

CENTRO PAULA SOUZA 4 ANOS
COMPETÊNCIA EM EDUCAÇÃO PÚBLICA PROFISSIONAL

 **GOVERNO DE
SÃO PAULO**

**FACULDADE DE TECNOLOGIA DE AMERICANA
PROCESSAMENTO DE DADOS**

Danuza Santana Rocha

**Alta Disponibilidade em SQL Server
Replicação de dados**

Americana - SP

Maio – 2010

Danuza Santana Rocha

RA: 071303

Alta Disponibilidade em SQL Server
Replicação de dados

Trabalho de conclusão de curso apresentada à banca examinadora da Faculdade de Tecnologia de Americana, como parte dos requisitos necessários para obtenção do título de Tecnólogo em Processamento de Dados, sob orientação do Prof. José Alberto F. Rodrigues Filho.

FATEC - Americana

Americana - Maio / 2010

Dedicatória

Dedico este trabalho a minha família e amigos fofos que me colocaram pra cima e ajudaram (ou tentaram) manter meu bom humor.

Agradecimentos

Agradeço a Deus, que me deu a ferramenta, e que afinal, me fez assim. Minha família que me educou e me mostrou o certo. Aos meus amigos e pessoas que passaram pela minha vida e adicionaram conhecimento e experiência e enfim, aos meus professores que me incentivaram e tiveram muita paciência.

Epígrafe

“Ama-se mais o que se conquista com esforço”
Benjamin Disraeli

Resumo

Este trabalho tem como objetivo mostrar a tecnologia de replicação de dados para SGBD Microsoft SQL Server como uma solução de Alta Disponibilidade voltada para banco de dados. No início são descritos alguns conceitos sobre banco de dados, e sua importância, conceitos sobre SGBD, sua história, linguagem padrão (SQL) e modelo de dados relacional. A Alta Disponibilidade será vista como um assunto importante e necessário para as empresas nos dias de hoje. Algumas soluções de Alta Disponibilidade em SQL Server serão descritas superficialmente, mas a replicação de dados será descrita detalhadamente, com seus componentes, tipos e cenários de aplicação. E finalmente, no final deste trabalho um exemplo real de aplicação.

Palavras-chave: Replicação de dados, alta disponibilidade, duplicação de dados, SQL Server.

Abstract

This paper aims to show the technology of data replication for Microsoft SQL Server DBMS as a High Availability solution-oriented database. At first we describe some concepts of the database, and its importance, DBMS concepts, its history, standard language (SQL) and relational data model. High Availability will be seen as an important and necessary for business today. Some solutions for High Availability in SQL Server will be described superficially, but the data replication will be described in more detail, with its components, types and application scenarios. And finally, at the end of this work a real example of application.

Keywords: data replication, high availability, duplication of data, SQL Server.

Lista de ilustrações

Figura 1 – Modelo relacional	14
Figura 2 – SQL Server 2008.....	17
Figura 3 – Cluster de Failover	21
Figura 4 – Espelhamento de dados.....	23
Figura 5 – Envio de logs – Cenário simples	24
Figura 6 – Bancos de dados compartilhados evolutivos	25
Figura 7 – Visão geral dos componentes de uma replicação	27
Figura 8 – Principais componentes de replicação transacional.....	32
Figura 9 – Replicação voltada a Data Warehouse e relatórios	38
Figura 10 – Integrando dados de vários sites.....	39
Figura 11 – Integrando dados para outros bancos.....	40
Figura 12 – Integrando dados do Oracle.....	40
Figura 13 – Descarregando processamento em lote.....	41
Figura 14 – Integrando dados de diversos locais	42
Figura 15 – Integrando dados com usuários móveis - Laptops.....	43
Figura 16 – Integrando dados com usuários móveis - PDAs	43
Figura 17 – Aplicativos POS – Caixa eletrônico	44
Figura 18 – Conexões via linhas dedicadas entre as lojas.....	46
Figura 19 – Topologia replicação	47
Figura 20 – Visualização detalhada da replicação entre 2 lojas.....	47

Lista de abreviaturas e siglas

ANSI	American National Standards institute
CRM	Gerenciamento de Relacionamento com o Cliente
DBA	Administrador de Banco de Dados
DDL	Data Definition Language
DML	Data Manipulation Language
EILD	Exploração Industrial de Linha Dedicada
FFA	Automação de Força do Campo
ISO	International Organization for Standardization
OLAP	On-line Analytical Processing
OLTP	Online Transaction Processing
PDA	Personal Digital Assistant
POS	Ponto de Venda ao Consumidor
ROLAP	Relational On Line Analytical Processing
SAN	Storage Area Network
SFA	Automação de Força de Vendas
SGBD	Sistema Gerenciador de Banco de Dados
SQL	Structured Query Language
TI	Tecnologia da Informação
VB	Visual Basic

Sumário

1. Introdução	10
2. Banco de dados	11
2.1. SGBD	12
2.2. Modelo Relacional	14
2.3. SQL	15
2.4. Microsoft SQL Server	16
3. Alta Disponibilidade	17
3.1. Disponibilidade	18
3.2. Soluções de Alta Disponibilidade.....	19
4. Replicação de Dados - Conceitos	26
4.1. Tipos de replicação.....	29
4.1.1. Replicação Transacional	30
4.1.2. Replicação de Mesclagem	34
4.1.3. Replicação de Instantâneo	36
4.2. Ambiente	37
4.3. Exemplo de aplicação.....	45
5. Conclusão	49
Referências	50

1. Introdução

À medida que o tempo passa e a área de TI se torna mais dinâmica, ganha mais importância dentro das corporações e passa a ser indispensável. Infelizmente muitas empresas ainda não têm esta consciência e visualizam seu setor de TI como um bombeiro, simplesmente “apagando incêndios”, resolvendo problemas quando acontecem, sem planejamento ou monitoração do ambiente.

A informação hoje é um dos itens mais importantes dentro de qualquer corporação e isso gera a necessidade de segurança, tanto das informações contidas no banco de dados como em sua integridade e disponibilidade, o que aumenta o investimento na área de TI e cria tecnologias voltadas a estas necessidades.

Segurança das informações está associada não só a fraudes, perdas ou roubos de informações, mas também na integridade, agilidade e acessibilidade, o que faz com que as empresas escolham um SGBD que atenda melhor aos requisitos. Disponibilidade não significa apenas manter sistemas funcionando, o conceito vai além, pois envolve manutenção de aplicações e dados. Um servidor disponível, porém com aplicações indisponíveis, não tem o menor valor para o usuário, pois as informações na verdade estarão indisponíveis. O conceito se resume a Parada = Prejuízo.

Este trabalho foi desenvolvido com o objetivo de mostrar que a tecnologia de replicação de dados para SGBD Microsoft SQL Server é uma boa opção de Alta Disponibilidade. E no final deste trabalho será mostrado um exemplo real de aplicação.

2. Banco de dados

Segundo Korth (2006), banco de dados “é uma coleção de dados inter-relacionados, que contém informações relevantes para uma empresa”.

Os dados devem ser inter-relacionados, não existem vantagens em ter dados isolados. Quando os dados estão inter-relacionados e são então manipulados, traduzem-se facilmente em informações relevantes, se isolados causam grande custo de tempo e mecanismos de busca para esta tradução em informação.

Seguindo este conceito, pode ser citado como exemplo de banco de dados uma agenda telefônica, pois relaciona os números de telefones de um grupo de pessoas, até mesmo um caderno de anotações, que relaciona informações sobre as aulas assistidas num determinado período.

Em ambiente corporativo os bancos de dados contêm, por exemplo:

- carteira de clientes;
- carteira de fornecedores;
- estoque de produtos;
- documentos de compras e vendas;
- lançamentos contábeis;
- valores destinados ao fisco;
- movimentações bancárias;

Ou seja, informações importantíssimas que são essenciais para as empresas, toda sua história pode estar contida em seu banco de dados.

Uma importante característica dos bancos de dados é a possibilidade de compartilhar informações, tanto entre usuários dentro de um mesmo espaço físico até mesmo entre usuários separados por grandes distâncias, por exemplo, uma rede de hiper-mercados com 50 lojas dentro do estado, um cliente pode se cadastrar na loja de sua cidade, passar por toda a rotina de aprovação de crédito e então poder comprar em qualquer loja da rede quando estiver viajando sem precisar repetir todo esse processo burocrático.

Quando a pergunta “qual banco de dados você utiliza?” é feita, a maior parte das respostas vem com algum desses nomes: SQL Server, Oracle, DB2, Firebird, etc. Mas na verdade esses nomes estão relacionados à ferramenta que o gerencia, o sistema gerenciador de banco de dados (SGBD).

O conceito de banco de dados deixa claro de que é apenas um conjunto de dados, como um depósito, toda a interface de comunicação com o usuário é feita através dos SGBDs, assim esses dados podem se converter em informações úteis.

2.1. SGBD

Segundo Korth (2006), SGBD “é a coleção de dados inter-relacionados e um conjunto de programas para acessar estes dados”.

SGBD é o conjunto de programas que permite manipular o banco de dados e faz a interface entre o banco de dados e o usuário. Seu principal objetivo é fornecer de maneira eficiente a recuperação de informações do banco de dados Korth(2006).

Os bancos de dados são parte essencial de uma empresa hoje, seu uso cresceu a partir da década de 70 quando o modelo relacional foi estabelecido e simplificou a visão dos dados, assim os SGBDs surgiram, como evolução dos sistemas antigos de gerenciamento de arquivos, onde os dados eram mantidos em arquivos de sistema, a estrutura era definida pela aplicação e os relacionamentos em forma de arquivos, para a recuperação de dados era necessário varrer o arquivo em busca das informações necessárias Rezende (acesso 2010). Com os SGBDs, o armazenamento dos dados é feito segundo a metodologia de metadados¹, não só os dados estão contidos no banco, mas toda sua descrição, relacionamentos, regras de integridade e de acesso.

A recuperação dos dados é feita utilizando algoritmos complexos, que resultam rapidamente as informações.

¹ metadados : Dados que descrevem dados. Uma abstração de dados.

A integridade referencial definida também é gerenciada pelo SGBD, sem precisar do programa aplicativo, contendo instruções que permitam tanto a positivação das requisições quanto o seu cancelamento caso a integridade seja violada.

Outras funções interessantes são o controle de acessos simultâneos, evitando os erros de deadlock² tão comuns dos antigos gerenciadores de arquivos e o controle de acesso a usuários, permitindo definir usuários e quais tipos de acessos podem ter no banco NOVELLI (acesso 2010).

Resumindo, as vantagens de um SGBD:

- Rapidez e simplicidade no acesso e na manipulação dos dados, diminuindo o esforço do usuário e do programador;
- Redução da redundância e inconsistência, resultando em confiança nas informações;
- Integridade das informações;
- Compartilhamento dos dados;
- Redução de problemas de acessos concorrentes;
- Restrição e concessão de acesso aos usuários, resultando em segurança;
- Controle da distribuição física dos dados;
- Diminuição do isolamento dos dados;

Na década de 90, quando houve a explosão do uso da internet, é que uso dos SGBDs foi intensificado, com a grande quantidade de lojas, bancos e livrarias virtuais, emails, etc. Essas empresas têm uma grande necessidade em compartilhar informações de maneira rápida, ampla e segura. E são essas necessidades que fazem com que a tecnologia voltada ao banco de dados desenvolva grandes fornecedores de SGBDs, ou faça parte de setor de grande força dentro de empresas de software mais diversificadas Korth (2006).

Mas para que o sistema seja realmente funcional, precisa descrever os dados de forma que usuário entenda. A maioria dos bancos de dados são baseados no modelo de dados relacional, que oferece de forma simples de apresentação dos dados.

² deadlock : Situação em que ocorre um impasse e dois ou mais processos ficam impedidos de continuar suas execuções, ou seja, ficam bloqueados.

2.2. Modelo Relacional

É um modelo de dados, um conceito abstrato que define maneiras de armazenar, manipular e recuperar dados de maneira estruturada, baseada na teoria matemática das relações, o que faz com que a visão dos dados fique extremamente simples Korth (2006).

Foi desenvolvido na década de 70 por Edgar Frank Codd, que desligou a estrutura lógica do método de armazenamento físico de dados Rezende (acesso 2010).

Neste modelo existe um conjunto de tabelas, cada linha da tabela é um elemento da relação, uma entidade. Cada coluna corresponde a atributos da relação, tanto o armazenamento das informações quanto o resultado de buscas no banco é feito baseando-se em tabelas.

Exemplo:

Tabela / Relação / Entidade: Amigos

Tabela / Relação / Entidade: Telefones

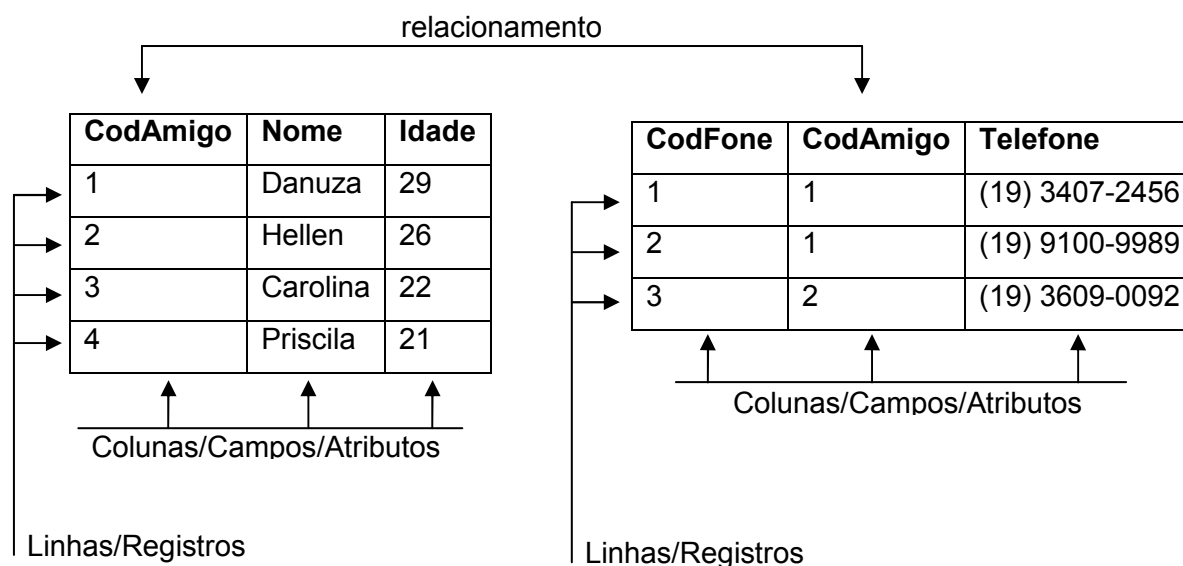


Figura 1

Tanto a tabela Amigos como Telefones tem suas colunas e linhas, mas note que a coluna CodAmigo existe em ambas, é através dela que o relacionamento

entre as linhas das tabelas é feito, assim o sistema de gerenciamento ao receber o comando para mostrar o telefone da Danuza, mostrará:

Telefone
(19) 3407-2456
(19) 9100-9989

Para estabelecer essas relações de maneira organizada, para que seja interpretado corretamente pelo SGBD, foi criada uma normalização, visando principalmente à qualidade das informações e o bom desempenho de acesso.

A forma simples de visualizar os dados, foi o motivo principal para o sucesso do banco de dados relacional, juntamente com a uniformização e padronização da linguagem de manipulação dos dados, o Structured Query Language (SQL).

2.3. SQL

É uma linguagem de pesquisa declarativa, criada nos anos 70, nos laboratórios da IBM. Devido à simplicidade de uso e a rapidez em que os programadores a dominavam, foi normalizada pela American National Standards Institute (ANSI) e International Organization for Standardization (ISO) e tornou-se a linguagem padrão para banco de dados relacionais na segunda metade da década de 80 Korth (2006).

A natureza declarativa da linguagem SQL era uma novidade para a época. Os sistemas de banco de dados existentes utilizavam o modelo de dados hierárquico ou em rede, as linguagens de acesso a dados eram procedimentais, isto fazia necessário que o programador navegasse nas estruturas de dados para obter as informações de interesse, já no SQL, se especifica apenas a forma do resultado, não importando como este será obtido, o SGBD é o responsável por escolher adequadamente os procedimentos a serem executados, de forma que os resultados sejam obtidos com a maior eficiência possível.

Mesmo que o SQL se defina como uma linguagem de consulta, não é, ela pode também definir estrutura, modificar, excluir dados, fazer restrições de

segurança, etc. Isto porque o SQL possui algumas outras linguagens incorporadas, as principais são:

- Linguagem de definição de dados ou Data Definition Language (DDL): fornece os comandos para definição de dados, ou seja, para criar, modificar e excluir estruturas, objetos e direitos de acesso do banco de dados.
- Linguagem de manipulação de dados ou Data Manipulation Language (DML): fornece comandos para manipulação dos dados, ou seja, para incluir, excluir e modificar os dados.

Apesar de ter sido padronizada, alguns fabricantes de SGBDs criaram extensões para o SQL, que permitem escrever a lógica empresarial usando linguagens de programação como Java, C++, .C#, Visual Basic (VB), J++, COBOL, etc, mas apesar dessas extensões, pode-se migrar de plataforma sem mudanças estruturais principais da linguagem Korth(2006).

2.4. Microsoft SQL Server

O SQL Server é um sistema gerenciador de banco de dados relacional criado pela Microsoft, usado desde Pocket PCs até servidores de grande porte, é robusto e por isso escolhido por empresas dos mais diversos portes.

Foi desenvolvido no final da década de 80 pela Sybase, originalmente usado em sistemas UNIX e depois então adaptado para Windows NT pela Microsoft. É uma plataforma de dados confiável, produtiva e inteligente que permite a execução de aplicações críticas, mais exigentes. Korth(2006).

Oferece várias ferramentas gráficas de gerenciamento que ajudam no trabalho dos Administradores de Banco de Dados (DBA) em criar objetos de banco de dados e restrições de acesso, ou seja, projetar o banco de dados.

Permite a utilização da extensão do SQL chamada Transact-SQL ou linguagens de programação .NET como C#, J++, VB ou COBOL. Korth(2006).



Figura 2 – SQL Server 2008

O SQL Server, em sua última versão, SQL Server 2008, teve algumas implementações de soluções de Alta Disponibilidade que junto com outras opções já existentes serão vistas mais a frente:

- Cluster de failover
- Espelhamento de banco de dados
- Envio de logs
- Bancos de dados compartilhados evolutivos
- Replicação ou Duplicação de dados

3. Alta Disponibilidade

Alta Disponibilidade ou High Availability no inglês, é um assunto indispensável e que todos os profissionais da área de banco de dados têm que compreender atualmente. Nichter (acesso 2010).

Devido à globalização, a aceleração do crescimento dos mercados digitais, o ditado “tempo é dinheiro” passou a ser uma verdade absoluta para as corporações. O acesso e a velocidade do acesso à informação tem sido cada dia mais exigida, podemos citar em qualquer setor, exemplos de aplicações que não podem parar, e também citar os transtornos e prejuízos ocorridos caso ocorra parada, como em lojas virtuais, hospitais, aeroportos, bancos, etc.

Evitar paradas não programadas e manter os sistemas operando, mesmo em casos adversos, aumentam a competitividade e evitam perdas financeiras e até mesmo perda de vidas caso o setor seja de saúde.

Mas antes de tratar da Alta Disponibilidade, é necessário saber o que é disponibilidade.

3.1. Disponibilidade

Segundo alguns autores “Disponibilidade é capacidade de um serviço ou componente executar sua função em um período determinado de tempo”.

Devido à exigência de exatidão na área de TI, o conceito foi traduzido em uma fórmula que expressa a porcentagem de disponibilidade que esses serviços ou componentes podem ter:

$$\text{Disponibilidade} = (TT - P) / TT \quad \times 100$$

Onde TT é o total em unidade de tempo e P é a parada em unidade de tempo. Brandão (acesso 2010).

Exemplo 1:

No mês de abril (720 horas) o servidor de banco de dados de uma empresa teve 1 hora de parada para atualização dos sistemas:

$$\text{Disponibilidade} = (720 - 1) / 720 \quad \times 100 = 99,8611 \%$$

Esta porcentagem parece boa, mas o objetivo é chegar mais próximo ainda de 100%, que seria a perfeição.

Exemplo 2:

No trimestre que compreende os meses de abril (720 horas), maio (744 horas) e junho (720 horas) o servidor de banco de dados de uma empresa teve 1 hora de parada para atualização dos sistemas:

$$\text{Disponibilidade} = (2184 - 1) / 2184 \quad \times 100 = 99,9542 \%$$

Comparando o exemplo 1 com o exemplo 2, vemos que o tempo analisado (TT) teve influência na disponibilidade, por isso deve-se utilizar a fórmula baseando-se nas necessidades de cada empresa.

Analisando a fundo a fórmula e os exemplos, concluímos que para aumentar a disponibilidade de um sistema ou componente, devemos diminuir ao máximo o tempo de parada (P), e com este objetivo veremos algumas soluções de Alta Disponibilidade para banco de dados SQL Server.

3.2. Soluções de Alta Disponibilidade

Segundo NICTER (acesso 2010), “Alta Disponibilidade refere-se à prontidão e acessibilidade de um sistema”.

Várias empresas de TI estão se especializando, e prestando consultoria em Alta Disponibilidade, e para chegar numa solução eficiente, aumentar a acessibilidade de um sistema, alguns passos são necessários. Kioskea (acesso 2010):

- Avaliar a estrutura e necessidades específicas;
- Avaliar a arquitetura de disponibilidade já existente e processos de gerenciamento;
- Analisar possíveis incidentes que possam causar interrupções;
- Identificar áreas de risco na arquitetura e os níveis desses riscos, para poder minimizá-los;
- Definir metas de disponibilidade;
- Definir o orçamento disponível.

Dentro dos possíveis incidentes estão os erros, dentro da avaliação deve-se:

- Prevenir erros: evitar os erros antecipando-os;
- Tolerar erros: definir uma redundância;
- Eliminar erros: reduzir os erros com ações corretivas.

Algumas classificações de erros, levando em consideração também atos criminosos e intencionais:

- Físicas
 - Desastres naturais (inundações, terremotos, incêndio);
 - Ambiente (umidade, temperatura);
 - Defeito/depreciação de material (hardware, cabeamento de rede, etc);
 - Quedas de eletricidade;
- Humanas
 - Falha de projeto (software, rede, etc);
 - Falha de instalação (rede, equipamentos, etc);
- Operacionais
 - Vírus;
 - Falha de software;

Uma solução de alta disponibilidade mascara os efeitos de uma falha de software ou hardware, mantendo a disponibilidade dos componentes, de forma que o tempo de inatividade seja menor e menos perceptível aos usuários, segundo Marques(aceso 2010).

Pensando em banco de dados SQL Server, as opções de solução incluem:

Cluster de failover

Segundo Manuais (aceso 2010), Cluster de failover proporciona suporte de alta disponibilidade para uma instância inteira do SQL Server. É uma combinação de um ou mais nós, ou servidores, com dois ou mais discos compartilhados. A qualquer momento, cada grupo de recursos pertence a apenas um nó do cluster. O serviço de aplicativo tem um nome virtual que é independente dos nomes de nó e é chamado de nome de instância do cluster de failover. Um aplicativo pode se conectar a instância do cluster de failover fazendo apenas referência ao nome dela. O aplicativo não tem precisa saber qual nó hospeda a instância do cluster de failover, aparece na rede como se fosse um único computador, mas tem funcionalidade que proporciona failover de um nó para outro se o nó atual se tornar indisponível. Por exemplo,

durante uma falha de hardware que não seja o disco, uma falha do sistema operacional ou uma atualização planejada do sistema operacional, você pode configurar uma instância do SQL Server em um nó de um cluster de failover para executar failover em qualquer outro nó do grupo de discos.

O detalhe é que ele não protege contra falhas de disco. É usado para reduzir tempo de inatividade do sistema e oferecer maior disponibilidade de aplicativos ao banco de dados.

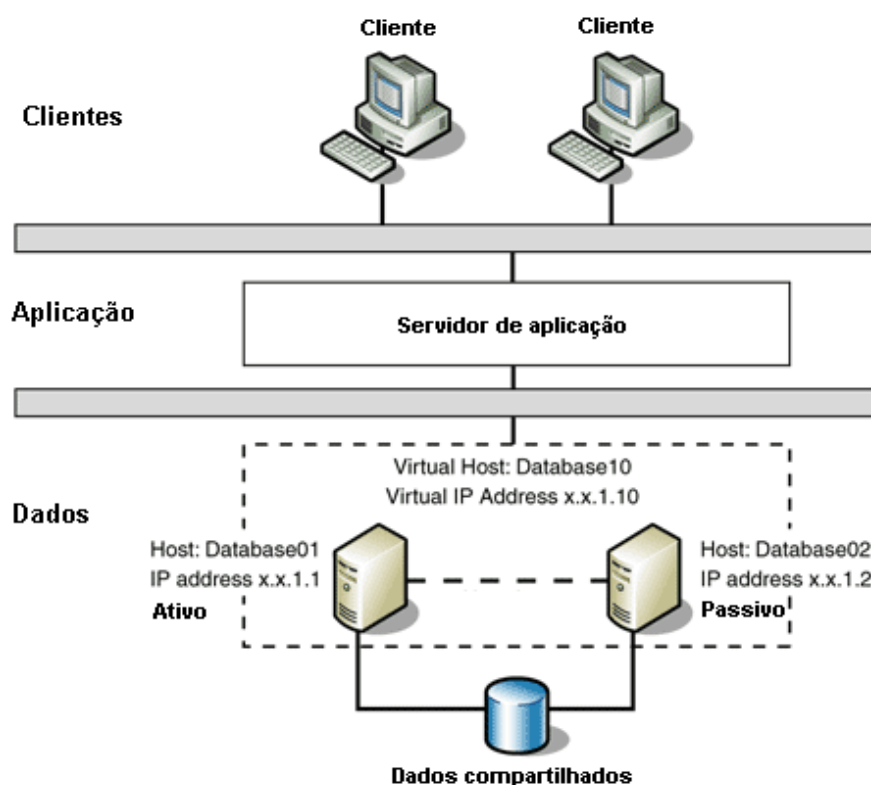


Figura 3 – Cluster de Failover

A figura 3 mostra um cenário onde um servidor de aplicação que tem 2 clientes, está ligado a uma instancia de Cluster de Failover, um endereço virtual que corresponde a dois servidores. Na falha do servidor ativo, o passivo pode ser acessado sem que a aplicação tenha que desviar o acesso.

Espelhamento de banco de dados

Segundo Manuais (acesso 2010), esta é uma solução de software para aumentar a disponibilidade do banco de dados, dando suporte a failover quase

instantâneo. Pode ser usado para manter um único banco de dados de espera (ou banco de dados espelho), para um banco de dados principal.

O banco de dados espelho é criado pela restauração de um backup do banco de dados principal sem recuperação ou com espera. Isso torna o banco de dados espelho inacessível aos clientes.

Cada configuração de espelhamento de banco de dados envolve um servidor principal que contém o banco de dados principal e um servidor espelho que contém o banco de dados espelho. O servidor espelho mantém o banco de dados espelho continuamente atualizado com o banco de dados principal.

Em sua configuração mais simples, o espelhamento de banco de dados envolve apenas os servidores principal e espelho. Nessa configuração, se o servidor principal for perdido, o servidor espelho poderá ser usado como um servidor em espera passiva, com possível perda de dados, pois pode não ter existido tempo suficiente de algumas informações serem espelhadas. Já o modo de alta segurança suporta uma configuração alternativa, o modo de alta segurança com failover automático. Essa configuração envolve uma terceira instância de servidor, conhecida como testemunha, que possibilita que o servidor espelho atue como um servidor em espera ativa, lembrando que o failover do banco de dados principal para o banco de dados espelho pode demorar alguns segundos.

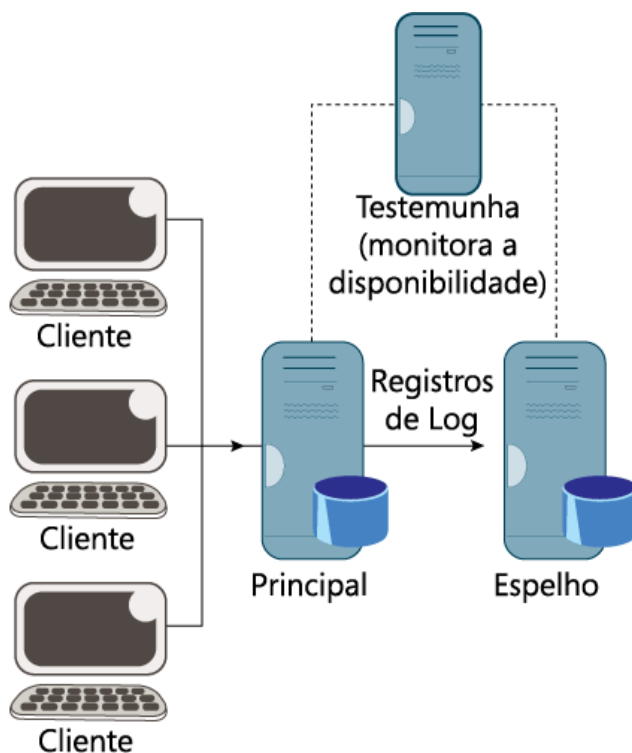


Figura 4 – Espelhamento de dados - Alta Segurança

A figura 4 mostra um esquema de espelhamento de dados em modo de alta segurança, ou seja, com um servidor de monitoramento, o servidor testemunha.

Envio de logs

O envio de logs, segundo Manuais (acesso 2010), como o espelhamento de banco de dados, opera no nível do banco de dados. Se diferem pois o envio de logs pode ser usado para manter um ou mais bancos de dados de espera passiva para um banco de dados de produção correspondente, chamado de banco de dados primário. Os bancos de dados de espera são chamados de bancos de dados secundários. Assim, cada banco de dados secundário é criado pela restauração de um backup do banco de dados primário sem recuperação ou com espera. A restauração com espera permite usar o banco de dados secundário para relatório limitado.

A configuração de envio de logs inclui um único servidor primário que contém o banco de dados primário, um ou mais servidores secundários que por sua vez têm um banco de dados secundário cada um, e um servidor de monitoramento. Cada

servidor secundário atualiza seu banco de dados secundário em intervalos definidos a partir de backups de log do banco de dados primário. O envio de logs envolve um atraso modificável pelo usuário entre o momento em que o servidor primário cria um backup de log do banco de dados primário e quando o servidor secundário restaura o backup do log. Antes que um failover possa ocorrer, um banco de dados deve ser atualizado completamente pela aplicação manual de quaisquer backups de log não restaurados.

O envio de logs fornece a flexibilidade de suportar vários bancos de dados de espera, podendo ser usado independentemente ou como um suplemento para o espelhamento de banco de dados. Quando essas soluções forem usadas conjuntamente, o banco de dados principal atual da configuração de espelhamento de banco de dados também será o banco de dados primário atual da configuração de envio de logs.

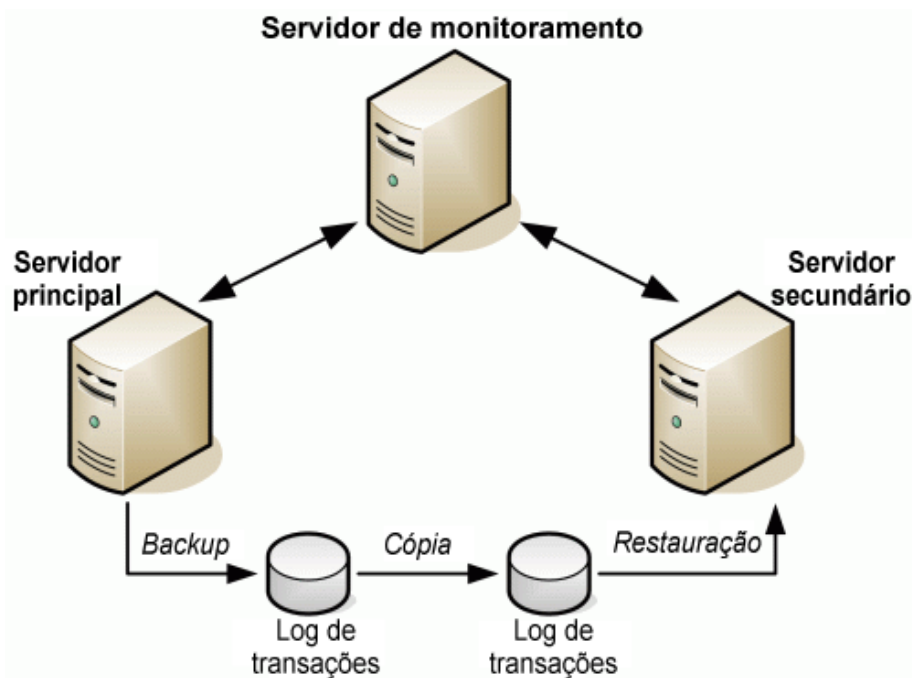


Figura 5 – Envio de logs – Configuração simples

A figura 5 mostra um esquema de envio de logs em sua configuração mais simples, com um servidor de monitoramento ligado a um servidor principal e um secundário.

Bancos de dados compartilhados evolutivos

Permite expandir um banco de dados somente leitura criado exclusivamente para relatórios. O banco de dados de relatório deve residir em um conjunto de volumes dedicados somente leitura cuja finalidade principal é hospedar o banco de dados. Depois de criado o banco de dados de relatório em um conjunto de volumes de relatório, os volumes são marcados como somente leitura e montados para vários servidores de relatório. Em cada servidor de relatório, o banco de dados de relatório é anexado a uma instância do SQL Server e se torna disponível como um banco de dados evolutivo compartilhado. Uma vez estabelecido como um banco de dados compartilhado evolutivo, o banco de dados de relatório pode ser compartilhado por clientes que usam servidores de relatório diferentes. Para consultar o banco de dados, um usuário ou um aplicativo pode se conectar a qualquer instância do servidor ao qual o banco de dados está anexado. Para uma determinada versão de banco de dados de relatório, os clientes em servidores diferentes obtêm uma exibição idêntica dos dados de relatório, tornando os resultados de consulta consistentes entre os servidores, segundo Manuais (acesso 2010).

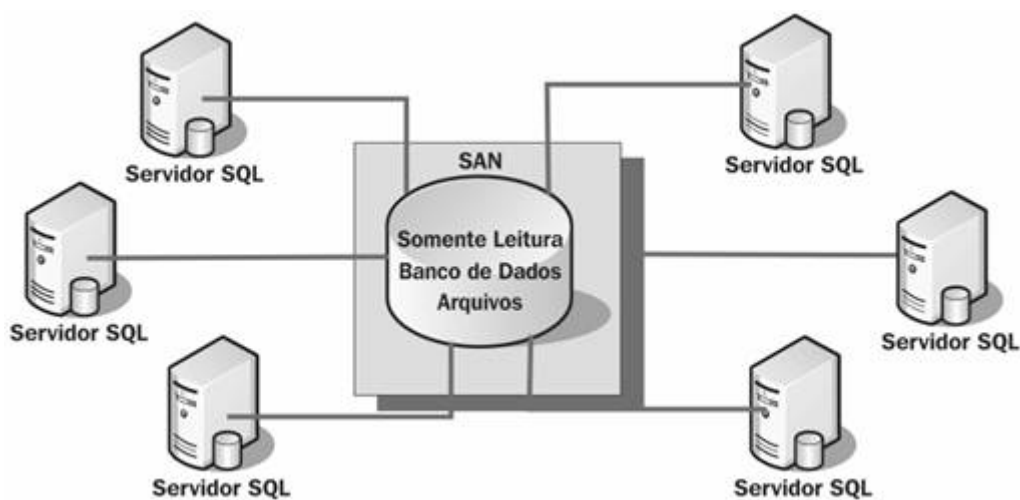


Figura 6 - Bancos de dados compartilhados evolutivos

A figura 6 mostra um esquema de banco de dados compartilhados evolutivos para 6 servidores de relatório. Um banco de dados compartilhados evolutivos só pode ser disponibilizado a partir de um Storage Area Network (SAN).

Replicação de dados

A replicação de dados é a única das soluções Microsoft SQL Server existente em todas as versões, considerada a mais utilizada por quem usa este SGBD e será descrita com profundidade mais a frente.

Mas porque existem tantas opções no SQL Server para se resolver um mesmo problema? As diversas opções de alta disponibilidade oferecidas visam atender a qualquer cenário, não importa o tamanho da empresa, tanto com banco de dados de 100MB quanto 30TB, existe uma solução mais apropriada para cada cenário. Deve-se analisar caso a caso e optar pela melhor opção, ou até mesmo utilizá-las de maneira combinada.

4. Replicação de Dados - Conceitos

A replicação é um conjunto de tecnologias para copiar e distribuir dados e objetos de um banco de dados para outro e, em seguida, sincronizar os bancos de dados para manter a consistência. Usando replicação, é possível distribuir dados para diferentes locais e para usuários remotos e móveis através de redes locais e de longa distância, conexões dial-up, conexões sem fio e a Internet.

Usa um modelo de publicação/assinatura, permitido que um servidor primário, conhecido como Publicador distribua os dados para um ou mais servidores secundários, chamados de Assinantes. A replicação possibilita disponibilidade em tempo real e escalabilidade entre esses servidores. Suporta filtragem para fornecer um subconjunto de dados aos Assinantes e também permite atualizações particionadas. Os assinantes ficam online e disponíveis para relatórios e outras funções, sem recuperação de consultas.

Para entender melhor, deve-se pensar da replicação SQL Server como uma revista. A replicação utiliza a metáfora da indústria de publicação para representar os componentes de uma topologia de replicação, que inclui o Publicador, Distribuidor, Assinantes, publicações, artigos e assinaturas.

- Um publicador de revista produz uma ou mais publicações
- Uma publicação contém artigos
- O publicador distribui a revista diretamente ou usa um distribuidor
- Os assinantes recebem as publicações que assinaram

Apesar da metáfora de revista ser útil para compreender a replicação, é importante observar que a replicação do SQL Server inclui a funcionalidade que não é representada nesta metáfora, particularmente o recurso de um Assinante efetuar atualizações e de um Publicador enviar alterações incrementais para os artigos de uma publicação.

Uma topologia de replicação define a relação entre os servidores e cópias de dados e esclarece a lógica que determina como os dados fluem entre os servidores. Existem diversos processos de replicação (mencionados como agentes) responsáveis pela cópia e movimentação de dados entre o Publicador e os Assinantes. A figura a seguir é uma visão geral dos componentes e processos envolvidos em replicação, retirada de Manuais (acesso 2010).

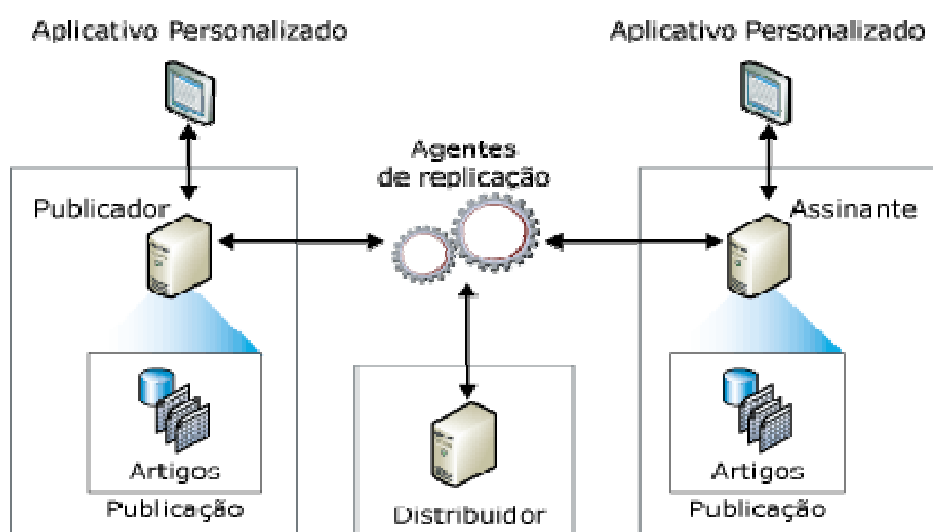


Figura 7 – Visão geral dos componentes de uma replicação

Abaixo serão descritas as funcionalidades de cada um dos componentes de replicação, segundo Manuais (acesso 2010):

Publicador:

É uma instância de banco de dados que disponibiliza dados para outros locais por meio de replicação. O Publicador pode ter uma ou mais publicações, cada uma definindo um conjunto de objetos relacionado de forma lógica e os dados para replicar.

Distribuidor

É uma instância de banco de dados que atua como um armazenamento para replicação de dados específicos associados a um ou mais Publicadores. Cada Publicador é associado a um único banco de dados (conhecido como um banco de dados de distribuição), o Distribuidor. O banco de dados de distribuição armazena os dados de status de replicação, metadados sobre a publicação e, em alguns casos, atua como uma fila para mover dados do Publicador para os Assinantes. Em muitos casos, uma única instância de servidor de banco de dados atua como Publicador e Distribuidor. Isto é conhecido como um Distribuidor local. Quando o Distribuidor e o Publicador estão configurados em instâncias separadas de servidor de banco de dados, o Distribuidor é conhecido como Distribuidor remoto.

Assinante

É uma instância de banco de dados que recebe dados replicados. Um Assinante pode receber dados de diversos Publicadores e publicações. Dependendo do tipo de replicação escolhida, o Assinante também pode passar as alterações de dados de volta ao Publicador ou republicar os dados para outros.

Artigo

Um artigo identifica um objeto de banco de dados que é incluído em uma publicação. Uma publicação pode conter tipos diferentes de artigos, incluindo tabelas, exibições, procedimentos armazenados e outros objetos. Quando as tabelas são publicadas como artigos, os filtros podem ser usados para restringir as colunas e linhas dos dados enviados aos Assinantes.

Publicação

Uma publicação é uma coleção de um ou mais artigos de um banco de dados. O agrupamento de diversos artigos em uma publicação facilita especificar um conjunto de objetos de banco de dados logicamente relacionado e os dados que são replicados como uma unidade.

Assinatura

Uma assinatura é uma solicitação para uma cópia de uma publicação a ser distribuída a um Assinante. A assinatura define qual publicação será recebida, onde e quando. Existem dois tipos de assinatura:

- **Push:** Em uma assinatura push, o Publicador propaga alterações para o Assinante sem solicitação do Assinante. As alterações podem ser empurradas para os Assinantes sob demanda continuamente ou com base em agendamento.
- **Pull:** Em uma assinatura pull, o Assinante solicita que alterações sejam feitas no Publicador. As assinaturas pull permitem que o usuário, no Assinante, determine quando as alterações de dados serão sincronizadas.

4.1. Tipos de replicação

O SQL Server oferece diferentes tipos de replicação para diferentes requisitos de aplicativo: replicação de instantâneo, replicação transacional e replicação de mesclagem.

4.1.1. Replicação Transacional

A replicação transacional, segundo Manuais (acesso 2010), normalmente é usada em cenários de servidor para servidor que requerem alta taxa de transferência, incluindo: melhora da escalabilidade e disponibilidade; armazenamento de dados data warehouse e relatórios; integração de dados de vários sites; integração de dados heterogêneos e descarregamento de processamento em lote.

A replicação transacional normalmente inicia com um instantâneo dos objetos e dados do banco de dados de publicação. Assim que o instantâneo inicial é tirado, as alterações subseqüentes nos dados e as modificações no esquema efetuadas no Publicador geralmente são distribuídas para o Assinante assim que ocorrem (quase em tempo real). As alterações nos dados são aplicadas ao Assinante na mesma ordem e dentro dos mesmos limites de transação conforme ocorreram no Publicador; por isso, dentro de uma publicação, a consistência transacional é assegurada.

Como citado anteriormente, a replicação transacional é normalmente usada em ambientes do tipo servidor para servidor e é apropriada em cada um dos seguintes casos:

- Você quer que as alterações com incremento sejam propagadas para os Assinantes à medida que ocorrem.
- O aplicativo requer baixa latência entre as mudanças de hora feitas no Publicador, assim as mudanças chegarão ao Assinante.
- O aplicativo requer acesso aos estados de dados intermediários. Por exemplo, se uma linha muda cinco vezes, a replicação transacional permite que um aplicativo responda a cada mudança (como acionar um gatilho), e não simplesmente uma mudança de dados da rede na linha.
- O Publicador tem um volume muito alto de atividade de inserção, atualização e exclusão.

- O Publicador ou Assinante é um banco de dados que não é do tipo SQL Server, como Oracle.

Por padrão, os Assinantes de publicações transacionais devem ser tratados como somente leitura, porque as alterações não são propagadas de volta para o Publicador. Porém, replicação transacional oferece opções que permitem atualizações ao Assinante.

A replicação transacional é implementada pelo Snapshot Agent, Log Reader Agent e Distribution Agent do SQL Server, abaixo serão descritos cada um desses agentes, segundo Manuais (acesso 2010).

Snapshot Agent

Prepara os arquivos de instantâneo que contêm o esquema e os dados das tabelas publicadas e os objetos do banco de dados, armazena os arquivos na pasta do instantâneo e registra os trabalhos de sincronização do banco de dados de distribuição no Distribuidor.

Log Reader Agent

Monitora o log de transações de cada banco de dados configurado para replicação transacional e copia as transações marcadas para replicação do log de transações no banco de dados de distribuição, que atua como uma fila confiável para armazenar e avançar.

Distribution Agent

Copia os arquivos do instantâneo inicial da pasta de instantâneo e as transações contidas nas tabelas do banco de dados de distribuição para os Assinantes.

As alterações incrementais feitas no Publicador fluem para os Assinantes de acordo com a programação do Distribution Agent, que pode ser executado continuamente para ficar com latência mínima, ou em intervalos programados. Como as alterações nos dados devem ser feitas no Publicador (quando a replicação transacional for usada sem as opções de atualização imediata ou atualizações em fila), os conflitos de atualizações serão evitados. Por fim, todos os Assinantes alcançarão os mesmos valores que o Publicador. Se as opções de atualização

imediate ou atualizações em fila forem usadas com a replicação transacional, as atualizações poderão ser feitas no Assinante, e com a atualização em fila, poderão ocorrer conflitos. A ilustração a seguir mostra os principais componentes de replicação transacional.

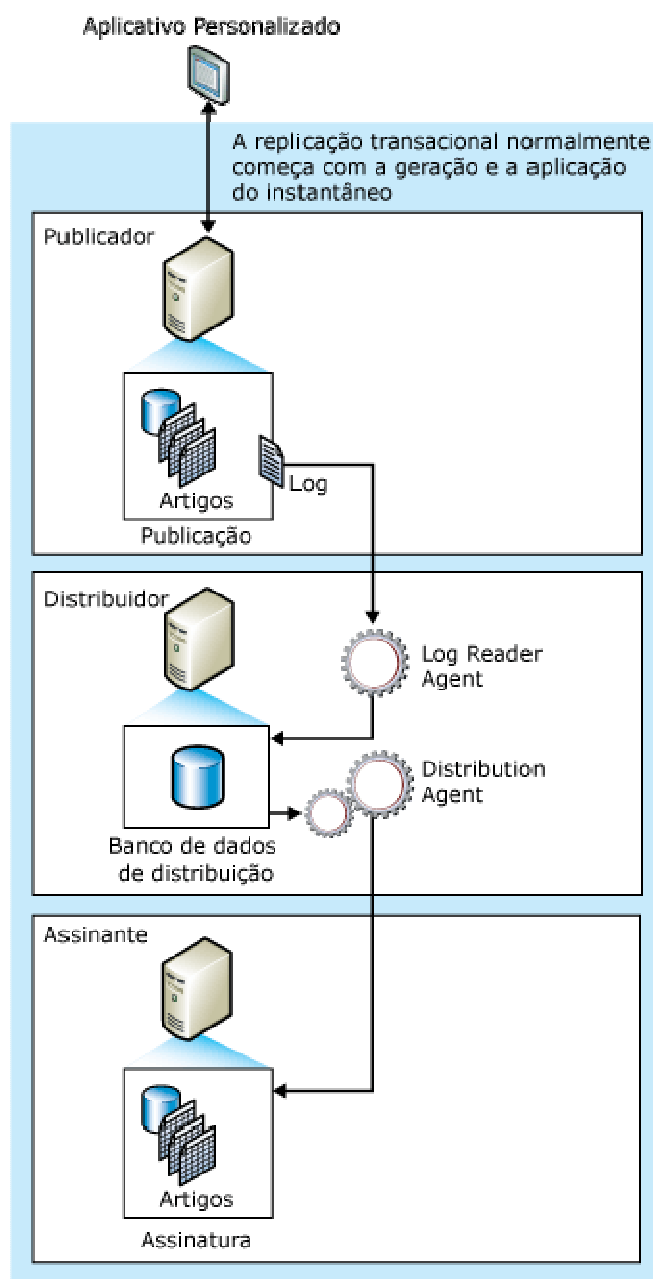


Figura 8 - Principais componentes de replicação transacional

Antes que um novo Assinante de replicação transacional possa receber alterações incrementais de um Publicador, o Assinante deverá conter as tabelas com o mesmo esquema e dados que as tabelas do Publicador. O conjunto de dados inicial normalmente é um instantâneo criado pelo Snapshot Agent e distribuído e

aplicado pelo Distribution Agent. O conjunto de dados inicial também pode ser fornecido por meio de um backup ou outros meios, como SQL Server Integration Services.

Quando os instantâneos forem distribuídos e aplicados aos Assinantes, somente esses Assinantes à espera dos instantâneos iniciais serão afetados. Outros Assinantes dessa publicação (que já foram inicializados) não serão afetados.

A replicação de instantâneo coloca bloqueios compartilhados em todas as tabelas publicadas como parte de replicação pela duração da geração do instantâneo. Isto pode impedir que as atualizações sejam feitas nas tabelas de publicação. O processamento simultâneo de instantâneos, o padrão com a replicação transacional, não mantém os bloqueios compartilhados no lugar durante toda a geração do instantâneo, o que permite que os usuários continuem trabalhando ininterruptamente enquanto a replicação cria os arquivos iniciais de instantâneo.

Os procedimentos pelos quais o Snapshot Agent implementa o instantâneo inicial na replicação transacional são os mesmos usados na replicação de instantâneo (exceto como é descrito acima em relação ao processamento simultâneo de instantâneos).

Depois que os arquivos de instantâneo foram gerados, você pode exibi-los na pasta de instantâneo usando o Windows Explorer Microsoft por exemplo.

O Log Reader Agent é executado no Distribuidor, em geral ele é executado continuamente, mas ele também pode executar de acordo com uma programação que você estabelece. Ao executar, o Log Reader Agent primeiro lê o log de transação de publicação (o mesmo log de banco de dados usado para rastreamento de transações e recuperação durante as operações regulares do Mecanismo do Banco de Dados do SQL Server), e identifica qualquer instrução de INSERT, UPDATE e DELETE ou outras modificações feitas nos dados em transações que foram marcadas para replicação. Em seguida, o agente copia essas transações em lotes para o banco de dados de distribuição no Distribuidor. O Log Reader Agent usa o procedimento armazenado interno para obter o conjunto seguinte de comandos do log marcados para replicação. O banco de dados de distribuição se torna em

seguida a fila de armazenamento e avanço da qual as alterações são enviadas aos Assinantes. Somente as transações confirmadas são enviadas ao banco de dados de distribuição.

Após o lote inteiro de transações ter sido gravado com êxito no banco de dados de distribuição, ele é confirmado. Após a confirmação de cada lote de comandos no Distribuidor, o Log Reader Agent chama um procedimento para marcar onde a replicação foi concluída por último. Por fim, o agente marca as linhas no log de transação que estão prontas para serem apuradas. As linhas que ainda esperam para serem replicadas não são enviadas.

Os comandos de transação são armazenados no banco de dados de distribuição até serem propagados a todos os Assinantes ou até que o período de retenção máxima de distribuição tenha sido atingido. Os Assinantes recebem as transações na mesma ordem em que foram aplicados no Publicador Manuais (acesso 2010).

O Distribution Agent é executado no Distribuidor para assinaturas push e no Assinante para assinaturas pull. O agente move as transações do banco de dados de distribuição para o Assinante. Se uma assinatura estiver marcada para validação, o Distribution Agent também verificará se os dados do Publicador e do Assinante coincidem.

4.1.2. Replicação de Mesclagem

Segundo Manuais (acesso 2010), a replicação de mesclagem é projetada principalmente para aplicativos móveis ou de servidor distribuído que possuem possíveis conflitos de dados. Os cenários comuns incluem: troca de dados com usuários móveis; aplicativos de POS (ponto de vendas) para o consumidor e integração de dados de vários sites.

A replicação de mesclagem, como a replicação transacional, normalmente inicia com um instantâneo dos objetos e dos dados do banco de dados de publicação. As alterações dos dados subseqüentes e as modificações de esquema feitas no Publicador e nos Assinantes são rastreadas com gatilhos. O Assinante sincroniza com o Publicador quando está conectado à rede e permuta todas as linhas que foram alteradas entre o Publicador e o Assinante desde a última vez que a sincronização ocorreu.

Normalmente, a replicação de mesclagem é usada em ambientes do tipo servidor para clientes e é apropriada em quaisquer das seguintes situações:

- Diversos Assinantes podem atualizar os mesmos dados diversas vezes e propagar essas alterações para o Publicador e outros Assinantes.
- Os Assinantes precisam receber dados, fazer alterações offline e sincronizar posteriormente as alterações com o Publicador e outros Assinantes.
- Cada Assinante requer uma partição diferente de dados.
- Conflitos podem ocorrer e, quando isto acontecer, você precisará do recurso para detectá-los e encontrar a solução.
- O aplicativo requer a alteração nos dados da rede no lugar do acesso aos estados de dados intermediários. Por exemplo, se uma linha for alterada cinco vezes em um Assinante antes de ele sincronizar com um Publicador, a linha será alterada somente uma vez no Publicador pra refletir a alteração dos dados na rede (ou seja, o quinto valor).

A replicação de mesclagem permite que diversos sites operem de forma autônoma e que as atualizações posteriores de mesclagem obtenham um único resultado uniforme. Como as atualizações são feitas em mais de um nó, os mesmos dados podem ter sido atualizados pelo Publicador e por mais de um Assinante. Portanto, os conflitos podem ocorrer quando as atualizações forem mescladas, mas a replicação de mesclagem fornece várias maneiras para controlar os conflitos.

4.1.3. Replicação de Instantâneo

Segundo Manuais (acesso 2010), a replicação de instantâneo é usada para fornecer o conjunto inicial de dados para replicação transacional e de mesclagem. Ela também pode ser usada quando as atualizações completas de dados estiverem apropriadas.

A replicação de instantâneo distribui os dados exatamente como eles aparecem em um momento específico no tempo e não monitora para as atualizações dos dados. Quando a sincronização ocorre, todo o instantâneo é gerado e enviado aos Assinantes.

Pode ser usada por si só, mas o processo de instantâneo (o que cria uma cópia de todos os objetos e dados especificados por uma publicação) também é usado regularmente para fornecer o ajuste inicial dos dados e dos objetos do banco de dados para publicações de mesclagem e transacionais.

O uso da replicação de instantâneo por si só é mais apropriado quando um ou mais dos itens a seguir for verdadeiro:

- As alterações de dados ocorrerem raramente.
- É aceitável ter cópias de dados desatualizadas em relação ao Publicador por um período de tempo.
- Replicação de pequenos volumes de dados.
- Um volume grande de alterações ocorre por um curto período de tempo.

Assim, este tipo de replicação é mais apropriada quando as alterações de dados forem significativas, mas pouco frequentes. Por exemplo, se uma empresa de vendas mantiver uma lista de preços de produtos e os preços forem todos atualizados ao mesmo tempo uma ou duas vezes por ano, é recomendada a replicação de todo o instantâneo de dados após ele ter sido alterado. Ao ter certos tipos de dados determinados, os instantâneos mais frequentes também podem ser

apropriados. Por exemplo, se uma tabela relativamente pequena for atualizada no Publicador durante o dia, mas alguma latência for aceitável, as alterações podem ser distribuídas pela noite como um instantâneo.

A replicação de instantâneo tem uma sobrecarga contínua no Publicador inferior à replicação transacional, porque as alterações incrementais não são rastreadas. No entanto, se o ajuste do conjunto de dados que estiver sendo replicado for muito grande, ele exigirá recursos substanciais para gerar e aplicar o instantâneo. Considere a dimensão de todo o conjunto de dados e a frequência de alterações nos dados ao avaliar a possibilidade de utilizar a replicação de instantâneo.

Avaliando os tipos de replicação, a replicação transacional é a mais indicada quando o volume de dados é alto e existe a necessidade de alta disponibilidade dos dados.

4.2. Ambiente

A replicação de dados pode ser classificada por dois ambientes: servidor para servidor e servidor para cliente.

Servidor – Servidor

O ambiente de replicação de servidor para servidor é mais usado pelas seguintes vantagens, segundo Manuais (acesso 2010):

- **Melhora da escalabilidade e a disponibilidade:** A manutenção de cópias de dados continuamente atualizadas permite que a atividade de leitura seja escalada por diversos servidores. A redundância resultante da manutenção de diversas cópias dos mesmos dados também é essencial durante a manutenção do sistema planejada e não planejada.

Muitos aplicativos podem se beneficiar do aumento da escalabilidade e disponibilidade. A manutenção de cópias de dados continuamente atualizadas

permite que a atividade de leitura seja escalada por diversos servidores. A redundância resultante da manutenção de diversas cópias dos mesmos dados também é útil durante interrupções planejadas e não planejadas. Se um servidor estiver indisponível, as solicitações podem ser direcionadas para um servidor diferente com os mesmos dados até que o servidor original (ou um substituto) retorne ao status online.

- **Data warehouse e relatórios:** Os servidores de data warehouse e de relatório usam freqüentemente dados de servidores de processamento de transações online (OLTP). A replicação é usada freqüentemente para distribuir dados de bancos de dados de processamento transacional (OLTP) para bancos de dados de preparo e bancos de dados usados para as finalidades de gerar relatórios, suporte à decisão e análise.

O diagrama que segue exibe um cenário típico com dados replicados de um servidor de processamento on-line para um servidor de relatórios e um servidor de preparo para análise OLAP e ROLAP:

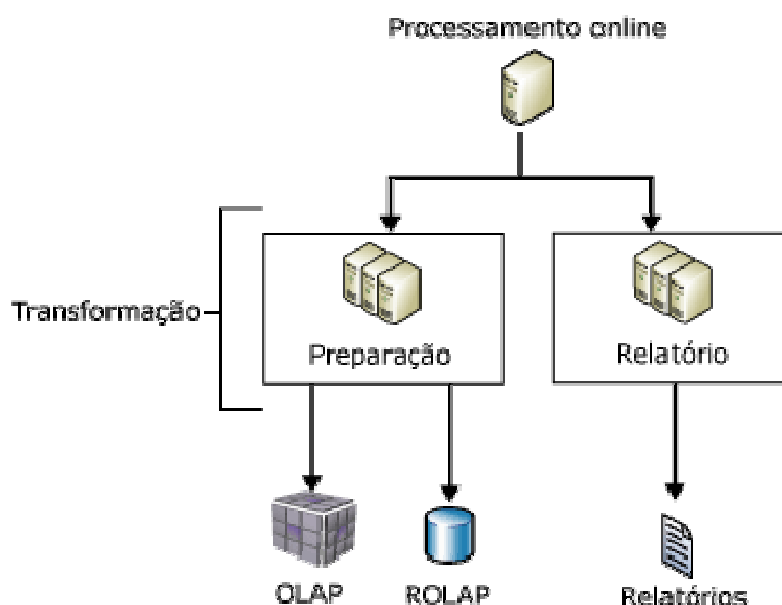


Figura 9 – Replicação voltada a Data Warehouse e relatórios

- **Integrando dados de vários sites:** Os dados são freqüentemente acumulados de escritórios remotos e consolidados em um escritório central. De forma semelhante, os dados podem ser replicados para escritórios remotos. Muitas

empresas têm escritórios ou entidades regionais que coletam e processam dados que devem ser enviados a um local central. O diagrama a seguir mostra um cenário típico no qual os dados de sites remotos são acumulados. Também são enviados dados somente leitura para cada site remoto.

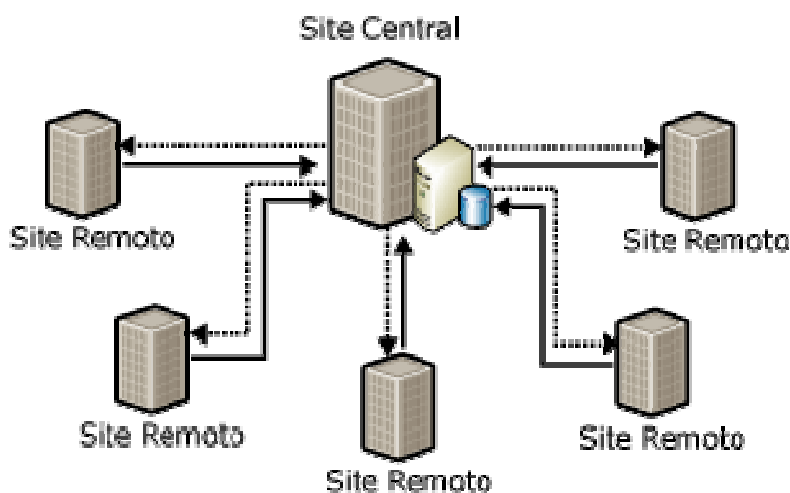


Figura 10 – Integrando dados de vários sites

- **Integrando dados heterogêneos:** Alguns aplicativos dependem dos dados que são enviados de ou para bancos de dados diferentes do SQL Server. Muitas empresas e negócios têm dados armazenados em bancos de dados de diversos fornecedores. A integração destes dados é freqüentemente um componente fundamental para permitir que os sistemas trabalhem juntos em uma empresa. A replicação permite integrar dados heterogêneos de dois modos:
 - Usar Oracle como uma fonte de dados que podem ser replicados para Microsoft SQL Server, bancos de dados IBM e Oracle.
 - Usar SQL Server como uma fonte de dados que podem ser replicados para bancos de dados IBM e Oracle.

O diagrama a seguir ilustra a replicação dos dados de SQL Server para IBM DB2 e Oracle.

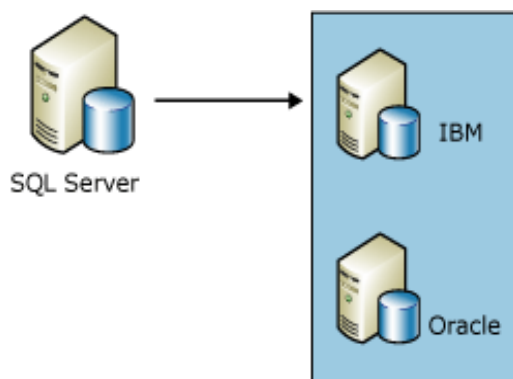


Figura 11 – Integrando dados para outros bancos

O diagrama a seguir ilustra a replicação dos dados de um banco de dados Oracle para outros bancos de dados. Os dados são inicialmente replicados para um banco de dados SQL Server e podem ser replicados em seguida para outros bancos de dados incluindo SQL Server, IBM DB2 e Oracle.

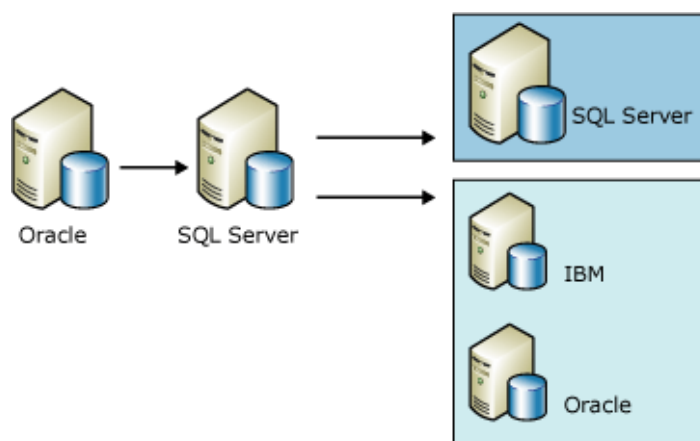


Figura 12 – Integrando dados do Oracle

- **Descarregando processamento em lote:** As operações em lote com frequência são um recurso muito intensivo para executar em um servidor OLTP.

Alguns aplicativos exigem que essas operações em lote de processamento intensivo sejam executadas nos dados. Em muitos casos, estas operações em lote não podem ser efetuadas no servidor OLTP, porque a sobrecarga no processamento interfere nas outras operações do servidor. Neste caso, é necessário executar o processamento em lote em um servidor separado. Em alguns casos, o processamento em lote é simplesmente descarregado; em outros casos, os resultados do processo em lote são propagados de volta para o servidor de processamento on-line.

O diagrama que segue exibe um cenário típico com dados replicados para um servidor de processamento em lote:

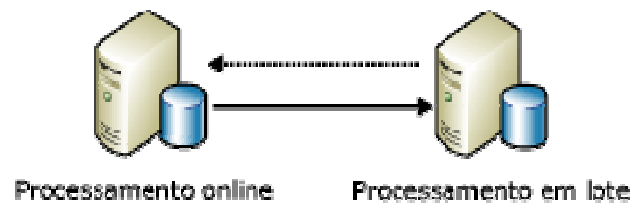


Figura 13 - Descarregando processamento em lote

Servidor - Cliente

Geralmente envolvem troca de dados entre servidores e estações de trabalho, laptops, tablets e dispositivos, classificadas de três formas, segundo Manuais (acesso 2010):

- **Integrando dados de diversos locais:** Quando os dados precisam ser integrados em diversos locais diferentes.

Exemplo: uma empresa tem várias filiais que coletam e processam dados que devem ser enviados ao escritório da matriz, que exige que:

- Os dados de inventário devem ser consolidados de um número de servidores em armazéns locais para o servidor central do escritório da matriz.
- As informações de cada filial devem ser enviadas para o servidor central.
- As informações de processamento de pedidos das filiais podem ser consolidadas.

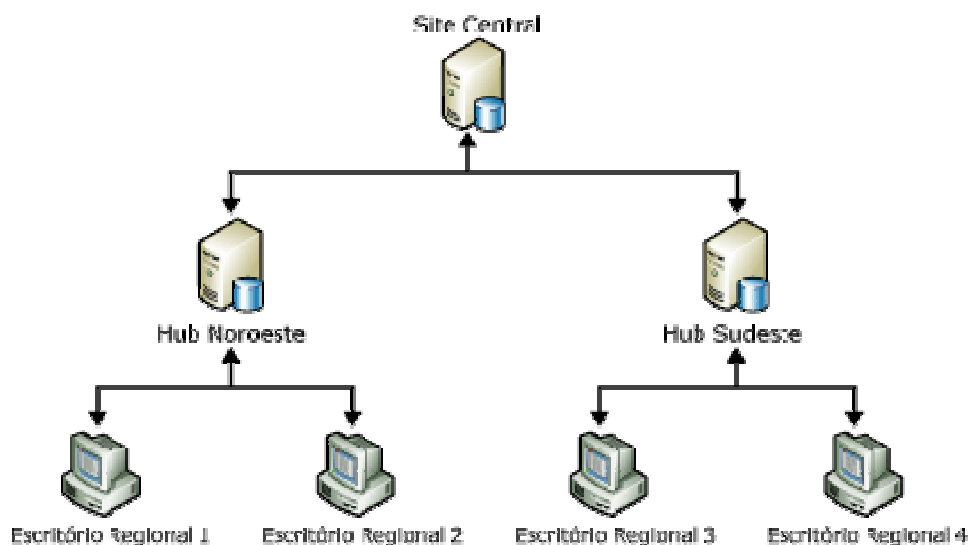


Figura 14 - Integrando dados de diversos locais

A figura x mostra este cenário comum, com os dados fluindo em duas direções entre um site central (matriz) e locais remotos(filiais).

- **Integrando dados com usuários móveis:** Quando há exigência de que os dados estejam disponíveis para usuários remotos, incluindo equipes de vendas, entregadores, e assim por diante.

Esta integração inclui a utilização de aplicativos CRM (Gerenciamento de Relacionamento com o Cliente), SFA (Automação de Força de Vendas) e FFA (Automação de Força do Campo).

A figura a seguir ilustra um cenário no qual um grupo de usuários se conecta por laptops diretamente a um servidor central:

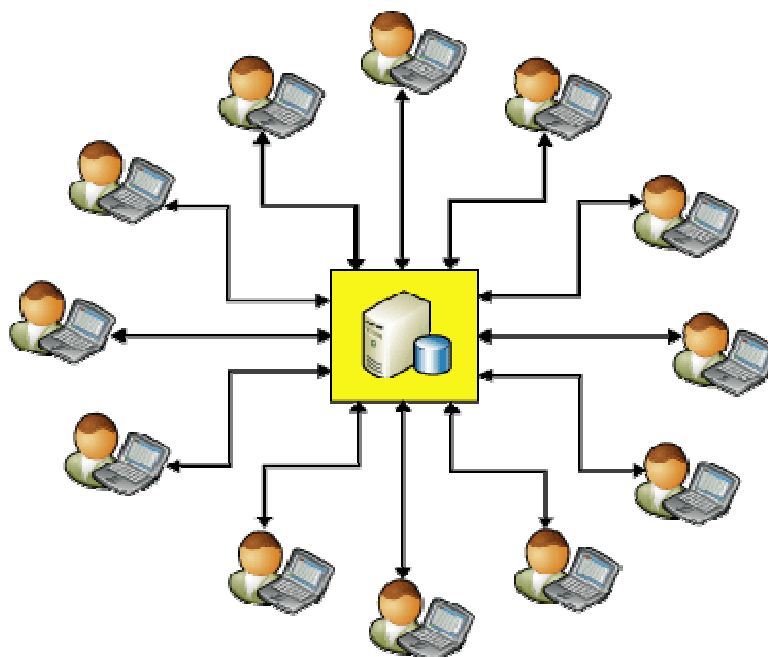


Figura 15 - Integrando dados com usuários móveis - Laptops

O próximo diagrama ilustra um cenário no qual um conjunto de usuários com Personal Digital Assistant (PDA), traduzido como Assistente Pessoal Digital, se conecta ao servidor central por meio de servidores de internet.

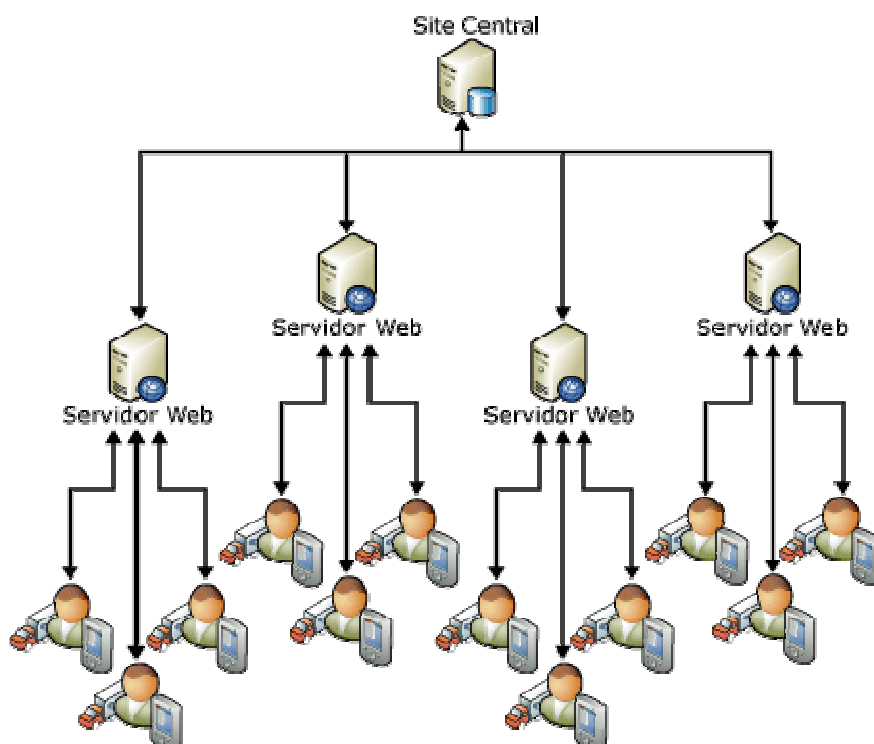


Figura 16 - Integrando dados com usuários móveis - PDAs

- **Aplicativos POS (Ponto de Venda ao Consumidor):** Os aplicativos POS, como os terminais de caixa e caixas eletrônicos, exigem que os dados sejam replicados de locais remotos para um local central.

Os exemplos incluem os terminais usados por caixas, bancos 24 horas, e em quiosques de lojas. Estes aplicativos coletam dados em sites remotos e os transmitem de volta a um local central, como um escritório central ou centro de dados. Nesses aplicativos é comum os dados serem coletados basicamente no ponto de vendas e em seguida carregados para os escritórios centrais sem conflitos, porque um único usuário remoto (normalmente um cliente ou atendente de vendas) está atualizando uma determinada parte dos dados.

A figura a seguir ilustra um cenário comum com os dados fluindo em duas direções entre um servidor central e locais remotos como caixas eletrônicos de um banco.

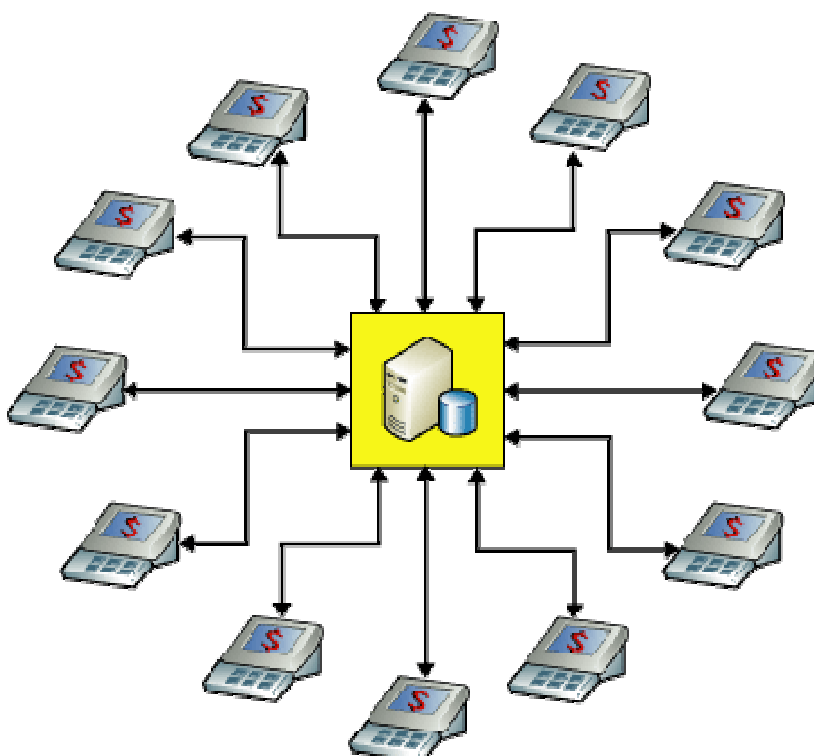


Figura 17 - Aplicativos POS – Caixa eletrônico

4.3. Exemplo de aplicação

A CK, uma empresa familiar brasileira, do ramo de confecção, fundada há quase 45 anos em Americana/SP a partir do trabalho artesanal de três irmãs, hoje atua no segmento de cama, mesa e banho oferecendo ao mercado produtos de alta qualidade, procurando ressaltar a riqueza e diversidade cultural do país CK (acesso 2010).

A empresa possui uma rede de 6 lojas dentro do estado de São Paulo, sendo quatro dessas lojas em Americana, uma em Limeira e uma em Piracicaba. Uma das unidades de Americana, a loja 4, também é o escritório central e a confecção, é recebedora de todos os produtos de revenda, assim, todos os artigos de confecção própria e produtos de revenda são distribuídos a partir desta loja para as outras lojas da rede.

Utiliza SGBD Microsoft SQL Server e sistemas ERP³ Microdata Sistemas, seu principal fornecedor de sistemas, também de Americana.

A CK ao contratar a Microdata Sistemas, tinha algumas necessidades iniciais:

- Integrar todos os dados da empresa desde cadastros a informações de Planejamento e Controle de Produção (PCP), por isso a escolha de um sistema ERP³ com suporte multi-empresa
- Compartilhar informações entre as lojas como cadastro de clientes, estoques, vendas, etc
- Alta disponibilidade do sistema e dos dados
- Ter controle dos dados dentro da própria empresa

Analisando essas necessidades, a replicação de dados foi escolhida como solução para alta disponibilidade, controle de dados dentro da empresa e compartilhamento de informações em tempo real para todas as lojas, e de maneira simplificada pelo uso de um sistema multi-empresa.

A conexão entre as lojas foi feita por meio de linhas dedicadas da Embratel, esse meio de conexão foi escolhido pela qualidade, pois foi considerada mais confiável quanto a quedas e também pelo fato dos equipamentos que permitem as

³ ERP (Enterprise Resource Planning) : São sistemas de informação que integram todos os dados e processos de uma organização em um único sistema.

comunicações podem ser controlados pela própria empresa. Apesar de ser mais cara que uma conexão VPN⁴, foi a que atendeu as necessidades particulares.

A figura 18 mostra um esquema da conexão entre lojas. Foram utilizadas 5 EILD, em topologia estrela, partindo da loja 4, o servidor distribuidor do escritório central.

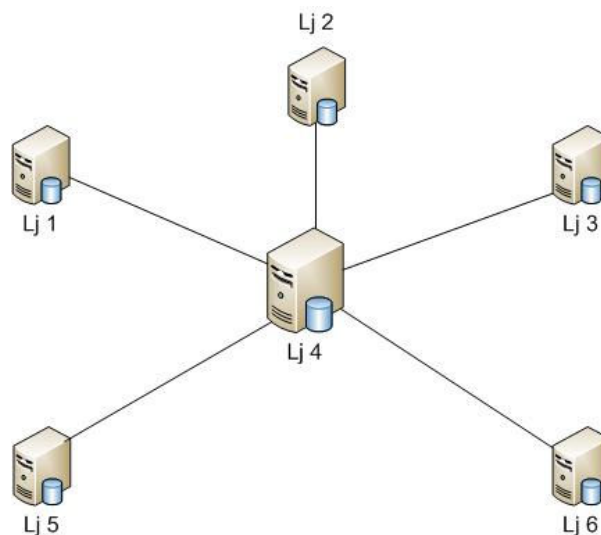


Figura 18 – Conexões via linhas dedicadas entre as lojas

A empresa desejava que as informações de todas as lojas estivessem disponíveis para todas as lojas, então foi escolhida a replicação transacional, pois:

- É a mais aconselhada em cenários de servidor para servidor que requerem alta taxa de transferência
- Os dados são replicados à medida que as alterações são feitas, quase que em tempo real para os assinante
- A opção que tem latência mais baixa

Seguindo essa descrição, todos os servidores das lojas são publicadores e assinantes, mas na loja 4, existem 2 servidores, um assinante e publicador e um servidor distribuidor da replicação.

Veja na figura 19, os servidores assinantes/publicadores das lojas, sendo o da loja 4 chamado de S1 e no centro, o servidor Distribuidor da loja 4, chamado de S2.

⁴ VPN (Virtual Private Network) : uma rede de comunicações privada normalmente utilizada por uma empresa ou um conjunto de empresas e/ou instituições, construída em cima de uma rede de comunicações pública (como por exemplo, a Internet).

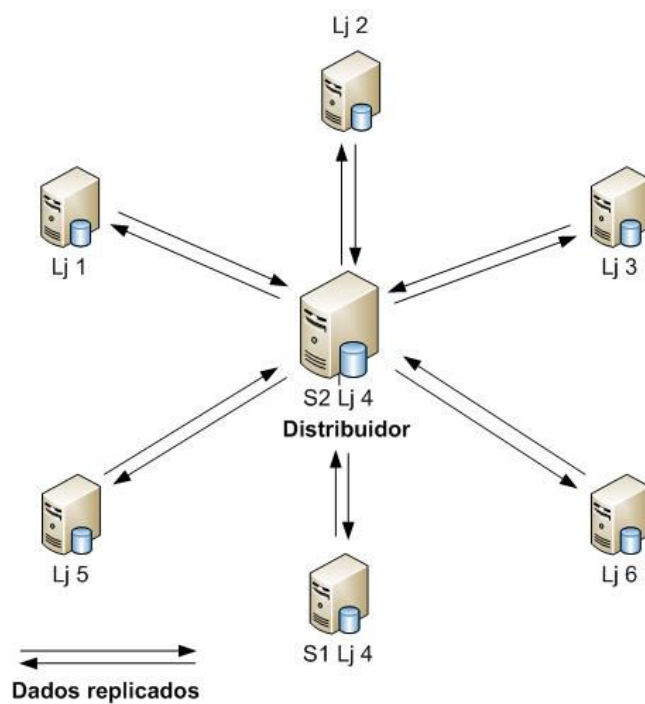


Figura 19 – Topologia replicação

Para entender melhor o mecanismo da replicação, a figura 20 mostra as lojas 1 e 2 ligadas ao Distribuidor, e um grupo de usuários ligados a cada uma das lojas.

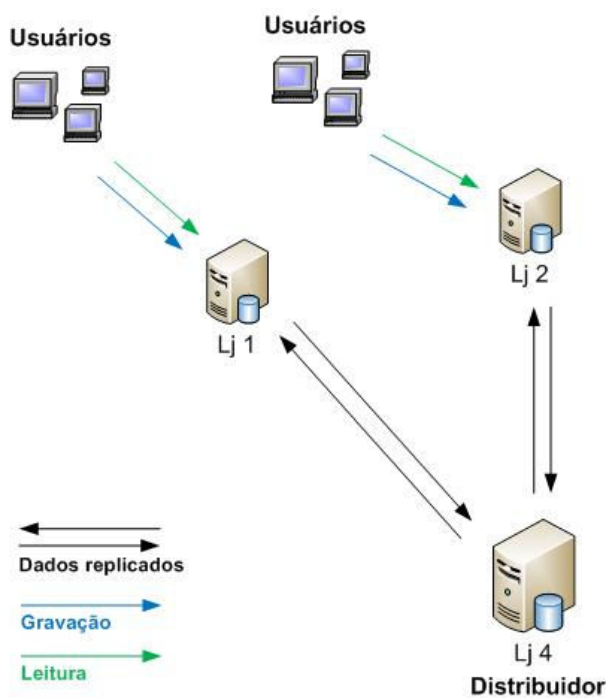


Figura 20 - Visualização detalhada da replicação entre 2 lojas

Desta forma, quando na loja 1, um cliente for cadastrado ou fizer uma compra por exemplo, estas informações serão publicadas pelo servidor da própria loja. As informações sendo publicadas, o Log Reader Agent que fica ativo no Distribuidor, e monitora os logs de transação de publicação, identifica as instruções de modificações feitas nos dados e em seguida copia essas transações em lotes para o banco de dados de distribuição. O Distribuidor então envia essas transações aos Assinantes, todos os assinantes receberão as mesmas informações, mantendo os bancos de dados atualizados.

Caso um servidor publicador/assinante caia, isso afetará somente os usuários conectados a este servidor, que poderão ter desviados os acessos para outro servidor. Quando este servidor voltar ao ar, o Distribuidor o atualizará com as informações que foram publicadas enquanto estava parado. Os dados não serão perdidos.

Caso o servidor Distribuidor caia, os servidores publicadores/assinantes funcionarão normalmente, todas as transações continuarão a ser publicadas. Ao voltar ao ar o Distribuidor continuará de onde parou a atualização dos dados para os assinantes. Os dados neste caso também não serão perdidos.

A replicação no caso da CK foi satisfatória, apesar de ser uma solução um pouco mais cara que as outras, pois a implantação de linhas dedicadas é mais cara e depende de profissionais habilitados na tecnologia, que não são facilmente encontrados.

5. Conclusão

Concluo que Alta Disponibilidade é um assunto atual e indispensável em qualquer área dentro de TI, não só quando o assunto é banco de dados, mas também em redes, aplicações, etc. Manter o acesso aos sistemas e dados é fundamental numa época em que as informações têm que ser rápidas, um segundo de atraso pode valer uma venda, um negócio fechado, até uma vida.

Os bancos de dados são essenciais para as corporações pela facilidade em compartilhar informações, tanto em curtas quanto longas distancias e a criação dos SGBDs, a normalização da linguagem de comunicação e do modelo relacional foram os responsáveis para que o sucesso acontecesse.

Voltado a banco de dados, vimos que o SQL Server é um SGBD que tem várias soluções disponíveis voltadas a Alta Disponibilidade, que atendem a todos os setores, tanto pequenas quanto a grandes empresas, em vários cenários.

Depois de ver as soluções de Alta Disponibilidade do SQL Server, conclui-se que a Replicação de Dados é a única que se aplica quanto se quer distribuir informações em tempo real tanto para outros servidores SQL Server quanto a clientes, e que não há perda de dados caso haja queda de algum dos servidores envolvidos na replicação, tanto distribuidores, quanto assinantes e publicadores.

Referências

- BRANDÃO, R. Conceitos sobre Disponibilidade - Parte I. Disponível em: <<http://technet.microsoft.com/pt-br/library/cc668492.aspx>>
Acesso em: 16 mar 2010. 21h56.
- CK. Confeção Kacyumara. Disponível em: <<http://www.kacyumara.com.br/>>
Acesso em: 08 mai 2010. 19h40.
- KIOSKEA. Elevada Disponibilidade. Disponível em:
< <http://pt.kioskea.net/contents/surete-fonctionnement/haute-disponibilite.php3>>
Acesso em: 15 mar 2010. 00.42h.
- MANUAIS Online do SQL Server 2008. MSDN.microsoft.com. Disponível em:
<<http://msdn.microsoft.com/pt-br/library/ms345392.aspx> >
Acesso em: 25 abr 2010. 14h01.
- MARQUES, C. Gerenciamento de Disponibilidade. Disponível em:
<<http://www.itcentral.com.br/default.asp?ACT=5&content=343&id=38&mnu=38>>
Acesso em: 21 mar 2010. 21h23.
- NICHTER, Z. Atingir a alta disponibilidade do SQL Server. Disponível em:
<<http://technet.microsoft.com/pt-br/magazine/2007.03.highavailability.aspx>>
Acesso em: 15 mar 2010. 00h35.
- NOVELLI, M. SGBD - Sistema Gerenciador de Banco de Dados. Disponível em:
<<http://www.plugmasters.com.br/sys/materias/108/1/SGBD---Sistema-Gerenciador-de-Banco-de-Dados>>
Acesso em: 15 mar 2010. 00h37.
- REZENDE, R. Conceitos Fundamentais de Banco de Dados - Parte 2. Disponível em: <<http://www.devmedia.com.br/articles/post-1678-Conceitos-Fundamentais-de-Banco-de-Dados-Parte-2.html>>
Acesso em: 15 mar 2010. 00h39.
- SILBERSCHATZ, A; KORTH, H.F e SUDARSHAN, S. **Sistema de Bancos de Dados**. Tradução Daniel Vieira. 5^a edição. Rio de Janeiro: Campus, 2006. p. 1-105;715-743.