

**Márcio Alexandre Pellis Delarolle**

**Um estudo da viabilidade de otimização de consultas em Data  
Warehouse utilizando Bancos de Dados Distribuídos com  
Replicação Heterogênea**

Trabalho de conclusão de curso apresentado ao  
Instituto Federal de São Paulo, como parte dos  
requisitos para a obtenção do grau de Tecnólogo  
em Sistemas para Internet.

Área de Concentração: Banco de Dados

Orientador: Prof. Gustavo Aurélio Prieto  
Co-orientador: Prof. Breno Lisi Romano

São João da Boa Vista

2014

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Ficha catalográfica preparada pela Seção de Tratamento  
da Informação do Serviço de Biblioteca – IFSP

Delarolle, Márcio.  
Um estudo da viabilidade de otimização de consultas em  
Data Warehouse utilizando Bancos de Dados Distribuídos  
com Replicação Heterogênea. / Márcio Alexandre Pellis  
Delarolle ; orientador: Gustavo Aurélio Prieto;  
co-orientador: Breno Lisi Romano. São João da Boa Vista,  
2014.

Trabalho de Conclusão de Curso, IFSP, 2014.

1. Banco de Dados Distribuídos. 2. Replicação.
3. Data Warehouse. 4. Otimização de Consultas.

I. Um estudo da viabilidade de otimização de  
consultas em Data Warehouse utilizando Bancos de  
Dados Distribuídos com Replicação Heterogênea.

# **AGRADECIMENTOS**

Primeiramente agradeço a minha querida mulher, Michele, que sempre me deu conselhos, força, coragem e incentivo para chegar ao fim desta jornada.

Agradeço à minha filha, Metzli, em razão das noites de ausência durante este longo percurso, obrigado pela compreensão, a minha irmã Stella Delarolle e a Kamel Bensebaa.

Ao meu orientador, professor Gustavo A. Prieto e co-orientador, professor Breno L. Romano, que acreditaram e confiaram em minhas considerações, compartilhando suas ideias, conhecimento e experiências, sempre me motivando. Expresso o meu reconhecimento e admiração pela competência profissional e pela forma que conduziram minha orientação em todas as etapas para conclusão e desenvolvimento deste trabalho.

Agradeço a todos os professores, que foram muito importantes na minha vida acadêmica.

Por fim agradeço aos meus amigos e colegas, que mesmo seguindo caminhos diferentes, sempre se fizeram presentes através de sua experiência e vivência, que foram essenciais neste percurso onde nos ajudamos mutuamente.



# RESUMO

DELAROLLE, MÁRCIO. (2014). **Um estudo da viabilidade de otimização de consultas em Data Warehouse utilizando Bancos de Dados Distribuídos com Replicação Heterogênea.** Trabalho de Conclusão de Curso - Instituto Federal de São Paulo, São João da Boa Vista, 2014.

Este trabalho aborda a otimização de consulta aos dados em *Data Warehouse*, concebendo um ambiente com bancos de dados distribuídos distintos, utilizando replicação heterogênea. Este trabalho, apresenta a otimização do tempo de consulta aos dados integrados, por meio de visões materializadas, para que estas informações, essenciais em um processo de tomada de decisão, sejam visualizadas de forma ágil e concisa, melhorando o desempenho de acesso aos dados para auxiliar os processos de tomada de decisão de forma rápida e segura, com estimativa de benefícios e custos associados ao *Data Warehouse*. Pode-se concluir com os resultados obtidos neste trabalho que a otimização de consulta aos dados, através do uso de banco de dados distribuídos e replicação heterogênea, não só otimiza o tempo de consulta de acesso aos dados mas também a performance do Banco de Dados, tornando-se uma opção viável para implementação de um *Data Warehouse*.

Palavras-chave: Banco de Dados Distribuídos. Replicação Heterogênea. *Data Warehouse*. Otimização de Consultas.



# **ABSTRACT**

**DELAROLLE, MÁRCIO. (2014). A viability study of optimization consultations Data Warehouse using the database with distributed heterogeneous replication. Course Conclusion Project – Instituto Federal de São Paulo, São João da Boa Vista, 2014.**

This work deals the consultation optimization of Data Warehouse, designing a database environment separate distributed data using heterogeneous replication. In this work, we present the optimization of consulting time to integrated data through materialized views for this essential information in decision process of making is viewed quickly and concisely, improving the performance of data access for assist the decision-making process quickly and safely, with the estimated benefits and costs associated with Data Warehouse. It can be concluded with the results obtained in this work that the consultation optimization of data, through the use of distributed database and heterogeneous replication, not only optimizes the query time data access but also the performance of the database, becoming a viable option for implementation of a Data Warehouse.

**Keywords:** Distributed Database. Heterogeneous Replication. Data Warehouse. Consultation Optimization.





# LISTA DE FIGURAS

Figura 1 – Restrição, projeção e junção (exemplos) .....	25
Figura 2 – Exemplo de Integração .....	39
Figura 3 – Estatísticas de tempo de execução .....	40
Figura 4 – Figura metodológica .....	43
Figura 5 – Ambiente de Análise.....	45
Figura 6 – <i>Script</i> de criação da Visão Materializada da tabela “pessoa” .....	46
Figura 7 – <i>Script</i> de criação da Visão Materializada da tabela “cidade” .....	46
Figura 8 – <i>Script</i> da consulta S1 (Experimento I).....	48
Figura 9 – Resultado do <i>Script</i> S1 (Experimento I) .....	48
Figura 10 – <i>Script da consulta</i> S2 (Experimento I).....	49
Figura 11 – Resultado do <i>Script</i> S2 (Experimento I).....	50
Figura 12 – Resultado do <i>Script</i> S1 (Experimento II).....	51
Figura 13 – Resultado do <i>Script</i> S2 (Experimento II).....	51
Figura 14 – <i>Script</i> da consulta S3 (Experimento III) .....	52
Figura 15 – Resultado do <i>Script</i> S3 (Experimento III) .....	53
Figura 16 – <i>Script</i> da consulta S4 (Experimento III) .....	54
Figura 17 – Resultado do <i>Script</i> S4 (Experimento III) .....	54
Figura 18 – Análise Estatística dos <i>Scripts</i> S1 e S2 .....	58
Figura 19 – Análise Estatística dos <i>Scripts</i> S3 e S4 .....	59



# LISTA DE TABELAS

Tabela 1 - Atributos desejáveis para Dados .....	22
Tabela 2 - Vantagens de utilizar Bancos de Dados .....	22
Tabela 3 - Exemplo de Tabela.....	24
Tabela 4 - Comparação do MySQL com outros produtos de SGBDR.....	35
Tabela 5 - Comparação de implementações de SQL .....	37
Tabela 6 - Conjunto de estatísticas padrão - PL/SQL .....	40
Tabela 7 - Conjunto de estatísticas analisadas .....	44
Tabela 8 – <i>Automatic statistics</i> - <i>Script S1</i> (Experimento I) .....	49
Tabela 9 – <i>Automatic statistics</i> - <i>Script S2</i> (Experimento I) .....	50
Tabela 10 – <i>Automatic statistics</i> - <i>Script S1</i> (Experimento II).....	51
Tabela 11 – <i>Automatic statistics</i> - <i>Script S2</i> (Experimento II) .....	51
Tabela 12 – <i>Automatic statistics</i> - <i>Script S3</i> (Experimento III) .....	53
Tabela 13 – <i>Automatic statistics</i> - <i>Script S4</i> (Experimento III) .....	55
Tabela 14 - Comparação Estatística dos <i>Scripts S1</i> e <i>S2</i> .....	57



# LISTA DE SIGLAS

BD	Banco de Dados
BDD	Banco de Dados Distribuído
DBLink	<i>DataBase Link</i> (Links de Banco de Dados)
DDL	Linguagem de Definição de Dados
DML	Linguagem de Manipulação de Dados
DW	<i>Data Warehouse</i>
PL/SQL	Linguagem Procedural /Linguagem de Consulta Estruturada
SEQUEL	<i>Structured English Query Language</i> (Linguagem de Pesquisa em Inglês Estruturado)
SGBD	Sistema de Gerenciamento de Banco de Dados
SGBDD	Sistema de Gerenciamento de Banco de Dados Distribuído
SGBDO	Sistema Gerenciador de Banco de Dados Orientado a Objeto
SGBDR	Sistema Gerenciador de Banco de Dados Relacional
SQL	<i>Structured Query Language</i> (Linguagem estruturada de consulta)
TCO	<i>Total Cost of Ownership</i> (Custo Total de Propriedade)
VM	Visão Materializada



# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>17</b>
1.1	Motivação .....	17
1.2	Objetivos.....	18
1.2.1	Objetivo Geral.....	18
1.2.2	Objetivos Específicos.....	18
1.3	Organização deste trabalho.....	19
<b>2</b>	<b>LEVANTAMENTO BIBLIOGRÁFICO .....</b>	<b>21</b>
2.1	Características de Banco de Dados.....	21
2.1.1	Vantagens de utilizar Banco de Dados.....	22
2.2	Evolução da Tecnologia de Banco de Dados.....	22
2.3	Bancos de Dados Relacionais .....	24
2.4	Bancos de Dados Distribuídos.....	25
2.4.1	Armazenamento Distribuído dos Dados.....	26
2.5	Visão Geral da Estrutura do Sistema de Banco de Dados.....	27
2.6	Sistema de Gerenciamento de Banco de Dados .....	29
2.7	Oracle .....	31
2.7.1	Serviços Heterogêneos (HS) .....	32
2.7.2	<i>DataBase Links</i> (Links de Banco de Dados).....	32
2.7.3	Visões de Apoio a Decisão.....	33
2.7.4	Materialização de Visão.....	33
2.7.5	PL/SQL ( <i>Procedural Language/Structured Query Language</i> ).....	35
2.8	MySQL.....	35
2.9	SQL .....	36
2.9.1	Histórico.....	36
2.10	<i>Data Warehouse</i> .....	37

2.10.1 Características da Tecnologia de <i>Data Warehouse</i> .....	38
2.11 Performance .....	39
2.12 Trabalhos Relacionados .....	41
<b>3 METODOLOGIA.....</b>	<b>43</b>
3.1 Estudo sobre BDD.....	43
3.2 Definição de critérios para avaliação das ferramentas .....	44
3.3 Projeto do ambiente para estudo de viabilidade .....	44
3.4 Criação dos <i>Databases</i> , Tabelas e VMs.....	45
3.5 Carga nos BD's .....	47
3.6 Análise Estatística .....	47
<b>4 RESULTADOS .....</b>	<b>57</b>
<b>5 CONCLUSÕES.....</b>	<b>61</b>
5.1 Trabalhos Futuros.....	62
<b>REFERÊNCIAS.....</b>	<b>63</b>



# 1 Introdução

A quantidade de informações disponíveis está crescendo exponencialmente e o valor dos dados é completamente reconhecido como um ativo organizacional. Para se obter estes complexos conjuntos de dados, é necessário o uso de ferramentas que simplifiquem as tarefas de gerenciamento de dados e a extração das informações, de forma oportuna (RAMAKRISHNAN, 2008).

O crescente poder de processamento e sofisticação das ferramentas e técnicas resultaram no desenvolvimento do *Data Warehouse* (depósito de dados). Esse poder, continuamente crescente, traz uma grande demanda pela melhoria de desempenho do acesso a dados em banco de dados (ELMASRI; NAVATHE, 2005).

Portanto, este trabalho pretende apresentar uma forma de viabilizar estas informações de maneira segura, rápida e consistente, otimizando o tempo de consulta aos dados de um banco de dados para que possam ser úteis em uma tomada de decisão.

## 1.1 Motivação

Em virtude do uso de tecnologias que tornam mais fácil e mais barato o armazenamento de informações, as organizações vêm acumulando um imenso volume de dados (WATSON, 2004) que tem aumentado a necessidade de acesso e atualização de dados de um conjunto de banco de dados preexistentes, com diferentes ambientes sob os quais esses dados estão armazenados (SILBERSCHATZ *et al.* 1999). Para muitas organizações, o gerenciamento de dados tornou-se uma função importante, compreendendo que podem

aprender muito a partir da análise dos dados que elas coletam como parte de suas operações diárias (WATSON, 2004).

As organizações têm dado cada vez mais ênfase às aplicações nas quais dados atuais e históricos são amplamente analisados, identificando tendências e resumindo os dados para apoiar a tomada de decisões de alto nível (RAMAKRISHNAN, 2008).

## **1.2 Objetivos**

Diante do exposto, apresentam-se nesta seção os objetivos gerais e específicos desta pesquisa.

### **1.2.1 Objetivo Geral**

Conceber um ambiente em que se possa analisar a viabilidade de otimização de consultas em um *Data Warehouse* utilizando Bancos de Dados Distribuídos Heterogêneos. Para tanto, serão replicadas as informações previamente calculadas e o resultado será armazenado em visões materializadas, a fim de se otimizar o tempo de consulta aos dados neste cenário.

### **1.2.2 Objetivos Específicos**

A fim de atingir o objetivo geral, as seguintes etapas se fazem necessárias:

- Realizar um levantamento bibliográfico a fim de pesquisar e apresentar assuntos diretamente relacionados a Bancos de Dados, *Data Warehouse*, Replicação e Performance;
- Conceber um ambiente para viabilizar a análise da otimização de consultas por meio de visões materializadas;
- Aplicar o ambiente concebido por meio de experimentos, visando validar seu funcionamento e apresentar os resultados obtidos;

- Analisar os resultados obtidos.

### ***1.3 Organização deste trabalho***

O trabalho é composto por quatro capítulos além desta introdução. No segundo capítulo apresenta-se o levantamento bibliográfico, onde são descritos os conceitos básicos envolvidos com o tema deste trabalho, elementos e fundamentos sobre Bancos de Dados, *Data Warehouse*, Replicação e Performance, fornecendo base para o desenvolvimento do objetivo proposto. No terceiro capítulo é apresentada a metodologia para o desenvolvimento deste trabalho. No quarto capítulo apresenta-se uma análise dos resultados obtidos com este trabalho. No quinto capítulo é apresentada a conclusão dos resultados obtidos.



## 2 Levantamento Bibliográfico

Nas próximas seções, serão apresentados os conceitos relacionados ao objetivo deste trabalho, fornecendo o embasamento teórico para o entendimento do mesmo. Serão abordados tópicos sobre Bancos de Dados, Replicação, *Data Warehouse* e Otimização de Consultas.

### 2.1 Características de Banco de Dados

Um banco de dados é uma coleção de dados que, tipicamente, descreve as atividades de uma ou mais organizações relacionadas (RAMAKRISHNAN, 2008).

Segundo Silberschatz *et al.* (1999), a tecnologia de banco de dados não melhora apenas as operações diárias das organizações, mas também as decisões que afetam nossas vidas. Os bancos de dados contêm uma enorme quantidade de dados sobre muitos aspectos da nossa vida. Essa tecnologia ajuda a consolidar essa massa de dados e a transformá-la em informação para tomada de decisão.

Segundo Silberschatz *et al.* (1999), apresentam-se abaixo importantes propriedades dos bancos de dados:

- **Persistente**: os dados são armazenados de forma permanente. Pelo fato de armazenar e manter dados ser dispendioso, apenas dados com possibilidades de relevância para a tomada de decisões devem ser armazenados;
- **Compartilhado**: um banco de dados pode ter múltiplas aplicações e usuários;
- **Inter-relacionado**: dados armazenados como unidades separadas podem ser ligados para fornecer um quadro completo. Os bancos de dados contêm tanto

entidades como relacionamento entre entidades. Uma entidade é um grupo de dados (*cluster*) normalmente sobre um único assunto.

Segundo Watson (2004) uma vez percebida a importância dos dados para as organizações, pode-se identificar alguns atributos desejáveis para esses dados, conforme apresentado na Tabela 1.

Tabela 1 – Atributos desejáveis para Dados. Fonte: Watson (2004)

Compartilhamento	Prontamente acessível a mais de uma pessoa ao mesmo tempo
Disponibilidade	Facilmente disponível para um tomador de decisão
Segurança	Protegido contra destruição e uso não autorizado
Precisão	Confiável, armazenado de maneira precisa
Oportunidade	Corrente e atual
Pertinência	Apropriado para decisão

### 2.1.1 Vantagens de utilizar Banco de Dados

De acordo com Suehring (2002), a velocidade, exatidão e inteireza dos bancos de dados os tornam críticos para troca de informações em alta-velocidade. Apresenta-se na Tabela 2 abaixo cada uma das vantagens.

Tabela 2 – Vantagens de utilizar bancos de dados. Fonte: Suehring (2002)

Vantagem	Descrição
Velocidade	O formato significa rápido armazenamento e recuperação das informações. Os usuários e as aplicações dispõem de um meio rápido de ler e gravar dados assincronamente.
Informação	As informações podem ser reunidas, quantificadas e analisadas de maneira personalizada com maior flexibilidade.
Exatidão	Desde que a inserção de dados seja cuidadosa, os bancos de dados fornecem resultados consistentes e exatos com base em seus dados.
Inteireza	Os bancos de dados podem armazenar e informar resultados de maneira tão completa e detalhada quanto seus dados permitirem.

## 2.2 Evolução da Tecnologia de Banco de Dados

Desde os primeiros computadores, armazenar e manipular dados tem sido o principal foco dos aplicativos (RAMAKRISHNAN, 2008).

No mundo comercial atual, os quatro mais importantes tipos de Sistemas de Gerenciamento de Banco de Dados (SGBDs) são os Sistema Gerenciador de Banco de Dados Relacional (SGBDR) e Sistema Gerenciador de Banco de Dados Orientado a Objeto (SGBDO), que se baseiam nos modelos de dados relacional e de objetos, respectivamente. Os dois outros tipos principais de produtos de SGBDs são, atualmente, referenciados como SGBDs legados. Eles são baseados nos modelos de dados hierárquico e de rede que foram introduzidos na metade dos anos 60 (ELMASRI; NAVATHE, 2005).

Segundo Ramakrishnan (2008), a base do modelo de dados de rede, padronizado pela *Conference on Data Systems Languages* (CODASYL), foi o primeiro SGBD de propósito geral, projetado por Charles Bachman, na *General Electric*, sendo chamado de Depósito de Dados Integrado (*Integrated Data Store*). Este SGBD influenciou bastante os Sistemas de Banco de Dados na década de 60. No final da década de 60, a IBM desenvolveu o Sistema de Gerenciamento de Informação (*IMS – Information Management System*), que constituiu a base da estrutura de representação alternativa de dados, chamada modelo de dados hierárquico.

Em 1970, Edgar Codd, do laboratório de Pesquisa de San Jose, da IBM, propôs uma nova estrutura de representação de dados chamada modelos de dados relacional, que veio a ser um marco histórico no desenvolvimento de Sistemas de Banco de Dados. Os SGBDO's foram propostos como uma alternativa aos sistemas relacionais e se destinam aos domínios de aplicação onde os objetos complexos desempenham um papel fundamental. Alguns exemplos de domínios de dados complexos incluem repositórios multimídia e gerenciamento de documentos (RAMAKRISHNAN, 2008).

## 2.3 Bancos de Dados Relacionais

Um Banco de Dados Relacional consiste em uma coleção de tabelas. Uma linha em uma tabela representa um relacionamento entre um conjunto de valores. Uma vez que essa tabela é uma coleção de tais relacionamentos, há uma estreita correspondência entre o conceito de tabela e o conceito matemático de relação, a partir das quais se origina o nome desse modelo de dados (SILBERSCHATZ *et al.*, 1999).

Na terminologia do modelo relacional, uma linha é chamada tupla, um cabeçalho de coluna é denominado atributo e a tabela é chamada relação (ELMASRI; NAVATHE, 2005).

Segundo Watson (2004), uma tabela é uma forma comum de armazenagem de dados da organização. Na Tabela 3, é demonstrado um exemplo de tabela.

Tabela 3 – Exemplo de Tabela. Fonte: Watson (2004)

Lista de Preços	
Produto	Preço
Canivete	4,50
Compasso	10,00
Sistema de geoposicionamento	500,00
Régua da escala para mapas	4,90

Os Sistemas de Banco de Dados Relacionais foram desenvolvidos, originalmente, por causa da familiaridade e simplicidade. Como as tabelas são utilizadas para comunicar ideias em vários campos, a terminologia de tabelas, linhas e colunas é familiar para muitos usuários. A matemática dos bancos de dados relacionais implica a conceitualização de tabelas como conjuntos. A combinação de familiaridade e simplicidade com a base matemática é tão poderosa que os SGBDs relacionais são dominantes em termos comerciais (MANNINO, 2008).

Um conceito importante em um Banco de Dados Relacional é o conceito de atributo chave, que permite identificar e diferenciar uma tupla de outra. Através do uso de chaves é



possível acelerar o acesso a elementos usando índices e estabelecer relacionamentos entre as múltiplas tabelas de um de Banco de Dados Relacional (RICARTE, 2002).

Segundo Date (2003), o modelo relacional é constantemente descrito como tendo os três aspectos a seguir:

- **Estrutural**: os dados são percebidos pelo usuário como tabelas e nada além de tabelas, no banco de dados;
- **Integridade**: essas tabelas satisfazem a certas restrições de integridade;
- **Manipulador**: os operadores disponíveis para que os usuários possam, manipular essas tabelas, são operadores que derivam tabelas a partir de outras tabelas. Três importantes operadores são de restrição, projeção e junção, apresentados na Figura 1.

Restrição:	Resultado:	DEPTO#	NOMEDEPTO	ORÇAMENTO
DEPTOs nos quais ORÇAMENTO > 8M		D1	Marketing	10M
		D2	Desenvolvimento	12M

Projeção:	Resultado:	DEPTO#	ORÇAMENTO
DEPTOs sobre DEPTO#, ORÇAMENTO		D1	10M
		D2	12M
		D3	5M

Junção:					
DEPTOs e EMPs sobre DEPTO#			Resultado:		
DEPTO#	NOMEDEPTO	ORÇAMENTO	EMP#	NOMEEMP	SALÁRIO
D1	Marketing	10M	E1	Lopez	40K
D1	Marketing	10M	E2	Cheng	42K
D2	Desenvolvimento	12M	E3	Finzi	30K
D2	Desenvolvimento	12M	E4	Saito	35K

Figura 1 – Exemplos de Restrição, Projeção e Junção. Fonte: (Date, 2003)

## 2.4 Bancos de Dados Distribuídos

O gerenciamento de bancos de dados distribuídos tem sido proposto por diversas razões que variam desde a descentralização organizacional e a economia de processamento até a maior autonomia (ELMASRI; NAVATHE, 2005).

Nos últimos anos, vem aumentando a necessidade de acesso e atualização de dados de um conjunto de banco de dados preexistentes, com diferentes ambientes sob os quais esses dados estão armazenados. Um sistema de banco de dados múltiplo (*multidatabase*) é uma camada de *software* que oferece tais coleções de banco de dados (SILBERSCHATZ *et al.*, 1999).

Segundo Elmasri e Navathe (2005), um Banco de Dados Distribuído (BDD) é uma coleção de múltiplos bancos de dados logicamente inter-relacionados distribuídos por uma rede de computadores. Já um Sistema de Gerenciamento de Banco de Dados Distribuído (SGBDD) é um sistema de *software* que gerencia um Banco de Dados Distribuído enquanto torna a distribuição transparente para o usuário.

Os bancos de dados distribuídos fornecem controle sobre a multiprogramação de operações que permitem um compartilhamento de recursos sem a possibilidade de invasão de um processo na área de memória de outros, prevenindo os canais secretos<sup>1</sup> orientados por oportunidade (ELMASRI; NAVATHE, 2005).

#### **2.4.1 Armazenamento Distribuído dos Dados**

Segundo Silberschatz *et al.* (1999), há diversos enfoques para o armazenamento dessas relações em um Banco de Dados Distribuído:

- **Replicação**: O sistema mantém réplicas idênticas (cópias) da relação. Cada réplica é armazenada em diferentes sites, resultando na replicação dos dados;
- **Fragmentação**: A relação é particionada em vários segmentos e o sistema mantém diversas réplicas de cada fragmento;
- **Replicação e Fragmentação**: A relação é particionada em vários segmentos.

O sistema mantém diversas réplicas de cada fragmento.

---

<sup>1</sup> Um canal secreto (*covert channel*) permite uma transferência de informação que viola a segurança ou a Política, em um canal de oportunidade a informação é transmitida pela oportunidade de eventos ou processos (ELMASRI; NAVATHE, 2005).

## 2.5 Visão Geral da Estrutura do Sistema de Banco de Dados

Segundo Silberschatz *et al.* (1999), um Sistema de Banco de Dados está dividido em módulos específicos, de modo a atender a todas as funções do sistema.

Os componentes funcionais do Sistema de Banco de Dados podem ser divididos pelos componentes de processamento de consultas e pelos componentes de administração de memória.

Segundo Silberschatz *et al.* (1999), os componentes de processamento de consulta incluem:

- **Compilador DML** (*Data-manipulation Language*): Que traduz comandos DML da linguagem de consulta em instruções de baixo nível, inteligíveis ao componente de execução de consultas;
- **Pré-compilador para comandos DML**: Inseridos em programas de aplicação, que convertem comandos DML em chamadas de procedimentos normais da linguagem hospedeira;
- **Interpretador DDL** (*Data-definition Language*): Interpreta os comandos DDL e registra-os em um conjunto de tabelas que contêm metadados<sup>2</sup>;
- **Componentes para o tratamento de consultas**: Executam instruções de baixo nível geradas pelo compilador DML.

Os componentes para administração do armazenamento de dados proporcionam a interface entre os dados de baixo nível, armazenados no banco de dados, os programas de aplicações e as consultas submetidas ao sistema.

Segundo Silberschatz *et al.* (1999), os componentes de administração de armazenamento de dados incluem:

---

<sup>2</sup> Metadados — são dados que descrevem a estrutura dos dados, suas restrições, suas aplicações, suas autorizações, e assim por diante (ELMASRI; NAVATHE, 2005).

- **Gerenciamento de autorizações e integridade**: Testam o cumprimento das regras de integridade e a permissão ao usuário no acesso dado;
- **Gerenciamento de transações**: Garante que o banco de dados permanecerá em estado consistente no sistema e que transações concorrentes serão executadas sem conflitos em seus procedimentos;
- **Administração de arquivos**: Gerencia a alocação de espaço no armazenamento em disco e as estruturas de dados usadas para representar estas informações armazenadas em disco;
- **Administração de buffer**: Responsável pela intermediação de dados do disco para a memória principal e pela decisão de quais dados colocar em memória *cache*.

Além disso, Silberschatz *et al.* (1999), ainda complementam que algumas estruturas de dados são exigidas como parte da implementação física do sistema:

- **Arquivo de dados**: Armazena o próprio banco de dados;
- **Dicionário de dados**: Armazena os metadados relativos à estrutura do banco de dados. O dicionário de dados é muito usado. Portanto, grande ênfase é dada ao desenvolvimento de um bom projeto com uma implementação eficiente do dicionário;
- **Índices**: Proporcionam acesso rápido aos itens de dados que são associados a valores determinados;
- **Estatística de dados**: Armazenam informações estatísticas relativas aos dados contidos no banco de dados. Essas informações são usadas pelo processador de consultas para seleção de meios eficientes para execução de uma consulta.

## 2.6 Sistema de Gerenciamento de Banco de Dados

Segundo Mannino (2008), um Sistema de Gerenciamento de Banco de Dados (SGBD) é um conjunto de componentes que dão suporte à criação, utilização e à manutenção de banco de dados. Devido às exigências do mercado e à inovação dos produtos, os SGBDs evoluíram e hoje fornecem uma ampla gama de recursos para aquisição, armazenamento, disseminação, manutenção, recuperação e formatação de dados.

Os SGBDs foram desenvolvidos para resolver alguns dos problemas e limitações com que se deparavam os usuários de sistemas baseados em arquivos de dados (DAMAS, 2007).

Segundo Silberschatz *et al.* (1999), obter informações organizacionais em sistemas de processamento de arquivos apresenta numerosas desvantagens:

- **Inconsistência e redundância de dados**: Já que arquivos e aplicações são criados e mantidos por diferentes programadores, é comum que os arquivos possuam formatos diferentes e os programas sejam escritos em diversas linguagens de programação. A mesma informação pode ser repetida em diversos lugares (arquivos). Essa redundância aumenta os custos de armazenamento e acesso. Ainda, pode originar inconsistências de dados;
- **Dificuldade de acesso aos dados**: Um ambiente com sistema de processamento de arquivos convencional não atende às necessidades de recuperação de informações de modo eficiente;
- **Isolamento de dados**: Como os dados estão dispersos em vários arquivos, estes arquivos podem apresentar diferentes formatos;
- **Problemas de integridade**: Os valores dos dados atribuídos e armazenados em um banco de dados devem satisfazer certas restrições para manutenção da consistência. Os programadores determinam o cumprimento dessas restrições por meio da adição de código apropriado aos vários programas de aplicações.

Entretanto, quando aparecem novas restrições, é difícil alterar todos os programas para incrementá-las;

- **Problema de atomicidade**: Um sistema computacional, como qualquer outro dispositivo mecânico ou elétrico, está sujeito a falhas. Em muitas aplicações é crucial assegurar que, uma vez detectada uma falha, os dados sejam salvos em seu último estado consistente, anterior a ela. É difícil garantir essa propriedade em um sistema convencional de processamento de arquivos;
- **Anomalias no acesso concorrente**: Muitos sistemas permitem atualizações simultâneas nos dados para aumento do desempenho do sistema como um todo e para melhores tempos de resposta. Nesses tipos de ambiente, a interação entre atualizações concorrentes pode resultar em inconsistência de dados. Para resguardar-se dessa possibilidade, o sistema deve manter algum tipo de supervisão. Como os dados podem sofrer acesso de diferentes programas, os quais não foram coordenados previamente, a supervisão é bastante dificultada;
- **Problemas de segurança**: Nem todos os usuários de banco de dados estão autorizados ao acesso a todos os dados. Uma vez que os programas de aplicação são inseridos no sistema como um todo, é difícil garantir a efetividade das regras de segurança.

Estas dificuldades, entre outras, provocaram o desenvolvimento dos SGBDs.

Segundo Damas (2007), o objetivo de um SGBD é garantir um ambiente apropriado para acessar e armazenar informações no banco de dados de forma confiável e eficiente.

## 2.7 Oracle

Há quase trinta anos, Larry Ellison vislumbrou uma oportunidade que outras companhias não haviam percebido, encontrando uma descrição de um protótipo funcional de um Banco de Dados Relacional. Ellison e os co-fundadores da Oracle, Bob Miner e Ed Oates, perceberam que havia um potencial de negócios no modelo de Banco de Dados Relacional. A tecnologia Oracle pode ser encontrada em quase todos os setores do mundo inteiro, sendo o principal fornecedor de *software* para gerenciamento de informações e a segunda maior empresa de *software* independente do mundo (ORACLE, 2013).

Oracle é um sistema de Banco de Dados Relacional flexível e extremamente poderoso, junto com esse poder vem a sua complexibilidade (URMAN, 2002).

Atualmente, a Oracle Corporation é a líder em *software* do servidor de banco de dados de nível corporativo sendo amplamente utilizado em vários tipos de aplicações grandes e muito popular, pois possui desvantagens mínimas (SUEHRING, 2002).

A fim de cumprir o objetivo deste trabalho, os seguintes recursos serão utilizados:

- **Servicos Heterogêneos (HS)**: Será utilizado o serviço HS *Generic Connectivity* (Conectividade Genérica) que permite a conexão entre as bases de dados utilizando-se de drives *Open Database Connectivity* (ODBC)<sup>3</sup>;
- **DataBase Links**: Ligação de dados, um ponteiro no local do banco de dados que permitirá acessar objetos em uma base de dados remota;
- **Visões Materializadas**: Serão utilizadas para cálculos, armazenamentos de dados e aumentar a agilidade na troca de informações entre os bancos de dados;

---

<sup>3</sup> *Open Database Connectivity* - Conectividade aberta de banco de dados - (ODBC) é a interface estratégica da Microsoft para acessar dados em um ambiente heterogêneo em sistemas de gerenciamento de banco de dados (MICROSOFT, 2013).

- **PL/SQL (*Procedural Language/Structured Query Language*)**: Importante linguagem, especialmente adequada para funcionar com o Oracle, devido a sua integração com a SQL;
- **Performance**: Por meio da linguagem PL/SQL, será analisado a performance de execução da solicitação aos dados.

### **2.7.1 *Serviços Heterogêneos (HS)***

Os Serviços Heterogêneos (HS) são componentes integrados dentro do Oracle e fornecem a tecnologia genérica para acessar sistemas não-Oracle, a partir do servidor de banco de dados Oracle. Os serviços heterogêneos permitem usar declarações Oracle SQL para acessar, de forma transparente, os dados armazenados em sistemas não-Oracle (ORACLE, 2013).

Para acessar um sistema não-Oracle, torna-se necessário utilizar um agente de serviços heterogêneos complementar. Pode-se conectar a um sistema não-Oracle utilizando a conectividade genérica através das interfaces ODBC (ORACLE, 2013).

### **2.7.2 *DataBase Links (Links de Banco de Dados)***

Um *DataBase Link (DBLink)* é um objeto de esquema em um banco de dados que permite o acesso a objetos em outros banco de dados. O outro banco de dados não precisa ser um sistema de banco de dados Oracle. No entanto, para acessar sistemas não-Oracle, deve-se usar o Serviço Heterogêneo do Oracle. Depois de ter criado um *link* de banco de dados, pode-se usá-lo para se referir a tabelas e exibições do outro banco de dados (ORACLE, 2013).

A grande vantagem dos *Database Links* é que eles permitem que um usuário local possa acessar um *link* para um banco de dados remoto, sem ter que ser um usuário no banco de dados remoto (ORACLE, 2013).



### 2.7.3 Visões e Apoio a Decisão

Uma visão é exatamente como uma tabela-base. Avaliar consultas feitas em visões é muito importante para aplicações de apoio à decisão. Diferentes grupos de analistas dentro de uma organização normalmente estão preocupados com diferentes aspectos do negocio e é conveniente definir visões que forneçam a cada grupo ideias dos detalhes do negócio que interessem em uma tomada de decisão (RAMAKRISHNAN, 2008).

Segundo Mesquita (2013), uma visão é como uma consulta predefinida baseada em uma ou mais tabelas. As visões podem receber consultas assim como uma tabela. Algumas das vantagens de se utilizar visões são:

- Restringir acesso a dados;
- Fazer consultas complexas de forma mais simplificada;
- Oferecer independência de dados.

### 2.7.4 Materialização de Visão

As consultas feitas em definições de visão complexas devem ser respondidas muito rapidamente, pois os usuários envolvidos em atividades de apoio à decisão exigem tempos de resposta interativos. Uma alternativa é calcular previamente a definição da visão e armazenar o resultado. Quando uma consulta é feita na visão materializada, a consulta é executada diretamente no resultado calculado previamente. Esta estratégia é chamada de materialização de visão. As visões materializadas podem ser usadas durante o processamento da consulta, da mesma maneira que as relações normais (RAMAKRISHNAN, 2008).

Segundo Oracle (2013), uma visão materializada (VM) é um objeto de banco de dados, que pode ainda ser chamado de *master tables* (um termo usado para replicação) ou *detail tables* (um termo usado para *data warehouse*). Para os objetivos da replicação, as visões materializadas permitem manter cópias de dados remotos em um banco de dados local.

As cópias podem ser atualizáveis por meio da característica de Replicação Avançada do Oracle.

Segundo Oracle (2013), as principais cláusulas para replicação de dados são:

- **Build**: esta cláusula especifica como a visão materializada será populada;
- **Immediate**: indica que a visão materializada será populada no momento de sua criação;
- **Deferred**: indica que a visão materializada será populada na próxima operação *REFRESH*;
- **Refresh**: refere-se aos métodos, modos e tempos para a atualização da visão materializada;
- **Fast**: método incremental que realiza a atualização de acordo com as mudanças que ocorreram;
- **Complete**: método completo que atualizará todos os dados da visão materializada, mesmo que seja possível realizar uma atualização incremental;
- **Force**: realiza uma atualização incremental se for possível, caso não seja, realiza uma atualização completa;
- **On Commit**: realiza uma atualização incremental toda vez que uma operação *COMMIT* for realizada;
- **On Demand**: especifica que não será feita uma atualização da visão materializada a menos que seja manual;
- **Start With**: define uma data de início para a atualização dos dados da visão materializada. Associada à cláusula *NEXT*, permite realizar atualizações temporárias.

### 2.7.5 PL/SQL (Linguagem Procedural /Linguagem de Consulta Estruturada)

O PL/SQL é extensão da Linguagem Procedural de SQL, combinando o SQL com os recursos procedurais de linguagens de programação. Ele foi desenvolvido pela Oracle Corporation no início dos anos 90, para melhorar as capacidades do SQL (PLSQL, 2013). A PL/SQL é uma linguagem importante, especialmente adequada para funcionar com o Oracle, devido a sua integração com a SQL, que é projetada para manipulação de dados, disponível em diversos ambientes, cada um dos quais tem vantagens diferentes (URMAN, 2002).

## 2.8 *MySql*

O MySQL surgiu da idéia de se criar um Banco de Dados Relacional estável, seguro, rápido e ainda pouco custoso, tendo quase tudo que seus concorrentes mais renomados têm, com a vantagem de ser gratuito e com o código fonte aberto (INFO ONLINE, 2013).

O MySQL é o banco de dados mais conhecido no mundo do *software* livre, uma das suas vantagens técnicas é ser um produto multiplataforma (SANTOS, 2013). Ele concorre e vence alguns de seus correspondentes em muitas áreas, particularmente em desempenho, escalabilidade e estabilidade, podendo ter um desempenho tão bom ou melhor que seus concorrentes (SUEHRING, 2002), conforme apresentado na Tabela 4.

Tabela 4 – Comparação do MySQL com outros produtos de SGBDR.  
Fonte: SUEHRING (2002)

Recurso	MySQL	Oracle	MS SQL Server	PostgreSQL
Transacional	Sim	Sim	Sim	Sim
De código-fonte aberto	Sim	Não	Não	Sim
TCO <sup>4</sup>	Baixo	Alto	Alto	Baixo
Linguagens de desenvolvimento	Muitas	Muitas	Poucas	Muitas
Base de usuários corporativos	Sim	Sim	Sim	Não
Suporte da empresa	Sim	Sim	Sim	Não
Opera em diversas plataformas	Sim	Sim	Não	Sim

<sup>4</sup> Custo Total de Propriedade (TCO) - é um cálculo que visa levar em conta todos os custos envolvidos no uso de um determinado equipamento ou solução. (MORIMOTO, 2005).

## 2.9 SQL

O SQL significa *Structured Query Language* (Linguagem Estruturada de Consulta) é uma linguagem de computador usada para obter informações de dados armazenadas em uma base de dados relacional (PATRICK, 2002).

Segundo Watson (2004), a SQL é uma linguagem de consulta interativa e uma linguagem de programação de banco de dados.

Complementando, Suehring (2002) diz que a SQL é um tipo de linguagem de programação especializada e desenvolvida para trabalhar com bancos de dados relacionais como MySQL, Oracle, Microsoft SQL Server, PostgreSQL, Informix e outros.

Atualmente, a linguagem SQL é considerada um padrão dos SGBDs relacionais (DAMAS, 2007).

### 2.9.1 Histórico

A SQL foi chamada inicialmente de SEQUEL (*Structured English Query Language* – Linguagem de Pesquisa em Inglês Estruturado), sendo projetada e implementada na IBM *Research* como uma interface para um sistema experimental de um Banco de Dados Relacional chamado SISTEMA R (ELMASRI; NAVATHE, 2005).

Em 1986, o *American National Standards Institute* (ANSI) e a *International Standards Organization* (ISO) publicaram os padrão para a SQL, chamada SQL-86. A IBM publicou seus próprios padrões para a SQL, a *System Application Architecture Database Interface* (SAA-SQL), em 1987. Uma extensão para o padrão SQL, a SQL-89, foi publicada em 1989 e um sistema de banco de dados típico atualmente suporta ao menos recursos dispostos na SQL-89 (SILBERSCHATZ *et al.*, 1999). A versão mais recente do padrão ANSI/ISSO SQL é a ISO/IEC 9075-11:2011 (ISO, 2013).

Cada Banco de Dados Relacional aplica sua própria versão do padrão ANSI SQL e, conseqüentemente, muitos aprimoram esse padrão e adicionam funcionalidades próprias. A padronização da linguagem de programação permite ao desenvolvedor construir um banco de dados de maneira muito parecida de uma plataforma para outra (SUEHRING, 2002). Na Tabela 5 é apresentada uma comparação de implementações de SQL entre alguns SGBDs relacionais.

Tabela 5 – Comparação de implementações de SQL. Fonte: Suehring (2002)

<b>SGBDR</b>	<b>Vantagens</b>	<b>Desvantagens</b>
Oracle	Estável, versátil e segura	TCO potencialmente alto
MS SQL Server	Estável e segura	TCO relativamente alto; proprietário
PostgreSQL	Banco de dados com boas perspectivas de êxito e TCO baixo	Ainda precisa ser amplamente implementado em utilização de negócios de larga escala
Informix	Estável	TCO geralmente mais alto
MySQL	Oferece um banco de dados com cenário de melhor caso sob vários aspectos; TCO baixo, estabilidade alta, segurança alta	Nem todas as versões disponíveis podem oferecer a linha completa de capacidades do MySQL

## 2.10 Data Warehouse

A tomada de decisão organizacional exige uma visão abrangente de todos os aspectos de uma empresa. Além disso, para tal análise complexa, frequentemente de natureza estatística, não é fundamental que seja usada uma versão mais atual dos dados. A solução natural é criar um repositório centralizado de todos os dados, chamado de *Data Warehouse* (DW) (RAMAKRISHNAN, 2008).

Segundo Machado (2000), a tecnologia de *Data Warehousing* é considerada como a evolução natural do Ambiente de Apoio à Decisão. Ele representa uma grande base de dados capaz de integrar as informações de interesse para a empresa, de forma concisa e confiável, para utilização em Sistemas de Apoio à Decisão. É um armazém de dados históricos, cuja finalidade é apresentar informações que permitam identificar indicadores e a evolução de

valores ao longo de uma grande janela de tempo. Sua crescente utilização está relacionada à necessidade do domínio das informações estratégicas para garantir respostas e ações rápidas.

### ***2.10.1 Características da Tecnologia de Data Warehousing***

Segundo Machado (2000), o DW possui um conjunto de características que o distingue de outros ambientes de sistemas convencionais:

- Extração de dados de fontes heterogênea;
- Transformação e integração dos dados antes de sua carga;
- Normalmente requer máquina e suporte próprio;
- Visualização dos dados em diferentes níveis;
- Utilização de ferramentas voltadas para acesso com diferentes níveis de apresentação.

Machado (2000) ainda complementa que uma das características de suma importância em um DW é a integração, pois todos os seus dados tem um alto nível de integração. Por exemplo, as convenções de nomes, valores de variáveis, tais como: sexo masculino e feminino e outros atributos físicos de dados, são formalmente unificados e integrados nesta base única.

A Figura 2, exemplifica a integração entre dois SGBDs.

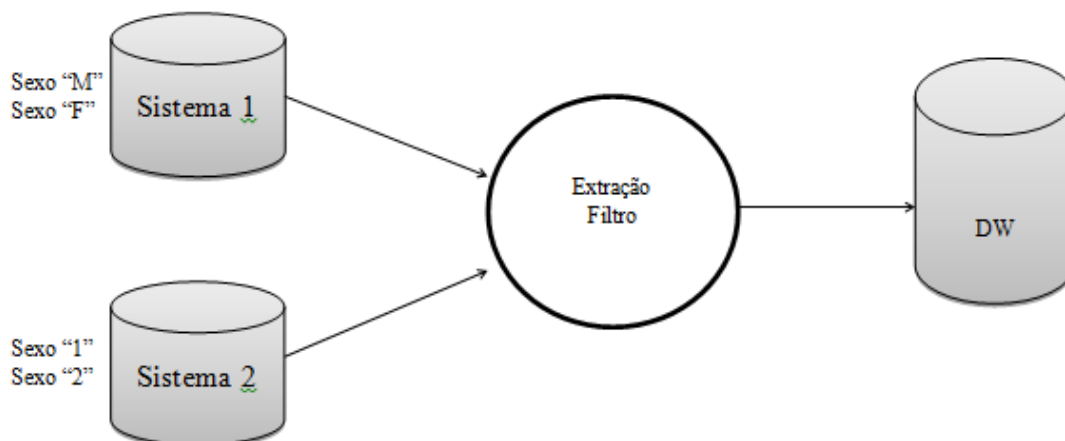


Figura 2 – Exemplo de Integração. Fonte: Machado (2000)

## 2.11 Performance

O desempenho do DW tem início na modelagem física e perdura durante toda a sua operação dentro de uma organização, tornando necessário um conjunto de ações coordenadas cujo objetivo é sustentar um patamar de desempenho adequado para a organização suportar seus processos de tomada de decisão (JOSKO, 2013).

Para saber justificar quantitativamente as melhorias de desempenho em um banco de dados é preciso realizar testes que sejam de acordo com o ambiente, pois necessita-se considerar uma série de fatores (PICHILIANI, 2013).

Para se otimizar a performance dos códigos SQL e PL/SQL, pode-se utilizar um recurso da ferramenta PL/SQL chamado *Automatic statistics*. Esta funcionalidade permite visualizar informações do tempo de execução de cada linha do código e, ainda, obter automaticamente estatísticas de tempo de execução de declarações SQL e programas PL/SQL (ALLROUND AUTOMATIONS, 2013).

Pode-se visualizar a utilização de recursos reais de uma instrução SQL ou PL/SQL exibindo estatísticas e gerando, automaticamente, um relatório de estatística desta execução, acessando a guia de estatísticas na tela *SQL Window*. Nestas estatísticas, pode-se incluir o uso de CPU, *block I/O*, registro de I/O, *table scans*, *sorts*, entre outros. Para cada estatística, é

possível visualizar o valor da última execução e o total para a sessão atual (ALLROUND AUTOMATIONS, 2007). O conjunto padrão de estatísticas está apresentado na Tabela 6.

Tabela 6 - Conjunto de estatísticas padrão - PL/SQL.  
Fonte: Allround Automations (2007)

Statistic	Meaning
CPU usado por esta sessão	O uso da CPU em centésimos de segundo
Leituras físicas	O número de blocos lidos a partir do disco
Escrita física	O número de blocos gravados no disco
Leitura sessão lógica	O número de blocos de leitura a partir do <i>buffer</i> ou a partir do disco
Ordenações (disco)	O número de ordenações realizadas num segmento temporário
Ordenações (memória)	O número de ordenações executada na memória
Ordenações (linhas)	O número de linhas que foram ordenadas
Linhas da tabela por Id	O número de linhas carregadas por id, geralmente como resultado de acesso a índice
Blocos obtidos na varredura	O número de blocos de leitura para a varredura de tabela completa
Linhas obtidas na varredura	O número de linhas de leitura para varredura de tabela completa
Varredura tabela (tabelas longas)	O número de tabela completa na varredura em tabelas longas
Varredura tabela (tabelas curtas)	O número de tabela completa na varredura em tabelas curtas

Na Figura 3 é apresentado um conjunto de estatísticas em tempo de execução.

Name	Last	Total
CPU used by this session	0	0
physical reads	5	12
physical writes	0	0
session logical reads	17	45
sorts (disk)	0	0
sorts (memory)	2	2
sorts (rows)	5	5
table fetch by rowid	1	6
table scan blocks gotten	3	5
table scan rows gotten	19	21
table scans (long tables)	0	0
table scans (short tables)	3	5

Executed in 0,084 seconds

Figura 3 - Estatísticas de tempo de execução. Fonte: Allround Automations



## **2.12 Trabalhos Relacionados**

A seguir serão apresentados trabalhos cujo tema está relacionado ao desta pesquisa.

Parolin (2011) destacou, por meio de seu trabalho, a importância de conhecer diversas formas de como o DW pode ser empregado nas várias áreas dentro das organizações, sua origem, evolução, características, técnicas e como a ferramenta contribui para agregar valor no negócio e proporcionar agilidade e precisão nos processos de tomada de decisão.

Ziulkoski (2003) apresentou, por meio de seu trabalho, um estudo sobre o uso de técnicas de aquisição de conhecimento na coleta de requisitos e modelagem de dados para DW, enfatizando as características de um DW como sistema de apoio a decisão, desenvolvendo um estudo de caso envolvendo modelagem de dados e analisando as vantagens obtidas com a aplicação de técnicas de aquisição de conhecimento na coleta de requisitos em DW.

Italiano (2000) abordou o uso de Visões Materializadas em *Data Warehouses*, buscando analisar algumas das alternativas disponíveis para uma implementação mais eficiente e dinâmica, concluindo que vários estudos têm sido feitos com o objetivo de conferir ao DW uma característica mais dinâmica e otimizar o acesso aos dados, indicando o uso de visões materializadas como alternativa viável.

Neves (2009) destacou a materialização de visões como uma técnica comumente utilizada no projeto físico de um *Data Warehouse*, tornando persistentes visões pré-computadas, para que possam responder, de forma eficiente às consultas submetidas de acordo com a necessidade dos usuários. Apresentando uma abordagem que automatiza e define uma estratégia para a seleção de visões a materializar em tempo de projeto do DW.



### 3 Metodologia

A metodologia seguiu o método de estudo exploratório.

Segundo Gil (2002), o método exploratório visa proporcionar maior familiaridade com o problema, com intuito de torná-lo explícito ou construir hipóteses.

Para alcançar os objetivos, a metodologia deste trabalho foi dividida em etapas, levando em conta a dimensão do assunto tratado e o grau de importância do mesmo. Os conceitos sobre *Data Warehouse*, Replicação, Visões Materializadas e SQL são essenciais para o desenvolvimento deste trabalho.

Na Figura 4 são apresentadas as etapas que foram definidas para a metodologia deste trabalho.

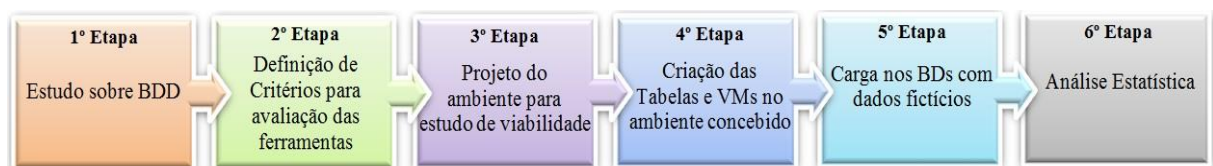


Figura 4 - Figura metodológica

As próximas seções irão reportar o que foi desenvolvido em cada uma das etapas apresentadas na Figura 4.

#### 3.1 *Estudo sobre BDD*

A primeira etapa foi focada no levantamento bibliográfico e realizadas pesquisas e estudos sobre bancos de dados distribuídos para definir quais bancos de dados seriam adotados. Com base nas Tabelas 4 e 5, devido a estabilidade, segurança e pelos recursos disponibilizados para viabilidade da análise de otimização de consultas, optou-se por utilizar os SGBDR's Oracle e MySQL.

### 3.2 Definição de critérios para avaliação das ferramentas

A análise da primeira etapa propiciou também a definição dos critérios para avaliação das ferramentas de replicação e do ambiente de desenvolvimento integrado, PL/SQL, avaliando suas características e recursos.

Utilizando o Serviço Heterogêneo do Oracle, serão criados os *DBLinks* para acesso aos BD MySQL e por meio de visões materializadas será feita a replicação.

A ferramenta PL/SQL será utilizada para gerar as estatísticas, por meio de seu recurso chamado *Automatic statistics*. Esta funcionalidade permite visualizar informações do tempo de execução de cada linha do código e, ainda, obter automaticamente estatísticas de tempo de execução de declarações SQL. A Tabela 7, a seguir, apresenta os dados que serão analisados por meio desta ferramenta.

Tabela 7 - Conjunto de estatísticas analisadas.

Statistic	Meaning
Tempo de execução	Tempo de execução do <i>script</i>
CPU usado por esta sessão	O uso da CPU em centésimos de segundo
Leituras físicas	O número de blocos lidos a partir do disco
Escrita física	O número de blocos gravados no disco
Leitura sessão lógica	O número de blocos de leitura a partir do <i>buffer</i> ou a partir do disco
Ordenações (linhas)	O número de linhas que foram ordenadas

### 3.3 Projeto do ambiente para estudo de viabilidade

Definidos os BDD, as ferramentas e os métodos a serem utilizados para alcançar os objetivos, foi projetado o ambiente, apresentado na Figura 5, com os seguintes recursos:

- **Processador Intel Core i3-2100 CPU 3.10GHz Memória RAM - 4,00 GB**
- **Windows Server 2008 R2:** sistema operacional Microsoft Windows baseado na arquitetura NT.

- **Oracle Database 10g Express Edition:** versão gratuita da família Oracle Database;
- **MySQL 5.5.22:** SGDB que utiliza a linguagem SQL;
- **MySQL Connector ODBC 5.2.2:** conector de banco de dados;
- **PL/SQL Developer:** extensão da linguagem padrão SQL para o SGDB Oracle;

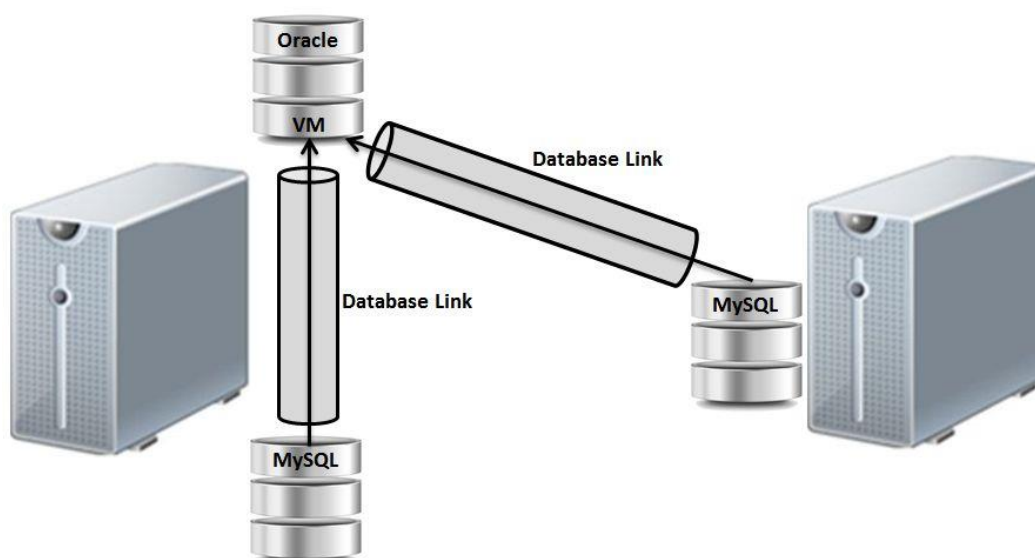


Figura 5 – Ambiente de Análise

### 3.4 Criação dos Databases, Tabelas e VMs

Após a concepção do ambiente para estudo, foi criado o *database* denominado “dbmysql” e, em seguida, foram criadas as tabelas “pessoa” e “cidade”, em ambos os BDs MySQL.

A partir deste ponto foram criados os *DBLinks*, no BD Oracle, para acesso a estes *databases* e a replicação das tabelas por meio de visões materializadas.

Apresenta-se a seguir o *script* de criação da Visão Materializada da tabela “pessoa”, que unifica os dados dos BDs MySQL por meio dos *DBLinks* criados.

```

1 CREATE MATERIALIZED VIEW VM_PESSOA
2 REFRESH FORCE ON DEMAND
3 START WITH TO_DATE('16-10-2013 08:01:34', 'DD-MM-YYYY HH24:MI:SS') NEXT SYSDATE + 1/86400
4 AS
5 SELECT
6     'MYSQL 1' AS DATABASE,
7     P1.NRSEQUENCIAL,
8     P1.NOPESSOA,
9     P1.DTNASCIMENTO,
10    P1.CDSEXO,
11    P1.CDESTADO_CIVIL,
12    P1.NRSEQUENCIAL_CIDADE
13 FROM
14     PESSOA@dbmysql1 P1
15 UNION ALL
16 SELECT
17     'MYSQL 2' AS DATABASE,
18     P2.NRSEQUENCIAL,
19     P2.NOPESSOA,
20     P2.DTNASCIMENTO,
21     P2.CDSEXO,
22     P2.CDESTADO_CIVIL,
23     P2.NRSEQUENCIAL_CIDADE
24 FROM
25     PESSOA@dbmysql2 P2;
26

```

Figura 6 – Script de criação da Visão Materializada da tabela “pessoa”

Apresenta-se a seguir o *script* de criação da Visão Materializada da tabela “cidade”, que unifica os dados dos BDs MySQL por meio dos *DBLinks* criados.

```

1 CREATE MATERIALIZED VIEW VM_CIDADE
2 REFRESH FORCE ON DEMAND
3 START WITH TO_DATE('14-10-2013 23:30:43', 'DD-MM-YYYY HH24:MI:SS') NEXT SYSDATE + 1/86400
4 AS
5 SELECT
6     'MYSQL 1' AS DATABASE,
7     C1.NRSEQUENCIAL,
8     C1.NOCIDADE,
9     C1.CDESTADO,
10    C1.CDIBGE
11 FROM
12     CIDADE@dbmysql1 C1
13 UNION ALL
14 SELECT
15     'MYSQL 2' AS DATABASE,
16     C2.NRSEQUENCIAL,
17     C2.NOCIDADE,
18     C2.CDESTADO,
19     C2.CDIBGE
20 FROM
21     CIDADE@dbmysql2 C2;
22

```

Figura 7 – Script de criação da Visão Materializada da tabela “cidade”

### 3.5 Carga nos BDs

Foi feita a carga de dados, com informações fictícias, em ambos os BDs MySQL. Estas informações foram replicadas para o BD Oracle.

A fim de analisar a performance no levantamento dos dados, foi feita uma carga inicial, considerando um volume de dados menor. Após a primeira análise, foi feita outra carga de dados, com um volume dados maior, para comparação do desempenho dos *scripts*.

### 3.6 Análise Estatística

A análise foi dividida em experimentos, considerando sempre a primeira execução da consulta, desta forma não buscando resultados armazenados na memória do banco de dados, onde cada experimento possui uma complexidade diferente e uma carga de dados diferente, para que os resultados possam ser conclusivos e objetivos.

A seguir serão apresentados os experimentos deste trabalho.

#### Experimento I:

Foi feita uma carga em ambos os BDs MySQL, contendo dados fictícios, sendo:

- Tabela PESSOA: Carga com 500.000 mil registros;
- Tabela CIDADE: Carga com 5.341 mil registros;

Neste experimento, foram utilizadas, a fim de obter resultados satisfatórios para análise de desempenho das consultas, funções para conversão de tipos de dados, funções de agregação, funções de agregação como argumento de chamada de função, expressões condicionais e de ordenação.

Após a replicação a partir das visões materializadas, foram criados dois *scripts* para fins estatísticos, montando um levantamento para análise da performance da consulta.

O *script* S1 consulta os dados com base na replicação, feita a partir da visão materializada da tabela “pessoa”, projetando a quantidade de pessoas por sexo e faixa etária,

enquanto o *script* S2 calcula previamente o resultado da mesma consulta S1 e armazena em outra visão materializada.

Apresenta-se a seguir o *script* da consulta S1.

```

1 SELECT
2   DECODE(CDSEXO, 'M', 'MASCULINO', 'F', 'FEMININO') SEXO,
3   SUM("0 A 18") "0 A 18",
4   SUM("19 A 21") "19 A 21",
5   SUM("22 A 33") "22 A 33",
6   SUM("34 A 41") "34 A 41",
7   SUM("42 A 53") "42 A 53",
8   SUM("54 A 61") "54 A 61",
9   SUM("ACIMA DE 61") "ACIMA DE 61"
10  FROM
11  (
12    SELECT
13      DATABASE, NRSEQUENCIAL, CDSEXO,
14      CASE WHEN IDADE BETWEEN 0 AND 18 THEN COUNT(DISTINCT(NRSEQUENCIAL))
15        ELSE 0 END AS "0 A 18",
16      CASE WHEN IDADE BETWEEN 19 AND 21 THEN COUNT(DISTINCT(NRSEQUENCIAL))
17        ELSE 0 END AS "19 A 21",
18      CASE WHEN IDADE BETWEEN 22 AND 33 THEN COUNT(DISTINCT(NRSEQUENCIAL))
19        ELSE 0 END AS "22 A 33",
20      CASE WHEN IDADE BETWEEN 34 AND 41 THEN COUNT(DISTINCT(NRSEQUENCIAL))
21        ELSE 0 END AS "34 A 41",
22      CASE WHEN IDADE BETWEEN 42 AND 53 THEN COUNT(DISTINCT(NRSEQUENCIAL))
23        ELSE 0 END AS "42 A 53",
24      CASE WHEN IDADE BETWEEN 54 AND 61 THEN COUNT(DISTINCT(NRSEQUENCIAL))
25        ELSE 0 END AS "54 A 61",
26      CASE WHEN IDADE > 61 THEN COUNT(DISTINCT(NRSEQUENCIAL))
27        ELSE 0 END AS "ACIMA DE 61"
28    FROM
29    (
30      SELECT
31        VP.DATABASE, VP.NRSEQUENCIAL, VP.CDSEXO,
32        FN_IDADE(VP.DTNASCIMENTO, SYSDATE) IDADE
33      FROM VM_PESSOA VP
34    )
35    GROUP BY DATABASE,
36             NRSEQUENCIAL,
37             CDSEXO,
38             IDADE
39  )
40  GROUP BY
41    CDSEXO
42  ORDER BY
43    CDSEXO
44

```

Figura 8 – *Script* da consulta S1 (Experimento I)

Na Figura 9 apresenta-se o resultado da consulta S1, com tempo de execução de aproximadamente 13 minutos, e na Tabela 8, seus resultados estatísticos.

SEXO	0 a 18	19 a 21	22 a 33	34 a 41	42 a 53	54 a 61	Acima de 61
FEMININO	72262	18828	150590	150344	65600	23026	45316
MASCULINO	81216	19478	132654	125418	55488	19684	40096

1:1	13:00	2 rows selected in 780,583 seconds
-----	-------	------------------------------------

Figura 9 – Resultado do *Script* S1 (Experimento I)



Tabela 8 – *Automatic statistics - Script S1* (Experimento I)

Estatística	Significado	Resultado
Tempo de execução	Tempo de execução do <i>script</i> (em minutos)	13
CPU usado por esta sessão	O uso da CPU em centésimos de segundo	8417
Leituras físicas	O número de blocos lidos a partir do disco	54660
Escrita física	O número de blocos gravados no disco	7877
Leitura sessão lógica	O número de blocos de leitura a partir do <i>buffer</i> ou do disco	3085
Ordenações (linhas)	O número de linhas que foram ordenadas	2000002

O *script* S2 apresentado a seguir, retorna o mesmo levantamento do *script* S1, porém armazenando seu resultado em outra VM denominada VM\_QTDE\_PES\_SX\_FXET.

Apresenta-se a seguir o *script* de criação da VM da consulta S2, para armazenar o resultado previamente calculado.

```

1 CREATE MATERIALIZED VIEW VM_QTDE_PES_SX_FXET
2 REFRESH FORCE ON DEMAND
3 START WITH TO_DATE('16-10-2013 14:22:40', 'DD-MM-YYYY HH24:MI:SS') NEXT SYSDATE + 1/86400 AS
4 SELECT
5     DECODE(CDSEXO, 'M', 'MASCULINO', 'F', 'FEMININO') SEXO,
6     SUM("0 A 18") "0 A 18",
7     SUM("19 A 21") "19 A 21",
8     SUM("22 A 33") "22 A 33",
9     SUM("34 A 41") "34 A 41",
10    SUM("42 A 53") "42 A 53",
11    SUM("54 A 61") "54 A 61",
12    SUM("ACIMA DE 61") "ACIMA DE 61"
13 FROM
14     ( SELECT
15         DATABASE, NRSEQUENCIAL, CDSEXO,
16         CASE WHEN IDADE BETWEEN 0 AND 18 THEN COUNT(DISTINCT(NRSEQUENCIAL))
17         ELSE 0 END AS "0 A 18",
18         CASE WHEN IDADE BETWEEN 19 AND 21 THEN COUNT(DISTINCT(NRSEQUENCIAL))
19         ELSE 0 END AS "19 A 21",
20         CASE WHEN IDADE BETWEEN 22 AND 33 THEN COUNT(DISTINCT(NRSEQUENCIAL))
21         ELSE 0 END AS "22 A 33",
22         CASE WHEN IDADE BETWEEN 34 AND 41 THEN COUNT(DISTINCT(NRSEQUENCIAL))
23         ELSE 0 END AS "34 A 41",
24         CASE WHEN IDADE BETWEEN 42 AND 53 THEN COUNT(DISTINCT(NRSEQUENCIAL))
25         ELSE 0 END AS "42 A 53",
26         CASE WHEN IDADE BETWEEN 54 AND 61 THEN COUNT(DISTINCT(NRSEQUENCIAL))
27         ELSE 0 END AS "54 A 61",
28         CASE WHEN IDADE > 61 THEN COUNT(DISTINCT(NRSEQUENCIAL))
29         ELSE 0 END AS "ACIMA DE 61"
30     FROM
31         ( SELECT
32             VP.DATABASE, VP.NRSEQUENCIAL, VP.CDSEXO,
33             FN_IDADE(VP.DTNASCIMENTO, SYSDATE) IDADE
34             FROM VM_PESSOA VP
35         )
36     GROUP BY DATABASE,
37             NRSEQUENCIAL,
38             CDSEXO,
39             IDADE
40     )
41 GROUP BY CDSEXO
42 ORDER BY CDSEXO
43

```

Figura 10 – *Script* da consulta S2 (Experimento I)

Apresenta-se na Figura 11 o resultado da consulta S2, a partir da VM VM\_QTDE\_PES\_SX\_FXET, com tempo de execução de aproximadamente 0,1 milissegundo e, a Tabela 9, apresenta seus resultados estatísticos.

SEXO	0 A 18	19 A 21	22 A 33	34 A 41	42 A 53	54 A 61	ACIMA DE 61
FEMININO	72262	18828	150590	150344	65600	23026	45316
MASCULINO	81216	19478	132654	125418	55488	19684	40096

1:1	2 rows selected in 0,01 seconds
-----	---------------------------------

Figura 11 – Resultado do *Script S2* (Experimento I)

Tabela 9 – *Automatic statistics - Script S2* (Experimento I)

Estatística	Significado	Resultado
Tempo de execução	Tempo de execução do <i>script</i> (em minutos)	0,01
CPU usado por esta sessão	O uso da CPU em centésimos de segundo	2
Leituras físicas	O número de blocos lidos a partir do disco	0
Escrita física	O número de blocos gravados no disco	0
Leitura sessão lógica	O número de blocos de leitura a partir do <i>buffer</i> ou do disco	20
Ordenações (linhas)	O número de linhas que foram ordenadas	55

Por meio das estatísticas geradas com a execução dos *scripts* S1 e S2, pode-se verificar, neste experimento, que o *script* S2, que calcula previamente a definição da visão e armazena o resultado em uma visão materializada, teve seus resultados de desempenho superior ao *script* S1.

#### Experimento II:

No segundo experimento, foram utilizados os mesmos *scripts* S1 e S2 utilizados no experimento I, porém foi feita outra carga em ambos os BDs MySQL, contendo mais dados fictícios, sendo:

- Tabela PESSOA: Carga com 50.000.000 milhões de registros;
- Tabela CIDADE: Carga com 5.341 registros.

A Figura 12 apresenta o resultado da consulta S1, com tempo de execução de aproximadamente 2 horas e meia e, a Tabela 10, apresenta seus resultados estatísticos.

	SEXO	0 A 18	19 A 21	22 A 33	34 A 41	42 A 53	54 A 61	ACIMA DE 61
1	FEMININO	7752430	2077392	13361958	9134624	8191240	3798764	7963322
2	MASCULINO	8870410	2178960	12003044	7645696	6821266	3176396	7024498

80:1 155:55 2 rows selected in 9355,582 seconds

Figura 12 – Resultado do *Script S1* (Experimento II)

Tabela 10 – *Automatic statistics - Script S1* (Experimento II)

Estatística	Significado	Resultado
Tempo de execução	Tempo de execução do <i>script</i> (em minutos)	155
CPU usado por esta sessão	O uso da CPU em centésimos de segundo	231962
Leituras físicas	O número de blocos lidos a partir do disco	1085649
Escrita física	O número de blocos gravados no disco	1085591
Leitura sessão lógica	O número de blocos de leitura a partir do <i>buffer</i> ou do disco	2773564
Ordenações (linhas)	O número de linhas que foram ordenadas	200001198

Apresenta-se na Figura 13 o resultado da consulta S2, a partir da VM VM\_QTDE\_PES\_SX\_FXET, com tempo de execução de aproximadamente 0,3 milissegundos e, a Tabela 11, apresenta seus resultados estatísticos.

	SEXO	0 A 18	19 A 21	22 A 33	34 A 41	42 A 53	54 A 61	ACIMA DE 61
1	FEMININO	7752430	2077392	13361958	9134624	8191240	3798764	7963322
2	MASCULINO	8870410	2178960	12003044	7645696	6821266	3176396	7024498

2:14 2 rows selected in 0,321 seconds

Figura 13 – Resultado do *Script S2* – Experimento II

Tabela 11 – *Automatic statistics - Script S2* (Experimento II)

Estatística	Significado	Resultado
Tempo de execução	Tempo de execução do <i>script</i> (em minutos)	0,321
CPU usado por esta sessão	O uso da CPU em centésimos de segundo	2
Leituras físicas	O número de blocos lidos a partir do disco	1
Escrita física	O número de blocos gravados no disco	0
Leitura sessão lógica	O número de blocos de leitura a partir do <i>buffer</i> ou do disco	42
Ordenações (linhas)	O número de linhas que foram ordenadas	63

Com base neste experimento, pode-se verificar que o *script S1* teve uma performance bastante inferior e com grande queda de desempenho, quando aumentado o volume de dados, já o *script S2*, manteve um bom desempenho.

### Experimento III:

Neste experimento foram utilizadas, a fim de obter resultados satisfatórios para análise de desempenho das consultas, funções de agregação, ordenação e relacionamento entre as visões das tabelas de pessoa e cidade.

Foram criados mais dois *scripts* para fins estatísticos, buscando-se a quantidade de pessoas por cidade.

O *script* S3 consulta os dados com base na replicação feita a partir das visões materializadas das tabelas “pessoa” e “cidade”, enquanto o *script* S4 calcula previamente o resultado e armazena em outra visão materializada, conforme apresentado a seguir.

Apresenta-se a seguir, na Figura 14, o *script* da consulta S3.

```

Code section  Select
1  SELECT CDIBGE,
2      NOCIDADE,
3      CDESTADO,
4      COUNT (NRSEQUENCIAL) QTDE_PESSOAS
5  FROM
6  ( SELECT
7      DISTINCT
8      VMP.DB,
9      VMC.CDIBGE,
10     VMC.NOCIDADE,
11     VMC.CDESTADO,
12     VMP.NRSEQUENCIAL
13  FROM
14     BI.VM_PESSOA VMP,
15     BI.VM_CIDADE VMC
16  WHERE
17     VMP.NRSEQUENCIAL_CIDADE = VMC.NRSEQUENCIAL
18  )
19  GROUP BY
20     CDIBGE,
21     NOCIDADE,
22     CDESTADO
23  ORDER BY
24     CDESTADO,
25     NOCIDADE
26

```

Figura 14 – *Script* da consulta S3 (Experimento III)

A Figura 15 a seguir, apresenta o resultado da consulta S3, com tempo de execução de aproximadamente 35 minutos e, a Tabela 12, apresenta seus resultados estatísticos.

	CDIBGE	NOCIDADE	CDESTADO	QTDE_PESSOAS
1	2706000	OLIVENCA	AL	1806
2		NOVO BRASIL	GO	200
3		NOVO CRUZEIRO	MG	1998000
4	4300646	AMETISTA DO SUL	RS	1498
5	3500303	AGUAI	SP	10000498
6	3500402	AGUAS DA PRATA	SP	10000000
7	3509502	CAMPINAS	SP	1980000
8	3515186	ESPIRITO SANTO DO PINHAL	SP	10018000
9	3549102	SAO JOAO DA BOA VISTA	SP	30000000
10	3550308	SAO PAULO	SP	31999998
11	3556404	VARGEM GRANDE DO SUL	SP	4000000

8:10 34:58 11 rows selected in 2098,898 seconds

Figura 15 – Resultado do *Script S3* (Experimento III)

Tabela 12 – *Automatic statistics - Script S3* (Experimento III)

Estatística	Significado	Resultado
Tempo de execução	Tempo de execução do <i>script</i> (em minutos)	34
CPU usado por esta sessão	O uso da CPU em centésimos de segundo	42412
Leituras físicas	O número de blocos lidos a partir do disco	2880934
Escrita física	O número de blocos gravados no disco	2463285
Leitura sessão lógica	O número de blocos de leitura a partir do <i>buffer</i> ou do disco	442425
Ordenações (linhas)	O número de linhas que foram ordenadas	17256

O *script* S4 apresentado a seguir, na Figura 16, retorna o mesmo levantamento do *script* S3, porém armazenando seu resultado em outra VM denominada VM\_QTDE\_PES\_CIDADE.

Apresenta-se a seguir o *script* de criação da VM da consulta S4, que armazenar o resultado em outra visão materializada.

```

1 CREATE MATERIALIZED VIEW VM_QTDE_PES_CIDADE
2 REFRESH FORCE ON DEMAND
3 START WITH TO_DATE('28-03-2014 14:22:40', 'DD-MM-YYYY HH24:MI:SS') NEXT SYSDATE + 1/86400
4 AS
5 SELECT
6     CDIBGE,
7     NOCIDADE,
8     CDESTADO,
9     COUNT(NRSEQUENCIAL) QTDE_PESSOAS
10 FROM
11 (
12     SELECT
13         DISTINCT
14         VMP.DB,
15         VMC.CDIBGE,
16         VMC.NOCIDADE,
17         VMC.CDESTADO,
18         VMP.NRSEQUENCIAL
19     FROM
20         BI.VM_PESSOA VMP,
21         BI.VM_CIDADE VMC
22     WHERE
23         VMP.NRSEQUENCIAL_CIDADE = VMC.NRSEQUENCIAL
24 )
25 GROUP BY
26     CDIBGE,
27     NOCIDADE,
28     CDESTADO

```

Figura 16 – Script da consulta S4 (Experimento III)

A Figura 17 mostra o resultado da consulta S4, a partir da VM VM\_QTDE\_PES\_CIDADE, com tempo de execução de aproximadamente 0,02 milissegundos e, a Tabela 13, apresenta seus resultados estatísticos.

	CDIBGE	NOCIDADE	CDESTADO	QTDE_PESSOAS
1	2706000	OLIVENCA	AL	1806
2		NOVO BRASIL	GO	200
3		NOVO CRUZEIRO	MG	1998000
4	4300646	AMETISTA DO SUL	RS	1498
5	3500303	AGUAI	SP	10000498
6	3500402	AGUAS DA PRATA	SP	10000000
7	3509502	CAMPINAS	SP	1980000
8	3515186	ESPIRITO SANTO DO PINHAL	SP	10018000
9	3549102	SAO JOAO DA BOA VISTA	SP	30000000
10	3550308	SAO PAULO	SP	31999998
11	3556404	VARGEM GRANDE DO SUL	SP	4000000

11 rows selected in 0,02 seconds

Figura 17 – Resultado do Script S4 (Experimento III)

Tabela 13 – *Automatic statistics* – *Script S4* (Experimento III)

<b>Estatística</b>	<b>Significado</b>	<b>Resultado</b>
Tempo de execução	Tempo de execução do <i>script</i> (em minutos)	0,02
CPU usado por esta sessão	O uso da CPU em centésimos de segundo	0
Leituras físicas	O número de blocos lidos a partir do disco	0
Escrita física	O número de blocos gravados no disco	0
Leitura sessão lógica	O número de blocos de leitura a partir do <i>buffer</i> ou do disco	4
Ordenações (linhas)	O número de linhas que foram ordenadas	66

Pode-se verificar, neste experimento, que o *script* S4, que calcula previamente a definição da visão e armazena o resultado em uma visão materializada, teve seus resultados de desempenho superior ao *script* S4.

Foram finalizados os experimentos e o próximo capítulo tem como finalidade analisar as estatísticas dos resultados obtidos.





## 4 Resultados

Neste capítulo serão verificados os resultados obtidos com a criação dos *database links* e da replicação, por meio das visões materializadas no banco de dados Oracle, com diferentes volumes de dados, para análise do comportamento de desempenho dos *scripts* no levantamento das informações.

Com base nas estatísticas geradas pelo PL/SQL dos dois *scripts* (S1 e S2), pode-se obter os seguintes resultados apresentados na Tabela 14 e na Figura 18.

Tabela 14 – Comparação Estatística dos *Scripts* S1 e S2.

Carga de dados	1 Milhão		100 Milhões	
	S1	S2	S1	S2
Tempo de Execução (minutos)	13	0,01	155	0,321
Uso da CPU (centésimos de segundo)	8417	2	231962	2
Blocos de Leitura	54660	0	1085649	1
Escrita Física	7877	0	1085591	0
Sessão Lógica	3085	20	2773564	42
Ordenação (linhas)	2000002	55	200001198	63

Pode-se concluir pela análise da Tabela 14, que o desempenho do *script* S1 sofreu uma grande queda de desempenho quando o volume de dados aumentou de 1 milhão para 100 milhões de registros. Já o *script* S2, que calcula previamente o resultado e o armazena em

uma visão materializada, manteve sua performance dentro de um padrão estável, com tempo de resposta muito mais rápido.

A Figura 18 mostra a análise de cada item da Tabela 14, segundo as estatísticas analisadas pelo recurso *Automatic statistics* da ferramenta PL/SQL.

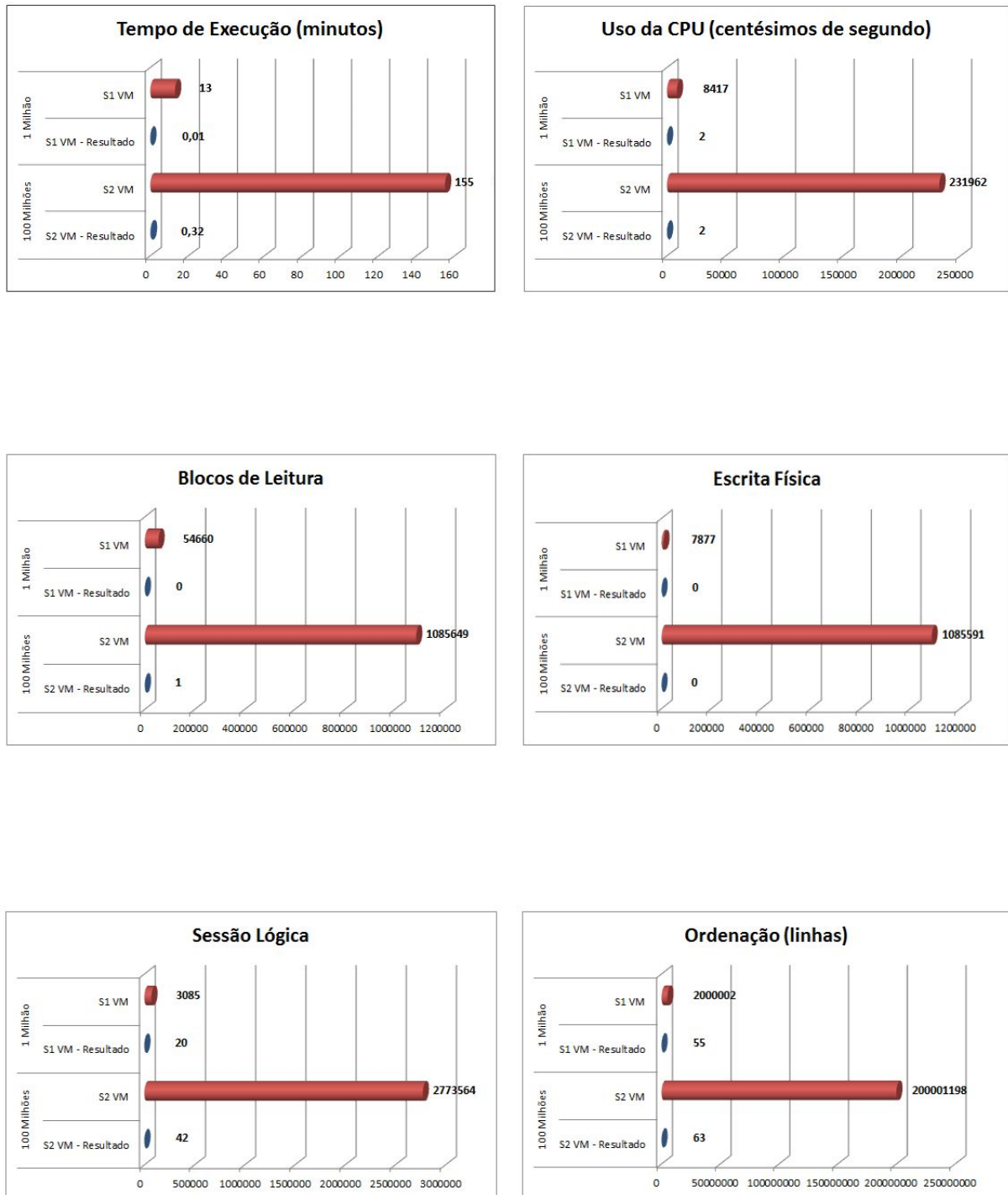


Figura 18 – Análise Estatística dos *Scripts* S1 e S2

Com base nas estatísticas geradas dos dois *scripts* (S3 e S4), pode-se obter os seguintes resultados apresentados na Figura 19.

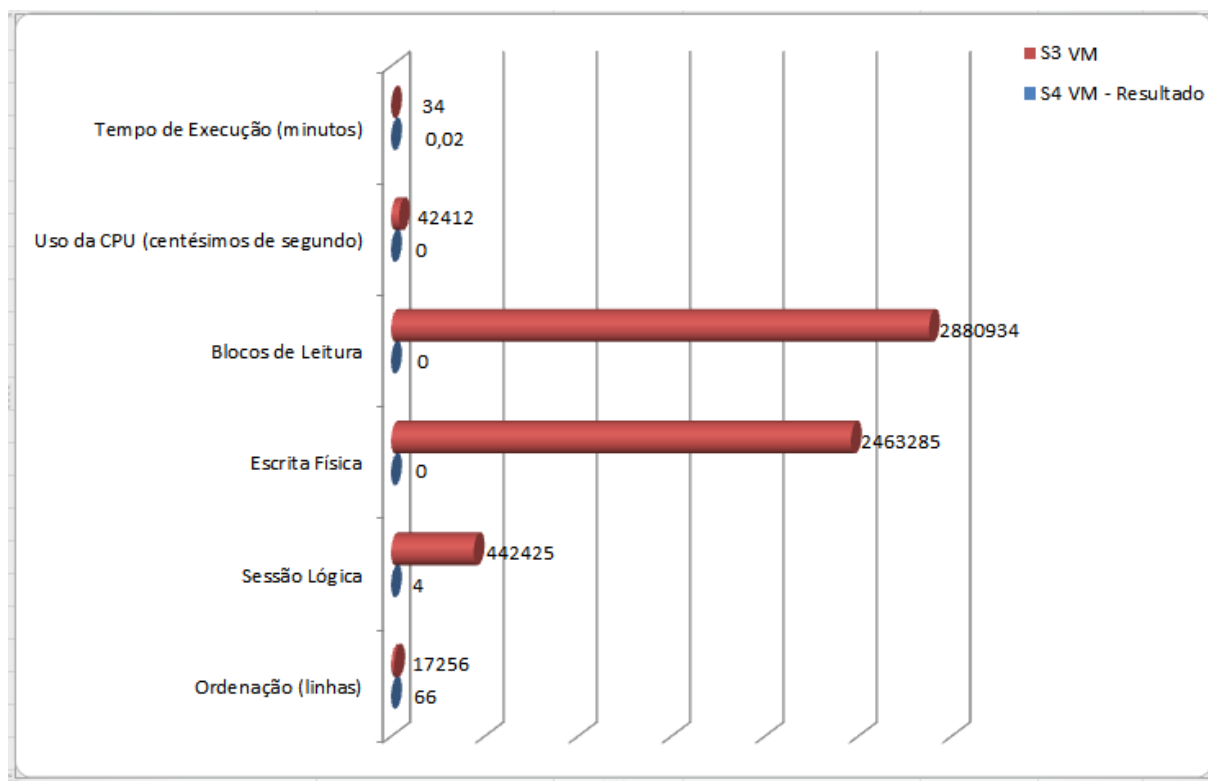


Figura 19 – Análise Estatística dos *Scripts* S3 e S4

Pode-se concluir pela análise da Figura 19 que, o comportamento do *script* S3 teve um tempo de execução para o retorno da consulta muito maior, e um desempenho muito inferior ao *scprit* S4, que armazena o resultado previamente calculado em uma visão materializada.



## 5 Conclusões

Para o desenvolvimento deste trabalho, foram realizados pesquisas e estudos sobre Bancos de Dados Distribuídos, *Data Warehouse*, Replicação, Visões Materializadas e SQL, a fim de proporcionar a compreensão sobre os conceitos, estrutura e funcionamento de todo o processo de consulta em *Data Warehouse* utilizando Banco de Dados Distribuídos, para que fosse possível propiciar um ambiente para estudo de viabilidade de otimização da performance de consultas, por meio de visões materializadas.

Após a concepção do ambiente para estudo, a fim de obter resultados satisfatórios para análise de desempenho das consultas, foram desenvolvidos alguns *scripts* SQL, utilizando funções para conversão de tipos de dados, funções de agregação, funções de agregação como argumento de chamada de função, expressões condicionais e de ordenação, buscando as informações por meio de visões materializadas e também calculando o resultado e o armazenando em outra visão materializada. A obtenção dos resultados estatísticos para análise foi feita a partir do recurso da ferramenta PL/SQL chamado *Automatic statistics*, que permite visualizar informações do tempo de execução de cada linha do código e, ainda, obter automaticamente estatísticas de tempo de execução de declarações SQL. A análise destes resultados foram divididas em experimentos, cada qual com uma complexidade diferente e uma carga de dados diferente, para que os resultados pudessem ser conclusivos e objetivos.

Os *scripts* que calcularam previamente o resultado e o armazenaram em uma visão materializada, mantiveram sua performance dentro de um padrão estável, mesmo com variação de volume de dados, enquanto os *scripts* que fizeram o levantamento com base nas

visões materializadas tiveram uma performance bastante inferior e com grande queda de desempenho quando aumentado o volume de dados.

Pode-se concluir com os resultados obtidos neste trabalho que a otimização de consulta aos dados, através do uso de banco de dados distribuídos e replicação heterogênea, não só otimiza o tempo de consulta de acesso aos dados mas também a performance do Banco de Dados, tornando-se uma opção viável para implementação de um *Data Warehouse*.

### **5.1 Trabalhos Futuros**

A análise estatística do desempenho e otimização de consultas em Data Warehouse utilizando Bancos de Dados Distribuídos com Replicação Heterogênea, não se encerram por aqui, o ambiente apresentado para a análise deste trabalho foi baseado em apenas duas tabelas, portanto, como contribuição teórica e prática para este trabalho, apresento as seguintes sugestões:

- Fazer a análise em um Banco de Dados com maior complexidade;
- Utilizar bancos heterogêneos com grandes diferenças entre os seus tipos de dados;
- Comparar o desempenho de bancos de dados distribuídos de diferentes arquiteturas.
- A fim de avaliar a performance relativa das consultas, efetuar uma série de execuções e registrar seus padrões (benchmark).

## Referências

- ALLROUND AUTOMATIONS. **PL/SQL developer**. Disponível em:  
<<http://www.sunsoftware.com.br/solucoes/ambiente.desenvolvimento/plsql.pdf>>. Acesso em:  
28 Set. 2013.
- ALLROUND AUTOMATIONS. **PL/SQL developer 7.1 user's guide**. Disponível em:  
<<http://www.allroundautomations.com/download/Manual71.pdf>>. Acesso em: 02 Out. 2013.
- DAMAS, L. **SQL: Structured Query Language**, 6 ed., Rio de Janeiro: LTC, 2007.
- DATE, C. J. **Introdução a sistemas de banco de dados**. 8 ed. Rio de Janeiro: Elsevier, 2003.
- ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 4. ed. Addison Wesley. São Paulo: Pearson, 2005.
- GIL, A. C. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002.
- INFO ONLINE. **Ponha o MySQL em seu desktop**. Disponível em:  
<<http://info.abril.com.br/dicas/arquivos/ponha-o-mysql-em-seu-desktop-749.shtml>>. Acesso em: 25 Out. 2013.
- ISO. **ISO/IEC 9075-11:2011**. Disponível em:  
<[http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=53685](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=53685)>. Acesso em: 24 Out. 2013.
- ITALIANO, I. **O Uso de visões materializadas em data warehouses**. 2000. 17 f. Dissertação (Tópicos em Ciência da Computação) - Universidade de São Paulo, São Paulo, 2000.
- JOSKO, J. M. B. **Pensando o desempenho no data warehouse**. Disponível em:  
<<http://www.devmedia.com.br/pensando-o-desempenho-no-data-warehouse/6940>>. Acesso em: 31 Out. 2013.
- MACHADO, F.. **Projeto de data warehouse: uma visão multidimensional**. São Paulo: Érica, 2000.
- MANNINO, M. **Projeto, desenvolvimento de aplicações & administração de banco de dados**. 3. ed. São Paulo: Mcgrawhil, 2008.
- MESQUITA, E. **Trabalhando com views no Oracle**. Disponível em:  
<<http://www.devmedia.com.br/trabalhando-com-views-no-oracle/26030>>. Acesso em: 13 Nov. 2013.

MICROSOFT. **ODBC visão geral da conectividade aberta de banco de dados**. Disponível em: <<http://support.microsoft.com/kb/110093/pt-br>>. Acesso em: 29 Set. 2013.

MORIMOTO, C. **TCO**. Disponível em: <<http://www.hardware.com.br/termos/tco>>. Acesso em: 24 Abril 2014.

NEVES, R. **Automatização do processo de criação de visões para modelagem de datawarehouse**. 2009. 99 f. Dissertação (Ciência da Computação) - Universidade Metodista de Piracicaba, São Paulo, 2009.

ORACLE. **A história do Oracle: inovação, liderança e resultados**. Disponível em: <<http://www.oracle.com/br/corporate/press/story-346137-ptb.html>>. Acesso em: 30 Set. 2013.

ORACLE. **Create database link**. Disponível em: <[http://docs.oracle.com/cd/B19306\\_01/server.102/b14200/statements\\_5005.htm](http://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_5005.htm)>. Acesso em: 28 Set. 2013.

ORACLE. **Database links**. Disponível em: <[http://docs.oracle.com/cd/B28359\\_01/server.111/b28310/ds\\_concepts002.htm](http://docs.oracle.com/cd/B28359_01/server.111/b28310/ds_concepts002.htm)>. Acesso em: 28 Set. 2013.

ORACLE. **Database SQL reference 10g release 2 (10.2)**. Disponível em: <[http://docs.oracle.com/cd/B19306\\_01/server.102/b14200/statements\\_6002.htm](http://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_6002.htm)>. Acesso em: 29 Set. 2013.

ORACLE. **Heterogeneous services concepts**. Disponível em: <[http://docs.oracle.com/cd/A84870\\_01/doc/server.816/a76960/hs\\_conce.htm](http://docs.oracle.com/cd/A84870_01/doc/server.816/a76960/hs_conce.htm)>. Acesso em: 28 Set. 2013.

PAROLIN, E. **Data warehouse aplicado ao negócio**, 2011. 71 f. Monografia (Tecnólogo em Processamento de Dados) – Faculdade de Tecnologia de São Paulo, São Paulo, 2011.

PATRICK, John J. **SQL fundamentos**. 2. ed. São Paulo: Editora Berkeley, 2002.

PICHILIANI, M. **O mito do melhor desempenho em banco de dados**. Disponível em: <<http://imasters.com.br/banco-de-dados/db2/o-mito-do-melhor-desempenho-em-banco-de-dados/>>. Acesso em: 30 Out. 2013.

PLSQL. **What is PL/SQL?** Disponível em <<http://plsql-tutorial.com/>>. Acesso em: 27 Set. 2013.

RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de gerenciamentos de bancos de dados**. 3. ed. São Paulo: McGraw Hill Artmed, 2008.

RICARTE, Ivan. **Banco de dados relacionais**. Disponível em: <<http://www.dca.fee.unicamp.br/cursos/PooJava/javadb/bdrel.html>>. Acesso em: 27 Set. 2013.



SANTOS, A. **MySQL: quem é você?** Disponível em:  
<[http://www.sqlmagazine.com.br/Colunistas/AdrianoSantos/02\\_MYSQL.asp](http://www.sqlmagazine.com.br/Colunistas/AdrianoSantos/02_MYSQL.asp)>. Acesso em:  
25 Out. 2013.

SILBERSCHATZ, A.; KORTH, H. e SUDARSHAN, S. **Sistema de banco de dados** 3. ed.  
Makron Books do Brasil Editora Ltda, 1999.

SUEHRING, Steve. **MySQL a bíblia**. 6. ed. Rio de Janeiro: Elsevier, 2002.

URMAN, Scott. **Programação PL/SQL**. Rio de Janeiro: Elsevier, 2002.

WATSON, R. **Data management** - Banco de Dados e Organizações. 3. ed. Rio de Janeiro:  
LTC – Livros técnicos e científicos Editora S.A., 2004.

ZIULKOSKI, L. **Coleta de requisitos e modelagem de dados para data warehouse**: um  
estudo de caso utilizando técnicas de aquisição de conhecimento. 2003. 63 f. Trabalho de  
conclusão de curso (Bacharelado em Ciências da Computação) – Universidade do Rio Grande  
do Sul, Porto Alegre, 2003.