

FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA - UNIVEM
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

JOÃO PAULO COLTRI MARRI

**ORACLE E FIREBIRD UMA ANÁLISE DE DESEMPENHO A PARTIR
DE IMPLEMENTAÇÃO DE FUNÇÕES**

MARÍLIA
2006

JOÃO PAULO COLTRI MARRI

**ORACLE E FIREBIRD UMA ANÁLISE DE DESEMPENHO A PARTIR
DE IMPLEMENTAÇÃO DE FUNÇÕES**

Monografia apresentada ao Curso de Ciência da Computação da Fundação de Ensino Eurípides Soares da Rocha, Mantenedora do Centro Universitário Eurípides de Marília - UNIVEM, para obtenção do grau de Bacharel em Ciência da Computação.

Orientador:

Prof^a. Dra. Fátima de L. S. Nunes Marques

MARÍLIA
2006

JOÃO PAULO COLTRI MARRI

ORACLE E FIREBIRD UMA ANÁLISE DE DESEMPENHO A PARTIR DE
IMPLEMENTAÇÃO DE FUNÇÕES

BANCA EXAMINADORA DA MONOGRAFIA PARA OBTENÇÃO DO GRAU DE
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO

CONCEITO FINAL: _____(_____)

ORIENTADOR: _____
Prof^a. Dra. Fátima L. S. Nunes Marques

1º EXAMINADOR: _____
Prof. Dr.

2º EXAMINADOR: _____
Prof. Dr.

Marília, _____ de _____ de 2006.

AGRADECIMENTOS

A Deus por me dar forças para alcançar meus objetivos e metas nesse trabalho e em toda minha vida.

A todos os amigos por sempre estarem do meu lado, nos momentos mais difíceis.

A Prof^a. Fátima L. S. Nunes Marques pelo apoio, ajuda, compreensão e conhecimentos passados durante todo esse trabalho.

A todos os professores do curso de Ciência da Computação do Univem pelo conhecimento transmitido durante todo curso.

Aos companheiros e amigos da Faculdade por sempre estarem juntos nas incansáveis horas de estudos.

MARRI, João Paulo Coltri. Oracle e Firebird uma de análise desempenho a partir de implementação de funções. 2006. 67 f.
Monografia (Bacharelado em Ciência da Computação) – Curso de Ciência da Computação, Fundação de Ensino “Eurípides Soares da Rocha”, mantenedora do Centro Universitário Eurípides de Marília – UNIVEM.

Resumo

O Oracle é um SGDB amplamente conhecido no mercado. Oferece inúmeras funções que possibilitam o aumento da produtividade e desempenho no desenvolvimento de sistemas. O Firebird é outro SGDB, relativamente novo no mercado, gratuito e que tem uma quantidade limitada de funções implementadas.

Esse trabalho propõe a implementação de determinadas funções de manipulação de data presentes no Oracle para o Firebird, com o objetivo de avaliar os desempenhos dessas novas funções. Para fazer as comparações foram realizados testes nos dois sistemas de banco de dados com uma quantidade variável de execuções para uma melhor avaliação dos resultados.

Os resultados mostram todas as comparações dos testes realizados, concluindo que o SGDB Oracle obteve um desempenho bem superior em relação ao SGDB Firebird.

Palavras-chave: Banco de dados, *Stored Procedures*, implementação, testes de desempenho, funções.

MARRI, João Paulo Coltri. Oracle e Firebird uma análise de desempenho a partir de implementação de funções. 2006. 67 f.

Monografia (Bacharelado em Ciência da Computação) – Curso de Ciência da Computação, Fundação de Ensino “Eurípides Soares da Rocha”, mantenedora do Centro Universitário Eurípides de Marília – UNIVEM.

Abstract

The Oracle is an widely known SGDB in the market. It offers innumerable functions that make possible the increase of the productivity and performance in the development of systems. The Firebird is another SGDB, relatively new in the market, gratuitous and that it has an limited amount of implemented functions. This work considers the implementation of definitive functions of manipulation of date gifts in the Oracle for the Firebird, with the objective to evaluate the performances of these new functions. To make the comparisons tests in the two systems of data base with a changeable amount of executions for one better evaluation of the results had been carried through. The results show to all the comparisons of the carried through tests, concluding that the SGDB Oracle got a well superior performance in relation to the SGDB Firebird.

Keywords: Data Base, *Stored Procedures*, implementation, performance tests, functions.

LISTA ABREVIACÕES

BD - Banco de Dados

EMBRAPA – Empresa Brasileira de Pesquisa Agropecuária

ESALQ – Escola Superior de Agronomia Luis de Queiroz

GHz - *Giga Hertz*

GB - *Giga Byte*

HD - *Hard Disk*

HTTP - *Hyper Text Transfer Protocol*

MB - *Mega Byte*

JVM - *Java Virtual Machine*

PSQL - *Procedural SQL*

RAM – *Random Access Memory*

SID – Nome da instância do Banco de Dados Oracle

SQL - *Structured Query Language*

SP - *Stored Procedure*

SGDB - Sistema Gerenciador de Banco de Dados

UDF - *User Defined Function*

UTC - *Coordinated Universal Time*

USP – Universidade de São Paulo

SUMÁRIO

Resumo	5
1. Introdução.....	11
1.1. Objetivos.....	12
1.2. Justificativa.....	12
1.3 Disposição do Trabalho	13
2. SGBDs Oracle e Firebird.....	14
2.1. O SGBD Oracle e suas funções de manipulação de data	14
2.2. O SGDB Firebird.....	16
2.2.1. <i>Stored Procedures</i>	17
3. Metodologia.....	20
3.1. Tecnologias utilizadas	20
3.1.1. Instalação e configuração dos SGDBs.....	23
3.1.2. Linguagem de implementação.....	23
3.2. Funções implementadas no Firebird.....	26
3.3. Definição e descrição dos algoritmos das funções	26
3.4. Forma de execuções das funções.....	33
4. Resultados e Discussões	35
4.1. Descrição dos resultados no Oracle.....	35
4.2. Descrição dos resultados no Firebird	36
4.3. Análise de desempenho por função	37
4.3.1. Função ADD_MONTHS	38
4.3.2. Função LAST_DAY.....	40
4.3.3. Função MONTHS_BETWEEN	41
4.3.4. Função NEXT_DAY	43
4.3.5. Função TO_CHAR.....	44
4.3.6. Função TO_DATE	45
4.3.7. Função TO_TIMESTAMP	46
4.3.8. Função ROUND	48
4.3.9. Função TRUNC.....	49
4.3.10. Análise geral.....	50
5. Conclusões.....	52
Referências Bibliográficas.....	54
Apêndices	55
A. Instalação e configuração do Firebird 1.5	55
B. Instalação e configuração do Oracle 9i.....	61

LISTA DE FIGURAS

Figura 2.1 – Estrutura básica de Stored Procedure.....	18
Figura 2.2 – Tela de exemplo de chamada de SP	19
Figura 3.1 - Tela de conexão do SQL*Plus.....	21
Figura 3.2 - Tela do <i>prompt</i> da SQL*Plus	21
Figura 3.3 - Tela da ferramenta IBEpert.....	22
Figura 3.4 - Definição do algoritmo da função auxiliar DIV	27
Figura 3.5 - Definição do algoritmo da função auxiliar CONSULTA_DIA.....	27
Figura 3.6 – Definição do algoritmo da função ADD_MONTHS	28
Figura 3.7 - Definição do algoritmo da função LAST_DAY.....	28
Figura 3.8 - Definição do algoritmo da função MONTHS_BETWEEN	28
Figura 3.9 - Definição do algoritmo da função NEXT_DAY	29
Figura 3.10 - Definição do algoritmo da função TO_DATE	29
Figura 3.11 - Definição do algoritmo da função TO_TIMESTAMP.....	30
Figura 3.12 – Definição do algoritmo da função TO_CHAR	31
Figura 3.13 – Definição do algoritmo da função ROUND.....	32
Figura 3.14 – Definição do algoritmo da função TRUNC	33
Figura 4.1 – Fórmulas para os cálculos das médias	38
Figura 4.2 – Gráfico com os resultados dos testes da função ADD_MONTHS	39
Figura 4.3 - Gráfico com os resultados dos testes da função LAST_DAY.....	40
Figura 4.4 - Gráfico com os resultados dos testes da função MONTHS_BETWEEN	42
Figura 4.5 - Gráfico com os resultados dos testes da função NEXT_DAY	43
Figura 4.6 - Gráfico com os resultados dos testes da função TO_CHAR.....	44
Figura 4.7 - Gráfico com os resultados dos testes da função TO_DATE	45
Figura 4.8 - Gráfico com os resultados dos testes da função TO_TIMESTAMP.....	47
Figura 4.9 - Gráfico com os resultados dos testes da função ROUND	48
Figura 4.10 - Gráfico com os resultados dos testes da função TRUNC.....	49
Figura A.1 - Tela Inicial de instalação do Firebird.....	55
Figura A.2 - Tela de escolha do servidor Firebird.....	56
Figura A.3 - Tela de configuração da inicialização do Servidor	57
Figura A.4 - Tela de informação sobre a instalação do Firebird	58
Figura A.5 - Tela do IBEpert para registrar uma base de dados	59
Figura A.6 - Tela de exemplo de conexão com uma base a dados usando o isql.....	60
Figura B.1 - Tela Inicial da instalação do Oracle 9i.....	61
Figura B.2 - Tela para verificação dos produtos Oracle.....	62
Figura B.3 - Tela de definição do Oracle Home.....	62
Figura B.4 - Tela para escolha do tipo de produto	63
Figura B.5 - Tela para escolha do tipo da instalação.....	64
Figura B.6 - Tela de configuração do Banco de dados.....	64
Figura B.7 - Tela para nomeação do banco de dados e o SID.....	65
Figura B.8 - Tela para informar caminho do banco de dados	65
Figura B.9 - Tela para escolher o conjunto de caracteres do banco de dados	66
Figura B.10 - Tela de resumo da instalação do Oracle.....	66
Figura B.11 - Tela de configuração do Oracle	67
Figura B.12 - Tela final da instalação do Oracle.....	67

LISTA DE TABELAS

Tabela 2.1 – Sintaxe e descrição das funções de data do Oracle.....	14
Tabela 2.2 – Sintaxe e a descrição das funções de manipulação de data do Firebird	17
Tabela 3.1 – Formas de execução das funções no Firebird	34
Tabela 4.1 – Exemplos e os resultados das funções executadas no Oracle.....	35
Tabela 4.2 - Exemplos e os resultados das funções executadas no Firebird	36
Tabela 4.3 – Média dos testes com a função ADD_MONTHS.....	39
Tabela 4.4 - Média dos testes com a função LAST_DAY	41
Tabela 4.5 - Média dos testes com a função MONTHS_BETWEEN.....	42
Tabela 4.6 - Média dos testes com a função NEXT_DAY	43
Tabela 4.7 - Média dos testes com a função TO_CHAR	44
Tabela 4.8 - Média dos testes com a função TO_DATE.....	46
Tabela 4.9 - Média dos testes com a função TO_TIMESTAMP	47
Tabela 4.10 - Média dos testes com a função ROUND.....	48
Tabela 4.11 - Média dos testes com a função TRUNC	49
Tabela 4.12 – Média geral das funções nos SGDBs Oracle e Firebird	50

1. Introdução

Um Banco de Dados (BD) é uma coleção de dados armazenados (banco físico) que pode ser acessado e manipulado através de um sistema gerenciador ou um aplicativo (DATE, 2004).

A definição de banco de dados segundo a Wikipedia (2006): “bancos de dados são conjuntos de dados com uma estrutura regular que organizam informações. Um banco de dados é usualmente mantido e acessado por meio de um software conhecido como Sistema Gerenciador de Banco de Dados (SGBD)”.

O sistema gerenciador de banco de dados (SGDB) é o software que esta entre o banco de dados físico e os usuários. Korth e Silberchatz (2003) definem SGDB como sendo um conjunto de dados relacionados entre si e uma coleção de programas para acessá-los tendo como principal objetivo possuir um meio eficiente para recuperar e armazenar informações.

Um SGDB também deve ser capaz de fazer a proteção, contra falhas (hardware ou software) e a manutenção do banco de dados por um longo período de tempo (ELMASRI e NAVATHE, 2005).

Há diversas vantagens na utilização de um SGDB, como o controle de redundância, acesso restrito, armazenamento persistente, processamento eficiente de consultas, processo de backup e restauração, restrições de integridade, redução no desenvolvimento de aplicações, flexibilidade, disponibilidade para atualizar as informações, entre outros (ELMASRI e NAVATHE, 2005).

As funções implementadas em SGDB trazem determinadas vantagens e desvantagens. Vantagens: redução do tráfego na rede, produtividade, facilidade de manutenção, desempenho e segurança. Desvantagens: limitação da linguagem utilizada e portabilidade (TODD, 2000).

1.1. Objetivos

Este trabalho tem como objetivo implementar algumas funções de manipulação de data do Oracle no banco de dados Firebird e apresentar um comparação de desempenho dessas funções nos dois SGBDs. O escopo foi delimitado a essas funções devido ao fato de que geralmente estas não estão disponíveis em SGDBs gratuitos. Também foi visado nesse trabalho fazer testes para comparar o tempo de execução das funções entre o Oracle e o Firebird, isto é, verificar em qual SGDB apresenta melhor desempenho em relação às funções testadas.

O SGDB Oracle foi escolhido para esse trabalho devido as seguintes características: inúmeros recursos, robusto e por ser um dos mais utilizados do mercado. O outro SGDB escolhido foi o Firebird devido as seguintes características: gratuito, relativamente novo e com poucos recursos.

1.2. Justificativa

Assim torna-se possível utilizar alguns dos recursos do Oracle em um SGDB gratuito e Open Source, no caso o Firebird, utilizado por milhares de desenvolvedores em todo mundo.

Algumas funções de manipulação de data do Oracle ajudarão muito os desenvolvedores, diminuindo a necessidade de implementar novos códigos. Em SGDBs gratuitos, em geral, há poucas funções deste tipo. Se essas já estiverem definidas no banco de dados, podem reduzir o tempo e, conseqüentemente, os custos no desenvolvimento de uma aplicação.

1.3 Disposição do Trabalho

No Capítulo 1 são apresentados a justificativa, os objetivos e alguns conceitos sobre o banco de dados e um SGBD.

No Capítulo 2 são descritos os SGDBs Oracle e Firebird com algumas de suas características e suas respectivas funções de manipulação de data.

No Capítulo 3 é apresentada a metodologia utilizada para a realização desse trabalho, descrevendo todas as tecnologias que estão envolvidas e também a definição dos algoritmos das funções a serem implementadas.

No Capítulo 4 são apresentados os resultados e uma análise de desempenho, comparando-se os dois SGDBs.

O Capítulo 5 descreve as conclusões obtidas por meio dos testes realizados no Capítulo 4.

2. SGBDs Oracle e Firebird

2.1. O SGBD Oracle e suas funções de manipulação de data

O Oracle é um SGDB considerado um dos mais robustos, flexíveis e mais utilizados do mercado.

Esse SGDB possui diversos recursos, entre eles uma linguagem de programação chamada de PL/SQL que mistura o poder e a flexibilidade do SQL (*Structured Query Language*) com os comandos e estruturas de uma linguagem de programação comum (SCOTT, 2002).

No Oracle existem três formas para o armazenamento de data: DATE, TIMESTAMP e INTERVAL. O tipo DATE armazena a data e hora (inclusive século, ano, mês, dia, hora, minuto e segundo). O tipo TIMESTAMP é semelhante ao DATE, mas pode armazenar segundos fracionários. Já o tipo INTERVAL serve para armazenar a diferença entre dois timestamps. Internamente esses tipos de dados são armazenados como números e ocupam 7 bytes (SCOTT, 2002).

O Oracle oferece um conjunto grande de funções para manipulação de data, essas funções recebem argumentos do tipo DATE e a maioria delas também retorna valores DATE. As funções são apresentadas na Tabela 2.1.

Tabela 2.1 – Sintaxe e descrição das funções de data do Oracle

Função	Sintaxe	Descrição
ADD_MONTHS	ADD_MONTHS(D, N) , onde D é a data de entrada e N é o numero de meses que será acrescido em D.	Retorna a data D Acrescida de N meses. Se N for um a data D menos N meses.
CURRENT_DATE	CURRENT_DATE	Retorna a data atual do fuso horário da sessão com valor DATE.
CURRENT_TIMESTAMP	CURRENT_TIMESTAMP [precisão] , onde [precisão] é	Retorna a data atual do fuso horário da sessão com valor

	um numero que se for especificado indica a precisão em frações de segundos.	TIMESTAMP WITH TIMEZONE.
DBTIMEZONE	DBTIMEZONE	Retorna o fuso horário do banco de dados.
EXTRACT	EXTRACT (Opção From D) , onde Opção define o que será extraído do campo D. O argumento de D pode ser DATE ou TIMESTAMP.	Retorna os dados de acordo com o argumento Opção, que pode ser YEAR, DAY, HOUR, MINUTE, SECOND, entre outros.
LAST_DAY	LAST_DAY (D) , onde D é do Tipo DATE.	Retorna a data do último dia do mês da data de entrada.
LOCALTIMESTAMP	LOCALTIMESTAMP [precisão] , onde [precisão] é um numero que se for especificado indica a precisão em frações de segundos.	Retorna a data atual do fuso horário da sessão com valor TIMESTAMP.
MONTHS_BETWEEN	MONTHS_BETWEEN (data1, data2) , onde data1 e data2 são as datas de entrada com valor DATE.	Retorna o numero de meses entre a data1 e data2.
NEW_TIME	NEW_TIME (D, zona1, zona2) , onde D é a data e zona1 e zona2 são strings de caractere que podem ser consultados na visão v\$timezone_names.	Retorna a data e a hora do fuso horário da zona2 quando a data e a hora da zona 1 for igual ao argumento D.
NEXT_DAY	NEXT_DAY (D, string) , onde D é a data de entrada e string especifica o dia da semana para fazer aproximação com D.	Retorna a data do dia da semana mais próximo ao argumento de D.
ROUND	ROUND (D, [formato]) , onde D é a data de entrada e formato especifica como fazer o arredondamento.	Retorna a data de acordo com o formato. O formato pode ser: DD, MM, YYYY, HH, MI, SS, entre outros.
SESSIONTIMEZONE	SESSIONTIMEZONE	Retorna o fuso horário da sessão atual com valor string.
SYS_EXTRACT_UTC	SYS_EXTRACT_UTC (D) , onde D é data de entrada com valor DATE.	Retorna a hora em UTC (<i>Coordinated Universal Time</i>) ou meridiano de <i>Greenwich</i> a partir do argumento D.
SYSDATE	SYSDATE	Retorna a data e a hora atual do sistema com valor DATE.
SYSTIMESTAMP	SYSTIMESTAMP	Retorna a data e a hora atual do sistema com valor TIMESTAMP WITH ZONE.

TRUNC	TRUNC (D, [formato]), onde D é a data de entrada e formato especifica como fazer o truncamento.	Retorna a data D de acordo com o formato. O formato pode ser: DD, MM, YYYY, HH, MI, SS, entre outros.
TZ_OFFSET	TZ_OFFSET (fuso horário), onde fuso horário é o nome do fuso horário.	Retorna o deslocamento entre o fuso horário fornecido e UTC.

2.2. O SGDB Firebird

O SGBD Firebird foi originado a partir do código fonte do Interbase, quando a Borland abriu o fonte para a comunidade (*Open Source*), podendo ser usado livremente, sem o pagamento de nenhuma licença (FIREBIRD, 2004).

Todo o código fonte do Firebird foi revisado e convertido da linguagem C para a linguagem C++ surgindo, então, a versão 1.5. Essa versão do Firebird está disponível para a plataforma *Windows* e *Linux* e traz diversas melhorias em relação a sua primeira versão (FIREBIRD, 2004).

Algumas empresas como a EMBRAPA (Empresa Brasileira de Pesquisa Agropecuária), a ESALQ da USP (clínica do leite) e várias outras empresas de desenvolvimento de software adotaram o Firebird em seus projetos (CANTU, 2005).

No Firebird há possibilidade de criar uma banco de dados usando um dos seus dialetos, o dialeto¹ 1, dialeto 2 ou dialeto 3. O dialeto 2 é recomendado apenas para fazer e depurações no banco de dados; o dialeto 1 e o dialeto 3 são mais recomendados para serem utilizados, sendo que o dialeto 3 é o mais utilizado hoje em dia (CANTU, 2005).

O Firebird tem a capacidade de armazenar diversos tipos de dados: inteiro, caractere, decimal, decimal de dupla precisão, data, hora, e até textos e imagens. Para armazenar datas o Firebird se dispõe de três formas: o tipo DATE, para armazenar a data, o tipo TIME, para armazenar a hora, e o tipo TIMESTAMP armazena tanto a data como a hora (Dialeto 3).

¹ O dialeto é um tipo de linguagem que o SGDB vai utilizar para determinar os tipos de dados disponíveis e a maneira como serão armazenados.

Internamente esses tipos de são armazenados como inteiros longos e ocupam 8 *bytes* (CANTU, 2005).

O Firebird possui algumas funções para manipulação de data que já estão implementadas, conforme mostra a Tabela 2.2.

Tabela 2.2 – Sintaxe e a descrição das funções de manipulação de data do Firebird

Função	Sintaxe	Descrição
CURRENT_DATE	CURRENT_DATE	Retorna a data atual do sistema com valor DATE.
CURRENT_TIME	CURRENT_TIME [(precisão)], onde [precisão] é um numero que se for especificado indica a precisão em frações de segundos.	Retorna a hora atual do sistema com valor TIME.
CURRENT_TIMESTAMP	CURRENT_TIMESTAMP [(precisão)], onde [precisão] é um numero que se for especificado indica a precisão em frações de segundos.	Retorna a data e ao hora atual do sistema com valor TIMESTAMP.
EXTRACT	EXTRACT (Opção from Data) , onde Opção define o que será extraído do campo D. O argumento de D pode ser DATE ou TIMESTAMP.	Retorna os dados de acordo com o argumento Opção (DAY, MONTH, YEAR, WEEKDAY, HOUR, MINUTE, SECOND).

2.2.1. Stored Procedures

O Firebird possui um grande recurso que é a capacidade de se criar *Stored Procedures* (SP), isto é, um procedimento armazenado no banco de dados que pode ser executados através de comandos SQL (CANTU, 2005).

A utilização de SP traz algumas vantagens importantes para o servidor Firebird, como a melhora no desempenho com a redução de dados trafegados na rede e por não precisar de uma nova compilação em cada chamada. Uma outra vantagem das SP é a possibilidade de

reutilizá-la onde precisar (reutilização de código) ganhando tempo no desenvolvimento e manutenção de uma aplicação (TODD, 2000).

Para se criar uma SP no Firebird é necessário conhecer a linguagem PSQL que é uma extensão da linguagem SQL. A PSQL define vários comandos, sendo que muitos são parecidos com os comandos de programação utilizados em outras linguagens. A estrutura básica para se criar uma stored procedure é apresentada na Figura 2.1.

```
CREATE PROCEDURE [<nome da procedure>] ([<parâmetros de entrada>])
RETURNS ([<parâmetros de saída>])
AS
[<declaração de variáveis>]
BEGIN
    <bloco de código>
END;
```

Figura 2.1 – Estrutura básica de Stored Procedure

Na construção de uma SP também podem ser usados comandos da linguagem SQL, por exemplo, *select*, *insert*, *update*, *delete*, *alter* (CANTU, 2005).

As SP podem receber valores por meio dos parâmetros de entrada e retornar valores por meio dos parâmetros de saída. Assim, os programas aplicativos podem requisitar uma SP passando valores por meio de seus parâmetros de entrada e receberem resultados da SP passados através dos parâmetros de saída (TODD, 2000).

Para executar uma SP, pode ser utilizado o seguinte comando:

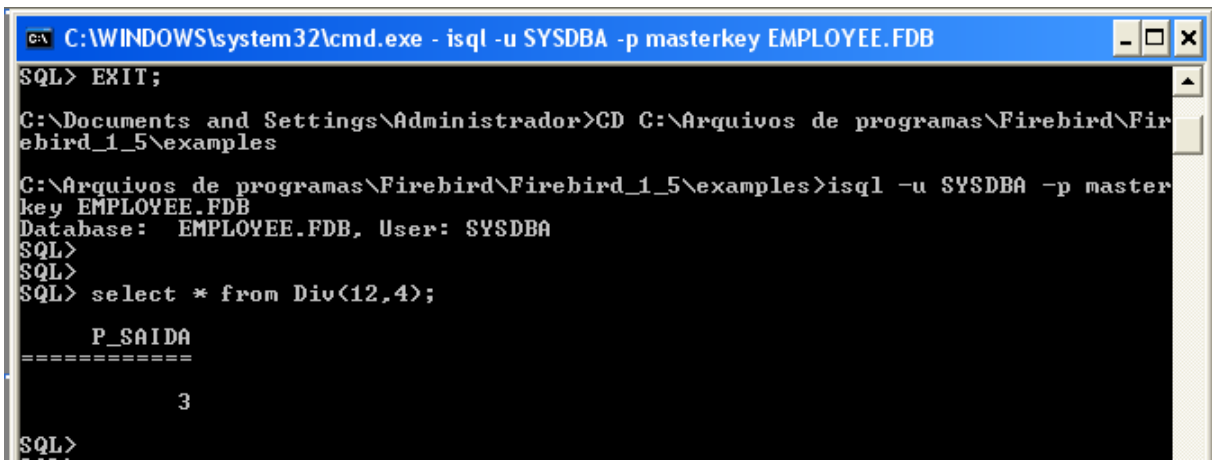
*Select * from <nome da procedure> ([parâmetros]);*

Outra maneira de executar uma procedure é usando o seguinte comando: *execute*

procedure <nome da procedure> ([parâmetros]);

A primeira forma de executar uma SP deve ser utilizada quando essa retorna algum valor. A segunda forma pode ser usada para chamar uma SP que esteja dentro de outra ou por uma aplicação. Essa forma deve ser usada quando a procedure não retorna nenhum valor, por exemplo, uma procedure que executa o comando *insert* para inserir os dados em uma tabela (TODD, 2000).

A Figura 2.2 apresenta um exemplo de uma chamada de SP e o seu resultado.



```
C:\WINDOWS\system32\cmd.exe - isql -u SYSDBA -p masterkey EMPLOYEE.FDB
SQL> EXIT;
C:\Documents and Settings\Administrador>CD C:\Arquivos de programas\Firebird\Firebird_1_5\examples
C:\Arquivos de programas\Firebird\Firebird_1_5\examples>isql -u SYSDBA -p masterkey EMPLOYEE.FDB
Database:  EMPLOYEE.FDB, User:  SYSDBA
SQL>
SQL>
SQL> select * from Div(12,4);
      P_SAIDA
=====
              3
SQL>
```

Figura 2.2 – Tela de exemplo de chamada de SP

3. Metodologia

Para a realização desse trabalho foi necessário passar por diversos passos: delimitação do escopo, pesquisa das funções de manipulação de data do Oracle; determinar quais das funções do Oracle não estão presentes no Firebird; um estudo literário detalhado das funções do Oracle escolhidas para a implementação; definição dos algoritmos para implementar as funções; estudo detalhado de como criar uma função no Firebird; implementação propriamente dita no SGDB Firebird.

3.1. Tecnologias utilizadas

Para a realização desse trabalho foram utilizadas as seguintes tecnologias: SGDB Oracle 9i e a ferramenta SQL*Plus, SGDB Firebird 1.5 e a ferramenta chamada IBExpert (CANTU, 2005).

A ferramenta IBExpert foi desenvolvida por uma empresa independente chamada HK-Software. Existe uma versão gratuita dessa ferramenta, chamada *Personal* e uma outra versão, a *Trial*. A versão *Trial* é uma versão de avaliação e só pode ser utilizada por apenas 45 dias. Essas duas versões podem ser obtidas no site: www.hksoftware.net/download. Essa ferramenta pode ser utilizada com os SGDBs Firebird e o Interbase.

Nesse trabalho também foi realizado teste com uma função do tipo UDF (*User Defined Function*), funções criadas pelo usuário e que são armazenadas em bibliotecas externas ao banco de dados.

O Oracle possui um utilitário padrão para a manipulação de uma base de dados, o SQL*Plus. Esse utilitário permite manipular o BD por meio de linhas de comando, o que pode trazer algumas dificuldades para os usuários que estão acostumados com interfaces mais sofisticadas com a presença de botões, ícones, grids e outros componentes visuais.

Antes de abrir o *prompt* para a entrada de comandos é necessário conectar-se a um servidor fazendo o login do usuário, informando qual é o usuário, a senha e o *Host* (servidor) como são mostradas na Figura 3.1.

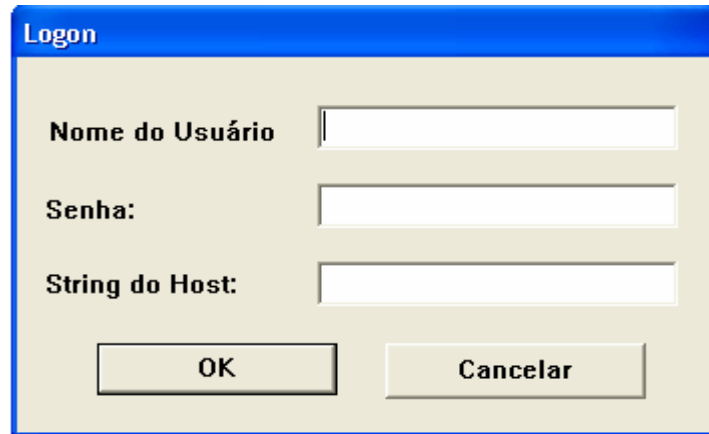
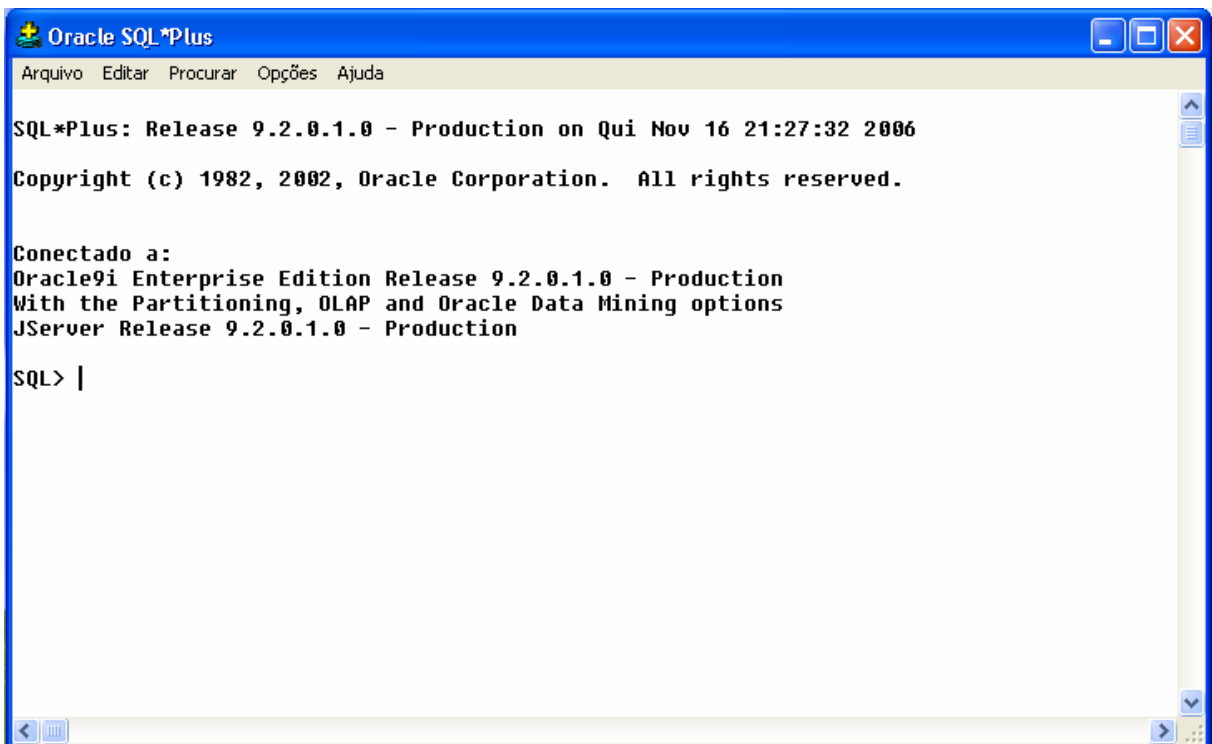
A imagem mostra uma caixa de diálogo intitulada "Logon" com um fundo bege e uma borda azul. Ela contém três campos de entrada de texto: "Nome do Usuário", "Senha:" e "String do Host:". Abaixo dos campos, há dois botões: "OK" e "Cancelar".

Figura 3.1 - Tela de conexão do SQL*Plus

Depois que o usuário passa pela tela de *Logon*, o *prompt* de do SQL*Plus estará disponível para receber os comandos desse usuário como é mostrado na Figura 3.2.

A imagem mostra a interface de usuário do Oracle SQL*Plus. O título da janela é "Oracle SQL*Plus". O menu de opções inclui "Arquivo", "Editar", "Procurar", "Opções" e "Ajuda". O conteúdo da janela é o seguinte:

```
SQL*Plus: Release 9.2.0.1.0 - Production on Qui Nov 16 21:27:32 2006
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Conectado a:
Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production

SQL> |
```

Figura 3.2 - Tela do *prompt* da SQL*Plus

A SQL*Plus foi utilizada nesse trabalho para a realização dos testes das funções de data do Oracle, e também para obtenção dos resultados na execução das funções onde foi verificado como cada função operava.

O IBExpert traz diversas facilidades para o usuário, como por exemplo, a criação de um novo banco de dados, criação de tabelas e outras diversas funcionalidades apenas com manipulação da ferramenta, sem que seja necessário digitar comandos para isso. Praticamente tudo que se faz com os comandos para manipular um BD se faz também de pelo IBExpert. Na Figura 3.3 é apresentada a tela do IBExpert junto com o seu editor de script SQL.

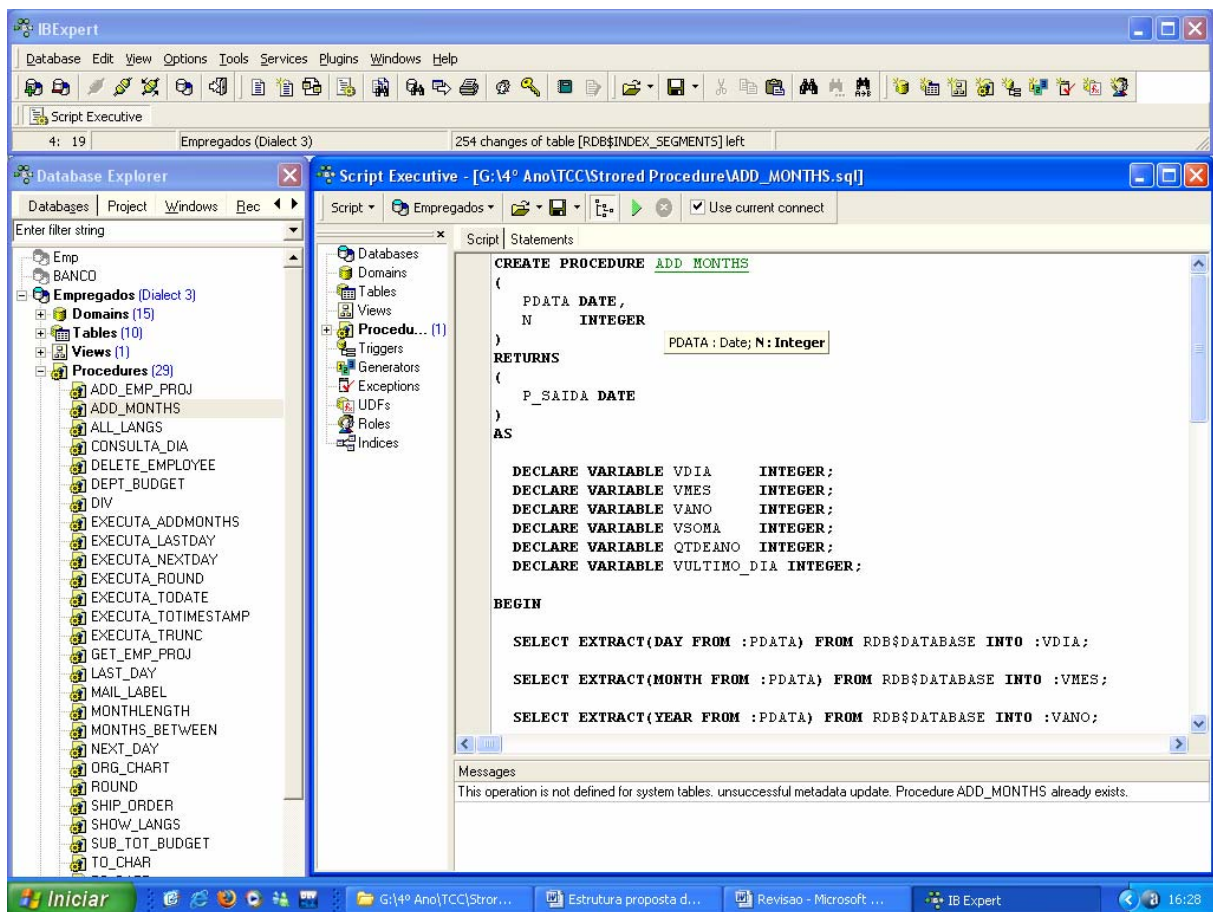


Figura 3.3 - Tela da ferramenta IBExpert

O IBExpert foi utilizado para a construção, execução, manutenção e testes das funções implementadas no Firebird.

3.1.1. Instalação e configuração dos SGDBs

Nesse trabalho foram instalados os SGDBs Oracle 9i e o Firebird 1.5. Os procedimentos de instalação e configuração podem ser conferidos nos apêndices A e B.

3.1.2. Linguagem de implementação

A linguagem utilizada na implementação das funções no Firebird foi a PSQL que é a linguagem padrão do Firebird para criação de SP. Essa linguagem combina a flexibilidade da linguagem SQL com os comandos de uma linguagem de programação comum (CANTU, 2005).

A seguir será apresentado um resumo sobre os comandos e características dessa linguagem.

- **Comentários:** os comentários podem ser de duas formas: o comentário de uma única linha definido pelo uso do "--" ou o comentário de várias linhas definido pelo uso dos delimitadores "/* */".
- **Variáveis:** as variáveis são sempre criadas de forma local; por isso podem ser utilizadas apenas dentro das SP onde foram criadas. Sintaxe:


```
DECLARE VARIABLE <nome> <tipo> ;
```
- **Bloco:** uma estrutura de bloco é o trecho que está entre um comando BEGIN e um END e que pode ainda conter outros blocos.
- **Exception:** serve para gerar uma exceção que é pré-definida pelo usuário. Para definir uma exceção deve-se usar o comando CREATE EXCEPTION. Sintaxe:

```
CREATE EXCEPTION <nome> ;
```

- **Execute Procedure:** serve para chamar uma Stored Procedure que esteja dentro de outra geralmente para armazenar o resultado nos seus parâmetros de saída. Sintaxe:

EXECUTE PROCEDURE <nome> [parâmetros de entrada]

[**RETURNING_VALUES** parâmetros de saída] ;

- **If Then Else:** é um comando condicional utilizado quando há necessidade de verificar uma determinada condição para então executar um trecho de código. Se uma dada condição for verdadeira, o bloco1 é executado, senão é executado o bloco seguinte. Esse comando também pode ser utilizado sem o Else. Sintaxe do If com Else:

If (<condição>) **Then** <bloco1> **Else** <bloco2> ;

- **Post_Event:** esse comando é utilizado para enviar um evento do servidor para aplicações que estão registradas para recebê-los. Sintaxe:

POST_EVENT <nome do evento> | <campo> ;

- **Select Into:** esse comando também pode ser usado em SPs para armazenar um valor retornado em uma variável para posterior utilização. Sintaxe:

Select <função> **from** <tabela> **into** :<variavel> ;

- **Suspend:** esse comando é utilizado para suspender a execução de uma SP para que ela devolva o primeiro registro com o valor atribuído no seu parâmetro de saída.

- **When Do:** esse comando é utilizado para tratar erros em SPs. Quando um erro é gerado o comando When é procurado dentro de um bloco. Se nenhum comando desse for encontrado todas as ações executadas são desfeitas.
- **While Do:** esse comando é utilizado quando se deseja fazer um laço, para executar um determinado comando ou uma instrução varias vezes. É o único comando de laço genérico para utilizar em SPs. Sintaxe do comando While com a estrutura de bloco:

While (<condição>) Do Begin <bloco> End

- **For Select do:** é um tipo de select com laço que permiti percorrer um conjunto de dados (campos de tabela), criando um cursor para isso. É necessário a clausula INTO nesse tipo de select para armazenar os valores retornados desse select. Sintaxe desse comando com a estrutura de bloco:

For Select <campos> From <tabela> into : <variavel> Do Begin <bloco> End

- **Case:** é um comando condicional que permite escolher um valor baseado nas condições criadas nesse comando. Sintaxe:

**Select <campo1>, <campo2>,
Case campo3 when <condição1> : <valor>;**

- **Break:** esse comando permite que um determinado laço seja interrompido. O processamento é desviado para a próxima instrução depois do fim do laço.

3.2. Funções implementadas no Firebird

De todas as funções do Oracle para manipulação de data, apresentadas no Capítulo 2, algumas foram escolhidas neste trabalho para serem implementadas no SGDB Firebird. A escolha por funções de manipulação de data dá-se pelo fato da constante necessidade dessas manipulações quando são desenvolvidos sistemas de informação que fazem conexões a SGBDs.

Foram escolhidas funções de manipulação de data disponibilizadas no SGBD Oracle e não disponibilizadas no SGBD Firebird. Funções que não fazem manipulação de data relacionada ao fuso horário da sessão do usuário, ao *Meridiano de Greenwich* ou *UTC* ou, ainda, a zona de tempo (*time zone*). Não foram escolhidas porque essas características não estão presentes no SGDB Firebird.

Dentre todas as funções apresentadas foram implementadas: **ADD_MONTHS**, **LAST_DAY**, **MONTHS_BETWEEN**, **NEXT_DAY**, **TO_CHAR**, **TO_DATE**, **TO_TIMESTAMP**, **ROUND** e **TRUNC**.

3.3. Definição e descrição dos algoritmos das funções

Antes de começar a definir os algoritmos das funções, foram implementadas duas funções como Stored Procedure, usadas como auxiliares na implementação das demais funções. A primeira função auxiliar a ser definida foi a função **DIV** e a segunda foi a **CONSULTA_DIA**.

A função **DIV** tem como finalidade obter a divisão inteira de um numero por outro. É utilizada principalmente onde é necessário avaliar se um determinado ano é bisexto. A função **CONSULTA_DIA** tem como finalidade obter o último dia de um mês. É utilizada para comparar se um determinado dia é o ultimo dia do mês.

A seguir são apresentados os algoritmos das funções implementadas nesse trabalho inclusive o algoritmo das duas funções auxiliares.

A Figura 3.22 mostra a definição do algoritmo da função auxiliar **DIV**.

```
inicio
  contador = 0
  aux = n1 - n2
  Enquanto (aux for maior ou igual a zero) faça
  inicio
    incrementa o contador em 1;
    aux = aux - n2
  fim
fim.
```

Figura 3.4 - Definição do algoritmo da função auxiliar **DIV**

A Figura 3.23 mostra a definição do algoritmo da função auxiliar **CONSULTA_DIA**.

```
inicio
  Se (mes = 2) então
  inicio
    Se (ano for bisexto) então
      resultado = 29
    Senão
      resultado = 28
  fim se;
  Senão
  inicio
    resultado = o ultimo dia do mes
  fim senão
fim.
```

Figura 3.5 - Definição do algoritmo da função auxiliar **CONSULTA_DIA**

A Figura 3.24 mostra a definição do algoritmo da função **ADD_MONTHS**.

```

inicio
    separa data em dia, mês e ano
    soma = mes + n;
    qtdeano = divisão inteira de soma por 12;

    Se (soma > 12) então
        inicio
            Ajusta o mes e o ano
        fim se;

    mes = soma;

    Se (dia > ultimo dia do mes) então
        dia = ultimo dia do mes;

    monta a nova data no formato dia.mes.ano;

fim.

```

Figura 3.6 – Definição do algoritmo da função ADD_MONTHS

A Figura 3.23 mostra a definição do algoritmo da função **LAST_DAY**.

```

inicio
    separa o dia, mes e ano da data
    pega o dia retornado pelo procedimento CONSULTA_DIA
    monta a nova data no formato dia.mes.ano
fim.

```

Figura 3.7 - Definição do algoritmo da função LAST_DAY

A Figura 3.26 mostra a definição do algoritmo da função **MONTHS_BETWEEN**.

```

inicio
    separa o mes da data1 e da data2
    separa o ano da data1 e da data2

    calcula a diferença de meses entre a data1 e data2
    Se (ano da data1 <> ano da data2) então
        inicio
            armazena a diferença de anos entre a data1 e data2
            calcula a quantidade de meses entre a data1 e data2
        fim se
    fim.

```

Figura 3.8 - Definição do algoritmo da função MONTHS_BETWEEN

A Figura 3.27 mostra a definição do algoritmo da função **NEXT_DAY**.

```

inicio

  separa da data o dia, mes e o ano
  extrai o dia da semana da data
  troca o valor de day pelo numero do dia da semana correspondente

  faz o deslocamento de quantos dias da semana falta para chegar em day
  dia = dia + deslocamento
  se (dia > ultimo dia do mes) então
  inicio
    ajusta o dia, mes e o ano
  fim se

  monta a nova data no formato dia.mes.ano
fim.

```

Figura 3.9 - Definição do algoritmo da função NEXT_DAY

A Figura 3.28 mostra a definição do algoritmo da função **TO_DATE**.

```

inicio

  Se (formato = 'DD/MM/YYYY' ou 'DD-MM-YYYY' ou 'DD.MM.YYYY') então
  inicio
    separa da data o dia, mes e o ano
  fim se;

  Se (formato = 'MM/DD/YYYY' ou 'MM-DD-YYYY' ou 'MM.DD.YYYY') então
  inicio
    separa da data o dia, mes e o ano
  fim se;

  Se (formato = 'DD/MON/YYYY' ou 'DD-MON-YYYY' ou 'DD.MON.YYYY') então
  inicio
    separa da data o dia, mes e o ano
    troca as letras iniciais do mes pelo seu numero correspondente
  fim se

  monta a data no formato dia.mes.ano
fim.

```

Figura 3.10 - Definição do algoritmo da função TO_DATE

A Figura 3.29 mostra a definição do algoritmo da função **TO_TIMESTAMP**.

```
inicio

Se (formato = 'DD/MM/YYYY HH:MI:SS' ou 'DD-MM-YYYY HH:MI:SS' ou
  'DD.MM.YYYY HH:MI:SS' ou vazio) então
  inicio
    separa da data o dia, mes, ano, e a parte da hora
  fim se;

Se (formato for igual a 'MM/DD/YYYY HH:MI:SS' ou 'MM-DD-YYYY HH:MI:SS' ou
  'MM.DD.YYYY HH:MI:SS') então
  inicio
    separa da data o dia, mes, ano, e a parte da hora
  fim se;

Se (formato for igual a 'DD/MON/YYYY HH:MI:SS' ou 'DD-MON-YYYY HH:MI:SS'
  ou 'DD.MON.YYYY HH:MI:SS') então
  inicio
    separa da data o dia, mes, ano, e a parte da hora
    troca as letras iniciais do mes pelo seu numero correspondente
  fim se;

monta a data no formato dia.mes.ano com a parte da hora
fim.
```

Figura 3.11 - Definição do algoritmo da função TO_TIMESTAMP

A Figura 3.30 mostra a definição do algoritmo da função **TO_CHAR**.

```

inicio
  Se (formato = 'D') então
    inicio
      extrai da data o numero
      correspondente ao dia da semana
      incrementa o numero extraído
    fim se

  Se (formato = 'DAY' ou 'DY') então
    inicio
      extrai da data o numero
      correspondente ao dia da semana

      se (formato = 'DY') então
        troca numero do dia da semana
        pelo nome do dia abreviado
        Senão
          troca o numero do dia da
          semana pelo nome do dia

    fim se

  Se (formato = 'DDD') então
    inicio
      extrai o numero do dia no ano
    fim se

  Se (formato = 'MM') então
    inicio
      separa o mes da data
    fim se

  inicio
    separa a hora da data
  fim se

  Se (formato = 'MI') então
    inicio
      separa o minuto da data
    fim se

  Se (formato = 'SS') então
    inicio
      separa o segundo da data da data
    fim se

  Se (formato = 'DD/MM/YYYY' ou 'DD-
  MM-YYYY' ou 'DD.MM.YYYY' ou
  DD/MON/YYYY' ou 'DD-MON-YYYY'
  ou 'DD.MON.YYYY' ou vazio) então
    inicio
      pega o caracter separador que
      esta em formato
      separa da data o dia, mes e o
      ano

      Se (parte do mes no formato for
      igual a 'MON') então
        Se (formato = 'MONTH' ou 'MON')
        então
          inicio
            separa o mes da data

            se (formato = 'MON') então
              retorna o nome abreviado do
              mes

            Senão
              retorna nome do mes da data
            fim se

          Se (formato = 'YYYY' ou 'RRRR')
          então
            inicio
              separa o ano da data com 4
              dígitos
            fim se

          Se (formato = 'YY' ou 'RR') então
            inicio
              separa ano da data com os 2
              últimos dígitos
            fim se;

          Se (formato = 'Y') então
            inicio
              separa o ano da data com o ultimo
              dígito
            fim se

          Se (formato = 'HH' ou 'HH12' ou
          'HH24') então
            troca o mes pelo seu nome
            abreviado de acordo;

            Se (tiver a parte da hora) então
              inicio
                separa a hora, minuto e o
                segundo
              fim se

            monta data de acordo com o
            separador
            fim se

          Se (formato = 'Q') então
            inicio
              separa o mes da data
              verifica qual o trimestre
              pertence o mes da data
            fim se;

          Se (formato = 'RM') então
            inicio
              separa o mes da data
              troca o numero do mes para
              algarismo romano
            fim se;
          fim.

```

Figura 3.12 – Definição do algoritmo da função **TO_CHAR**

A Figura 3.31 mostra a definição do algoritmo da função **ROUND**.

```

início
    separa data em dia, mês, ano, hora,
    minuto, segundo
    Se (formato for vazio) então
        início
            Se ( hora > 12 ) então
                Retorna a data do próximo dia
            Senão
                retorna a mesma data de entrada
        fim se

    Se ( formato = 'DAY' ou 'D' ou 'DY'
    ) então
        início
            se (data for antes da Quarta-
            feira e antes do meio dia) então
                retorna a data inicial da
                semana corrente
            Senão
                retorna a data inicial da
                próxima semana
        fim se

    Se ( formato = 'MONTH' ou 'MM' ou
    'MOM' ) então
        início
            se (data for antes do dia 15 do
            mes corrente) então
                retorna a data inicial do mes
                corrente
            Senão

                retorna a mesma parte da data,
                a hora, passa para o próximo minuto e
                zera os segundos

        fim se

    Se ( formato = 'Q' ) então
        início

            Se ( mes = 1 ou 4 ou 7 ou 10 )
            então
                início
                    retorna o dia inicial desse
                mes
                fim se

            Se ( mes = 2 ou 5 ou 8 ou 9 )
            então
                início

                retorna a data inicial do
                próximo mes

            fim se

        fim.

        retorna a data inicial do
        próximo mes
        fim se

    Se ( formato = 'YEAR' ou 'YYY' ou
    'YY' ) então
        início
            se (mes < 7) então
                retorna a data inicial da ano
                corrente
            Senão
                retorna a data inicial do
                próxima ano
        fim se

    Se ( formato = 'HH' ) então
        início
            se (minuto >= 30) então
                retorna a mesma parte da data
                de entrada com a mesma hora inicial
            Senão
                retorna a mesma parte da data
                de entrada com a próxima hora
        fim se

    Se ( formato = 'MI' ) então
        início
            se (segundo >= 30) então
                retorna a mesma parte da data,
                a hora, o minuto e zera os segundos
            Senão

            Se (dia > 15) então
                retorna a data inicial do
                próximo trimestre
            Senão
                retorna a data inicial do
                trimestre corrente
        fim se

    Se ( mes = 3 ou 6 ou 9 ou 12 )
    então
        início
            retorna a data inicial do
            próximo trimestre
        fim se

    fim se

```

Figura 3.13 – Definição do algoritmo da função **ROUND**

A Figura 3.32 mostra a definição do algoritmo da função **TRUNC**.

```

inicio

separa data em dia, mês, ano, hora, minuto, segundo
Se (formato = 'DAY' ou 'D' ou 'DY') então
  inicio
    retorna a data inicial da semana corrente
  fim se

Se (formato = 'DD' ou 'DDD' ou vazio) então
  inicio
    retorna a mesma data de entrada
  fim se

Se (formato = 'MONTH' ou 'MON' ou 'MMM') então
  inicio
    retorna a data inicial do mes corrente
  fim se

Se (formato for igual a 'YEAR' ou 'YYY' ou 'YY') então
  inicio
    retorna a data inicial do ano corrente
  fim se

Se (formato for igual a 'HH') então
  inicio
    retorna a mesma data, hora e zero os minutos e segundos da hora
  fim se

Se (formato for igual a 'MI') então
  inicio
    retorna a mesma data, hora, minutos e zero os segundos da hora
  fim se

fim.

```

Figura 3.14 – Definição do algoritmo da função TRUNC

3.4. Forma de execuções das funções

Há duas formas de se executar uma função (*Stored Procedure*) no Firebird. Uma delas é usando o comando *EXECUTE PROCEDURE* <nome da função> e a outra é através do comando *SELECT * FROM* <nome da função>.

A Tabela 3.1 mostra as formas de execução de cada função implementada no Firebird.

Tabela 3.1 – Formas de execução das funções no Firebird

Função	Forma de execução com Execute procedure	Forma de execução com Select
ADD_MONTHS	Execute procedure ADD_MONTHS (data,numero)	Select * from ADD_MONTHS (data,numero)
LAST_DAY	Execute procedure LAST_DAY (data)	Select * from LAST_DAY (data)
MONTHS_BETWEEN	Execute procedure MONTHS_BETWEEN (data1,data2)	Select * from MONTHS_BETWEEN (data1,data2)
NEXT_DAY	Execute procedure NEXT_DAY (data,day)	Select * from NEXT_DAY (data,day)
TO_CHAR	Execute procedure TO_CHAR (data,formato)	Select * from TO_CHAR (data,formato)
TO_DATE	Execute procedure TO_DATE (data,formato)	Select * from TO_DATE (data,formato)
TO_TIMESTAMP	Execute procedure TO_TIMESTAMP (data,numero)	Select * from TO_TIMESTAMP (data,formato)
ROUND	Execute procedure ROUND (data,numero)	Select * from ROUND (data,formato)
TRUNC	Execute procedure TRUNC (data,numero)	Select * from TRUNC (data,formato)

No próximo capítulo serão apresentados os resultados desse trabalho, a descrição dos testes no SGDB do Oracle 9i e do Firebird 1.5.

4. Resultados e Discussões

4.1. Descrição dos resultados no Oracle

Os resultados no Oracle foram observados através de seu utilitário SQL*Plus. Esse utilitário mostra os resultados de cada função que é executada na mesma janela, pois seu ambiente é um *prompt* de comandos (interface em modo texto). O *prompt* de comandos é uma janela que espera a entrada do usuário e retorna a saída na mesma tela.

A Tabela 4.1 mostra o resultado de cada função executada no Oracle:

Tabela 4.1 – Exemplos e os resultados das funções executadas no Oracle

Função	Exemplo de Execução	Resultado
ADD_MONTHS	select ADD_MONTHS ('29-02-04',12) from Dual;	Um atributo do tipo DATE com valor 28/02/05
LAST_DAY	SELECT LAST_DAY (SYSDATE) FROM DUAL;	Um atributo do tipo DATE com valor 31/10/06, onde sysdate = 19/10/2006
MONTHS_BETWEEN	SELECT MONTHS_BETWEEN ('01-09-95', '11-01-94') FROM DUAL;	Um atributo do tipo NUMBER com valor 19,6774194
NEXT_DAY	select NEXT_DAY (sysdate,6) from Dual;	Um atributo do tipo DATE com valor 20/10/06, onde sysdate = 17/10/06
TO_CHAR	SELECT TO_CHAR (SYSDATE, 'DD/MM/YYYY') FROM DUAL;	Um atributo do tipo STRING, com o valor 20/10/2006, onde sysdate = 20/10/2006
TO_DATE	SELECT TO_DATE ('01/10/2006', 'DD/MM/YYYY') FROM DUAL;	Um atributo do tipo DATE com valor 01/10/06
TO_TIMESTAMP	select TO_TIMESTAMP ('01/10/2006', 'DD/MM/YYYY HH:MI:SS') from Dual;	Um atributo do tipo TIMESTAMP com valor 01/10/2006 20:18:40
ROUND	SELECT ROUND (SYSDATE) FROM DUAL;	Um atributo do tipo DATE com valor 25/10/06, sendo sysdate = 24/10/2006
TRUNC	SELECT TRUNC (SYSDATE) FROM DUAL;	Um atributo do tipo DATE com valor 24/10/2006, sendo que sysdate = 24/10/2006

4.2. Descrição dos resultados no Firebird

Os resultados no Firebird foram observados através da ferramenta IBExpert. Essa ferramenta mostra o resultado de cada função (*Stored Procedure*) que é executada no seu ambiente. A Tabela 4.2 mostra o resultado de cada função que foi implementada no Firebird.

Tabela 4.2 - Exemplos e os resultados das funções executadas no Firebird

Função	Exemplo de Execução	Resultado
ADD_MONTHS	select * from ADD_MONTHS ('29.02.2004',12);	Um atributo do tipo DATE com valor 28.02.2005
LAST_DAY	select * from LAST_DAY (current_date);	Um atributo do tipo DATE com valor 31.10.2006, onde current_date = 19.10.2006
MONTHS_BETWEEN	select * from MONTHS_BETWEEN ('01.09.1995', '11.01.1994');	Um atributo do tipo NUMBER com valor 20
NEXT_DAY	select * from NEXT_DAY (current_date,6);	Um atributo do tipo DATE com valor 20.10.2006, onde current_date = 17/10/06
TO_CHAR	select * from TO_CHAR ('12.11.2006','DD/MM/YYYY');	Um atributo do tipo STRING, com o valor 20/10/2006, onde current_date = 20.10.2006
TO_DATE	select * from TO_DATE ('01/10/2006','DD/MM/YYYY');	Um atributo do tipo DATE com valor 01.10.2006
TO_TIMESTAMP	select * from TO_TIMESTAMP ('01/10/2006 20:18:40','DD/MM/YYYY HH:MI:SS');	Um atributo do tipo TIMESTAMP com valor 01.10.2006 20:18:40
ROUND	select * from ROUND (current_timestamp,');	Um atributo do tipo TIMESTAMP (Date Time) com valor 25.10.2006 00:00:00, onde current_timestamp = 24.10.2006
TRUNC	select * from TRUNC (current_timestamp,');	Um atributo do tipo TIMESTAMP com valor 24.10.2006 00:00:00, sendo que current_timestamp = 24.10.2006

4.3. Análise de desempenho por função

Nessa seção será feita uma comparação de cada função implementada, comparando o desempenho entre os SGDBs Oracle e Firebird, com relação ao número de execuções e o tempo gasto em milisegundos. Tanto no Oracle quanto no Firebird foram criados *scripts* para cada função com objetivo de testar a execução da mesma para as seguintes quantidades de vezes: 100, 500, 1000 e 5000.

Para os testes realizados no Firebird foi utilizada a Ferramenta IBExpert para a execução dos *scripts* de teste e para marcar o tempo gasto com a execução de cada *script*. Essa ferramenta mostra o tempo em que foi gasto, em milisegundos, para executar cada função em seu ambiente.

Para os testes realizados no Oracle foram utilizados os seguintes passos para a obtenção do tempo gasto na execução dos *scripts* de teste:

1. Criar uma classe em Java com um método que retorne o tempo corrente em milisegundos.
2. Carregar a classe no Oracle através do utilitário *loadjava*. Para adicionar essa classe no Oracle é necessário compilá-la com a JVM (*Java Virtual Machine*) que é instalada junto com Oracle ou utilizar uma versão compatível. Após compilar a classe, deve-se executar o utilitário *loadjava* fornecendo o seguinte comando:

c:> loadjava -user scott/tiger nome_da_classe.class

3. Após esses dois passos, deve ser criada uma *Stored Procedure* no Oracle para receber o valor que é retornado pelo método da classe que foi adicionada no Oracle. Uma vez criada essa *Stored Procedure* ela pode ser utilizada diretamente no ambiente da SQL*Plus ou dentro de uma *Stored Procedure*.

4. Criar os *scripts* de teste com uma chamada dessa Stored Procedure no início do *script* e outra chamada no fim do *script* para que possa ser marcado o tempo inicial e o tempo final da execução. Ao final da execução do *script* é obtido o tempo gasto em milisegundos, realizando a subtração do tempo final pelo tempo inicial.

Os testes das funções foram realizados em uma máquina com processador *AMD Athlon XP* 1,60 GHz com 256 MB de memória *RAM* e 40 GB HD (*Hard Disk*).

Os cálculos das médias dos testes foram feitos aplicando-se as fórmulas mostradas na figura 4.1.

$$\text{MGF} = \frac{\left(\frac{T1}{N1} + \frac{T2}{N2} + \frac{T3}{N3} + \frac{T4}{N4}\right)}{4}, \text{ onde MGF} = \text{Média Geral do Firebird}$$

$$\text{MGO} = \frac{\left(\frac{T1}{N1} + \frac{T2}{N2} + \frac{T3}{N3} + \frac{T4}{N4}\right)}{4}, \text{ onde MGO} = \text{Média Geral do Oracle}$$

$$\text{MGT} = \left(\frac{\text{MGF}}{\text{MGO}}\right) - 1, \text{ onde MGT} = \text{Média Geral Total}$$

onde:
T1, T2, T3, T4 são os tempos gastos para executar cada teste
N1, N2, N3, N4 é a quantidade de execuções de cada teste.

Figura 4.1 – Fórmulas para os cálculos das médias

4.3.1. Função **ADD_MONTHS**

O gráfico da Figura 4.2 mostra a comparação da função **ADD_MONTHS** implementada no Firebird com sua correspondente no Oracle:

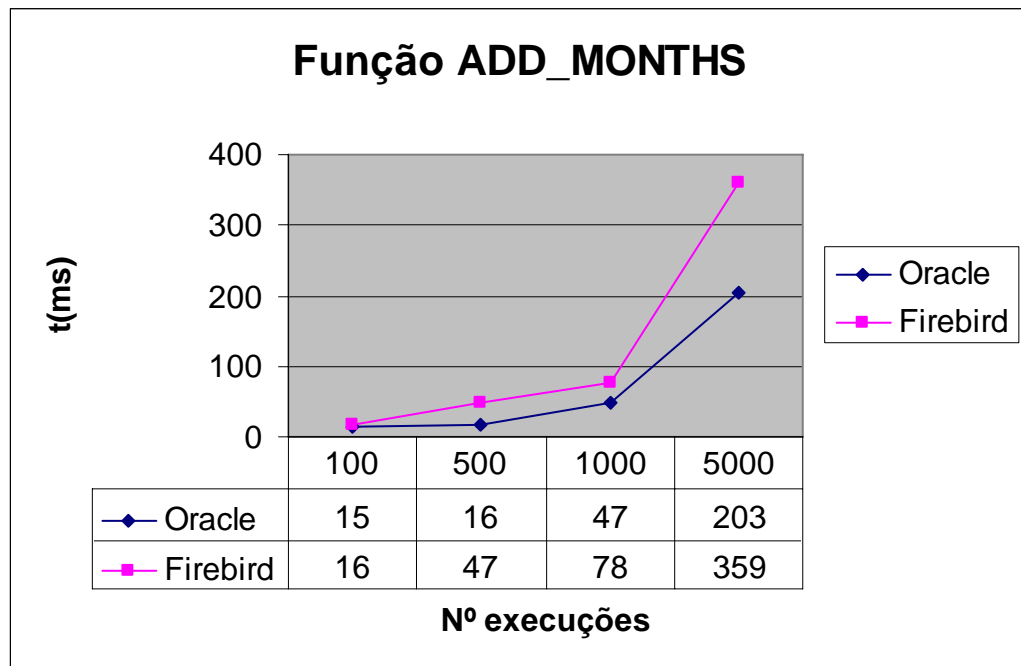


Figura 4.2 – Gráfico com os resultados dos testes da função ADD_MONTHS

A Tabela 4.3 mostra um resumo do gráfico acima, com o resultado das médias individuais e a média geral obtida nos testes realizados para os dois SGDBs.

Tabela 4.3 – Média dos testes com a função ADD_MONTHS

SGDB	Média de tempo para 100 execuções	Média de tempo para 500 execuções	Média de tempo para 1000 execuções	Média de tempo para 5000 execuções	Média Geral
Oracle	0,15	0,032	0,047	0,0406	0,0674
Firebird	0,16	0,094	0,078	0,0718	0,10095

Como pode ser visto na Figura 4.2 e na Tabela 4.3 o Oracle obteve melhor desempenho na execução em todos os testes realizados para essa função. Para o teste com 100 execuções o Firebird gastou em média 6.66 % de tempo a mais que o Oracle, nesse caso o desempenho do Firebird ficou bem próxima da performance do Oracle. Já para os testes com 500, 1000 e 5000 o desempenho do Oracle foi bem superior ao Firebird que levou em média 193.75 % (aproximadamente 2 vezes mais), 65.95 %, 76.84 % do tempo a mais para realizar os respectivos testes.

Na média geral o Firebird gastou 49.77 % a mais de tempo para realizar todos os testes com essa função. Isso pode ter ocorrido devido a vários fatores, entre eles a forma com que esse SGDB faz a alocação de recursos.

4.3.2. Função LAST_DAY

O gráfico da Figura 4.3 mostra a comparação da função **LAST_DAY** implementada no Firebird com sua correspondente no Oracle:

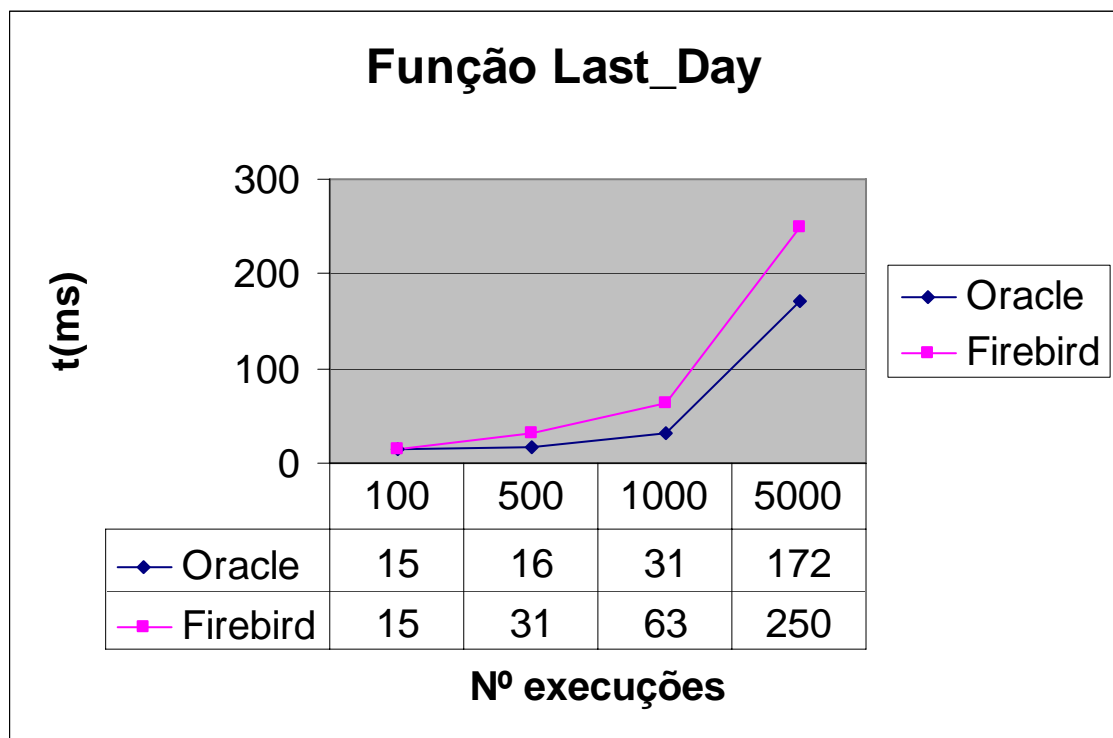


Figura 4.3 - Gráfico com os resultados dos testes da função **LAST_DAY**

A Tabela 4.4 mostra um resumo do gráfico acima, com o resultado das médias individuais e a média geral obtida nos testes realizados para os dois SGDBs.

Tabela 4.4 - Média dos testes com a função LAST_DAY

SGDB	Média de tempo para 100 execuções	Média de tempo para 500 execuções	Média de tempo para 1000 execuções	Média de tempo para 5000 execuções	Média Geral
Oracle	0,15	0,032	0,031	0,0304	0,06185
Firebird	0,15	0,062	0,063	0,05	0,08125

Como pode ser visto na Figura 4.3 e na Tabela 4.4 o Oracle obteve, em geral, melhor desempenho para a maioria dos testes realizados o que não aconteceu nos testes da função anterior. Para o teste com 100 execuções o Firebird teve um bom desempenho se igualando ao do Oracle. Já para os testes com 500 e 1000 o desempenho do Oracle foi bem superior ao Firebird que gastou em média 93.75 % e 103.22 % para os respectivos testes. Isso resulta em um gasto de aproximadamente o dobro do tempo em relação ao Oracle. Para o teste com 5000 o Firebird melhorou seu desempenho gastando em média 45.34 % reduzindo cerca de 57 % do tempo que foi gasto para executar o teste anterior, realizado com 1000 execuções.

Na média geral o Firebird gastou 31.36 % a mais de tempo para realizar todos os testes com essa função.

4.3.3. Função MONTHS_BETWEEN

O gráfico da Figura 4.4 mostra a comparação da função **MONTHS_BETWEEN** implementada no Firebird com sua correspondente no Oracle:

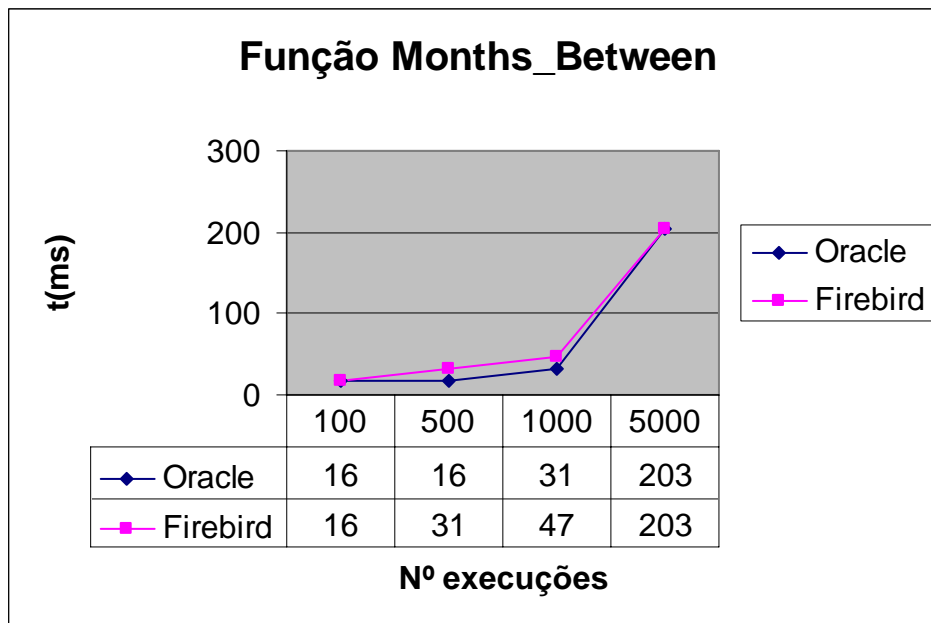


Figura 4.4 - Gráfico com os resultados dos testes da função MONTHS_BETWEEN

A Tabela 4.5 mostra um resumo do gráfico acima, com o resultado das médias individuais e a média geral obtida nos testes realizados para os dois SGDBs.

Tabela 4.5 - Média dos testes com a função MONTHS_BETWEEN

SGDB	Média de tempo para 100 execuções	Média de tempo para 500 execuções	Média de tempo para 1000 execuções	Média de tempo para 5000 execuções	Média Geral
Oracle	0,16	0,032	0,031	0,0406	0,0659
Firebird	0,16	0,062	0,047	0,0406	0,0774

Como pode ser visto na Figura 4.4 e na Tabela 4.5 o Firebird obteve um bom desempenho se igualando ao do Oracle para os testes realizados com 100 e 5000 execuções. Já para os testes com 500 e 1000 o desempenho do Oracle foi bem superior ao Firebird que gastou em média 93.75 % e 51.61 % para os respectivos testes. Para o teste com 5000 o Firebird melhorou o seu desempenho e voltou a se igualar ao Oracle.

Na média geral o Firebird gastou 17.45 % a mais de tempo para realizar todos os testes com essa função.

4.3.4. Função NEXT_DAY

O gráfico da Figura 4.5 mostra a comparação da função **NEXT_DAY** implementada no Firebird com sua correspondente no Oracle:

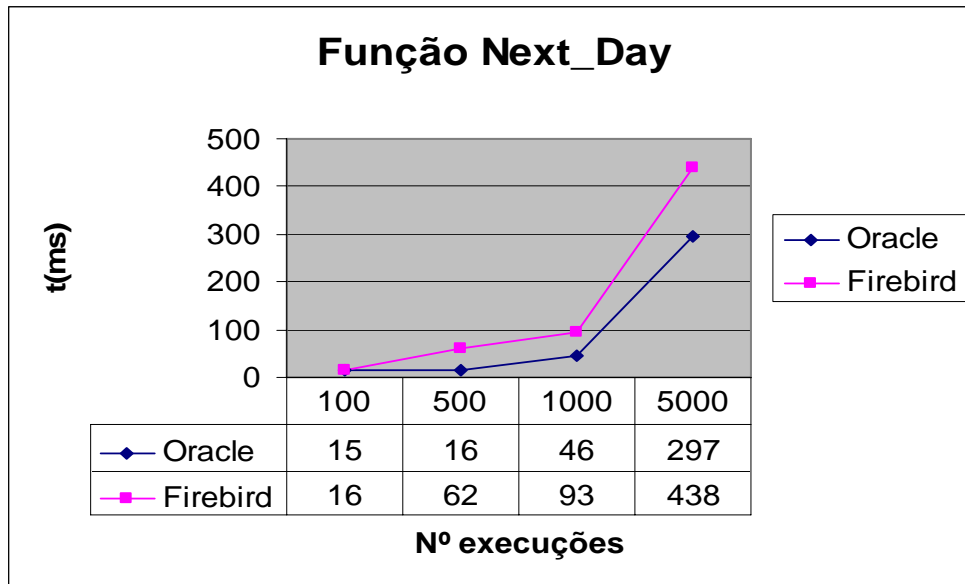


Figura 4.5 - Gráfico com os resultados dos testes da função **NEXT_DAY**

A Tabela 4.6 mostra um resumo do gráfico acima, com o resultado das médias individuais e a média geral obtida nos testes realizados para os dois SGDBs.

Tabela 4.6 - Média dos testes com a função **NEXT_DAY**

SGDB	Média de tempo para 100 execuções	Média de tempo para 500 execuções	Média de tempo para 1000 execuções	Média de tempo para 5000 execuções	Média Geral
Oracle	0,15	0,032	0,046	0,0594	0,07185
Firebird	0,16	0,124	0,093	0,0876	0,11615

Como pode ser visto na Figura 4.5 e na Tabela 4.6 o Oracle obteve melhor desempenho na maioria dos testes realizados perdendo apenas para o teste com 5000 execuções. Para o teste com 100 execuções o Firebird gastou em média 6.66 % de tempo a mais que o Oracle, ficando bem próxima da performance do Oracle. Já para os testes com 500, 1000 e 5000 o desempenho do Oracle foi bem superior ao Firebird que levou em média

287 % (aproximadamente 3 vezes mais) para 500 execuções e 102 % (aproximadamente o dobro do tempo) para 1000 execuções e 47.47 % para 5000 execuções.

Na média geral o Firebird gastou 61.65 % a mais de tempo para realizar todos os testes com essa função.

4.3.5. Função TO_CHAR

O gráfico da Figura 4.6 mostra a comparação da função **TO_CHAR** implementada no Firebird com sua correspondente no Oracle:

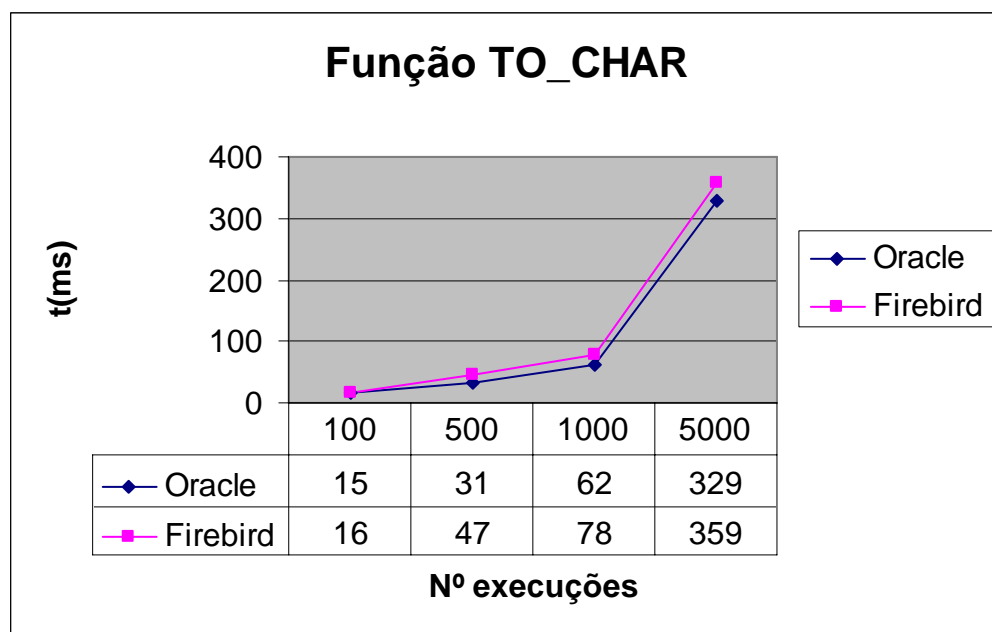


Figura 4.6 - Gráfico com os resultados dos testes da função **TO_CHAR**

A Tabela 4.7 mostra um resumo do gráfico acima, com o resultado das médias individuais e a média geral obtida nos testes realizados para os dois SGDBs.

Tabela 4.7 - Média dos testes com a função **TO_CHAR**

SGDB	Média de tempo para 100 execuções	Média de tempo para 500 execuções	Média de tempo para 1000 execuções	Média de tempo para 5000 execuções	Média Geral
Oracle	0,15	0,062	0,062	0,0658	0,08495
Firebird	0,16	0,094	0,078	0,0718	0,10095

Como pode ser visto na Figura 4.6 e na Tabela 4.7 o Oracle obteve melhor desempenho para todos os testes realizados. Para o teste com 100 execuções o Firebird gastou em média 6.66 % de tempo a mais que o Oracle, ficando bem próxima da performance do Oracle. Já para os testes com 500, 1000, 5000 o desempenho do Oracle foi superior ao Firebird que gastou em média 51.61 %, 25.80 % e 9.11% para os respectivos testes.

Na média geral o Firebird gastou em média 18.83 % a mais de tempo para realizar todos os testes com essa função.

4.3.6. Função TO_DATE

O gráfico da Figura 4.7 mostra a comparação da função **TO_DATE** implementada no Firebird com sua correspondente no Oracle:

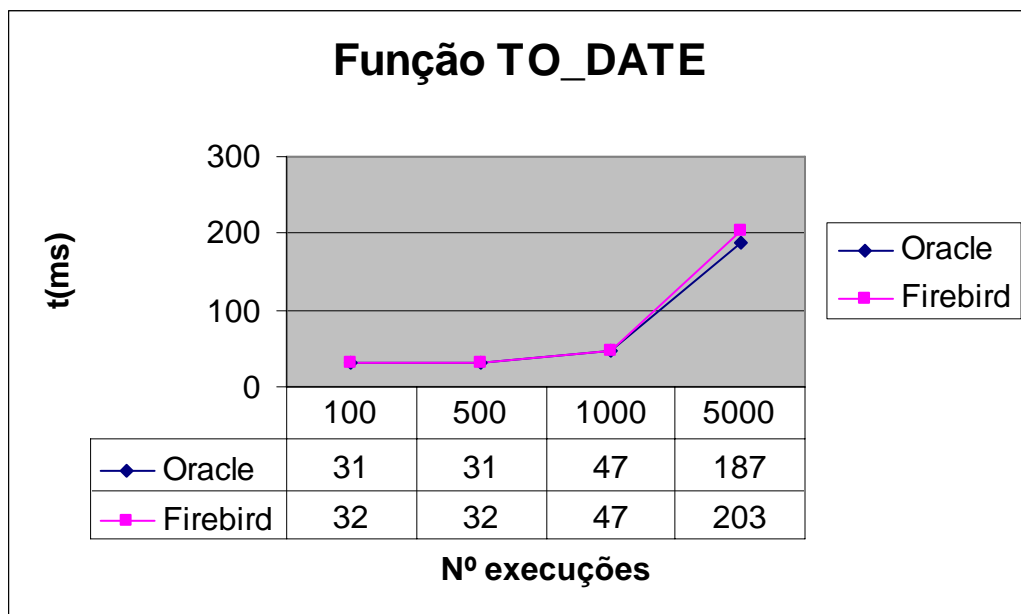


Figura 4.7 - Gráfico com os resultados dos testes da função TO_DATE

A Tabela 4.8 mostra um resumo do gráfico acima, com o resultado das médias individuais e a média geral obtida nos testes realizados para os dois SGDBs.

Tabela 4.8 - Média dos testes com a função TO_DATE

SGDB	Média de tempo para 100 execuções	Média de tempo para 500 execuções	Média de tempo para 1000 execuções	Média de tempo para 5000 execuções	Média Geral
Oracle	0,31	0,062	0,047	0,0374	0,1141
Firebird	0,32	0,064	0,047	0,0406	0,1179

Como pode ser visto na Figura 4.7 e na Tabela 4.8 o Firebird obteve um desempenho próximo ao do Oracle. Para o teste com 100 execuções o Firebird obteve uma performance inferior ao do Oracle, gastando em média 3.22 % de tempo a mais para executar. Para o teste com 1000 execuções o Firebird conseguiu se igualar ao desempenho do Oracle e para os testes com 500 e 5000 o desempenho do Oracle foi um pouco superior ao Firebird que gastou em média 3.22 %, 8.55 % para realizar esses testes.

Na média geral o Firebird gastou em média 3.33 % a mais de tempo para realizar todos os testes com essa função.

4.3.7. Função TO_TIMESTAMP

O gráfico da Figura 4.8 mostra a comparação da função **TO_TIMESTAMP** implementada no Firebird com sua correspondente no Oracle:

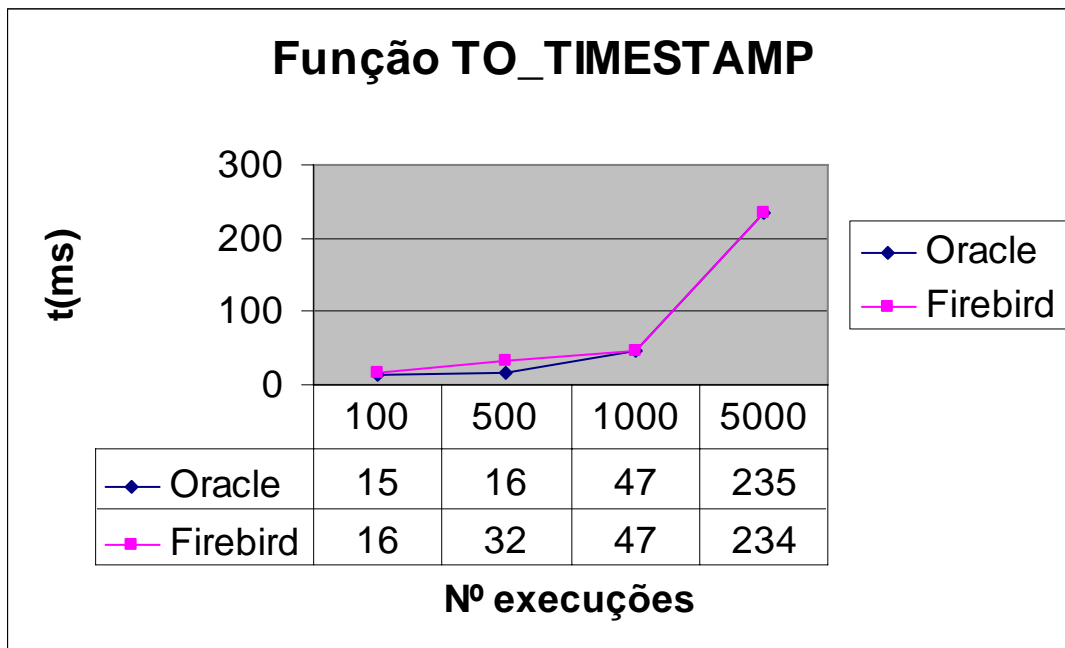


Figura 4.8 - Gráfico com os resultados dos testes da função TO_TIMESTAMP

A Tabela 4.9 mostra um resumo do gráfico acima, com o resultado das médias individuais e a média geral obtida nos testes realizados para os dois SGDBs.

Tabela 4.9 - Média dos testes com a função TO_TIMESTAMP

SGDB	Média de tempo para 100 execuções	Média de tempo para 500 execuções	Média de tempo para 1000 execuções	Média de tempo para 5000 execuções	Média Geral
Oracle	0,15	0,032	0,047	0,047	0,069
Firebird	0,16	0,064	0,047	0,0468	0,07945

Como pode ser visto na Figura 4.8 e na Tabela 4.9 o Oracle obteve, em geral, um desempenho melhor que o do Firebird para a maioria dos casos. Para o teste com 100 execuções o Firebird gastou em média 6.66 % de tempo a mais que o Oracle, ficando bem próxima da performance do Oracle. Para o teste com 1000 execuções o Firebird conseguiu se igualar ao desempenho do Oracle e para os testes com 500 e 5000 o desempenho do Oracle foi superior ao Firebird que gastou em média 100 % (o dobro do tempo), 8.55 % para realizar esses testes.

Na média geral o Firebird gastou 15.14 % a mais de tempo para realizar todos os testes com essa função.

4.3.8. Função ROUND

O gráfico da Figura 4.9 mostra a comparação da função **ROUND** implementada no Firebird com sua correspondente no Oracle:

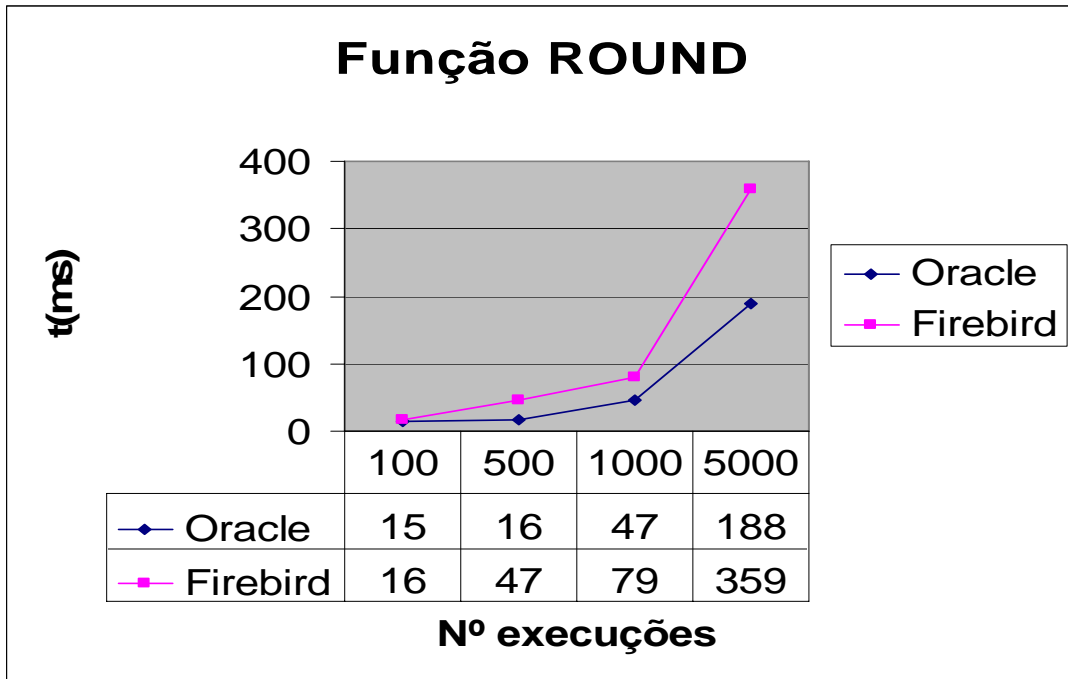


Figura 4.9 - Gráfico com os resultados dos testes da função ROUND

A Tabela 4.10 mostra um resumo do gráfico acima, com o resultado das médias individuais e a média geral obtida nos testes realizados para os dois SGDBs.

Tabela 4.10 - Média dos testes com a função ROUND

SGDB	Média de tempo para 100 execuções	Média de tempo para 500 execuções	Média de tempo para 1000 execuções	Média de tempo para 5000 execuções	Média Geral
Oracle	0,15	0,032	0,047	0,0376	0,06665
Firebird	0,16	0,094	0,079	0,0718	0,1012

Como pode ser visto na Figura 4.9 e na Tabela 4.10 o Oracle obteve um desempenho superior que o do Firebird para a maioria dos casos. Para o teste com 100 execuções o Firebird gastou em média 6.66 % de tempo a mais que o Oracle. Já para os demais testes o Oracle manteve o desempenho bem superior ao Firebird que em média gastou 193,75 %

(aproximadamente 2 vezes mais) para 500 execuções, 68,08 % para 1000 e 90,95% (aproximadamente o dobro do tempo) para 5000.

Na média geral o Firebird gastou 51.83 % a mais de tempo para realizar todos os testes com essa função.

4.3.9. Função TRUNC

O gráfico da Figura 4.10 mostra a comparação da função **TRUNC** implementada no Firebird com sua correspondente no Oracle:

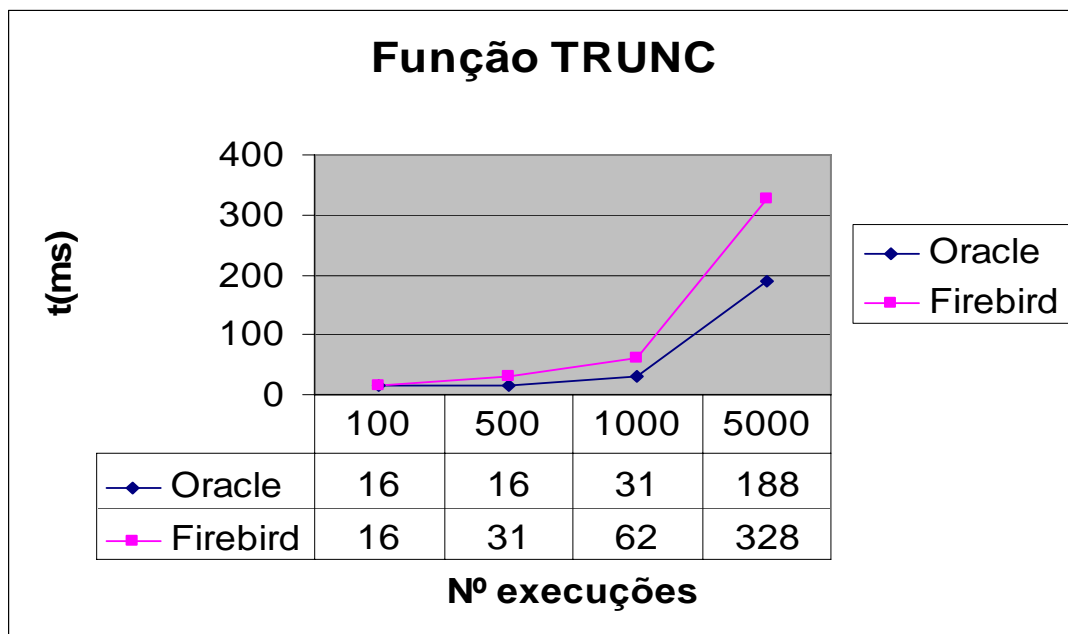


Figura 4.10 - Gráfico com os resultados dos testes da função TRUNC

A Tabela 4.11 mostra um resumo do gráfico acima, com o resultado das médias individuais e a média geral obtida nos testes realizados para os dois SGDBs.

Tabela 4.11 - Média dos testes com a função TRUNC

SGDB	Média de tempo para 100 execuções	Média de tempo para 500 execuções	Média de tempo para 1000 execuções	Média de tempo para 5000 execuções	Média Geral
Oracle	0,16	0,032	0,031	0,0376	0,06515
Firebird	0,16	0,062	0,062	0,0656	0,0874

Como pode ser visto na Figura 4.10 e na Tabela 4.11 o Oracle obteve um desempenho superior ao do Firebird para a maioria dos testes. Para o teste com 100 execuções o Firebird obteve um bom desempenho se igualando ao Oracle. Já para os testes com 500 e 1000 execuções o Firebird gastou aproximadamente o dobro do tempo para executá-los, com média de 93,75 % para 500 e 100 % para 1000 execuções (aproximadamente o dobro do tempo).

Na média geral o Firebird gastou 34,15 % a mais de tempo para realizar todos os testes com essa função.

4.3.10. Análise geral

A Tabela 4.12 mostra a média geral obtida com os testes realizados para cada função e a média do total geral através da razão entre a soma de todas as médias gerais do Firebird com as do Oracle.

Tabela 4.12 – Média geral das funções nos SGDBs Oracle e Firebird

Função	Oracle	Firebird
ADD_MONTHS	0,0674	0,10095
LAST_DAY	0,06185	0,08125
MONTHS_BETWEEN	0,0659	0,0774
NEXT_DAY	0,07185	0,11615
TO_CHAR	0,08415	0,10095
TO_DATE	0,1141	0,1179
TO_TIMESTAMP	0,069	0,07945
ROUND	0,06665	0,1012
TRUNC	0,06515	0,0874
Média Geral	0,074094	0,09585

Com pode ser visto na Tabela 4.12, o Oracle obteve desempenho superior ao do Firebird para todas funções utilizadas nos testes. No geral o Firebird gastou em média 29.36% a mais de tempo para executar os mesmos testes realizados no Oracle. Isso pode ter ocorrido pela forma com que o SGDB do Firebird aloca os recursos utilizados.

5. Conclusões

Pelos resultados obtidos no Capítulo 4, verifica-se que o SGDB Oracle apesar de ser considerado um SGDB “pesado”, devido a todos recursos que oferece, suas regras de integridade e segurança, obteve desempenho superior comparado ao desempenho do SGDB Firebird que é considerado bem mais “leve”.

O Firebird apresentou desempenho inferior, visto que não possuía as funções utilizadas nos testes. As funções foram implementadas através de *Stored Procedures*, sendo que esses tipos de procedimentos não são nativos do banco de dados, o que contribuiu para uma grande perda de performance na hora de executá-los.

A linguagem utilizada na implementação das *Stored Procedures* é uma linguagem de 4ª geração, como a linguagem SQL. Esse tipo de linguagem não pode ser traduzido (compilada) para uma linguagem de máquina. Passam por um outro processo de tradução bem mais lento que uma tradução comum e não podem ser executadas diretamente pelo sistema operacional são executadas dentro de seu sistema gerenciador, portanto acaba afetando o desempenho das funções.

Uma outra forma de implementar funções no Firebird é com uso de UDF, recurso que permite escrever funções para serem utilizadas dentro do banco de dados. Mesmo com o uso de UDFs não se consegue bom desempenho, pois são armazenadas em bibliotecas externas ao banco de dados.

As vantagens de se implementar as funções com SPs mesmo com baixo desempenho são por essas serem independentes de sistema operacional, e podem ser facilmente adaptadas para outros SGDBs.

Esse trabalho tem uma boa contribuição, principalmente para os SGDBs gratuitos, visto que foi possível implementar um dos recursos do SGDB Oracle para o Firebird, um SGDB gratuito com poucos recursos.

A continuação desse trabalho pode ser feita por meio dos recursos que foram implementados no Firebird também serem implementados em outros SGDBs.

Referências Bibliográficas

CANTU, Carlos H.; Firebird Essencial. Rio de Janeiro: Editora Ciência Moderna, 2006.

DATE, C.J. Introdução a sistemas de banco de dados. Editora Campus, 2004.

ELMASRI, Ramez E.; NAVATHE Shamkant. Sistemas de Banco de dados. Editora Person Addison Wesley, 2005.

FIREBIRD. Notas da Versão v.1.5 08 Fev. 2004. Disponível em: www.ibphoenix.com/downloads/Firebird_v15.108_ReleaseNotesPortugBrasil.pdf. Acessado em 16/09/2006.

HARTEL, Anderson R. Apostila de Firebird 1.0. Disponível em: http://www.comunidade-firebird.org/cflp/downloads/CFLP_O012.PDF . Acessado em 14/09/2006.

KORTH, Henry F; SILBERCHATZ, Abraham. Sistemas de Banco de Dados. Editora Makrom Books, 1995.

ORACLE. Oracle Database SQL Reference. Disponível em: http://download-east.oracle.com/docs/cd/B19306_01/install.102/b14318.pdf .Acessado em 20/09/2006.

SCOTT, Urman. Oracle 9i: programação PL/SQL. Editora Elsevier, 2002.

TODD, Bill. Introdução a *Stored Procedures* e Triggers no Firebird. San Diego 2000. Disponível em: http://www.comunidade-firebird.org/cflp/downloads/CFLP_T003.PDF. Acessado em 19/10/2006.

WIKIPEDIA. Banco de Dados. Disponível em: http://pt.wikipedia.org/wiki/Banco_de_dados. Acessado em 11/10/2006.

Apêndices

A. Instalação e configuração do Firebird 1.5

A instalação do SGDB Firebird 1.5 é considerada simples se comparada à instalação de outros SGDBs. Para começar o processo de instalação basta executar o instalador, que pode ser obtido no site www.firebirdsql.com. Após a execução do instalador aparecerá a tela inicial da instalação, reproduzida na Figura A.1.



Figura A.1 - Tela Inicial de instalação do Firebird

A próxima tela informa sobre a licença. O usuário deve concordar com a licença para continuar a instalação. Se o usuário concordar a tela para escolher o tipo de instalação será exibida (Figura A.2).

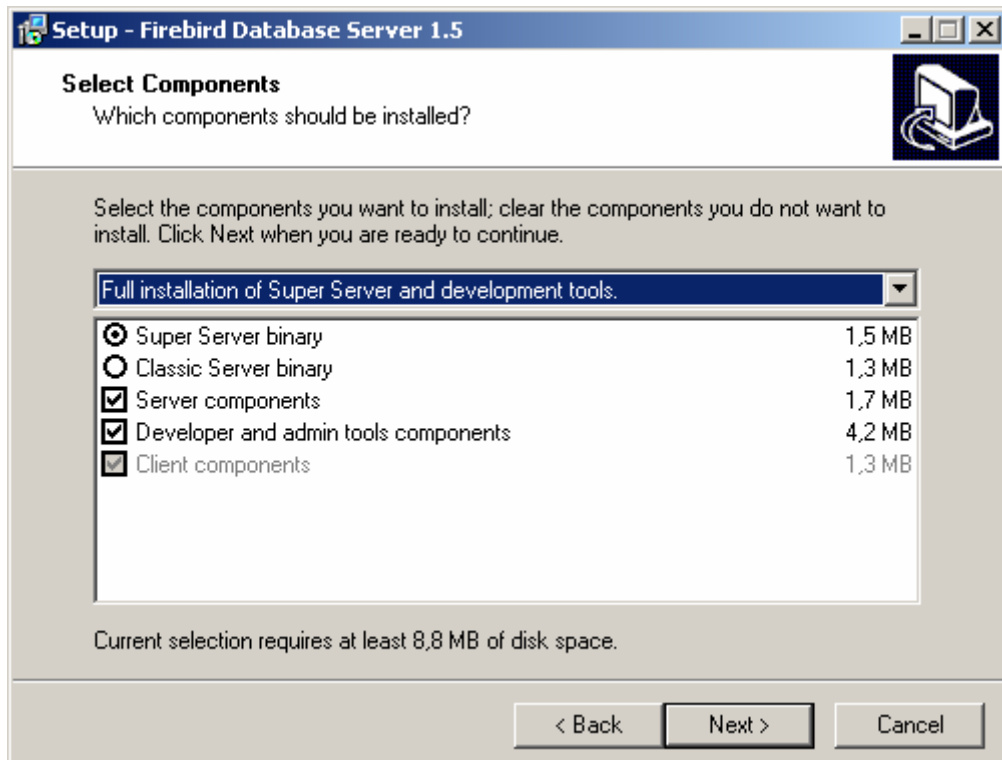


Figura A.2 - Tela de escolha do servidor Firebird

O servidor Firebird possui duas versões: a *Classic* e a *SuperServer*. A principal diferença entre as duas é que a *Classic* cria um novo processo para cada conexão e cada conexão com sua própria região de memória, sendo melhor opção para computadores com múltiplos processadores. A versão *SuperServer* cria um único processo do servidor e cria *threads*² para cada conexão e faz o compartilhamento das regiões de memória, sendo melhor opção para computadores com único processador (CANTU, 2005).

A próxima tela a ser exibida é para informar como o Firebird será inicializado, conforme pode ser observado na Figura A.3.

² *thread* é um tipo de processo ‘leve’, isto é, a sobrecarga causado pelo escalonamento de *thread* é muito menor do que o escalonamento de um processo comum.

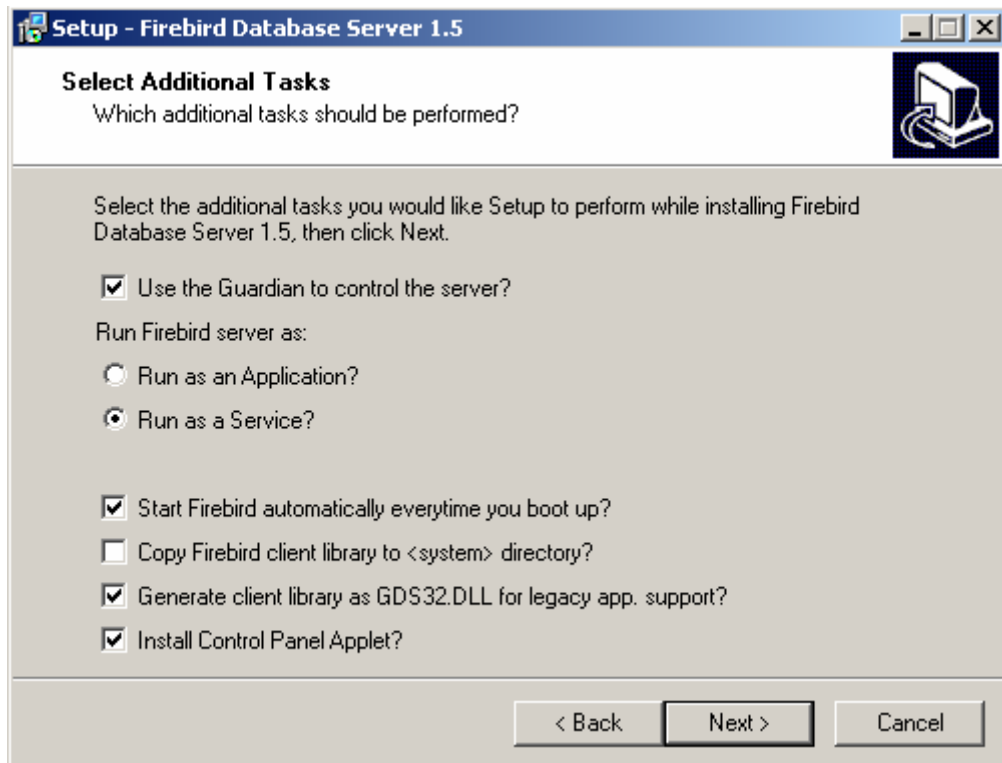


Figura A.3 - Tela de configuração da inicialização do Servidor

Nessa etapa da instalação são mostradas algumas opções de configuração do servidor Firebird:

- **Use de Guardian to control the server:** se marcar essa opção o aplicativo chamado Guardian vai reiniciar automaticamente o servidor Firebird caso o servidor pare por algum problema.
- **Run as na Application:** vai inicializar o servidor como uma aplicação. Utilizado para as versões mais inferiores do *Windows* (98, ME).
- **Run as a service:** vai inicializar o servidor como um serviço.
- **Start Firebird automatically everytime you boot up:** se marcar essa opção o servidor Firebird será inicializado automaticamente com o *Windows*.
- **Copy Firebird client library to <system> directory:** se marcar essa opção as DLLs (gds32.dll ou fbclient.dll) serão copiadas para o diretório do sistema do *Windows*.

- **Generate client library as GDS32.DLL for legacy app. Support:** se marcar essa opção o instalador vai gerar a *DLL* cliente com o nome de *gds32.dll* por questão de compatibilidade com aplicações que usam a *gds32.dll*. Para executar aplicações com Firebird 1.5 recomenda-se não marcar essa opção para que as aplicações não usem a *gds32.dll* e sim a *fbclient.dll*.
- **Install Control Panel Applet:** se marcar essa opção o instalador criará um ícone do controlador do servidor Firebird no painel de controle do *Windows*.

Depois de escolher as opções desejadas será iniciada a cópia dos arquivos de instalação. Se não ocorrer nenhum problema na instalação aparecerá a seguinte tela reproduzida na Figura A.4.

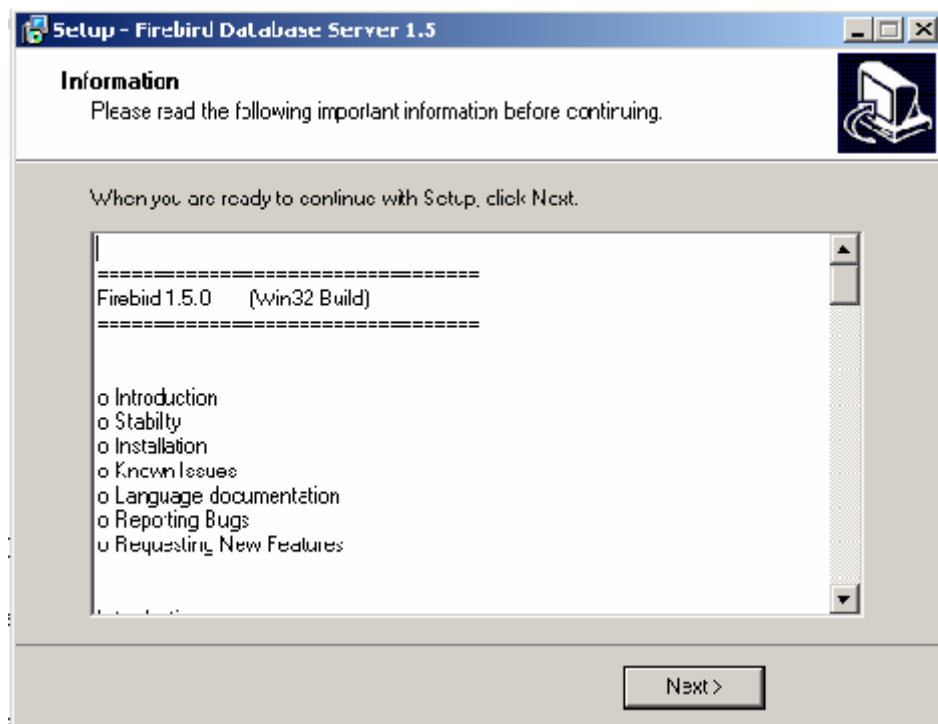


Figura A.4 - Tela de informação sobre a instalação do Firebird

Após terminado o processo de instalação basta fazer a conexão com uma base de dados para utilizar o Firebird. Para conectar uma base de dados no Firebird por meio do IBExpert é preciso primeiro registrar uma base de dados existente. Para registrar uma base de

dados no IBExpert deve-se clicar em *Database* e depois *Register Database*. Ao executar essas opções abrirá a tela apresentada na Figura A.5.

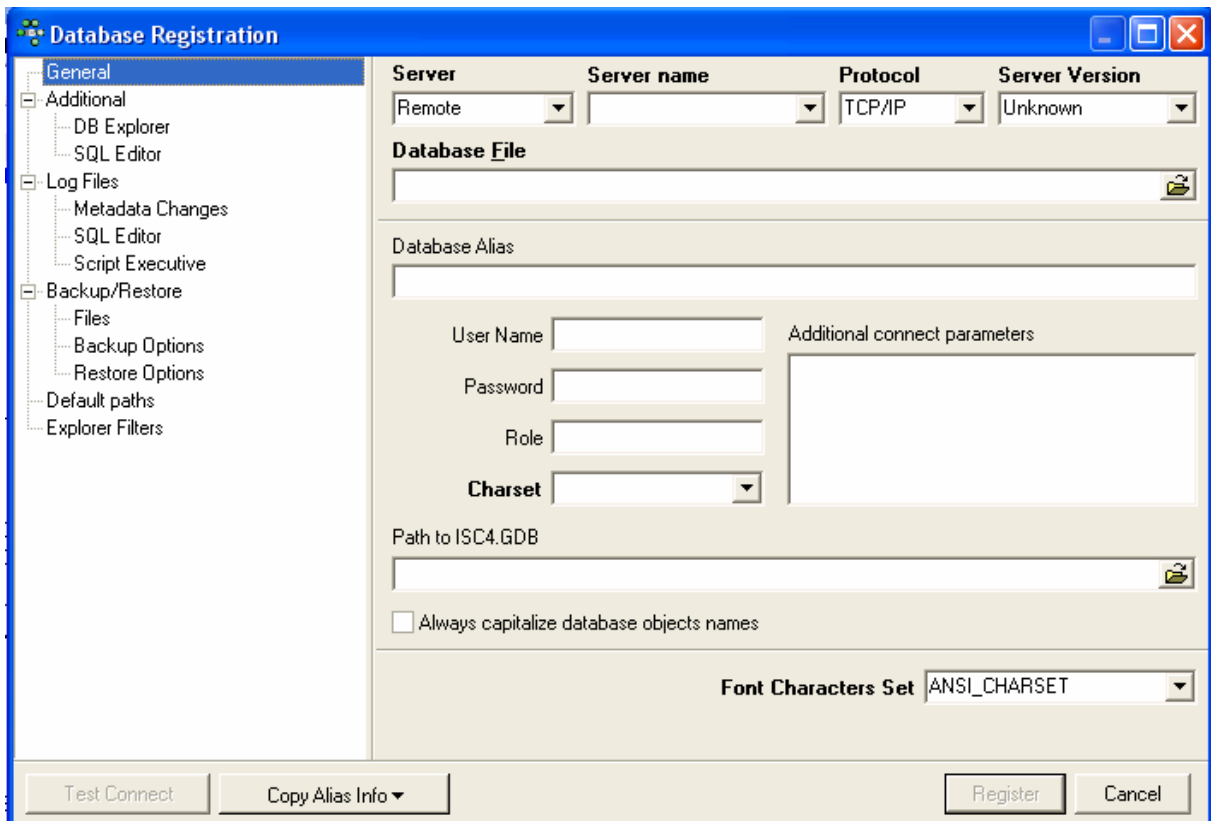
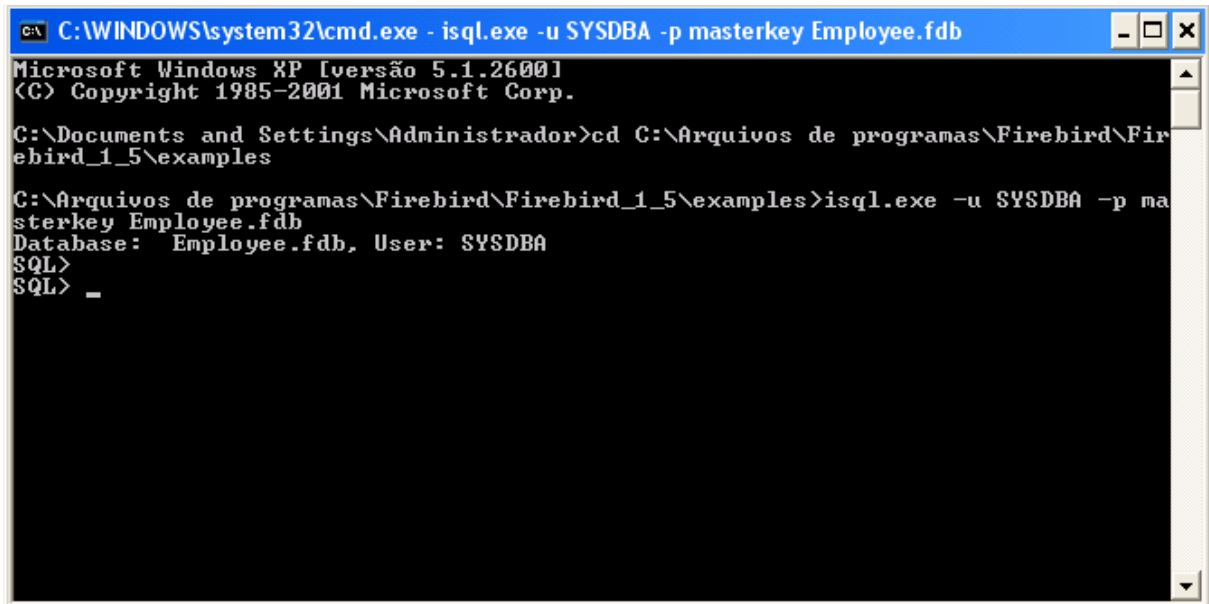


Figura A.5 - Tela do IBExpert para registrar uma base de dados

Essa tela é utilizada para que o usuário faça o registro da base de dados propriamente dita. Alguns campos nessa tela são de preenchimentos obrigatórios para que a conexão entre servidor e o banco seja realizada. Os campos obrigatórios são: *Server* (identifica se vai ser uma conexão local ou remota), *Server name* (identifica o nome do servidor), *Server Version* (identifica a versão do Firebird) e o campo *Database File* (identifica o local da base de dados no disco). Os campos *User name* e *Password* não são obrigatórios, mas serão requisitados toda vez em que houver uma tentativa de conexão.

O Firebird também possui um utilitário padrão para se conectar a uma base de dados, o isql (*Interactive SQL*). Esse utilitário não possui os mesmos recursos do IBExpert. Sua forma de interação com o usuário é através do *prompt* de comandos, não apresentando nenhum dos recursos visuais como IBExpert. No entanto essa característica não quer dizer

que seja inferior, pois o seu objetivo é ser uma ferramenta para simplificar a manipulação de uma base de dados. Na Figura A.6 é apresentado um exemplo de como conectar uma base de dados através do isql:



```
C:\WINDOWS\system32\cmd.exe - isql.exe -u SYSDBA -p masterkey Employee.fdb
Microsoft Windows XP [versão 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrador>cd C:\Arquivos de programas\Firebird\Firebird_1_5\examples

C:\Arquivos de programas\Firebird\Firebird_1_5\examples>isql.exe -u SYSDBA -p masterkey Employee.fdb
Database: Employee.fdb, User: SYSDBA
SQL>
SQL> _
```

Figura A.6 - Tela de exemplo de conexão com uma base a dados usando o isql

Os parâmetros passados para o isql nesse exemplo são: [-u] para identificar o usuário que está conectando ao banco, [-p] para a senha do usuário e o caminho (*path*) da base de dados no disco.

B. Instalação e configuração do Oracle 9i

A configuração do SGDB Oracle pode ser feita no momento da sua instalação. Os requisitos mínimos para efetuar a instalação do Oracle 9i é uma máquina com no mínimo 3 ou 5 GB de espaço livre em disco dependendo do sistema de arquivos e recomenda-se 256 MB no mínimo de memória *RAM*.

Ao começar o processo de instalação do Oracle 9i aparecerá a tela reproduzida na Figura B.1.

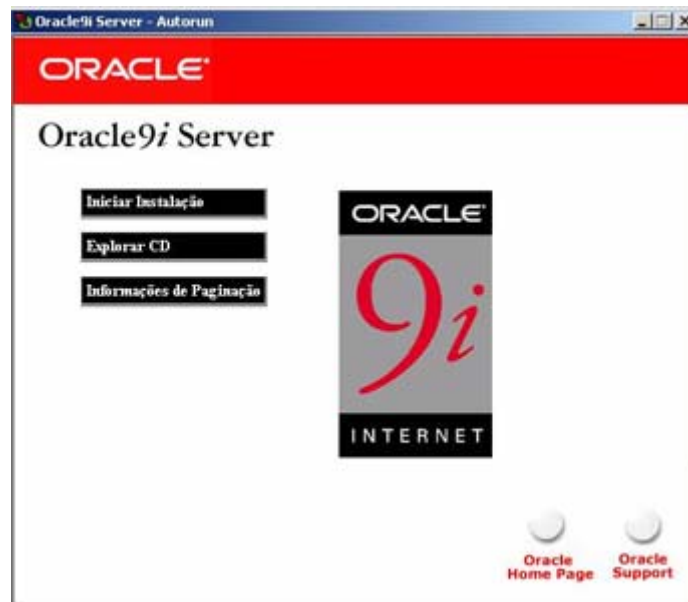


Figura B.1 - Tela Inicial da instalação do Oracle 9i

Em seguida, será disponibilizada uma tela mostrando que a instalação está sendo carregada e logo após aparecerá apresentada na Figura B.2.



Figura B.2 - Tela para verificação dos produtos Oracle

Nessa tela o usuário tem a opção de visualizar quais os produtos Oracle estão instalados na máquina. Posteriormente o usuário pode optar por instalar algum outro componente que não tenha sido instalado ou mesmo desinstalar algum produto, bastando apenas recomeçar o processo de instalação e escolher os produtos nessa tela.

Em seguida aparecerá uma nova tela (Figura B.3) onde usuário vai escolher o caminho dos arquivos de instalação e o diretório de destino.

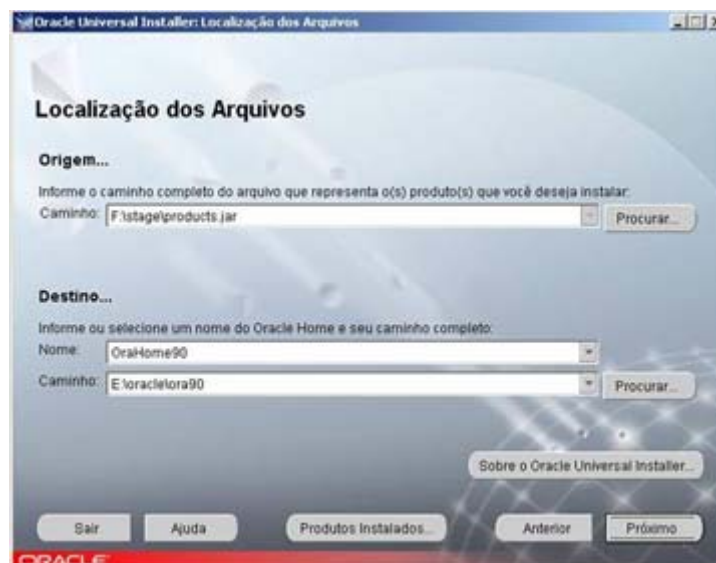


Figura B.3 - Tela de definição do Oracle Home

Nessa etapa da instalação é definido o nome referente à instância e o caminho (local). Se não existir nenhuma instância do Oracle na máquina, pode-se manter o nome e o caminho que foi sugerido pelo instalador.

A próxima etapa da instalação é para a escolha de uma das três opções mostradas na Figura B.4.



Figura B.4 - Tela para escolha do tipo de produto

Para esse trabalho foi escolhida a primeira opção que já define um banco de dados com as configurações iniciais e diversas ferramentas para gerenciamento.

Na etapa seguinte (Figura B.5) o usuário deverá escolher o tipo de instalação de acordo com suas necessidades. Se nenhuma das três primeiras opções atenderem as necessidades do usuário, esse poderá optar pela instalação personalizada.



Figura B.5 - Tela para escolha do tipo da instalação

Na próxima etapa o usuário deverá escolher que tipo de configuração deseja para o banco de dados, se vai ser utilizado para uso geral, para uso específico ou ainda a opção para não criar um banco de dados no momento da instalação, conforme é apresentada na Figura B.6.

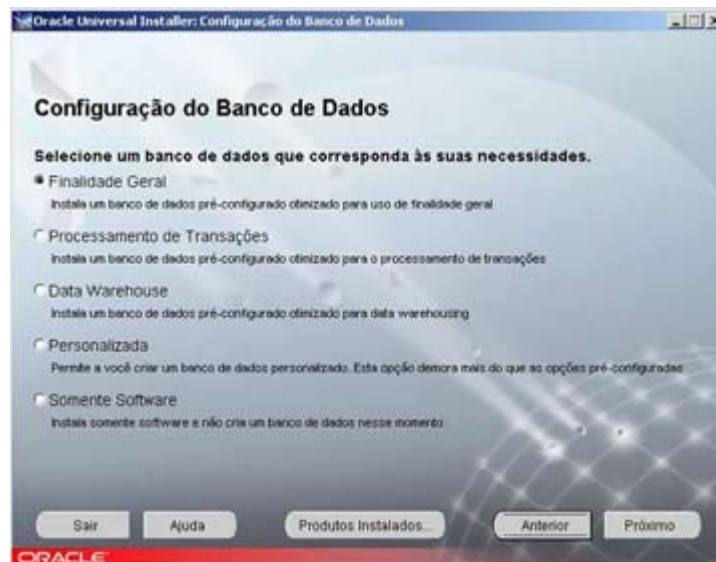


Figura B.6 - Tela de configuração do Banco de dados

Na tela apresentada na Figura B.7 o usuário deve definir o nome do banco de dados e da instância do banco que será criado. O campo SID é o nome da instância do Oracle que deve ser único no mesmo computador, pois servirá para referenciar um banco de dados.



Figura B.7 - Tela para nomeação do banco de dados e o SID

Essa fase da instalação é onde o usuário vai escolher o local de armazenamento físico da base de dados. Um caminho padrão é sugerido para o usuário (Figura B.8).



Figura B.8 - Tela para informar caminho do banco de dados

Na tela apresentada na Figura B.9 o usuário vai definir qual o conjunto de caracteres que o banco de dados vai manipular. Depois da escolha do conjunto de caracteres aparecerá uma tela de resumo sobre a instalação (Figura B.10).

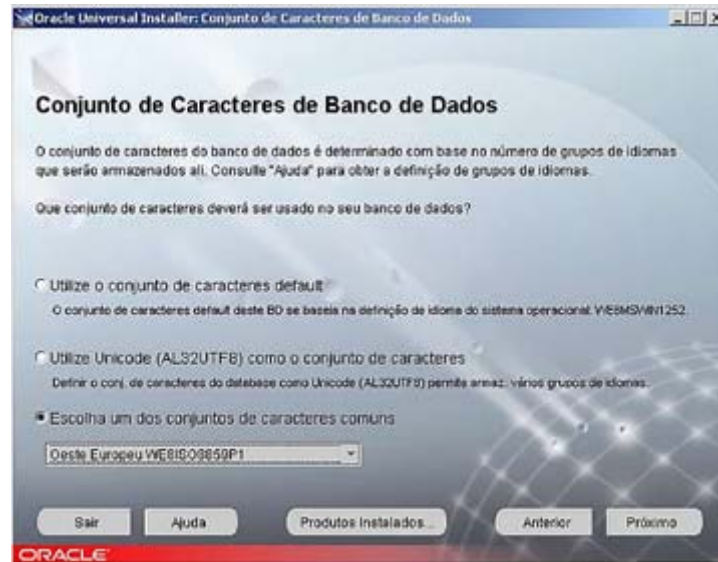


Figura B.9 - Tela para escolher o conjunto de caracteres do banco de dados



Figura B.10 - Tela de resumo da instalação do Oracle

Após terminar o processo de cópia dos arquivos que serão instalados aparecerá a seguinte tela reproduzida na Figura B.11 para configuração.

Nessa tela o usuário pode mudar a senha do usuário SYS e do usuário SYSTEM. É recomendável trocar a senha neste momento para obter maior segurança no acesso ao banco de dados.

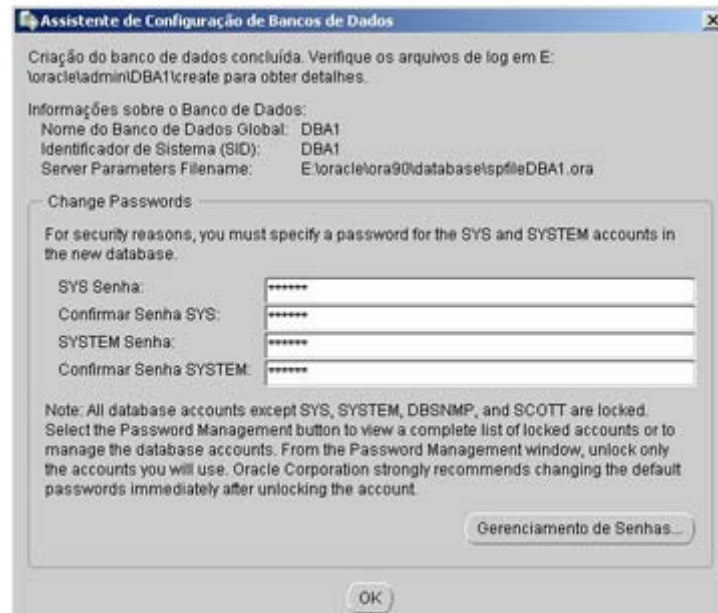


Figura B.11 - Tela de configuração do Oracle

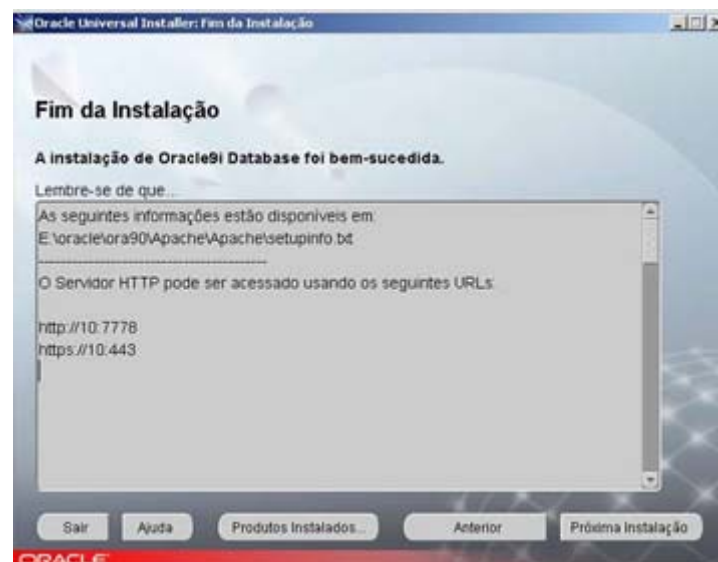


Figura B.12 - Tela final da instalação do Oracle

A tela observada na Figura B.12 indica o final da instalação do Oracle 9i. Também são mostrados nessa tela os dados para conexão do servidor via HTTP (*Hyper Text Transfer Protocol*).