

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
CENTRO UNIVERSITÁRIO DO RIO GRANDE DO NORTE
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

RAFAEL STROPPA GONÇALVES

**ALTA DISPONIBILIDADE COM REPLICAÇÃO DE BANCO DE DADOS
UTILIZANDO O MICROSOFT SQL SERVER 2008**

NATAL/RN

2012

RAFAEL STROPPA GONÇALVES

**ALTA DISPONIBILIDADE COM REPLICAÇÃO DE BANCO DE DADOS
UTILIZANDO O MICROSOFT SQL SERVER 2008**

Monografia apresentada ao Centro
Universitário do Rio Grande do Norte –
UNI-RN, como parte dos requisitos para
obtenção do título de Bacharel em
Sistemas de Informação.

ORIENTADOR:

Prof. Esp. Romulo Fagundes Cantanhede

NATAL/RN

2012

Catalogação na Publicação – Biblioteca do UNI-RN
Setor de Processos Técnicos

Gonçalves, Rafael Stroppa.

Alta disponibilidade com replicação de banco de dados utilizando o
Microsoft SQL Server 2008 / Rafael Stroppa Gonçalves. – Natal, 2012.
78 f.

Orientador: Prof^o. Esp. Romulo Fagundes Cantanhede.

Monografia (Graduação em Sistemas de Informação) – Centro
Universitário do Rio Grande do Norte.

1. Banco de dados - Monografia. 2. Replicação de dados - Monografia. 3.
SQL server 2008 - Monografia. 4. Alta disponibilidade - Monografia. I.
Cantanhede, Romulo Fagundes. II. Título.

RN/UNI-RN/BC

CDU 004.65

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
CENTRO UNIVERSITÁRIO DO RIO GRANDE DO NORTE
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

Rafael Stroppa Gonçalves

**ALTA DISPONIBILIDADE COM REPLICAÇÃO DE BANCO DE DADOS
UTILIZANDO O MICROSOFT SQL SERVER 2008**

Monografia apresentada e aprovada em 12 de dezembro de 2012, pela banca examinadora
composta pelos seguintes membros

Prof. Esp. Romulo Fagundes Cantanhede
Orientador

Profa. Msc. Lívia Maria Martins da Silva
Membro examinador

Prof. Esp. Itamir de M. Barroca Filho
Membro examinador

Natal/RN, 12 de dezembro de 2012.

A minha mãe Suze, minha esposa Adrilúcia e aos meus filhos Reno e Pedro Henrique. Dedico este trabalho por estarem sempre ao meu lado nos momentos de conquistas.

AGRADECIMENTOS

Em primeiro lugar a **Deus!** Por sempre, de forma milagrosa, permitir que momentos como este sejam possíveis em minha vida.

A minha esposa **Adrilúcia**, por ter acreditado em mim, dado forças para completar essa jornada e por ter compreendido e proporcionado momentos de dedicação para este trabalho.

Aos meus filhos, **Reno Neto e Pedro Henrique**, que são o motivo de tanto esforço e dedicação em minha vida.

A minha mãe **Suze**, que sempre esteve do meu lado nos momentos de grandes conquistas e por sempre ter a palavra certa nos momentos de fraquezas.

Ao meu pai **Reno**, que tenho certeza, sempre esteve presente em espírito apoiando e guiando meus passos nestes momentos de grandes realizações.

A **minha família**, que esteve ao meu lado dando apoio para esta conquista.

A **Alexandre Carvalho**, diretor da Peggasus, por ter apoiado a realização deste sonho.

Aos **amigos da faculdade**, em especial **Diego, Helvetius, e Maxwell**, que sempre unidos conseguimos vencer várias “batalhas” ao longo do curso.

Ao meu orientador **Romulo Fagundes**, um grande profissional e que incentivou bastante a conclusão deste curso.

Por fim, A **UNI-RN**, por ter me proporcionado este importante passo profissional e acadêmico.

SUMÁRIO

LISTAS DE ILUSTRAÇÕES.....	I
LISTAS DE TABELAS.....	III
LISTA DE ABREVIATURAS E SIGLAS	IV
RESUMO.....	V
ABSTRACT	VI
1. INTRODUÇÃO	1
2. OBJETIVO	2
2.1. Objetivo Geral	2
2.2. Objetivos Específicos.....	2
2.3. Justificativa	2
3. REFERENCIAL TEÓRICO	3
3.1. Banco de Dados.....	3
3.2. Sistema Gerenciador de Banco de Dados (SGBD)	3
3.3. Modelo de Dados.....	5
3.3.1. Hierárquico	6
3.3.2. Em rede.....	6
3.3.3. Relacional	7
3.3.4. Entidade-Relacional	8
3.3.5. Orientado a Objetos	9
3.4. Alta Disponibilidade	9
3.4.1. Tipos de Disponibilidade.....	10
3.4.1.1. Básica.....	10
3.4.1.2. Alta.....	10
3.4.2. Aplicação	10
3.4.3. Tempo de Indisponibilidade	11

3.4.4.	Conceitos	12
3.4.4.1.	Falha	12
3.4.4.2.	Erro.....	12
3.4.4.3.	Defeito.....	13
3.4.4.4.	Failover.....	13
3.4.4.5.	Failback	13
3.5.	Replicação de banco de dados	13
3.5.1.	Fatores de ambiente	14
3.5.2.	Aplicação	14
3.5.3.	Vantagens	15
3.5.4.	Desvantagens.....	16
3.5.5.	Estratégia de Replicação	16
3.5.5.1.	Síncrona.....	16
3.5.5.2.	Assíncrona.....	17
3.5.6.	Modelos de Replicação	17
3.5.6.1.	Mestre-escravo.....	17
3.5.6.2.	Multimestre.....	18
3.6.	História do Microsoft SQL Server	19
3.7.	Arquitetura do Microsoft SQL Server	20
3.7.1.	Páginas e Extensões	20
3.7.2.	Arquivos	22
3.7.3.	Transações.....	23
3.7.4.	Log de transações	23
3.8.	Microsoft SQL Server 2008	24
3.8.1.	Visão Geral.....	24
3.8.2.	Edições	24
3.8.3.	Soluções de Alta Disponibilidade	25

3.8.3.1.	Cluster de Failover	25
3.8.3.2.	Espelhamento de banco de dados	27
3.8.3.3.	Envio de log	28
3.8.3.4.	Replicação	29
3.8.3.4.1.	Componentes de Replicação	30
3.8.3.4.1.1.	Publicador (Publisher)	30
3.8.3.4.1.2.	Distribuidor (Distributor)	30
3.8.3.4.1.3.	Assinante (Subscriber)	30
3.8.3.4.2.	Agentes de Replicação	30
3.8.3.4.2.1.	Snapshot Agent	30
3.8.3.4.2.2.	Distribution Agent	31
3.8.3.4.2.3.	Merge Agent	31
3.8.3.4.2.4.	Log Reader Agent	31
3.8.3.4.2.5.	Queue Reader Agente	31
3.8.3.4.3.	Topologias de Replicação	31
3.8.3.4.3.1.	Publicador central.....	31
3.8.3.4.3.2.	Assinante Central.....	32
3.8.3.4.4.	Variantes de Replicação	33
3.8.3.4.4.1.	Replicação Snapshot.....	33
3.8.3.4.4.2.	Replicação Transacional	34
3.8.3.4.4.3.	Replicação Merge.....	36
4.	CONFIGURANDO UMA REPLICAÇÃO TRANSACIONAL DE BANCO DE DADOS DO MICROSOFT SQL SERVER 2008	38
4.1.	Replicação Transacional	38
4.1.1.	Modelo.....	38
4.1.2.	Controle de alteração.....	38
4.1.3.	Preparando banco de dados para replicação	38

4.1.4.	Configurando um distribuidor	43
4.1.5.	Configurando um Publicador	51
4.1.6.	Configurando um Assinante	61
4.1.7.	Testando a replicação	72
5.	CONCLUSÃO.....	75
6.	BIBLIOGRAFIA	76

LISTAS DE ILUSTRAÇÕES

Figura 1: Representação gráfica de um SGBD.....	5
Figura 2: Modelo Hierárquico	6
Figura 3: Modelo em Rede	7
Figura 4: Modelo Relacional	8
Figura 5: Diagrama entidade-relacionamento	9
Figura 6: Cálculo de disponibilidade de um serviço	11
Figura 7: Replicação mestre-escravo.....	18
Figura 8: Replicação Multimestre	18
Figura 9: Exemplo de uma página de dados.....	21
Figura 10: Exemplo de alocação de extensões	21
Figura 11: Exemplo da arquitetura de arquivos do SQL Server.....	22
Figura 12: Representação do log de transações.....	24
Figura 13: Cluster de dois nós no Windows.....	26
Figura 14: Componentes de um espelhamento de banco de dados	28
Figura 15: Componentes do envio de log.....	29
Figura 16: Topologia publicador central	32
Figura 17: Topologia assinante central.....	33
Figura 18: Replicação Snapshot	34
Figura 19: Replicação Transacional	36
Figura 20: Replicação Merge	37
Figura 21: Conectando na instância S1 do SQL Server	39
Figura 22: Restauração de backup do banco de dados AWTransaction	40
Figura 23: Conectando na instância S2 do SQL Server	41
Figura 24: Criação do banco de dados AWTransaction	42
Figura 25: Configuração de um Distribuidor	43
Figura 26: Configuração inicial do Distribuidor	44
Figura 27: Configuração da instância padrão do Distribuidor	45
Figura 28: Configuração de pasta padrão para snapshot	46
Figura 29: Configuração do banco de dados de distribuição.....	47
Figura 30: Configuração da instância padrão para distribuição	48
Figura 31: Confirmação da configuração do Distribuidor.....	49
Figura 32: Finalização da configuração do Distribuidor	50

Figura 33: Configuração de um Publicador.....	51
Figura 34: Selecionando banco de dados de publicação	52
Figura 35: Selecionando o tipo de publicação.....	53
Figura 36: Selecionando objetos para publicação	54
Figura 37: Filtro para objetos publicados	55
Figura 38: Configuração do Snapshot Agent	56
Figura 39: Configuração de Segurança do Snapshot Agent.....	57
Figura 40: Senhas de acesso para o Snapshot Agent.....	58
Figura 41: Confirmação da configuração do Publicador.....	59
Figura 42: Finalização da configuração do Publicador	60
Figura 43: Configuração de um Assinante	61
Figura 44: Configuração inicial do Assinante	62
Figura 45: Configuração da publicação padrão.....	63
Figura 46: Confiando a publicação padrão	64
Figura 47: Configuração do Agent local de Distribuição.....	65
Figura 48: Configuração da instância e banco de dados do assinante.....	66
Figura 49: Configuração de Segurança do Distribution Agent	67
Figura 50: Senhas de acesso para o Distribution Agent.....	68
Figura 51: Configurando agendamento de sincronização	69
Figura 52: Configurando inicialização da assinatura	70
Figura 53: Confirmação da configuração do Assinante	71
Figura 54: Selecionando registros na tabela Culture	72
Figura 55: Inserindo registros na tabela Culture	73
Figura 56: Confiando replicação de dados.....	74

LISTAS DE TABELAS

Tabela 1: Disponibilidade e Downtime associado	12
Tabela 2: Replicação x Edição disponível.....	37

LISTA DE ABREVIATURAS E SIGLAS

MTBF	Mean Time Between Failures
MTTR	Mean Time To Repair
OLTP	Online Transaction Processing
OLAP	On Line Analytical Processing
SGBD	Sistema de Gerenciamento de Banco de Dados
SGBDR	Sistema de Gerenciamento de Banco de Dados Relacional
ER	Entidade-Relacionamento
DER	Diagrama entidade-relacionamento
SGBDOO	Sistema de Gerenciamento de Banco de Dados Orientados a Objetos

RESUMO

O presente trabalho tem como finalidade o estudo da alta disponibilidade de banco de dados utilizando a ferramenta Microsoft SQL Server 2008 através do uso de replicação de dados. O uso da tecnologia de *alta disponibilidade* é uma premissa para qualquer organização que necessita ter dados disponíveis o maior tempo possível. O uso da replicação de dados utilizando o SQL Server 2008 fornecido pela Microsoft proporciona este tipo de segurança. Utilizando esta ferramenta e implementando a rotina de replicação é possível assegurar o uso constante dos sistemas de informação dessas empresas, fornecendo um ambiente “*on-line*” o maior tempo possível para que estas informações indispensáveis estejam sempre disponíveis.

Palavras-Chave: Banco de Dados, Replicação, Alta Disponibilidade, Informação.

ABSTRACT

The present work aims to study the high availability of database using the Microsoft SQL Server 2008 by using data replication. The use of high-availability technology is a premise for any organization that needs to have data available for as long as possible. Using data replication using SQL Server 2008 provided by Microsoft provides this type of security. Using this tool and implementing replication routine you can ensure the constant use of information systems of these companies by providing an environment online as long as possible so that these essential information is always available.

Keywords: database, replication, high availability, information.

1. INTRODUÇÃO

Armazenar informações é uma necessidade. As pessoas ao longo dos séculos anotavam informações em papiros, madeira, pedras e papel, ou seja, em qualquer coisa que assegurasse que o conhecimento obtido não seria perdido. Quando surgiram os primeiros sistemas de bancos de dados, eles foram projetados para armazenar todos os dados em arquivos simples. Os índices eram construídos sobre o sistema para localizar respostas a perguntas específicas. Estes sistemas de banco de dados evoluíram com o passar do tempo e resultaram em sistemas de banco de dados relacionados, orientados a objetos e objetos-relacionais. (SOLID QUALITY LEARNING, 2006)

Os bancos de dados tem um importante impacto no crescimento do uso de computadores. Pode-se afirmar que os bancos de dados desempenham um papel crítico em quase todas as áreas em que os computadores são usados, que incluem áreas de negócio, comércio eletrônico, engenharia, medicina, etc.. (ELMASRI; NAVATHE, 2010)

De acordo com ELMASRI e NAVATHE (2010), banco de dados e sistema de banco de dados são um componente essencial da vida na sociedade moderna. Para eles diariamente diversas atividades envolvem um banco de dados. Por exemplo, ir a um banco para retirada de dinheiro, reservas em hotéis, comprar através de um site de comércio eletrônico, entre outras atividades. Provavelmente estas atividades envolvem alguém ou programa de computador que acessa um banco de dados.

Tendo em vista toda essa importância e dependência da informação contida em um banco de dados se faz necessário mantê-lo disponível o maior tempo possível. Para esta necessidade surge o conceito de alta disponibilidade, que busca manter estas informações disponíveis o maior tempo possível para com isso diminuir o impacto causado pela paralização dos sistemas de informação.

2. OBJETIVO

2.1. Objetivo Geral

Visto que os dados para as empresas são de suma importância e precisam estar disponíveis o maior tempo possível, o objetivo deste trabalho é o estudo da alta disponibilidade de banco de dados, utilizando a tecnologia de replicação oferecida pela Microsoft através do SQL Server 2008, bem como suas vantagens e desvantagens.

2.2. Objetivos Específicos

- Estudo da alta disponibilidade, com ênfase na replicação de banco de dados;
- Vantagens e desvantagens de se utilizar replicação de dados;
- Configurar uma solução de replicação utilizando a ferramenta Microsoft SQL Server 2008.

2.3. Justificativa

A justificativa para este trabalho parte da necessidade de empresas estarem cada vez mais necessitando de sistemas de informação disponíveis em tempo integral. Hoje a dependência tecnológica para prestação de serviços é uma premissa para qualquer empresa. Algumas empresas precisam estar com seus serviços disponíveis 24 horas, sete dias por semana. Desta forma, este trabalho busca demonstrar umas das possíveis soluções para a alta disponibilidade de banco de dados através de replicação utilizando a ferramenta SQL Server 2008.

3. REFERENCIAL TEÓRICO

3.1. Banco de Dados

O Banco de dados pode, ele próprio, ser visto como o equivalente eletrônico de um armário de arquivamento. Em outras palavras, é um repositório ou recipiente para uma coleção de arquivos de dados computadorizados. (DATES, 2000)

Para DATES (2000), um banco de dados é uma coleção de dados persistentes utilizadas pelo sistema de aplicações de uma determinada empresa.

Em geral, o gerenciamento eficiente de dados exige a utilização de um banco de dados computacional. Um banco de dados é uma estrutura computacional compartilhada e integrada que armazena um conjunto de dados do usuário final ou *metadados*, dados sobre dados, pelos quais os dados do usuário final são integrados e gerenciados. (ROB; CORONEL, 2011)

Em resumo, existe a necessidade do armazenamento de dados. Entretanto, o formato e o propósito desses dados irão diferir dependendo do negócio específico, situação atual, o desempenho passado e como ele será exposto no futuro. (SOLID QUALITY LEARNING, 2006)

3.2. Sistema Gerenciador de Banco de Dados (SGBD)

No passado, o mecanismo de banco de dados armazenava os dados enquanto o sistema de informação do usuário era responsável por tarefas adicionais como: Ler os dados a partir do disco, Autorizar ou negar acesso aos dados, Verificar dados contra as regras de negócio para assegurar qualidade nos dados, etc., mas não era possível garantir que os dados sempre seriam gerenciados pelo sistema de informação do usuário, levando, em muitas vezes, a corrupção dos dados. Por essa razão, se faz necessário o uso de um aplicativo servidor que possui a gerência dos dados e realiza estas e outras tarefas necessárias para manter um banco de dados íntegro. (SOLID QUALITY LEARNING, 2006)

Entre o banco de dados físico e os usuários do sistema há uma camada de software, conhecida como sistema gerenciador do banco de dados ou servidor de banco de dados ou ainda, com maior frequência, sistema de gerenciamento de banco de dados (SGBD), conforme representado na figura 1. Os sistemas de banco de dados estão disponíveis em máquinas que vão desde um computador de pequeno porte (dispositivos móveis) até máquinas de grande porte. (DATES, 2000)

Todas as solicitações feitas ao banco de dados são tratadas pelo SGBD, por exemplo, buscar, atualizar, remover ou inserir dados em tabelas ou arquivos. Este tipo de aplicação nada mais é do que uma coleção de dados inter-relacionados e uma coleção de programas para prover o acesso a base de dados. O objetivo principal de um SGBD é garantir um ambiente adequado e eficiente para uso na recuperação e armazenamento de informações. (DATES, 2000)

Algumas vantagens que tornam o gerenciamento de dados utilizando um SGBD mais eficiente e eficaz são:

- **Aprimoramento e compartilhamento de dados:** Cria um ambiente onde os usuários finais possuem um melhor acesso aos dados, em maior quantidade e bem gerenciados.
- **Aprimoramento da segurança de dados:** Quanto mais usuários acessando os dados, maiores são os riscos de falhas de segurança. O SGBD fornece um modelo melhor aplicado a políticas de privacidade e segurança de dados.
- **Melhoria na integração de dados:** Acesso mais amplo aos dados bem gerenciados promove uma visualização mais ampla do panorama geral.
- **Minimização da inconsistência de dados:** Problema reduzido por um adequado projeto de banco de dados e um bom gerenciamento dos dados.
- **Aprimoramento de acesso aos dados:** O SGBD torna possível obter respostas rápidas as consultas ou manipulação dos dados.
- **Aprimoramento na tomada de decisão:** Dados bem gerenciados e com acesso aprimorado geram melhor resultados para tomadas de decisões mais rápidas e confiáveis.

- **Aumento de produtividade do usuário final:** Os dados disponíveis, aliados a uma ferramenta que os transformam em informações úteis, capacitam o usuário final a tomar decisões rápidas e confiáveis.



Figura 1: Representação gráfica de um SGBD

Fonte: BORGES, 2012

3.3. Modelo de Dados

Para ROB e CORONEL (2011), um modelo de dados é uma representação relativamente simples, normalmente gráfica, de estruturas de dados reais mais complexas. De uma forma geral, modelo é abstração de um objeto ou evento real de mais complexidade. No ambiente de banco de dados, um modelo representa estruturas de dados e suas características, relações, restrições, transformações e outros elementos que tenham por fim dar suporte ao problema específico.

3.3.1. Hierárquico

Desenvolvido na década de 60 para gerenciamento de grandes quantidades de dados para projetos complexo de fabricação. É uma estrutura hierárquica que contém níveis, também chamados de seguimentos, que é o equivalente ao tipo de registro em um sistema de arquivo, porém, o modelo hierárquico resulta em diversas vantagens em relação ao sistema de arquivos. No interior da hierarquia, a camada superior é vista como “pai” do segmento “filho” logo abaixo dela. Em resumo, este modelo representa um conjunto de relacionamento de um para muitos entre pai e seus segmentos filhos, como representado na figura 2. Neste modelo cada pai pode ter muitos filhos, porém, cada filho possui apenas um pai. Os modelos de dados atuais foram constituídos a partir do modelo hierárquico. Este modelo foi muito utilizado na década de 70, porém, apresentava algumas limitações, entre elas a dificuldade de programar e gerenciar e a não disposição de independência estrutural. (ROB; CORONEL, 2011)

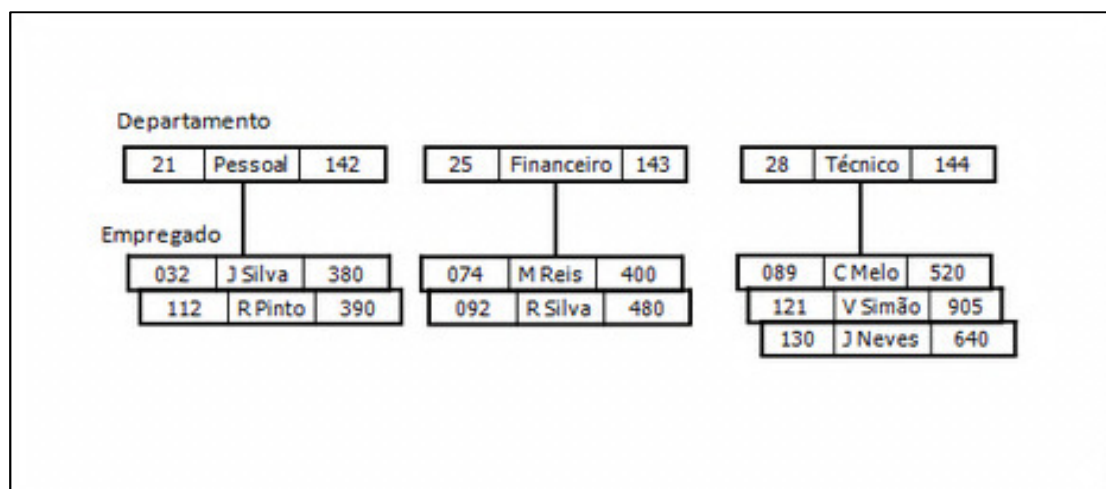


Figura 2: Modelo Hierárquico

Fonte: CLARINDO, 2012

3.3.2. Em rede

Foi criado para representar relacionamentos de dados complexos com mais eficiência do que o modelo hierárquico, além de, melhorar o desempenho dos bancos de dados e impor um padrão a eles. No modelo em rede é possível perceber o banco de dados em rede como uma coleção de registros em relacionamentos um para muitos.

Contudo, diferentemente do hierárquico, este modelo permite que um registro possua mais de um pai, como podemos observar na figura 3. (ROB; CORONEL, 2011)

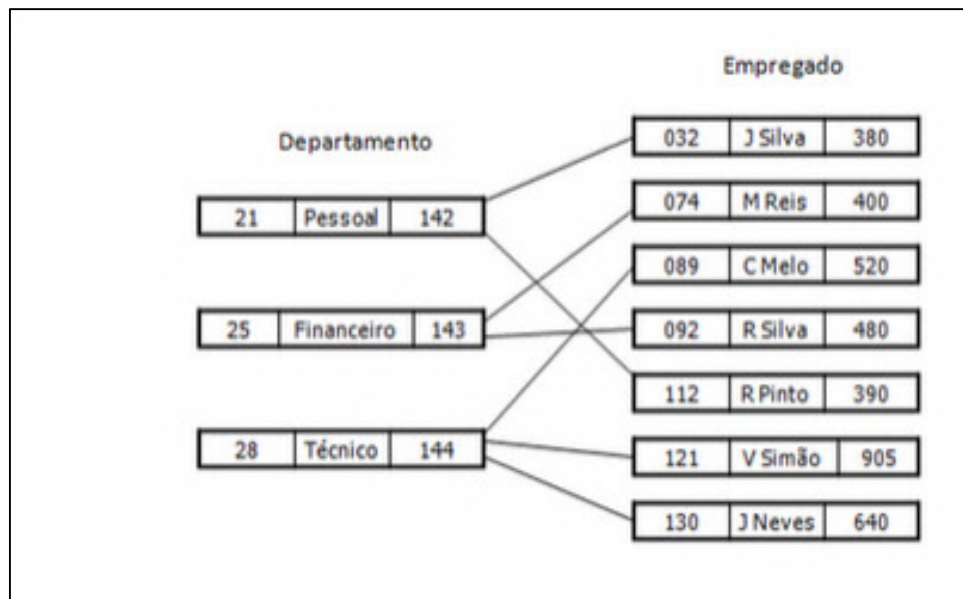


Figura 3: Modelo em Rede

Fonte: CLARINDO, 2012

3.3.3. Relacional

Foi apresentado na década de 70 através de um artigo “A Relational Model Data for Large Shared Databanks” escrito por E. F. Codd cientista da IBM. A partir deste modelo houve uma revolução conceitual nos bancos de dados. O fundamento básico deste modelo é um conceito matemático conhecido como relação, também chamado tabela, como uma matriz composta de interseções de linhas e colunas. Cada linha de uma relação é chamada de *tupla*. Cada coluna representa um *atributo*. O grande problema deste modelo para a época foi a não capacidades das maquinas em suportar a implementação deste momento. Com a evolução dos computadores, e claro, com o crescimento exponencial de hardware e de software este cenário mudou completamente. (ROB; CORONEL, 2011)

Conforme representado na figura 4, neste modelo as tabelas são relacionadas umas com as outras por meio de um atributo comum entre elas. Na figura 4 temos o campo *Dept* na tabela *Empregado* que faz um relacionamento com o campo *Dept* da

tabela *Departamento*. O tipo de relacionamento pode ser um para um, um para muitos e muitos para muitos. (ROB; CORONEL, 2011)

Neste modelo temos também o SGBDR (Sistema gerenciador de banco de dados relacional), que executa as mesmas funções de um SGBD para modelos hierárquicos e em rede, mas, além disso, abriga outras funções que tornam o modelo relacional mais fácil de compreender e implantar. É possível dizer que uma das vantagens do SGBDR é a capacidade de ocultar do usuário as complexidades do modelo relacional. Ele faz toda gerência dos detalhes físicos, enquanto o usuário vê o banco de dados como uma coleção de tabelas onde os dados são armazenados. (ROB; CORONEL, 2011)

Empregado			
NumEmp	NomeEmp	Salário	Dept
032	J Silva	380	21
074	M Reis	400	25
089	C Melo	520	28
092	R Silva	480	25
112	R Pinto	390	21
121	V Simão	905	28
130	J Neves	640	28

Departamento		
NumDept	NomeDept	Ramal
21	Pessoal	142
25	Financeiro	143
28	Técnico	144

Figura 4: Modelo Relacional

Fonte: CLARINDO, 2012

3.3.4. Entidade-Relacional

A simplicidade utilizada na tecnologia de banco de dados relacional incentivou a utilização de SGBDR. Por sua vez, esta demanda fez crescer rapidamente a exigência de transações e informações mais complexas. Embora o modelo relacional fosse um amplo aprimoramento aos modelos anteriores, ainda carecia de recursos que o tornasse

uma ferramenta de projeto eficiente. Assim, na década de 70, Peter Chen, cientista americano, apresentou o modelo Entidade-Relacional, representado na figura 5. Tratava-se de uma representação gráfica de entidades e seus relacionamentos na estrutura do banco de dados. Os modelos de ER são geralmente apresentados em um DER. (ROB; CORONEL, 2011)

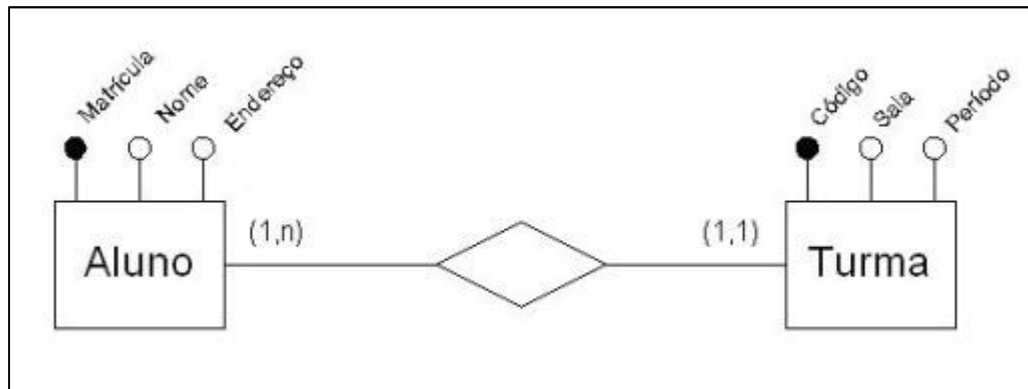


Figura 5: Diagrama entidade-relacionamento

Fonte: PENHA, 2011

3.3.5. Orientado a Objetos

Problemas reais cada vez mais complexos exigiam um modelo que se assemelhasse ao mundo real. No modelo de dados orientado a objeto, os dados e seus relacionamentos são contidos em uma estrutura chamada objeto. Este modelo é a base para o SGBDOO (Sistema gerenciador de banco de dados orientado a objetos). (ROB; CORONEL, 2011)

3.4. Alta Disponibilidade

Alta disponibilidade não é apenas um produto ou aplicação que se instale, na verdade é uma característica de um sistema computacional. Existem diversos mecanismos e técnicas que são utilizados para aumentar e manter a disponibilidade de um sistema. A simples utilização dessas técnicas não garante esta disponibilidade se não for acompanhado de um completo estudo e projeto de configuração. (SZTOLTZ; TEIXEIRA; RIBEIRO, 2003)

Disponibilidade é definida como a propriedade de um sistema estar pronto para ser usado imediatamente. Em geral, refere-se a probabilidade de o sistema estar funcionando corretamente em qualquer momento determinado e estar disponível para executar suas funções. Em outras palavras, um sistema de alta disponibilidade é aquele que mais provavelmente estará funcionando em dado instante no tempo. (TANENBAUM; STEEN, 2006, p. 194)

3.4.1. Tipos de Disponibilidade

3.4.1.1. Básica

Este tipo de disponibilidade normalmente é encontrado em máquinas comuns, sem nenhum mecanismo especial, de software ou hardware, que vise, de alguma forma sustentar eventuais falhas dessas máquinas. Neste tipo de disponibilidade costuma-se apresentar uma disponibilidade de 99% a 99,99%, em outras palavras, isso quer dizer que em um ano de operação a máquina pode ficar indisponível por um período de 9 horas há quatro dias. (SZTOLTZ; TEIXEIRA; RIBEIRO, 2003)

3.4.1.2. Alta

Adicionando mecanismos especializados de detecção, recuperação e mascaramento de falhas, pode-se aumentar a disponibilidade de um sistema, desta forma, este se enquadrará em uma classe de alta disponibilidade. Neste tipo de disponibilidade costuma-se apresentar uma disponibilidade de 99,99% a 99,999%, ou seja, podendo ficar indisponível por um período de pouco mais de 5 minutos até 1 hora em um ano de operação. Neste tipo se encaixam a grande massa de aplicações comerciais de alta disponibilidade. (SZTOLTZ; TEIXEIRA; RIBEIRO, 2003)

3.4.2. Aplicação

A principal aplicação da Alta Disponibilidade é buscar uma forma de manter os serviços prestados por um sistema de informação, mesmo que este sistema venha a sofrer modificações internas por causa de falhas. Neste ponto fica implícito o mecanismo utilizado para o mascaramento de falhas, seja através de redundância ou replicação de dados. Em outras palavras, um determinado serviço, que se quer

altamente disponível, é colocado por trás de uma camada de abstração, que permite mudanças em seus mecanismos internos mantendo intacta a interação com elementos externos. Uma subárea de tolerância a falhas, que visa manter a disponibilidade dos serviços prestados por um sistema de informação, através de redundância de hardware e reconfiguração de software. Vários computadores agindo como um só, monitorando uns aos outros e assumindo seus serviços no caso de algum deles falhar. (SZTOLTZ; TEIXEIRA; RIBEIRO, 2003)

3.4.3. Tempo de Indisponibilidade

Em um sistema real, se um computador apresenta falhas, ele é reparado ou substituído por um novo componente. Se este novo falha, é substituído ou reparado também por outro e assim por diante. Durante sua vida útil um componente pode ter um estado *funcionando* e *em reparo*. O estado *funcionando* significa que ele está em operação e *em reparo* significa que ele parou e ainda não foi substituído por um novo componente. No caso de um sistema, ele vai de *funcionando* para *em reparo* e quando substituído ele retorna para o estado *funcionando*, logo, pode-se dizer que o sistema apresenta ao longo de sua vida um tempo médio até apresentar falha (MTBF) e um tempo médio de reparo (MTTR). De forma simplificada, como vemos na figura 6, diz-se que a disponibilidade de um sistema é a relação entre o tempo de vida útil deste sistema e seu tempo total de vida. (SZTOLTZ; TEIXEIRA; RIBEIRO, 2003)

$$\text{Disponibilidade} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

Figura 6: Cálculo de disponibilidade de um serviço

Fonte: NATÁRIO, 2011

A tabela 1 ilustra um comparativo normalmente utilizado na avaliação de soluções de alta disponibilidade através da demonstração do *Downtime*, que é o tempo de indisponibilidade de um serviço ou equipamento. Ao de tratar de *Downtime*, podem existir duas partes envolvidas, os *planejados* e os *não planejados*. Os *não planejados* são falhas não controladas decorrentes de hardware, software ou falta de energia. Já os

planejados são normalmente períodos que chamamos de janelas, onde será realizado procedimento de manutenção agendada, podendo ser um reparo, atualização de componentes ou instalação de novos equipamentos. Os *planejados* estão excluídos desta tabela, pois os tempos de *Downtime* estimados para este tipo de situação são alheios às soluções e diferentes variáveis.

Disponibilidade (%)	<i>Downtime</i>/ano	<i>Downtime</i>/mês
95%	18 dias 6:00:00	1 dias 12:00:00
96%	14 dias 14:24:00	1 dias 4:48:00
97%	10 dias 22:48:00	0 dias 21:36:00
98%	7 dias 7:12:00	0 dias 14:24:00
99%	3 dias 15:36:00	0 dias 7:12:00
99,9%	0 dias 8:45:35.99	0 dias 0:43:11.99
99,99%	0 dias 0:52:33.60	0 dias 0:04:19.20
99,999%	0 dias 0:05:15.36	0 dias 0:00:25.92

Tabela 1: Disponibilidade e Downtime associado

Fonte: INÁCIO (2011)

3.4.4. Conceitos

3.4.4.1. Falha

Uma falha acontece no universo físico, ou seja, em nível de hardware, por exemplo, uma interferência eletromagnética é considerada uma falha, pois acontece no universo físico e afeta o funcionamento de um computador ou partes dele. (SZTOLTZ; TEIXEIRA; RIBEIRO, 2003)

3.4.4.2. Erro

A ocorrência de uma falha pode acarretar um erro, que representa uma falha, por exemplo, um computador lê bits, cada bit contém um 0 ou um 1. Uma falha

pode fazer com que um ou mais bit troque de valor inesperadamente, o que certamente afetará o funcionamento normal do computador. Uma falha, portanto, pode gerar um erro em alguma informação. (SZTOLTZ; TEIXEIRA; RIBEIRO, 2003)

3.4.4.3. Defeito

Uma falha, se não percebida e tratada, pode gerar um defeito. O sistema simplesmente trava, mostra uma mensagem de erro, ou ainda perde os dados do usuário sem maiores avisos. Isto é percebido no universo do usuário. (SZTOLTZ; TEIXEIRA; RIBEIRO, 2003)

3.4.4.4. Failover

É o processo onde uma máquina assume os serviços de outra em caso de uma falha. Este pode ser automático ou manual, porém, é claro, sendo o automático o que se espera de uma solução de alta disponibilidade. Mesmo assim, alguns serviços não críticos podem suportar um tempo maior de recuperação e podem utilizar o modelo manual.

3.4.4.5. Failback

Após manutenção da máquina defeituosa se faz necessária à recolocação desta em serviço, logo, é executado um processo inverso ao failover, o failback. (SZTOLTZ; TEIXEIRA; RIBEIRO, 2003)

3.5. Replicação de banco de dados

Replicação de banco de dados é o processo de manter duas ou mais réplicas dos dados em diferentes instancias de um servidor de banco de dados. (BATTISTI, 2005)

Para RAMOS (2009), replicação é um meio de se copiar (replicar) de forma gerenciada os dados entre servidores, que podem estar próximos ou a centenas de quilômetros de distância.

3.5.1. Fatores de ambiente

No planejamento para implantação de um ambiente de replicação devem ser considerados alguns fatores, tais como:

- **Latência:** Tempo de espera para que dois conjuntos de dados sejam sincronizados.
- **Autonomia de réplica:** Este fator leva em consideração o tempo que um servidor que está em réplica pode continuar em funcionamento, ou seja, caso um servidor perca contato com os demais servidores devido a uma parada da internet. O servidor continuará funcionando sem problemas, porém, quando a conexão for restabelecida será necessário sincronizar os dados, isso é feito conforme uma programação.
- **Conflitos de atualizações:** Quando os dados podem ser alterados em diversos locais se faz necessário um mecanismo de solução de conflito de replicação.
- **Velocidade de link:** Talvez o fator mais importante, pois replicação implica em informações enviadas pela rede.

3.5.2. Aplicação

A aplicação mais óbvia de uma replicação é a possibilidade de se manter, em diferentes servidores, cópias fiéis de um conjunto de dados. (BATTISTI, 2005)

Com base nesta afirmação, é possível dizer que a replicação deve ser considerada uma solução em situações que:

- Seja necessário que diferentes locais tenham a disposição uma cópia atualizada de um banco de dados. Por exemplo, a matriz de uma empresa enviando catálogo de produtos para as suas filiais.

- Seja necessário manter servidores separados, por questão de desempenho, duas cópias idênticas do mesmo banco de dados. Por exemplo, utilização de um banco de dados OLTP e um OLAP. Um ambiente OLTP onde são realizados basicamente entrada de dados ou consulta simples e um ambiente OLAP onde são realizadas consultas complexas que exigem maior desempenho do servidor.
- Sempre seja necessário copiar e distribuir dados para diferentes localidades.

3.5.3. Vantagens

De acordo com MELO (2010), estas são algumas vantagens que a replicação de banco de dados nos traz:

- **Disponibilidade:** Considerando um conjunto de objetos, caso um deles se torne indisponível, o outro pode ser acessado mantendo a continuidade da operação que estava sendo realizada;
- **Desempenho:** Partindo do princípio que vários objetos são replicados, uma aplicação que necessita desse objeto pode escolher, dentre os objetos disponíveis, o que lhe proporciona um tempo de acesso menor, aumentando assim o desempenho da aplicação. Além disso, como existem vários objetos disponíveis, caso sejam necessários vários acessos à esses objetos, esses acessos podem ser divididos entre as diferentes réplicas evitando a sobrecarga em algum objeto específico e possibilitando um maior paralelismo na execução de operações;
- **Redução de custos:** É possível na replicação, em determinadas circunstâncias, propiciar uma redução de custos, por exemplo, com conectividade, pois se pode utilizar soluções de conectividade mais baratas mantendo localmente objetos replicados nos pontos de acesso e configurando a replicação para ocorrer em momentos de menor tráfego;
- **Redundância da informação:** As informações estão preservadas nas demais réplicas, no caso de falhas permanentes nos objetos replicados, visto que eles são replicados entre si;

3.5.4. Desvantagens

Porém, segundo MELO (2010), a replicação de dados nos traz algumas desvantagens, onde podemos citar:

- **Redundância nas operações:** As operações realizadas em um objeto necessitam ser realizadas novamente nas réplicas;
- **Maior espaço para armazenamento e memória:** Estando vários objetos replicados entre si, em cada local que esses objetos estejam, será necessário reservar espaço para os mesmos, além disso, se todos os objetos estiverem recebendo informação e replicando para os demais, cada objeto acumulará as informações de todos os objetos juntos;
- **Complexidade no controle de inconsistências:** Em diferentes modelos de replicação, podem ocorrer inconsistências e a resolução das mesmas aumenta a complexidade na gestão dos objetos envolvidos na replicação;

3.5.5. Estratégia de Replicação

3.5.5.1. Síncrona

Com replicação síncrona, todas as cópias dos dados são mantidas exatamente sincronizadas e consistentes. Se uma cópia é atualizada as mudanças serão imediatamente aplicadas para todas as outras bases de dados com a mesma transação que fazem parte do grupo de replicação. (HOPKINS; THOMAS, 1998)

Para MELO (2010), os sistemas mais propícios a trabalhar com a replicação síncrona são aqueles que possuem os seguintes requisitos:

- Exige alta consistência entre réplicas;
- Exige latência baixa na propagação das informações;
- As operações de leitura dos objetos predominam sobre as operações de escrita;
- Possuem um número reduzido de objetos replicados;

- Os meios de comunicação entre os objetos são confiáveis e de alto desempenho;
- Não necessitam de alto desempenho na utilização dos objetos

3.5.5.2. Assíncrona

Com replicação assíncrona as cópias dos dados ficarão temporariamente dessincronizadas. Se uma cópia é atualizada, a mudança será aplicada e propagada para as outras cópias num segundo passo, em uma transação separada que pode ocorrer em segundos, horas ou dias mais tarde. As cópias por outro lado podem ficar temporariamente sem sincronia, mas a certa altura os dados irão convergir para um mesmo valor em todas as cópias. (HOPKINS; THOMAS, 1998)

Esta estratégia de replicação é utilizada em situações que não há necessidade de se obter o dado totalmente atualizado e que são pouco propícios a conflitos entres transações executadas em réplicas distintas, mesmo assim, pode haver a necessidade de solucionar possíveis conflitos na atualização. (MELO 2010)

3.5.6. Modelos de Replicação

3.5.6.1. Mestre-escravo

De acordo com MELO (2010), Na replicação mestre-escravo, um dos objetos é considerado como mestre e as alterações só podem ser feitas nesse objeto. Os demais objetos são considerados como escravos e só podem ser utilizados para leitura. O fluxo das informações sempre se dá do objeto mestre para os objetos escravos, vemos este fluxo de operação exemplificado na figura 7, se configurando como uma arquitetura de menor complexidade. Eventualmente pode-se ter nesse tipo de modelo o cascadeamento entre objetos escravos, isto é, um objeto escravo que recebe as modificações do objeto mestre pode repassar essas modificações para outros objetos escravos.

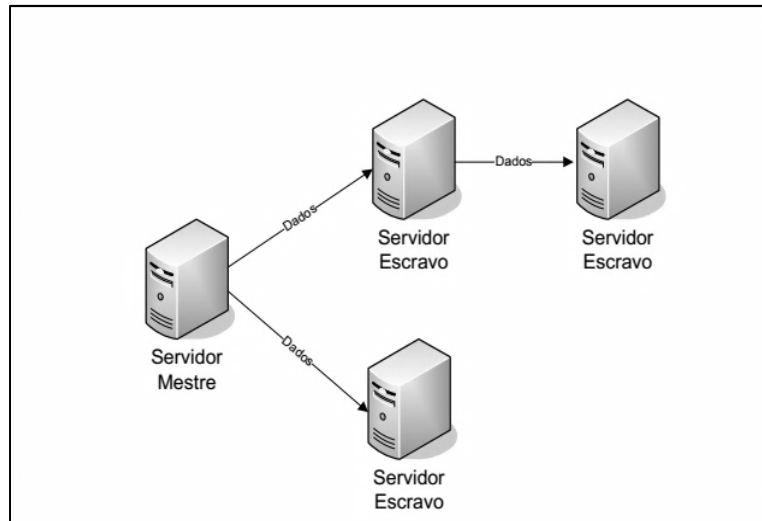


Figura 7: Replicação mestre-escravo

Fonte: MELO, 2010

3.5.6.2. Multimestre

Segundo MELO (2010), Na replicação multimestre, todos os objetos podem ser modificados e essas modificações são transmitidas para os demais objetos. Neste modelo de replicação o fluxo das informações pode partir de qualquer objeto participante para os demais, conforme vemos na figura 8, e por isso a complexidade de controle da replicação é maior.

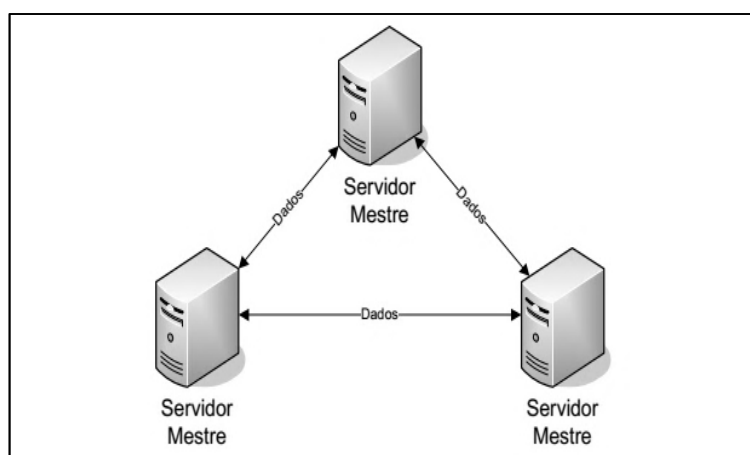


Figura 8: Replicação Multimestre

Fonte: MELO, 2010

3.6. História do Microsoft SQL Server

Originalmente, o Microsoft SQL Server surgiu de uma parceria entre as empresas Microsoft e Sybase, empresa fundada por Mark Hoffman e Bob Epstein em 1984, em Berkeley, na Califórnia. Em 1987 a Sybase produzia o Data Server para sistemas operacionais OS/2, sistema operacional da IBM. Então, em 27 de Março deste ano a Microsoft assinou um contrato onde obteria direitos do Data Server. Para ganhar a aceitação do mercado o “novo” sistema gerenciador de banco de dados da Microsoft (licenciado pela Sybase) precisava ganhar adeptos da comunidade dBase, primeiro SGBD utilizado largamente por empresas, então, outro acordo foi realizado junto a Ashton-Tate, empresa que desenvolveu o dBase, assim, a primeira versão do Microsoft SQL Server se chamava Ashton-Tate/Microsoft SQL Server e chegou ao mercado na metade de 1988. Em maio do ano seguinte a versão 1.0 para OS/2 era lançada com participação da Microsoft, limitando-se apenas a poucas ferramentas, testes e um projeto visando tornar o aplicativo mais simples de ser instalado. Na segunda metade de 1990 a união da Microsoft com a Ashton-Tate se encerrou e a versão 1.1 do SQL Server passou a oferecer suporte para o Windows 3.0. Apesar disso, a base do SQL Server era produzida pela Sybase e a Microsoft não tinha acesso ao código-fonte. Qualquer defeito deveria ser relatado para a Sybase e corrigido apenas por ela. Em 1991 a Microsoft passou a poder acessar o código fonte do SQL Server apenas para leitura. Com o lançamento da versão 4.2 para OS/2 no fim de 1991 a Microsoft já desenvolvia o banco de dados em conjunto com a Sybase, o que só mudou em agosto de 1993 quando a Microsoft finalmente conseguiu a totalidade da criação do SQL Server e lançou a versão para o Windows NT 3.1. Essa nova versão já era em 32-bits. (MSDN, 2008)

Então, pouco tempo depois, a parceria entre as duas empresas acabava. O primeiro beta do SQL Server 6.0, também chamado de Microsoft SQL Server 95, surgiu no final de outubro de 1994, e em julho de 1995 era lançado oficialmente o Microsoft SQL Server 6.0. Em dezembro de 1995 é liberada uma versão beta completa do SQL Server 6.5. O SQL Server 7.0, codinome Sphinx, chegou ao mercado no início de 1999 e colocou o SQL Server no mercado de sistemas de gerenciamento de base de dados relacionais de empresas. Em setembro de 1999 é distribuído o Beta 1 do SQL Server 2000 (versão 8.0 de codinome Shiloh), o Beta 2 foi lançado em abril de 2000 e em agosto

a versão final do Microsoft SQL Server 2000. Em novembro de 2005 é lançado ao público o SQL Server 2005 (codinome Yukon). A versão final do Microsoft SQL Server 2008, codinome Katmai, foi lançada em agosto de 2008 e em 1º de abril de 2012 foi lançada a versão 2012, codinome Denali. Esta última sendo a versão mais atual do SGBD da Microsoft. (MSDN, 2008)

3.7. Arquitetura do Microsoft SQL Server

3.7.1. Páginas e Extensões

Uma página é a unidade fundamental de armazenamento de dados no SQL Server. Uma extensão consiste em uma coleção de páginas fisicamente ligadas. Estas extensões ajudam a gerenciar as páginas de maneira eficaz. (MSDN, 2008)

O espaço em disco alocado a um arquivo de dados em um banco de dados é logicamente dividido em páginas numeradas de forma contígua de 0 a n. As operações de E/S de disco são executadas no nível de página, ou seja, o SQL Server lê ou grava páginas de dados inteiras. (MSDN, 2008)

No SQL Server uma página possui 8K. Isso significa dizer que o banco de dados SQL Server possui 128 páginas por megabyte. Cada página é iniciada por um cabeçalho de 96 bytes usados para armazenamento de informações de sistema sobre a página. (MSDN, 2008)

As linhas de dados são inseridas em sequência na página, logo após o cabeçalho de página. Uma tabela de deslocamento da linha tem início no final da página, e cada tabela de deslocamento da linha é referenciada a uma entrada para cada linha na página. Podemos ver este tratamento de dados na figura 9. (MSDN, 2008)

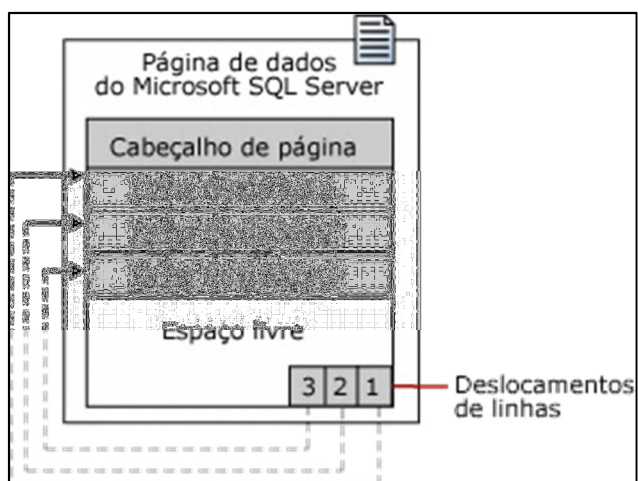


Figura 9: Exemplo de uma página de dados

Fonte: MSDN, 2008

Uma extensão tem oito páginas fisicamente contíguas ou 64 KB. Isso significa que os bancos de dados SQL Server têm 16 extensões por megabyte. (MSDN, 2008)

Para tornar o espaço em disco eficiente, o SQL Server não aloca extensões inteiras para tabelas com quantidade pequenas de dados. Ele as divide em dois tipos de extensões:

- **Uniforme:** Estas pertencem a um único objeto. Todas as oito páginas são utilizadas pelo objeto proprietário.
- **Mistas:** São compartilhadas por até oito objetos e cada uma das oito páginas podem pertencer a um objeto diferente.

Vemos a representação deste tratamento de alocações na figura 10.

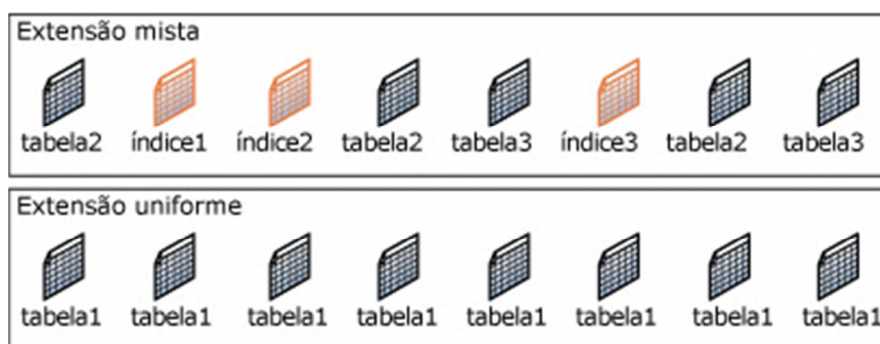


Figura 10: Exemplo de alocação de extensões

Fonte: MSDN, 2008

3.7.2. Arquivos

O SQL Server mapeia um banco de dados de um conjunto de arquivos do sistema operacional. As informações de log e dados nunca ficam embaralhados no mesmo arquivo, e os arquivos individualmente são usados apenas por um banco de dados. Os grupos de arquivos são conhecidos como coleções de arquivos e são usados para ajudar no posicionamento de dados e em tarefas administrativas, como operações de backup e restauração, por exemplo. (MSDN, 2008)

Bancos de dados SQL Server possuem três tipos de arquivos, representados graficamente na figura 11 e descritos abaixo, são eles:

- **Arquivos de dados primários:** Este é o ponto de partida do banco de dados e aponta para outros arquivos do conjunto. Necessariamente, todo banco de dados SQL Server possui um arquivo de dados primário. Normalmente é utilizada a extensão *.mdf para este tipo de arquivo.
- **Arquivo de dados secundários:** São arquivos que compõe o conjunto de dados. É possível existir bancos de dados SQL Server sem arquivos secundários, porém alguns bancos possuem vários arquivos secundários normalmente escritos com a extensão *.ndf.
- **Arquivo de log:** Neste arquivo estão todas as informações de log utilizados para recuperar um banco de dados. É necessário haver pelo menos um arquivo de log para cada banco de dados, embora possa existir mais de um. São normalmente escritos com a extensão *.ldf.

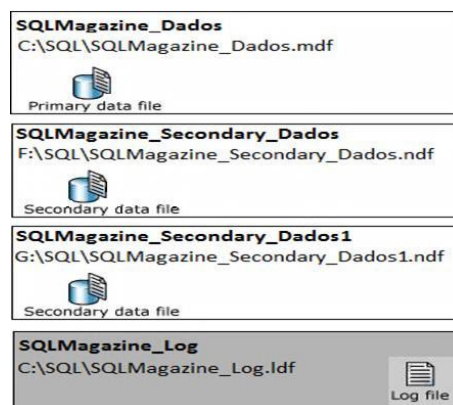


Figura 11: Exemplo da arquitetura de arquivos do SQL Server

Fonte: MSDN, 2008

3.7.3. Transações

De uma forma explícita, uma transação é qualquer operação de escrita em uma tabela de um banco de dados. Executar uma instrução UPDATE é uma transação por si só, assim como da mesma forma executar um DELETE ou um INSERT. Esse trio de comandos efetivamente altera o conteúdo dos dados dentro de uma tabela através de transações. (REIS, 2003)

3.7.4. Log de transações

No momento que uma transação é iniciada e as operações de INSERT, DELETE ou UPDATE começam a ser enviadas para o banco de dados, exemplificado na figura 12 através do comando INSERT, algumas premissas serão executadas em um servidor SQL Server.

- As páginas que contém os dados que serão afetados são localizadas no disco e carregadas em memória, também chamada de chave de dados. Sempre, antes de carregar uma página de disco, o SQL Server verifica se esta página já está no cache, caso contrário, fará a busca em disco.
- As alterações, exclusões, inclusões nunca são afetadas diretamente no disco, mas sim nas páginas carregadas em memória. Estas modificações nos dados são registradas primeiramente no arquivo de log do banco de dados.
- Em seguida um processo chamado checkpoint faz com que as transações que tenham sido completadas com sucesso no log sejam aplicadas ao banco de dados, ou seja, gravado em disco.

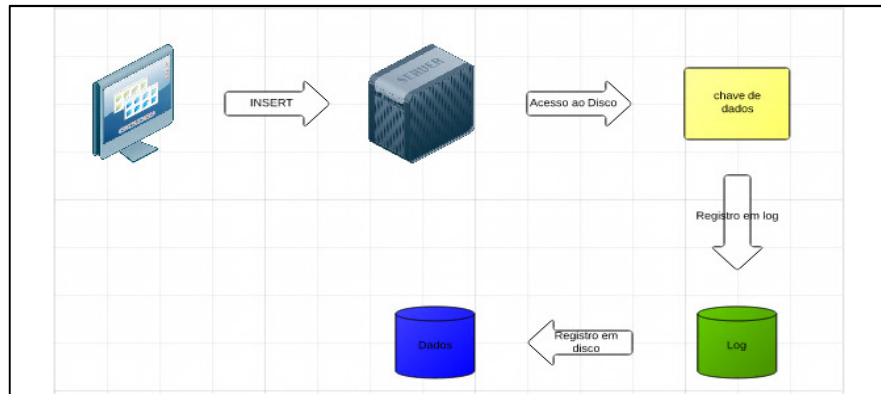


Figura 12: Representação do log de transações

Fonte: Resultado de pesquisa

3.8. Microsoft SQL Server 2008

3.8.1. Visão Geral

O SQL Server 2008 é um SGBDR – sistema gerenciado de banco de dados relacional – e está disponível em várias edições, desde edições projetadas para aplicações móveis até edições com suporte a petabytes de informações ou dados manipulados por milhões de usuários. (HOTEK, 2009)

3.8.2. Edições

- **Enterprise:** Projetada para grandes empresas que precisam aproveitar o máximo da plataforma SQL Server 2008.
- **Standard:** Projetada para empresas de pequeno e médio porte e que não precisam de todos os recursos disponíveis.
- **Workgroup:** Indicado para projetos departamentais, pois possui um conjunto limitado de recursos.
- **Express:** Versão gratuita projetada para atender necessidades básicas de armazenamento.
- **Compact:** Projetada para uso em dispositivos móveis.
- **Developer:** Possui toda capacidade da edição Enterprise, porém, possui limitações de ambiente de desenvolvimento.

- **Evaluation:** Versão de avaliação do com validade de 180 dias. Possui todas as funcionalidades da edição Enterprise.

3.8.3. Soluções de Alta Disponibilidade

O Microsoft SQL Server 2008 oferece várias soluções para ambientes de alta disponibilidade para um servidor de banco de dados. Dentre as opções temos o cluster de failover, espelhamento de banco de dados, envio de logs e replicação.

3.8.3.1. Cluster de Failover

Esta solução é baseada no cluster do Microsoft Windows Server, que é um grupo de computadores independentes trabalhando em conjunto para aumentar a disponibilidades de aplicativos e serviços. A base para a construção de um cluster failover do SQL Server é o cluster do Windows Server, (MSDN, 2008)

Neste tipo de solução, os aplicativos são instalados em um grupo de cluster, conhecidos como grupo de recursos ou recursos de cluster, que consiste em tudo que pode ser configurado em um servidor. Endereço IP, nomes de rede, unidades de disco, são exemplo de recursos de cluster. Podemos ver a representação de um cluster de dois nós na figura 13. (HOTEK, 2009)

Cada grupo de recurso pertence a apenas um servidor do cluster. Os aplicativos não precisam saber qual servidor hospeda a instância do cluster de failover. Um cluster de failover do SQL Server aparece em uma rede de servidores como se fosse um único servidor, este possui funcionalidades que proporcionam failover de um servidor para outro, caso o principal se torne indisponível, por exemplo, durante uma falha de hardware, uma falha do sistema operacional ou uma atualização planejada do sistema operacional. (MSDN, 2008)

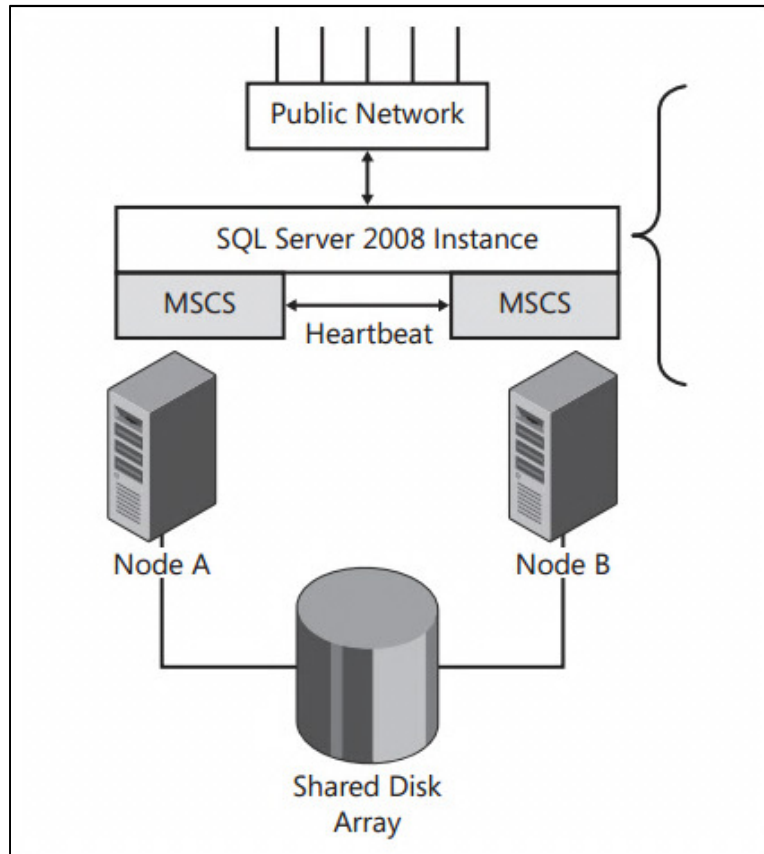


Figura 13: Cluster de dois nós no Windows

Fonte: HOTEK, 2009

É possível utilizar o cluster de failover para reduzir tempo de inatividade do sistema e oferecer maior disponibilidade de aplicativos. (MSDN, 2008)

Vantagens do uso do cluster de failover:

- Failover automático;
- Redirecionamento transparente para os clientes em caso de falhas;

Desvantagens do uso do cluster de failover:

- Requer hardware compatível;
- Custo de implantação;
- Dependência cluster do Windows Server;

3.8.3.2. Espelhamento de banco de dados

Trata-se de uma solução de software para aumento de disponibilidade do banco de dados, dando suporte a um failover automático quase instantâneo. (MSDN, 2008)

O espelhamento de banco de dados permite a criação de servidores de reserva quentes. O servidor principal é o servidor de origem com o banco de dados principal. O servidor espelho é o servidor de destino com o banco de dados de destino espelhado. É possível ainda existir um servidor de monitoramento que realiza o failover automático em caso de falha do servidor principal. Podemos ver a representação deste espelhamento utilizando os três servidores na figura 14. (STANEK, 2010)

O Espelhamento de banco de dados dispõe de três tipos de operações, são elas:

- *Alta Disponibilidade* – fornece transferência síncrona durável entre os bancos de dados principal e secundário, assim como detecta falhas automaticamente e executa o processo de failover, pois faz uso de um servidor de monitoramento. (HOTEK, 2009)
- *Alto Desempenho* – fornece transferência assíncrona e não faz uso do servidor de monitoramento, logo, não é possível o failover automático. (HOTEK, 2009)
- *Alta Segurança* – utiliza um modo de transferência síncrona onde as transações efetivadas são primeiramente gravadas no servidor secundário para depois serem gravadas do servidor principal. Este modo não utiliza o servidor de monitoramento, logo, o failover automático não está disponível. (HOTEK, 2009)

Vantagens do uso do Espelhamento de banco de dados:

- Detecção de falhas e failover automático;
- Redirecionamento transparente para os clientes em caso de falhas;
- Fácil gerenciamento;

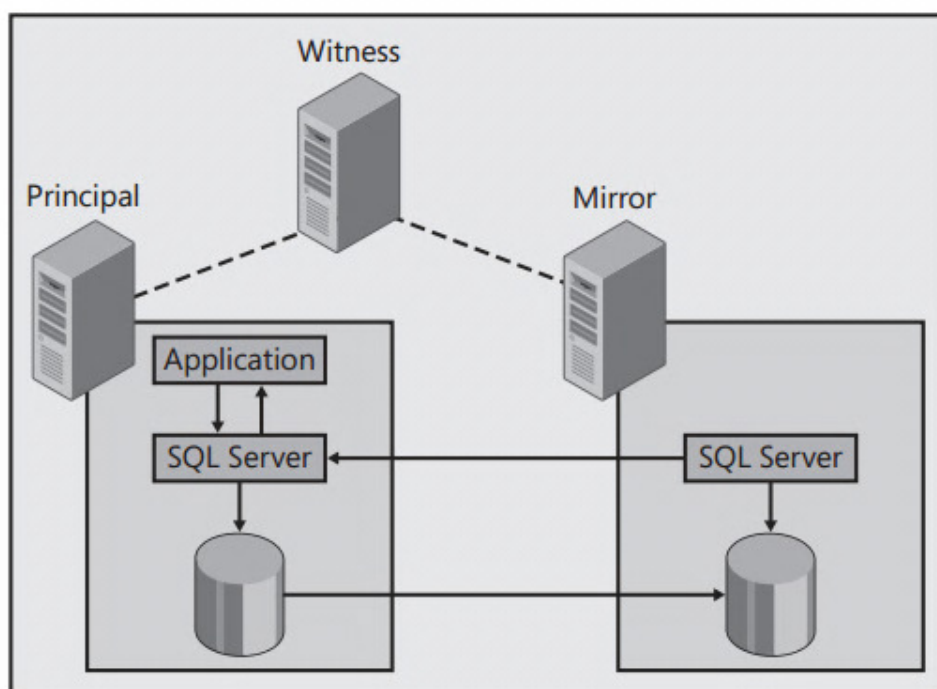


Figura 14: Componentes de um espelhamento de banco de dados

Fonte: HOTEK, 2009

3.8.3.3. Envio de log

Conforme exemplificado na figura 15, consiste em um banco de dados de origem e outro de destino, onde as alterações realizadas na origem são guardadas em backups de log, estes backups são copiados para um diretório do servidor de destino e em seguida restaurados no banco de dados de destino. (STANEK, 2010)

A configuração do Envio de log inclui um servidor primário, secundário e um terceiro como servidor de monitoramento. (MSDN, 2008)

Este tipo de solução pode ser utilizado como uma opção de disponibilidade, porém, comumente, este tipo de solução é utilizado como uma forma de failover secundário para o cluster de failover e para o espelhamento de banco de dados. (HOTEK, 2009)

A grande vantagem de se utilizar o envio de logs é a possível recuperação de dados acidentalmente alterados coletando os dados dos servidores secundários, isto

é possível, pois o tempo de backup do servidor primário para restauração nos servidores secundários é definido pelo usuário. (HOTEK, 2009)

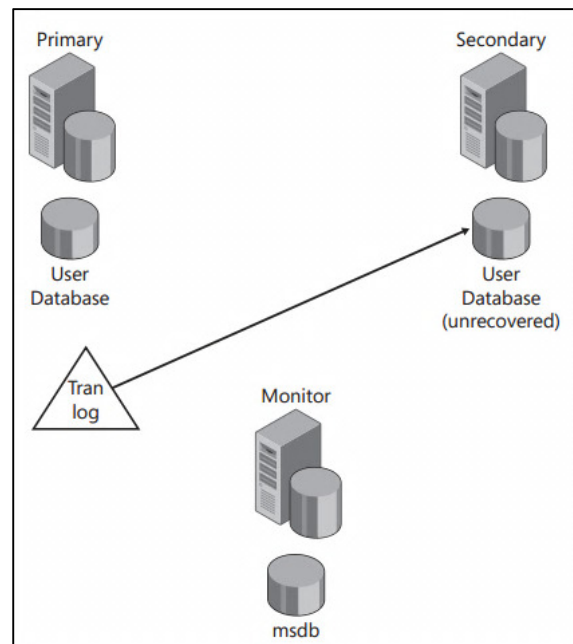


Figura 15: Componentes do envio de log

Fonte: HOTEK, 2009

3.8.3.4. Replicação

A arquitetura de replicação do SQL Server é dividida em vários processos, procedimentos e componentes diferentes. Esta arquitetura de replicação inclui os componentes de replicação que são os componentes de dados utilizados na replicação, chamados de Publicador, Distribuidor e Assinante. Possui também agentes de replicação, que são aplicativos que auxiliam no processo de replicação. Topologias de replicação, que podem ser do tipo Publicador Central ou Assinante central. Por fim, as variantes de replicação ou tipos de replicação, que são três: Replicação Snapshot, Replicação Transacional e Replicação Merge. (STANEK, 2010)

Vantagens do uso da Replicação:

- É possível definir que subconjunto de dados serão replicados para servidores secundários;
- É possível existir mais de uma cópia redundante do banco de dados;
- Disponibilidade dos dados em tempo real;

- Permite a disponibilidade completa dos bancos de dados secundários.

3.8.3.4.1. Componentes de Replicação

3.8.3.4.1.1. Publicador (Publisher)

São servidores que tornam dados disponíveis para replicação em outros servidores. Além disso, os publicadores monitoram as alterações feitas nos dados e mantêm informações sobre banco de dados de origem. Cada estrutura de replicação de dados do SQL Server tem apenas um publicador. (STANEK, 2010)

3.8.3.4.1.2. Distribuidor (Distributor)

São servidores que, como o próprio nome diz, distribuem os dados replicados. Eles armazenam o banco de dados chamado *distribution*, metadados, dados históricos e, para replicação transacional, transações. (STANEK, 2010)

3.8.3.4.1.3. Assinante (Subscriber)

São os servidores de destino da replicação. Tais servidores armazenam os dados replicados e recebem atualizações. Estes também podem fazer alterações nos dados. Por fim, você pode publicar para vários assinantes. (STANEK, 2010)

3.8.3.4.2. Agentes de Replicação

3.8.3.4.2.1. Snapshot Agent

Cria instantâneos dos dados. Este é executado no distribuidor. Cada banco de dados publicado possui o seu Snapshot Agent que se conecta no publicador. (STANEK, 2010)

3.8.3.4.2.2. Distribution Agent

Aplica os dados de replicação de instantâneos ou transações de replicação de transação nos assinantes. Pode ser executado no distribuidor ou no assinante. É utilizado no distribuidor para assinatura de inserção e no assinante para assinatura de extração. (STANEK, 2010)

3.8.3.4.2.3. Merge Agent

Sincroniza as alterações que ocorrem depois que um instantâneo é criado ou iniciado. Se ocorrerem conflitos, estes serão resolvidos seguindo as regras definidas no solucionador de conflitos. Pode ser utilizado no publicador ou no assinante. (STANEK, 2010)

3.8.3.4.2.4. Log Reader Agent

Move as transações marcadas para replicação do log de transações do publicador para o distribuidor. Cada banco de dados publicado usando replicação transacional possui seu agente Log Reader. (STANEK, 2010)

3.8.3.4.2.5. Queue Reader Agente

Armazena as alterações feitas no banco de dados em uma fila, onde são propagadas para o publicador de forma assíncrona. Isso permite que os assinantes modifiquem os dados publicados e sincronizem alterações sem conexão ativa com o publicador. (STANEK, 2010)

3.8.3.4.3. Topologias de Replicação

3.8.3.4.3.1. Publicador central

Como podemos observar na figura 16, consiste em um único publicador central que possui um ou mais assinantes. O publicador central possui a cópia mestra dos dados e é usado para configurar a arquitetura de replicação. Neste tipo de topologia, as alterações de dados ocorrem em uma única origem, o publicador, e são replicadas para os assinantes. (HOTEK, 2009)

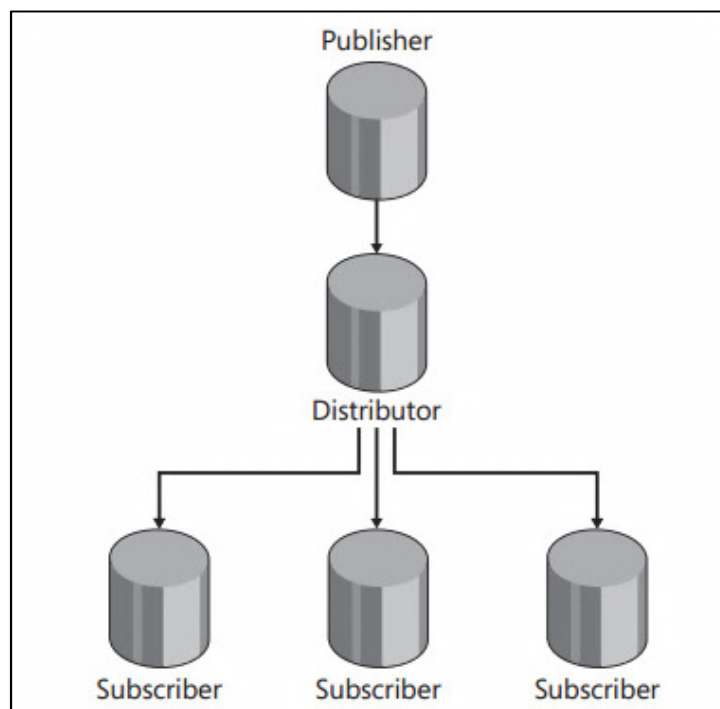


Figura 16: Topologia publicador central

Fonte: HOTEK, 2009

3.8.3.4.3.2. Assinante Central

Nesta topologia um único assinante possui mais de um distribuidor, como vemos na figura 17. As alterações são gravadas em vários publicadores e, então, consolidadas em um único assinante. Este tipo de topologia é comumente utilizado para consolidar vários bancos de dados ou um banco de dados central para produção de relatórios. (HOTEK, 2009)

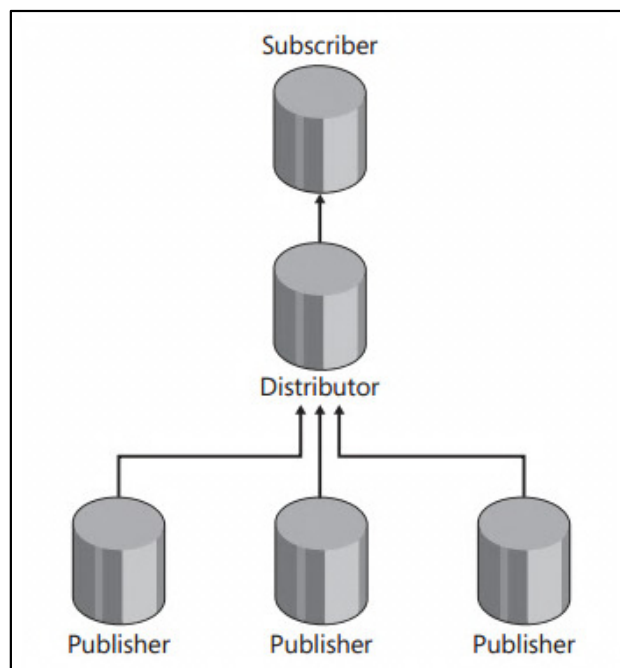


Figura 17: Topologia assinante central

Fonte: HOTEK, 2009

3.8.3.4.4. Variantes de Replicação

3.8.3.4.4.1. Replicação Snapshot

Tira um snapshot (foto) dos dados atuais e substitui a cópia dos dados inteira em um ou mais assinantes. Nos snapshots subsequentes, a cópia inteira do banco de dados é distribuída novamente para os assinantes. Esta cópia exata da base de dados é uma vantagem deste tipo de replicação, porém, em contra partida, é uma desvantagem já que o tráfego de rede aumenta consideravelmente causando uma sobrecarga. Além disso, este tipo de replicação é executada apenas periodicamente, ou seja, os assinantes não possuem informações mais atuais. (STANEK, 2010)

Este tipo de replicação é indicado nas seguintes situações:

- Replicação de pequenos bancos de dados;
- Replicação de bancos de dados que sofreram muitas alterações em um curtíssimo espaço de tempo;

- Bancos de dados que sofrem alterações com pouquíssima frequência.

Na figura 18 temos o exemplo de uma replicação snapshot, onde existe um publicador, origem dos dados, um distribuidor, que faz a distribuição dos dados para os assinantes, que também estão representados na figura.

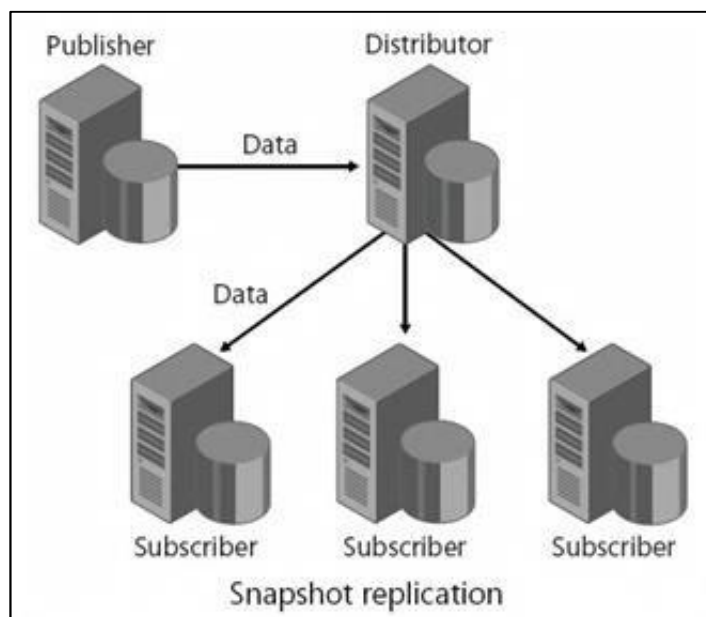


Figura 18: Replicação Snapshot

Fonte: HOTEK, 2009

3.8.3.4.2. Replicação Transacional

Usa transações para distribuir alterações. Principalmente em ambientes servidor-para-servidor. Após iniciada, a replicação cria um snapshot dos dados e envia para os assinantes. Depois de enviado, as transações selecionadas no log de transação do publicador são marcadas para replicação. Em seguida, snapshots periódicos são gerados para garantir que os dados estejam sempre sincronizados. Uma vantagem deste tipo de replicação é que é possível replicar transações individuais ao invés de um conjunto de dados inteiro. Esta pode ocorrer continua ou periodicamente. (STANEK, 2010)

Este tipo de replicação é indicado nas seguintes situações:

- Quando se deseja replicar as modificações incrementais no banco de dados de origem para o banco de dados de destino à medida que elas forem ocorrendo;
- Em aplicações que exigem uma latência baixa entre o momento da mudança e a efetivação da mesma no banco de dados de destino;
- Em aplicações que requerem acessos aos estados intermediários dos dados. Por exemplo, se um mesmo registro foi alterado cinco vezes na base de dados de origem, a replicação transacional executará as mesmas cinco alterações no registro da base de dados de destino;
- Em banco de dados que são modificados frequentemente.

Na figura 19 temos o exemplo de uma replicação transacional, onde possuímos um publicador, que inicialmente faz um snapshot do banco de dados de origem na máquina assinante, um distribuidor, que faz a distribuição dos dados para os assinantes e este último por sua vez, após receber o primeiro snapshot criando uma replica exata do banco de dados do publicador, passa a receber as transações de log realizadas no banco de dados do publicador. Podemos observar também que os agentes de replicação auxiliam este processo.

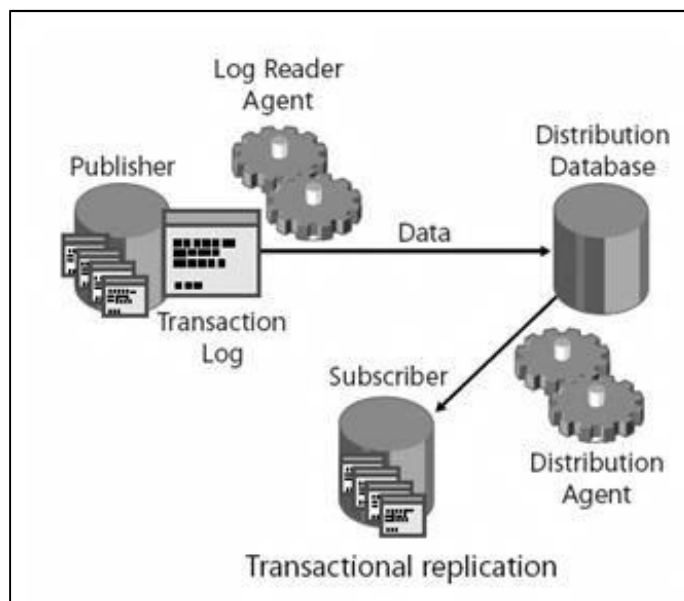


Figura 19: Replicação Transacional

Fonte: HOTEK, 2009

3.8.3.4.4.3. Replicação Merge

Este tipo de replicação permite que os assinantes façam alterações nos dados replicados independentemente, e principalmente em um ambiente servidor-para-cliente. Posteriormente as alterações são mescladas em todos os bancos de dados de origem e destino relacionados. (STANEK, 2010)

Este tipo de replicação é indicado nas seguintes situações:

- Os bancos de dados envolvidos na replicação precisam possibilitar alterações nos seus dados;
- Em banco de dados que necessitam realizar modificações off-line nos dados recebidos de outro banco de dados;
- Cada banco de dados trabalha com porções diferentes dos dados;
- Conflitos são permitidos e podem ser detectados e tratados posteriormente;
- Quando a aplicação não necessita acessar os estados intermediários dos dados. Por exemplo, se um mesmo registro foi alterado cinco vezes na base de dados de origem, a replicação merge só executará

a última das cinco alterações no registro da base de dados de destino;

Na figura 20, podemos observar o fluxo de dados realizado em uma replicação merge, onde tanto o publicador quanto o assinante podem modificar seus dados. Podemos observar também o uso do agente de resolução de conflitos.

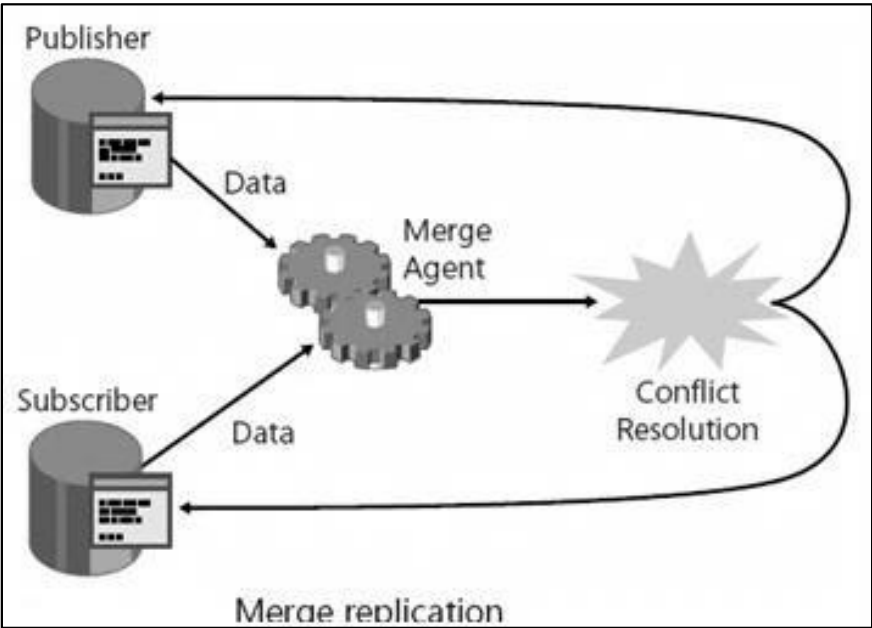


Figura 20: Replicação Merge
Fonte: HOTEK, 2009

A tabela 2 faz um comparativo entre as edições do SQL Server 2008 e a disponibilidade das variantes de replicação.

Replicacao	Datacenter	Enterprise	Standard	Web	Workgroup	Express
Snapshot replication	x	x	x	Subscriber only	x	Subscriber only
Merge replication	x	x	x	Subscriber only	Restricted	Subscriber only
Transactional replication	x	x	x	Subscriber only	Restricted	Subscriber only

Tabela 2: Replicação x Edição disponível
Fonte: TRIPOLI, 2011

4. CONFIGURANDO UMA REPLICAÇÃO TRANSACIONAL DE BANCO DE DADOS DO MICROSOFT SQL SERVER 2008

4.1. Replicação Transacional

4.1.1. Modelo

A replicação transacional é iniciada com um snapshot sendo aplicado no assinante, isto garante que os dois bancos de dados serão sincronizados corretamente. A medida que as transações são executadas no publicador o mecanismo de replicação as aplica no assinante. Este fluxo de transação incremental do publicador torna este tipo de replicação uma ótima opção para alta disponibilidade. (HOTEK, 2009)

4.1.2. Controle de alteração

A replicação transacional é gerenciada por dois agentes de replicação: O Log Reader e o Distribution Agent. O Log Reader Agent é responsável por mover alterações do log de transação do publicador para o banco de dados de distribuição. O Distribution Agent é responsável por mover lotes de alterações do banco de dados de distribuição para cada assinante. (HOTEK, 2009)

4.1.3. Preparando banco de dados para replicação

Para este ambiente de simulação foram utilizadas duas máquinas virtuais criadas no **Virtual Box versão 4.0.4** com **Windows Server 2003 Enterprise Edition SP2**. Foi Utilizado o **SQL Server 2008 R2 Developer** em ambas as máquinas. As máquinas foram configuradas em rede utilizando um grupo de trabalho. O servidor principal foi chamado de **S1** e o servidor de redundância foi chamado de **S2**. As instâncias do SQL Server foram instaladas com as configurações padrões e nomeadas como ***\SQL2008** em ambas as máquinas.

Na máquina virtual **S1** abra o **SQL Server Management Studio** e conecte na instância **S1\SQL2008**, ver figura 21.

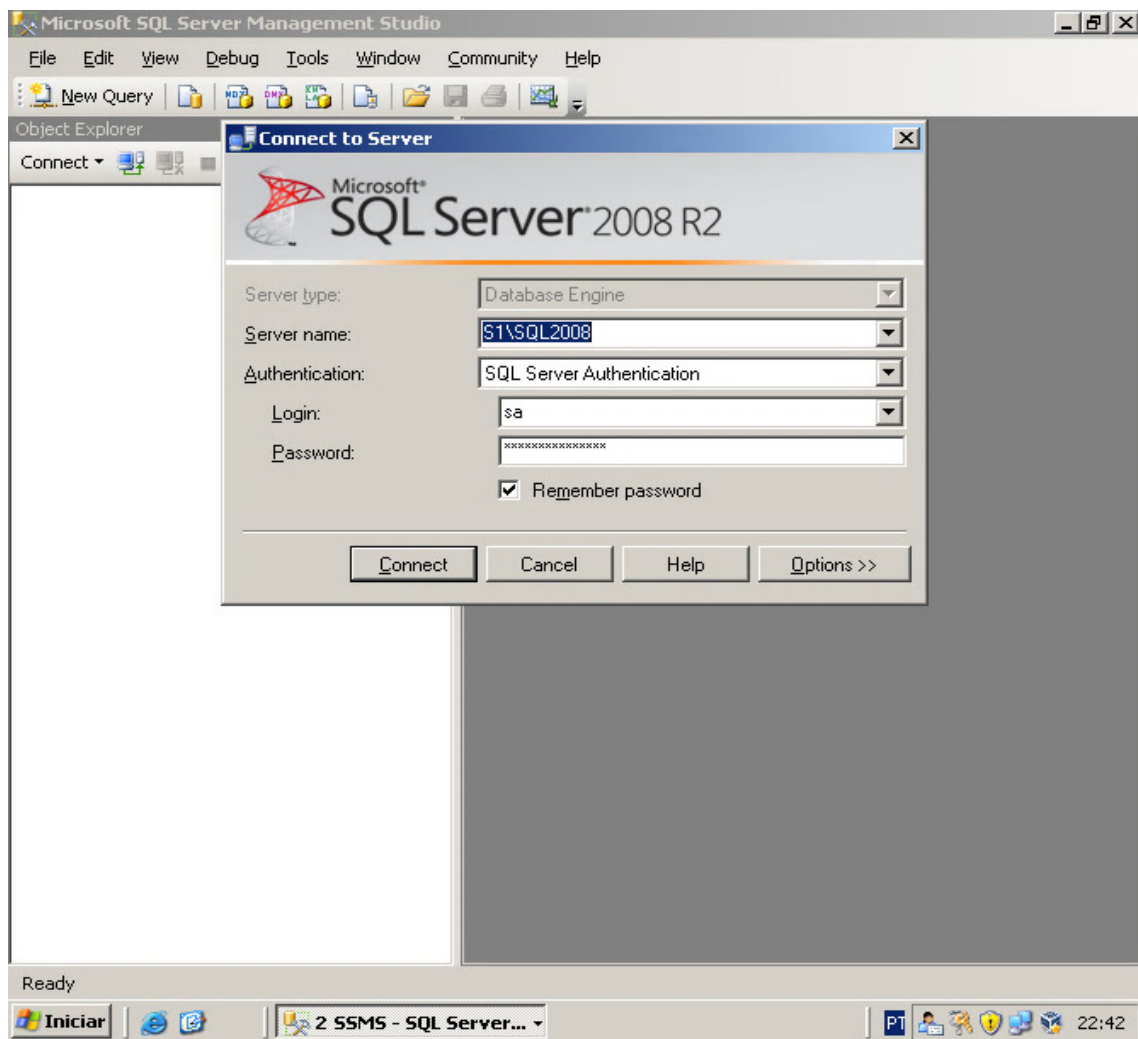


Figura 21: Conectando na instância S1 do SQL Server

Fonte: Resultado de pesquisa

Conforme figura 22, crie um banco de dados chamado **AWTransaction** através de uma cópia do banco de dados **AdventureWorks2008R2**.

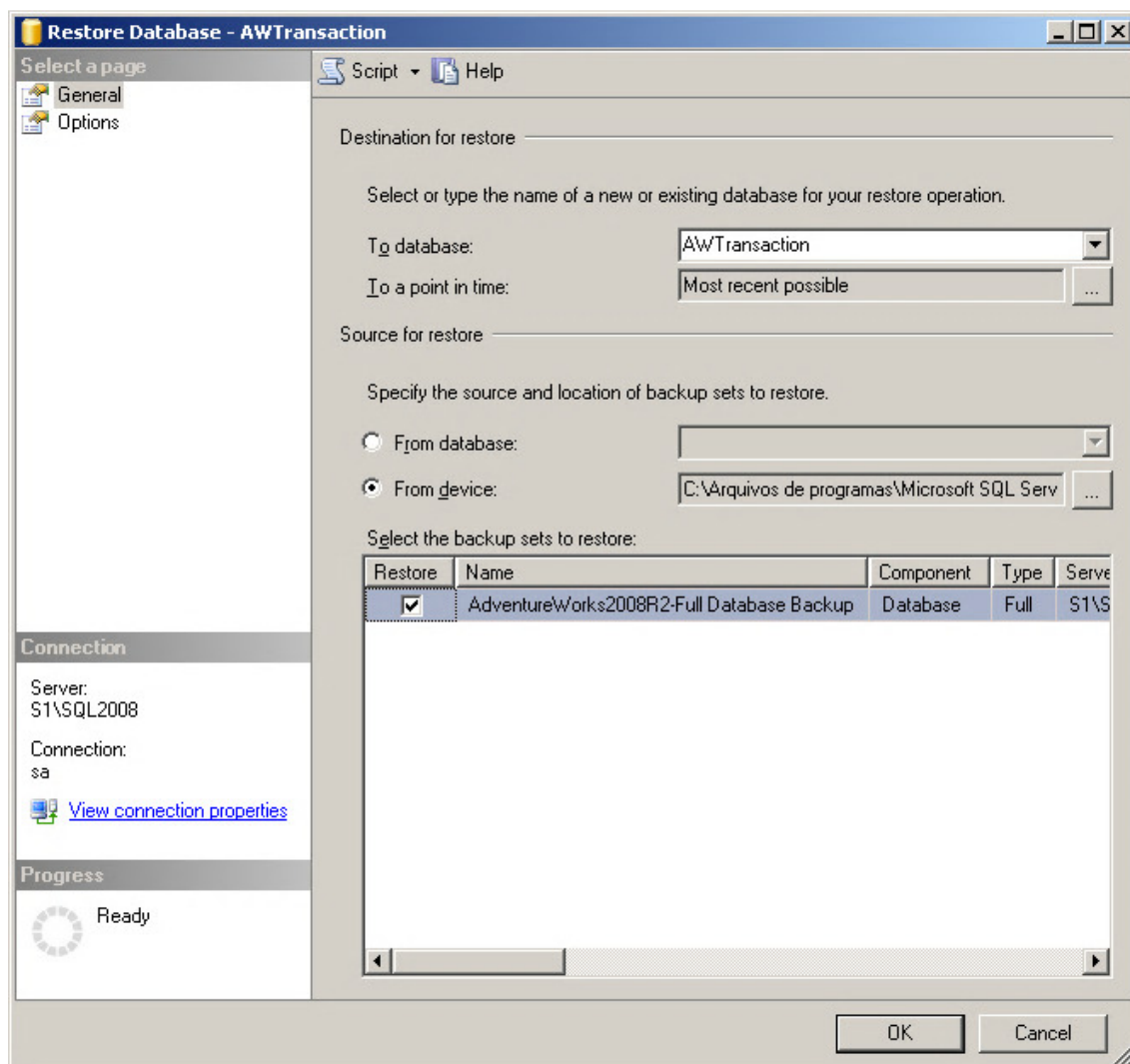


Figura 22: Restauração de backup do banco de dados AWTransaction

Fonte: Resultado de pesquisa

Na máquina virtual **S2** abra o **SQL Server Management Studio** e conecte na instância **S2\SQL2008**, ver figura 23.

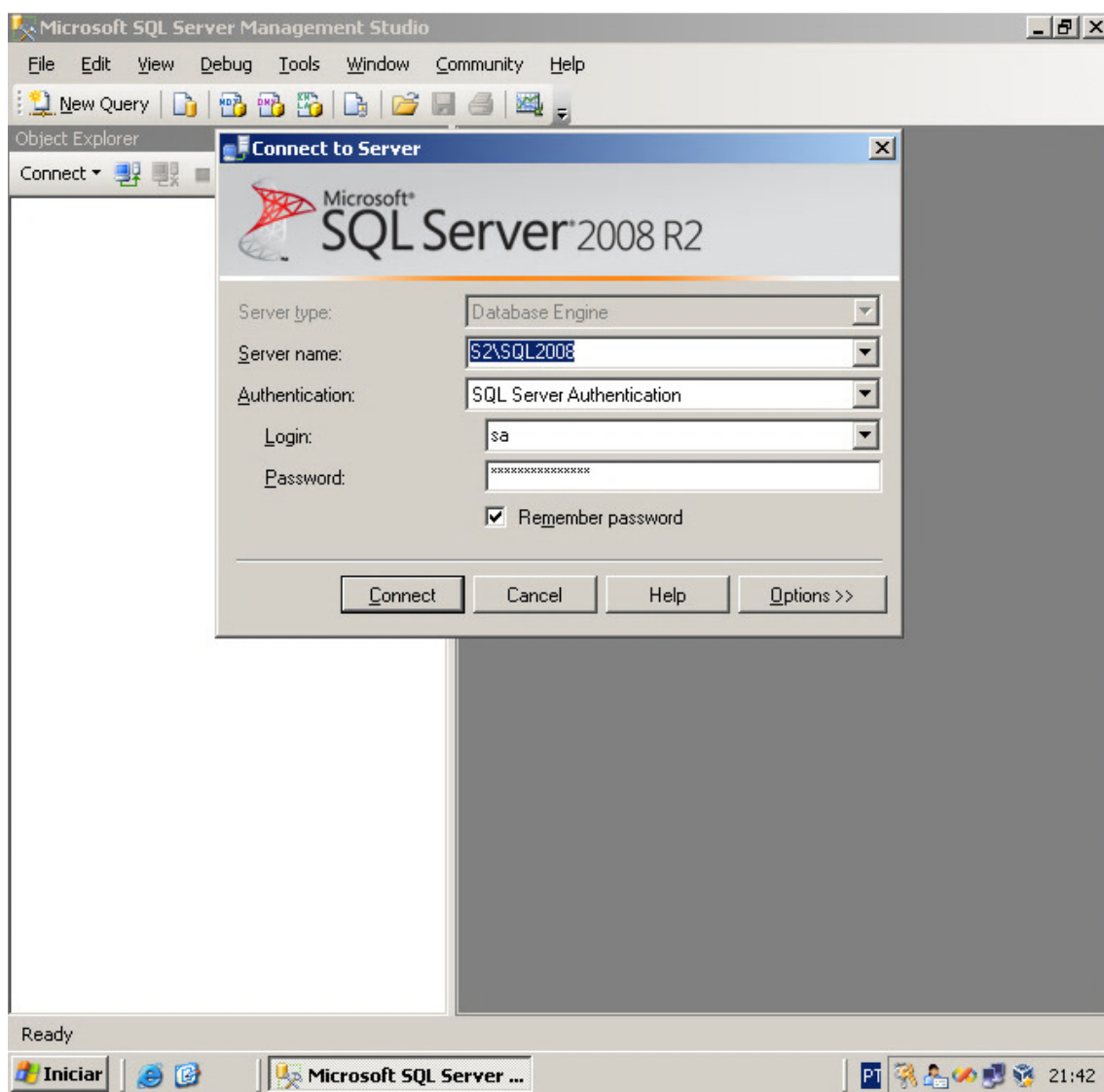


Figura 23: Conectando na instância S2 do SQL Server

Fonte: Resultado de pesquisa

Crie um banco de dados chamado **AWTranSubscriber**, conforme exemplificado na figura 24.

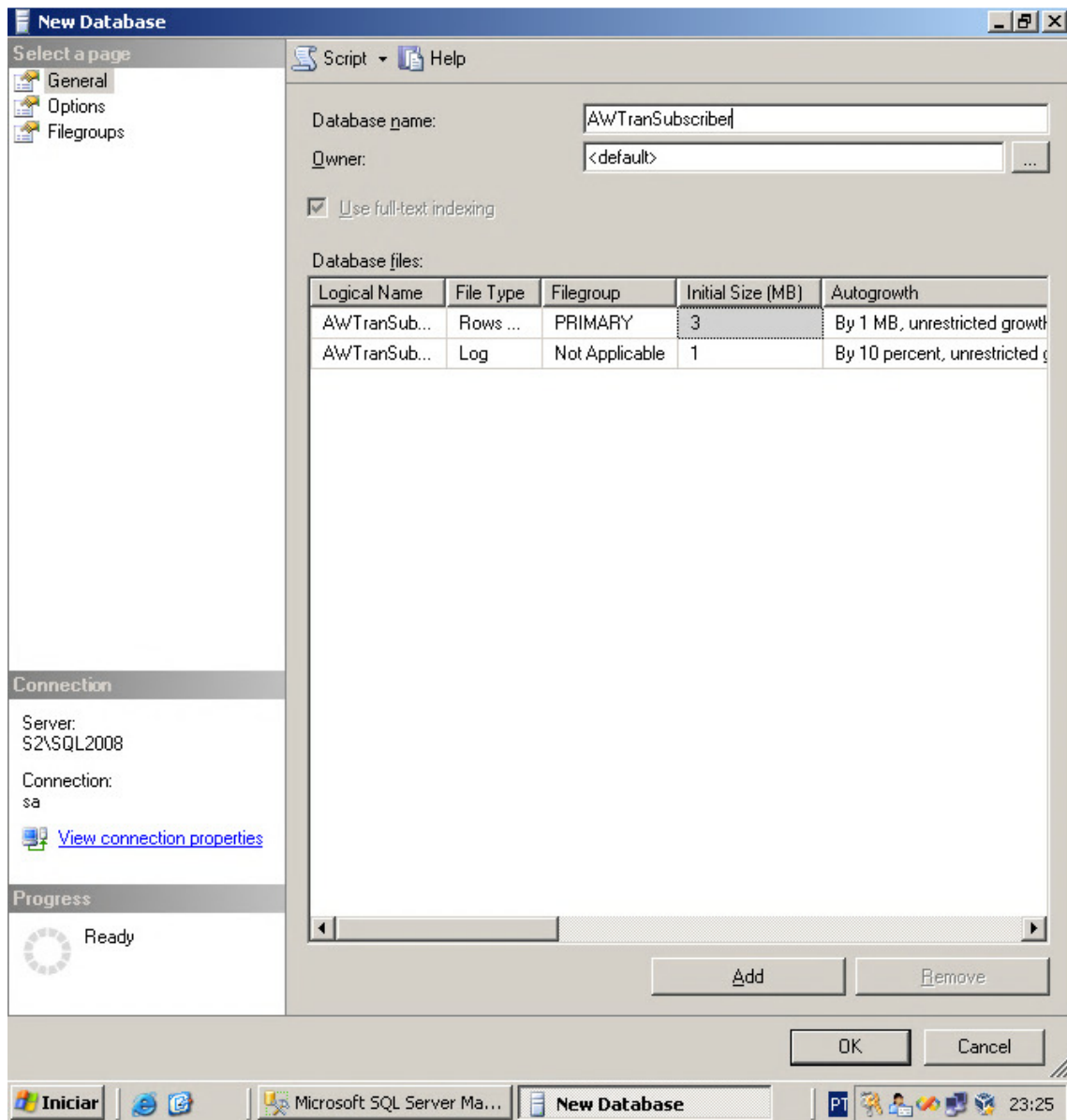


Figura 24: Criação do banco de dados AWTransaction

Fonte: Resultado de pesquisa

O banco de dados **AWTransaction** será o banco de dados principal (publicador) onde serão executadas as transações do ambiente de produção e o banco de dados **AWTranSubscriber** será o banco de dados redundante (assinante) e receberá as alterações efetuadas no banco de dados principal.

4.1.4. Configurando um distribuidor

Na máquina virtual **S1**, no **SQL Server Management Studio** clique com o botão direito do mouse em **Replication** e em seguida em **Configure Distribution**, ver figura 25.

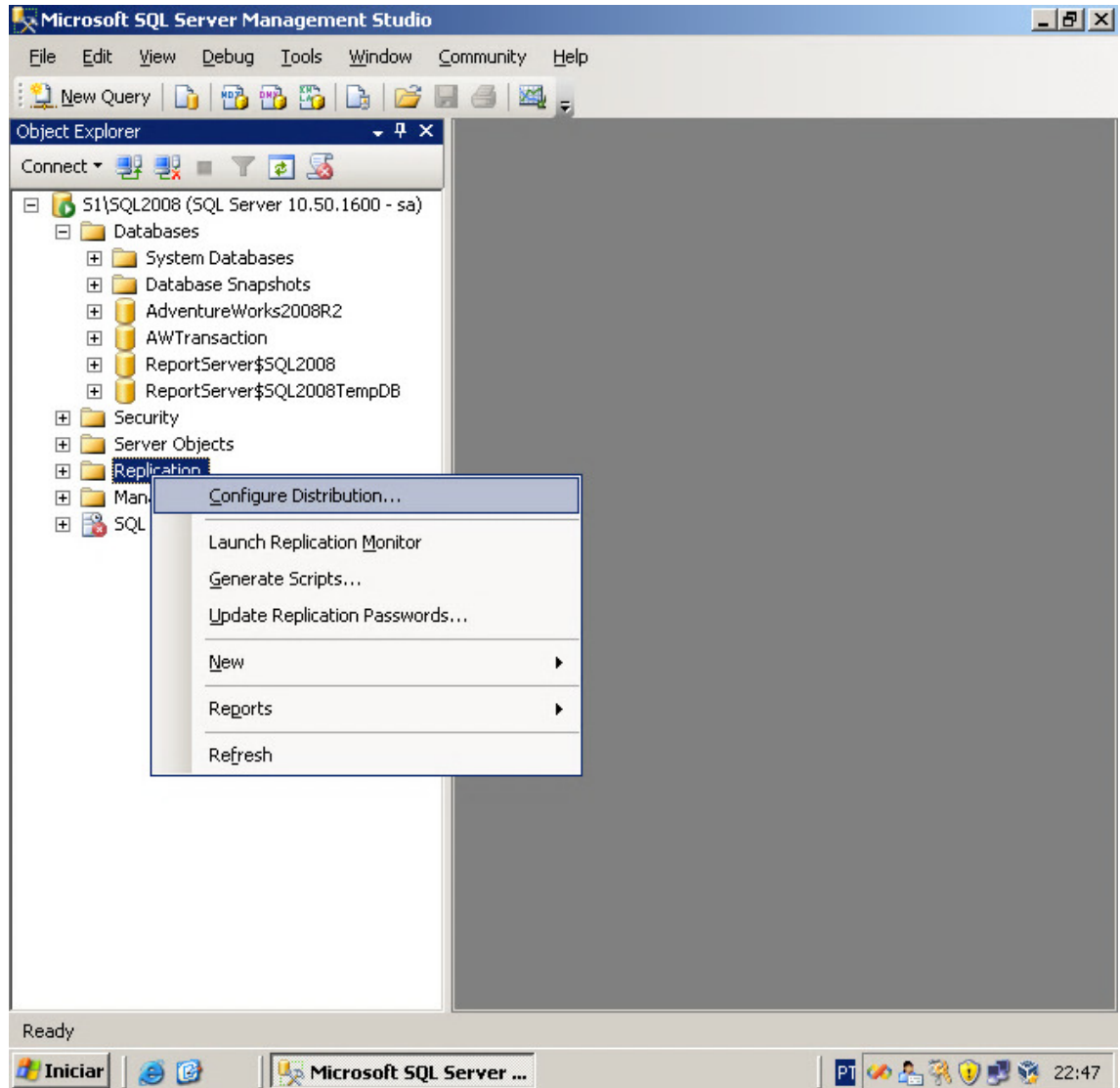


Figura 25: Configuração de um Distribuidor

Fonte: Resultado de pesquisa

Na primeira tela exibida clique em *Next*, ver figura 26.

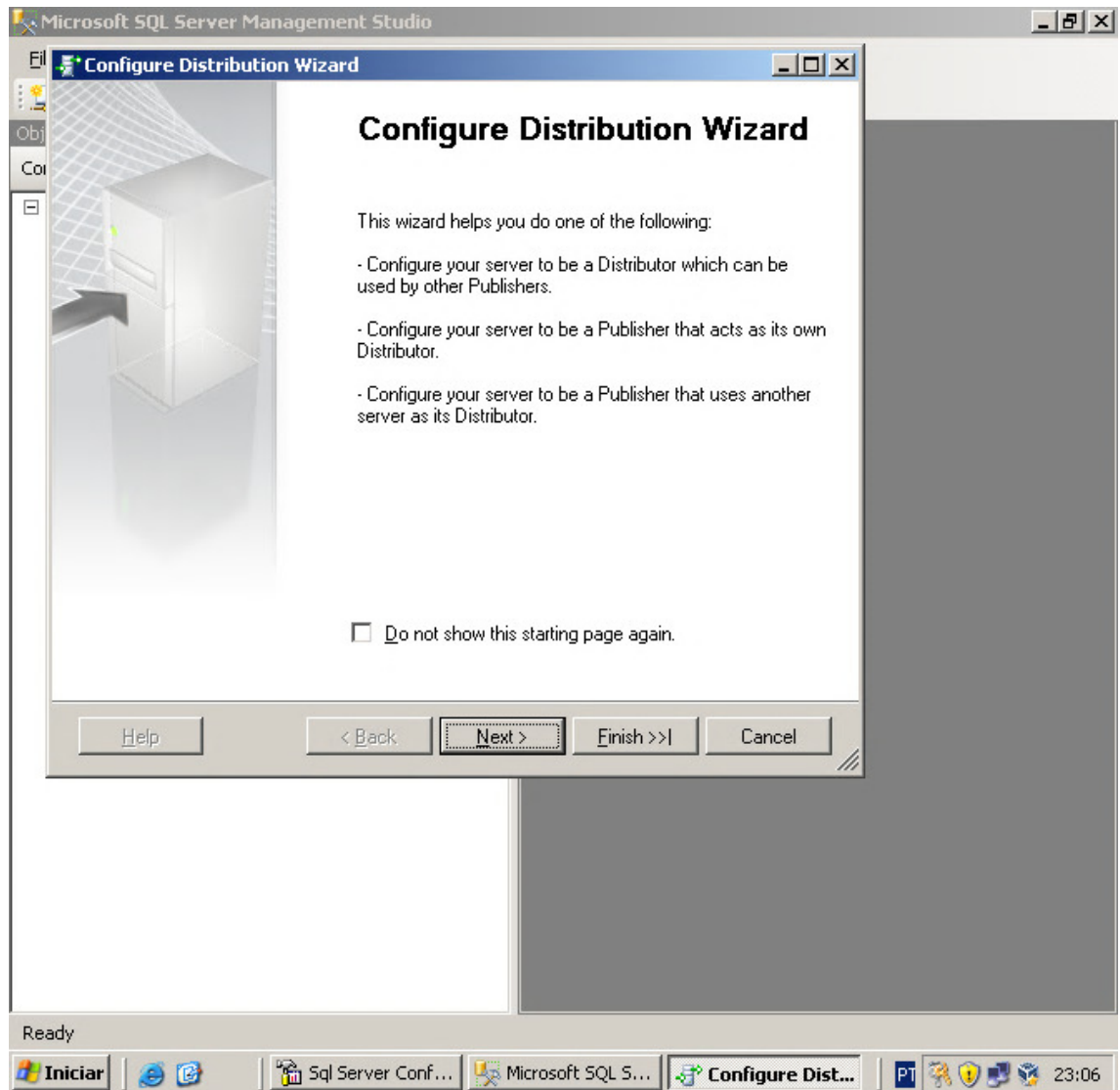


Figura 26: Configuração inicial do Distribuidor

Fonte: Resultado de pesquisa

Marque a opção *'S1\SQL2008' will act as its own Distributor; SQL Sever will create a distribution database nad log*, conforme figura 27. Isto fará com que a instância padrão **S1\SQL2008** atue como distribuidor. Em seguida, clique em *Next*.

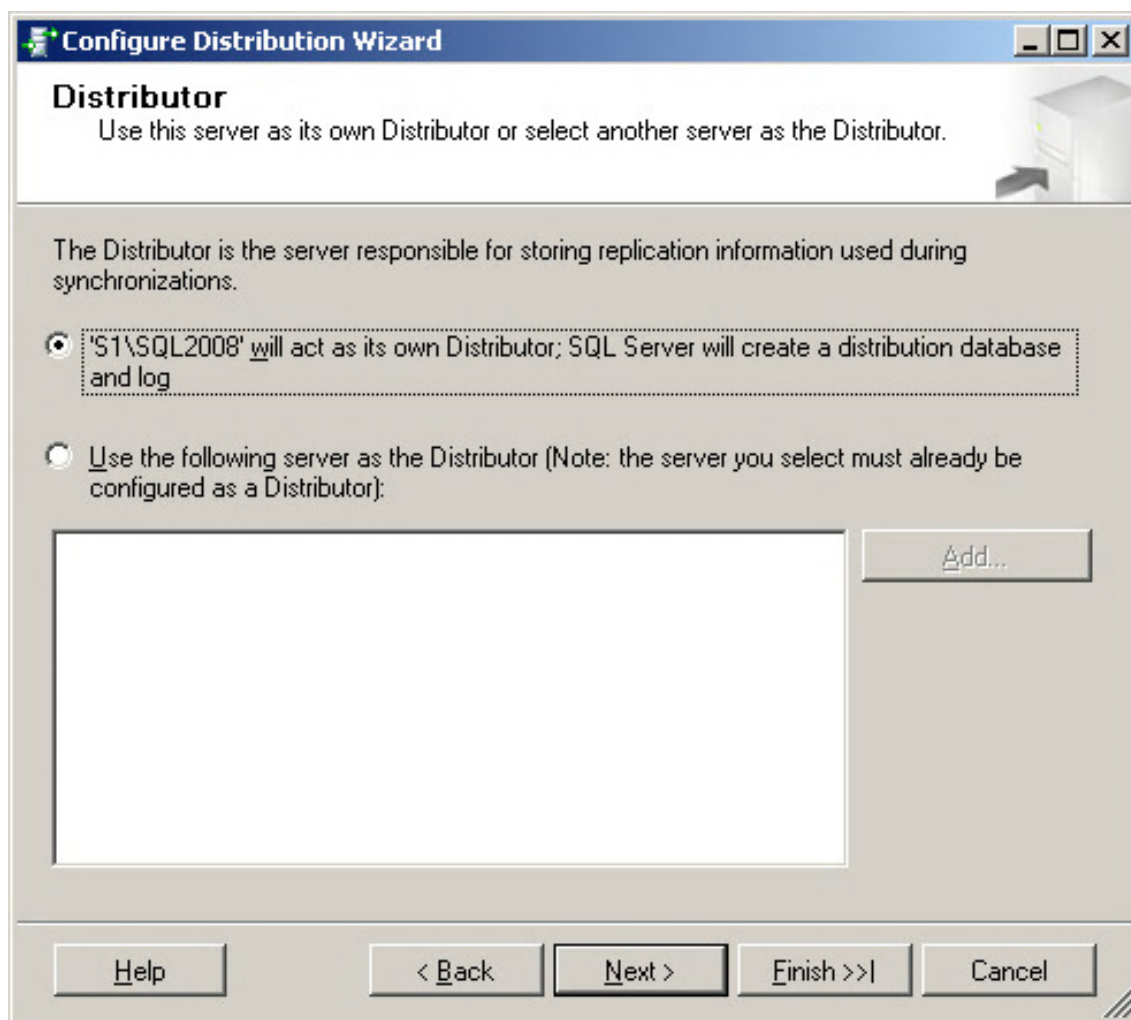


Figura 27: Configuração da instância padrão do Distribuidor

Fonte: Resultado de pesquisa

Deixe a **pasta de snapshot** configurada com o valor padrão, conforme figura 28, e, em seguida, clique em *Next*.

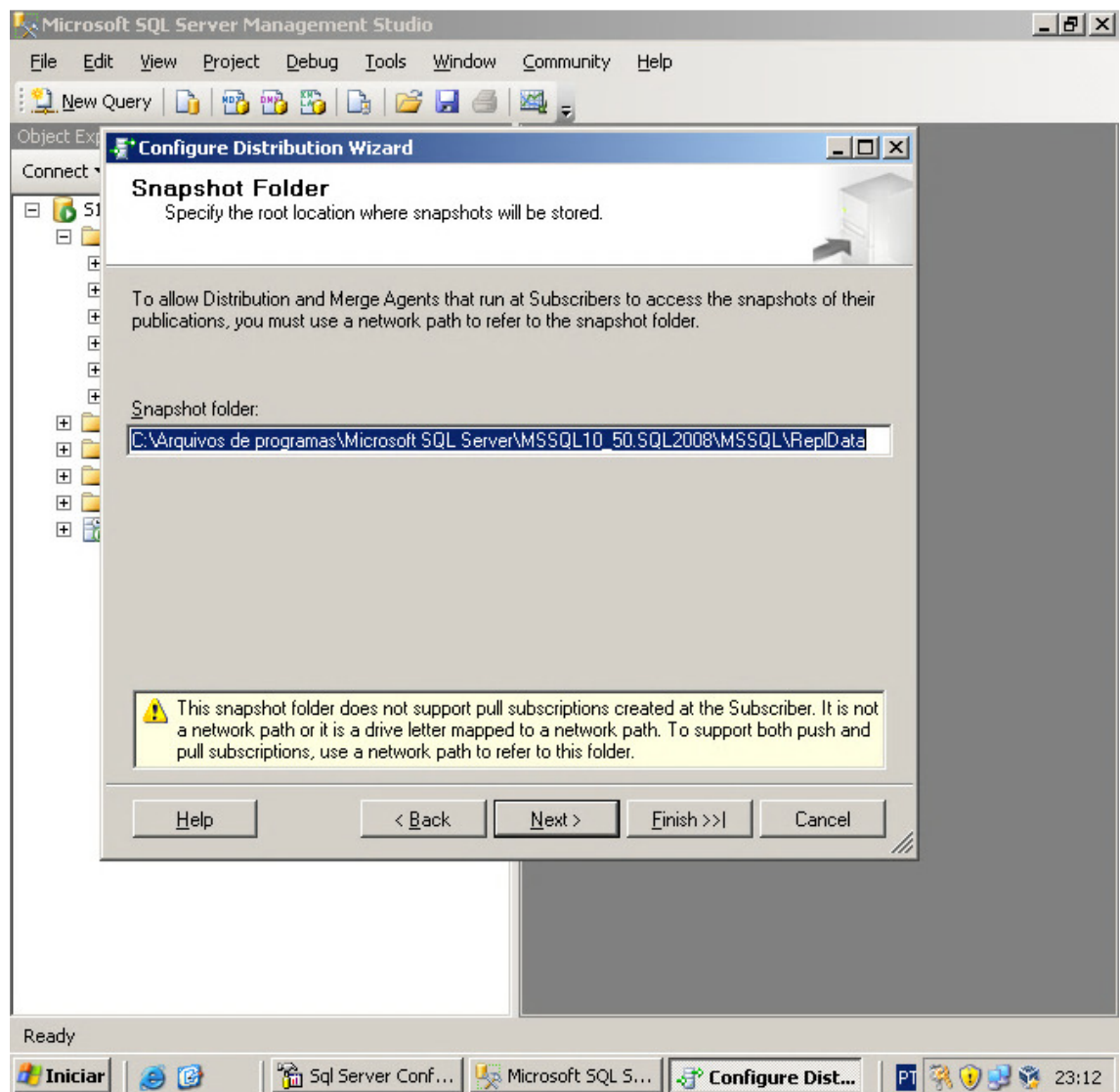


Figura 28: Configuração de pasta padrão para snapshot

Fonte: Resultado de pesquisa

Deixe o nome e o local do banco de dados de distribuição configurados conforme valores padrões, conforme figura 29, e, em seguida, clique em *Next*.

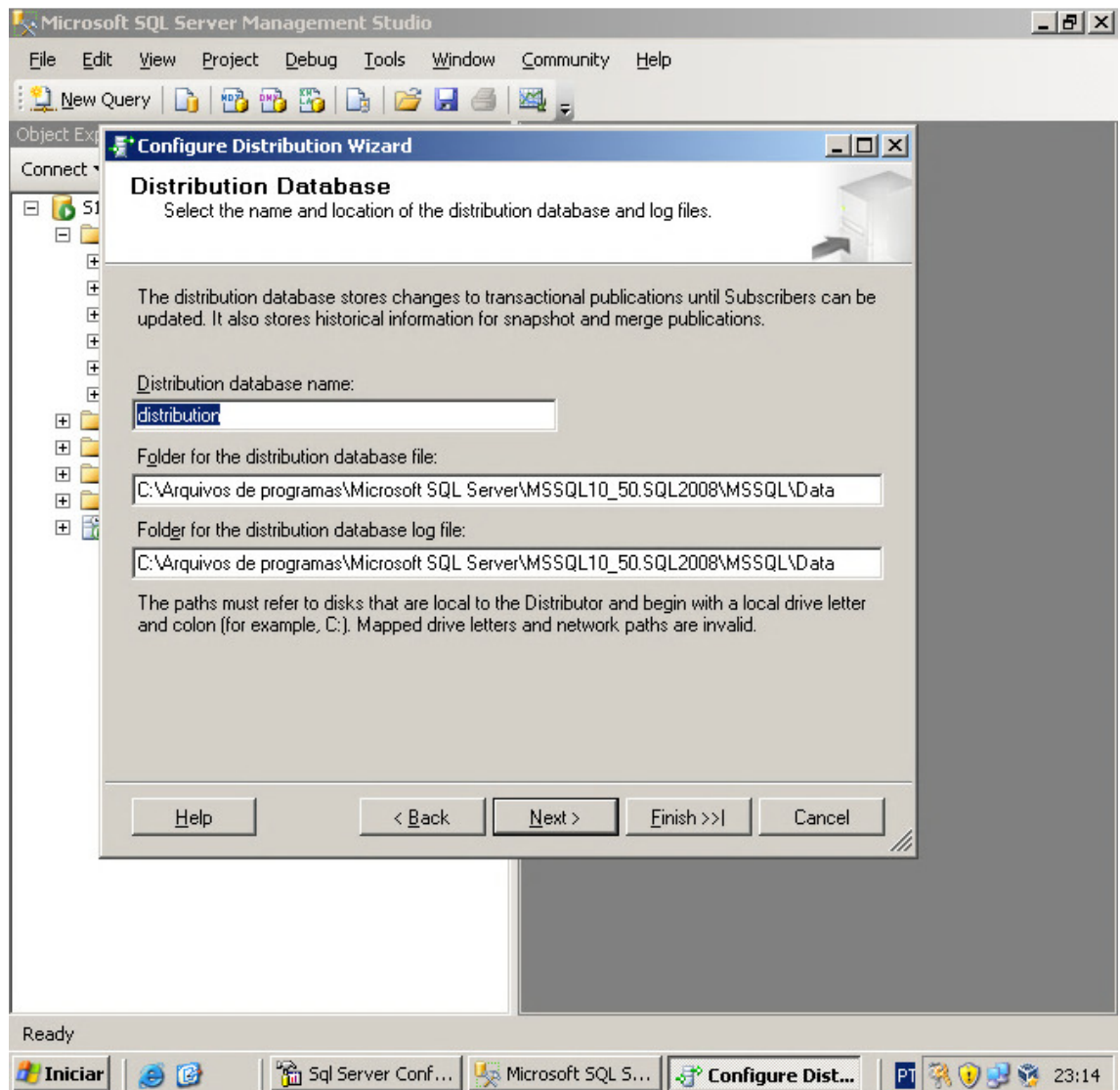


Figura 29: Configuração do banco de dados de distribuição

Fonte: Resultado de pesquisa

Verifique se a instância **S1\SQL2008** está marcada como **publicador**, conforme figura 30.

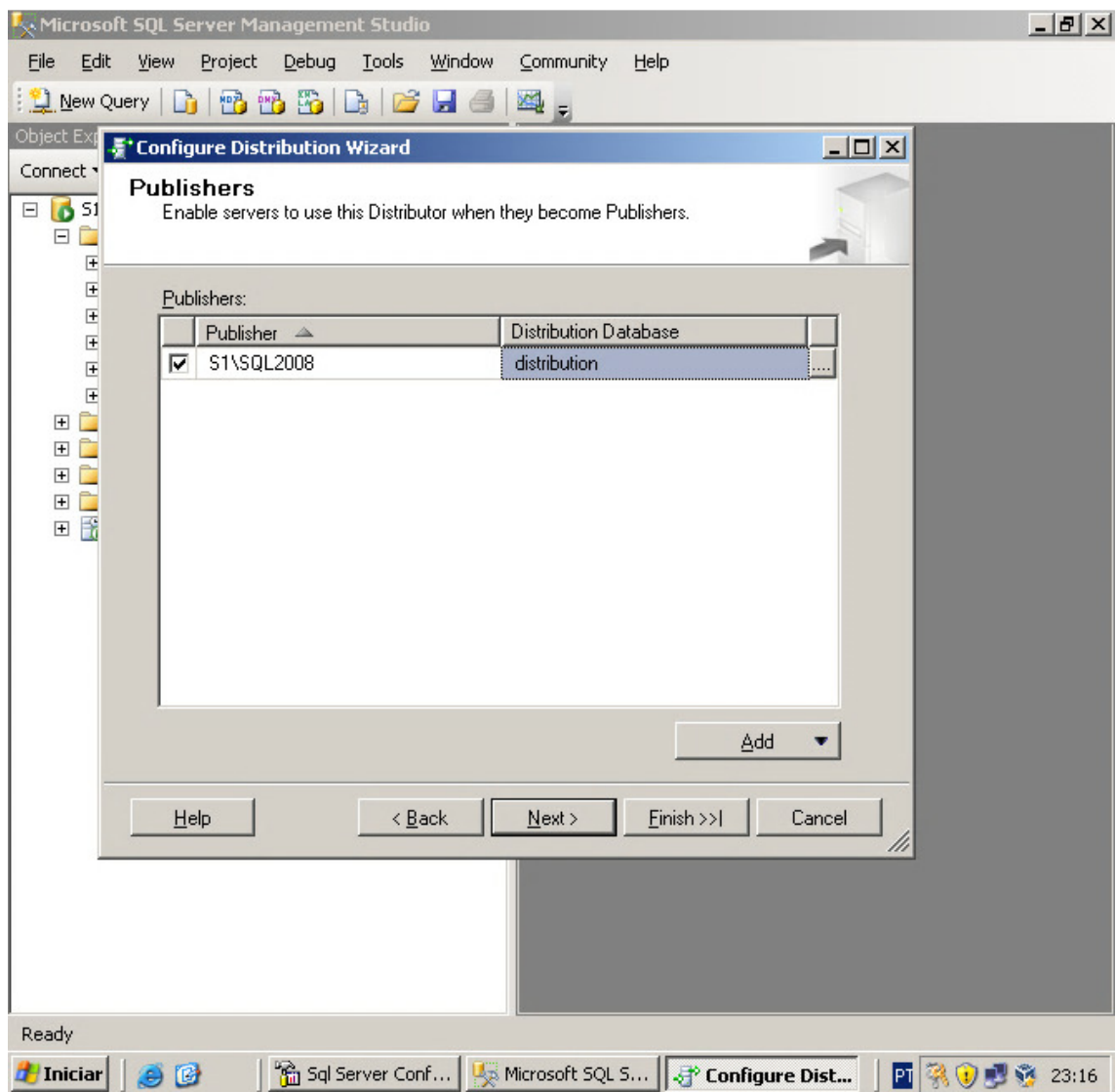


Figura 30: Configuração da instância padrão para distribuição

Fonte: Resultado de pesquisa

Verifique se a caixa de seleção **Configure Distribution** está marcada, conforme figura 31, e, em seguida, clique em *Next*.

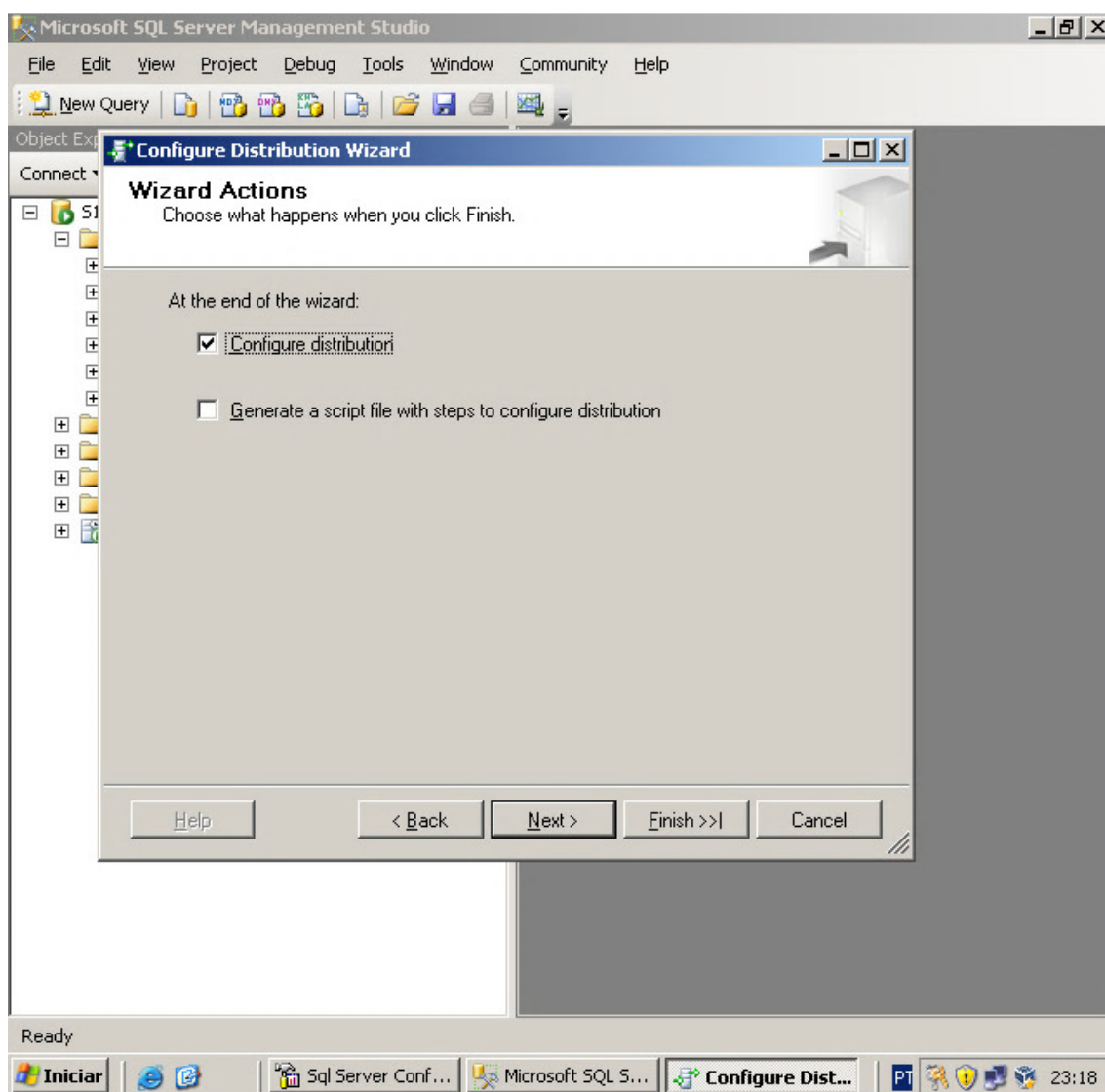


Figura 31: Confirmação da configuração do Distribuidor

Fonte: Resultado de pesquisa

Clique em *finish* para concluir a configuração, conforme figura 32, e, em seguida, clique em *close*.

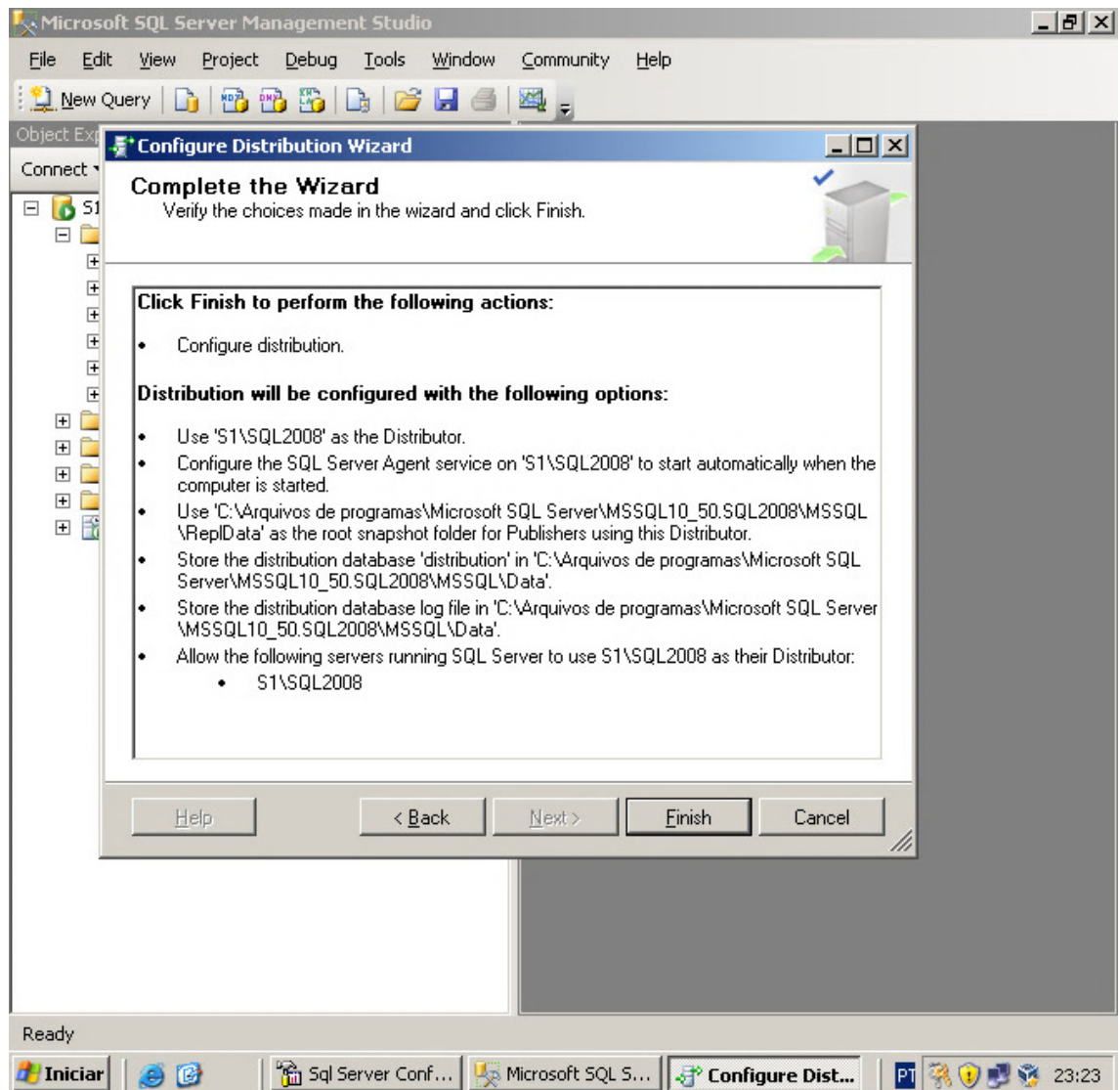


Figura 32: Finalização da configuração do Distribuidor

Fonte: Resultado de pesquisa

4.1.5. Configurando um Publicador

Na máquina virtual **S1**, no **SQL Server Management Studio** expanda **Replication**. Clique com o botão direito do mouse em **Local Publications** e escolha **New Publication**, ver figura 33.

Na primeira tela exibida clique em *Next*.

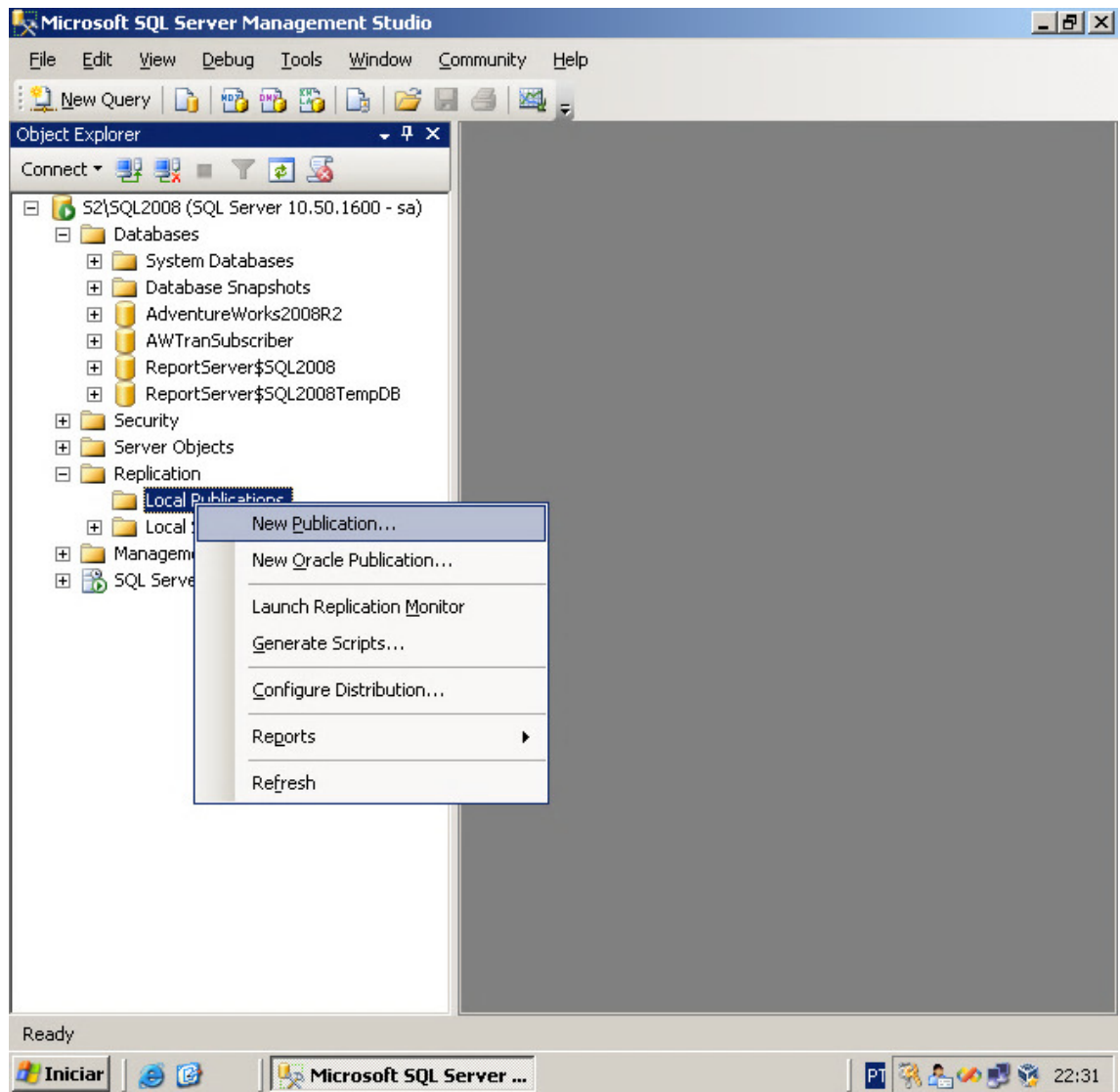


Figura 33: Configuração de um Publicador

Fonte: Resultado de pesquisa

Selecione o banco de dados *AWTransaction*, conforme figura 34, e, em seguida, clique em *Next*.

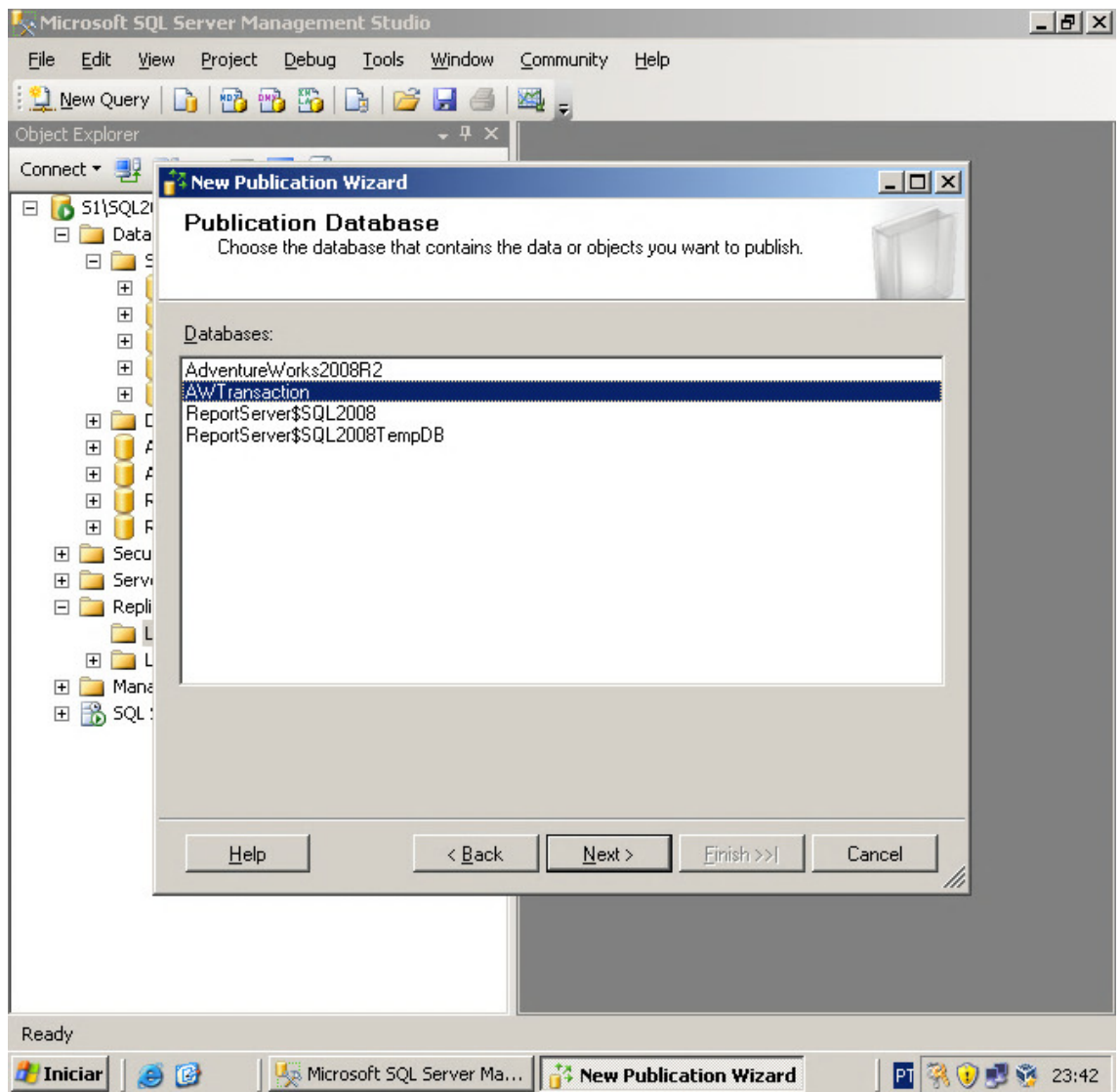


Figura 34: Selecionando banco de dados de publicação

Fonte: Resultado de pesquisa

Selecione **Transactional Publication**, conforme figura 35, e, em seguida, clique em *Next*.

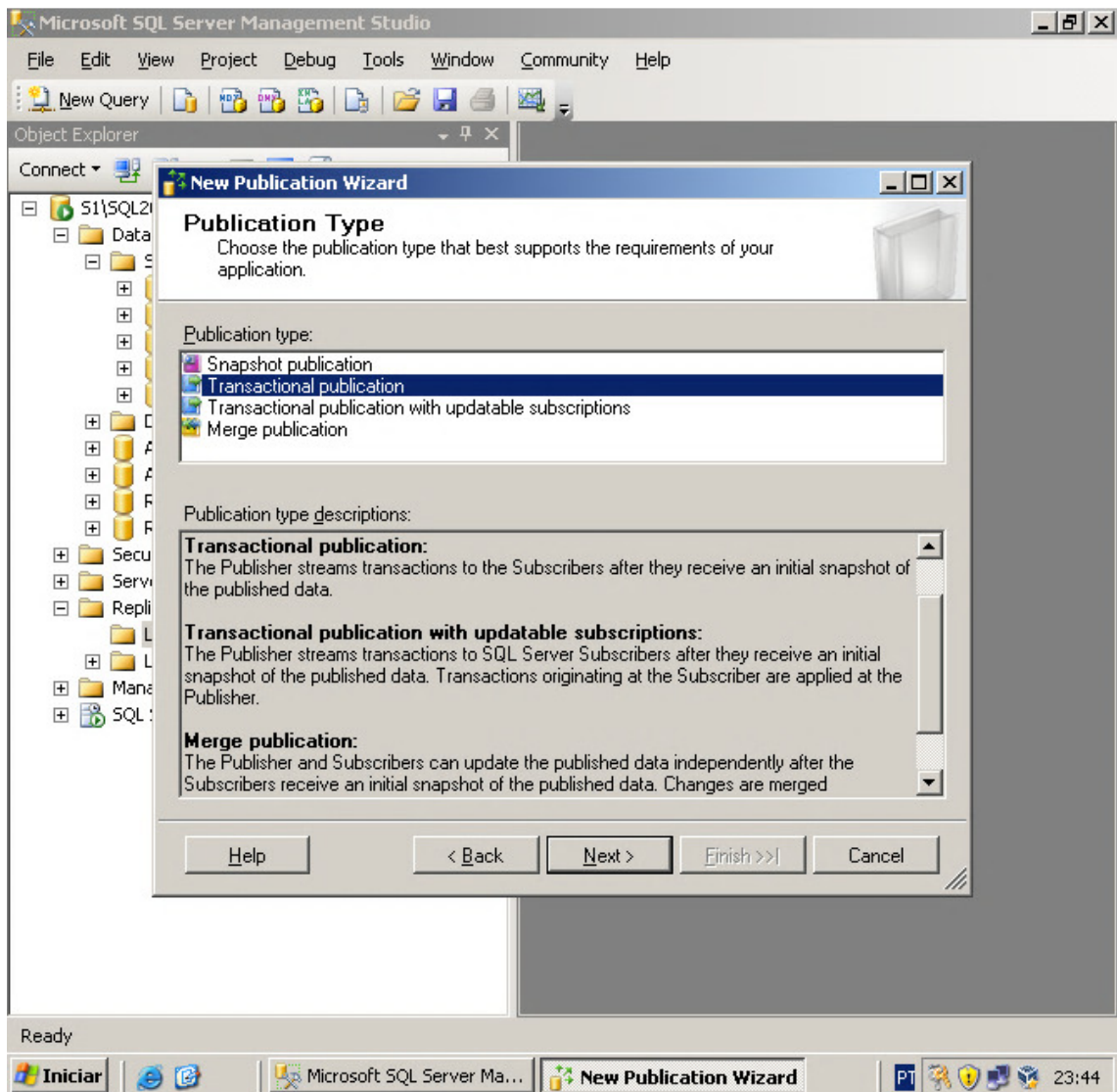


Figura 35: Selecionando o tipo de publicação

Fonte: Resultado de pesquisa

Por se tratar de uma simulação, selecione apenas a opção *Tables* e *User Defined Functions*, conforme figura 36, esta última é necessária porque existem tabelas dentro do banco de dados AdventureWorks que possuem constraints definidas com uma UDF.

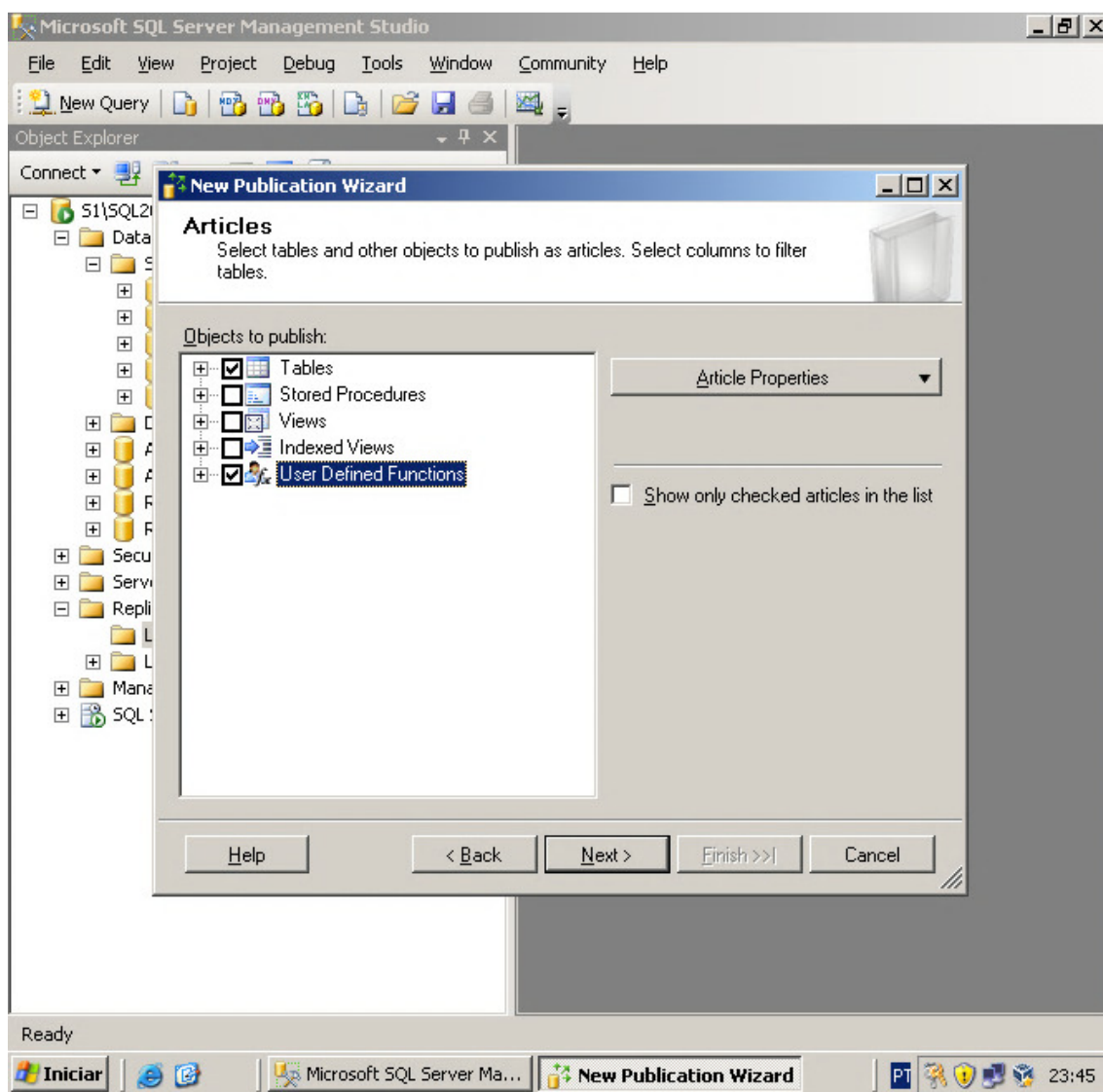


Figura 36: Selecionando objetos para publicação

Fonte: Resultado de pesquisa

Conforme figura 37, nesta tela clique em *Next*. Por se tratar de um processo de replicação para alta disponibilidade, não será necessário adicionar filtros para os artigos publicados.

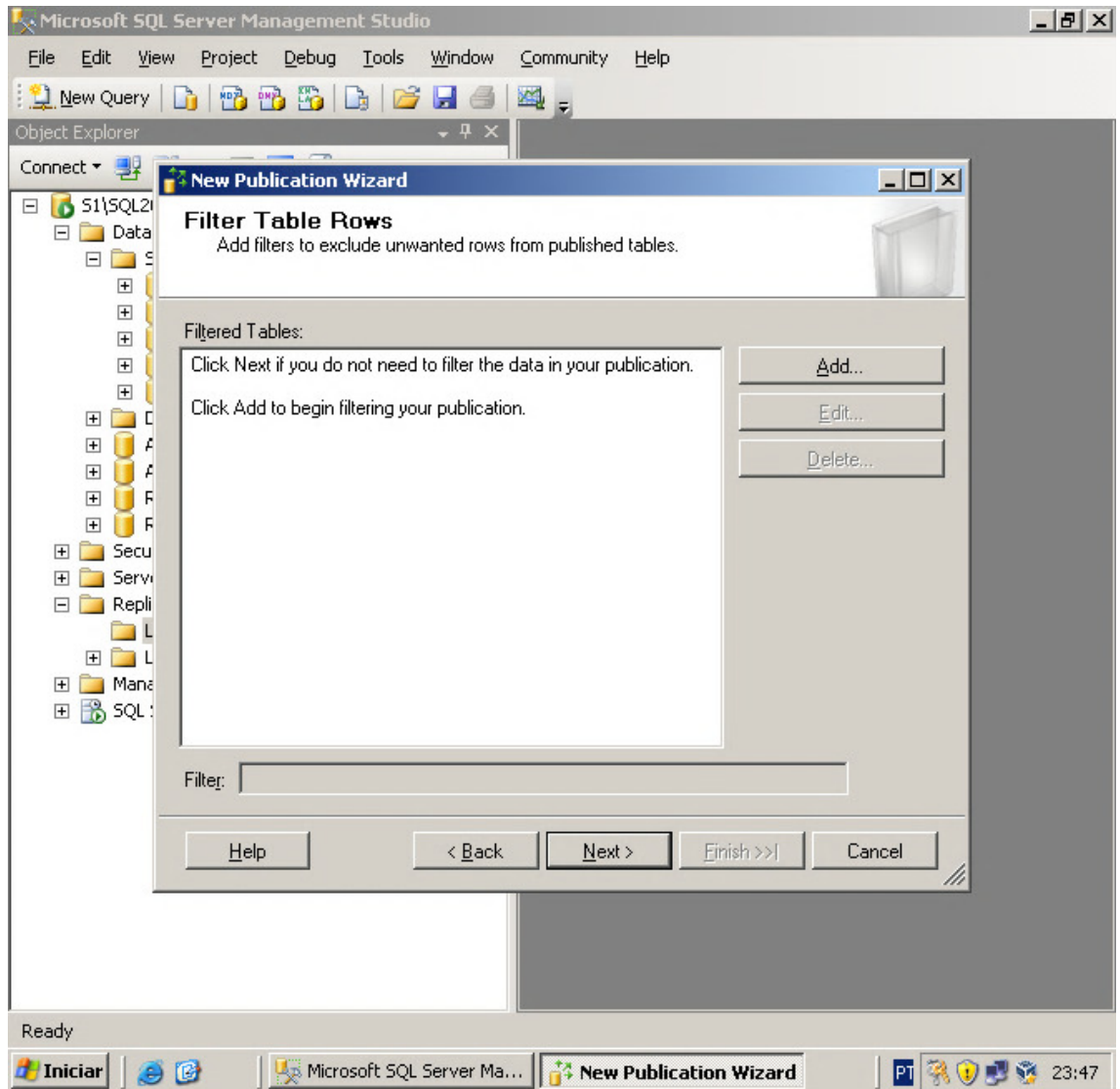


Figura 37: Filtro para objetos publicados

Fonte: Resultado de pesquisa

Marque a opção *Create A Snapshot Immediately and keep the sanpshot available to intialize subscriptions.*, conforme figura 28, e, em seguida, clique em *Next*.

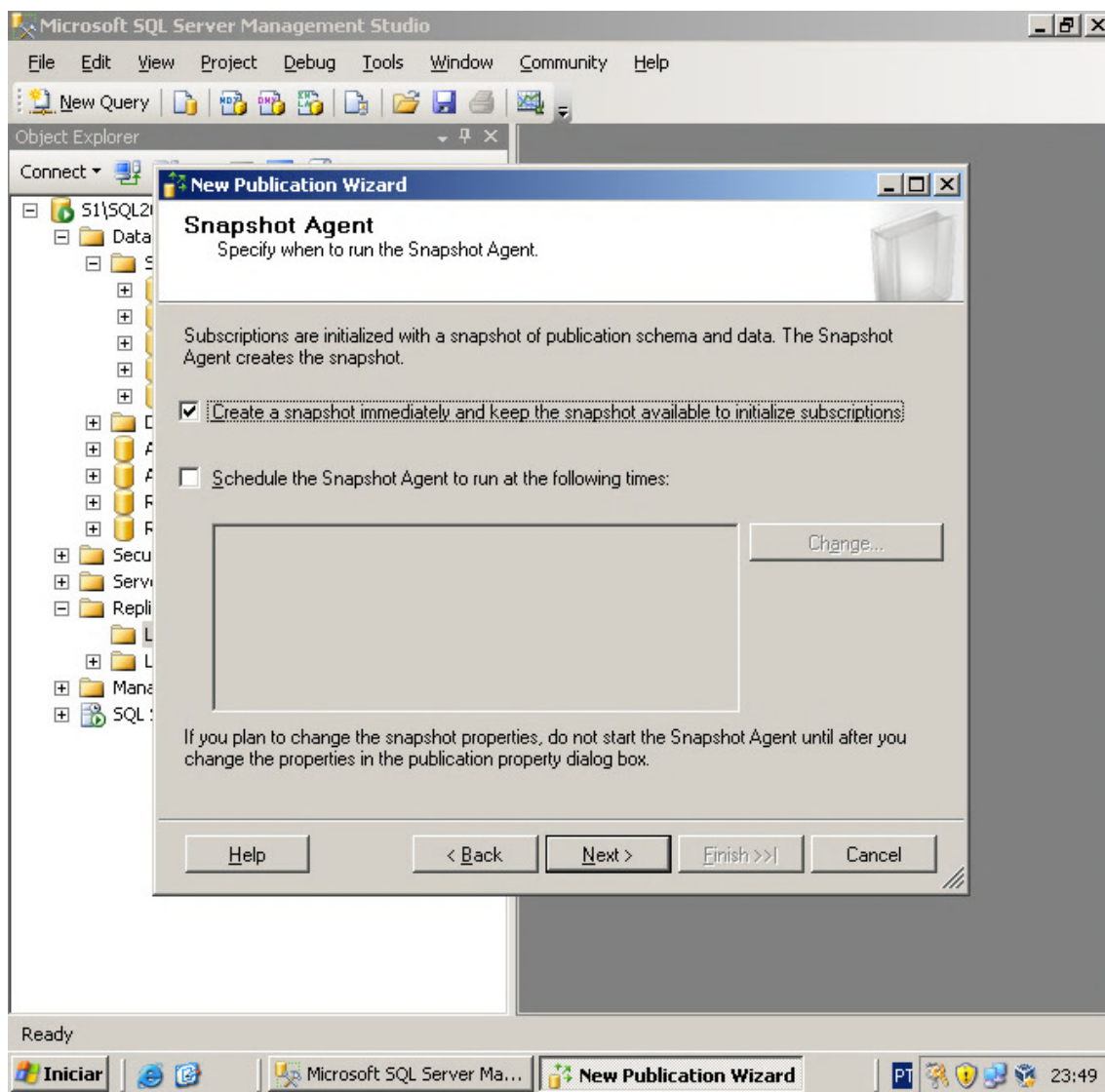


Figura 38: Configuração do Snapshot Agent

Fonte: Resultado de pesquisa

Clique em Security Settings, conforme figura 39.

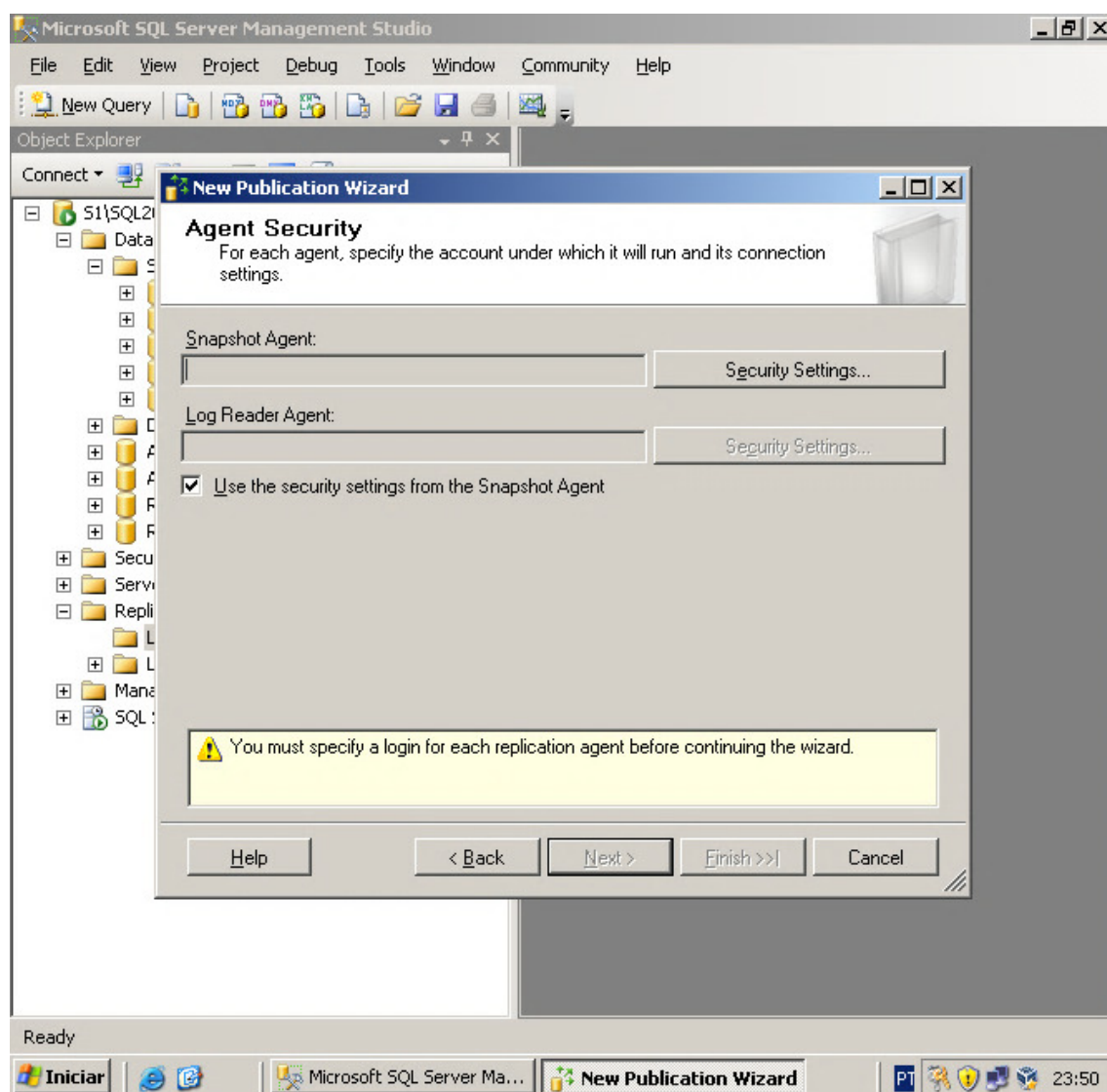


Figura 39: Configuração de Segurança do Snapshot Agent

Fonte: Resultado de pesquisa

Seguindo o padrão da figura 40, selecione **Run Under The SQL Server Agent Account**, selecione **By Impersonating The Process Account** e clique em **OK**.

Snapshot Agent Security

Specify the domain or machine account under which the Snapshot Agent process will run.

☐ Run under the following Windows account:

Process account:

Example: domain\account

Password:

Confirm Password:

☒ Run under the SQL Server Agent service account (This is not a recommended security best practice.)

Connect to the Publisher

☒ By impersonating the process account

☐ Using the following SQL Server login:

Login:

Password:

Confirm Password:

OK Cancel Help

Figura 40: Senhas de acesso para o Snapshot Agent

Fonte: Resultado de pesquisa

Não é recomendado o uso da conta de serviço do SQL Server Agent, porém, por ser um ambiente de simulação, não serão executadas configurações de segurança.

Na tela seguinte, ver figura 41, verifique se está marcada a opção **Create The Publication** e clique em *Next*.

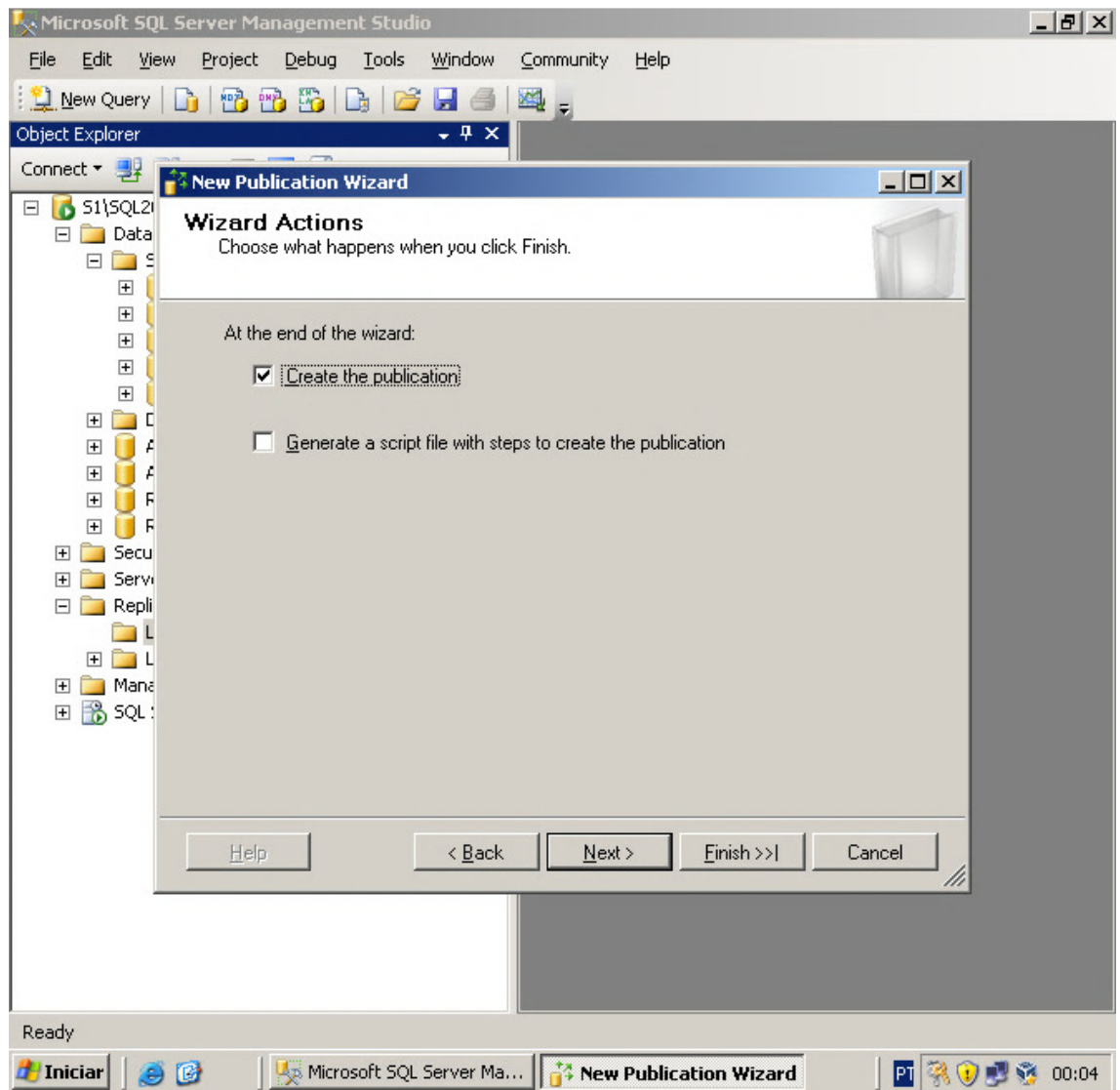


Figura 41: Confirmação da configuração do Publicador

Fonte: Resultado de pesquisa

Informe um nome para a publicação, ver figura 42, e, em seguida, clique em *finish*.

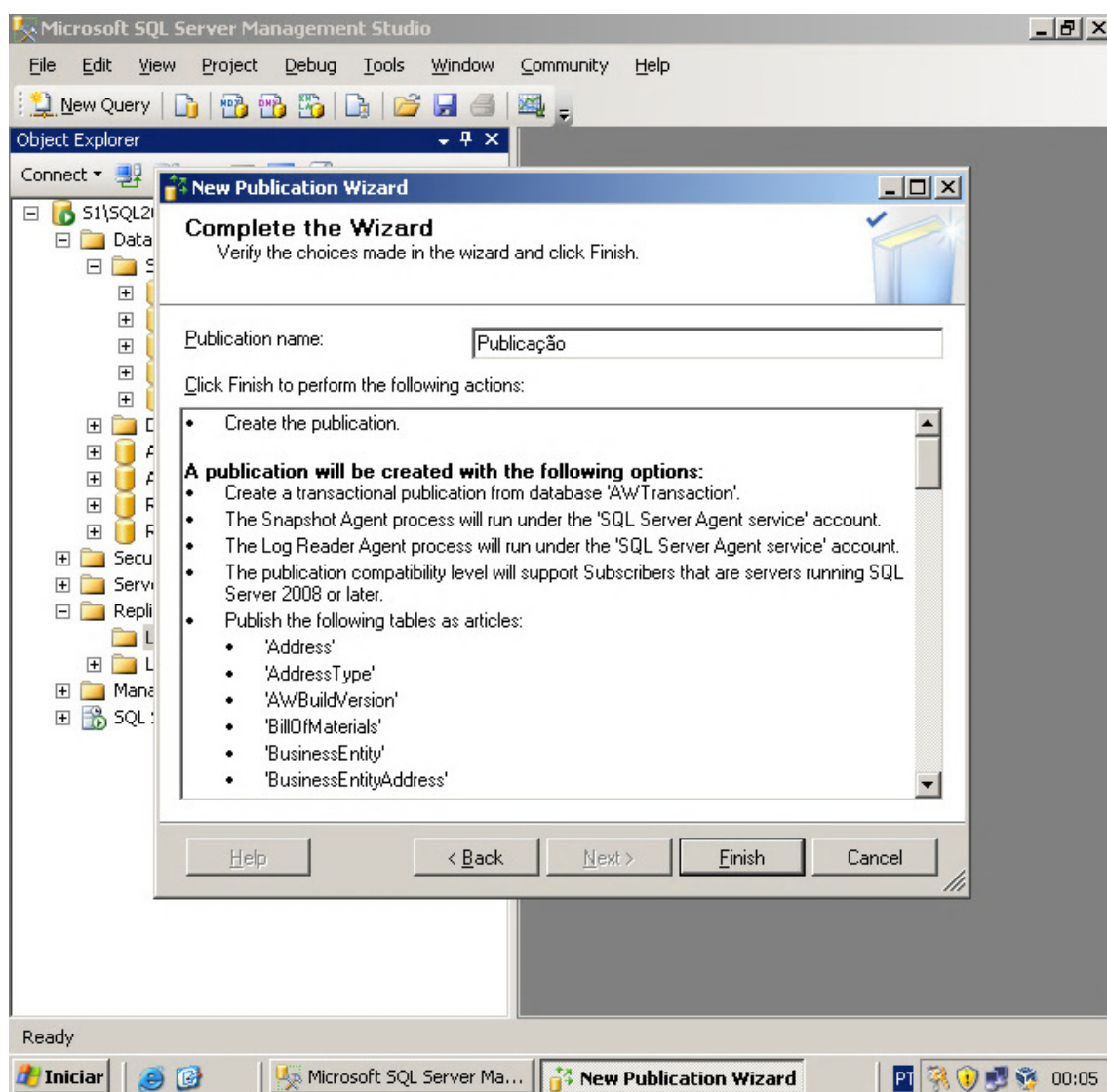


Figura 42: Finalização da configuração do Publicador

Fonte: Resultado de pesquisa

Após concluir a criação da publicação clique em *close*.

4.1.6. Configurando um Assinante

Na máquina virtual **S2**, no **SQL Server Management Studio** expanda **replication**. Clique com o botão direito do mouse em **Local Subscription** e escolha **New Subscription**, ver figura 43.

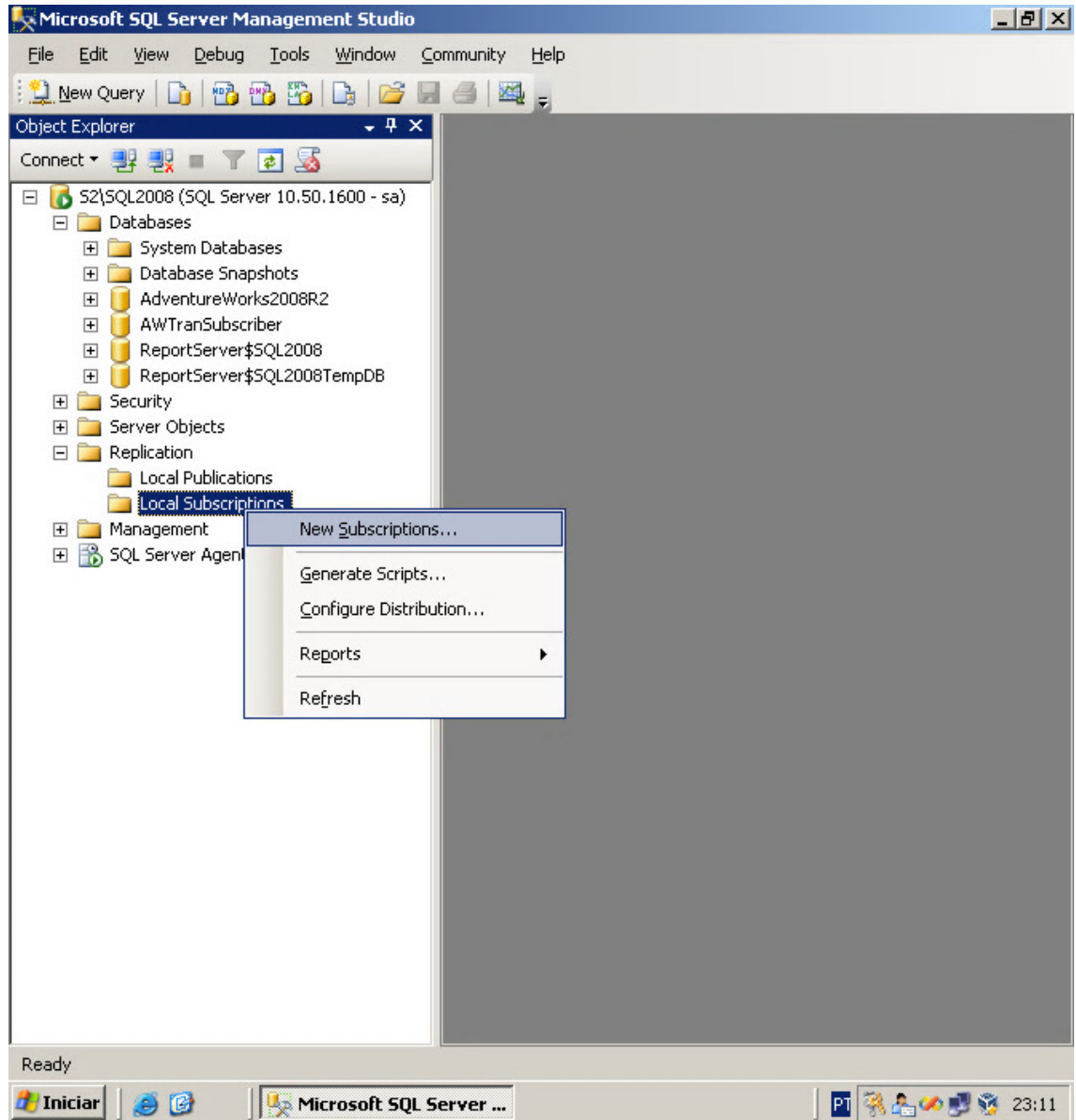


Figura 43: Configuração de um Assinante

Fonte: Resultado de pesquisa

Conforme figura 44, na primeira tela exibida clique em *Next*.

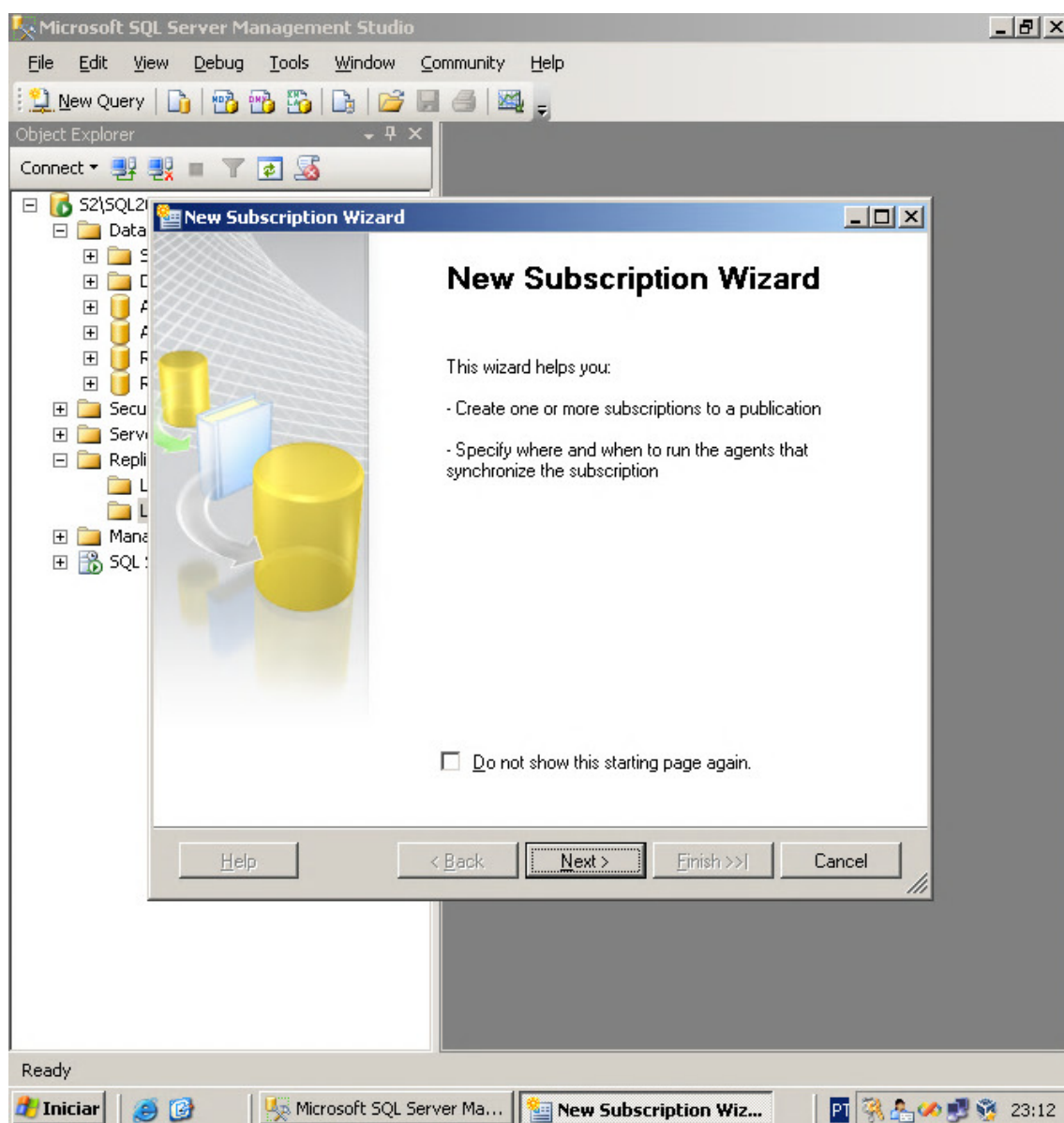


Figura 44: Configuração inicial do Assinante

Fonte: Resultado de pesquisa

Nesta tela, selecione a **Publicação** configurada no **S1**, conforme figura 45.

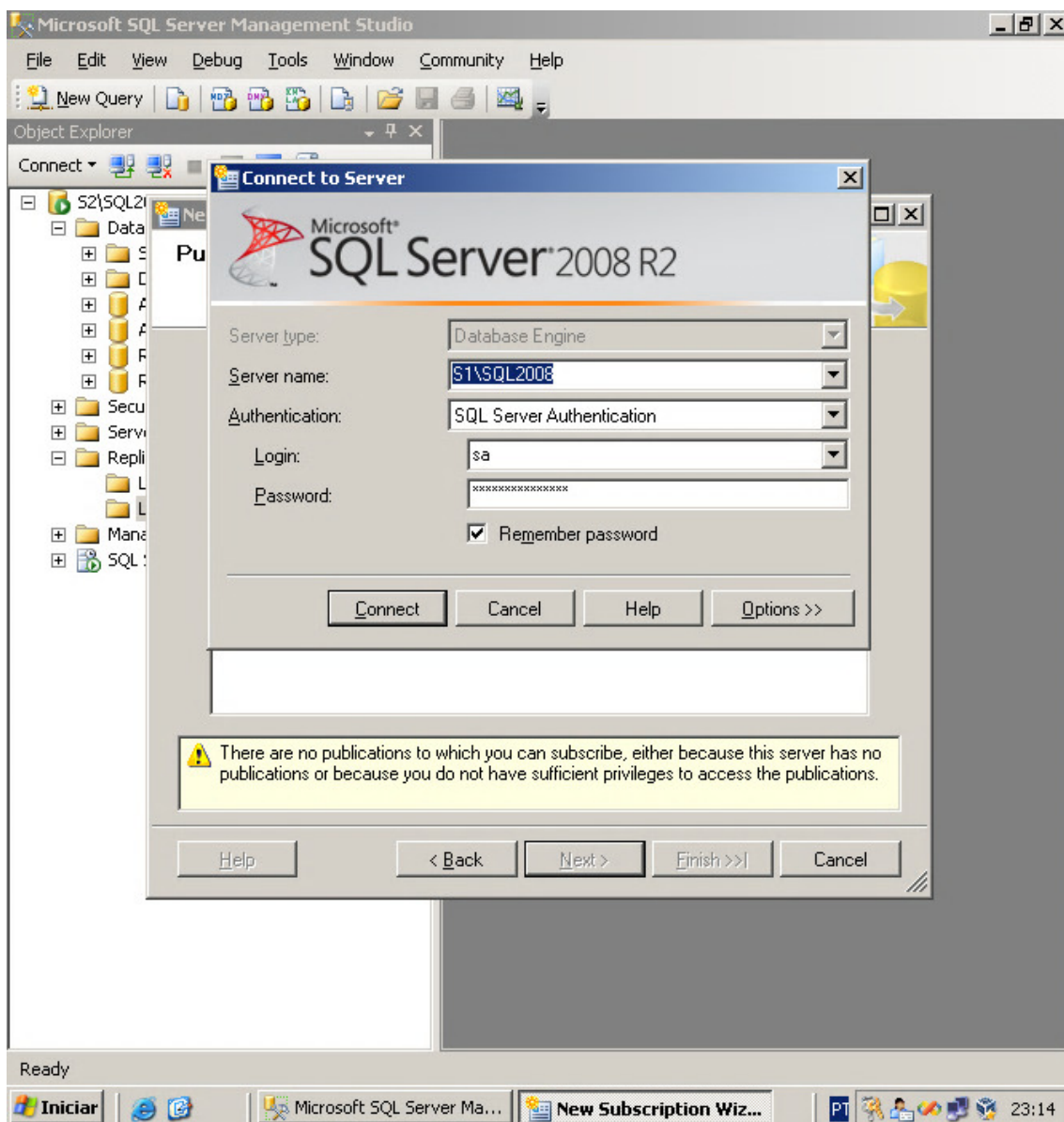


Figura 45: Configuração da publicação padrão

Fonte: Resultado de pesquisa

Na figura 46 podemos ver a confirmação da publicação selecionada.

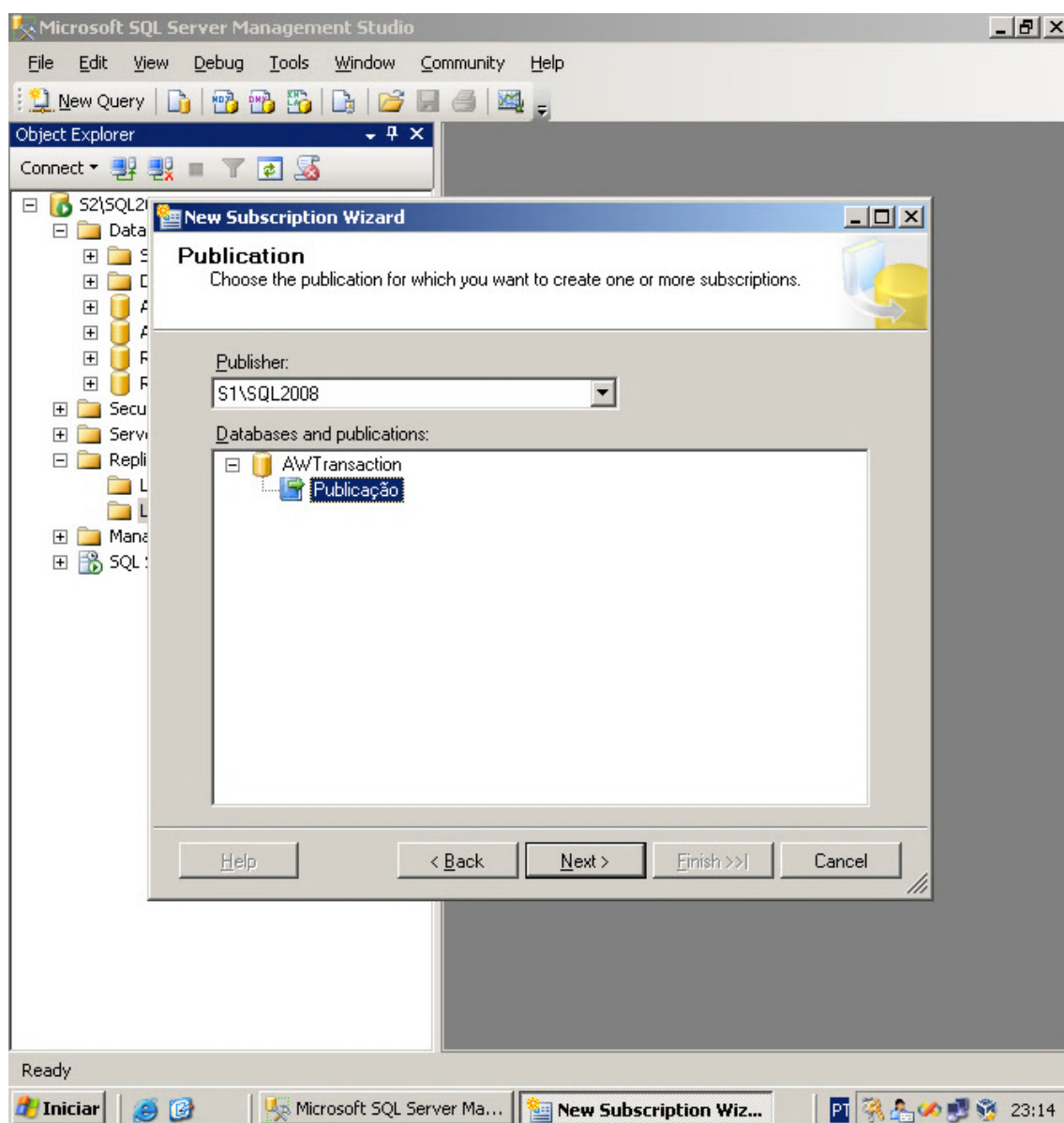


Figura 46: Confirmando a publicação padrão

Fonte: Resultado de pesquisa

Na tela seguinte, conforme figura 47, verifique se a opção **Run All Agents The Distribuidor** está marcada e clique em **Next**.

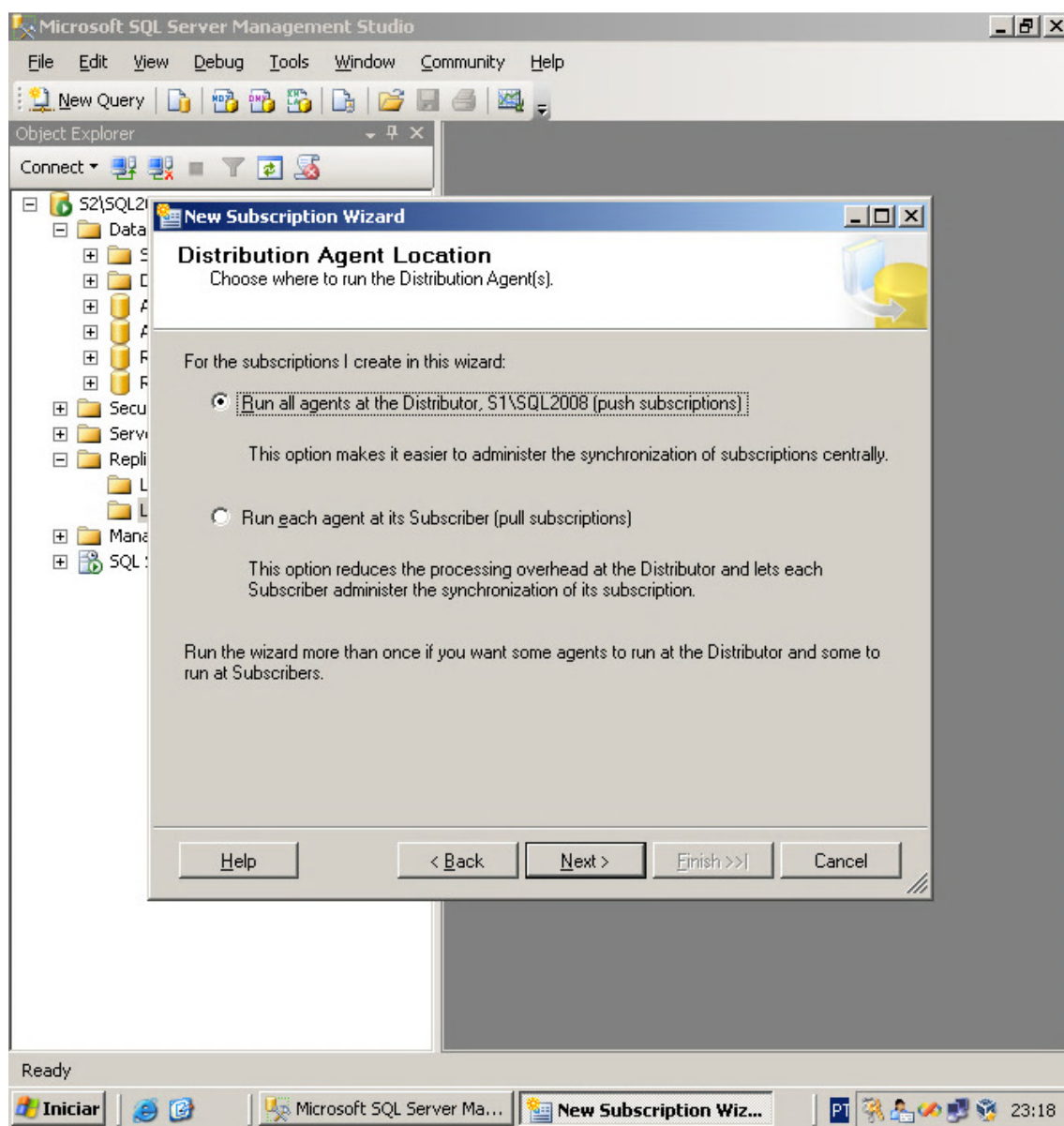


Figura 47: Configuração do Agent local de Distribuição

Fonte: Resultado de pesquisa

Conforme figura 48, deixe marcada a caixa de seleção da instância principal (S2\SQL2008) e selecione o banco de dados **AWTransSubscriber**.

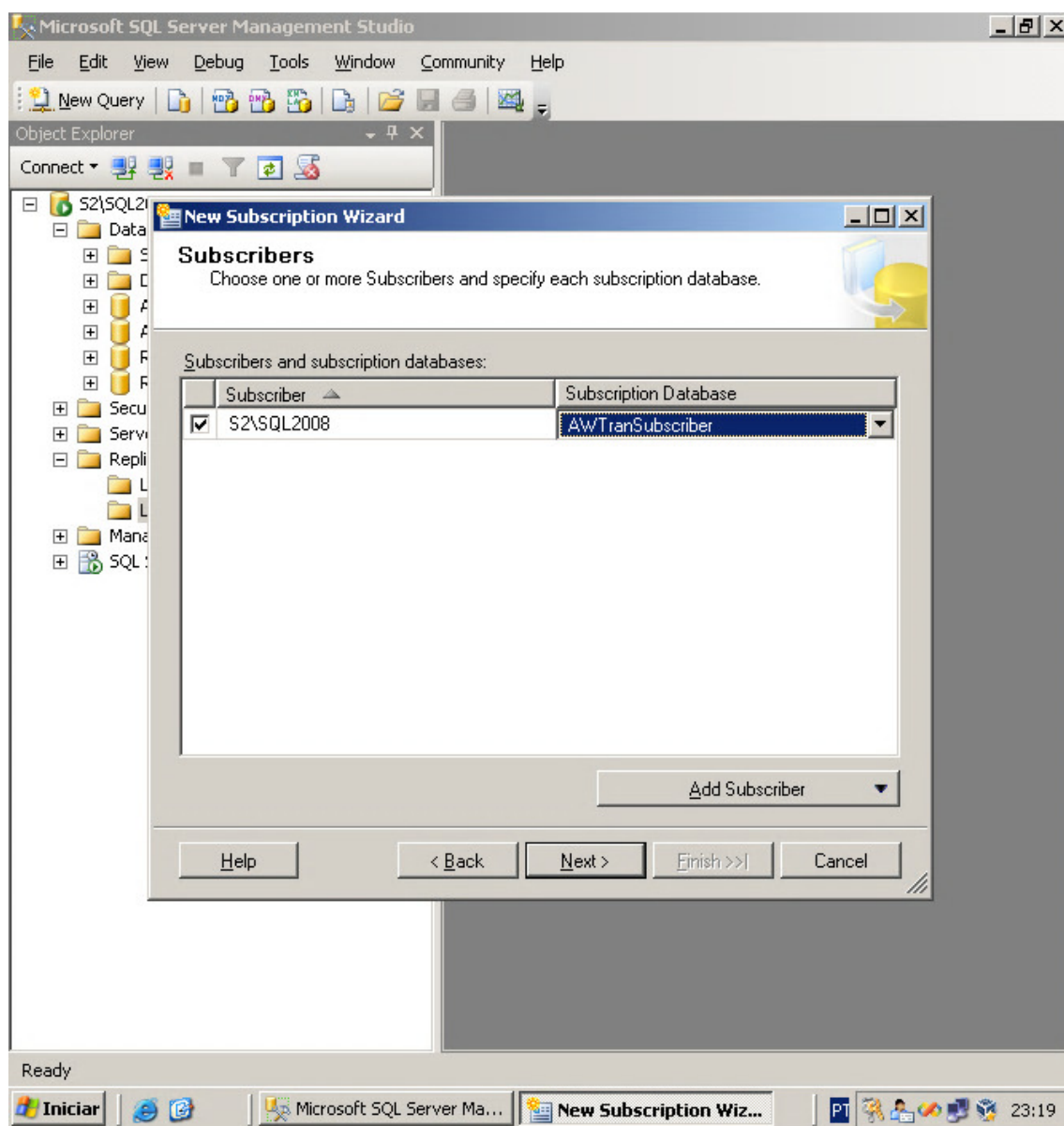


Figura 48: Configuração da instância e banco de dados do assinante

Fonte: Resultado de pesquisa

Na tela seguinte clique no botão de reticências, ver figura 49.

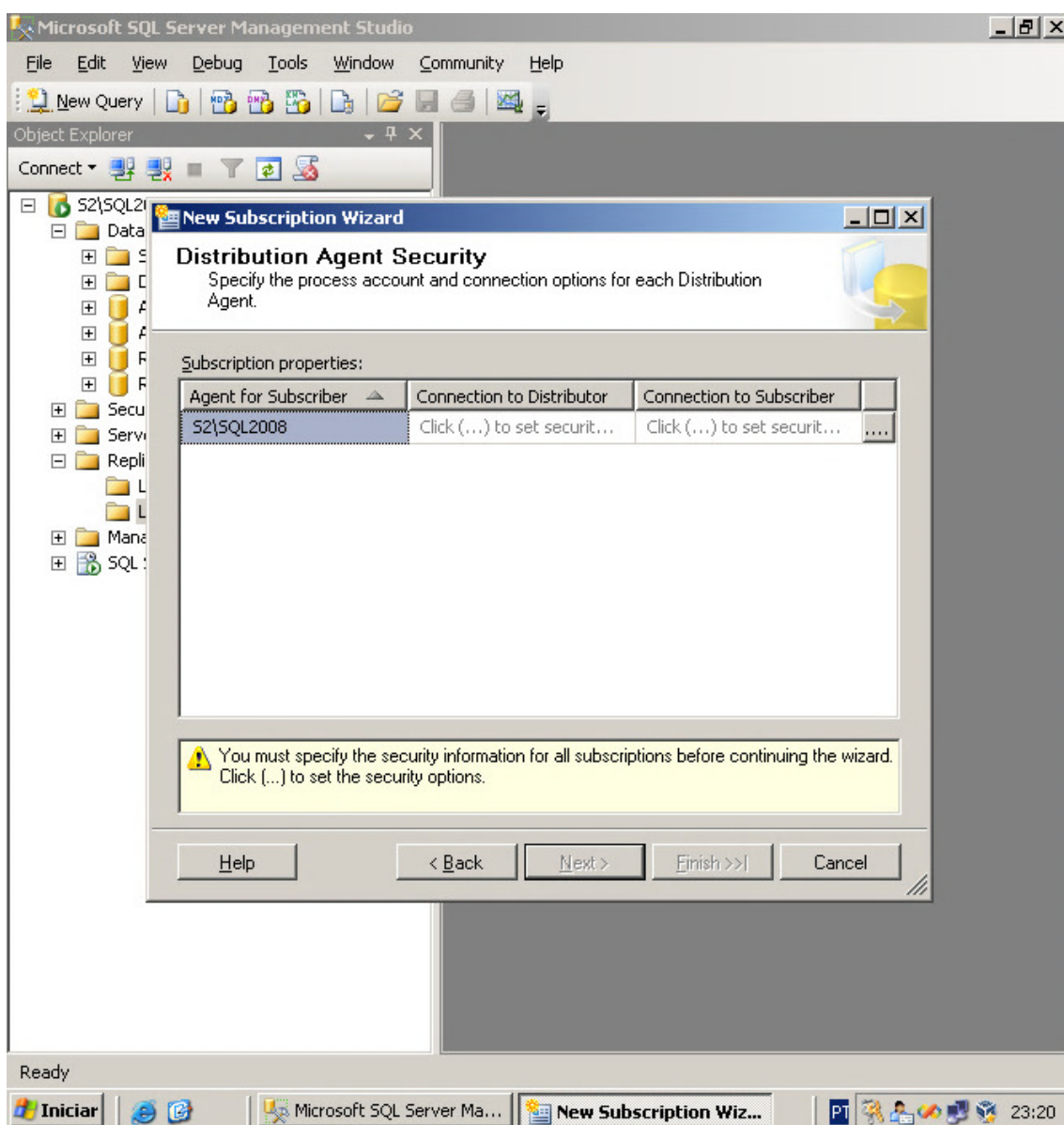


Figura 49: Configuração de Segurança do Distribution Agent

Fonte: Resultado de pesquisa

Configure a tela conforme opções da figura 50. Clique em **OK** e em seguida clique em **Next**.

Distribution Agent Security

Specify the domain or machine account under which the Distribution Agent process will run when synchronizing this subscription.

☐ Run under the following Windows account:

Process account:
Example: domain\account

Password:

Confirm Password:

☒ Run under the SQL Server Agent service account (This is not a recommended security best practice.)

Connect to the Distributor

☒ By impersonating the process account

☐ Using a SQL Server login

The connection to the server on which the agent runs must impersonate the process account. The process account must be a member of the Publication Access List.

Connect to the Subscriber

☒ By impersonating the process account

☐ Using the following SQL Server login:

Login:

Password:

Confirm password:

The login used to connect to the Subscriber must be a database owner of the subscription database.

OK Cancel Help

Figura 50: Senhas de acesso para o Distribution Agent

Fonte: Resultado de pesquisa

Na página de agendamento de sincronização deixa a opção **Run Continuously**, que é o padrão, configurada, conforme figura 51, e, em seguida, clique em *Next*.

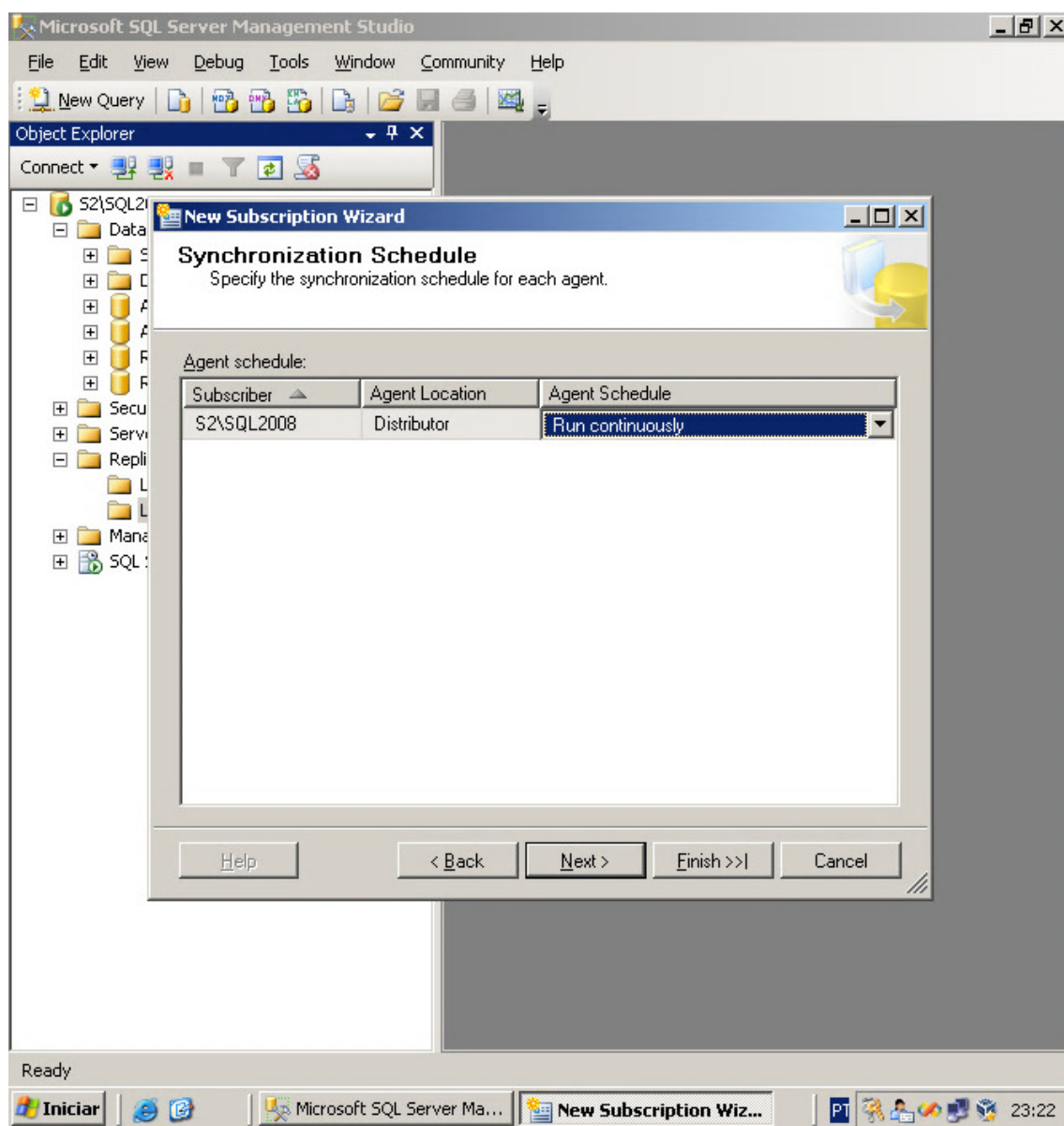


Figura 51: Configurando agendamento de sincronização

Fonte: Resultado de pesquisa

Deixe a opção **Initialize When** com a opção padrão, que é **Immediately**, conforme figura 52, e, em seguida, clique em *Next*.

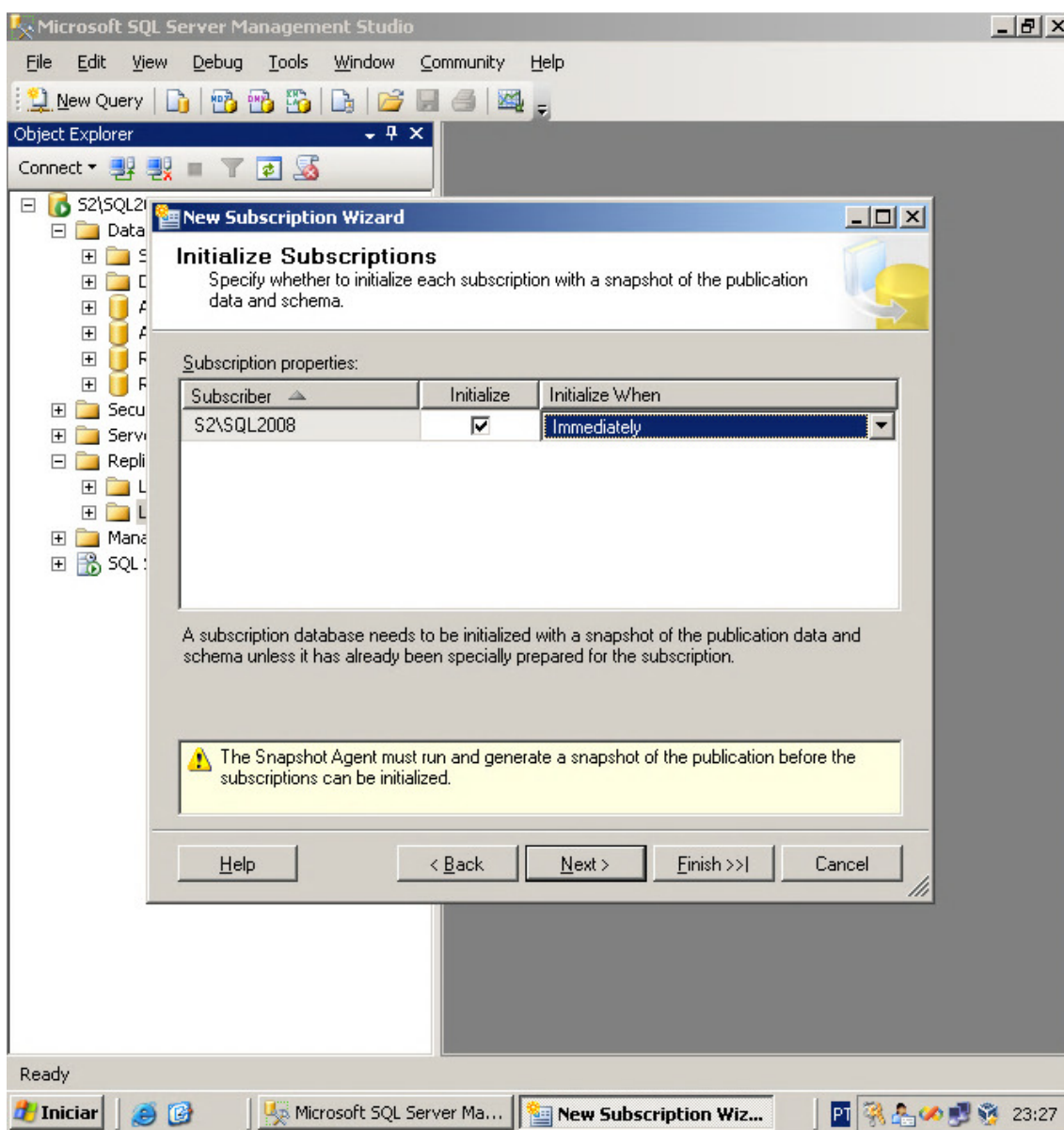


Figura 52: Configurando inicialização da assinatura

Fonte: Resultado de pesquisa

Verifique se a caixa de seleção **Create Subscription(s)** está marcada, conforme figura 53, e, em seguida, clique em **Next**.

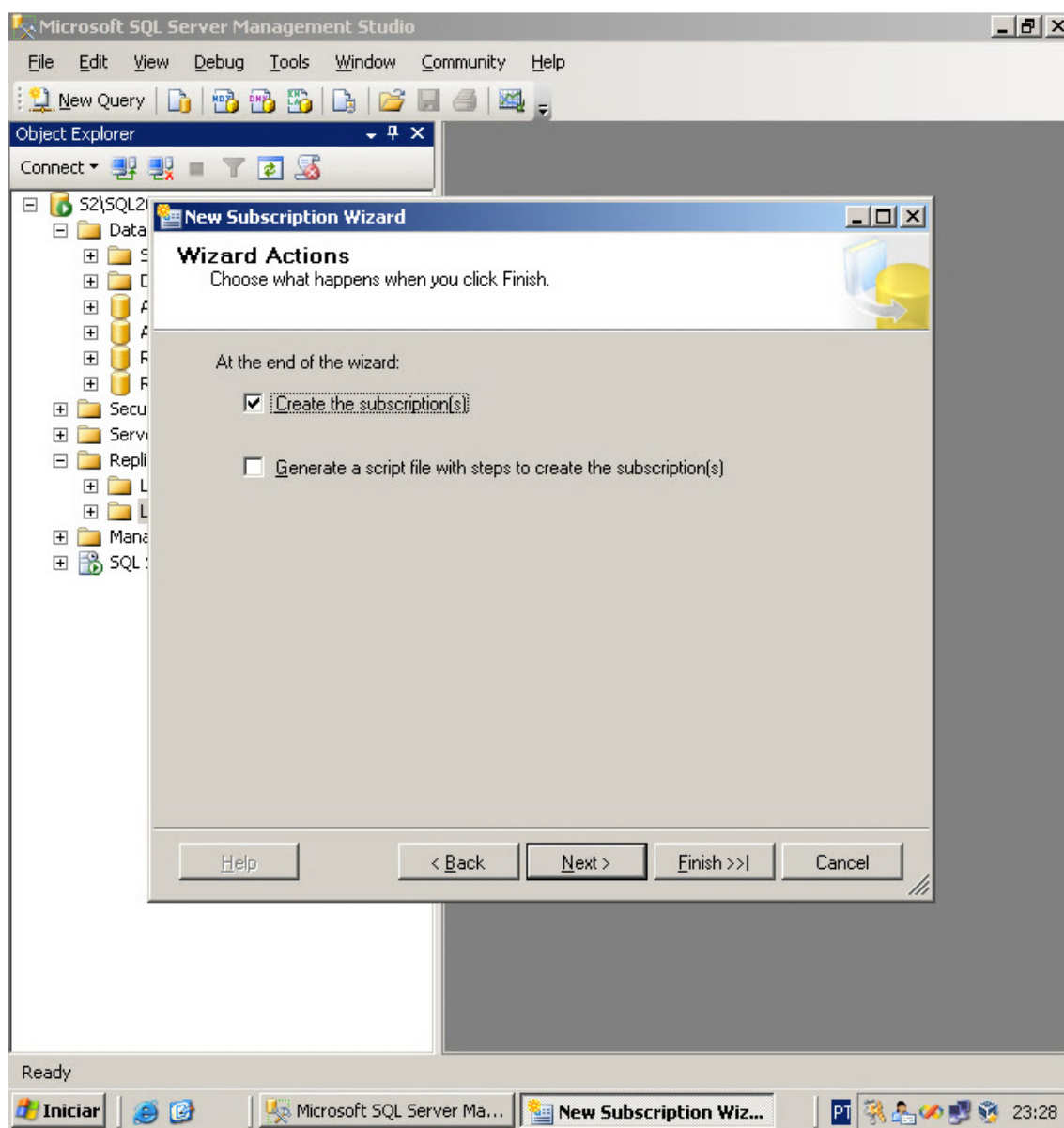


Figura 53: Confirmação da configuração do Assinante

Fonte: Resultado de pesquisa

Na tela seguinte clique em **Finish** e quando a assinatura estiver criada clique em **close**.

4.1.7. Testando a replicação

Na máquina virtual S1 abra o **SQL Server Management Studio** e crie uma nova *query*. Execute o comando: **select * from Production.Culture** no banco de dados **AWTransaction**. O resultado será o mesmo mostrado na figura 54.

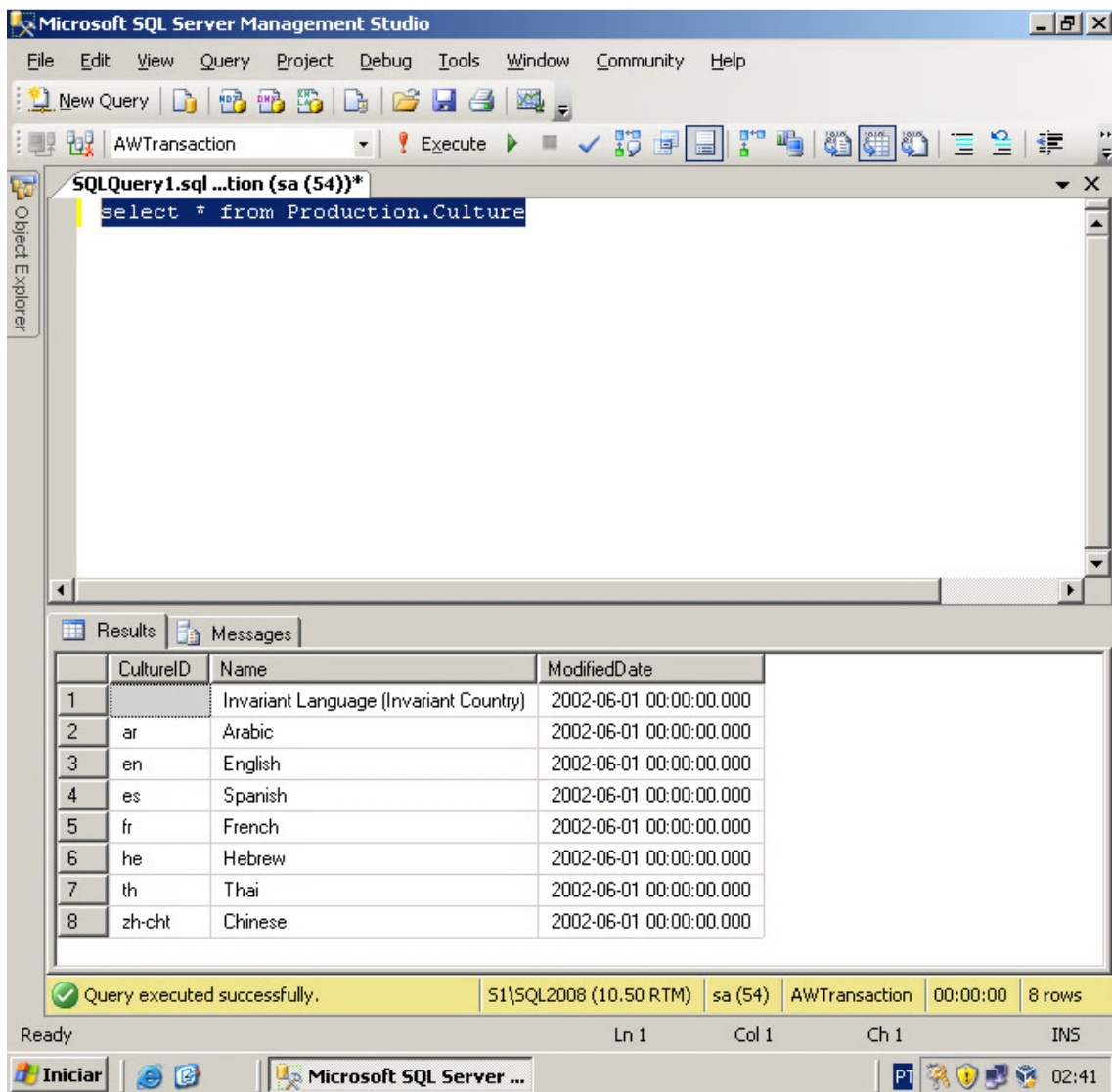


Figura 54: Selecionando registros na tabela Culture

Fonte: Resultado de pesquisa

Execute o mesmo procedimento na máquina virtual S2, porém, no banco de dados **AWTranSubscriber**. O resultado da query deve ser o mesmo.

Execute o comando **INSERT INTO Production.Culture (CultureID, Name, ModifiedDate) VALUES ('TT','TESTE DE REPLICACAO',GETDATE())** na query que está aberta na máquina virtual S1, conforme imagem 55.

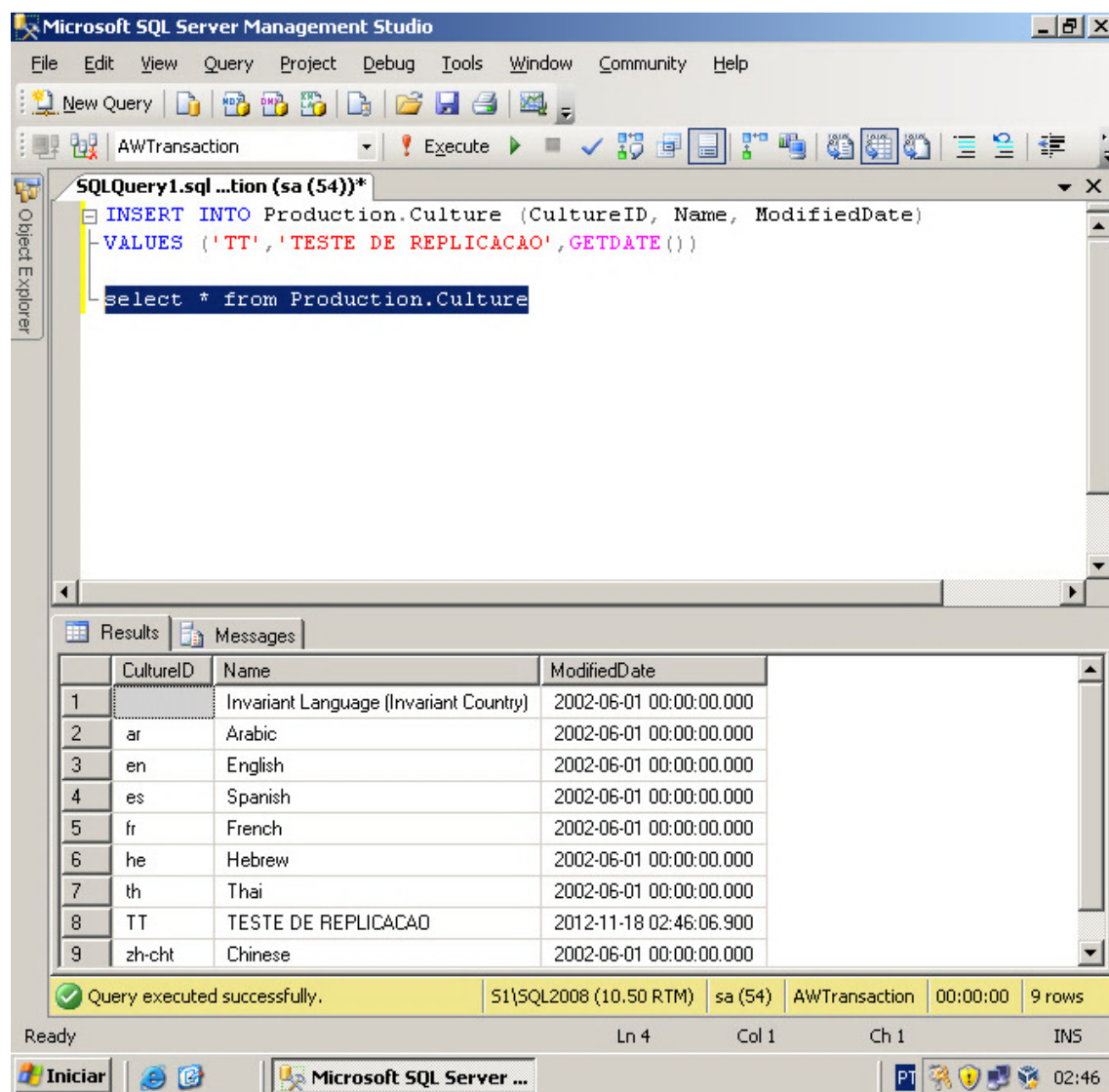


Figura 55: Inserindo registros na tabela Culture

Fonte: Resultado de pesquisa

Execute o comando **select * from Production. Culture** na query que está aberta na máquina virtual S2. O registro ‘TESTE DE REPLICACAO’ deve estar presente na tabela **Culture** do banco de dados **AWTranSubscriber**, conforme observamos na figura 56.

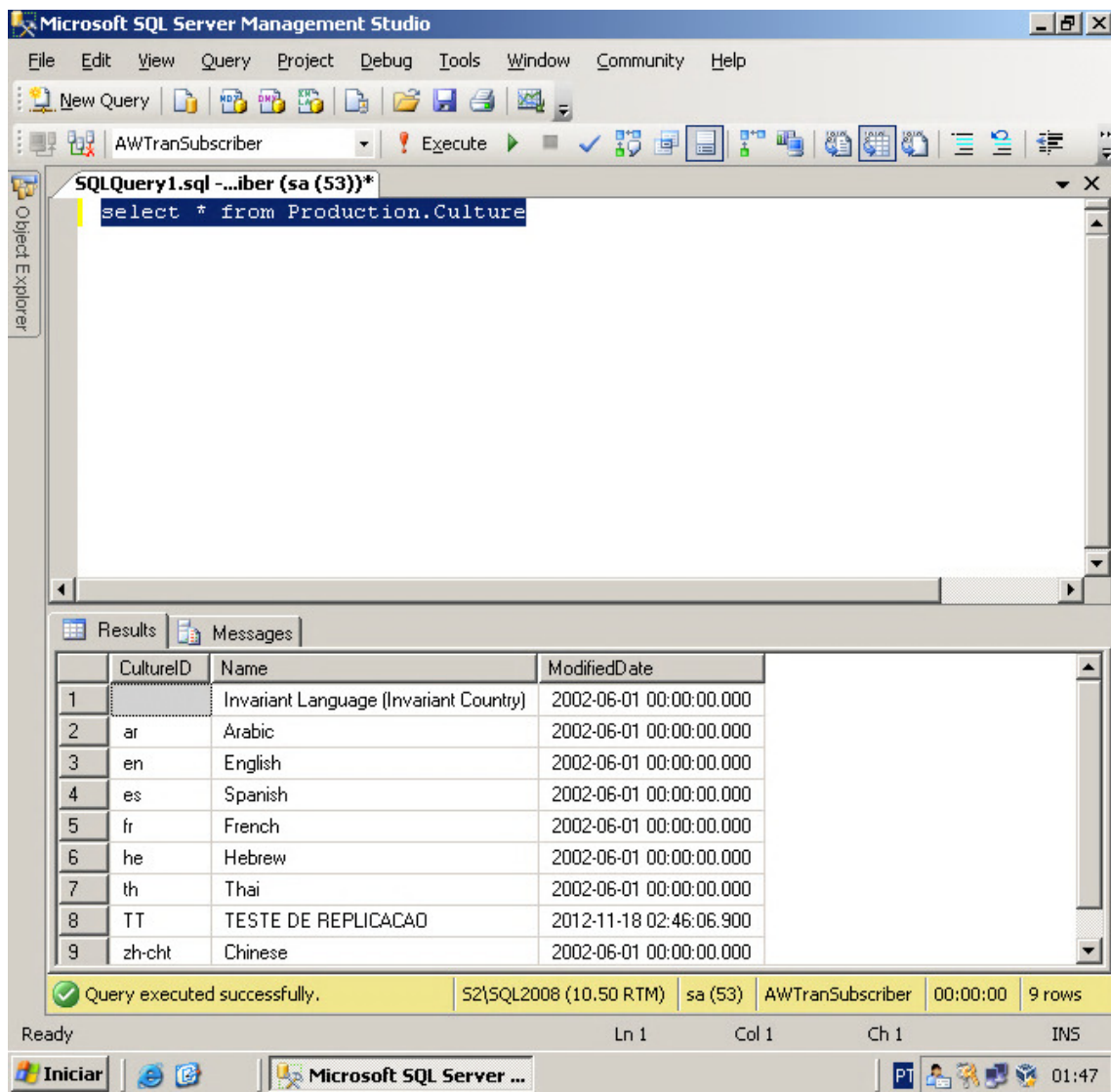


Figura 56: Conferindo replicação de dados

Fonte: Resultado de pesquisa

5. CONCLUSÃO

No mundo corporativo as informações são armazenadas em banco de dados originados por sistema de informações. Estas informações são extremamente necessárias e auxiliam desde os gestores das empresas até mesmo o corpo operacional da organização. As informações contidas nestas bases de dados geram relatórios gerenciais que auxiliam a tomada de decisão estratégica e que garantem uma visão futura de mercado gerando novas expectativas de investimentos. Estas informações também fazem funcionar de forma organizada todo o universo operacional que depende destes dados para realização de compras de produtos, vendas de produtos, controle de pedidos, etc..

Estando o mundo corporativo tão dependente destas informações, o conceito de alta disponibilidade deve ser um fator primordial para qualquer organização. Implantar uma solução de alta disponibilidade envolve custos que muitas vezes são tidos como desnecessários por alguns gestores, porém, a paralização dos serviços pode gerar perda de valores inestimáveis. Implantar uma solução de alta disponibilidade está diretamente associada a continuidade de negócio, de uma forma bem planejada, pode garantir a continuidade quase que 24 horas dos serviços.

Existem diversas soluções de alta disponibilidade no mercado, porém, a replicação de dados é uma excelente alternativa. Esta solução permite a criação de bancos de dados idênticos em um ou mais servidores e garante a redundância dos dados.

Mediante ao cenário, este trabalho teve como objetivo o estudo de uma solução de replicação para criação de alta disponibilidade utilizando a ferramenta Microsoft SQL Server 2008. Inicialmente foram abordadas os conhecimentos específicos em banco de dados, alta disponibilidade e replicação. Em seguida foi configurada uma replicação de dados do tipo transacional em um ambiente virtual que buscou simular um ambiente real onde os dados são mantidos em redundância.

Podemos concluir que a solução oferecida pela Microsoft para replicação de dados é uma ferramenta que pode assegurar a constante operação dos negócios da empresa e garantir o funcionamento constante do sistema de informação gerando uma redundância necessária nos dados. A ferramenta SQL Server se mostra também de fácil configuração e com ela é possível garantir de forma eficiente que a replicação de dados estará em constante funcionamento.

6. BIBLIOGRAFIA

BATTISTI, Júlio; SQL Server 2005 Administração e Desenvolvimento Curso Completo, Axcel Books , 2006.

BORGES, Lincoln; Entendendo o SGBD, Disponível em <http://www.tron.com.br/blog/2010/04/entendendo-o-sgbd-sistema-gerenciador-de-banco-de-dados/> acessado em 17 de outubro de 2012.

CLARINDO, Rodrigo; Modelos de rede, Disponível em <http://rodrigoclf.blogspot.com.br/> acessado em 17 de outubro de 2012.

DATE, C. J.; Introdução a Sistemas de Banco de Dados, Rio de Janeiro, Ed. Campus, 2000.

ELMASRI, Ramesz; NAVATHE Shamkant B.; Sistemas de banco de dados, Pearson Education, 2010.

HOPKINS, Matt; THOMAS, Dunstan; Replicação de Dados com InterBase/Firebird, Disponível http://www.comunidade-firebird.org/cflp/downloads/CFLP_T022.PDF acessado em 17 de outubro de 2012.

HOTEK, Mike; Exam 70-432: Microsoft SQL Server 2008 Implementation and Maintenance, Microsoft Press, 2009.

INÁCIO, Alexandre; Alta Disponibilidade, Disponível <http://desmontacia.wordpress.com/2011/01/02/alta-disponibilidade/> acessado em 17 de outubro de 2012.

MELO, Peterson; Desenvolvimento de um Sistema de Replicação de Dados entre Bancos de Dados Relacionais, Natal, 2010.

MICROSFT; Visão geral do Microsoft SQL Server, Disponível em <http://www.microsoft.com/sqlserver/pt-br/default.aspx> acessado em 17 de outubro de 2012.

MSDN; Compreendendo páginas e extensões. Disponível em: [http://msdn.microsoft.com/pt-br/library/ms190969\(v=sql.105\).aspx](http://msdn.microsoft.com/pt-br/library/ms190969(v=sql.105).aspx). acessado em 15 de outubro de 2012.

MSDN; Planejamento e arquitetura (Mecanismo de Banco de Dados), Disponível em: [http://msdn.microsoft.com/pt-br/library/cc280361\(v=sql.105\).aspx](http://msdn.microsoft.com/pt-br/library/cc280361(v=sql.105).aspx) acessado em 16 de outubro de 2012.

MSDN; Selecionando uma solução de alta disponibilidade, Disponível em: [http://msdn.microsoft.com/pt-br/library/bb510414\(v=sql.105\).aspx](http://msdn.microsoft.com/pt-br/library/bb510414(v=sql.105).aspx) acessado em 16 de outubro de 2012.

MSDN; The Evolution of Microsoft SQL Server: 1989 to 2000, Disponível em: <http://www.insidesqlserver.com/companion/History%20of%20SQL%20Server.pdf> acessado em 15 de outubro de 2012.

MSDN; Visão geral (Replicação), Disponível em: [http://msdn.microsoft.com/pt-br/library/bb500346\(v=sql.100\).aspx](http://msdn.microsoft.com/pt-br/library/bb500346(v=sql.100).aspx) acessado em 16 de outubro de 2012.

MSDN; Visão geral das soluções de alta disponibilidade. Disponível em: [http://msdn.microsoft.com/pt-br/library/ms190202\(v=sql.105\).aspx](http://msdn.microsoft.com/pt-br/library/ms190202(v=sql.105).aspx) acessado em 15 de outubro de 2012.

NATÁRIO, Ruy; Alta Disponibilidade - Terminologia (I), Disponível em <http://redes-e-servidores.blogspot.com.br/2011/02/alta-disponibilidade-terminologia.html> acessado em 17 de outubro de 2012.

PENHA, Marcos; Modelo Entidade-Relacionamento, Disponível em http://marcospenhaschuler.wordpress.com/2011/03/17/modelo_entidade-relacionamento/ acessado em 17 de outubro de 2012.

RAMOS, Wagner; Conceitos Gerais sobre Replicação de Banco de Dados. Disponível em: <http://www.object.com.br/wikiold/ConceitosGerais> acessado em 16 de outubro de 2012.

REIS, Eugênio; Transações. Disponível em: <http://www.delphibr.com.br/artigos/transacoes.php> acessado em 16 de outubro de 2012.

ROB, Peter; CORONEL, Carlos; Sistemas de Banco de Dados, projeto, implementação e administração. São Paulo, Ed. Cengage Learning, 2011.

SOLID QUALITY LEARNING, Microsoft SQL Server 2005: Database Essentials Step by Step, Microsoft Press, 2006.

STANEK, William R.; Guia de Bolso para Administradores do Microsoft SQL Server 2008, Bookman, 2010.

SZTOLTZ, Lisiane; TEIXEIRA, Roberto Selbach; RIBEIRO, Evelyne de Oliveira Ferraz; Guia do Servidor Conectiva Linux, Conectiva S.A., 2003.

TANENBAUM, Andrew; STEEN, Maarten; Sistemas Distribuídos: princípios e paradigmas, Pearson Education, 2006.

TRIPOLI, Tarcísio. Disponível em: <http://ttmaia.files.wordpress.com/2011/03/versoes-sql-server.png> acessado em 30 de novembro de 2012.