melhor desempenho na estrutura, facilidade de gestão e aumento na segurança [9].

3.3 – PostgreSql

[9] - Java runtime environment

[10] - Instalador universal da Oracle – utilizado para manipular componentes de software e criar bancos de dados [5]

O PostgreSQL pode ser instalado em um computador pessoal com 8MB de memória RAM e 100MB de espaço em disco, os sistema também oferece suporte a diversos sistemas operacionais como MAC OS X, Windows, Linux, Sun Solaris e outros. Dentre as principais características do SGBD PostgreSQL destacam-se o suporte a triggers, stored procedures, suporte a transações, controle de integridade referencial, não há limite quanto ao número de usuários, oferece controle de transações concorrentes e suporte nativo às quatro propriedades ACID [10].

Um grande sucesso de implantação do PostgreSQL a ser citado é a adoção do SGBD pela BASF (Divisão de produtos para agricultura na América do Norte). Esta empresa vende cerca de US$29 bilhões de dólares por ano e tem cerca de 92.000 (noventa e dois mil) empregados ao redor do mundo. Com o objetivo de se tornar líder de mercado, a empresa desenvolveu um projeto de vendas on line para atender cerca de 25.000 (vinte e cinco mil usuários) simultâneos e pretendia-se um tempo de resposta máximo entre 3 e 5 segundos [11].

Dentre os principais requisitos no projeto da BASF estavam o uso de banco de dados relacional com suporte a transações ACID, stored procedures, triggers e replicação. A empresa avaliou os SGBD´s Oracle e SQL Server entre outros e optou pelo PostgreSQL por sua licença livre, manutenção e suporte [11].

4 - Estudo de Caso

Realiza-se um estudo de caso, em um ambiente de testes padronizado e considerando as características comuns entre SGBD´s, para que deste modo, o programador de banco de dados possa se orientar melhor na escolha do SGBD para sua aplicação.

4.1 – Ambiente de Teste

Os chamados sistemas de processamento de transações (OLTP) são empregados para registrar os dados de uma empresa. Estes sistemas são empregados em áreas de mercado completamente distintas como sistemas bancários ou sistemas de controle de vendas e reservas de passagens aéreas mas, possuem algumas características em comum dentre as quais destacam-se o suporte a grande quantidade de usuários, restrições sobre o tempo de resposta, disponibilidade vinte e quatro horas e sete dias por semana do sistema e um forte padrão de acesso a dados definido por meio de transações [12].

Um dos mais utilizados e respeitados testes para medir desempenho em Sistemas de bancos de dados é o TPC-C. A TPC (*Transaction Processing Performance*) é uma organização mantida pelos principais fabricantes de hardware e desenvolvedores de sistemas de bancos de dados do mundo para fins de teste de *Benchmark* em sistemas de bancos de dados [13].

O ambiente de testes do TPC-C simula um ambiente de pedidos que entra no sistema e sua distribuição. Sucintamente, o teste calcula quantas transações de pedidos o SGBD pode realizar por minuto, simulando o acesso de uma grande quantidade de usuários simultaneamente espalhados ao redor do mundo[13].

Uma companhia fictícia é criada com vários armazéns e pontos de venda associados geograficamente distribuídos. Cada armazém tem dez pontos de venda associados e cada ponto de venda tem três mil clientes. Todos os armazéns mantêm

estoque para os cem mil artigos comercializados pela companhia. Na Figura 2 o ambiente de teste simulado pelo benchmark TPC-C está representado [13].

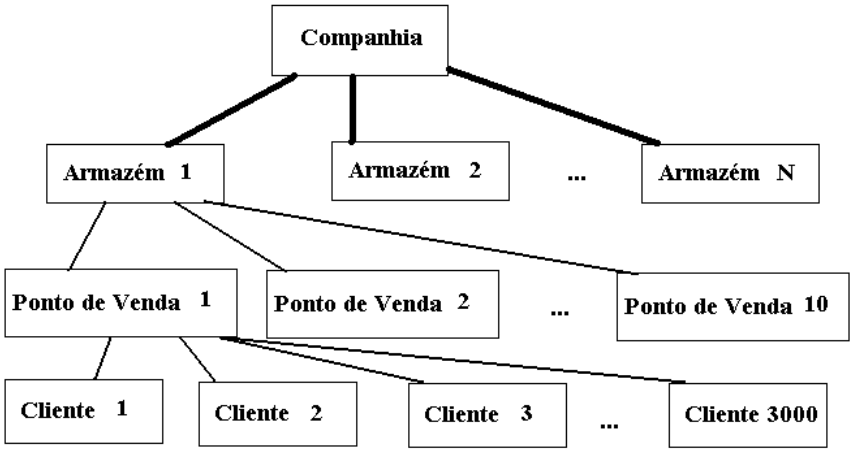

Figura 2 – Ambiente de testes TPC-C

4.2 – Análise de Desempenho

O *Benchmark* TPC-C leva em conta características desejáveis em um sistema de banco de dados como as propriedades ACID além do desempenho alcançado pelos SGBD´s.

Para medir o desempenho de um SGBD em um ambiente OLTP o TPC-C indica o índice tpmc que representa quantitativamente quantas transações são realizadas por minuto no banco de dados. Durante os testes cinco transações de tipos e complexidades diferentes são executadas em nove tabelas distintas que constituem o banco de dados durante o tempo de 20 minutos [13].

Para cada transação executada também é calculado o tempo de resposta e ao final do teste o TPC-C indica o tempo médio de resposta para o SGBD avaliado.

Ao avaliar as propriedades ACID o *Benchmark* TPC-C leva em conta os mais importantes requisitos de cada uma delas. No manual técnico do TPC-C as características desejáveis para cada propriedade, a carga de trabalho que compõe as transações de *Benchmark* e todas as fórmulas utilizadas para calcular tempos de respostas e outros podem ser encontrados [13].

4.3 – Estudo Comparativo

Com base nas propriedades ACID, níveis de isolamento e desempenho, realizou-se um estudo comparativo entre os SGBD´s Oracle, SQL Server e PostgreSQL.

A Tabela 3 apresenta as propriedades ACID implementadas em cada SGBD.

Tabela 3 – Comparativo de Propriedades ACID

SGBD	Atomicidade	Consistência	Isolamento	Durabilidade
Oracle	x	x	x	X
SQL Server	x	x	x	X

PostgreSQL	x	x	x	X

A atomicidade é implementada nos três SGBD´s através dos comandos BEGIN, COMMIT e ROLLBACK. Estes comandos definem blocos de transação, ou seja, todos os comandos colocados após um comando BEGIN podem ser efetivados com o comando COMMIT ou cancelados com um comando ROLLBACK, tratando todos os comandos como uma única operação. Caso um comando BEGIN seja executado em qualquer um dos SGBD´s e não seja encontrado um comando COMMIT ou ROLLBACK por decorrência de uma falha do sistema, o sistema de recuperação do SGBD se encarrega de restaurar o banco de dados ao estado que estava antes do início da transação.

A consistência dos dados é mantida nos três SGBD´s através da tipação dos dados e restrições de integridade impostas pelo programador. Quando se fala de tipação dos dados significa que nos três SGBD´s encontram-se tipos de dados bem definidos. Como exemplo pode-se imaginar que em nenhum dos três SGBD´s os usuários conseguiriam armazenar uma cadeia de caracteres em um campo do tipo inteiro. As restrições impostas pelo programador do banco de dados são restrições de integridade referencial dos dados, suportadas pelos três SGBD´s ou validação dos dados para manter a consistência com o mundo real, também suportado pelos três SGBD´s. A consistência dos dados ainda pode ser implementada através de *triggers* ou *stored procedures* ou ainda através do comando CHECK que pode ser utilizado para verificar valores válidos para cada atributo nos três SGBD´s.

O isolamento das transações é implementado e podem ser definidos níveis de isolamento diferentes para cada transação nos três SGBD´s. Assim, os três SGBD´s definem níveis de isolamento diferentes e cabe ao programador do banco de dados definir qual será o melhor nível de isolamento para cada transação.

Os três SGBD´s possuem ainda um sistema de recuperação que possibilita a implementação da durabilidade dos dados. Uma vez que uma transação é efetivada, os dados são mantidos. Se uma transação não é efetivada, todas as suas alterações serão perdidas.

A Tabela 4 mostra os resultados obtidos em uma análise de desempenho levando em consideração a quantidade de usuários e o índice tpmc obtido pelo *Benchmark* TPC-C.

Os três SGBD´s foram instalados no sistema operacional Windows XP da Microsoft, e o programa *Benchmark* Factory 5.7.1 foi utilizado para simular a carga de trabalho TPC-C. A conexão com os SGBD´s foi feita via driver ODBC fornecido pelos fabricantes em suas homepages.

Tabela 4 – Resultado obtido através do *Benchmark* Factory

	1 usuário	8 usuários	10 usuários	100 usuários
Oracle	1,328	12,590	14,815	141,236
SQL Server	1,490	12,614	14,908	141,537
PostgreSQL	1,479	11,952	14,735	140,335

Os resultados dos testes realizados pelo programa *Benchmark* factory mostram o desempenho de cada SGBD quando instalados em um computador e com toda a carga de trabalho TPC-C simulada através de software. Realizou-se o teste em 4 situações diferentes com números pequenos de usuários e o SQL Server obteve melhor resultado em todos os casos, em segundo lugar aparece o SGBD Oracle e em terceiro o PostgreSQL. Observa-se ainda que para um usuário o PostgreSQL foi mais eficiente que o Oracle mas menos eficientes nas outras situações.

A organização TPC divulgou em 15/09/2008 os resultados mais recentes obtidos através do *Benchmark* TPC-C realizados no ambiente padrão criado pela organização. Todos os testes de *Benchmark* realizados pela TPC são auditados e acompanhados pelas principais empresas desenvolvedoras de SGBD do mundo e outros membros da organização, afim de legitimar os resultados obtidos [13]. A Tabela 5 mostra uma parte dos resultados obtidos nos testes dos SGBD´s Oracle, PostgreSQL e SQL Server instalados em um servidor UniSys modelo ES7000 [13].

Tabela 5 – Resultados oficiais TPC-C

	Servidor	tpmC/Windows	tpmC/Linux
Oracle	UniSys ES7000	291413	322805
SQL Server	UniSys ES7000	309036.53	-
PostgreSQL	UniSys ES7000	298534	325685

No ambiente de testes do TPC-C o SQL Server foi superior quando comparado com o Oracle e o PostgreSQL rodando em um servidor Windows. Porém, em servidores com sistema operacional Linux não é possível instalar o SGBD SQL Server. O PostgreSQL superou o Oracle em ambos os casos.

Nos SGBD´s Oracle e PostgreSQL o nível de isolamento pode ser definido pelo programador de banco de dados através do comando SET TRANSACTION antes de realizar a transação. No SQL Server o comando é SET TRANSACTION ISOLATION LEVEL e também deve ser definido antes de uma transação.

No Oracle e no PostgreSQL são possíveis duas opções para os comandos SET TRANSACTION conforme observa-se na Tabela 6. Isto implica que em ambos os casos é possível dois níveis de isolamento diferentes, *read commited* e *serializable* [14].

Na Tabela 6 é apresentado um estudo comparativo entre os SGBD´s no que tange suas opções de isolamento.

Tabela 6 – Comparação de Níveis de Isolamento

Isolamento	Oracle	SQL Server	PostgreSQL
Read uncommited		**x**	
Read commited	**x**	**x**	**X**
Repeatable read		**x**	
Serializable	**x**	**x**	**X**

Snapshot		**x**	

Quando compara-se os níveis de isolamento o SQL Server é o mais completo e que oferece mais opções de configuração ao programador. As opções *read uncommited*, *read commited*, *repeatable read* e *serializable* estão disponíveis para serem utilizadas junto ao comando SET TRANSACTION ISOLATION LEVEL. Além destas, o SQL Server tem a opção *snapshot* que especifica que a transação pode reconhecer apenas modificações que estavam efetivadas antes do início da transação e modificações efetivadas por outras transações após o início da transação atual não são vistas [15].

5 – Considerações Finais

Antes de decidir qual SGBD é o melhor, é necessário que se tenha definido bem qual será o ambiente onde ele será aplicado. Observa-se que em casos onde o número de usuários é pequeno ou o sistema operacional utilizado é o Windows, se obtém uma melhor performance quando a opção escolhida é o SQL Server.

Ainda considerando apenas o desempenho o PostgreSQL apresenta melhores resultados que o Oracle quando se opta pelo sistema operacional Linux.

Entretanto, deve-se considerar outras características importantes em um SGBD como seu comportamento em relação às propriedades ACID e níveis de isolamento que podem ser implementados.

Simular um ambiente de software para teste e avaliação de todas as propriedades possíveis de um SGBD é uma tarefa bastante complicada. Em situações reais, ocorrem possíveis problemas como queda de energia que são de difícil implementação.

Conclui-se que não há um SGBD que atenda a todos os projetos de forma melhor que os demais. Deve-se estudar o desempenho e as necessidades de cada projeto em relação às propriedades ACID para tomar a decisão de qual SGBD utilizar.

Para trabalhos futuros, sugere-se temas como: implementação de um software para simulação e teste das propriedades ACID, implementação de um software para demonstração dos diferentes efeitos em diferentes níveis de isolamento ou análise de log dos SGBD´s por tempo determinado.

6 - Bibliografia

[1] SILBERSCHATZ, Abraham; KORTH, Henry F. e SUDARSHAN, S. **Sistema de Banco de Dados**. São Paulo. Makron Books, 1999.

[2] ELMASRI, Ramez. **Sistemas de banco de dados / Ramez Elmasri e Shamkant B. Navathe**. São Paulo. Pearson Addison Wesley, 2005.

[3] PICHILIANI, Mauro. **Criação e Uso de Stored Procedures**. Disponível em: <http://imasters.uol.com.br/artigo/223/sql_server/criacao_e_uso_de_stored_procedures/>. Acesso em 29/10/2008

[4] SYBASE. **SQL Server Reference Manual**. Disponível em: <http://manuals.sybase.com/onlinebooks/group-asarc/srg1100e/sqlref/@ebt-link;pt=16395;lang=pt?target=%25N%15_70685_START_RESTART_N%25>. Acesso em 07/11/2008

[5] ORACLE, Brasil. **A História do Oracle: Inovação, Liderança e Resultados**. Disponível em: <http://www.oracle.com/global/br/corporate/story.html>. Acesso em 17/10/2008

[6] PRASS, Fábio; PRASS, Fernando. **Oracle e Delphi uma visão geral do melhor dos bancos de dados**. Disponível em: <http://www.devmedia.com.br/articles/viewcomp.asp?comp=1311&hl=*oracle*>. Acesso em 17/10/2008

[7] ORACLE, PME. **Gailivro implementa Oracle como base de um programa de gestão integrada de informação**. Disponível em: <http://www.oracle.com/global/pt/pmes/references/case_study_gailivro_30_agosto_06.pdf>. Acesso em 20/11/2008.

[8] MICROSOFT CORPORATION. **SQL Server 2005**. Disponível em: <http://www.microsoft.com/brasil/servidores/sql/2005/prodinfo/overview/default.mspx>. Acesso em 12/11/2008.

[9] MICROSOFT BRASIL. **Casos de Sucesso Nossa Caixa – Nossa Caixa amplia número de transações a partir do SQL Server 2005**. Disponível em: <http://www.microsoft.com/brasil/Casos/interna.aspx?id=588>. Acesso em 20/11/2008.

[10] The Postgre Global Development Group. **Documentação do PostgreSQL 8.0.0**. Rio de Janeiro, 2005.

[11] MILANI, André. PostgreSQL – Guia do Programador. Disponível em: <http://novatec.com.br/livros/postgre/capitulo9788575221570.pdf>. Acesso em 10/11/2008

[12] MENDES, Marcelo; MACIEL, Paulo. **Análise de Desempenho de Sistemas OLTP utilizando o *Benchmark* TPC-C**. Disponível em: <http://www.cin.ufpe.br/~tg/2006-1/mrnm.pdf>. Acesso em 16/11/2008.

[13] TRANSACTION PROCESSING PERFORMANCE COUNCIL (TCP). **Standard Specification Revision 5.9**. Junho, 2007.

[14] ORACLE. **Oracle Database SQL Reference**. Disponível em: <http://download-east.oracle.com/docs/cd/B14117_01/server.101/b10759/statements_10005.htm> Acesso em 21/11/2008.

[15] MICROSOFT BRASIL. **Set Transaction Isolation Level (Transact-SQL)**. Disponível em: <http://msdn.microsoft.com/pt-br/library/ms173763.aspx>. Acesso em 22/11/2008.