Centro Universitário - IESB

BANCO DE DADOS DISTRIBUÍDOS

Um Estudo de Caso em um banco de dados Homogêneo Distribuído visando a Alta Disponibilidade de Dados

Matrícula: 0931004012 - Sidnei Fernandes das Neves

Brasília-DF
Abril de 2012

Sidnei Fernandes das Neves

BANCO DE DADOS DISTRIBUÍDO

Um Estudo de Caso em um banco de dados Homogêneo Distribuído visando à alta disponibilidade de Dados

Trabalho de Conclusão de Curso apresentado ao Curso de Pós-graduação com ênfase em SGBD Oracle do Centro Universitário - IESB, como requisito para a obtenção do grau de pós-graduado, sob a orientação do professor Eder Couto.

Brasília-DF
Abril de 2012

Sidnei Fernandes das Neves

BANCO DE DADOS DISTRIBUÍDO

Um Estudo de Caso em um banco de dados Homogêneo Distribuído visando à alta disponibilidade de Dados

Trabalho de Conclusão de Curso apresentado e aprovado em 13 de abril de 2012, pela banca examinadora constituída pelos professores:

_____

Prof. Eder Couto - Orientador

_____

Prof. Esp. Paulo Lima Machado - Examinador

_____

Prof. Esp. André Nagy - Examinador

AGRADECIMENTOS

Agradeço a todos os professores que se empenharam na ministração do curso de Pós-Graduação em Banco de Dados no Centro Universitário - IESB, os quais me ajudaram a desenvolver e agregar conhecimento suficiente para que eu pudesse prosseguir com as pesquisas e desenvolver este Trabalho de Conclusão de Curso.

A minha esposa, Letícia Lins, a qual tem me dado toda força e incentivo para continuar nesta empreitada.

A uma pessoa espetacular, que tive a honra de conhecer no Centro Universitário – IESB, chamado André Baracho, que foi uma das pessoas que se dedicou bastante em me apoiar nos momentos de dúvidas, se tornando uma peça chave para que eu obtivesse êxito neste trabalho.

Aos Diretores da Object Sistemas, pelo apoio e confiança depositados em mim para se tornasse possível a conclusão deste trabalho.

Além destes, não poderia me esquecer das pessoas que fazem parte minha vida e que tanto contribuíram com meu ensino básico, sem o qual eu não teria conseguido chegar ate aqui, são eles, meu pai, Paulo Sérgio e minha segunda mãe, Antonia Estelita.

RESUMO

Com o avanço tecnológico voltado para as redes de computadores e com as facilidades oferecidas por elas, é cada vez maior a quantidade de empresas que adéquam sua estrutura para um funcionamento em rede fazendo com que nenhum computador fique simplesmente isolado.

Em um ambiente corporativo é essencial que haja alta disponibilidade de acesso aos dados através de uma rede, mas somente o acesso não basta, tem que existir a garantia de disponibilidade desses dados, o que pode ter como uma alternativa de solução a distribuição do banco de dados em vários servidores, dentro de uma mesma empresa ou em diversas empresas distribuídas por várias localidades geográficas.

Este trabalho tem como objetivo principal demonstrar como funciona uma estrutura para a distribuição de dados em diversos Sistemas de Gerenciamento de Banco de Dados, através de redes computacionais, bem como expor quais são os benefícios e as perdas em se eleger uma estrutura como esta para se trabalhar. Serão expostas, ainda, algumas formas de se consolidar os dados distribuídos e os prós e contras de cada metodologia.

Todo o estudo é voltado para um problema real de Gestão de Continuidade do Negócio de uma empresa de Previdência Complementar, a qual, neste trabalho terá o nome fictício de PREVIDAS. Todas as soluções e análises ocorrerão em laboratório (Máquinas Virtuais).

ABSTRACT

With technological advances toward computer networks and the facilities offered by them is increasing the number of companies that adapt their structure to a network operation so that no computer is simply isolated.

In a corporate environment is essential to have high availability data access across a network, but only access is not enough, there must be a guarantee of availability of these data, which may have an alternative solution to the distribution database across multiple servers within a single company or several companies spread across multiple geographic locations.

This paper aims to demonstrate how a main structure for the distribution of data in various Management Systems Database, via computer networks, as well as expose what are the benefits and losses to be elected a structure like this to work. Will be exposed, still some ways to consolidate the data spread and the pros and cons of each methodology.

The whole study is a real problem facing the Management of a Business Continuity in a Pension Funds, which, in this work will have the fictitious name to PREVIDAS. All solutions and analysis will occur in the laboratory (Virtual Machines).

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AD – Administrador de Dados.

API – (*Application Programming Interface*) – Interface de Programação de Aplicações.

ARCn – (*Archiver Process*) – Processo de arquivamento.

BDD – Banco de Dados Distribuído.

BI – Business Intelligence.

CKPT – (*Ckecpoint Process*) – Marcador de pontos de verificação de processo.

CPU – (*Central Processing Unit*) – Unidade Central de Processamento.

DBLink – Data Base Link

DBWn – (Database Write Process) – Processo de escrita na base de dados.

DCL – (Data Control Language) – Linguagem de Controle de Dados.

DDL – (Data Definition Language) – Linguagem de Definição de Dados.

DML – (Data Manipulation Language) – Linguagem de Manipulação de Dados.

E/S – Entrada e Saída.

GB – (*Gigabytes*).

GHz – (*Gigahertz*).

IO – (*Input/Output*) - A mesma definição de E/S.

LGWR – (Log Write) – Escrita de Log.

MB – (Megabytes).

Mbps – Megabits por segundo.

MER – Modelo Entidade Relacionamento.

NTP – (*Network Time Protocol*) – Protocolo de hora da rede

OLTP – (*Online Transaction Processing*) – Processamento de transações em tempo real.

PMON – (*Process Monitor Process*) – Processo monitor de processos.

RAM – (*Random Access Memory*) – Memória de acesso randômico.

RPM – Rotações por minuto.

SBDD – Sistema de Banco de Dados Distribuído.

SGA – System Global Area.

SGBD – Sistema de Gerenciamento de Banco de Dados.

SGBDD – Sistema de Gerenciamento de Banco de Dados Distribuído.

SMON – (*System Monitor Process*) – Processo monitor do sistema.

SQL – (*Structured Query Language*) – Linguagem estruturada de pesquisa.

TI – Tecnologia da Informação.

WEB – Rede mundial de computadores.

SUMÁRIO

1. INTRODUÇÃO

Este trabalho tem como objetivo demonstrar o funcionamento de um banco de dados distribuído e explicar os seus principais conceitos, bem como compará-lo ao modelo centralizado, expondo as vantagens e desvantagens em se utilizar esta tecnologia e demonstrar a sua utilização em um caso de uso de uma empresa de Previdência Complementar.

Um Sistema de Banco de Dados Distribuído (SBDD) nada mais é do que uma coleção de múltiplos bancos de dados homogêneos ou heterogêneos, logicamente inter-relacionados distribuídos através de uma rede de computadores e gerenciado por um Sistema de Gerenciamento de Banco de Dados Distribuído (SGBDD) que é responsável por tornar a distribuição transparente para todos os usuários.

Este modelo traz a vantagem do processamento distribuído, através do qual vários computadores (nodos) podem dividir um grande e intratável problema em partes menores e menos complexas e os resolvem eficientemente de maneira coordenada.

Outra grande vantagem é a possibilidade de descentralização dos dados, o que proporciona alta disponibilidade, pois mesmo que um dos servidores do SBDD seja colocado fora de operação não indicará a falta de acesso aos dados, pois estes estarão distribuídos total ou parcialmente de forma confiável entre os outros nodos existentes.

Esta tecnologia surgiu a partir da fusão da tecnologia de Banco de Dados e a tecnologia de Rede e Comunicação de Dados, pois nos anos 70 e início dos anos 80 a idéia que se tinha era de centralizar os dados, o que resultou em bancos monolíticos e gigantescos, que consequentemente, exigiam muito mais poder de processamento. Este cenário logo se inverteu e ao final dos anos 80 a tendência passou a ser descentralizar os bancos de dados para haver autonomia no processamento dos dados, tendo vários reflexos positivos tais como a alta confiabilidade, disponibilidade e melhoria de desempenho.

1.1 CONTEXTUALIZAÇÃO

Atualmente, com o avanço da tecnologia, é exigida cada vez mais agilidade, segurança e alta disponibilidade nos processos, sejam de uma pequena empresa ou de uma multinacional. Isso tem sido um grande desafio para os profissionais de TI, os quais tem a obrigação de estarem sempre atentos às mudanças tecnológicas que auxiliem na árdua tarefa de melhoria dos processos no aspecto de eficiência, alta disponibilidade, segurança, confiabilidade e etc.

Um processo bem definido só traz bons resultados para as corporações, pois nas análises realizadas antes da implementação de uma solução, devem ser avaliados todos os aspectos positivos e negativos, bem como seus respectivos reflexos.

Já um processo mal definido pode levar uma empresa à falência, pois torna o trabalho tão ineficiente que pode prejudicar o negócio, ocasionando diversos problemas, tais como baixo tempo de resposta, falhas constantes, insegurança, o que pode gerar consequentemente a falta de confiabilidade. Todos estes problemas poderão ocasionar a perda de mercado para outras empresas mais estruturadas e preparadas para trabalhar com maior qualidade.

Com a utilização de um Banco de Dados Distribuído (BDD), vários dos fatores negativos são minimizados pelo fato deste se tornar mais eficiente dependendo da forma que for implementado, pois deixará de contar apenas com um banco de dados centralizado e trabalhará com vários bancos distribuídos em locais distintos, o que dá mais agilidade e segurança, pois quando um banco de dados não estiver disponível por algum tipo de falha, só este deverá ser tratado e os demais continuarão trabalhando normalmente.

Considerando que o volume diário de transações de uma empresa pode ser muito grande, é necessário elaborar a melhor solução para a atualização e replicação dos dados, pois o trafego na rede irá aumentar substancialmente, o que fará, dependendo da solução que for adotada, que as respostas às requisições feitas pelos usuários em suas estações de trabalho sejam muito lentas.

Por esse motivo, os profissionais de Tecnologia da Informação (TI) precisam adotar alternativas eficientes para realizar consultas e alterações dos dados sem que haja o comprometimento do desempenho da rede.

1.2 MOTIVAÇÃO E CARACTERIZAÇÃO DO PROBLEMA

Uma empresa precisa de acesso aos dados para poder trabalhar, pois sem eles os processos não conseguem seguir o fluxo normal.

Sendo assim, é necessário tornar mais estável e fácil o acesso e troca de dados entre uma empresa sede e seus sites alternativos, utilizando para este fim uma estrutura de rede computacional e os recursos disponibilizados pelo SGBDD.

Considerando que a empresa em questão atende a centenas de clientes diariamente, é indispensável que exista alta disponibilidade para acesso aos dados. Este é o principal motivo para a implementação de um banco de dados distribuído, pois mesmo que algum dos servidores deixe de funcionar, os dados sempre estarão disponíveis em outro servidor, fazendo assim com que o risco de perda dos dados seja minimizado e a disponibilidade dos dados maximizada, e que consequentemente os nodos que não tiveram falhas continuem operando normalmente.

1.2.1. Justificativa

A cada dia se torna mais evidente para as empresas que querem realmente crescer ou se manter no mercado, que tal objetivo é inalcançável se não houver um forte investimento em tecnologia, pois só assim haverá segurança e disponibilidade das informações a qualquer momento que se desejar, principalmente nos momentos estratégicos de negociações e atendimentos por exemplo.

1.2.2. Oportunidade

Analisando a justificativa acima é possível perceber que as empresas realmente deveriam investir em tecnologia, tendo em contrapartida aumento do volume de negócios, ora prejudicados pela falta de infra-estrutura. Este aumento de negociações se justifica, pelo fato dos processos passarem a fluir com maior rapidez e segurança após a implantação de uma nova estrutura mais sólida e confiável.

Em fase de projeto, para criar um cenário próximo ao real e avaliar as ocorrências que podem vir a acontecer após a implantação, será criado um laboratório de testes, no qual será montada toda a estrutura necessária para distribuir os dados de um SGBD centralizado. Neste momento serão iniciadas as análises e coleta de informações para identificar, na prática, quais são reais

dificuldades encontradas na implantação de um SGBDD. Após o processo de implantação serão simulados alguns processos inerentes a um SGBDD, tais como a fragmentação, replicação e alocação de dados.

1.2.3. Viabilidade

A arquitetura de um BDD, assim como todo investimento que é feito em tecnologia, possui um custo elevado, pois se trata de uma estrutura mais complexa a ser administrada. Mas dependendo da área de atuação e do volume de negociações realizadas diariamente por uma empresa, pode ser um investimento justificável se comparado aos prejuízos causados pela inoperabilidade do negócio, caso ocorra alguma falha numa arquitetura com banco de dados centralizado.

1.3 OBJETIVOS

1.3.1. Objetivo geral

Definir uma solução que garanta alta disponibilidade para um SBDD, sempre se preocupando com os aspectos relacionados à segurança, independente dos custos correspondentes.

1.3.2. Objetivo específico

Aplicar, de forma prática, o que está descrito nas documentações literárias que tratam dos assuntos correlatos ao projeto, que sirvam de embasamento para a definição da melhor solução para a distribuição dos bancos de dados e suas respectivas rotinas de replicação.

Avaliar algumas das formas e ferramentas de replicação disponíveis no mercado, focando nos prós e contras de cada uma.

Propor uma solução eficaz para replicação de dados entre servidores de banco de dados localizados em diferentes sites.

Montar um laboratório para o estudo de caso que simulará a replicação entre bancos de dados Oracle.

Analisar a qualidade dos dados replicados entre os sites criados em laboratório.

Demonstrar que é perfeitamente possível, com algum esforço adicional, implementar uma estrutura de Banco de Dados distribuído, o que adicionará segurança à estrutura de dados da organização.

1.4 RESULTADOS ESPERADOS

Após a criação da estrutura de BDD, é esperado o aumento do desempenho dos sistemas, a alta disponibilidade das bases de dados, a confiabilidade dos dados e informações geradas e consequentemente o aumento das negociações da empresa.

1.5 ESTRUTURA DO TRABALHO

O trabalho está estruturado em 10 (dez) capítulos. O primeiro trata da introdução, a contextualização do estudo, o que motivou o desenvolvimento desta pesquisa, os objetivos gerais e específicos, os resultados esperados e a estrutura do trabalho.

O capítulo dois trata de conceitos, sempre embasado por autores conhecidos na área de Banco de Dados, além da verificação das vantagens, desvantagem e problemáticas de tal implementação e alguns dos replicadores existentes no mercado.

No capítulo três é apresentada a metodologia de planejamento, desenvolvimento e implementação de um Banco de Dados Distribuído.

O capítulo quatro trata do estudo de caso e da descrição das soluções técnicas abordadas para a construção do BDD e seu respectivo processo de replicação de dados.

O capítulo cinco consiste na análise dos resultados obtidos a partir do estudo em questão, onde são feitas as considerações gerais e demonstrados os resultados obtidos.

No capítulo seis é feita a conclusão do trabalho com a análise dos aspectos positivos e negativos identificados e avaliados na implementação da solução para o estudo de caso em questão.

No capitulo sete é feita uma sugestão para trabalhos futuros que englobam assuntos correlatos ao desenvolvido neste trabalho.

Ao final as referências bibliográficas pesquisadas, o glossário, apêndices e anexos.

2. REVISÃO DA LITERATURA

2.1. SISTEMA DE BANCO DE DADOS

"Um sistema de banco de dados (SBD) é basicamente um sistema de manutenção de registros por computador" (DATE, 1991), que possui a missão de armazenar os registros inseridos pelos usuários e recuperá-los quando solicitado.

"Por estrutura de banco de dados entendemos os tipos de dados, relacionamentos e restrições que devem suportar os dados" (ELMASRI/NAVATHE, 2005).

"Um sistema de gerenciamento de bancos de dados (SGBD) consiste em uma coleção de dados inter-relacionados e em um conjunto de programas para acessá-los" (KORTH, 1995).

Para que haja uma fácil compreensão dos assuntos abordados, nos tópicos a seguir detalharemos os tipos de banco de dados.

2.2. TIPOS DE BANCOS DE DADOS QUANTO A ARQUITETURA

2.2.1. Banco de Dados Centralizado

É uma estrutura na qual todos os usuários do sistema acessam um mesmo servidor que está fisicamente localizado em um local único.

A conexão a este servidor ocorre através de uma rede de computadores, a qual pode utilizar de diversos meios para que ocorra a comunicação, tais como cabo UTP, coaxial, rádio, fibra-óptica e etc.

Este modelo estrutural atualmente não é recomendável por não ser tão seguro, pois dependendo da capacidade de processamento e armazenamento do servidor, quantidade de dados trafegados em rede, quantidade de usuários conectados e complexidade das operações, os resultados para o usuário final podem se tornar muito demorados ou até apresentar erros em se tratando de soluções WEB, além do risco de parada total da empresa em caso de inoperabilidade do servidor de banco de dados centralizado.

Estes são alguns dos motivos pelos quais as empresas atualmente, com o avanço da tecnologia, estão migrando para a arquitetura de banco de dados distribuído, que é mais complexa de se implantar e manter, porém mais rápida e segura se comparada à centralizada.

A figura 1 demonstra um exemplo de arquitetura centralizada, na qual os servidores trabalham de forma independente dentro da fronteira da rede de computadores, sem haver o compartilhamento de disco, *Central Processing Unit* (CPU) e memória. Cada usuário processa sua informação e armazena em um único *site* central.

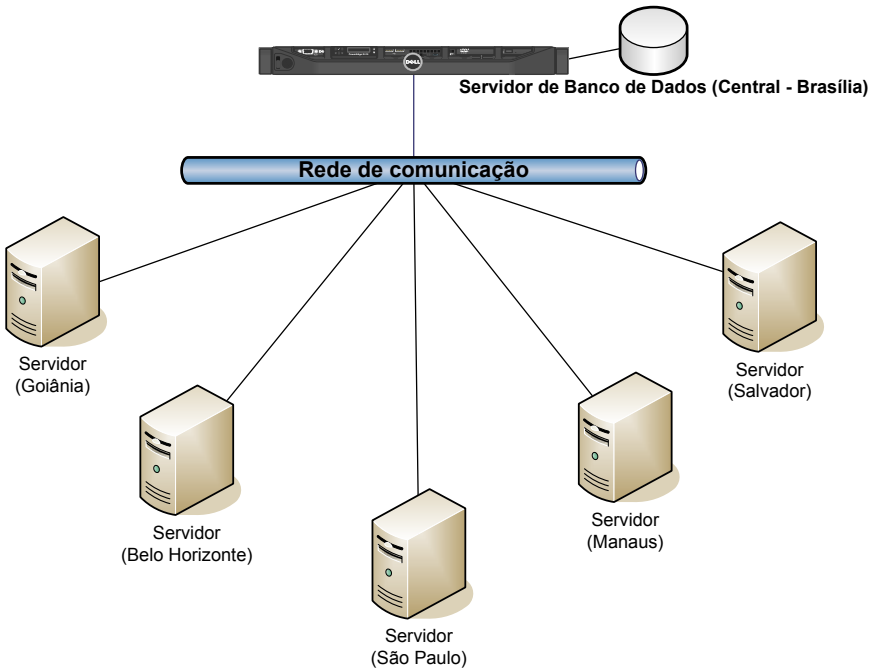


Figura 1 - Arquitetura Centralizada

Este modelo possui a desvantagem de que todas as informações precisam trafegar pela rede e chegar ao site onde está o banco de dados central, o que pode sobrecarregar o canal de comunicação deixando o serviço lento. Além do mais, se houver algum problema no site central, os demais sites também ficarão sem acesso aos dados.

2.2.2. Banco de Dados Distribuído

É “um conjunto de vários bancos de dados logicamente inter-relacionados, fisicamente separados em diferentes localidades, dispersos geograficamente e distribuídos por uma rede de computadores” (ÖZSU & VALDURIEZ, 2001).

De acordo com ELMASRI e NAVATHE, 2005, “Podemos definir banco de dados distribuído (BDD) como uma coleção de múltiplos bancos de dados logicamente inter-relacionados distribuídos por uma rede de computadores”.

Este modelo traz a vantagem da computação (processamento) distribuída, onde vários computadores interconectados dividem um problema grande e intratável em partes menores e o resolve eficientemente de maneira coordenada.

A figura 2 apresenta a ideia de uma estrutura de BDD, onde cada site possui seu respectivo SGBD, todos trabalhando de forma local e fazendo troca de informações quando necessário através de uma rede de comunicação. Neste modelo de arquitetura todos SGBDs podem ser iguais em todos os sites (arquitetura de BDD homogênea) ou diferentes em um ou mais sites (arquitetura de BDD heterogênea).

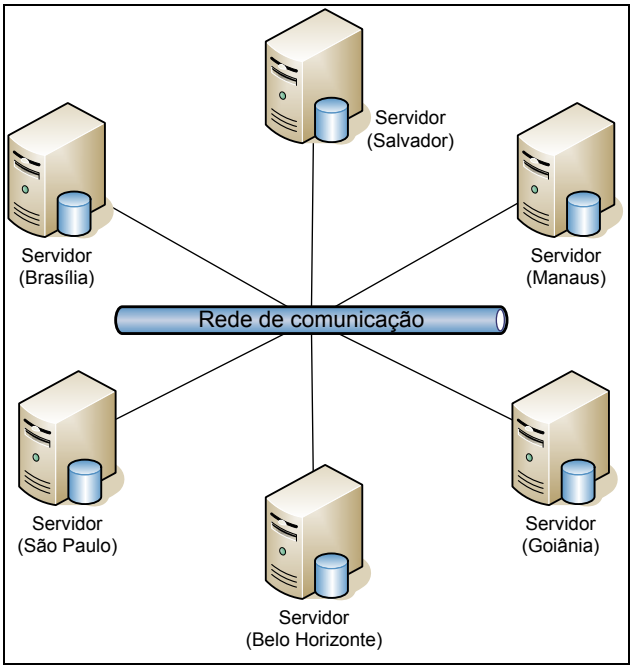


Figura 2 - Arquitetura Verdadeiramente Distribuída

Uma das principais vantagens deste modelo é que se houver algum incidente que ocasione a falha de algum dos sites, os outros poderão continuar operando normalmente. A falha ocorrida neste caso seria transparente para os demais.

2.3. VANTAGENS DE UM BANCO DE DADOS DISTRIBUÍDO

De acordo com KORTH e ÖZSU há diversas vantagens em se implantar um sistema de gerenciamento de bancos de dados distribuídos. Abaixo são destacadas algumas delas:

Gerenciamento de dados distribuídos - com níveis diferentes de transparência. Na forma ideal um SGBD deve ser transparente na distribuição, no sentido de camuflar os detalhes de onde cada arquivo se encontra armazenado

fisicamente, ou seja, mesmo que existam vários sites onde os dados são armazenados (distribuídos), isso deve ser transparente para o usuário final. A vantagem de um SGBD totalmente transparente é o alto nível de suporte que ele oferece para o desenvolvimento de aplicativos complexos.

Transparência de distribuição ou de rede - significa que o usuário não precisa conhecer detalhes técnicos ou operacionais relacionados à camada de rede. Este item subdivide-se em:

- *Transparência de localização* – significa que um comando utilizado para realizar determinada tarefa pode ser executado sem nenhum problema pelo usuário, independente da localização dos dados e da localização do sistema onde o comando foi emitido.
- *Transparência de nomenclatura* – significa que uma vez especificado o nome, os objetos nomeados podem ser acessados de forma não ambígua sem a necessidade de especificação adicional.
- *Transparência de replicação* – significa que as cópias de dados podem ser armazenadas em múltiplos locais para obter melhor disponibilidade, desempenho e confiabilidade. O usuário final não precisa saber da existência de cópias.
- *Transparência de fragmentação* – significa que os dados podem estar replicados em diversos sites de forma horizontal ou vertical, mas o usuário não precisa tomar conhecimento deste detalhe técnico.

Melhoria na confiabilidade e na disponibilidade:

- *Confiabilidade* – definida de forma ampla como a probabilidade de que o sistema esteja sempre em operação.
- *Disponibilidade* – é a probabilidade de que o sistema esteja disponível em um intervalo de tempo.

  Nota: Quando os dados e o software do SGBD são distribuídos por vários sites, se um site falhar os outros continuam funcionando. Apenas os dados do site que falhou não podem ser acessados, mas como se trata de banco distribuído, o usuário pode acessar outras partes do banco de dados.

Melhoria de desempenho – um banco de dados grande e complexo é quebrado em vários bancos com tamanhos menores e menos complexos, os dados ficam mais próximos de onde eles são realmente necessários, diminuindo a disputa

por CPU, reduzindo o IO e consequentemente reduzindo o atraso de acesso aos dados.

Facilidade de expansão – em um ambiente distribuído, tanto a expansão quanto ao acréscimo de mais dados, aumento do tamanho dos bancos de dados ou acréscimo de mais processadores é muito mais fácil, justamente pelo fato já citado de que estará se trabalhando com bancos menores e menos complexos.

Economia com *link* – o custo com comunicação de dados pode diminuir, pois os dados ficarão mais próximos do usuário, significando uma economia significativa nas situações em que o volume de tráfego de dados é tarifado por exemplo.

Economia com *hardware* – A redução de custos com servidores se dá pelo fato de que poderão ser utilizados computadores de menor valor e porte, tendo em vista que o volume de processamento é menor para os dados locais.

Aumento de performance – Uma estrutura distribuída apresenta melhor performance que um sistema centralizado pelo fato de haver a distribuição da carga de trabalho, pois cada site processa apenas parte do banco de dados, reduzindo em muito o esforço da CPU (*Central Processing Unit*) e o volume de E/S (entrada e saída).

2.4. DESVANTAGENS DE UM BANCO DE DADOS DISTRIBUÍDO

De acordo com KORTH e ÖZSU neste modelo existe uma complexidade muito superior, pois além de existirem os problemas encontrados em um ambiente de bancos de dados centralizados, terá vários problemas não resolvidos (aspectos de modelagem, processamento, consultas, concorrência, sistema operacionais, entre outros), além do custo adicional provocado pela necessidade de se investir em cursos com alto valor para o treinamento de pessoal, realização de aquisição de novos aplicativos e investimento em segurança da rede do SBDD, a qual precisa permanecer sempre muito bem protegida.

- ✓ **Distribuição de controle** - mesmo sendo considerada uma vantagem a distribuição gera problemas de sincronização e coordenação. Portanto, este controle distribuído exige mais responsabilidade e cuidados já que pode se tornar facilmente uma obrigação.
- ✓ **Segurança** – é um fator primordial para que qualquer negócio continue funcionando. Manter a segurança das redes é uma tarefa séria e complexa e

os problemas de segurança em sistemas de bancos de dados distribuídos são mais complexos do que os problemas nos sistemas de bancos de dados centralizados.

Entre os sites de banco de dados distribuídos existe uma rede de comunicação que deve estar resguardada por regras bem rígidas de segurança, o que é uma tarefa séria, complexa e cara. Por este motivo os problemas de segurança em SBDD são mais complexos que em bancos de dados centralizados.

2.5. PROBLEMÁTICAS DE UM BANCO DE DADOS DISTRIBUÍDO

Segundo KORTH e ÖZSU, vários problemas técnicos precisam ser resolvidos para se chegar ao melhor do potencial dos SGBDDs, pois estes possuem uma complexidade muito maior, o que pode influenciar na estabilidade do SBDD. Destacam-se dentro da problemática os tópicos a seguir:

- ✓ **Projeto de bancos de dados distribuídos** – Este é um dos principais pontos a serem discutidos quando se surge a idéia de distribuir dados por diversos sites, pois existem duas alternativas: particionar e replicar. O particionamento implica em estudos matemáticos com o intuito de redução de custos com tráfego de dados pela rede, sincronização entre sites e armazenamento de dados. Já a replicação é uma forma de se manter cópias dos dados em duas ou mais localidades, de forma a garantir que mesmo na extinção de determinado site, os dados estão preservados em outra localidade.
- ✓ **Processamento distribuído de consultas** – Em um SBDD a eficiência de atualização de dados é reduzida, mas em contrapartida, as consultas ficam muito mais rápidas, pelo fato dos dados consultados estarem próximos do usuário. O objetivo é melhorar os fragmentos, utilizando paralelismo para otimizar o desempenho nas consultas e atualizações.
- ✓ **Gerenciamento de diretório distribuído** – No dicionário de dados estão contidas informações como estrutura e localização sobre os itens de informações no banco de dados. O dicionário de dados pode permanecer centralizado em um único local ou distribuído por vários sites, onde pode haver uma cópia ou várias cópias do banco de dados e deverão estar em constante atualização.
- ✓ **Controle distribuído da concorrência** – Refere-se à sincronização de dados entre os bancos distribuídos, com a finalidade de manter a integridade dos

dados. O controle da concorrência em um contexto distribuído é um pouco diferente do que ocorre com a estrutura centralizada. Além de se preocupar com a integridade das informações de um único banco de dados, é importante, também, para a consistência de várias cópias do banco de dados.

- ✓ **Gerenciamento distribuído de impasses** – Se o mecanismo de sincronização se basear em bloqueios, a competição entre usuários pelo acesso a um conjunto de dados pode resultar em impasse. As alternativas bem conhecidas de prevenção, anulação e detecção também se aplicam a SBDDs.
- ✓ **Confiabilidade de SGBDs distribuídos** – Nos casos dos SBDDs, ocorrendo um defeito em que um ou vários sites fiquem inoperantes e sem acesso, os bancos de dados existentes nos sites que permanecerem operacionais devem continuar estáveis e atualizados.
- ✓ **Suporte do sistema operacional** – O suporte oferecido pelos sistemas operacionais para operações de bancos de dados não corresponde propriamente aos requisitos do software de gerenciamento dos bancos de dados. Os principais problemas são os sistemas de arquivos, o gerenciamento de memória, os métodos de acesso, a recuperação de falhas e o gerenciamento de processos.

2.6. ARQUITETURA DO BANCO DE DADOS ORACLE

De acordo com Watson (2010), a arquitetura do Oracle é composta por uma ou mais Instâncias (*Instance*) e Banco de Dados (*Database*), conforme figura a seguir:

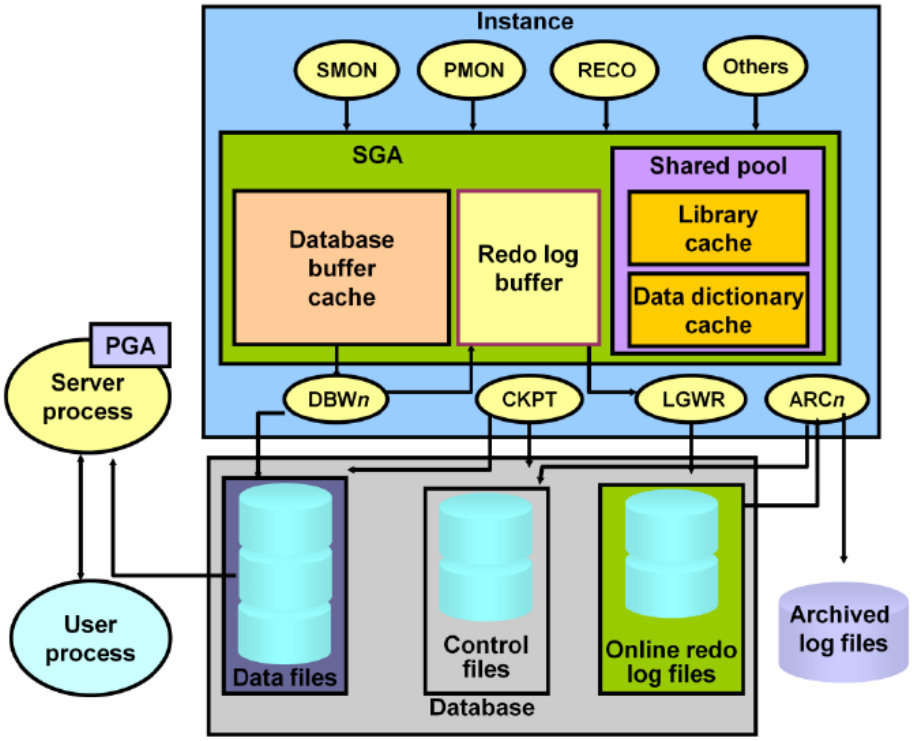


Figura 3 - Arquitetura do Oracle

Databases – são os armazenamentos físicos do banco de dados no sistema operacional, compostos por *data files*, *control files*, *online redo log file*, *archive log*, etc.

Instâncias – são vários processos rodando em background e na memória.

Dentro da instância Oracle existe o SGA (*System Global Area*), que é composto por *Database buffer cache*, *Redo Log buffer*, *Java pool*, *Streams Pool*, *Large Pool* e *Shared Pool*. Esta área responsável por compartilhar todos os processos do servidor e de *background*, para compartilhar dados e informações de controle da instância.

2.6.1. Funcionamento da *Instance*

Sempre que iniciada a instância do Oracle, uma área de *System Global Area* (SGA) é alocada, são iniciados os processos em *background* e a *database* é montada.

Após iniciada a *database* ocorre a conexão do usuário (*connection*) à instância Oracle, onde é iniciada a sessão (*session*). Do lado do servidor a conexão é feita com o processo de servidor (*Server process*) e do lado do usuário com o processo de usuário (*User process*).

2.6.2. Detalhamento da SGA

Conforme citado anteriormente a SGA é composta por diversos itens, os quais possuem o seguinte detalhamento:

Database Buffer Cache – nesta área ficam armazenados temporariamente os blocos de dados que foram lidos dos *data files* com a finalidade de que na próxima vez que os mesmos dados forem solicitados não haja a necessidade de fazer *I/O* em disco para a busca de dados, fazendo com isso que o banco se torne mais performático.

Redo Log Buffer – utilizado para registrar alterações realizadas no banco de dados. Possui a função de possibilitar o refazimento de uma determinada ação que possa ter sofrido algum tipo de problema. Os dados armazenados nessa área, após serem persistidos em banco de dados são transferidos para disco no *Online Redo Log Files* através do processo *LogWrite* (LGWR).

Shared Pool – é a área utilizada para armazenar o dicionário de dados, estrutura de controle, instruções SQL e PL/SQL executados recentemente, *cache* de resultado de função e *buffer* de mensagens de execução paralela.

2.6.3. Detalhamento da *Database*

Conforme demonstrado na Figura 1, assim como a SGA, a *database* também é composta por diversos outros itens conforme detalhamento a seguir:

Data files (Arquivos de dados) – é o conjunto de um ou mais arquivos onde são armazenados os dados do Banco de Dados.

Control Files (Arquivo de controle) – é um arquivo binário utilizado para rastrear o status do banco de dados e a estrutura física.

Online Redo Log Files – são arquivos de segurança que permitem que uma instância de Banco de Dados se recupere, caso haja alguma falha e não haja a perda de nenhum arquivo de dados.

2.6.4. Processos em *background*

Database Write Process (DBWn) - é responsável por escrever o conteúdo do *Database Buffer Cache* para o *Data File*.

LogWriter Process (LGWR) – é responsável por gerenciar o *Redo Log Buffer*, armazenando as entradas lá contidas no *Online Redo Log Files* quando executado o *commit* (confirmação) dos dados.

Checkpoint Process (CKPT) – é responsável por marcar todos os *datafiles* e *controlfiles*, utilizando *timestamps*, para indicar um ponto de recuperação em caso de falha.

System Monitor Process (SMON) – é responsável pela recuperação, se necessária, na inicialização da instância e pela limpeza dos segmentos temporários que não estejam em uso.

Process Monitor Process (PMON) – é responsável por fazer a recuperação quando um processo de usuário falhar, limpando *cache* no *buffer* do banco de dados e liberando recursos que o processo de usuário estava utilizando.

Archiver Process (ARCn) – é responsável por copiar os dados do *Online Redo Log Files* para o *Archived Log Files*. Este processo só funcionará se o Banco de Dados estiver configurado para o modo *Archivelog* e o arquivamento estiver automatizado.

2.7. *STRUCTURED QUERY LANGUAGE* (SQL)

De acordo com Machado e Abrei, “a sigla SQL significa Structured Query Language – Linguagem estruturada de pesquisa”.

A SQL é a linguagem padrão utilizada em bancos de dados relacionais, a qual pode ser utilizada nos modos interativo e embutido.

A SQL é chamada de interativa quando há a digitação de comandos e o resultado é visto logo em seguida. Já a SQL embutida recebe este nome por ficar dentro do código fonte de uma linguagem de programação.

Há a divisão da SQL nos grupos Linguagem de Definição de Dados (*Data Definition Language* - DDL), Linguagem de Controle de Dados (*Data Control Language* - DCL) e Linguagem de Manipulação de Dados (*Data Manipulation Language* - DML).

Linguagem de Definição de Dados – tem o objetivo de definir, alterar e eliminar objetos de banco de dados, tais como *tablespaces*, tabelas, *views*, *constraints* e outros.

Linguagem de Controle de Dados – tem o objetivo de controlar os usuários do banco de dados bem como o acesso aos seus recursos, mantendo a privacidade de informações importantes, a segurança de tabelas e o estabelecimento de fronteiras. Os comandos *Grant* e *Revoke* são exemplos de comandos DCL.

Linguagem de Manipulação de Dados – tem o objetivo de realizar a manipulação de dados armazenados em tabelas. Os comandos *Select*, *Update*, *Insert* e *Delete* são exemplos de comandos DML.

2.8. REPLICAÇÃO DE DADOS ENTRE BANCOS DE DADOS DISTRIBUÍDOS

Replicação é o nome dado ao processo sincronização de dados (inseridos, alterados ou excluídos) entre dois ou mais servidores de banco de dados com localizações lógicas ou geográficas diferentes, com a finalidade de que os dados estejam sempre prontos para uso em mais de um servidor. Com isso, a empresa ganha desempenho (os dados ficam mais próximos de quem os utiliza) e segurança (alta disponibilidade). O perfeito funcionamento da sincronização de dados depende de que a estrutura de dados dos servidores seja semelhante e, só é iniciada quando o dado se torna consistente na base de dados.

Elmasry e Navate (2005) descrevem que “A replicação é útil na melhoria da disponibilidade dos dados. O caso mais extremo é a replicação do banco de dados

inteiro em todos os sites do sistema distribuído, criando assim, um banco de dados distribuído completamente replicado. Isso pode melhorar notavelmente a disponibilidade porque o sistema pode continuar operando desde que pelo menos um site esteja em operação".

De acordo com Silberschatz, Korth e Sudarshan (2006), existem diversas vantagens e desvantagens na replicação de dados.

- **Disponibilidade**. Se um dos sites contendo os dados operacionais falhar, então estes mesmos dados podem ser encontrados em outro site. Desta forma o sistema pode continuar o processamento da consulta, apesar da falha de um site.
- **Paralelismo aumentado**. Como todos os sites terão os mesmos dados, as consultas realizadas geralmente ocorrerão nos servidores da rede local de cada cliente, com isso será minimizada a movimentação de dados entre os sites, melhorando a performance da rede que deixará de ficar sobrecarregada.
- **Maior sobrecarga da atualização**. Para garantir que todas as réplicas sejam consistentes é exigido mais controle do SGBD. Assim sempre que há uma atualização é necessária a atualização do todos os demais sites, com isso há uma maior sobrecarga para o sistema.

2.8.1. Modelo de Replicação

Segundo Silberschatz, Korth e Sudarshan (2006), o processo de replicação é dividido basicamente em dois tipos (síncrona e assíncrona).

Replicação Síncrona (*Eager*). É a replicação instantânea que ocorre no momento do processamento da transação, a qual possui a vantagem de permitir que os dados sejam atualizados em todas as réplicas de forma online, mas que em contrapartida possui a desvantagem de comprometer a performance e não permitir nenhum tipo de falha na rede. Neste tipo de replicação, quando há algum tipo de falha a operação não é concluída e o SGBD precisa fazer rollback na transação para que o sistema não fique travado aguardando que o falha seja normalizada. Este tipo de replicação é mais apropriado para aplicações comerciais (Ex. banco), onde é inevitável que todos os servidores sejam atualizados ao mesmo tempo.

Para o uso da replicação síncrona é indispensável o uso de um meio de transmissão de alta velocidade que garantam a eficiência e eficácia do processo, o que gera custos muito elevados, o que não ocorre com o tipo assíncrono.

Replicação Assíncrona (*Lazy*). A replicação não ocorre de forma instantânea. Neste modelo o replicador é responsável por montar um histórico das ações que ocorrem no banco de dados e que devem ser replicadas entre as bases de dados e, em um segundo momento ele se baseia neste histórico para gravar os dados nas réplicas. Com isso este modelo agrega várias vantagens: - O intervalo de replicação pode ser programado no replicador, a transação ocorrerá mesmo que algum dos sites esteja inoperante e dependendo do volume de dados, não precisa de uma grande quantidade banda de rede para sincronizar os dados, pois estes podem ser replicados utilizando até mesmo uma linha discada, com acesso de hora em hora do diário.

2.8.2. Métodos de Replicação

De acordo com Oliveira (2006), em sistemas de bancos de dados, toda replicação requer os mecanismos de Log de Transações e Triggers para para realizar as alterações feitas em dados replicados.

Log de transação – copia as transações marcadas para replicação para uma área de preparação para transmissão. Esta técnica possui a vantagem de ter baixo impacto no servidor porque exige o mínimo do uso de processador quando está lendo o log em memória e escrevendo em disco.

Triggers – propagam em uma tabela de log, as mudanças quando elas ocorrem, para transmissão posterior.

Como a linguagem procedural da base de dados pode ser usada para estes métodos, ela provê maior flexibilidade em decidir quais os dados serão replicados.

Logs de transação e Triggers podem ser usados individualmente ou conjuntamente em um SGBD, dependendo das características de cada sistema e de suas aplicações.

A vantagem de se usar Triggers é que estas demandam menor tempo e consomem menos recursos que os Logs de transação para realizar o processo de replicação.

2.8.3. Conflitos de Replicação

A ocorrência de conflito de replicação acontece quando o mesmo dado é atualizado em duas bases ao mesmo tempo.

Para assegurar a convergência, os conflitos de atualização devem ser detectados e resolvidos para que o dado tenha o mesmo valor em cada base. Quanto maior for a frequência de propagação das mudanças de dados, menor será a incidência de conflitos de atualização.

Os conflitos de atualização podem ser evitados pela limitação do direito de atualização dado a um elemento para uma base. Muitos acreditam, equivocadamente, que a resolução de conflitos é um processo que cabe ao desenvolvedor de software. O software jamais precisará resolver conflitos, pois eles nunca irão ocorrer se existir uma política de planejamentos e procedimentos para eliminar essa possibilidade no próprio banco de dados.

De acordo com Coutinho (2002) sempre deve ser criado um ambiente de replicação que evite a possibilidade de conflitos. Para que isso seja possível é necessário a utilização de algumas técnicas. Abaixo são listados alguns dos tipos de conflitos que podem ocorrer, conforme a replicação utilizada:

- ✓ **Conflito de Update** – poder acontecer quando duas ou mais transações, originadas de bancos de dados diferentes atualizam um mesmo registro ao mesmo instante. Este tipo de conflito pode ser reduzido ou até evitado reduzindo o intervalo de tempo entre uma replicação e outra.
- ✓ **Conflito de Chaves únicas e seqüenciais** – pode acontecer quando duas ou mais transações tentar fazer a inserção de um registro com mesma chave (*Primary Key*) ou (*Unique Key*) em uma mesma tabela, causando a violação de integridade.
- ✓ **Conflito de *Delete*** – pode acontecer quando duas ou mais transações tentam, ao mesmo instante, deletar um registro com a mesma chave de identificação. Neste caso só a primeira transação obterá sucesso, pois no momento em que a segunda for executada não encontrará mais o dado a ser excluído.

2.9. SOFTWARES DE REPLICAÇÃO

2.9.1. Dbmoto

É um software replicador que oferece suporte para atualização em tempo real, suporte a qualquer tamanho de conjunto de dados, suporte a múltiplas bases de dados e suporte a qualquer plataforma.

Este software possui a seguintes características:

- Identificação dos dados modificados para minimizar o acesso aos bancos de dados de origem e destino;
- A capacidade de ler os *logs* de transação do banco de dados;
- Suporte para transformação de dados; e,
- Apoio à integração de dados com soluções de *Data Warehousing*.

2.9.2. Oracle Goldengate

É um replicador adquirido pela Oracle em 2009, que promove a integração rápida de dados em tempo real e a transformação de informações em larga escala, com alta confiabilidade, bem como a solução de *Business Intelligence* (BI) instantânea, melhoria no desempenho de OLTP e disponibilidade contínua (24x7) dos dados para sistemas de missão crítica.

O Oracle Goldengate permite a movimentação de dados em tempo real entre sistemas heterogêneos de origem e destino, com sobrecarga mínima na infraestrutura de TI, habilitando soluções de BI em tempo real, sincronização de dados multidirecional para sistemas distribuídos, atualizações sem paralisações no sistema, recuperação de desastres e migrações entre diferentes bancos de dados, plataformas de sistema operacional e servidores.

2.9.3. DBLink

É um link criado em um banco de dados que possibilita acesso a outro banco de dados, sendo ele Oracle ou não. Esta conexão pode ser homogênea (mesmo banco de dados) ou heterogênea (banco de dados diferente), sendo que no Oracle para o acesso heterogêneo é necessário o uso do Oracle Heterogeneous Services.

Dependendo do nível de privilégio do usuário do DBLink é possível fazer a replicação de dados entre servidores, no entanto este aplicativo não pode ser

comparado a um replicador, pois é bem limitado com relação desempenho, controle de transação, automatização e etc.

Usar esta ferramenta implica em assumir a árdua tarefa de fazer de forma manual toda a complexa atividade que é executada por um replicador. Portanto, devido a estas limitações, o DBLink é mais indicado somente para consultas entre bancos de dados distribuídos.

2.9.4. Objectmmrs

É uma suíte de softwares para projetos de replicação de banco de dados, da empresa Object Sistemas. Criado em 2002, o software evoluiu e tornou-se uma verdadeira ferramenta multiuso de replicação de dados em tempo real de forma bidirecional entre servidores remotos, mesmo que estes estejam usando SGBDs de diferentes fabricantes.

A ferramenta possui tecnologia nacional e, o melhor: é totalmente adaptado às piores condições possíveis de internet, possibilitando assim a troca eficiente de dados mesmo em instáveis e restritas bandas de rede de até inacreditáveis 32Kbps.

É um *software* capaz de trabalhar com bancos Oracle, DB2, SQL Server, Postgresql, Firebird, HSQLDB, SQLite e Apache Derby.

As principais características do Objectmmrs são:

- Backups em tempo real de banco de dados, replicando os dados de um servidor principal para um servidor secundário localizado na rede local ou em outro local físico.
- Eliminação de rotinas de atualização de dados em batch. Ao usar o produto os dados são replicados em tempo real, continuamente, sem necessidade de rotinas de exportação e importação de dados, etc. A sincronização “tardia” de dados é fonte de muitos problemas, ao manter bases distintas atualizadas em poucos segundos ao invés de, por exemplo, realizar atualizações diárias, eliminará muitas inconsistências do dia-a-dia.
- Replicação entre bancos de dados de diferentes fabricantes, como por exemplo, a sincronização de uma base de dados Oracle com uma base de dados MySQL.
- Importação de dados a partir de arquivos texto, planilhas, etc.
- Replicação bidirecional e assíncrona.

- Resistência e auto-recuperação em casos de falhas de rede ou de banco de dados.
- Possibilidade de implementação em diversas topologias de rede, tais como estrela, hierárquica, ponto-a-ponto e etc.
- Arquitetura multi-thread e multi-engine, imprescindível em topologias do tipo estrela onde há um servidor central e N servidores nas pontas, de modo a permitir a replicação para um grande número de *slaves*.
- Baixo *overhead* nos servidores de banco de dados, o software de replicação pode ser instalado no próprio servidor de banco de dados, sem com isso comprometer o desempenho do servidor. Pode também, opcionalmente, em casos críticos, ser instalado em um ou mais servidores separados.
- Replicação a nível de tabela e de operação (*insert, update, delete*).
- Na replicação de *update* atualiza apenas as colunas alteradas, economizando assim banda de rede e minimiza conflitos de concorrência.
- Tratamento automático de conflitos de *update*, por tempo real ou prioridade de base de dados.
- Customizável, pois o cliente pode desenvolver classes java usadas para replicar casos especiais, como por exemplo, replicação para um middleware usando webservices, replicação de arquivos como fotos e vídeos via tunel ssh/sftp, etc.
- Possuiu *log* detalhado e configurável das operações DML nos níveis: DEBUG, INFO, WARN, ERROR e FATAL.
- É uma ferramenta multiplataforma, podendo ser instalado com facilidade em Windows, Linux, Unix, Mac e etc, desde que a plataforma permita o uso de Java 5 ou superior.
- Pode ser usado como ferramenta de sincronização de bases de dados em notebooks, PDAs e etc.
- Pode ser configurado para o envio de email automático em caso de falhas de rede prolongadas, etc. A saída do *log* pode ser integrada com ferramentas de monitoramento de *syslogs*.
- Não usa recursos de baixo nível dos SGBDs, é extremamente portável entre diferentes versões de um mesmo SGBD, facilitando assim upgrades

de versão do SGBD.

- É preparado para a replicação de grandes volumes de dados, como por exemplo, a partir de 500 mil operações chegando com bom desempenho a milhões de operações por dia.
- É uma tecnologia nacional, com possibilidade de customizações com desenvolvimento e suporte técnico local.

Quanto à replicação de DDL, o software possui uma interface web de aministração da ferramenta, que oferece a opção de disparar a execução de comandos SQL, sejam eles DDL ou DML entre todos os sites cadastrados no modelo. Após a execução é possível avaliar o log de execução nos servidores utilizados para a execução dos comandos.

2.9.5. DBReplicator

É uma API de replicação que possui como características mais significativas:

- Replicação heterogênea, ou seja, sincroniza dados entre o mesmo tipo de banco de dados ou entre outros bancos de dados;
- Replicação bi-direcional;
- A plataforma é 100% Java;
- Independência da aplicação, ou seja, o software não precisa conhecer da replicação, nem acessar a fonte de dados de forma diferente, o que é importante para aplicativos que não tem acesso ao código fonte;
- Detecção automática de conflitos de resolução;
- Facilidade de agendamento, onde o cronograma pode ser especificado por minutos, horas, etc, e executa a operação desejada sempre no intervalo especificado, a menos que haja algum erro.
- Depuração Verbose, que implementado usando apache log4j, pode ser bastante flexível em termos de produção.
- Tratamento de caracteres especiais. Atualmente, esses campos que contenham caracteres especiais devem ser especificados no EncodeConfig.ini, para o correto tratamento automático.
- Criação, caso não existam, as tabelas no assinante site slave.

2.9.6. Replication Server

É um software de replicação que trata os dados independentes do tipo de banco de dados, tanto na origem ou destino, enquanto mantém como prioridade o desempenho e estabilidade.

Esta ferramenta possui as seguintes características:

- Garante a recuperação em caso de desastres, sem interrupções para as aplicações críticas do negócio.
- Evita impacto no desempenho do banco de dados operacional e captura as mudanças em tempo real para em seguida replicá-las.
- Suporta replicação multi-master em ambiente heterogêneo de banco de dados, em diferentes localizações geográficas, garantido alto disponibilidade dos dados operacionais onde e quando precisar.
- Permite a migração de Sistema Operacional ou plataforma de banco de dados, sem a necessidade de interrupção das operações.
- Suporta replicação entre várias plataformas de banco de dados, incluindo Sybase ASE, Oracle, DB2 IBM e Microsoft SQL Server.

3. METODOLOGIA

3.1. METODOLOGIA UTILIZADA

O processo de pesquisa deste trabalho pode ser classificado sob os seguintes aspectos:

Quanto à natureza, ela é aplicada, uma vez que "o investigador é movido pela necessidade de contribuir para fins práticos mais ou menos imediatos, buscando soluções para problemas concretos. [CERVO e BERVIAN, 1996, p. 47].

Do ponto de vista dos objetivos, ela é exploratória, pois visa proporcionar maiores informações sobre determinado assunto; facilitar a delimitação de um tema de trabalho; definir os objetivos ou formular as hipóteses de uma pesquisa ou descobrir novo tipo de enfoque para o trabalho que se tem em mente. De acordo com Bonoma (1985), o método do Estudo de Caso tem sido visto mais como um recurso pedagógico ou como uma maneira para se gerar *'insights'* exploratórios, do que um método de pesquisa propriamente dito e isto tem ajudado a mantê-lo nesta condição.

Quanto aos procedimentos, ela é bibliográfica, pois, utiliza fontes secundárias, ou seja, livros e outros documentos bibliográficos. [ANDRADE, 2007, p. 115].

3.2. PLANEJAMENTO DE CRIAÇÃO DO BANCO DE DADOS DISTRIBUÍDO

Neste tópico, abordaremos o modelo utilizado para o experimento prático de criar, executar e manipular um BDD.

3.3. MODELO DE DADOS

3.3.1. Modelo entidade-relacionamento

O Modelo Entidade Relacionamento (MER) é uma extensão do modelo conceitual de dados, o qual foi introduzido por Peter Chen em 1976. (SETZER e SILVA, 2005)

“Segundo Chen, a visão de uma dada realidade baseia-se no relacionamento entre entidades, o qual retrata os fatos que governam essa mesma realidade, e cada um (entidade ou relacionamento) pode possuir atributos (qualificadores desta realidade)”. (MACHADO e ABREU, 2010)

Chen se dedicou a destacar a importância de reconhecer os objetos que compõem cada negócio independente de tratamento de informações,

procedimentos, programas, tecnologia, etc. Estes objetos ele classificou em dois grupos: Entidades e Relacionamentos, onde:

a. **Entidades** – é “aquele objeto que existe no mundo real com uma identificação distinta e com um significado próprio. São coisas que existem no negócio, ou ainda, descrevem o negócio em si”. (MACHADO e ABREU, 2010).
b. **Relacionamentos** – é o fato, o acontecimento que liga dois objetos, duas “coisas” existentes no mundo real (duas entidades). (MACHADO e ABREU, 2010)

Abaixo é apresentada a título de exemplo a representação gráfica de um MER simples, no qual são apresentados duas entidades (Departamento e Funcionário) e um relacionamento (Ter).

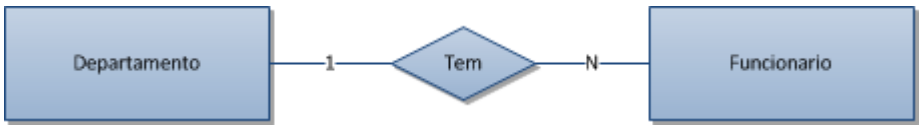


Figura 4 - Exemplo de MER Simples

3.4. ETAPAS PARA O DESENVOLVIMENTO E IMPLMENTAÇÃO

A construção de um SBDD exige bastante conhecimento do negócio, além de muitos estudos e análises na fase de projeto e exaustivos testes antes da total implementação, tudo isto para garantir que o sistema será estável e confiável e que a complexidade de sua implementação será transparente para o usuário final. Por este motivo serão respeitadas as seguintes etapas:

a. Preparação do modelo de dados – Nesta etapa do projeto será preparado o modelo de dados que retrate uma amostragem das entidades contidas no negócio. Serão mapeadas algumas das diversas entidades, seus respectivos atributos e relacionamentos, de forma a dar visibilidade parcial do modelo de dados.
b. Definição da política de sincronização e carga de dados – será definido, com base no modelo de dados preparado na etapa anterior.
c. Definição das entidades operacionais que serão sincronizadas entre os sites, de forma a manter a integridade, segurança e acessibilidade aos dados.

d. Implementação da ferramenta de replicação – Nesta etapa será definida a melhor forma de se fazer os processos de acesso aos dados encontrados no SBDD tanto para consulta quanto para atualização.

3.5. CONTEXTUALIZAÇÃO DO ESTUDO DE CASO

A PREVIDAS é uma empresa de previdência complementar, sem filiais e com a estrutura de banco de dados centralizada, esta empresa está sujeita ao cumprimento de normas regulamentares da Secretaria de Previdência Complementar, para que honre fielmente o seu propósito final que é o pagamento de aposentadoria para seus contribuintes.

Como esta empresa possui certificação ISO 9001/2008, sofre auditorias de qualidade constantemente e é alvo de observação de seus contribuintes, auditores independentes, auditores dos patrocinadores e de seus órgãos regulamentadores, existe uma forte preocupação de sua Diretoria com relação à Segurança da Informação e a Gestão de Continuidade do Negócio. A partir desta preocupação, surgiu a necessidade de se iniciar um projeto de replicação do banco de dados em mais de uma localidade, com a finalidade de garantir que se houver algum tipo de problema na sede, seja possível o acesso aos dados em locais alternativos para a continuidade das atividades normalmente.

A sede da empresa utiliza SGBD Oracle 10G, e os sites alternativos, também utilizarão Oracle 10G. A replicação será assíncrona e abrangerá qualquer tipo de alteração de dados ocorrida em algum dos sites, os quais farão atualização entre si.

3.6. PROCEDIMENTO DE PESQUISA

Todo o processo ocorre com embasamento no disposto na revisão literária, análise para preparação do ambiente de testes, preparação do ambiente, testes e auditoria dos resultados.

4. ESTUDO DE CASO

Nesta etapa será criado um laboratório com um Sistema de Gerenciamento de Banco de Dados Distribuído homogêneo (Oracle 10G) que simulará o ambiente da empresa de previdência complementar (sede) e dois sites alternativos de replicação, os quais terão os dados replicados entre si de forma automática, a partir da implementação de um software de replicação de Banco de Dados confiável que atenda a tal necessidade.

4.1. DIAGRAMA DA SOLUÇÃO

O projeto será composto de três sites alocados em diferentes pontos geográficos, os quais serão interligados através de links ponto a ponto de fibra-óptica com velocidade de tráfego *full* de 2 Mbps.

A distância aproximada dos dois sites alternativos para a sede é entre 1 e 3 quilômetros.

As alterações realizadas na sede serão replicadas para os dois sites alternativos. Já as alterações dos sites alternativos só serão replicadas para a sede e nunca entre eles, conforme demonstrado no diagrama a seguir. Todo esse processo de replicação ocorrerá de forma assíncrona.

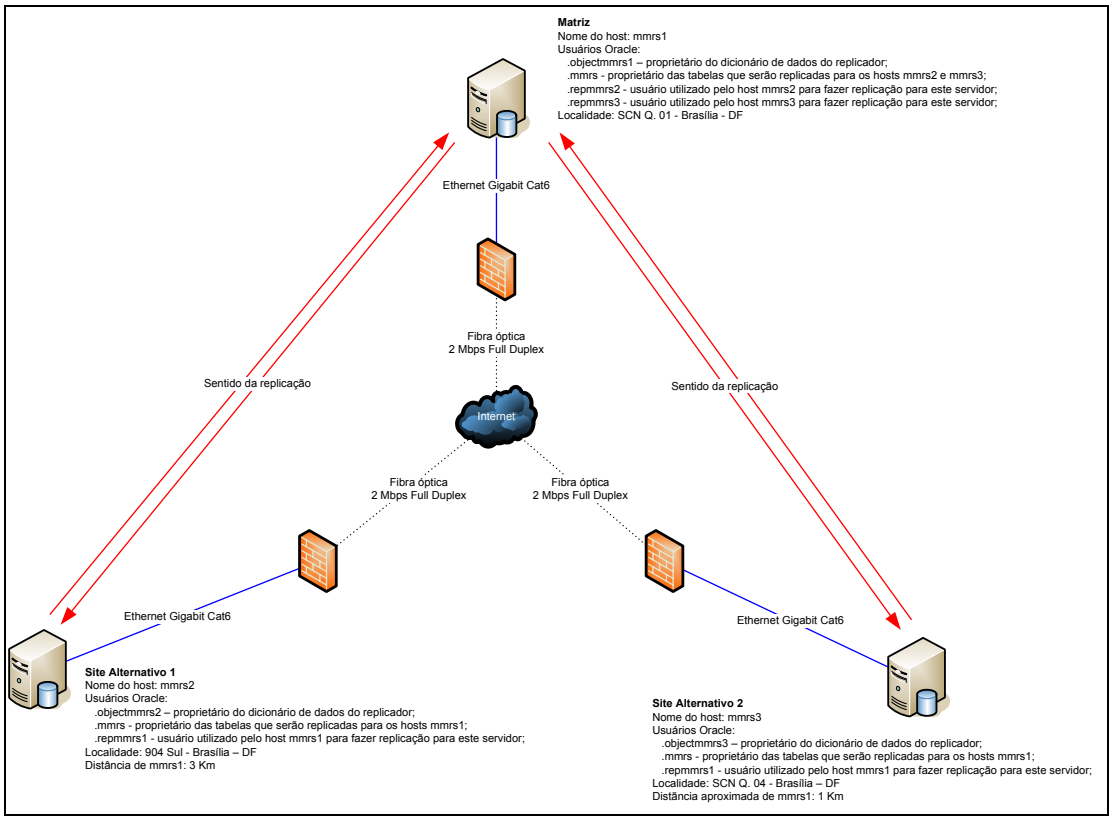


Figura 5 – Diagrama da solução de replicação

4.2. SGBD UTILIZADO

Para este projeto serão utilizados, tanto na sede quanto nos sites alternativos o Oracle 10G Express Edition Release 10.2.0.1.0.

4.3. INFRA-ESTRUTURA

Para simular o ambiente da PREVIDAS serão utilizados servidores virtuais, criados com o Virtualbox, os quais serão hospedados em um servidor com as seguintes características:

Processador CORE I5 3.0GHz;
Disco Rígido de 1TB - 7200RPM;
Memória RAM de 16 GB;
Linux Debian 6 (Kernel 2.6.39.4 compilado para 64 GB de memória RAM); e,
Virtualbox 4.0.12.

Os servidores virtuais terão configurações básicas, que servirão para simular o processo de replicação, sem comprometimento do desempenho, conforme as características a seguir:

Sede

Servidor de Banco de Dados

- Nome de Host: mmrs1
- IP: 192.168.1.171 / Máscara de Rede: 255.255.255.0
- Sistema Operacional Linux Debian 6;
- SGBD Oracle 10G Express; e
- Memória RAM de 2 GB.

Site Alternativo 1

Servidor de Banco de Dados

- Nome de Host: mmrs2
- IP: 192.168.1.172 / Máscara de Rede: 255.255.255.0
- Sistema Operacional Linux Debian 6;
- SGBD Oracle 10G Express; e
- Memória RAM de 2 GB.

Site Alternativo 2

Servidor de Banco de Dados

- Nome de Host: mmrs3
- IP: 192.168.1.173 / Máscara de Rede: 255.255.255.0
- Sistema Operacional Linux Debian 6;
- SGBD Oracle 10G Express; e
- Memória RAM de 2 GB.

4.4. DESCRIÇÃO DA SOLUÇÃO

A alternativa encontrada para proporcionar segurança aos dados operacionais da PREVIDAS e atender aos requisitos de Gestão de Continuidade do Negócio desta empresa, foi seguir as melhores práticas de mercado, adotando a tecnologia de banco de dados distribuídos e a replicação de banco de dados, o que trará grandes benefícios, pois os dados ficarão mais protegidos e, mesmo que aconteça algum tipo de problema ou até mesmo uma catástrofe em algum dos sites, sempre haverá mais duas cópias idênticas para a continuidade das atividades de forma transparente para os usuários.

Atualmente existem no mercado várias ferramentas de replicação de banco de dados, cada uma com suas características técnicas e preços que variam bastante de uma para a outra. Para a decisão da ferramenta que seria utilizada nesta solução, foi elaborada uma tabela comparativa, demonstrada logo a seguir, onde foram seguidos alguns critérios de avaliação e aprovação:

- Análise da ferramenta GoldenGate, a qual é nativa do SGBD Oracle;
- Análise de outras ferramentas replicadoras de Banco de Dados, paralelas ao GoldenGate, disponíveis no mercado; e,
- Avaliação de custo benefício de cada uma das ferramentas analisadas.

Item de avaliação	DBMoto	Oracle GoldenGate	DBLink	ObjectMMRS	DBReplicator
Compatível com Oracle	SIM	SIM	SIM	SIM	SIM (Mas não testado)
Replicação Multi-master	SIM	SIM	SIM	SIM	SIM
Replicação Assíncrona	SIM	SIM	NÃO	SIM	SIM
Detecção e controle de conflitos de update	SIM	SIM	NÃO	SIM	SIM
Uso de criptografia SSL	SIM	SIM	NÃO	SIM	NÃO
Captura de alteração dos dados baseados em triggers padrão do banco de dados	NÃO	SIM	NÃO	SIM	SIM
Ferramenta de sincronização de tabelas	SIM	SIM	NÃO	SIM	SIM
Replicação de DDL	NÃO	SIM	NÃO	SIM	NÃO
Proteção de usuários e senhas de banco de dados, não expondo os arquivos de configuração ou setup de tabelas	SIM	SIM	NÃO	SIM	NÃO
Ferramenta de monitoramento que pode enviar emails e sms alertando os mecanismos de replicação de banco de dados indisponíveis	NÃO	SIM	NÃO	SIM	NÃO
Arquitetura Multi-thread	SIM	SIM	NÃO	SIM	SIM
compatível com servidor Linux	NÃO	SIM	SIM	SIM	SIM
compatível com servidor Windows	SIM	SIM	SIM	SIM	SIM
Ferramenta de monitoramento enterprise com interface web habilitada para controle de vários projetos diferentes de replicação	SIM	SIM	NÃO	SIM	NÃO
Preço Anual	R$ 12.000,00	R$ 20.000,00	R$ 0,00 (Nativo)	R$ 1.000,00	R$ 0,00 (Open Source)

Tabela 1 - Matriz de Comparação de Replicadores

O item preço anual, demonstrado na sede acima é para servir de referência para este estudo, pois varia de projeto para projeto por depender de diversas variáveis, como por exemplo, a quantidade de tabelas a replicar, quantidade de conectores de bancos diferentes necessários, volume diário de dados, se precisará ou não de criptografia, se haverá redundância de links e etc.

Com relação à análise técnica, todas as ferramentas possuem grande potencial para atender à solução da PREVIDAS, com exceção do DBLink que seria uma solução lenta, trabalhosa e complicada de administrar. Após análise dos itens

de avaliação da tabela de comparação, foi identificado que as ferramentas mais completas para a solução são o Oracle Goldengate e o ObjectMMRS.

O ObjectMMRS da Object Sistemas foi escolhido como a solução ideal pelo seu baixo custo e pela robustez apresentada para o processo de replicação. Este será utilizado para replicar toda e qualquer operação de inserção, alteração e exclusão de dados que houver em algum dos sites do projeto.

Para o funcionamento do processo de replicação este software utiliza o controle do dicionário de dados próprio da ferramenta, criado especificamente para esta finalidade.

Todas as informações de inserção, alteração e exclusão são armazenadas em tabelas temporárias e excluídas após a conclusão, com sucesso, do processo de replicação dos dados lá armazenados.

O modelo de dados da PREVIDAS é constituído de aproximadamente 800 tabelas, as quais devem ser 100% replicadas entre os sites. Neste projeto está sendo feita uma amostragem de somente 4 tabelas (modelo abaixo), que servirão para demonstrar o uso da ferramenta de replicação.

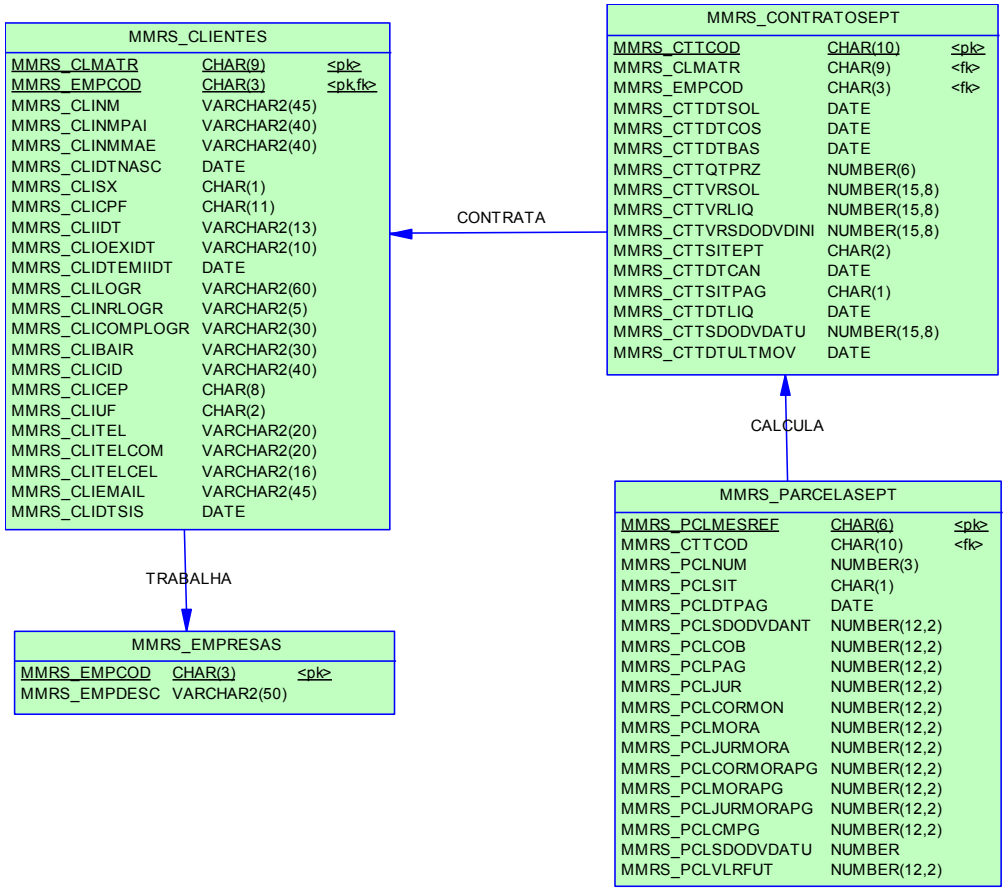


Figura 6 – Diagrama Físico do Banco de Dados

4.5. RESOLUÇÃO DE CONFLITOS NO PROCESSO DE REPLICAÇÃO

Conforme citado no item 2.9.4, a ferramenta ObjectMMRS, escolhida neste projeto para replicação, possui o tratamento automático de conflitos de *update*, por tempo real ou prioridade de base de dados.

Para que a ferramenta funcionasse adequadamente conforme o previsto pelo fabricante foi necessária a sincronização de horário com servidores NTP (Network Time Protocol), para garantir que a hora dos servidores de replicação e de banco de dados sempre estejam sincronizados de acordo com a hora atual do observatório nacional, evitando assim conflitos de update, insert e delete que comumente ocorrem em replicadores multi-master sem horários sincronizados.

4.6. CRIAÇÃO DO PROCESSO DE REPLICAÇÃO

Para instalação do replicador foi necessário solicitar à empresa Object Sistemas, uma versão Trial da ferramenta ObjectMMRS, e descompactá-la no diretório home do usuário de sistema operacional “object”, criado especificamente para este projeto. A estrutura de diretórios do replicador ficou “/home/object/objectmmrs”.

Todos os servidores que desempenharão o papel de master (replicador) precisarão da ferramenta ObjectMMRS instalada neles.

Em conformidade com o manual, no banco de dados será necessário criar no mínimo três usuários, sendo um proprietário do dicionário de dados do replicador (objectmmrs), um usuário que será o proprietário das tabelas a serem replicadas (mmrs) e um usuário que será utilizado por outro servidor master para replicar dados para este (repmmrs), quando este estiver no papel de slave, conforme detalhado na figura seguinte.

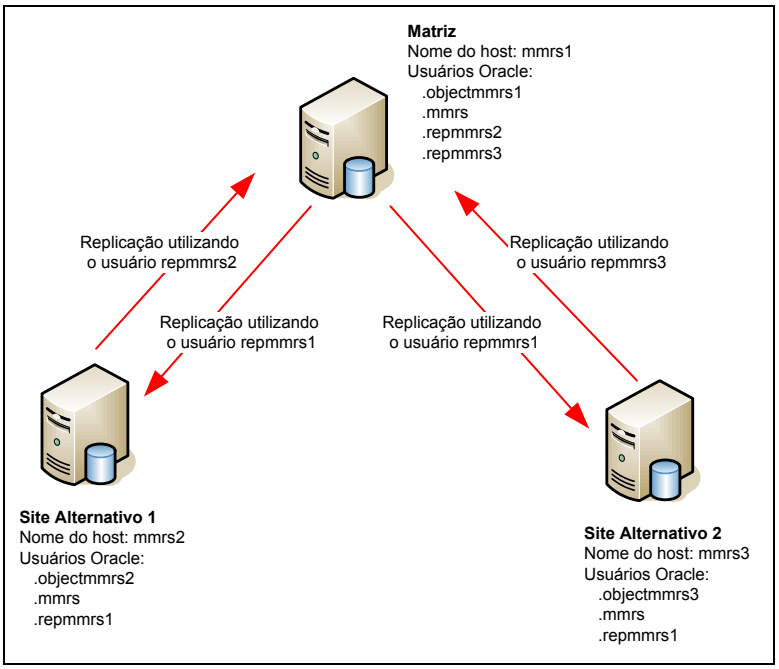


Figura 7 - Usuário de Banco de Dados

Conforme o diagrama acima, todos os bancos de dados deste projeto serão multimaster, ou seja, replicarão e receberão replicação. Por este motivo, em cada servidor precisará existir no mínimo os usuários citados no parágrafo acima. Logo adiante são demonstradas as telas com todo o processo de criação de um servidor de replicação.

Redimensionamento da tablespace System pelo fato do tamanho default do Oracle Express (350 MB) não suportar uma grande quantidade de operações realizadas neste projeto. O replicador trabalhará mais estável se for utilizada uma tablespace com bastante espaço, como o deste caso, que foi redefinida para 3GB.

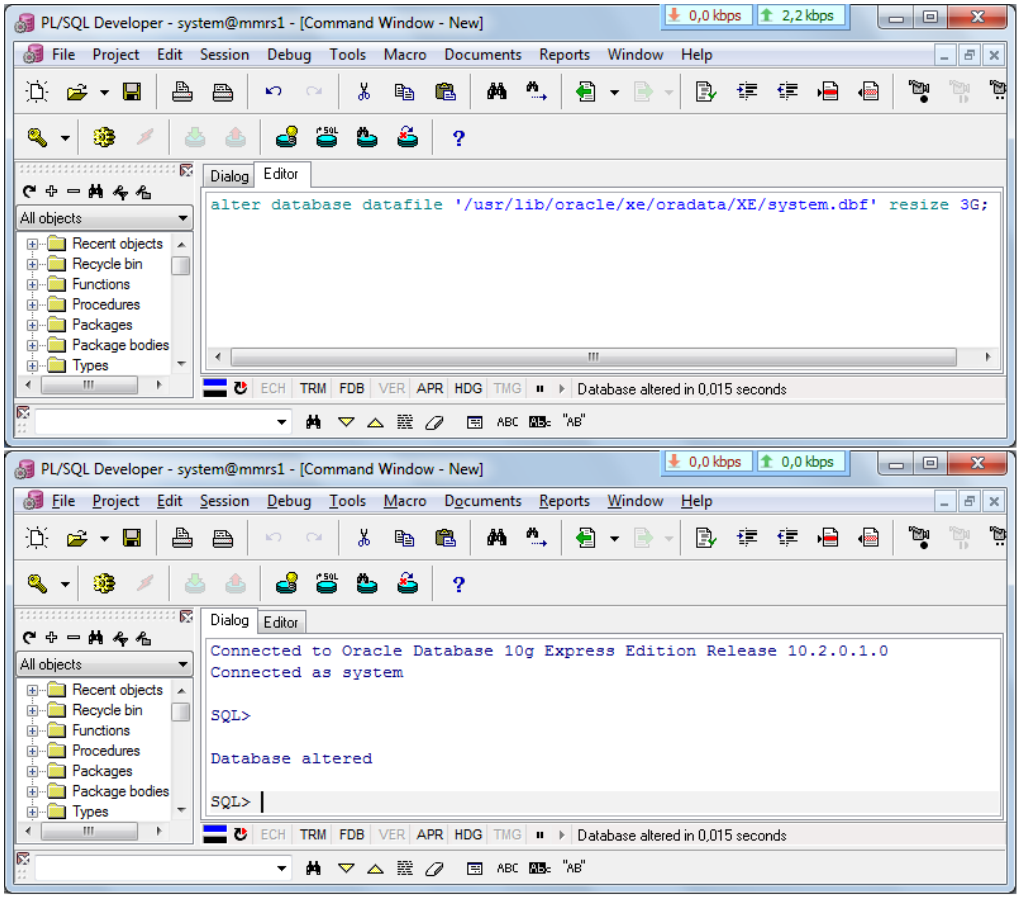


Figura 8 - Redefinição de tamanho da tablespace System

Criação do usuário (mmrs) proprietário das tabelas que serão replicadas aos sites alternativos, objectmmrs1 (proprietário do dicionário de dados do replicador), repmmrs2 e repmmrs2 (que serão utilizados pelos sites alternativos 1 e 2 para replicar dados para este servidor sede).

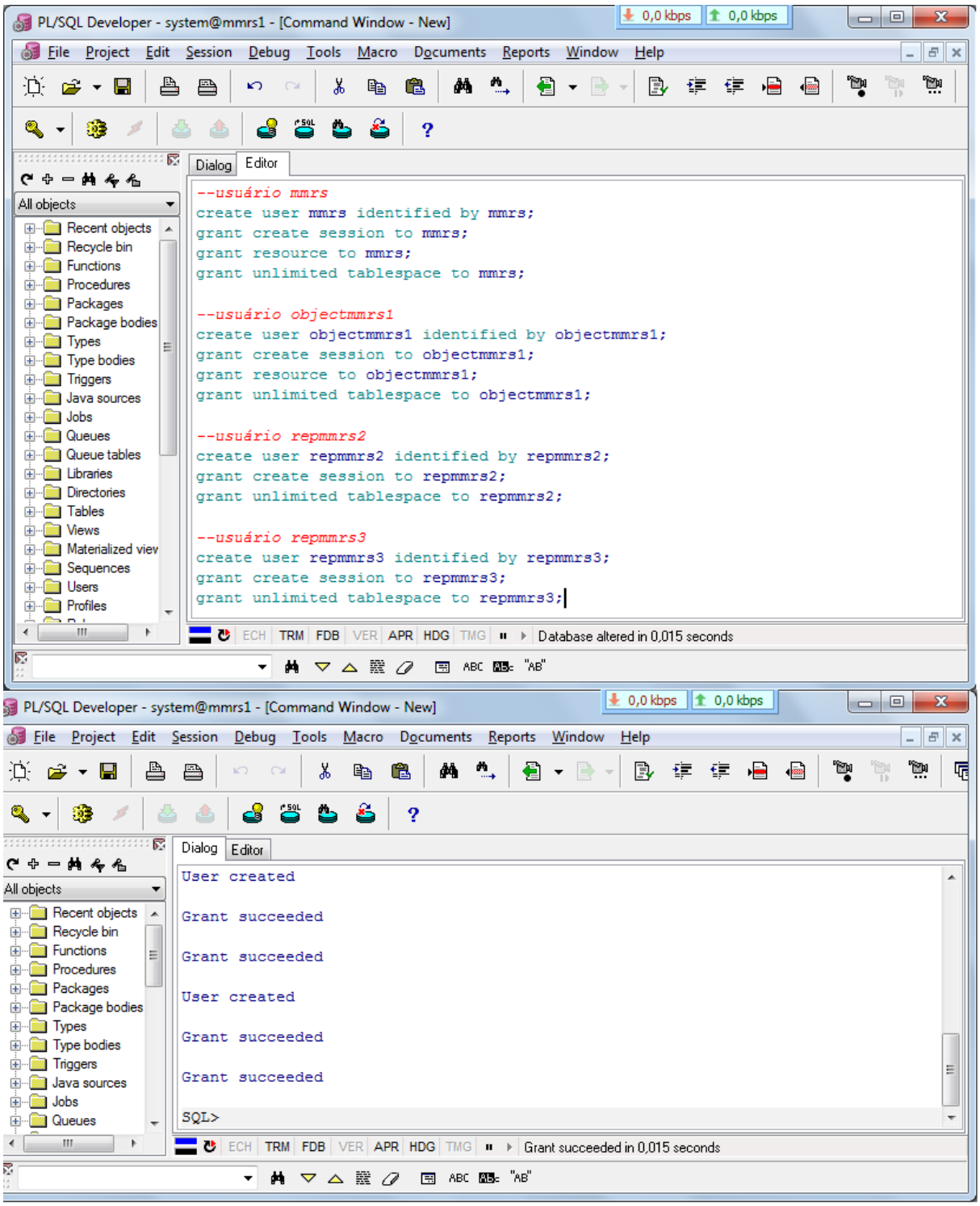


Figura 9 - Criação dos usuários do Banco de Dados

Criação das tabelas do usuário “mmrs”, as quais serão replicadas para os servidores slaves dos sites alternativos.

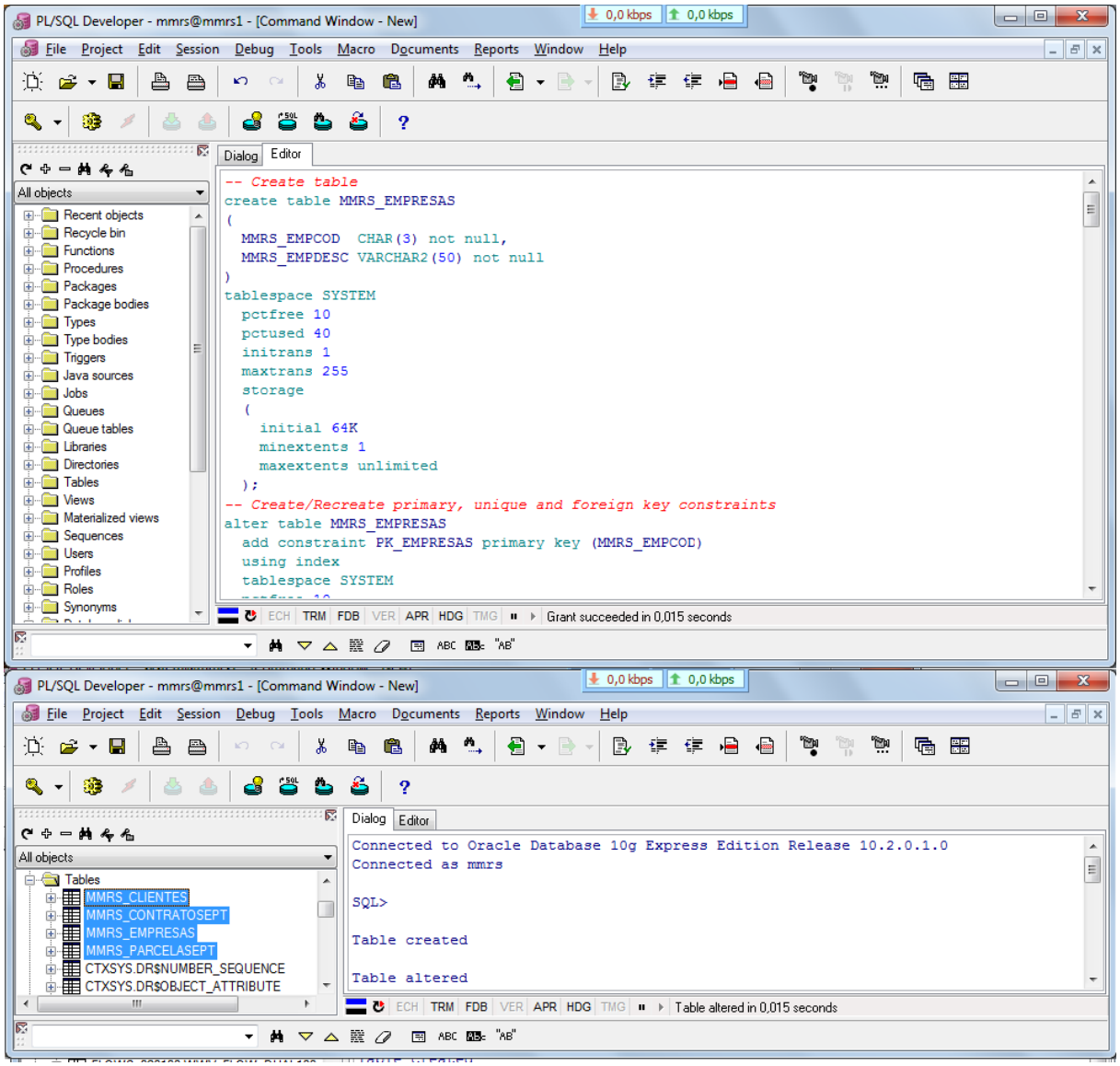


Figura 10 - Criação das tabelas que serão replicadas

Concessão de privilégios aos usuários de banco de dados (repmmrs2, repmmrs3 e objectmmrs1), para que possam manipular as pertencentes ao usuário mmrs, as quais serão replicadas.

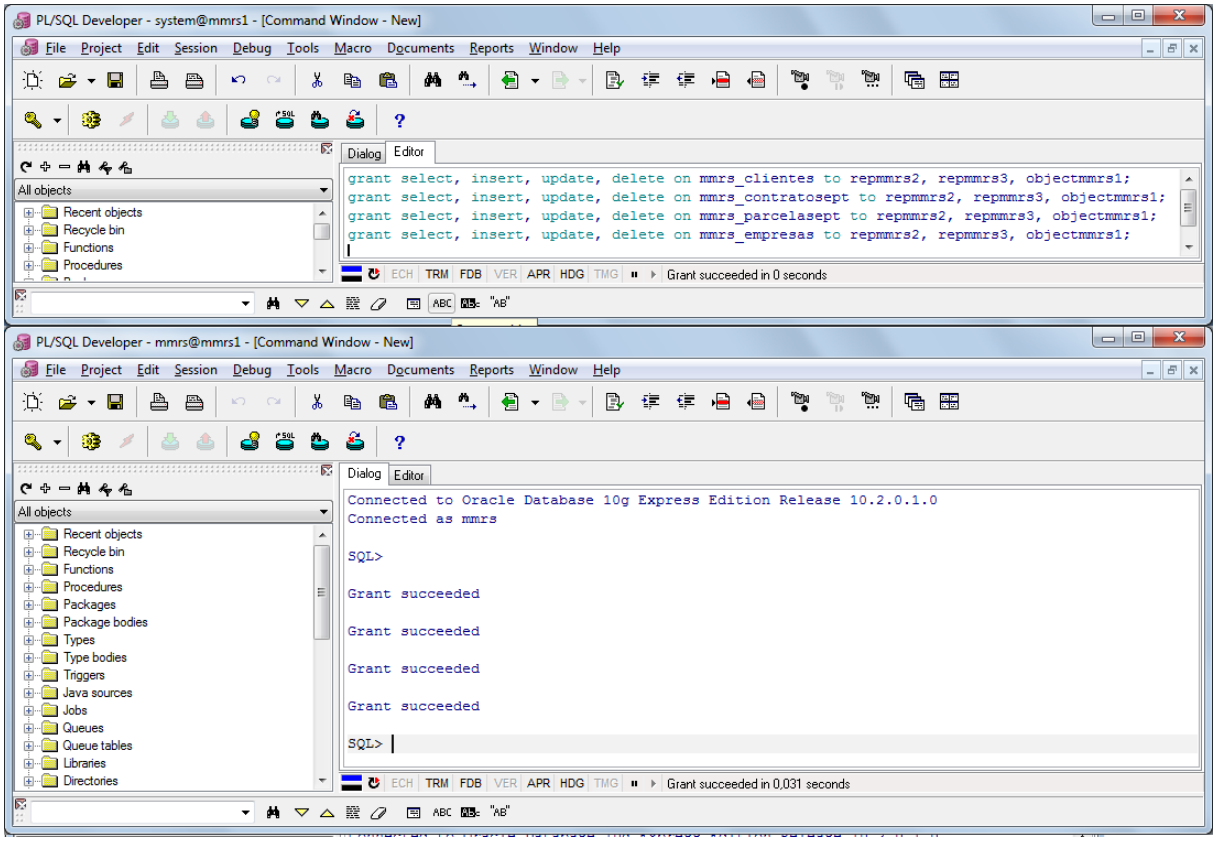


Figura 11 - Concessão de privilégios aos usuários de banco

Além dos privilégios já concedidos, é pré-requisito da ferramenta que o usuário proprietário do dicionário de dados (objectmmrs1) do replicador possua privilégio de “create any trigger”.

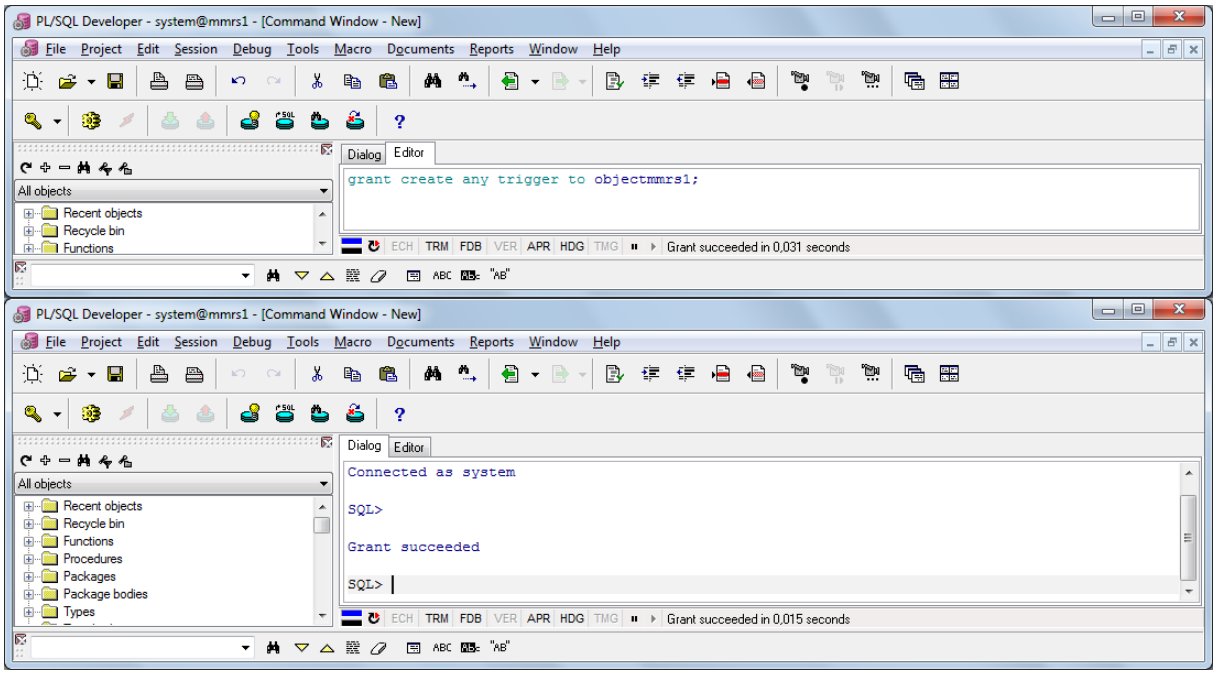


Figura 12 - Concessão de privilégio de "Create Any Trigger"

Criação do dicionário de dados do replicador em conformidade com o manual da ferramenta de replicação, a qual prevê um diretório para cada tipo de Banco de Dados a ser configurado. Neste caso o diretório do Oracle é o /home/object/objectmmrs/sql/oracle, onde estão os arquivos a serem executados (or_master_schema.sql e or_setup.sql).

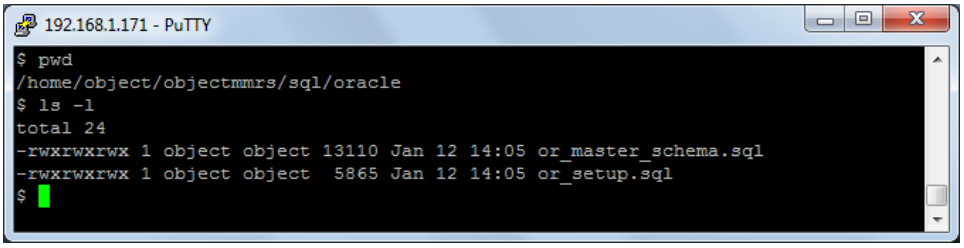


Figura 13 – Arquivos de criação do Dicionário de Dados

Execução dos scripts de criação do dicionário de dados do replicador (or_master_schema.sql).

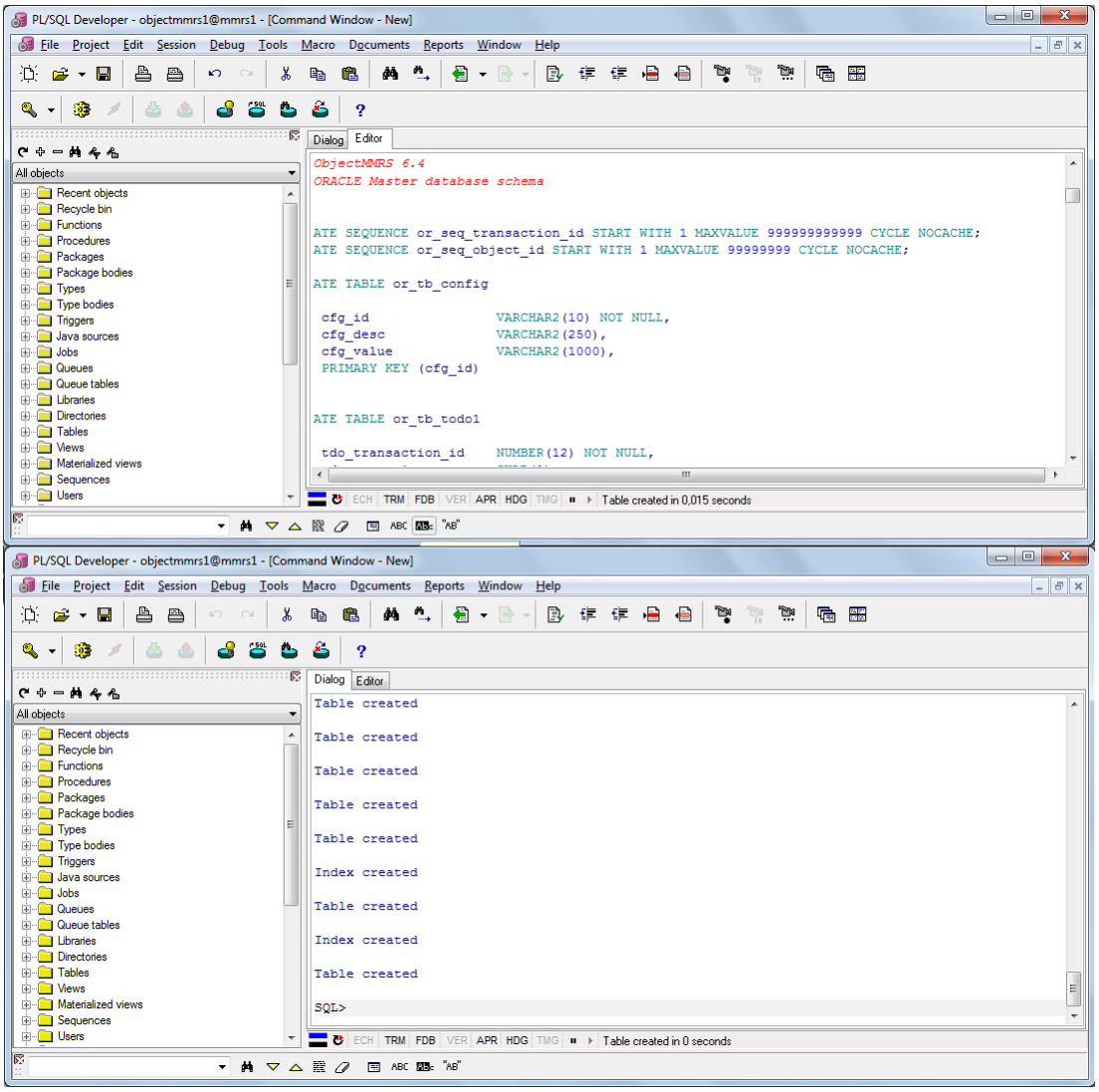


Figura 14 - Execução do Script de Criação do Dicionário de Dados

Execução do script de carga inicial nos parâmetros do dicionário de dados do replicador.

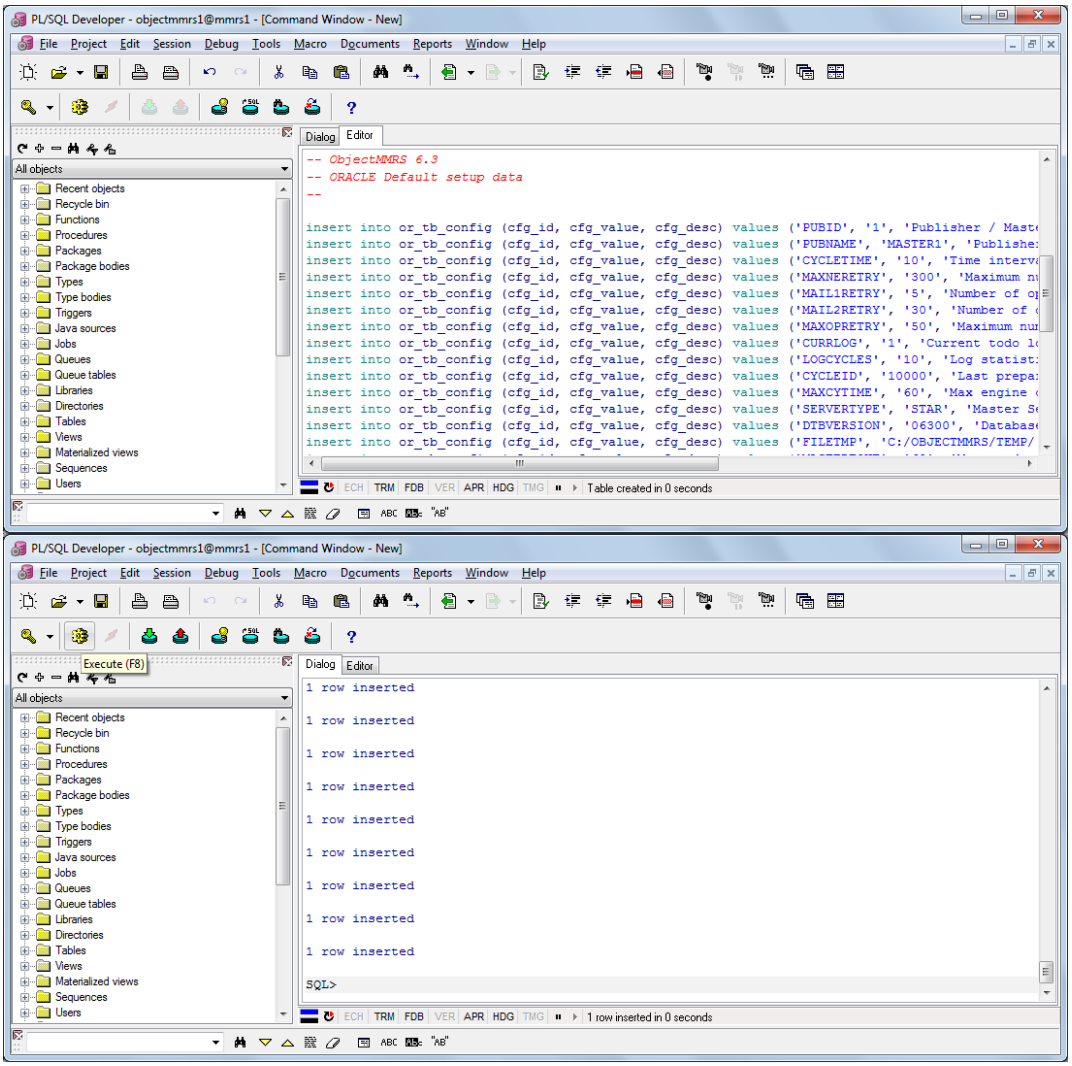


Figura 15 - Carga inicial do Dicionário de Dados

Criação das tabelas de log de replicação, conforme previsto no manual, com base nas tabelas modelo (or_tb_logoutput e or_tb_logoutput_t), criadas pelo script do dicionário da dados do replicador. As tabelas que estão sendo criadas são or_tb_logoutputmmrs2/ or_tb_logoutputmmrs2_t, as quais armazenarão os logs os dados a serem replicados ao servidor alternativo mmrs2 e as tabelas or_tb_logoutputmmrs3/ or_tb_logoutputmmrs3_t, as quais armazenarão os logs dos dados a serem replicados ao servidor alternativo mmrs3.

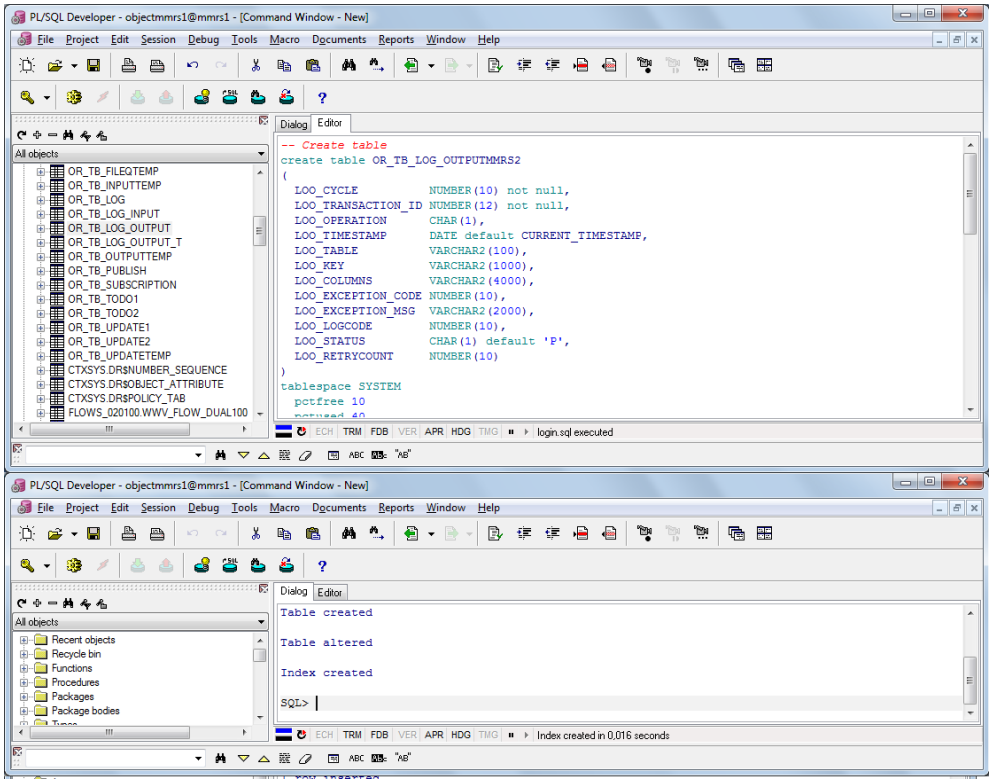


Figura 16 - Criação das tabelas de Log de replicação

Concessão de privilégio de “select” para os usuários repmmrs2 e repmmrs3, nas tabelas or_tb_* e “select e alter” nos sequences or_seq_*.

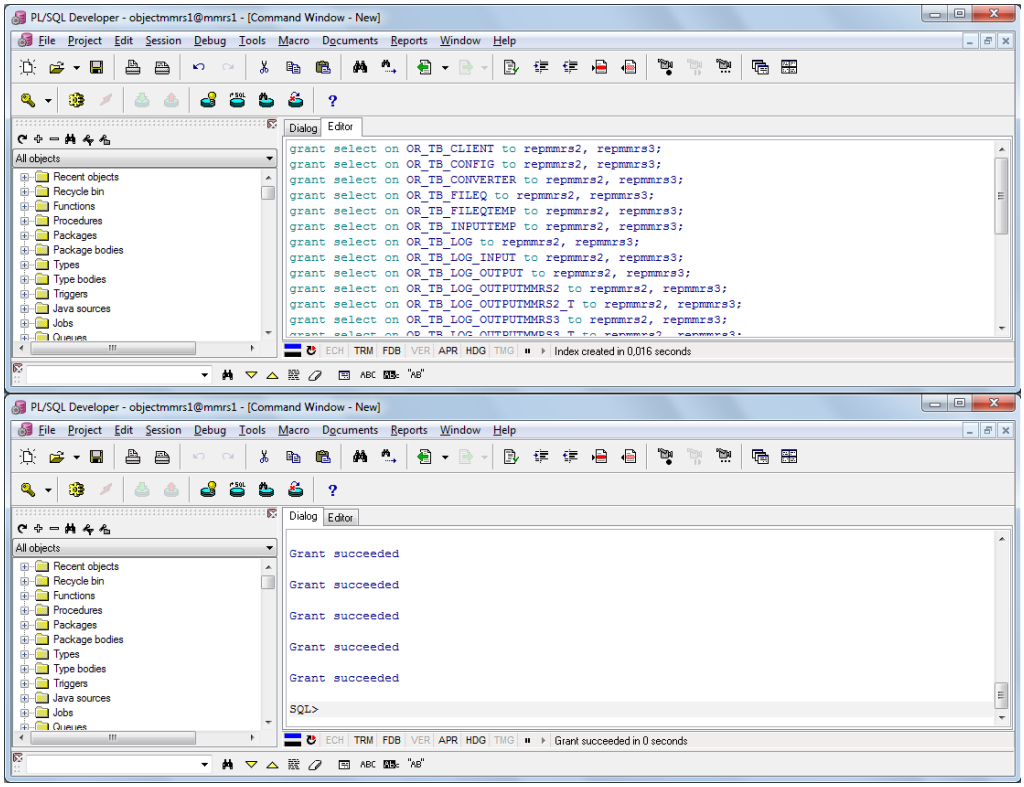


Figura 17 - Grants no dicionário de dados para os usuários de replicação

Concessão de privilégio de “select, insert, update e delete” para o usuário mmrs, nas tabelas or_tb_* e “select e alter” nos sequences or_seq_*.

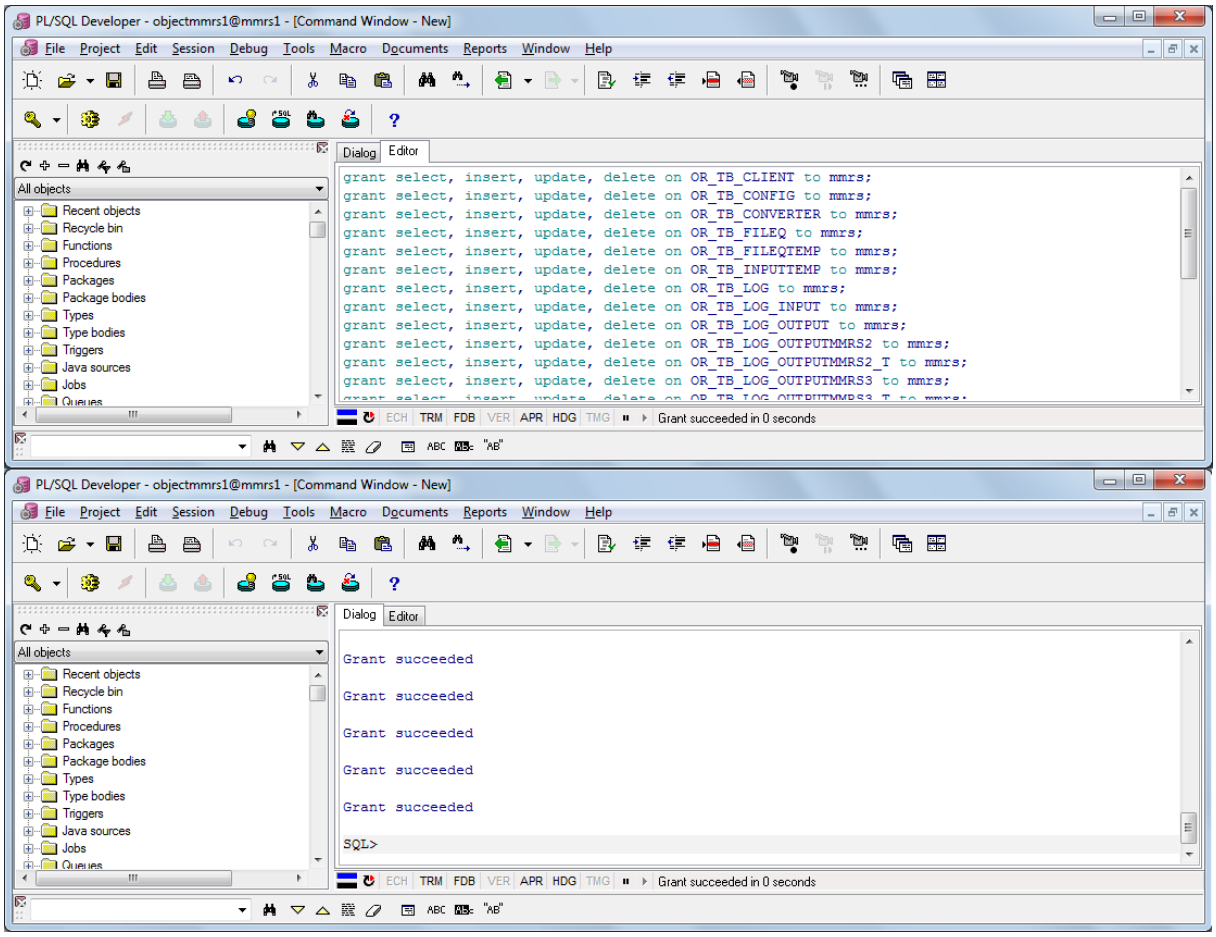


Figura 18 - Grants no dicionário de dados para o proprietário das tabelas

Após a preparação do dicionário de dados do replicador, é necessário configurar a aplicação no servidor Linux.

Originalmente os arquivos disponibilizados não estão prontos para execução no Linux então é necessário prepará-los antes do uso, tornando-os executáveis, conforme segue (Figuras 19 e 20).

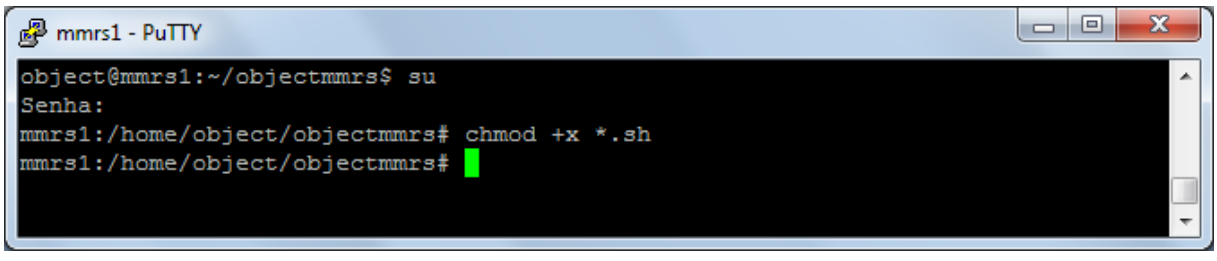


Figura 19 - Tornando scripts Executáveis

Convertendo os arquivos *.properties e *.sh para o formato Unix.

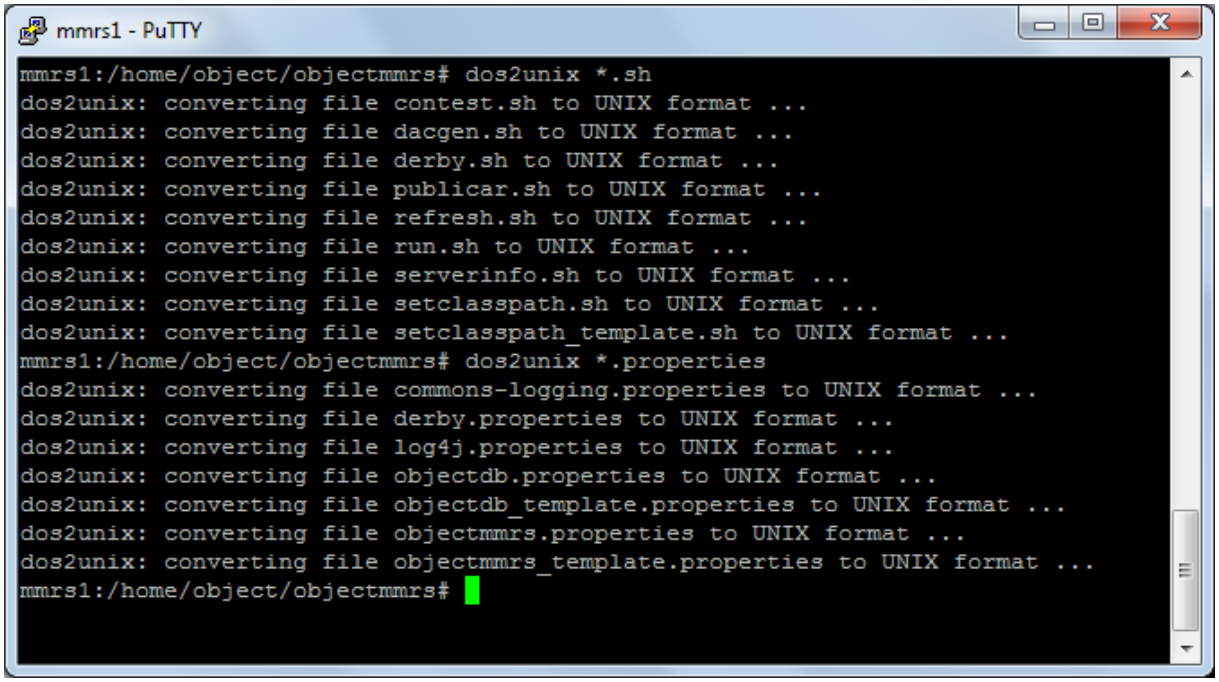


Figura 20 - Conversão de arquivos para o formato Unix

Agora que foi tudo devidamente preparado é iniciado o processo de configuração do replicador. Inicialmente serão configurados os arquivos objectmmrs.properties e objectdb.properties.

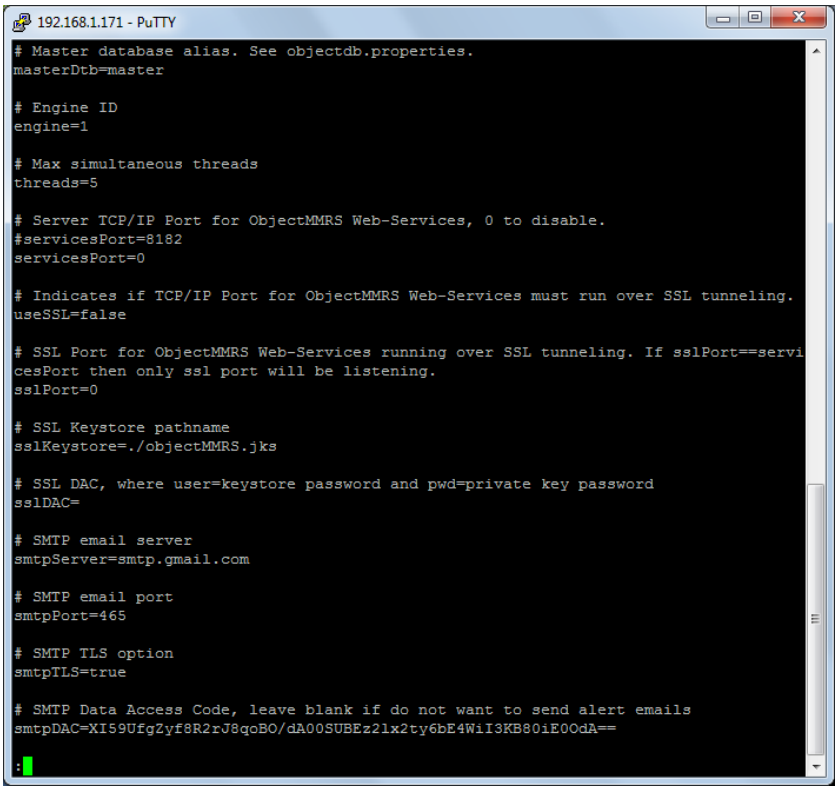


Figura 21 - Arquivo objectmmrs.properties

No parâmetro databases devem ser identificados todos os servidores que participarão do processo de replicação. Neste caso, master é o servidor que está enviando a replicação (mmrs1 neste caso), slave1 será o servidor mmrs2 e slave2 o servidor mmrs3. Estes nomes são super importantes e devem estar em conformidade com a parametrização realizada nas tabelas or_tb_config (master) e or_tb_client (slaves).

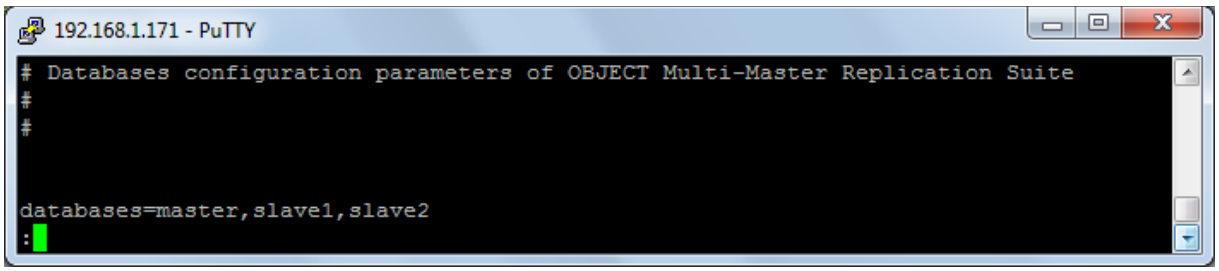


Figura 22 – Arquivo objectdb.properties (Parâmetro databases)

Nas linhas seguintes são detalhadas as propriedades do master e dos slaves, as quais precisam ter alguns parâmetros ajustados: dtbType, dtbDriver, dtbUrl, dtbDAC e connectionTestStatement, deixando conforme abaixo (Figuras 22, 23 e 24).

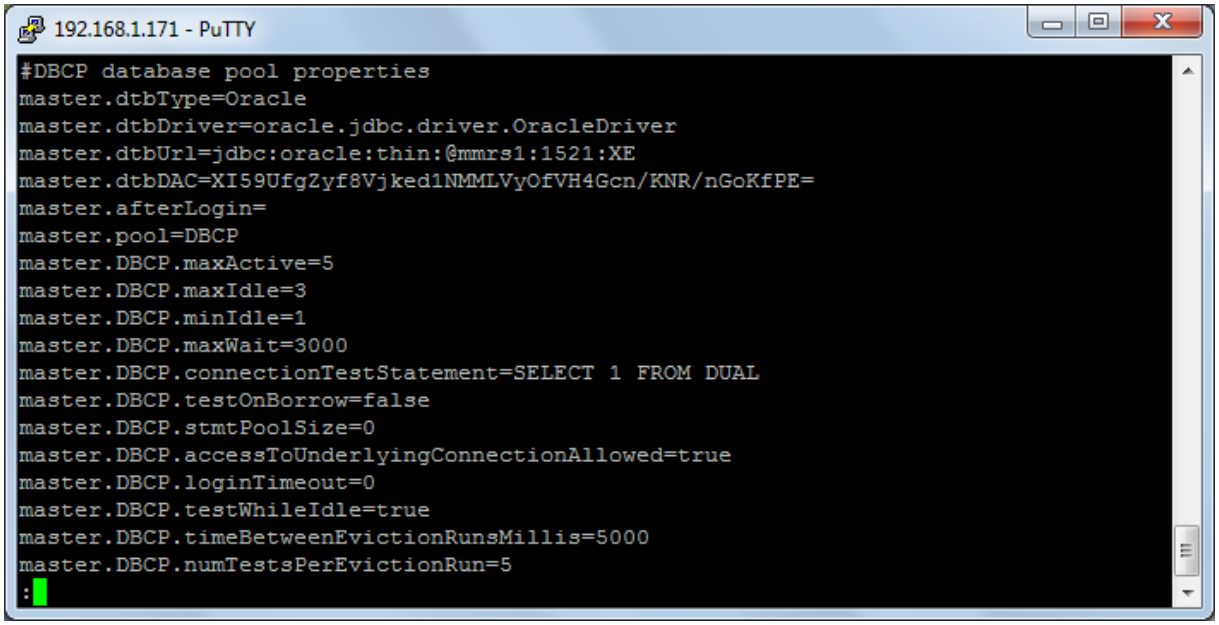


Figura 23 - Arquivo objectdb.properties (Parâmetro do master)

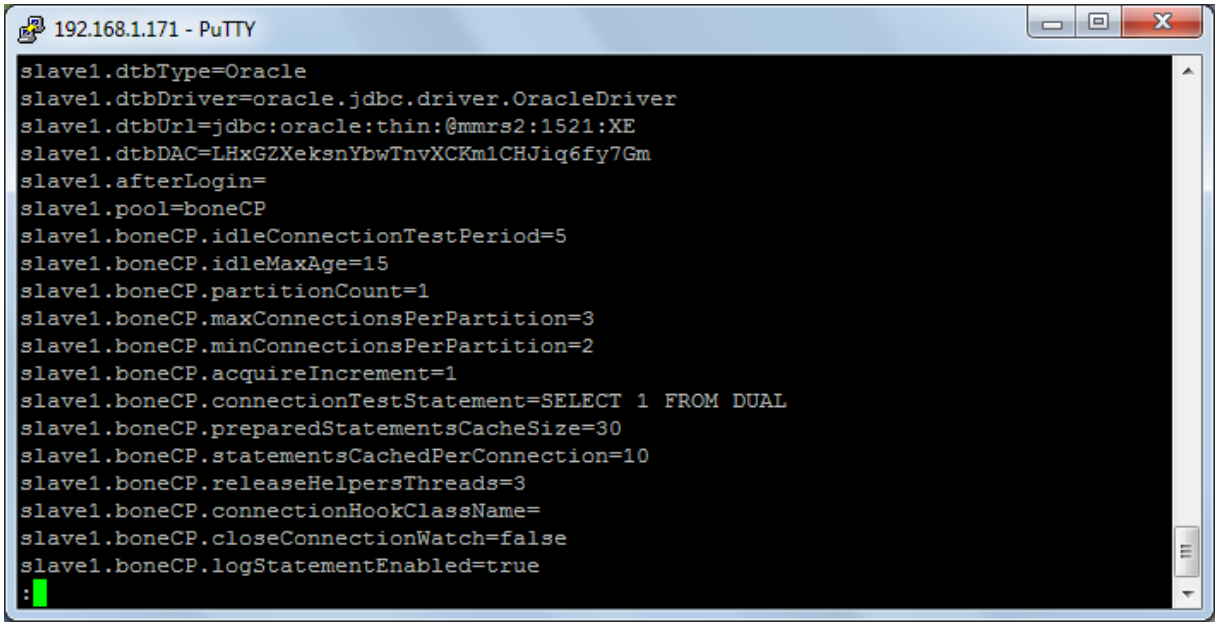


Figura 24 - Arquivo objectdb.properties (Parâmetro do Slave1)

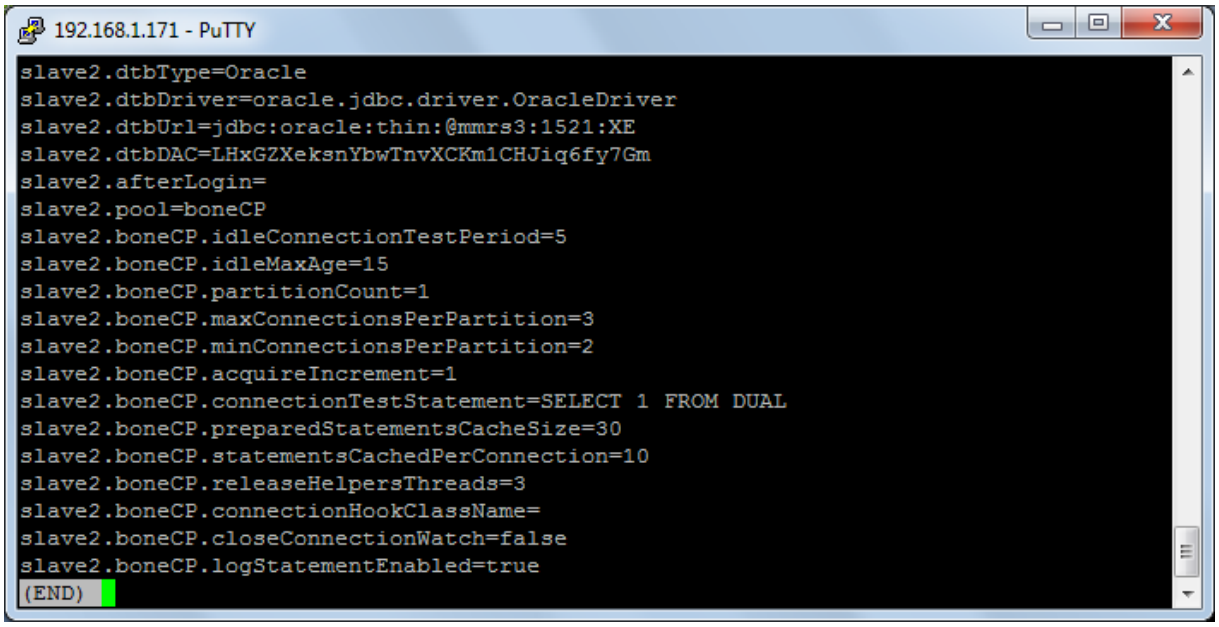


Figura 25 - Arquivo objectdb.properties (Parâmetro do Slave2)

Para o funcionamento da ferramenta foi baixado no site da Oracle o driver de conexão “ojdbc14.jar”, o qual foi alocado no diretório “/home/object/objectmmrs/lib/” do Linux. Após isso a configuração do CLASSPATH foi realizada conforme instruções do manual.

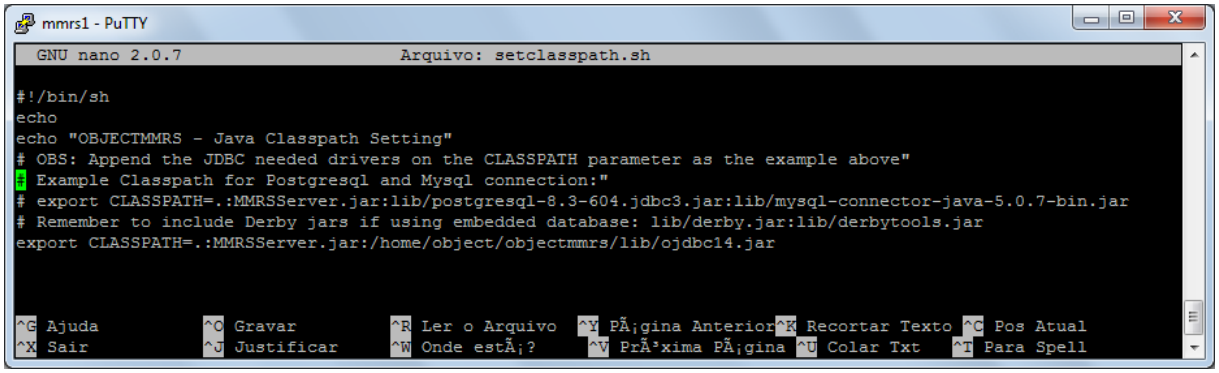


Figura 26 - Configuração do Classpath (Conforme Manual)

A figura abaixo apresenta uma configuração alternativa do Classpath, a qual é realizada no profile do Linux para garantir que a variável sempre seja setada e exportada para o Sistema Operacional independente sempre que este for carregado.

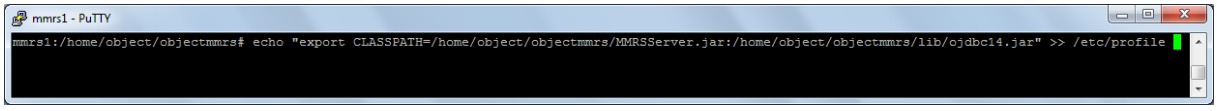


Figura 27 - Export do Classpath no Profile do Linux

Neste momento é possível a realização de um teste preliminar para teste de conexão com as base de dados utilizando o script “contest.sh” do replicador.

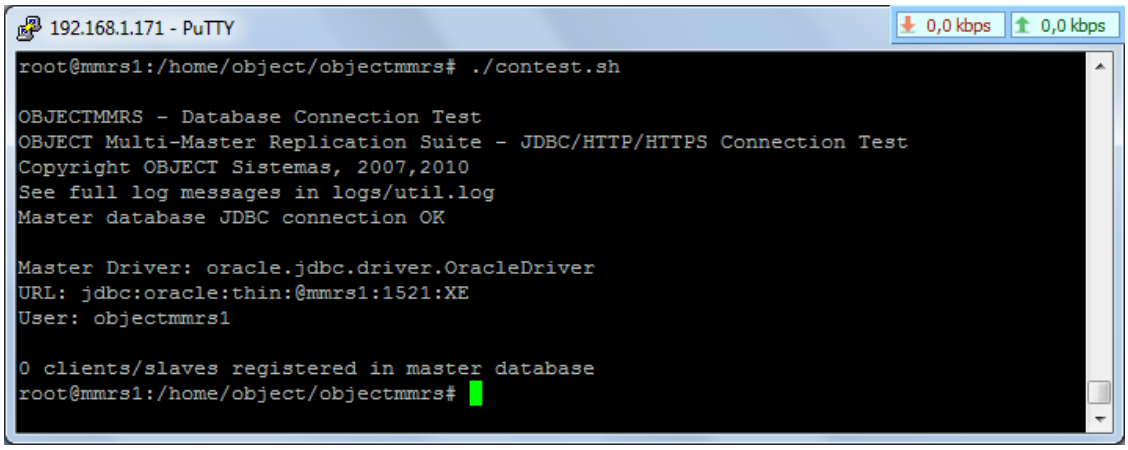


Figura 28 - Teste preliminar de conexão com as bases de dados

O teste acima foi realizado parcialmente com sucesso, pois a avaliação apresentou o status de OK para a conexão com a base de dados master, no entanto não conectou com nenhuma slave pelo fato não haver nenhum registro na tabela or_tb_client do dicionário de dados do replicador. Isto será solucionado após a conclusão dos passos seguintes.

Edição do arquivo publicar.sh que se encontra no diretório raiz do replicador, preenchendo os parâmetros em conformidade com o manual da ferramenta.

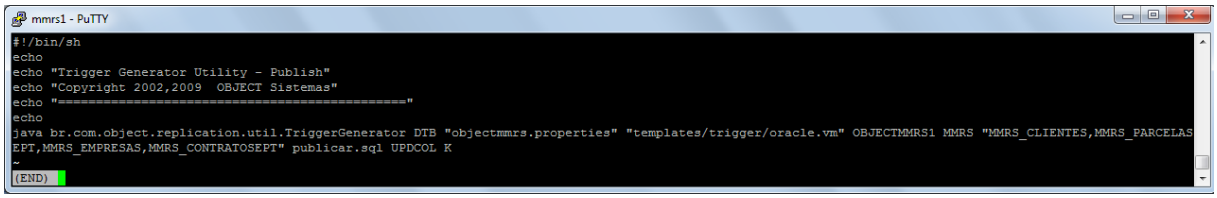


Figura 29 - Arquivo Publicar.sh

Execução do script publicar.sh, após a edição realizada na figura acima, o qual gerou o arquivo publicar.sql, que se trata das triggers de replicação que devem ser criadas.

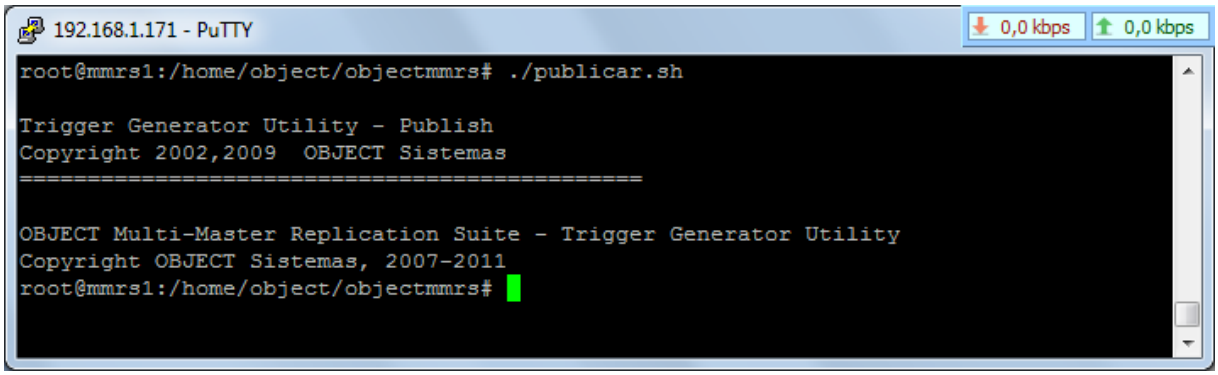


Figura 30 - Execução do script publicar.sh

Visão parcial do arquivo publicar.sql:

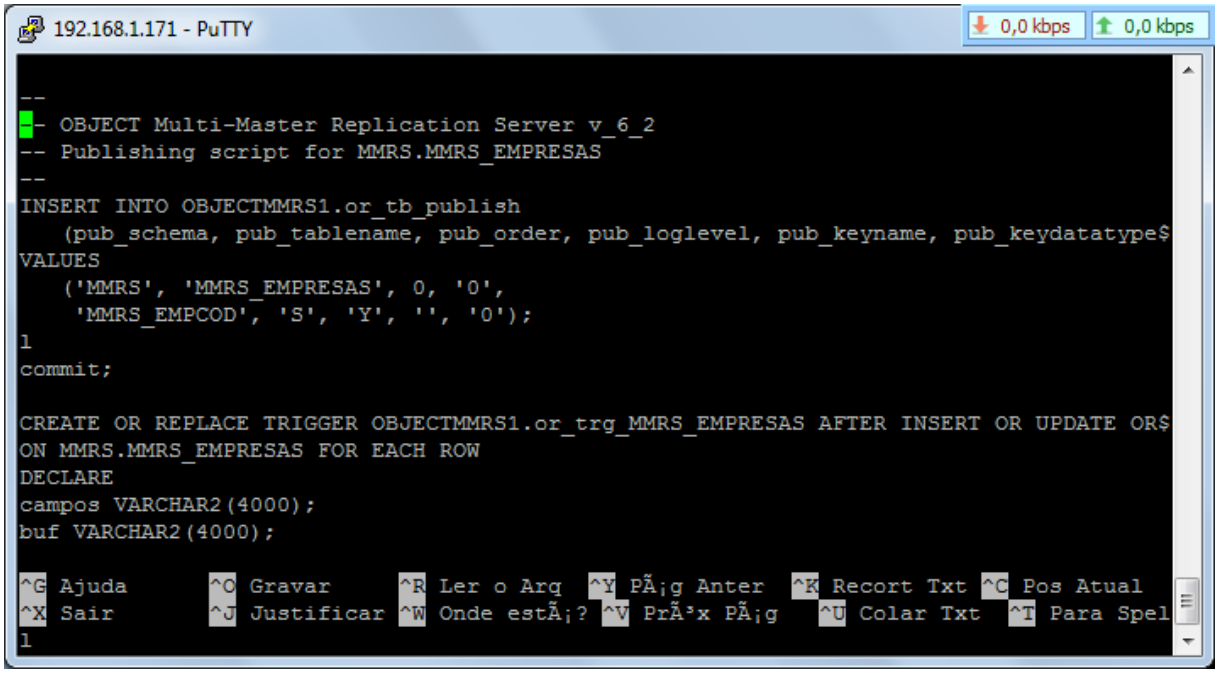


Figura 31 - Visão parcial do arquivo publicar.sql

Publicação das tabelas que serão replicadas e criação das triggers no dicionário de dados do replicador, com os dados contidos no arquivo publicar.sh (Figura 30).

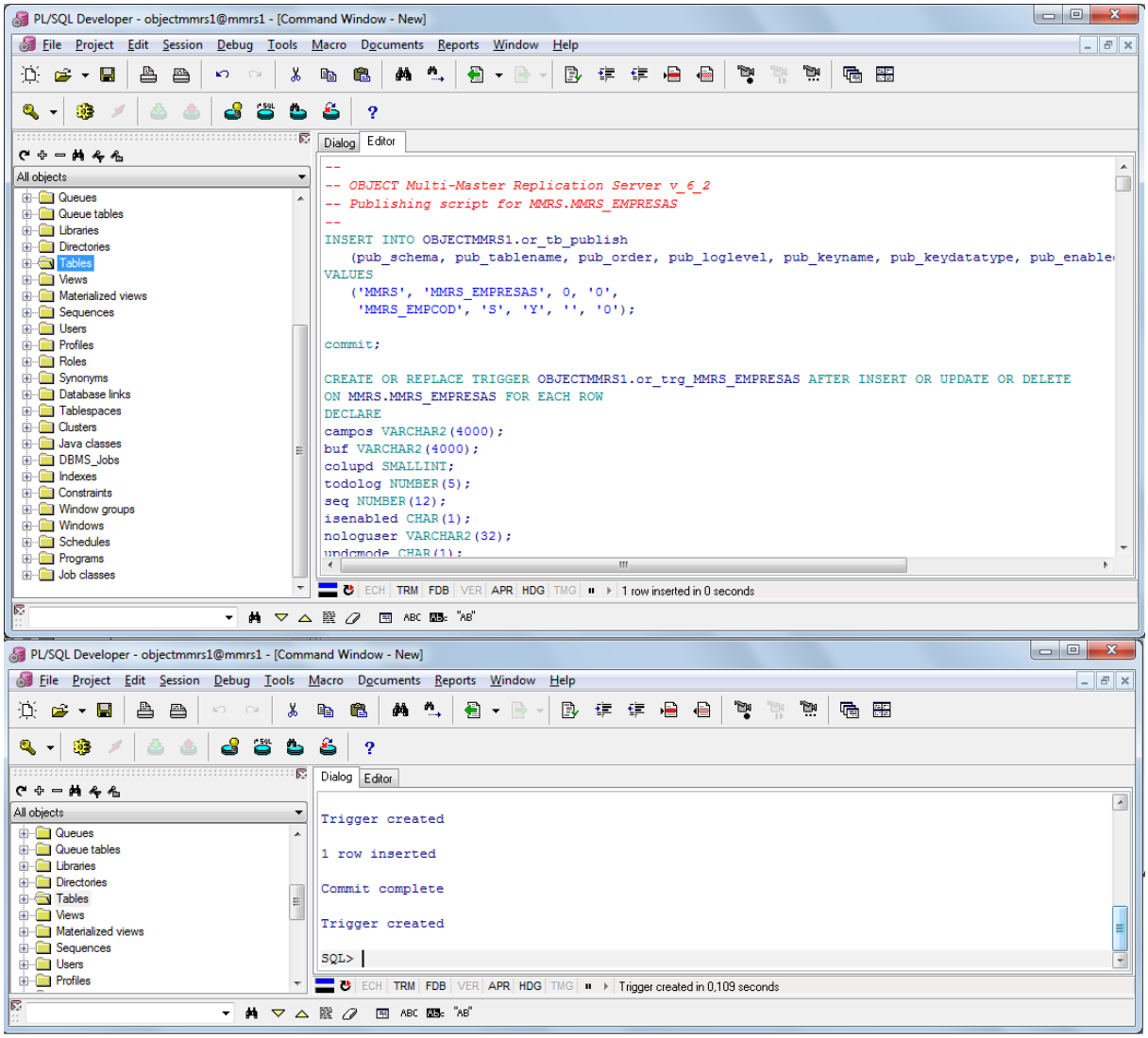


Figura 32 – Criação de Triggers de replicação

Registro dos slaves na tabela or_tb_client do dicionário de dados.

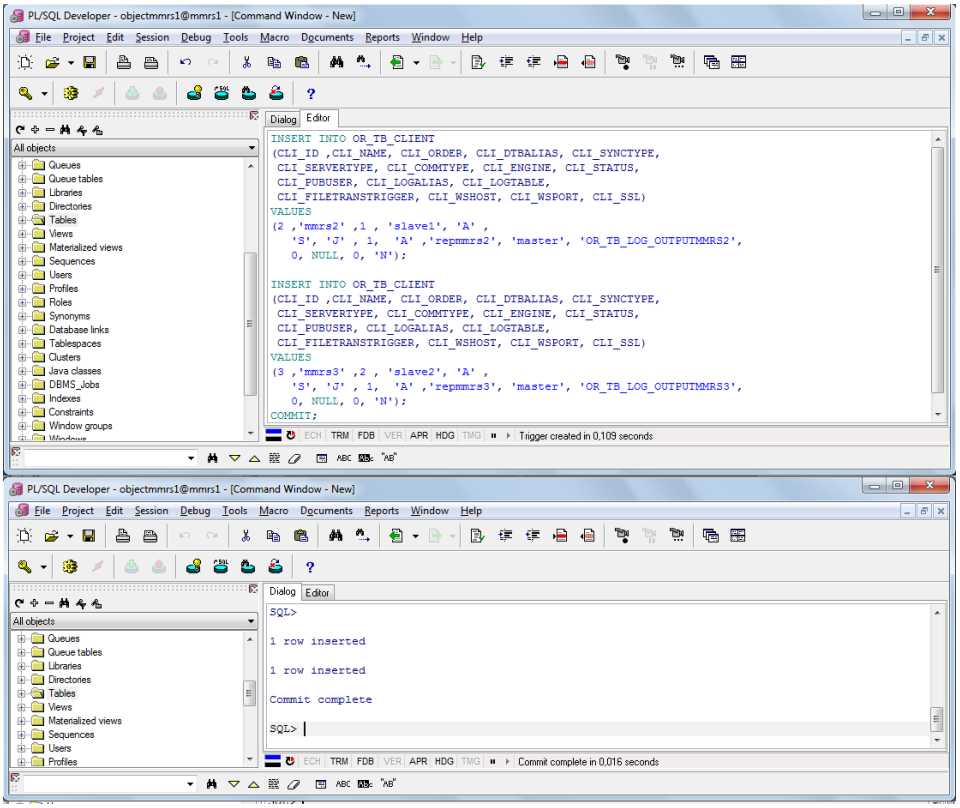


Figura 33 - Cadastro dos servidores slaves de replicação

O teste de conexão preliminar (figura 27) apresentou problema de conexão com os slaves porque eles não estavam cadastrados na tabela or_tb_config do replicador. Como no passo acima (figura 33) foi realizado o cadastramento dos slaves, é possível realizar um novo teste para identificar que o resultado será OK tanto para o servidor master quanto para os slaves.

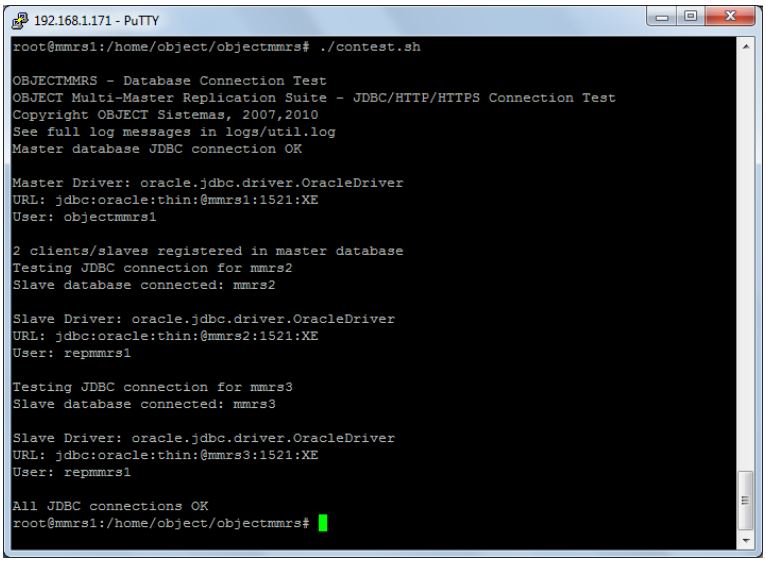


Figura 34 - Teste final de conectividade entre servidores

Cadastramento das assinaturas conforme script disponibilizado no manual de instalação do replicador.

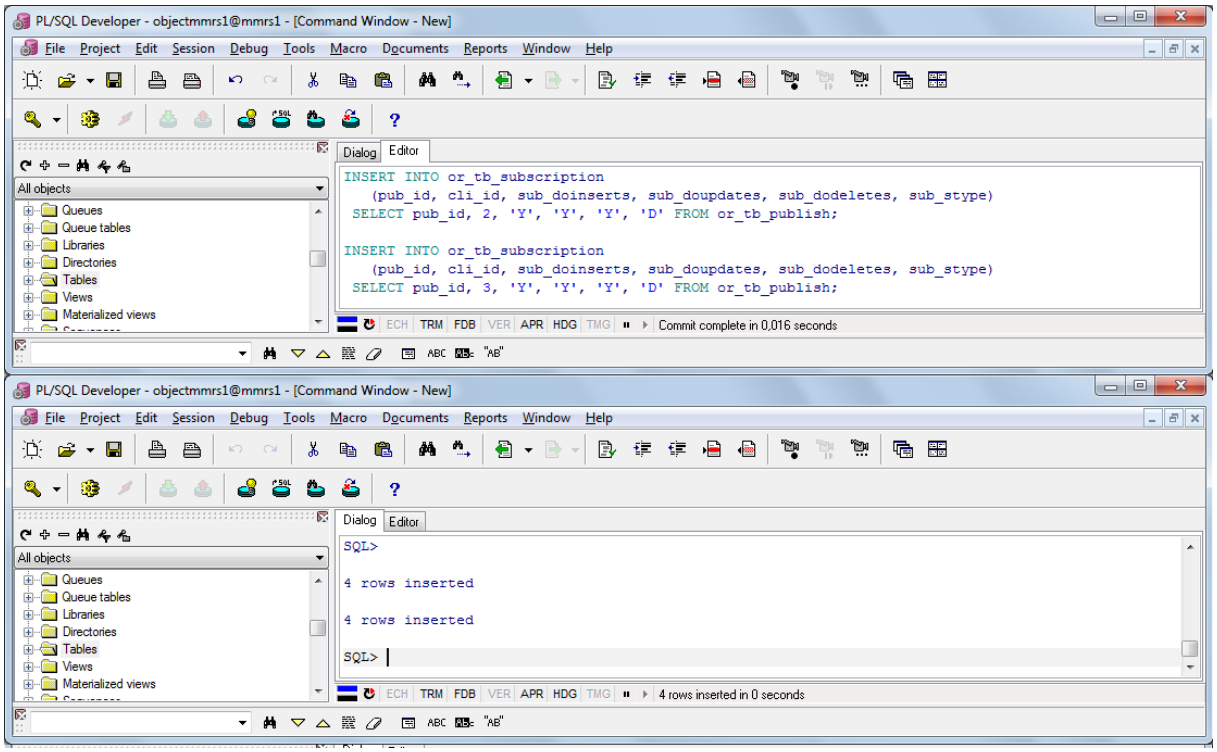
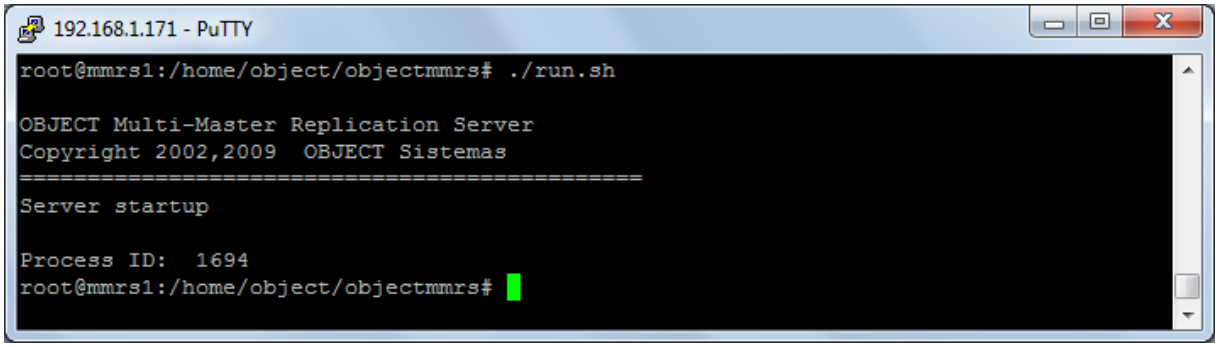


Figura 35 - Cadastramento de Assinaturas

Após estas configurações, o serviço de replicação foi iniciado com a execução do script "./run.sh", o qual se encontra no diretório raiz do replicador (/home/object/objectmmrs).



Todas as configurações realizadas na sede foram também realizadas nos sites alternativos com a finalidade de haver a replicação bidirecional.

4.7. TESTES DE REPLICAÇÃO

Com a finalidade de validar se o processo de replicação está ocorrendo adequadamente entre todos os sites, a seguir são realizadas algumas operações de Insert, Update e Delete em um servidor master de cada vez e, em seguida está sendo avaliado se os eventos estão se replicados aos demais servidores slaves. Em cada figura está destacado com um retângulo o servidor que está sofrendo a alteração ou ajuste.

4.7.1. Operações de Insert.

Inserção de um registro na tabela mmrs_empresas do servidor mmrs1 a verificação do dado utilizando o select.

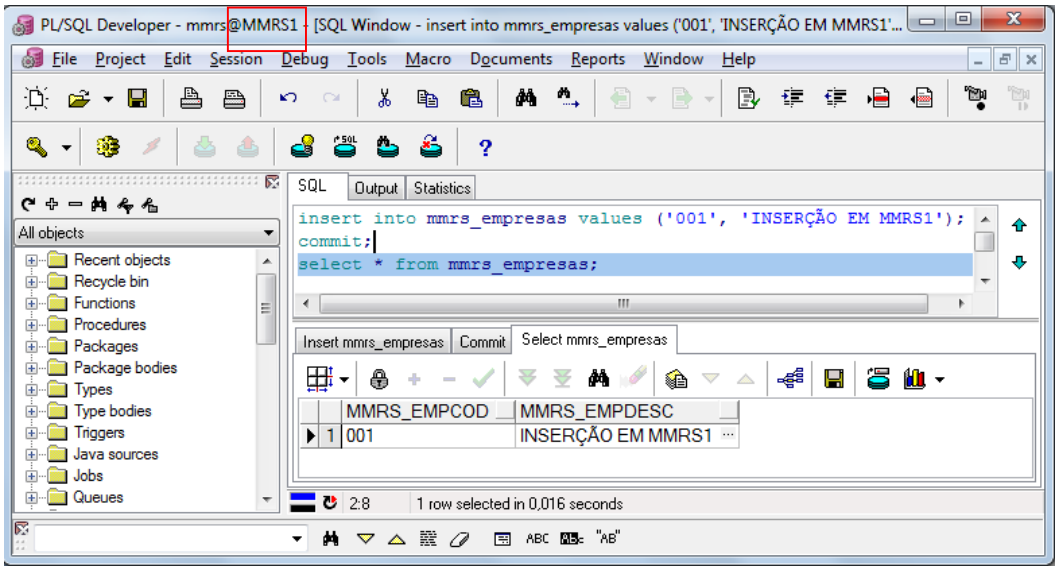


Figura 36 – Inserção e Consulta em mmrs1

Consulta da tabela mmrs_empresas no servidor mmrs2 para verificar se o registro replicado pelo servidor mmrs1 foi gravado com sucesso.

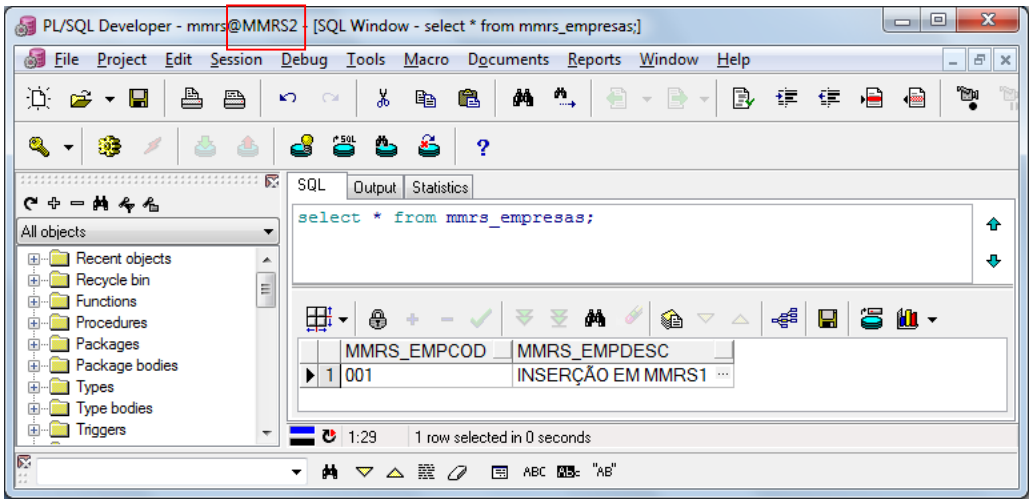


Figura 37 - Consulta em mmrs2

Consulta da tabela mmrs_empresas no servidor mmrs3 para verificar se o registro replicado pelo servidor mmrs1 foi gravado com sucesso.

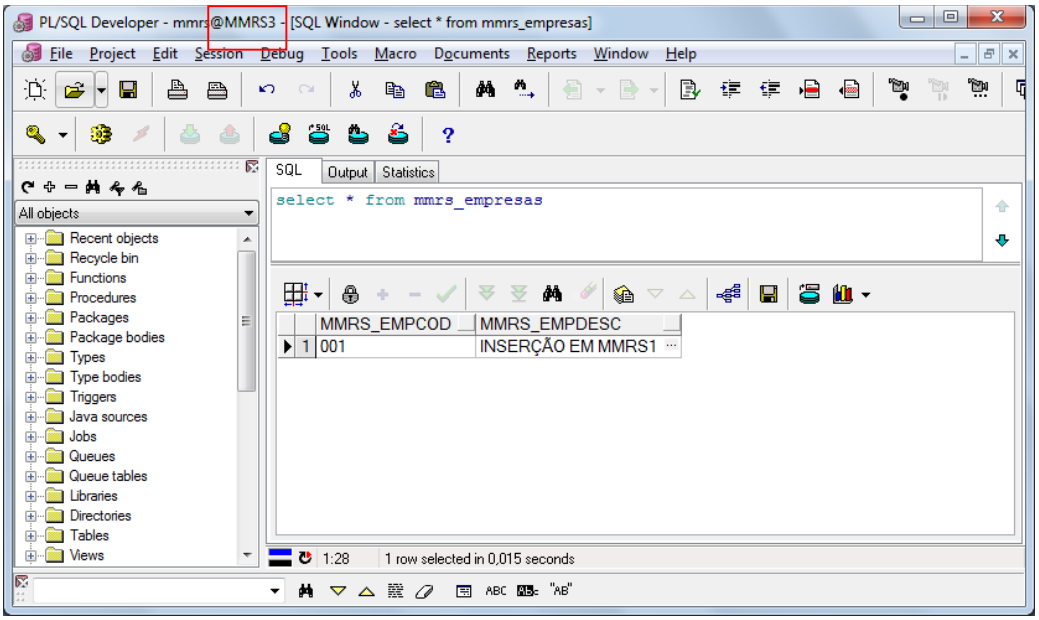


Figura 38 - Consulta em mmrs3

Inserção de mais um registro na tabela mmrs_empresas do servidor mmrs2 e a respectiva consulta através de select.

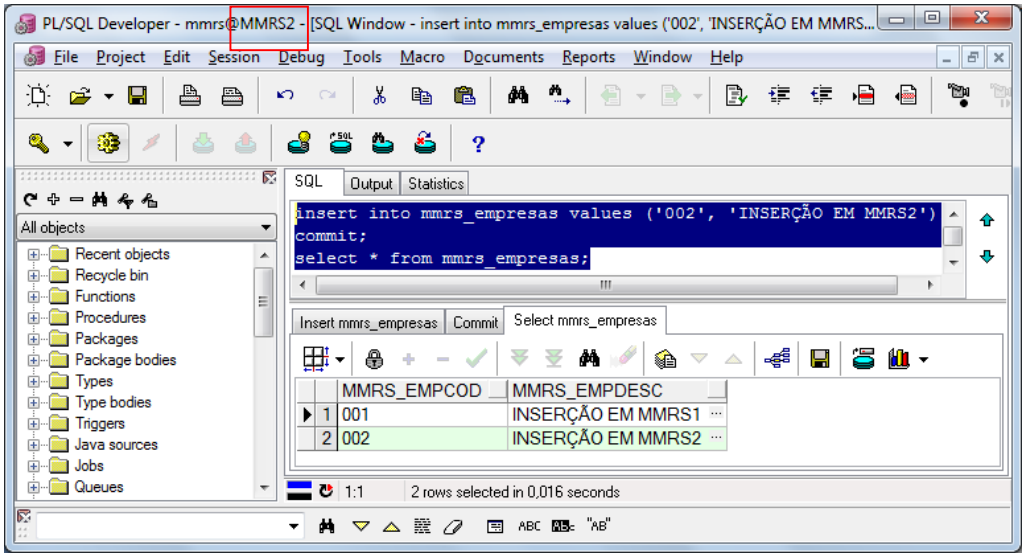


Figura 39 - Inserção e Consulta em mmrs2

Inserção de um registro na tabela mmrs_empresas do servidor mmrs3 e verificação do registro inserido.

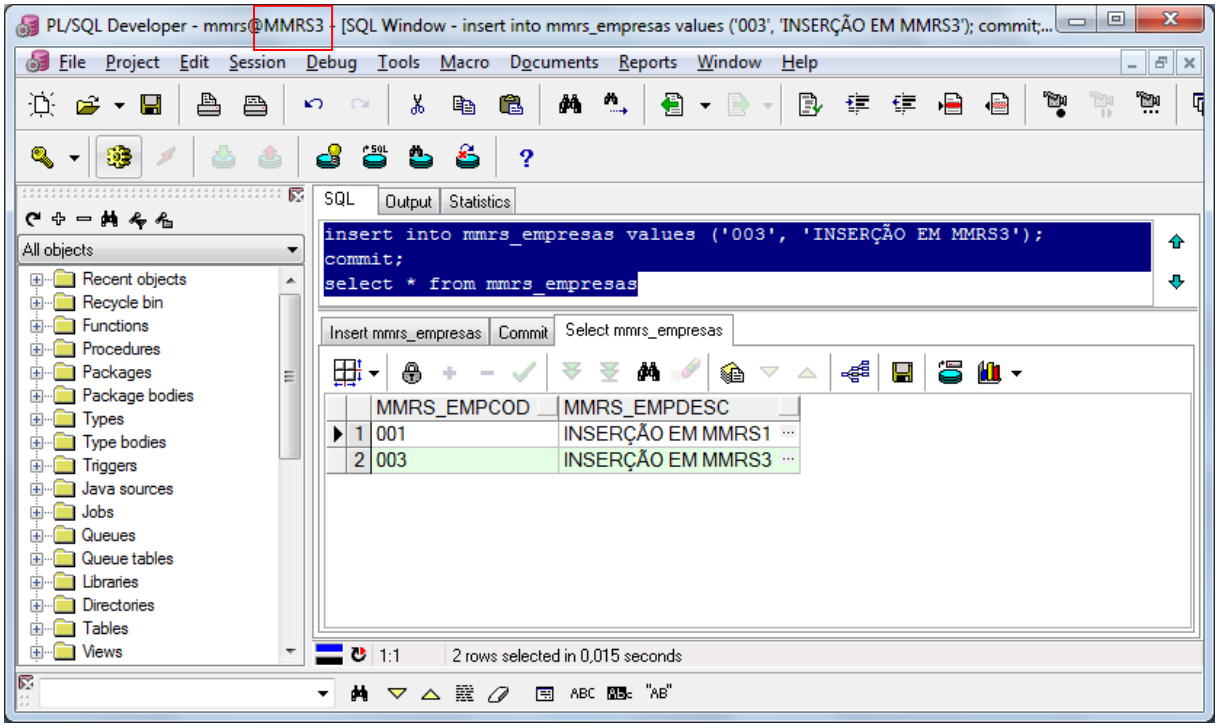


Figura 40 - Inserção e Consulta em mmrs3

Consulta ao servidor mmrs1, no qual está evidente que os registros inseridos nos servidores mmrs2 e mmrs3 já foram enviados pelos replicadores daqueles sites e já estão disponíveis na tabela deste servidor.

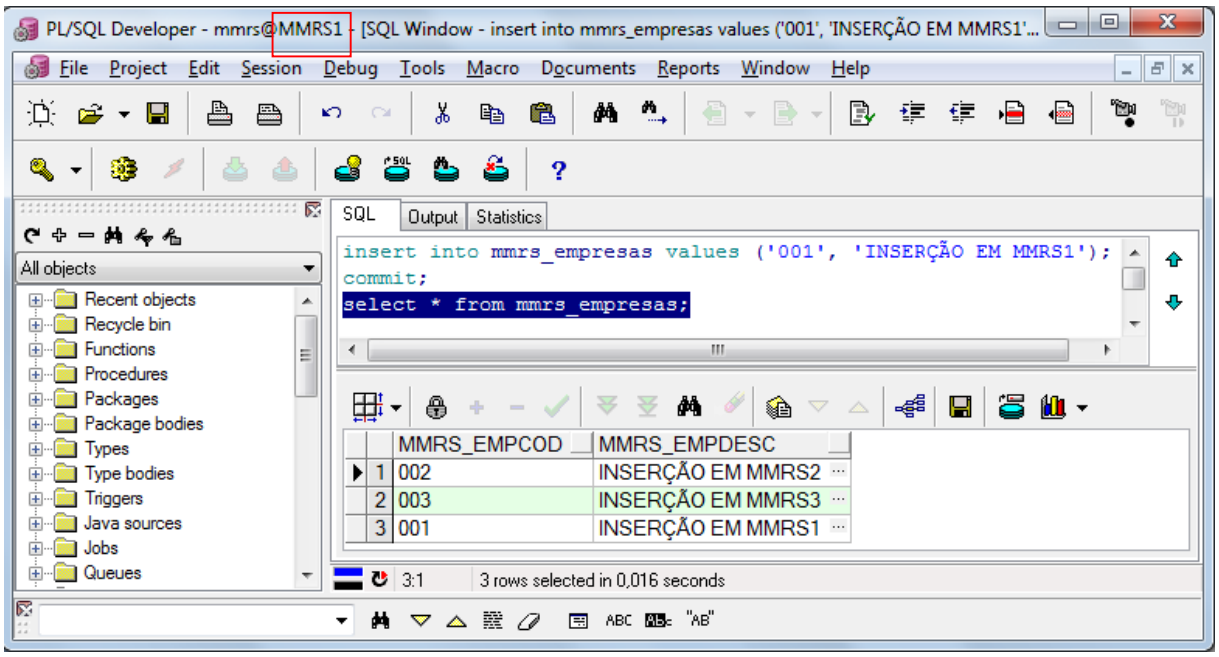


Figura 41 - Consulta em mmrs1

Consulta ao servidor mmrs2 e identificação de que o registro inserido em mmrs3 já está disponível neste servidor.

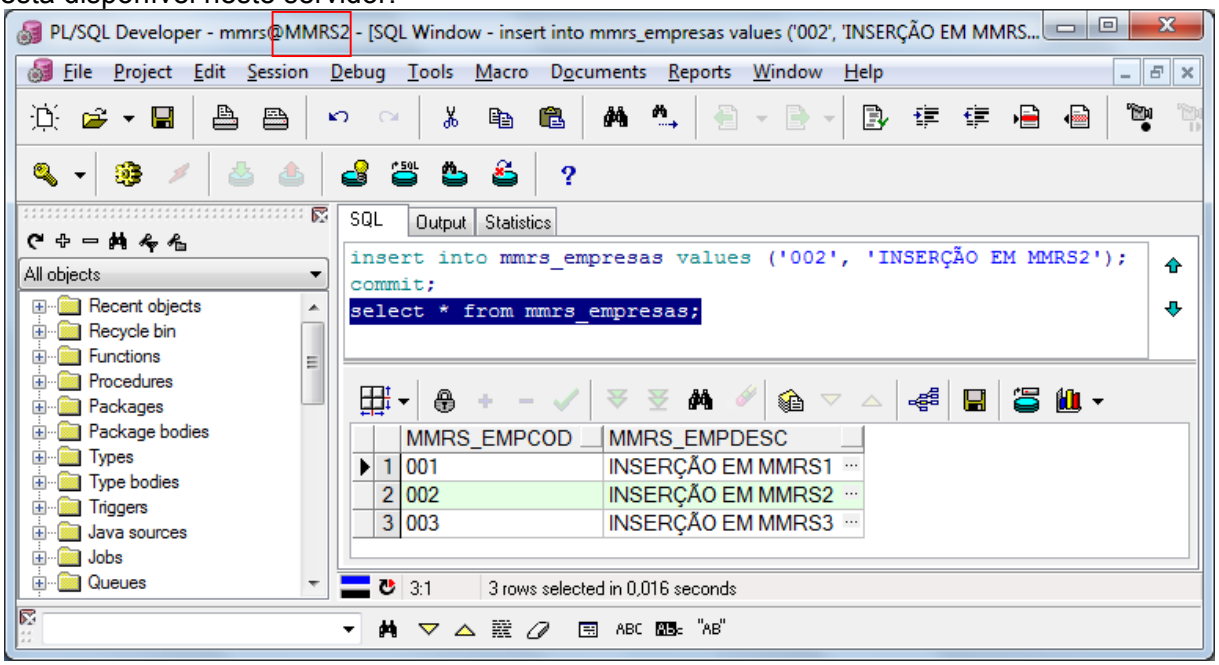


Figura 42 - Consulta em mmrs2

Consulta ao servidor mmrs3 e identificação de que o registro inserido em mmrs2 já está replicado para este servidor.

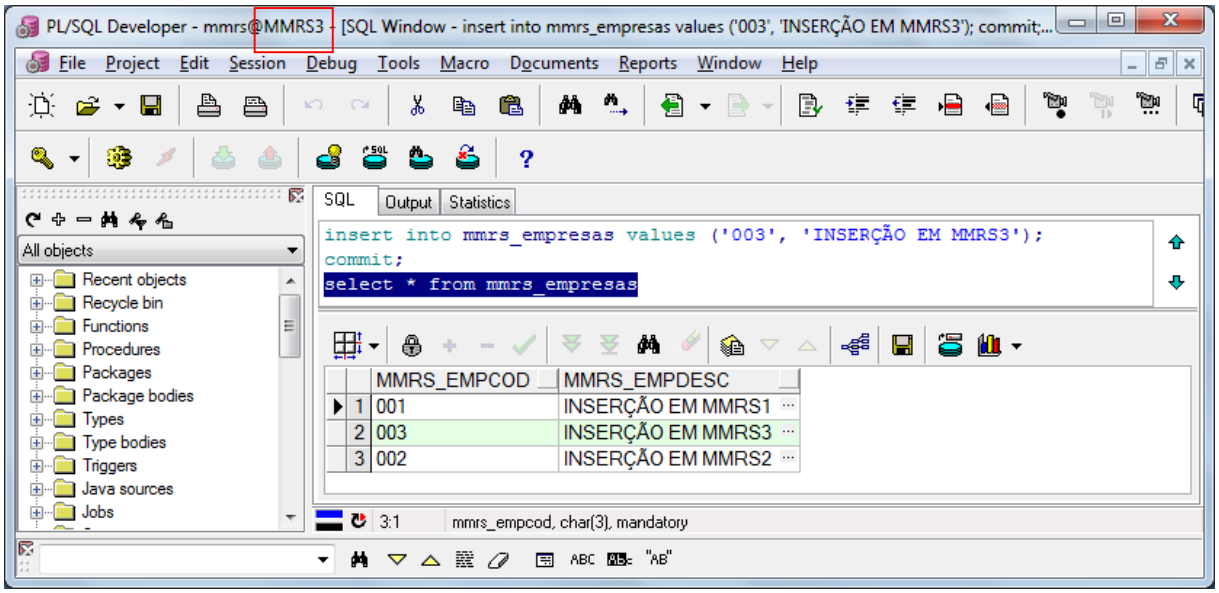


Figura 43 - Consulta em mmrs3

4.7.2. Operações de Update

Alteração de registro no servidor mmrs3, para testar a propagação das operações de update.

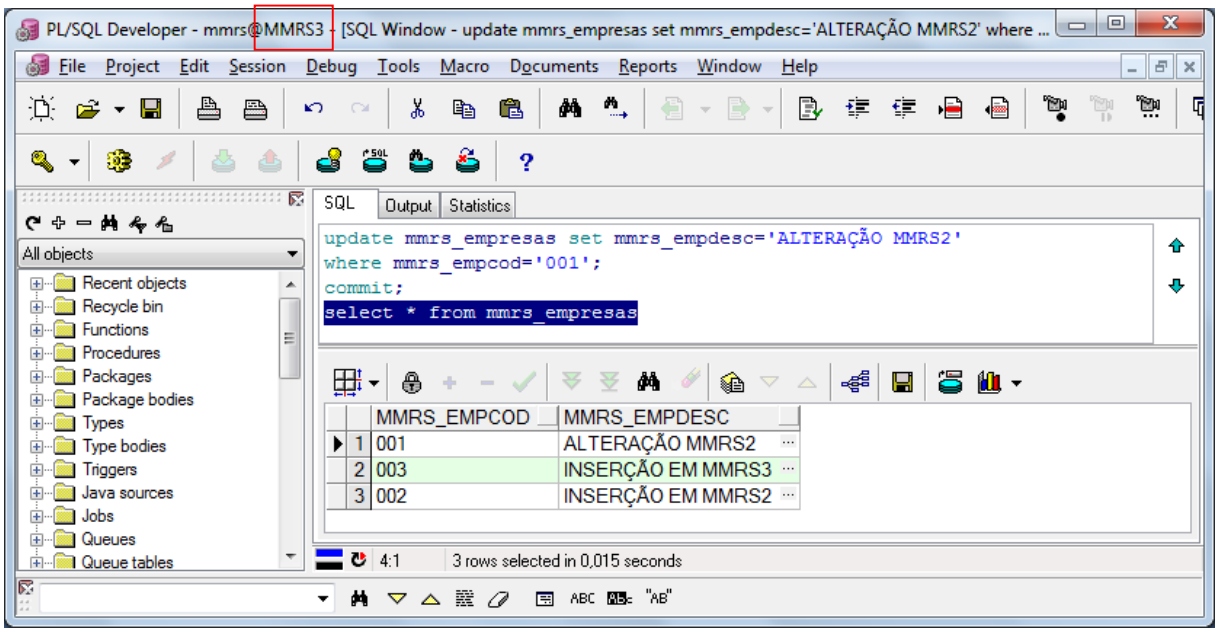


Figura 44 - Alteração e Consulta aos Dados

Consulta no servidor mmrs1 para verificar se o registro (001) alterado no servidor mmrs3 foi replicado corretamente.

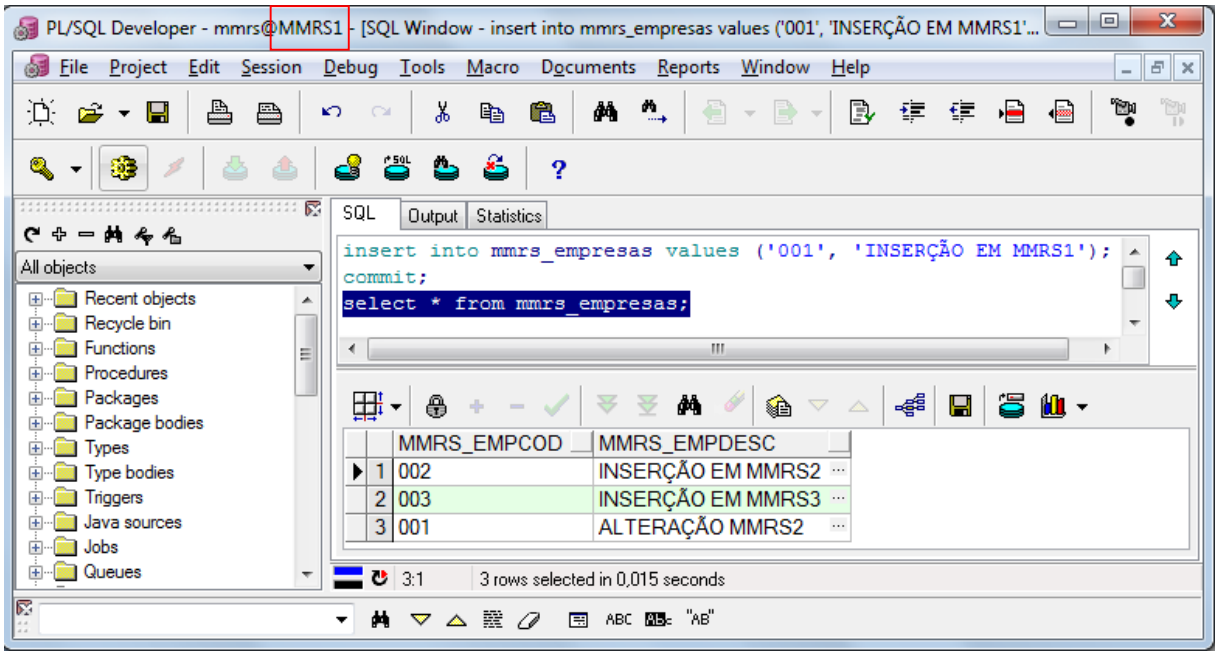


Figura 45 - Consulta de dado alterado replicado ao mmrs3

Consulta no servidor mmrs2 para verificar se o registro (001) alterado no servidor mmrs3 foi replicado corretamente

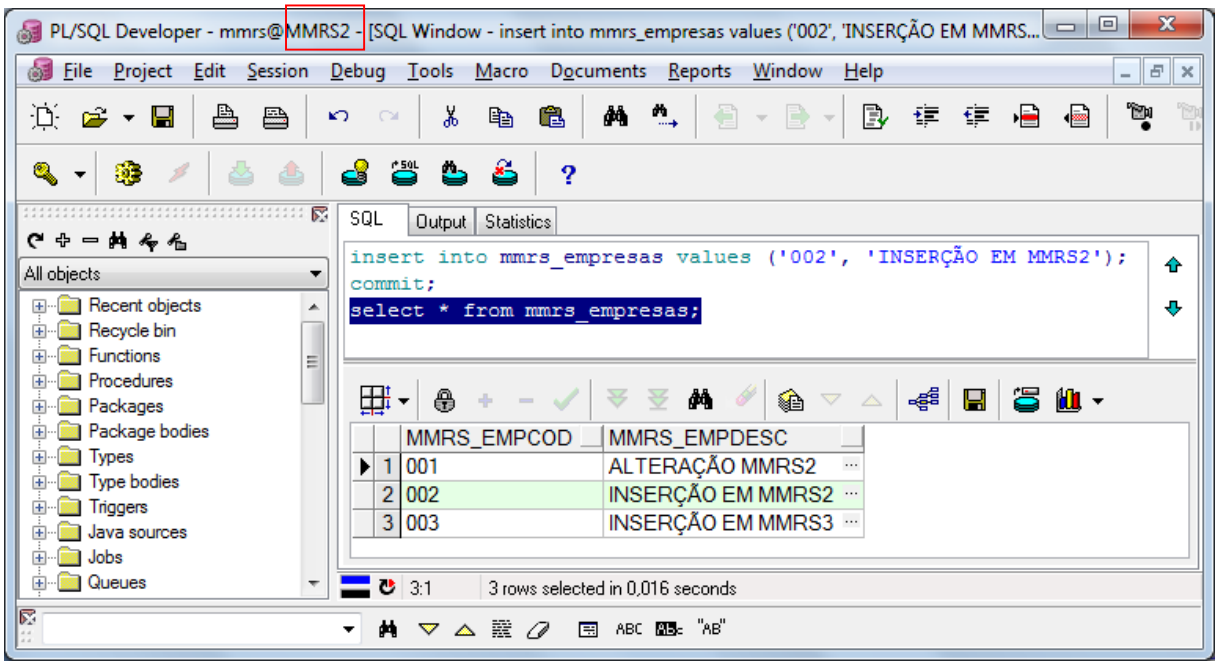


Figura 46 - Consulta de dado alterado replicado ao mmrs2

4.7.3. Operações de Delete

Exclusão de registro no servidor mmrs2, para testar a propagação das operações de delete.

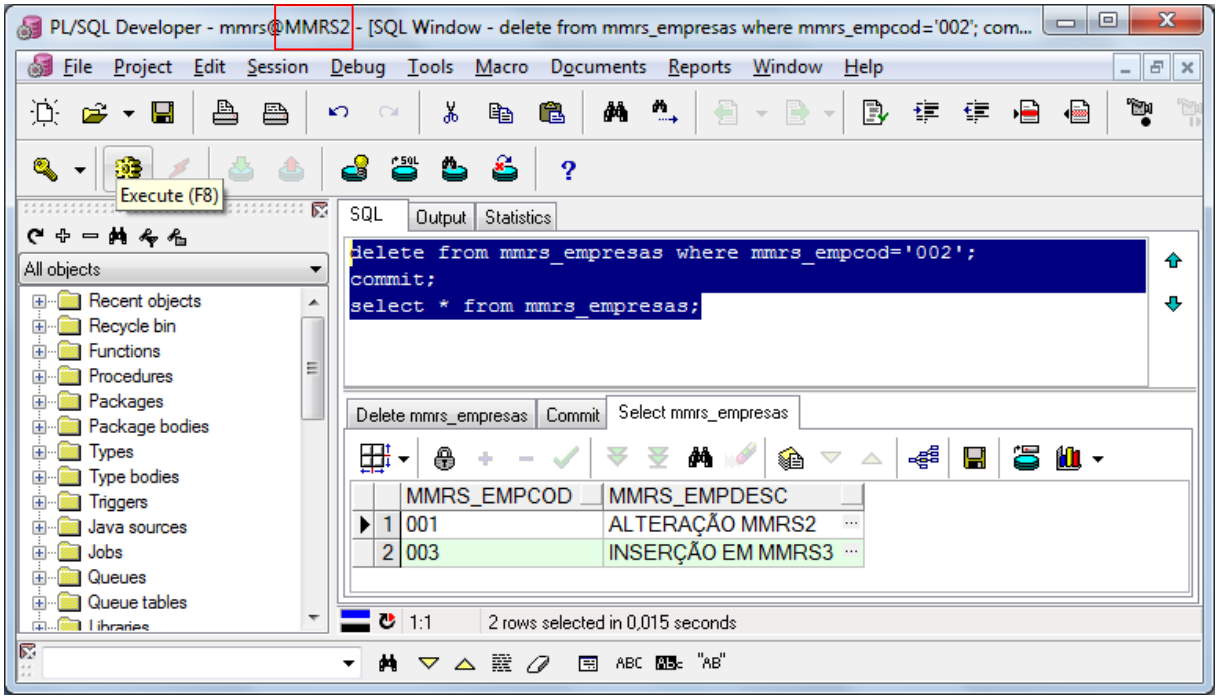


Figura 47 - Exclusão e consulta a dados

Consulta no servidor mmrs1 para verificar se o registro (002) excluído no servidor mmrs2 foi replicado corretamente.

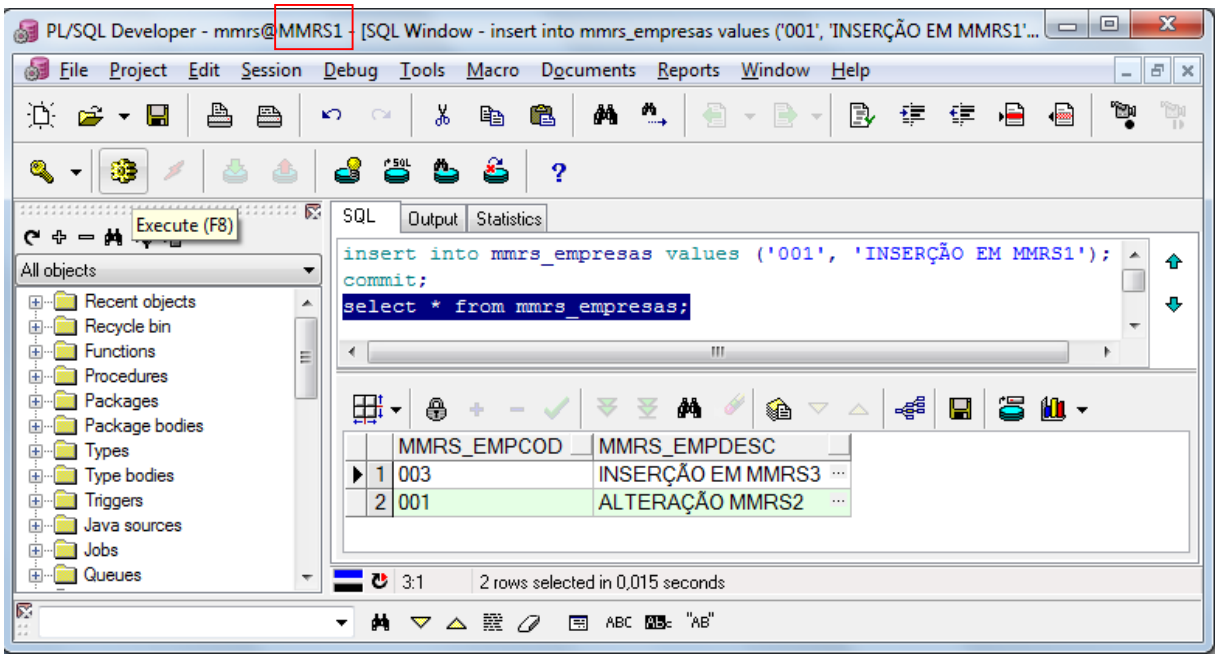


Figura 48 - Consulta a replicação em mmrs1

Consulta no servidor mmrs3 para verificar se o registro (002) excluído no servidor mmrs2 foi replicado corretamente.

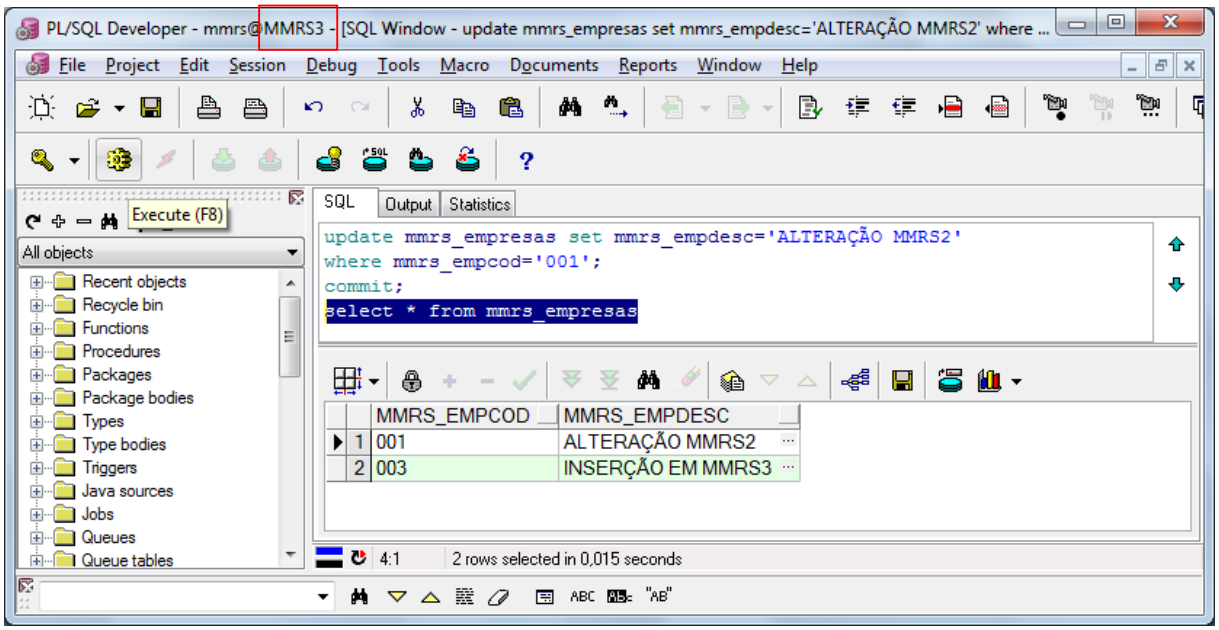


Figura 49 - Consulta a replicação em mmrs3

De acordo com os testes realizados nesta fase, o replicador funcionou corretamente o que indica que todo o processo de configuração foi feito em conformidade com o manual da ferramenta.

4.8. DESEMPENHO DO REPLICADOR

No processo de validação do replicador foram realizados diversos testes com as operações DML (update, delete e insert), com uma amostragem de 807 registros sendo replicados para um dos sites alternativos, devido ao limite de operações de replicação da ferramenta estar limitada a 1000 em sua versão Trial. Todos os processos funcionaram adequadamente, conforme previsto pela ferramenta.

Antes do início do processo de carga no servidor da sede, foi iniciado no Linux o processo responsável pela replicação (/home/object/objectmmrs/run.sh) e em seguida iniciado o processo de carga das tabelas, conforme demonstrado nas figuras a seguir:

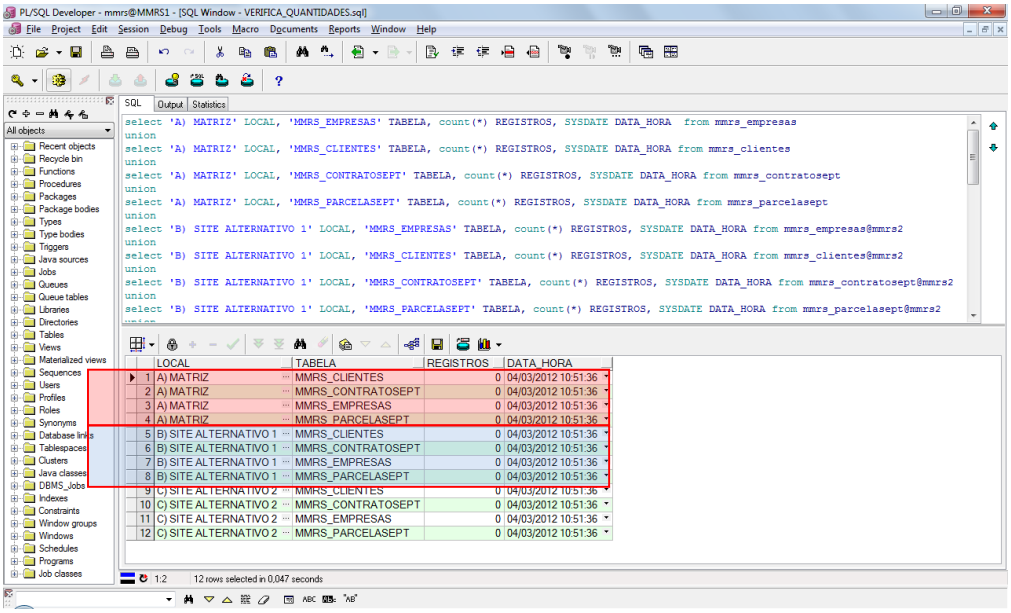


Figura 50 - Tabelas a serem replicadas vazias as 10:51:36

Na figura acima são demonstradas as tabelas de todos os sites (sede - mmrs1, site alternativo 1- mmrs2 e site alternativo 2 - mmrs3).

Devido a limitações da ferramenta Trial, será demonstrada a replicação de 807 registros para apenas um dos sites alternativos.

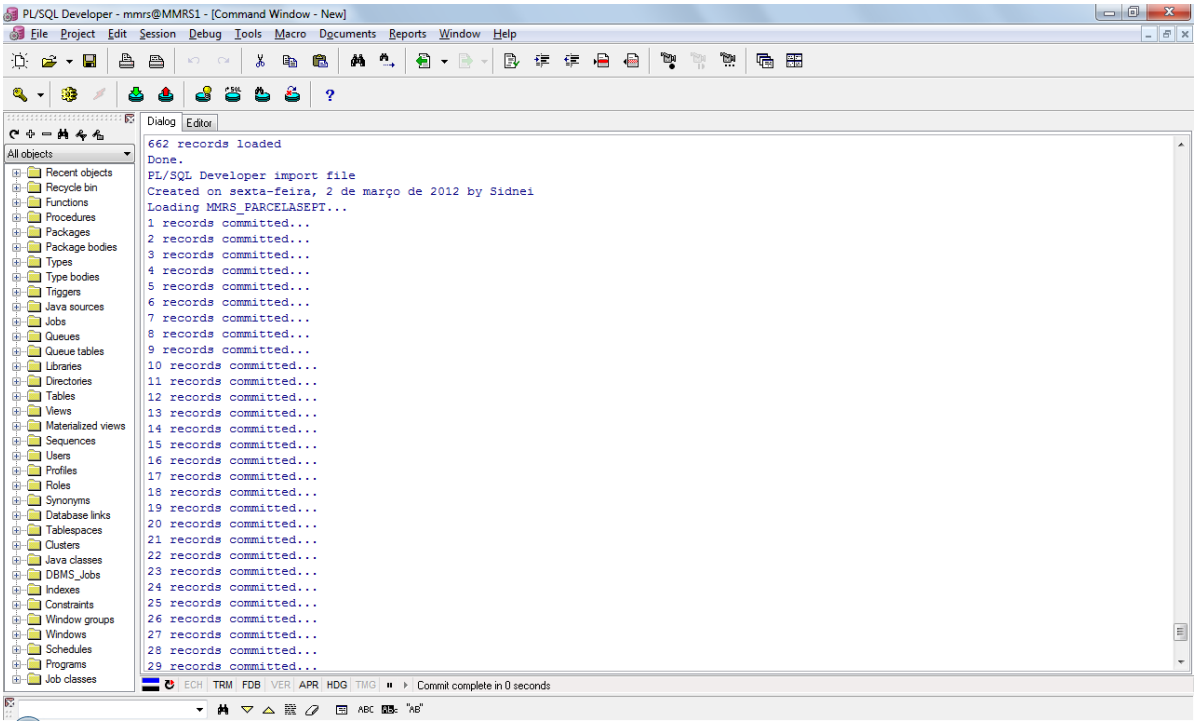


Figura 51 - Realização da Carga no site mmrs1

Na figura acima está sendo demonstrado o andamento do processo de carga no servidor da sede e logo adiante a são apresentadas as tabelas da sede

devidamente carregadas e do site alternativo com seus dados já recebidos da sede através do processo de replicação.

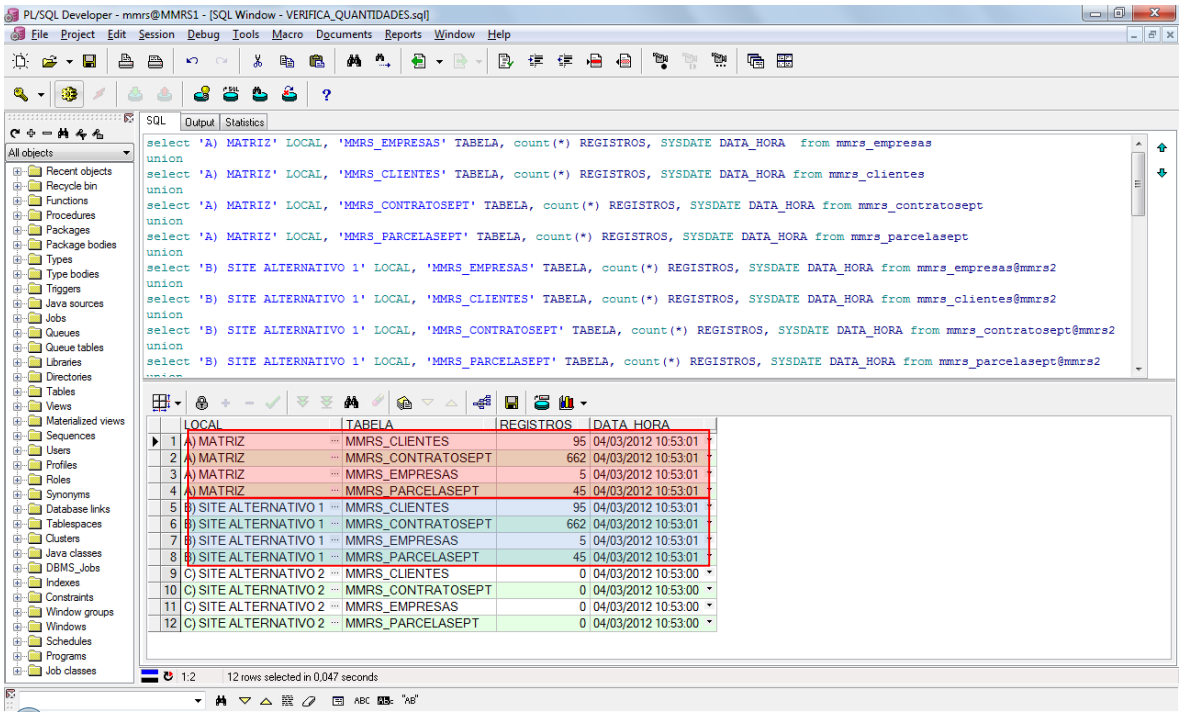


Figura 52 - Tabelas da sede e do o site alternativo 1 às 10:53:01

Além dos testes de funcionamento da ferramenta, foi verificado o desempenho do processo de replicação, utilizando o aplicativo IPTraf para medir a quantidade de Bytes trafegados no servidor slave no momento da replicação.

Durante o intervalo de 1min25seg (Tempo decorrido até 100% da replicação), trafegou pelo servidor mmrs2 o total de 3.351.950 Bytes, conforme figura a seguir.

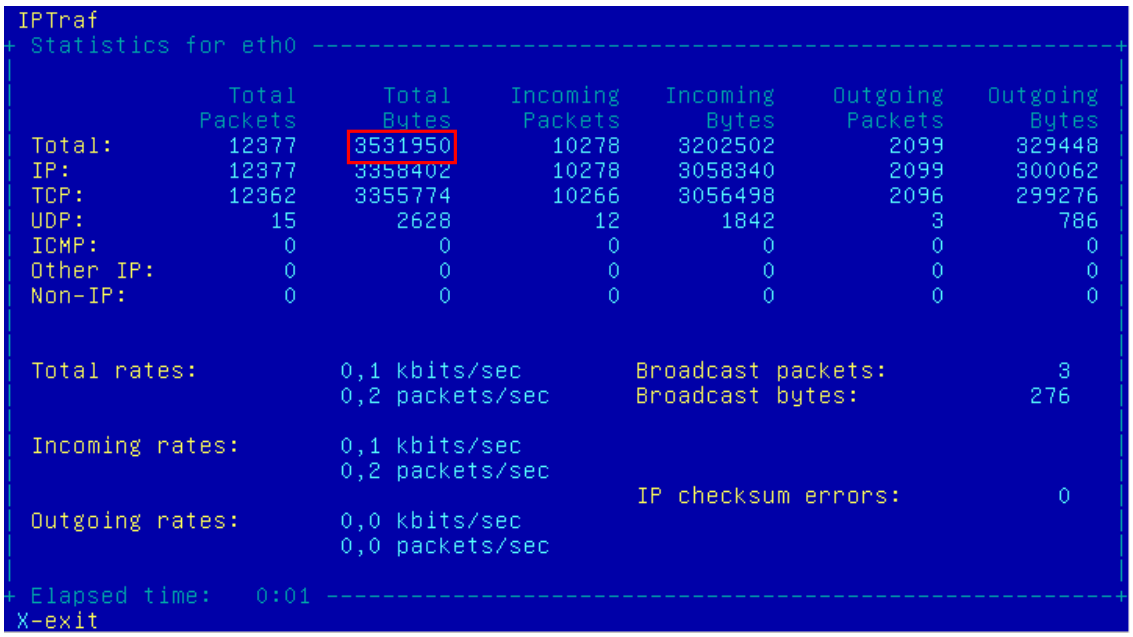


Figura 53 - Bytes trafegados na replicação para o Servidor Alternativo 1 (mmrs2)

Com base nas informações coletadas foi possível elaborar uma planilha para o cálculo do desempenho da ferramenta em kbps e Mbps de acordo com o volume de dados.

Cálculo Estatístico com Base e Dados Coletados		
Qtde. registros da tabela mmrs_clientes	95	
Qtde. registros da tabela mmrs_contratosept	662	
Qtde. registros da tabela mmrs_empresas	5	
Qtde. registros da tabela mmrs_parcelasept	45	
Total de Registros:	**807**	
Hora de início da replicação	10:51:36	
Hora em que a replicação foi concluída	10:53:01	
Tempo total gasto no processo de replicação	**00:01:25**	==> Equivale a 85 segundos
Bytes Trafegados no período da replicação:	3.531.950	==> Equivale a 3,37 Mbytes
Bytes Trafegados (convertidos em Kbits):	27.593,36	Fórmula: (Dados em Bytes/1024*8)
Velocidade Média de replicação (kbps):	**324,63**	Fórmula: (Bytes Trafegados (convertidos em Kbits)/Tempo gasto em segundos)
Velocidade Média de replicação (Mbps):	**0,32**	Fórmula: (Velocidade em Kbps / 1024)

Figura 54 **- Apuração do desempenho do replicador**

Conforme quadro acima, a velocidade de replicação, para 807 registros (3,37 MB) foi de 324,63 Kbps, o que significa que esta amostragem passaria com folga pelo canal de fibra óptica de 2 Mbps definido no item 4.1 do estudo de caso deste projeto.

4.9. GRÁFICO DE INVESTIMENTO

O gráfico a seguir demonstra que a replicação sem o uso de software específico para este fim pode gerar um custo elevado para a empresa, além da probabilidade de erros e o surgimento da necessidade de correções manuais, as quais muitas das vezes poderiam ser evitadas através do uso de um software replicador.

Os dados analisados para geração deste gráfico são:

- O custo do link de 10Mbps (R$ 6.000,00), necessário para a Replicação Rústica (sem software), acrescido de 10% de correção a cada 12 meses; e,
- O custo do link de 2Mbps (R$ 1.700,00), necessário para a Replicação via software, acrescido de 10% de correção a cada 12

meses, somado a aquisição de licença do software replicador e respectiva renovação anual.

O custo da solução sem o replicador, no período de 5 (cinco) anos seria por aproximadamente R$ 439.000,00, enquanto com a utilização do software replicador Goldengate ficaria aproximadamente R$ 161.000,00 e com o Objectmmrs R$ 130.000,00.

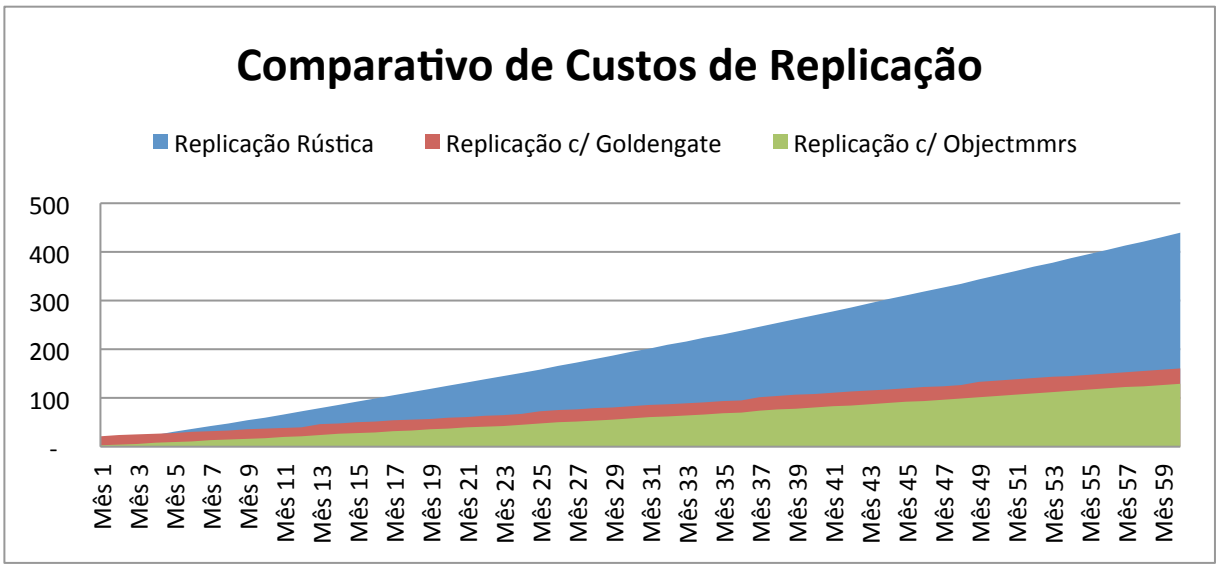


Figura 55 - Gráfico de Custos da Replicação

A solução tem um custo elevado, mas levando em consideração que a replicação gera segurança e alta disponibilidade para os dados, tal custo é perfeitamente justificável, pois estará preservando dados que são altamente importantes para a continuidade do negócio da empresa, pois caso haja uma catástrofe e a empresa perder todos os dados, esta ficará impossibilitada de pagar salários dos funcionários, aposentadorias, pensões e outros tipos de benefícios que fazem parte do negócio.

Consequentemente a empresa pode falir, ter problemas de imagem perante o público e mídia, pode receber multas de órgãos reguladores devido à impossibilidade de prestar informações dentro dos prazos legais, além de expor o corpo Funcional e Diretoria a sanções judiciais.

4.10. FUNCIONAMENTO E MANUTENÇÃO DO REPLICADOR

O serviço do objectmmrs pode ser customizado para ser inicializado junto com o sistema operacional ou de forma manual. Após inicializado ele permanece em

constante funcionamento, sempre verificando em determinado intervalo de tempo (customizável) se existe alguma replicação a ser realizada.

Este serviço só será parado por requisição do operador do servidor ou por alguma outra anormalidade no sistema (falta de memória, falta de espaço em disco, tablespace estourada, falha de hardware, etc). No entanto estas paradas anormais não implica na perda da replicação, pois todos os processo que vão acontecendo no servidor de banco de dados vão sendo registrados na tabela de log do objectmmrs e quando o serviço é restabelecido todos os dados que estão na tabela de log com o status de pendente de replicação, são automaticamente replicados.

Antes do servidor master enviar algum registro ao slave, ele analisa se o destino estão operando normalmente e, se não estiver, os registros são armazenados em tabelas temporárias até a solução do problema. Após o restabelecimento do serviço o replicador sincroniza os dados que ficaram pendentes na fila.

Não existe nenhuma rotina manutenção a ser seguida para o uso do replicador, apenas é recomendável que os servidores estejam com seus horários sincronizados e que o DBA sempre se esteja atento aos arquivos de log a fim de identificar e tratar eventuais erros que possam vir a ocorrer.

5. ANÁLISE

Neste capítulo, são descritas algumas observações a respeito deste trabalho de conclusão de curso, bem como os resultados obtidos.

5.1. CONSIDERAÇÕES GERAIS

A implementação da tecnologia de BDD é uma inovação interessante para empresas de todos os portes, pois através dela é possível tornar consultas a banco de dados mais ágeis, deixar os mesmos dados disponíveis em vários sites ao mesmo tempo, ou em intervalos pré-definidos, dando assim, segurança à guarda de informações e alta disponibilidade no caso de instabilidade ou pane em algum servidor que faz parte do conjunto de distribuição.

Existem diversas formas de se implementar um modelo de BDD, cabendo às empresas interessadas em desenvolver esta tecnologia a avaliação da melhor solução, sempre levando em consideração o custo/benefício, sem deixar de lado os quesitos relacionados a segurança, disponibilidade, confidencialidade, integridade, transparência.

Apesar da complexidade adicional para administração da estrutura e dos dados, o uso desta tecnologia vale a pena.

5.2. RESULTADOS OBTIDOS

Ficou evidente que no decorrer do trabalho, que o processo de replicação de dados em um SBDD depende da preparação de toda uma estrutura, para que a solução funcione de forma satisfatória. Pôde-se observar que a estrutura de replicação montada funciona bem, conforme o planejamento inicial. Os sites ficam com acesso independente aos dados, o que possibilita o funcionamento da empresa mesmo em situações em que haja problema de ligação da rede da empresa à rede de outra, fazendo com que não haja prejuízos decorrentes de eventuais falhas. Isso é benéfico para a empresa, pois no cenário anterior, quando se tinha um Banco de Dados centralizado, a situação era bem diferente, ou seja, caso um problema ocorresse na rede que impossibilitasse o acesso ao Banco de Dados significaria a impossibilidade da empresa poder operar. Do ponto de vista estratégico das empresas, os benefícios são vistos imediatamente após a implementação, o que justifica e valida todo o trabalho e investimento em um projeto deste tipo.

6. CONCLUSÕES

6.1. QUANTO AO PROBLEMA E A SOLUÇÃO

Os problemas inicialmente levantados foram a necessidade de se manter o sistema corporativo de uma empresa sempre operante, independente de algum servidor de banco de dados estar com alguma tipo de pane, reduzir o troughput de rede quando da necessidade da realização de consultas e atualizações de dados e aumentar o nível de segurança dos dados, mantendo-os em mais de um servidor de BD.

Na fase de desenvolvimento deste trabalho, foi possível perceber que é completamente possível a implementação de um BDD, no entanto, cabe ao gestor de TI avaliar qual a melhor solução, sempre levando em consideração a estratégia da empresa.

Apesar de ser uma solução de custo elevado, um BDD proporciona vantagens em relação ao centralizado, como por exemplo a redução de troughput no canal de rede e a replicação de dados em mais e um site, proporcionando assim mais segurança e disponibilidade das informações.

Como citado, esta não é uma solução simples de ser implementada, precisando de bastante conhecimento técnico e o uso de ferramentas adequadas para o perfeito funcionamento da política de replicação, principalmente quando se tratar de bancos de dados muito grandes e complexos.

7. SUGESTÃO DE TRABALHOS FUTUROS

A seguir são apontados alguns itens que podem servir de complementação de estudo a cerca do assunto abordado neste trabalho:

- Replicação de Dados em um SBDD heterogêneo;
- Sincronização de hora com base em servidores atômicos;
- Replicação de Dados e suas respectivas estruturas utilizando Archivelogs.

8. REFERÊNCIAS BIBLIOGRÁFICAS

ALVARES, R. V. **Modelagem**. Disponível em http://www.sqlmagazine.com.br/Colunistas/Reinaldo/06_Modelagem_P3.asp Acesso em 25 de out. 2010.

ANDRADE, M. M. **Introdução à metodologia de trabalho científico: elaboração de trabalhos na graduação**. 8. Ed. São Paulo: Atlas, 2007.

BONOMA, T. V. **Case research in marketing: opportunities, problems and a process**. *Journal of Marketing Reserarck*, Chicago, v. XXII, may 1985.

CERVO, A. L.; BERVIAN, P. A. **Metodologia científica**. 4. ed. São Paulo: Makron Books, 1996.

DATE, C. J. **Introdução a sistemas de banco de dados**, tradução (da 4 ed. original) de Contexto traduções, Rio de Janeiro. Campus, 1991.

ELMASRI, R. and NAVATHE, S. B. **Sistemas de Banco de Dados**, 4 ed., São Paulo. Addison-Wesley, 2005.

KORTH, H. F., **Sistema de Banco de Dados**, tradução [da 2 ed. ver] Maurício Heihachiro Galvan Abe. São Paulo. Makron Books, 1995.

ÖZSU, M. T.; VALDURIEZ, P.. **Principles of Distributed Database Systems**. tradução [da 2 ed. americana] Vandenberg D. de Souza. Rio de Janeiro. Campus, 2001.

MACHADO, F.; ABREU, M.. **Projeto de Banco de Dados – Uma Visão Prática**. 16ª edição, São Paulo. Érica, 2010.

OLIVEIRA, Alexandre Pereira de. **Modelo de Replicação de Dados entre SGBD Heterogêneos**. Monografia (Graduação em Ciências da Computação). 2006. Gravataí. Universidade Luterana do Brasil.

SETZER, V. V., SILVA, F. S. C. **Bancos de Dados**. 1 ed., São Paulo. Edgard Blucher, 2005.

SILBERSCHATZ, A., KORTH, H. F., SUDARSHAN, S. **Sistema de Banco de Dados**. 5 ed., Rio de Janeiro. Campus, 2006.

SMANIOTO, C. A. **Replicação e Alta Disponibilidade no PostgreSQL**. Disponível em http://www.devmedia.com.br/artigo-da-sql-magazine-24-replicaca-e-alta-disponibilidade-no-postegresql/6140
Acesso em 08 de mar. 2012.

SISTEMAS, O. **Conceitos Gerais de Replicação de Banco de Dados**. Disponível em http://www.object.com.br/wiki/ConceitosGerais
Acesso em 08 de mar. 2012.

WATSON, John. **OCA Oracle Database 11g Administração I: Guia do Exame 1Z0-052**. Porto Alegre: Brookman, 2010.

9. GLOSSÁRIO

API (*Application Programming Interface*) **–** É um conjunto de rotinas e padrões estabelecidos por um programa de computador para a utilização de suas funcionalidades pro programas aplicativos que não querem se envolver nos detalhes da implementação do software, mas apenas usar seus serviços.

Background – Em se tratando de processos, refere-se aos processos que são processados em computador ou servidor e que o usuário não consegue enxergar sem a execução de comandos específicos de listagem de processos ativos.

Buffer – É a área de memória reservada para armazenar dados que estão sendo utilizados em tempo de processamento.

Business Intelligence – É o nome dado ao processo de coleta, organização, análise, compartilhamento e monitoramento de informações que oferecem suporte a tomada de decisão.

Cache – é o armazenamento temporário de dados para tornar mais veloz as consultas aos mesmos tipos de dados já consultados anteriormente.

Clock – Controlador de tempo para execução de atividades por um computador.

Commit – Processo utilizado para persistir alterações realizadas nos dados de uma ou várias tabelas de bancos de dados.

Constraints – Regras restritivas.

Control files – São arquivos binários utilizados para rastrear o status do banco de dados e a estrutura física.

CPU – O mesmo que processador. É a parte física de um sistema de computador responsável pela execução as instruções de um programa de computador.

Data Warehouse – É um sistema de computação utilizado para armazenar informações relativas às atividades de uma organização em bancos de dados, de forma consolidada.

Database – Na arquitetura de banco de dados refere-se à área onde ficam armazenados os *data files*, *control files* e os *online redo log files, archivelog e etc.*

Data files – São os arquivos onde ficam armazenados em disco os dados do banco de dados.

DBLink – É um canal utilizado pelo Oracle para fazer com que, através de uma rede de computadores, um banco de dados localizado em determinado site possa acessar os dados existentes em outro site.

Delete – comando SQL utilizado para excluir dados em uma ou mais tabelas.

Driver – É um programa de computador que permite ao sistema operacional utilizar as funcionalidades de um dispositivo ou programa específico.

Eager – A tradução desta palavra é “ansioso”. Voltado para replicação de BDD, refere-se ao modo que a replicação é realizada, que é direta, sem buscar dados nos logs.

Full – cheio, completo.

Grant – Conceder privilégio.

Hardware - é a parte física (tangível) de um computador e caracteriza-se pelo conjunto de componentes eletrônicos, circuitos integrados e placas.

Insert – comando SQL utilizado para inserir dados em uma ou mais tabelas.

Instance – Na arquitetura de banco de dados refere-se a área onde ficam os processos rodando em background e memória.

Lazy – A tradução desta palavra significa “preguiçoso”. Voltado para replicação de BDD, refere-se ao modo que a replicação é realizada, que é através de logs dentro de determinado intervalo de tempo conforme parametrizado no replicador.

Link – é o elo que une através de um meio específico (fibra-óptica, rádio, etc) o usuário da informação à própria informação.

Log – registro detalhado de eventos que ocorreram no banco de dados.

Master – Em se tratando de replicação, esta é a denominação do servidor que envia dados ao servidor slave.

Nodos – Vide o termo “Site” neste glossário.

Online redo log file – Arquivos de segurança que permitem que uma instância de banco de dados se recupere em caso de falha.

OLTP – (Online Transaction Processing) - são sistemas que se encarregam de registrar todas as transações contidas em uma determinada operação organizacional.

Overhead – é qualquer processamento ou armazenamento em excesso, seja de tempo de computação, memória, largura de banda ou qualquer outro recurso requerido ou gasto para executar uma determinada atividade.

Select – comando SQL utilizado para selecionar dados em uma ou mais tabelas.

Site – No contexto deste trabalho, site significa uma instalação (localidade) que possua um servidor com um sistema de banco de dados distribuído nele instalado.

Slave – Em se tratando de replicação de banco de dados distribuído, é a denominação do servidor que receberá dados de um servidor master.

Suíte – conjunto de *softwares* aplicativos.

BDD – Banco de Dados Distribuído.

Primary Key – Restrição de integridade que define o atributo que deverá ser única em uma tabela.

Revoke – Revogar, retirar o privilégio dado através do comando *grant*.

Rollback – Nome dado ao processo que volta os dados ao estado original, caso haja algum problema na transação.

SBDD – Sistema de Banco de Dados Distribuído

SGBD – Trata-se de um o conjunto de programas de computador responsáveis pelo gerenciamento de uma base de dados, que possui como objetivo principal retirar do cliente a responsabilidade de gerência de acesso, manipulação e organização de dados. O SGBD disponibiliza uma *Application Programming Interface* (API) ou *Driver,* através da qual os clientes podem incluir, alterar e consultar dados através de comandos SQL.

SGBDD - É o software responsável por controlar todas as ações efetuadas em um Banco de Dados Sistema de Gerenciamento de Bancos de Dados Distribuídos.

Software – Programa de computador criado para ser instalado em um hardware para desempenhar atividades específicas.

Tablespace – Sub-divisão lógica de um banco de dados utilizado para agrupar estruturas lógicas relacionadas.

Timestamp – É o registro do tempo ou momento em que determinado processo é executado.

Troughput - é a quantidade de dados transferidos de um lugar a outro, ou a quantidade de dados processados em um determinado espaço de tempo, pode-se usar o termo throughput para referir-se a quantidade de dados transferidos em discos rígidos ou em uma rede.

Unique Key – Restrição de integridade que define que a chave do registro não poderá se repetir na tabela. Toda Primary Key é Unique Key.

Update – comando SQL utilizado para alterar dados em uma ou mais tabelas.

View – É a visão parcial ou total de tabelas de banco de dados.

10. ANEXOS

DVD com o seguinte conteúdo:

- Este trabalho de conclusão de curso;
- Scripts de preparação da base de dados do replicador;
- Scripts de criação das tabelas utilizadas no estudo de caso;
- Scripts de carga em tabelas utilizadas no estudo de caso;
- Replicador ObjectMMRS na versão trial (configurada para mmrs1);
- Script de criação da Database Link;
- Script de contagem de registros das tabelas do estudo de caso; e,
- Vídeo apresentação do funcionamento do replicador.