

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Framework para dimensionamento de plataformas Oracle Retail

Paulo Ricardo Lemos Marques

VERSÃO DEFINITIVA

Relatório de Projecto
Mestrado Integrado em Engenharia Informática e Computação

Orientador: Gabriel David (Professor Associado do DEI)

Junho de 2009

Framework para dimensionamento de plataformas Oracle Retail

Paulo Ricardo Lemos Marques

Relatório de Projecto
Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: José Magalhães Cruz (Professor Auxiliar)

Arguente: José Manuel Moreira (Professor Auxiliar Convidado)

Vogal: Gabriel David (Professor Associado)

13 de Julho de 2009

Resumo

Nos últimos tempos, o comércio a retalho tem-se concentrado em grandes superfícies, que conjugam múltiplos sectores de negócio. Com o crescimento acentuado deste negócio, têm aumentado o número de transacções efectuadas, os produtos a disponibilizar e as pessoas e processos envolvidos. Este crescimento verificou-se também ao nível da informação gerada, que rapidamente ultrapassou os níveis aceitáveis a uma gestão exclusivamente humana. Assim, o recurso a Tecnologias de Informação, nomeadamente a implementação e o uso de Sistemas de Informação, tornou-se indispensável à sobrevivência das empresas detentoras de grandes superfícies de venda a retalho. A utilização destes sistemas permitiu-lhes reduzir custos, otimizar processos e, consequentemente, melhorar o serviço prestado aos clientes, resultando em aumentos de procura e de lucros.

Nesta área, um dos sistemas com maior preponderância é o Oracle Retail. Este consiste numa suite de produtos orientados à gestão de informação e de operações das mais diversas áreas relacionadas com o negócio do retalho. A implementação destes sistemas é um processo sensível e demorado, uma vez que altera todo o funcionamento do negócio e das infra-estruturas. Desta forma, exige-se um planeamento adequado, para que o processo decorra sem problemas e se atinjam os objectivos propostos. Uma das etapas deste planeamento é o dimensionamento das bases de dados que suportam estes sistemas.

O projecto descrito no presente relatório, realizado na Wipro Retail, uma das maiores empresas a nível mundial no ramo dos Sistemas de Informação para retalho, consiste na elaboração de uma ferramenta capaz de estimar as necessidades a nível de *hardware* para a implementação de um dos módulos da suite Oracle Retail, o Oracle Retail Merchandising System, que é o cerne de todo o sistema, concentrando as funcionalidades e a informação mais importantes e críticas de todo o negócio. A importância deste projecto reside na sensibilidade que esta parte do planeamento apresenta, pois um sistema com este nível de criticidade resulta normalmente num investimento avultado por parte dos clientes, muitas vezes excessivo, por razões de prudência, a menos que se conheçam com rigor as necessidades reais. Assim, este projecto pretende caracterizar essas necessidades, o que permitirá minimizar este investimento, mediante a aquisição de sistemas que permitam satisfazer os requisitos impostos.

A solução para o problema foi conseguida através de um estudo teórico sobre todo o sistema e as suas componentes. Foi analisado um questionário já existente, que visa sobretudo traduzir as dimensões do negócio do retalhista e as suas necessidades. Este questionário foi cruzado com o modelo de dados, tendo sido criada uma relação entre as respostas obtidas e os volumes de dados previstos nas tabelas da base de dados. Finalmente, efectuou-se uma recolha de estatísticas de casos práticos de sistemas implementados e em fase de produção, que permitiram criar fórmulas para o cálculo da estimativa do volume de dados da base de dados do sistema. Estas fórmulas têm como valores de entrada as respostas do questionário e operam com valores resultantes da manipulação das estatísticas obtidas dos casos reais, produzindo as estimativas desejadas. Este trabalho foi compilado numa ferramenta de dimensionamento.

Abstract

Lately, retail commerce has become concentrated in big surfaces that aggregate multiple business sectors. Following the marked growth of this business, the number of transactions made, the sellable products and the people and processes involved have risen. This raise has also been verified on levels of generated information, which has quickly passed the bearable limits of an exclusive human management. Therefore, the call on Information Technologies, namely the implementation and use of Information Systems, became indispensable to the survival of the companies holding big surfaces of retail. The use of these systems enabled cost reduction, process optimization and, thus, the improvement of customer service, which has led to bigger levels of demand and profit.

On this area, one of the most important systems is Oracle Retail. It consists on a product suite focused on operation and information management of many related areas concerning retailing business. The process of implementation of these systems is very delicate and slow, as it changes all business processes and infrastructures. Thus, it is mandatory to have an adequate planning, so the process can run seamlessly and all goals are matched. One of the stages of this planning is the sizing of the database which supports these systems.

The project described on this report, accomplished at Wipro Retail, one of the major companies worldwide on Information Systems for retailing, is mainly the creation of a tool capable of estimating the hardware requirements for the implementation of one of the modules from Oracle Retail suite, Oracle Retail Merchandising System, which is the centre of all the system, putting together all the features and information which are the most critical of this business. The importance of this project stands on the sensibility that this planning stage presents, as a system with this level of criticism normally results on a big investment by the clients, often excessive, for prudence purposes, unless the real needs are known. Therefore, this project intends to detail those needs, which will lead to a minimization of the investment, by acquiring systems that can satisfy the imposed requirements.

The solution for the problem was achieved by studying the theory surrounding the system and its components. An existing questionnaire was analysed, which aims to get a notion of the business requirements and dimensions. This questionnaire was crossed with the data model, resulting in the creation of a relationship between the given answers and the volume of data predicted on the database's tables. Finally, some statistics were collected from practical cases of already implemented systems and in live phase, which allowed the creation of mathematical formulas for the calculation of the estimation of the data volume of the system's database. These formulas have the answers from the questionnaire as an input and operate with values resulting from the manipulation of the statistics, resulting in the desired estimates. This work was then compiled in a sizing tool.

Agradecimentos

Em primeiro lugar quero agradecer às instituições a que este projecto está afecto, a Faculdade de Engenharia da Universidade do Porto e a Wipro Retail, pois sem o seu contributo e as condições oferecidas, este trabalho não teria sido realizado.

A nível mais pessoal, um agradecimento especial à pessoa que me acompanhou na empresa, Rui Collaço, por todo o apoio, orientação e motivação dados, tendo sido uma ajuda muito preciosa e crítica na elaboração de todo o trabalho e deste documento.

Ao Professor Gabriel David, por todo o interesse demonstrado e por toda a ajuda prestada, quer nas linhas orientadoras do trabalho, quer na revisão e correcção deste documento.

A todos os estagiários na Wipro, pelos bons momentos proporcionados, que trouxeram a motivação necessária para levar a cabo todo este projecto.

Finalmente, um agradecimento especial aos meus familiares e amigos, por toda a força que me deram, desde o início do curso, até esta fase final, inclusive, sendo que sem eles tudo teria sido mais complicado e árduo.

O Autor

Índice

1	Introdução	1
1.1	Contexto/Enquadramento	1
1.1.1	A empresa	2
1.1.2	O negócio	3
1.1.3	Os maiores clientes	5
1.2	Tema do Projecto	6
1.3	Motivação e Objectivos	6
1.4	Estrutura da Dissertação	6
2	Revisão Bibliográfica.....	7
2.1	As bases de dados Oracle	7
2.2	O armazenamento nas bases de dados	7
2.2.1	Estruturas físicas	7
2.2.2	Estruturas lógicas	8
2.2.3	Índices	9
2.2.4	O dicionário de dados	11
2.2.5	A gestão do Undo.....	11
2.2.6	A gestão do <i>redo log</i>	12
2.3	A gestão da memória	13
2.3.1	System Global Area	13
2.3.2	Program Global Area	13
2.3.3	A gestão automática de memória	14
2.4	O dimensionamento e planeamento de capacidade	15
2.4.1	Amostras de dados e extrapolação	15
2.4.2	Tamanhos dos <i>datafiles</i>	16
2.4.3	Conteúdos dos <i>datafiles</i>	16
2.4.4	Uso do <i>package</i> DBMS_SPACE	16
2.4.5	Recolha de estatísticas	17
2.4.6	Tamanho exacto dos dados das colunas.....	17
2.5	Tecnologias de redundância de dados	18
2.5.1	RAID 0.....	18
2.5.2	RAID 1	18
2.5.3	RAID 5	19
2.6	Resumo e conclusões	19
3	Caracterização do problema.....	20
3.1	Oracle Retail	20
3.2	Descrição e objectivos principais	23
3.3	Oracle Retail Merchandising System	24
3.3.1	Dados fundamentais.....	24
3.3.2	Controlo de inventário	26
3.3.3	Compras	26
3.3.4	Contratos e negócios	27

3.3.5	Controlo de custos.....	27
3.3.6	Reposição.....	27
3.3.7	Gestão financeira	27
3.4	Modelo de dados do ORMS	28
3.4.1	Dados fundamentais.....	28
3.4.2	Manutenção de itens	28
3.4.3	Manutenção de custos	29
3.4.4	Manutenção de negócios.....	29
3.4.5	Ordens de compra	30
3.4.6	Gestão de inventário	30
3.4.7	Reposições	30
3.4.8	Stock Ledger	31
3.4.9	Variáveis de sistema	31
3.5	Processamentos de retaguarda	31
3.5.1	Tipos de programas.....	32
3.5.2	Janela de execução	32
3.5.3	Agendamento e fases	32
3.5.4	Processamento paralelo.....	33
3.6	Análise de casos de estudo	33
3.6.1	Caso 1.....	34
3.6.2	Caso 2.....	34
3.6.3	Caso 3.....	35
3.6.4	Caso 4.....	35
3.7	Resumo e conclusões	35
4	Desenvolvimento da solução	36
4.1	Metodologia de dimensionamento adoptada	36
4.1.1	Armazenamento físico	36
4.1.2	Carga de processador	37
4.1.3	Memória.....	37
4.2	Etapas de elaboração da solução.....	38
4.2.1	Aquisição de conhecimento teórico	38
4.2.2	Formação em retalho.....	39
4.2.3	Análise de questionário e modelo de dados	39
4.2.4	Recolha de estatísticas de casos práticos	52
4.3	Caracterização e apresentação da ferramenta desenvolvida	55
4.4	Resumo e conclusões	56
5	Conclusões e Trabalho Futuro.....	57
5.1	Retrospectiva	57
5.2	Satisfação de objectivos.....	58
5.3	Trabalho Futuro	58
Anexo A:	Modelo de dados do ORMS.....	59
A.1	Dados fundamentais.....	59
A.2	Manutenção de artigos.....	61
A.3	Manutenção de custos.....	64
A.4	Manutenção de negócios.....	64
A.5	Ordens de Compra	65
A.6	Gestão de inventário	65
A.7	Reposição.....	66
A.8	Stock Ledger	66
A.9	Variáveis de sistema	67
Anexo B:	Questionário da Oracle de dimensionamento.....	68
Referências.....		79

Lista de Figuras

Figura 1 – Países onde a Wipro Retail tem clientes	2
Figura 2 – Áreas funcionais do negócio de venda a retalho	3
Figura 3 – Áreas de negócio suportadas pelo Oracle Retail	4
Figura 4 – Alguns clientes da carteira da Wipro Retail	5
Figura 5 – Diferença entre árvores binárias e <i>B-Tree</i>	10
Figura 6 – Folha de árvore e o seu ramo (a) e a nova estrutura após inserção do valor 72 (b) ...	10
Figura 7 – Exemplo de uma <i>B-Tree</i>	11
Figura 8 – Diagrama representativo das áreas de memória	14
Figura 9 – Diagrama dos diversos módulos Oracle Retail e interações	241
Figura 10 – Diagrama representativo das diversas áreas funcionais do negócio	242
Figura 11 – Ecrã inicial do sistema	24
Figura 12 – Excerto da vista <i>restart_control</i>	37
Figura 13 – Etapas de desenvolvimento da solução	38
Figura 14 – Excerto do questionário de dimensionamento	39
Figura 15 – Exemplo de exercício de mapeamento	40
Figura 16 – Operações possíveis sobre ordens de compra	45
Figura 17 – Exemplo de estatísticas recolhidas para tabelas	53
Figura 18 – Exemplo de estatísticas recolhidas para os índices	53
Figura 19 – Diagrama de fluxo representativo do algoritmo	54
Figura 20 – Excerto da ferramenta desenvolvida, cálculo das tabelas	55
Figura 21 – Excerto da ferramenta desenvolvida, parte dos índices	56

Lista de Tabelas

Tabela 1 – Descrição dos diferentes tipos de <i>segments</i>	9
Tabela 2 – Hierarquia organizacional	25
Tabela 3 – Hierarquia mercadológica	26
Tabela 4 – Dados relativos ao caso 1	34
Tabela 5 – Dados relativos ao caso 2	34
Tabela 6 – Dados relativos ao caso 3	35
Tabela 7 – Dados relativos ao caso 4	35

Abreviaturas e Símbolos

BD	Base de Dados
IVA	Imposto sobre Valor Acrescentado
OR	Oracle Retail
ORMS	Oracle Retail Merchandising System
UDA	User-Defined Attributes
RPM	Retail Price Management

1 Introdução

O negócio de retalho consiste na venda de produtos, quer sejam eles mercearias, roupas, electrodomésticos ou materiais de construção, numa área fixa criada para esse efeito, normalmente chamada de “loja” [WIKI09a]. Os consumidores podem ser pessoas individuais, comprando para consumo próprio, ou representantes de um qualquer negócio, que pela sua dimensão não conseguem contratos vantajosos com os fornecedores e encontram nos retalhistas os produtos a preços que lhes permitem acrescentar uma margem de lucro.

Os retalhistas adquirem os seus produtos em fornecedores de grande escala, geralmente os representantes da marca, evitando ao máximo os intermediários, para assim diminuir o preço pago pelos bens. A sua estratégia de venda passa por adicionar ao custo do produto uma margem de lucro, que varia consoante o tipo de produto, a marca, o público-alvo ou mesmo as cláusulas do contrato firmado com o fornecedor. Ao longo dos anos, esta área de negócio sofreu uma grande evolução, devido à melhoria dos serviços prestados, aos preços geralmente económicos e à facilidade que representa para o consumidor, uma vez que com uma viagem a um único sítio consegue suprir grande parte das suas necessidades de produtos.

Este crescimento acentuado do volume de negócios dos retalhistas provocou um aumento da informação associada a esta actividade, que rapidamente ultrapassou os níveis aceitáveis a uma gestão humana. Controlar fornecedores, contratos, encomendas, distribuição pelas lojas, clientes, preços, promoções e demais aspectos inerentes a esta actividade é uma tarefa complexa e morosa para ser feita apenas com recurso a ficheiros de arquivo em papel. Desta forma, a necessidade de implementar Sistemas de Informação para o processamento de todos estes dados tornou-se evidente, como que uma condição *sine qua non* para o funcionamento e subsistência dos grandes retalhistas.

1.1 Contexto/Enquadramento

O projecto desenvolvido insere-se na área do planeamento da implementação de sistemas Oracle Retail nos clientes da Wipro Retail.

1.1.1 A empresa

A Wipro Retail nasce da aquisição da Enabler pela Wipro Technologies, em 2006. A Enabler era anteriormente uma empresa que resultou da autonomização do departamento de Sistemas de Informação da Modelo Continente, um dos maiores retalhistas em Portugal, cuja actividade principal era a implementação e gestão de toda a estrutura de Sistemas de Informação da Modelo Continente. Partindo da implementação de uma solução informática base para a gestão do negócio de retalho nas suas várias vertentes (na altura Retek, agora conhecido como Oracle Retail) foi evoluindo gradualmente, sofrendo diversas adaptações ao longo dos anos, constituindo o suporte actual do sistema informático em produção. Esta actividade, dotada de especial envergadura e complexidade, permitiu adquirir uma vasta experiência e competência nesta área dos sistemas de venda a retalho, bem como na implementação do produto Oracle Retail, o que possibilitou a criação da empresa do grupo Sonae, Enabler, com o principal objectivo de expandir toda esta base de conhecimento a diversos clientes no resto do mundo e competir entre os principais implementadores deste tipo de soluções.

Assim, a Enabler focou-se nos retalhistas líderes dos maiores mercados europeus, registando um crescimento significativo e sustentável, que a conduziram a elevados níveis de vendas e de performance. A empresa especializou-se na suite de produtos Oracle Retail, entre os quais o Retail Merchandise System, que é a base de suporte ao *online*, tornando-se numa consultora de referência na área dos Sistemas de Informação para venda a retalho a nível Europeu, tendo para isto contribuído a elevada preocupação com a qualidade dos serviços prestados, sendo a certificação de qualidade ISO 9001:2000 o princípio orientador. Durante cerca de 9 anos, a Enabler expandiu-se um pouco por todo o Mundo, contando na sua carteira de clientes com os maiores retalhistas de países como Reino Unido, Itália, Espanha, Alemanha e Brasil.

No seguimento do processo normal de crescimento da empresa, num mercado dominado por implementadores de grande dimensão, como por exemplo a Accenture, impunha-se um ganho extraordinário de competitividade. A aquisição por parte da Wipro Technologies veio responder a essa questão. A Wipro Technologies é uma empresa multinacional sediada em Bangalore, Índia, com cerca de 90.000 colaboradores e cuja principal actividade é a aplicação das Tecnologias de Informação às mais variadas áreas de negócio. Após a aquisição da Enabler, a Wipro Technologies passou a ter uma forte representação na área de retalho, denominada Wipro Retail. A constituição desta nova empresa permitiu uma representação mais forte junto de retalhistas cada vez mais importantes, resultando num aumento do volume de negócios e natural crescimento. Actualmente conta com projectos em todo o Mundo, representados na figura abaixo.

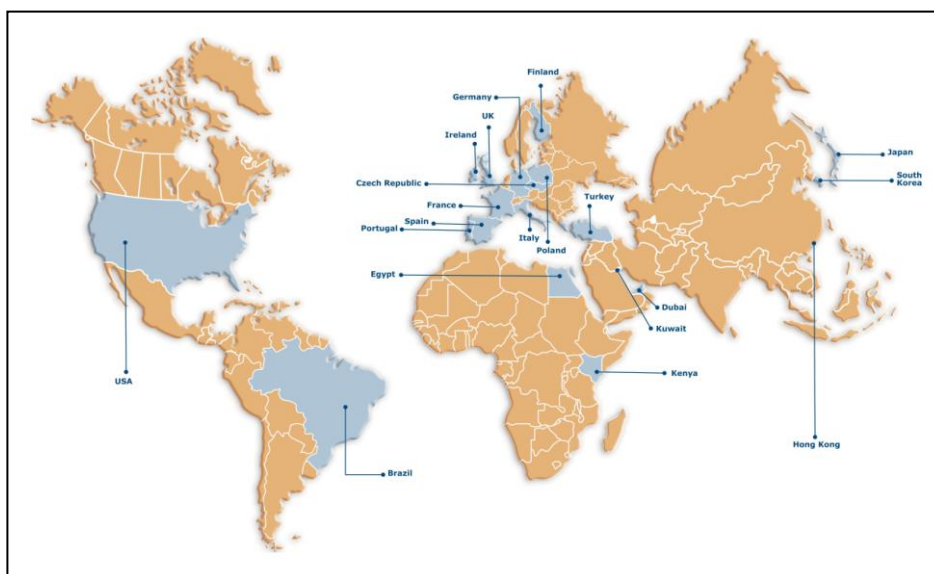


Figura 1 – Países onde a Wipro Retail tem clientes

1.1.2 O negócio

A actividade da Wipro Retail consiste em diversas operações, desde o aconselhamento dos clientes até à implementação, manutenção e actualização dos produtos da suite Oracle Retail adquiridos pelos retalhistas. Estabelecendo relações de parceria com estes, a empresa transmite todo o seu conhecimento e meios técnicos, no sentido de criar soluções à medida de cada caso, conseguindo otimizar processos de Sistemas de Informação, transformando infra-estruturas complexas em ferramentas de suporte orientadas ao lucro, o que resulta na criação de vantagens competitivas para os seus clientes, que desta forma aumentam os seus níveis de vendas, reduzem custos e mostram-se mais eficientes no cumprimento das necessidades dos seus consumidores.

De acordo com a figura abaixo, pode ter-se uma ideia generalizada das áreas funcionais de um retalhista sobre as quais a Wipro opera.

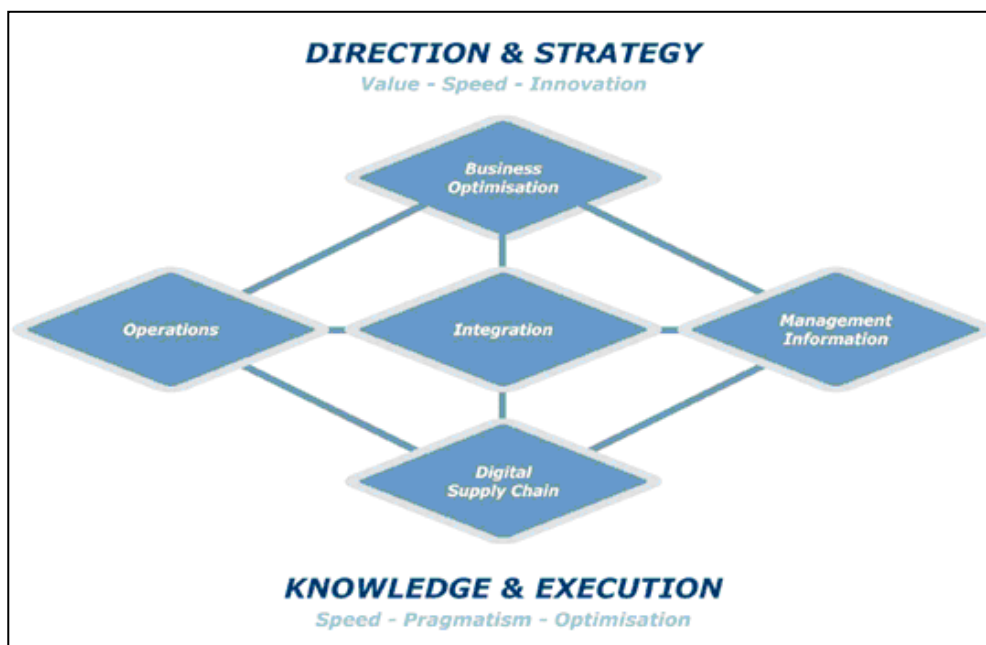


Figura 2 – Áreas funcionais do negócio de venda a retalho

- **Business Optimization** – utilizando tecnologias dominadas pela empresa, e aplicando todo o conhecimento sobre vendas a retalho adquirido ao longo dos anos, os processos de negócio dos retalhistas são otimizados;
- **Operations** – envolve a utilização de ferramentas que suportam a gestão das operações relacionadas com o normal funcionamento do negócio, otimizando e automatizando processos;
- **Digital Supply Chain** – a acção da Wipro resulta numa melhoria da cadeia de distribuição, ao gerir e agilizar as actividades e operações que dela fazem parte;
- **Management Information** – toda a informação que circula dentro da organização, resultante das actividades das áreas descritas, é gerida centralmente, estando disponível e acessível a todas as unidades funcionais;
- **Integration** – as diversas áreas funcionais estão devidamente integradas, havendo circulação de dados entre todas elas, estando os processos propriamente organizados e sincronizados.

A suite de produtos Oracle Retail abrange todas as operações e áreas funcionais de um retalhista, desde a gestão de *stocks* e armazéns, à reposição de produtos, ao planeamento de actividades, à gestão financeira, até à cadeia de distribuição, entre outras. Analisando o diagrama seguinte pode ter-se uma ideia geral das áreas abrangidas pelos produtos Oracle Retail.

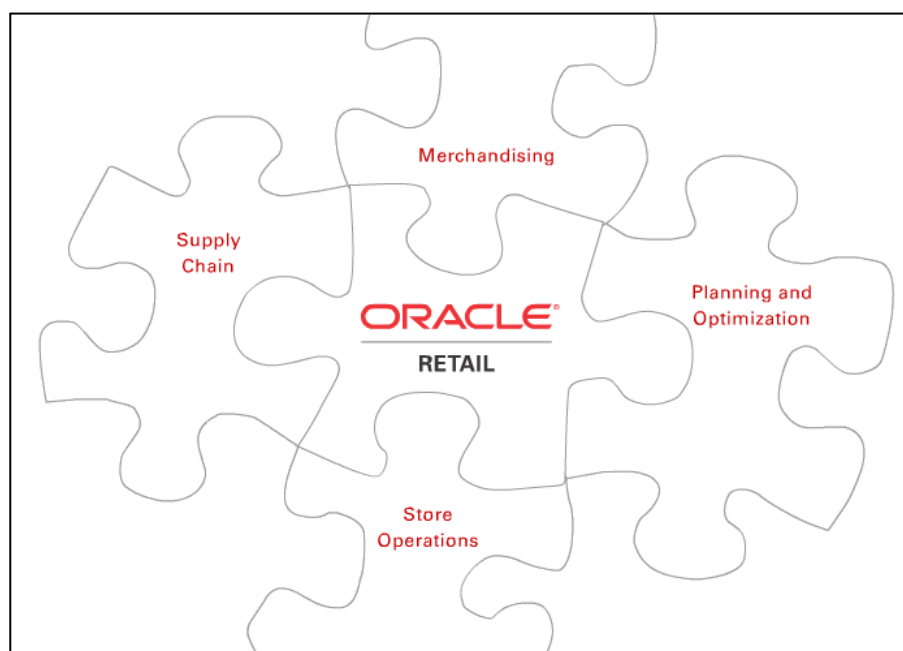


Figura 3 – Áreas de negócio suportadas pelo Oracle Retail

Além da actividade principal de implementação destas soluções, a Wipro oferece ainda outros serviços aos seus clientes, apresentando-se como a única entidade a que os clientes necessitam de recorrer para satisfazer todas as suas necessidades de Sistemas de Informação, tais como:

- **Suporte de aplicações** – envolve o aconselhamento, o planeamento e a implementação das soluções Oracle Retail requisitadas pelo cliente;
- **Recuperação do *batch* nocturno** – consiste numa actividade de constante monitorização dos processos de manutenção da base de dados que ocorrem numa janela de tempo em que não há actividade dos utilizadores, e na resolução de eventuais problemas com este *batch*;
- **Manutenção de aplicações** – as equipas da Wipro Retail comprometem-se a resolver *bugs* das aplicações, planear e gerir novas versões do produto implementado, analisar *patches* de correcção e efectuar revisões periódicas ao funcionamento de todo o sistema;
- **Actualização de produtos Oracle Retail** – baseia-se na implementação de novas versões dos produtos e em revisões ao estado do sistema;
- **Administração de bases de dados Oracle** – como a suite Oracle Retail está suportada por uma base de dados Oracle, a Wipro efectua operações de administração destas bases de dados nos clientes;
- **Consultadoria** – os clientes podem ser aconselhados, no sentido de desenvolver um centro de ajuda e suporte, melhorar a sua gestão de *hardware*, *software* ou *networking* ou ainda melhorar a sua arquitectura para a tornar mais eficiente a nível de custos;

- **Outros serviços** – a Wipro Retail executa ainda operações de manutenção de *hardware*, *network* e outros sistemas dos clientes, qualquer que tenha sido a entidade responsável pela instalação.

1.1.3 Os maiores clientes

A carteira de clientes da Wipro Retail conta com alguns dos maiores retalhistas a nível mundial, distribuídos pelas mais diversas áreas de negócio. Uma amostra destes pode ser encontrada na figura abaixo.



Figura 4 – Alguns clientes da carteira da Wipro Retail

O volume de negócios e a dimensão de algumas destas organizações retalhistas é elevado, conduzindo a volumes de informação que põem à prova as capacidades e a performance dos sistemas mais avançados na área, tais como os da Oracle. Englobando milhares de lojas e fornecedores, milhões de artigos, sem deixar de fora os milhares de transacções efectuadas diariamente, a gestão de todos os dados gerados, consultados e enviados de ponto para ponto torna-se um caso sério no domínio dos Sistemas de Informação, resultando em bases de dados com um número de registos e necessidades de *hardware* acima dos níveis vulgares, sendo mesmo algumas alvo de estudo por parte de fabricantes de sistemas, no sentido de melhorar a performance dos seus produtos.

1.2 Tema do Projecto

A implementação de soluções Oracle Retail constitui-se como uma actividade muito sensível, na medida em que altera por completo o funcionamento normal da entidade adquirente. Assim, existe um planeamento a ser feito que engloba o levantamento dos requisitos de negócio, a identificação dos módulos a instalar, a implementação propriamente dita, os testes ao sistema e a sua posterior manutenção. Este planeamento contém uma vertente que consiste no dimensionamento da base de dados que serve de suporte à aplicação. O dimensionamento abrange os servidores de bases de dados, ao nível de armazenamento, memória e processamento, resultando na especificação dos equipamentos a obter, que consigam garantir a performance esperada pelo cliente.

1.3 Motivação e Objectivos

O projecto descrito neste relatório é resultado de uma necessidade da Wipro em adoptar uma posição crítica face a um dimensionamento proposto pelo fabricante do *software*, neste caso a Oracle.

O exercício de dimensionamento da Oracle consiste no preenchimento de um questionário, pelo cliente, onde este expressa a dimensão da sua organização, os volumes transaccionais e outros requisitos do seu negócio. Este questionário, além de extenso, contém perguntas que um retalhista poderá não ser capaz de responder, e que para o efeito, não são relevantes, uma vez que o seu contributo poderá ser estimado recorrendo às respostas às perguntas mais fulcrais.

Assim, o objectivo deste projecto é produzir uma ferramenta de dimensionamento, reformulando o questionário existente e calcular as necessidades de *hardware* para a implementação do sistema Oracle Retail, apoiado nas respostas do cliente, que representam o seu volume de negócios e que podem ser convertidas em volume de dados. A ferramenta deverá conduzir a resultados semelhantes, e ser o mais autónoma possível, minimizando a necessidade de ajustes. O produto deste projecto deverá permitir aos colaboradores da Wipro compreender melhor o procedimento de dimensionamento das bases de dados, podendo criticar uma proposta do fabricante e ajustá-la às necessidades reais dos clientes, uma vez que as conhecem melhor e mais de perto, removendo o carácter generalista das soluções propostas pela Oracle.

Esta ferramenta justifica-se devido às necessidades geralmente apresentadas por projectos de implementação de produtos Oracle Retail serem elevadas e acima dos níveis vulgares, resultando em investimentos avultados por parte dos clientes, pelo que revelou-se importante aumentar o rigor na determinação dos equipamentos a adquirir, afim de reduzir os custos do cliente em Tecnologias de Informação.

1.4 Estrutura da Dissertação

Para além da introdução, este relatório contém mais 4 capítulos. No capítulo 2 é efectuada uma revisão bibliográfica de algum material teórico de suporte ao problema em questão, que permitirá uma melhor compreensão dos conceitos associados. Segue-se o capítulo 3, onde se faz uma caracterização do problema, apresentando-se o sistema envolvido, uma descrição do problema resolver e outros aspectos necessários para a sua compreensão. No capítulo 4, explica-se o desenvolvimento da solução, detalhando todas as etapas seguidas para a sua conclusão, desde a formulação até à criação da ferramenta de desenvolvimento. Finaliza-se o documento com um capítulo dedicado a conclusões e trabalho futuro, onde se realiza uma retrospectiva do projecto, apontando os detalhes mais preponderantes, seguindo-se uma análise aos objectivos propostos e os realmente atingidos, terminando com algumas sugestões para um desenvolvimento futuro do trabalho efectuado.

2 Revisão Bibliográfica

Neste capítulo faz-se uma apresentação de alguma informação de suporte teórico sobre o funcionamento das bases de dados Oracle, desde as estruturas envolvidas, até à gestão da memória, seguindo-se uma revisão a algumas técnicas de dimensionamento de bases de dados em prática na área.

2.1 As bases de dados Oracle

Uma base de dados consiste numa colecção de dados tratados como um todo. A sua principal função é armazenar e apresentar informação que está de alguma forma relacionada entre si. A gestão de toda esta informação é assegurada por um servidor de bases de dados, no qual é permitido guardar e classificar os dados em causa. Para além disso, estes servidores têm ainda a capacidade de possibilitar o acesso simultâneo de diferentes utilizadores à mesma informação, facilitando-lhes a leitura, a alteração, a introdução e a remoção de dados, sempre garantindo a máxima segurança e fornecendo processos de recuperação de falhas. As diferentes necessidades destes utilizadores, e os seus interesses sobre a informação que consta da base de dados são orientados por um mecanismo de permissões, que gere os acessos de cada utilizador, deixando-o operar apenas sobre os dados a que está habilitado pelo administrador, sendo que poderá ser-lhe barrada a escrita e alteração de uns, ou a leitura de outros, conforme o caso.

2.2 O armazenamento nas bases de dados

Um sistema deste tipo está organizado em estruturas lógicas e estruturas físicas, que se apresentam independentes umas das outras, pelo que a sua gestão poderá ser feita, sem o risco de existirem influências mútuas. São estas estruturas que guardam a informação e que permitem a sua leitura e classificação.

2.2.1 Estruturas físicas

As estruturas físicas das BD Oracle são os *datafiles*, *control files*, *redo log files*, *archive log files*, *parameter files*, *alert* e *trace log files*, e os *backup files*.

Os **datafiles** são os ficheiros que armazenam fisicamente os dados da base de dados, e todos os dados das estruturas lógicas, como tabelas e índices. Um ficheiro deste tipo só pode estar associado a uma única BD, e possui características que lhe permite expandir-se automaticamente, caso a BD fique sem espaço livre. Um conjunto de um ou mais ficheiros deste tipo forma uma unidade lógica chamada *tablespace*, que funciona como um classificador das diferentes estruturas lógicas de uma BD.

Para melhorar a performance e reduzir a frequência de operações de escrita e leitura de discos, os dados são guardados em memória e escritos para o disco em bloco, sendo tudo controlado por um processo de segundo plano, o *database writer* (DBWn).

Os **control files** existem em todas as BD Oracle e contêm informação relativa à BD, tal como o nome da própria BD, as localizações e os nomes dos *datafiles* ou dos *redo log files*, e a data e hora de criação da BD. Estes ficheiros são actualizados sempre que são feitas mudanças à constituição física da BD, como a adição de ficheiros. São também importantes numa operação de recuperação de dados, pois identificam os ficheiros necessários ao processo e apresentam os diversos *checkpoints*, para que se saiba com exactidão quais os dados já armazenados e que não necessitam de ser lidos dos *redo log files*. Os factores que determinam o tamanho destes ficheiros são o número máximo de *datafiles*, *log files*, instâncias, membros do *log* e máximo de histórico de *log*.

Qualquer BD Oracle possui dois ou mais **redo log files**. Estes ficheiros têm como função principal armazenar todas as alterações feitas à BD. Em caso de falha, que não permita a escrita para os *datafiles*, as mudanças estão todas nestes ficheiros, salvando todo o trabalho realizado até ao momento. Para protecção destes dados, é permitido manter cópias destes ficheiros em diferentes suportes de armazenamento. Se a opção de arquivamento estiver activa, estes ficheiros são arquivados quando atingem um determinado tamanho. Depois de arquivados, tomam a forma de **archive log files**.

Quando é criada uma BD, existe uma série de parâmetros a serem definidos, que configuram a BD à medida das exigências dos utilizadores e das funções que terá que desempenhar. Estes parâmetros são guardados em **parameter files**, para que estejam sempre acessíveis e possam ser modificados em qualquer altura.

Cada servidor e cada processo de segundo plano pode escrever para um **trace file** que lhe está associado. Estes ficheiros recebem informação quando ocorre um erro interno, sendo escritos dados sobre o erro para este ficheiro. A informação destes ficheiros destina-se ao administrador da BD e/ou aos serviços de suporte da Oracle, para a identificação e posterior resolução do problema, servindo ainda para ajustar aplicações e instancias. Neste contexto, existe também o **alert log**, que é um *trace file* especial. Estando associado à BD, apresenta um registo cronológico de todas as mensagens e erros ocorridos.

Para restaurar um ficheiro, este é substituído por um **backup file**. Estes ficheiros são cópias de segurança da BD efectuadas com uma frequência definida pelo administrador. Geralmente, o restauro de ficheiros é feito quando ocorre uma falha nos suportes de armazenamento, ou um erro de utilizador, que danificam ou eliminam o ficheiro original. O processo de cópia de segurança e de restauro de ficheiros pode ser gerido pelo utilizador ou pelo servidor.

2.2.2 Estruturas lógicas

As estruturas lógicas permitem à BD ter um controlo mais minucioso sobre o uso do espaço em disco. Estas estruturas são os *tablespaces*, os *data blocks*, os *segments* e os *extents*.

As BD Oracle estão divididas em unidades lógicas de armazenamento, chamadas **tablespaces**. Estas agrupam estruturas lógicas que se relacionam entre si, facilitando a classificação e assim algumas tarefas de administração. Os dados que constam destas estruturas são guardados fisicamente nos *datafiles* associados, sendo a soma de todos os ficheiros o

tamanho total do *tablespace*. Podem existir **smallfile tablespaces**, constituídos por diversos *datafiles* de tamanho reduzido, ou **bigfile tablespaces**, que são constituídos por um ficheiro singular de tamanho elevado, podendo atingir 8 exabytes. Os *tablespaces* podem ainda estar **online** ou **offline**, consoante o administrador os queira acessíveis ou não, respectivamente. Isto facilita inúmeras operações de administração da BD.

Os dados que constam das BD Oracle são guardados em **data blocks**, que são a estrutura de armazenamento mais elementar, constituída por um conjunto de bytes de espaço físico em disco, definidos na criação da BD. É nestes blocos que a BD usa e aloca espaço livre.

Em seguida aparecem os **extents**, que são um número específico de *data blocks* contíguos, obtidos com uma única alocação, para o alojamento de um tipo de dados específico.

No nível mais elevado de armazenamento aparecem os **segments**, que são um conjunto das estruturas anteriores, os *extents*. De acordo com o tipo de dados a armazenar, existem diferentes tipos de *segments*, detalhados na tabela seguinte.

Tabela 1 – Descrição dos diferentes tipos de *segments*

Segment	Descrição
Data segment	Estes <i>segments</i> armazenam dados de tabelas simples, de partições de tabelas e de <i>clusters</i> de tabelas.
Index segment	Aqui são guardados os dados referentes a todos os índices da BD.
Temporary segment	Este tipo de <i>segment</i> é utilizado por algumas operações SQL que necessitam de espaço temporário de trabalho, normalmente operações de ordenação. Apenas são utilizados caso o trabalho não possa ser feito em memória ou com recurso a índices.
Rollback segment	Em edições anteriores, a Oracle usava estes <i>segments</i> para armazenar informação sobre as mudanças feitas à BD, para que uma recuperação de dados posterior fosse possível. No entanto, a gestão desse processo é agora automática, tendo por isso este tipo de <i>segment</i> sido deprecado.

2.2.3 Índices

Um índice apresenta-se como um conjunto ordenado de elementos numa forma (x, α) , em que x é a chave de identificação e α a informação associada a essa chave. A informação contida em α poderá ser a que se procura, caso seja de tamanho reduzido, ou poderá ser um apontador para esta, sendo que este caso implica a existência de um outro nível de endereçamento. Os índices estão ordenados pelas chaves armazenadas em x [SCHO08].

O armazenamento dos índices é feito numa estrutura de dados lógica em árvore, de nome *B-tree*. Esta estrutura de dados, criada em 1969, é amplamente conhecida no mundo da Informática, devido às grandes vantagens que oferece na organização e gestão de volumes elevados de informação, normalmente residentes em suportes físicos de acesso aleatório. O acesso aos dados é independente do tamanho total do conjunto, uma vez que o crescimento do índice e do tempo de recolha apenas cresce numa proporção logarítmica ao tamanho total da informação.

Por ser uma estrutura que se organiza automaticamente após qualquer operação de adição ou subtração de dados, a *B-tree* consegue suportar os ambientes mais dinâmicos, em que a informação sofre constantes e frequentes mudanças.

Esta estrutura caracteriza-se por ser uma árvore de pesquisa com vários caminhos possíveis, sendo que uma visita em pré-ordem resulta vai retornando as chaves de forma ordenada, quer seja crescente ou decrescente. Para se encontrar uma chave e a informação associada, inicia-se a visita no nó de raiz e em cada nível examina-se um filho que direcciona a pesquisa no sentido do nó que contém a informação desejada. Tipicamente, uma árvore desta categoria contém entre 3 a 5 níveis, sendo que o pior caso de pesquisa seria a visita de 5 nós, pelo que se pode inferir a rapidez de acesso a dados proporcionada por esta estrutura de dados [SCHO08].

A principal característica das *B-trees* resume-se aos seus métodos de inserção e remoção de dados, que deixam a árvore sempre equilibrada. Nas árvores de pesquisa binárias, a inserção aleatória de dados pode deixar a árvore desequilibrada, conduzindo à existência de caminhos muito longos desde a raiz até às folhas, ao passo que uma *B-tree* tem as folhas sempre no mesmo nível de profundidade, devido à sua constante organização.

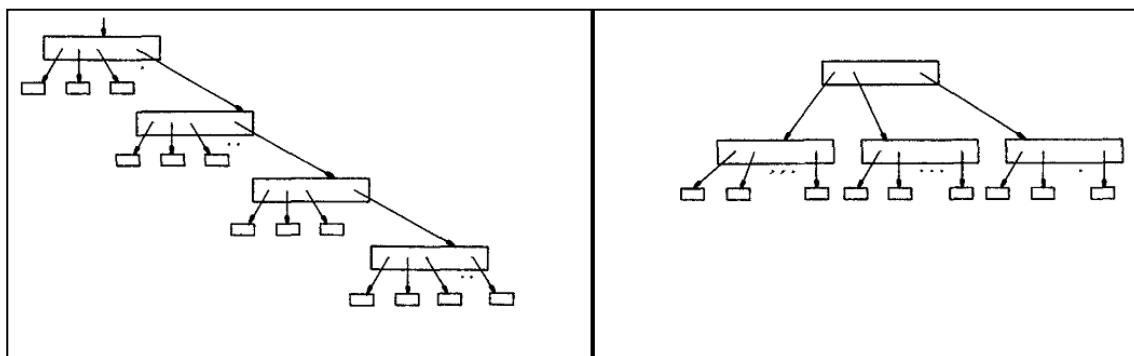


Figura 5 – Diferença entre árvores binárias e *B-Tree*

O método de inserção de dados consiste num processo de divisão dos nós. No início, existe apenas um nó na árvore, cujos elementos contidos estão ordenados. Quando um nó atinge a sua capacidade máxima, é dividido em dois nós-folha, e o elemento do meio é inserido no nó imediatamente acima, devidamente ordenado. Caso esse nó também já esteja cheio, o mesmo processo é seguido recursivamente, mesmo que se trate do próprio nó-raiz, sendo que neste caso a altura da árvore é incrementada.

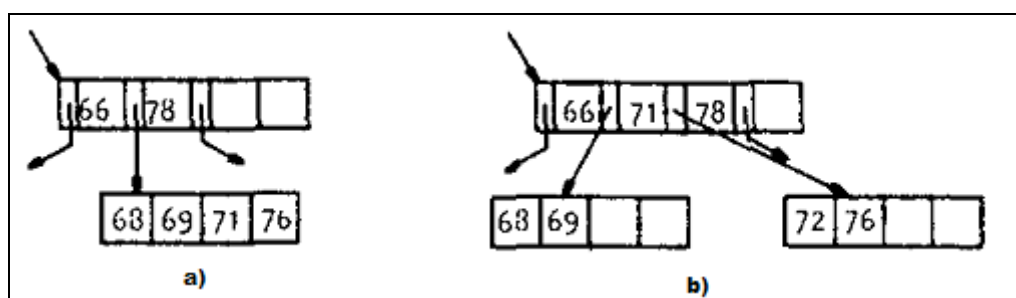


Figura 6 – Folha de árvore e o seu ramo (a) e a nova estrutura após inserção do valor 72 (b)

O método de remoção de dados também garante o equilíbrio da árvore. Depois de localizado o valor a apagar, existe 2 possibilidades: ou o valor reside num nó-folha, ou reside num nó intermédio. Remover de um nó-folha é simples e não implica qualquer alteração na estrutura; já num nó intermédio, há a necessidade de encontrar uma chave que ocupe o lugar agora vago. Esta chave encontra-se na folha mais à esquerda na sub-árvore direita¹ do espaço vazio [DOUG79].

¹ Uma sub-árvore resulta da extracção de um nó de uma árvore e de todos os nós que dele descendem.

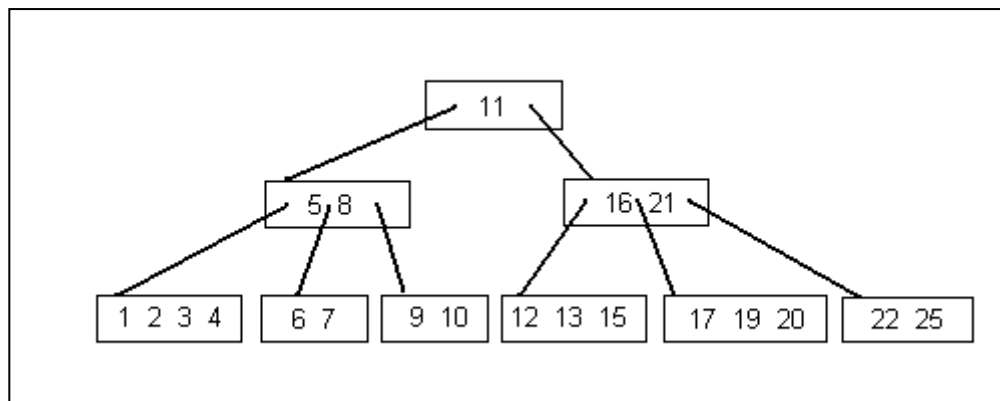


Figura 7 – Exemplo de uma B-Tree

O armazenamento dos índices pode ser feito em outras estruturas lógicas de dados, sendo a *B-tree* de longe a mais utilizada, pelo que não serão aqui detalhadas essas estruturas. O armazenamento físico das estruturas que suportam os índices é realizado em *index segments*. Todos os índices simples têm um *index segment* associado para armazenar todos os seus dados. Os índices particionados têm um *index segment* associado a cada uma das suas partições.

O espaço ocupado por um índice é aproximadamente metade do tamanho do bloco definido, por cada registo que dele faça parte. Quando se cria um índice, os dados são recolhidos e ordenados, sendo também guardados os seus *rowid* e o valor de índice.

2.2.4 O dicionário de dados

Uma das partes mais importantes de uma BD Oracle é o seu dicionário de dados. Apresenta-se como um conjunto de tabelas apenas de leitura, onde reside informação sobre a BD, tal como definições de todos os objectos, espaço alocado para estes, valores por defeito para as colunas, informação de integridade, nomes dos utilizadores, privilégios, informação de auditoria e outras informações gerais sobre a BD. Todas as tabelas e vistas que compõem o dicionário de dados estão armazenadas no *tablespace* do sistema (SYSTEM).

O objectivo da utilização deste dicionário é a recolha de informação sobre utilizadores, objectos e estruturas de armazenamento, a par de outras informações sobre a BD que sejam úteis ao utilizador.

2.2.5 A gestão do Undo

A BD mantém armazenada informação que permita anular todas as alterações feitas aos dados das suas tabelas, bem como a toda a sua estrutura. Os registos armazenados consistem nas acções de transacção, antes de serem confirmadas (antes da operação de *commit*). Numa tarefa de recuperação da BD, estes dados são utilizados para reverter todas as mudanças aplicadas pelo *redo log* que não tenham sido confirmadas. Além disso garantem consistência, pois mantêm uma cópia dos dados antes da sua alteração para apresentar a utilizadores que estejam a lê-los enquanto outro está a actualizá-los. Este processo é gerido automaticamente pela BD, sendo definido pelo administrador o tempo de retenção destes dados.

Toda a informação referente ao **undo** é guardada num *tablespace* destinado ao efeito, o que elimina a necessidade de alocar espaço em diversos *rollback segments* com tamanhos diferentes. É importante referir que esta informação não é permanente, sendo escrita numa forma cíclica, com a substituição a ser dependente do período máximo de retenção definido pelo administrador. O espaço ocupado pelo *tablespace* é automaticamente definido pelo sistema de gestão, sendo normalmente um pouco maior do que a *query* mais longa a ser executada.

2.2.6 A gestão do *redo log*

O **redo log** é a estrutura mais crítica no que diz respeito a operações de recuperação. Consiste em dois ou mais ficheiros que armazenam todas as alterações feitas à BD à medida que ocorrem. Este número mínimo de dois ficheiros prende-se com a necessidade de haver sempre um ficheiro disponível para escrita, enquanto outro é arquivado. Toda e cada instância da BD tem um *redo log* associado para a proteger em caso de falhas. Estes ficheiros contêm registos que são grupos de vectores de mudanças, cada vector com uma descrição da mudança feita a um único bloco da BD.

Os registos que constam dos ficheiros são armazenados inicialmente num *buffer* – o *Log Buffer* –, numa actividade circular, que se situa na SGA, sendo escritos para os suportes físicos pelo Log Writer (LGWR). O LGWR é um processo que actua em segundo plano e se encarrega de escrever os registos de *redo* nos ficheiros, sempre que ocorre uma operação de *commit* a uma transacção. Só quando os registos estão efectivamente escritos com segurança é que o processo iniciado pelo utilizador é notificado da conclusão da sua transacção. O LGWR escreve para os ficheiros numa actividade circular, ou seja, quando um ficheiro está cheio, ele escreve para o próximo. Quando o último ficheiro disponível enche, o LGWR volta a escrever no primeiro, substituindo a informação aí presente e reiniciando o ciclo.

Os *redo logs* arquivados podem ser utilizados para recuperação de uma BD, actualização de uma BD em *standby* ou recolha de informação de histórico com o recurso a ferramentas exteriores.

O dimensionamento destes ficheiros deve ter em consideração o tamanho total do suporte onde serão armazenados, para que não aconteçam situações de um ficheiro ocupar 51% do espaço disponível, ficando os outros 49% inutilizados. O tamanho destes deverá ter em conta o *redo* gerado pelo sistema, considerando-se uma média de 1 operação de *rollback* por cada 4 operações à BD.

O número apropriado de ficheiros depende de caso para caso. Se as mensagens indicarem que o LGWR tem que esperar diversas vezes, será melhor adicionar grupos de ficheiros.

2.2.6.1 Dimensionamento do *Log Buffer*

As aplicações que transaccionam elevados volumes de dados necessitam de alterar o tamanho definido por defeito do *Log Buffer*. O tamanho normalmente ocupado por este *buffer* é pequeno, comparado com o total da SGA, e se for bem dimensionado pode aumentar a performance em sistemas que realizam diversas operações de inserção, actualização ou remoção de dados.

Uma estimativa razoável para o tamanho deste *buffer* pode ser dada pela seguinte fórmula:

$$\text{MAX (0.5M, (128K * número de processadores))}$$

No entanto, a maioria dos sistemas não retira qualquer benefício de um *log buffer* superior a 1M [ORAC08a].

2.3 A gestão da memória

A BD usa a memória para armazenar diversa informação, tal como código de programas, detalhes de sessões, dados referentes à execução de programas, informação a ser partilhada entre diferentes e dados armazenados em *cache*, para um acesso mais rápido.

As duas estruturas básicas de memória são a *System Global Area* (SGA), comum a todos os processos, e a *Program Global Area* (PGA) que é privada e associada a cada processo, havendo uma PGA para cada um.

2.3.1 System Global Area

A SGA é um grupo de estruturas de memória partilhadas que contém dados e informação de controlo referentes a uma BD. Múltiplos utilizadores ligados simultaneamente partilham esta área entre si, pelo que algumas vezes ela é referida como *Shared Global Area*. O espaço necessário é imediatamente alocado quando a BD é iniciada, sendo depois reclamado pelo sistema operativo, no momento em que a BD é desligada. Esta área de memória é de leitura e escrita, sendo acessível a todos os utilizadores e processos durante as sessões estabelecidas.

A SGA agrupa um número de *pools* de memória que são usadas para satisfazer requisitos particulares de alocação de memória. Estas *pools* são a *shared pool*, que é usada para alocar memória destinada à execução de código SQL e PL/SQL, para agilizar o acesso a bibliotecas e a dados do dicionário de dados, a *Java pool* que, como o nome indica, se destina à alocação de memória para objectos Java e execução de código, a *streams pool*, usada para alocação de memória para o módulo Oracle Streams, e a *large pool* que é opcional e é utilizada por um determinado tipo de processos que requerem a alocação de grandes quantidades de memória. Além das *pools*, a SGA contém ainda um *buffer* para *cache* de dados da BD e outro para o *redo log*. Os dados referentes a estas categorias passam por estas estruturas de memória antes de serem escritos em suportes físicos, para reduzir o número de operações de E/S e assim melhorar a performance global. Há ainda a referir a *cache* do dicionário de dados e outra informação variada que também fazem parte das *pools* da SGA.

Todos os componentes da SGA reservam e libertam memória em unidades de **granules**. O tamanho destes *granules* é determinado pelo tamanho total da SGA, sendo normalmente um valor de 4MB ou 16MB, caso o tamanho total seja menor ou maior do que 1GB, respectivamente [ORAC05]. Algumas dependências de plataforma têm que ser consideradas. Deve ter-se em atenção o facto de a SGA residir em memória real, para um bom desempenho do sistema. Caso resida em memória virtual, as diversas operações de E/S reduzem drasticamente a performance global.

2.3.2 Program Global Area

Uma PGA é uma região de memória que contém dados e informação de controlo de um processo do servidor. É uma porção de memória não partilhada criada pela BD quando um processo do servidor é iniciado, sendo o acesso a essa memória exclusivo a esse processo, o que resulta na existência de uma PGA por cada processo do servidor iniciado. Os processos de segundo plano também reservam as suas próprias PGA.

O conteúdo da PGA varia consoante a opção de partilha de servidor esteja a ser usada ou não, mas na generalidade é composta pela **private SQL area** e pela **session memory**. A *private SQL area* contém informação atribuída a variáveis ou cursores, e estruturas de memória associadas à execução. Cada chamada de SQL tem a sua própria área de memória privada. A área de um *cursor* divide-se em duas partes: a área persistente, que contém a informação atribuída e só é desalocada quando este é fechado, e a área de execução, que é libertada quando

o *cursor* acaba de realizar a sua tarefa. A *private SQL area* pode residir em parte na SGA, caso a sessão seja ligada através de um servidor partilhado. Noutros casos, reside totalmente na PGA.

A *session memory* é a memória reservada para guardar as variáveis de sessão, como as informações de autenticação e outras relacionadas com a sessão. No caso de ser um servidor partilhado, esta área é também partilhada.

Além destas, existem ainda **SQL work areas**, destinadas à alocação de memória para operadores que necessitam de porções maiores de memória, como operadores de ordenação, *hash-join* e criação e combinação de *bitmaps*. Estes operadores executam grande parte do seu trabalho em memória, antes de apresentarem o resultado final, pelo que o tamanho destas áreas deve ser considerado e ajustado às necessidades da BD, sob pena de se verificar um aumento do tempo de resposta, devido a uma área de trabalho subdimensionada que obriga algumas operações a serem executadas recorrendo a espaço em disco temporário, cujos acessos são significativamente mais lentos.

Impõe-se ainda a referência às **software code areas** que armazenam código que está a ser ou que poderá ser executado. Estas áreas são normalmente estáticas em tamanho, que depende do sistema operativo, sendo apenas alteradas quando existe instalação ou actualização de *software*. São áreas apenas de leitura e são partilhadas sempre que possível, numa tentativa de aumentar a performance e reduzir os requisitos de espaço de memória.

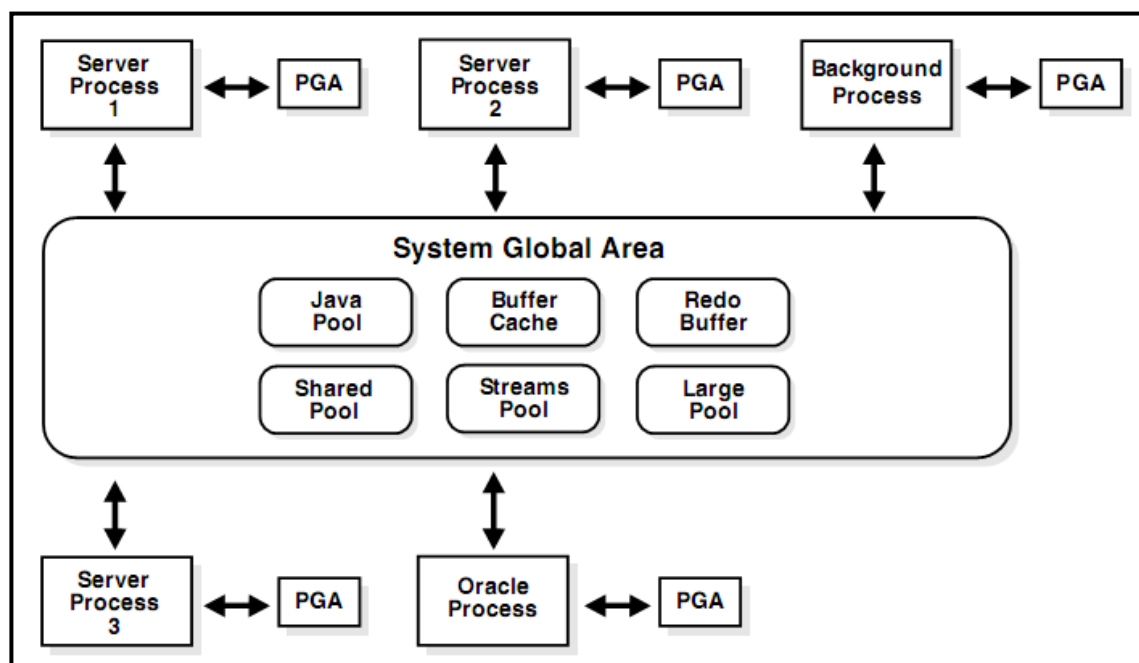


Figura 8 – Diagrama representativo das áreas de memória

2.3.3 A gestão automática de memória

A gestão dos espaços de memória pode ser deixada ao cargo da BD Oracle, escolhendo-se um de dois cenários. Com a gestão totalmente automática, o administrador apenas define um parâmetro de tamanho alvo e outro de tamanho máximo, ficando a BD responsável por gerir a memória ocupada pela SGA e pelas PGA, distribuindo o espaço entre estas duas estruturas conforme necessário, nunca ultrapassando o tamanho máximo definido. Neste cenário também os componentes individuais das duas estruturas são dinamicamente geridos. No outro cenário de gestão automática, o administrador define individualmente os tamanhos alvo e máximo para a SGA e para a PGA, e a BD trata de gerir os espaços alocados pelos componentes internos destas estruturas, dentro dos parâmetros definidos.

Apesar desta funcionalidade, a Oracle permite a gestão manual de todos os componentes de memória, mas como as necessidades variam significativamente consoante os instantes de tempo, esta gestão é pouco recomendada, estando inclusive a gestão automática activada por defeito.

2.4 O dimensionamento e planeamento de capacidade

O planeamento de capacidade é uma actividade que consiste na realização de uma estimativa do volume de dados de uma dada base de dados, para determinar o espaço físico necessário para a sua implementação e armazenamento. Além disso, também a carga de processador, os requisitos de memória e de rede são determinados, recorrendo a diferentes métricas.

O espaço de armazenamento pode ser estimado recorrendo-se a algumas fórmulas e cálculos. Estes cálculos consistem na recolha de tamanhos médios dos campos das tabelas, que poderão mesmo ser fornecidos nos manuais de administração da base de dados em questão. No entanto, dado que estes dados nem sempre estão disponíveis, torna-se necessário recorrer a outros métodos, sendo que alguns partem de uma base de dados vazia, enquanto outros requerem uma base de dados implementada, e estimam o seu potencial crescimento. Estes métodos vão ser explicados em detalhe nas próximas secções.

2.4.1 Amostras de dados e extrapolação

Esta metodologia envolve a introdução de dados no sistema, preferencialmente dados do mesmo domínio dos que serão armazenados posteriormente em ambiente de produção. Todas as tabelas devem ser populadas com cerca de 1000 registos, procedendo-se à recolha dos espaços médios ocupados por cada registo de cada tabela, tendo em atenção que os registos de tamanho variável raramente ocupam todo o espaço definido [WIPR06]. Outra forma de fazer esta recolha será o cálculo do número de registos armazenados em cada bloco, registando-se também o tamanho padrão dos blocos da base de dados. Depois de recolhida esta informação, as seguintes tarefas devem ser executadas:

1. Prever o número total de registos a ser introduzido em cada tabela, com a devida tolerância para a incrementação posterior;
2. Cruzando a informação recolhida sobre os tamanhos dos registos ou sobre o número de linhas por bloco com a previsão do número total de linhas de cada tabela, é possível então determinar o espaço total a ser ocupado por cada tabela e, desta forma, pela base de dados na sua totalidade;
3. Para os índices, pode considerar-se que o espaço por eles requerido é aproximadamente o mesmo ocupado pelas tabelas a que correspondem. Deve deixar-se um espaço de tolerância, de preferência igual ao tamanho do maior índice, para permitir a realização de tarefas de reorganização de índices;
4. O espaço temporário, normalmente ocupado por operações de ordenação, deve estimar-se como sendo no mínimo igual ou superior ao tamanho da maior tabela da base de dados. Este é um passo sensível, uma vez que algumas destas tarefas podem necessitar de mais espaço, pelo que devem ser conhecidas algumas características do sistema que funcionará sobre a base de dados, no sentido de avaliar os seus requisitos;
5. Para guardar os metadados, cerca de 20MB serão suficientes na esmagadora maioria dos casos [WIPR06];
6. Finalmente, uma percentagem do total calculado deve ser adicionada, para dotar o planeamento de alguma flexibilidade, uma vez que existe a possibilidade da

ocorrência de acontecimentos imprevistos. É de realçar que esta metodologia não compreende os requisitos do sistema que actua na base de dados, tais como tabelas e índices de sistema, *triggers*, blocos de PL/SQL ou arquivos binários de *software*.

Após este processo, há ainda a considerar o espaço para uso interno da base de dados. O dicionário de dados depende principalmente do número de objectos a serem criados e do uso dado a ferramentas do sistema.

O tamanho ocupado pelos segmentos de *rollback* é influenciado pelo número médio de transacções, e pelo volume de dados que movimentam. Quanto maiores forem estes números, maior será o número de *extents* por cada *rollback segment*.

Os ficheiros de *redo log* estão muitas vezes limitados no seu tamanho máximo pelo sistema de ficheiros do sistema operativo. No entanto, estes tipicamente encontram-se entre 50MB e 100MB, devendo considerar-se uma mudança de ficheiro em cada 30 minutos [WIPR06].

2.4.2 Tamanhos dos *datafiles*

Esta técnica de planeamento baseia-se no tamanho ocupado pelos ficheiros de dados de todos os *tablespaces*, conseguindo assim determinar o espaço total da base de dados. Recorrendo a um pequeno *script*, estes ficheiros poderão ser monitorizados durante um período finito, registando-se todas as alterações ocorridas, bem como a percentagem de espaço livre total nos suportes de armazenamento. Findo este processo, deverá realizar-se uma análise aos detalhes recolhidos pela monitorização, e determinar um factor de crescimento do volume de dados da BD, para assim efectuar mudanças a nível dos sistemas de armazenamento físico.

Note-se que esta técnica apenas planeia o crescimento de uma BD, e tem como requisito primário a existência prévia de uma implementação, para que a operação de monitorização seja possível.

2.4.3 Conteúdos dos *datafiles*

Uma alternativa ao método apresentado anteriormente consiste na recolha dos tamanhos ocupados pelo conteúdo dos *datafiles* da BD. Este processo é realizado consultando as vistas DBA_DATA_FILES e DBA_FREE_SPACE. Numa análise aos dados retornados, deve dar-se atenção ao número de *extents* associados a cada *datafile*, sendo que é com este número que se determina o tamanho da BD. Posteriormente, uma monitorização idêntica à relatada na sub-secção anterior pode ser feita, sendo que neste caso deve ter-se em atenção os *extents* que são adicionados a cada *datafile*, sem descurar o parâmetro PCTINCREASE, que determina uma percentagem a adicionar ao tamanho de um novo *extent*. Desta forma também será possível inferir um factor de variação do tamanho da BD, recorrendo a estes valores mais precisos que os anteriores, embora se possam desviar um pouco do pretendido, devido ao valor de PCTINCREASE, que resultará numa variação dos tamanhos dos *extents*.

Tal como o método anterior, também o planeamento efectuado se cinge ao crescimento potencial da BD, baseando-se na recolha de dados sobre uma BD implementada e na monitorização da actividade desta.

2.4.4 Uso do *package DBMS_SPACE*

O DBMS_SPACE é um conjunto de procedimentos escritos em PL/SQL, que permite o acesso a estatísticas que não estão disponíveis através da consulta das vistas existentes, como as referidas na sub-secção anterior. Um *script* PL/SQL que recorra ao procedimento DBMS_SPACE.UNUSED_SPACE produzirá resultados que possibilitam determinar o tamanho total da base de dados em termos de blocos usados e livres [GAVIN02].

O principal problema deste método é o parâmetro de armazenamento dos blocos, PCTUSED, que determina a percentagem de espaço num bloco permitida para novas inserções. Quando esta percentagem é alcançada, o restante espaço do bloco está reservado para operações de actualização. Assim, se um bloco sofrer remoção de dados, mas continuar acima da percentagem definida para este parâmetro, o bloco será marcado como estando em uso, pelo que os resultados finais poderão estar um pouco desviados do tamanho real da BD.

Tal como nos métodos anteriores, estes dados serão utilizados numa monitorização constante da BD e numa avaliação do seu crescimento e necessidades de armazenamento.

2.4.5 Recolha de estatísticas

O planeamento de capacidade pode ser feito com recurso a estatísticas actuais da base de dados, geradas por processos internos da própria BD, criados especificamente para esta finalidade [GAVIN02]. O comando **ANALYZE** é utilizado para calcular estatísticas de tabelas e índices, podendo, se preferido, produzir estimativas, caso a *performance* do sistema seja crítica e não possa ser ameaçada. Outra forma de conseguir estatísticas da BD é o uso das funções do *package* **DBMS_STATS**, pensado para este propósito, é mais rápido do que o procedimento anterior e produz estatísticas de uma ou várias tabelas, ou até mesmo de toda a BD. O uso deste *package* é recomendado pela Oracle, em detrimento do comando **ANALYZE**, embora este último seja a única forma de se saber estatísticas a nível de blocos livres e ocupados ou espaço médio. A versatilidade do **DBMS_STATS** permite o armazenamento das estatísticas em tabelas que podem ser manipuladas de qualquer maneira, ou exportadas para outra base de dados, ao passo que o **ANALYZE** popula algumas vistas do dicionário de dados.

Após a geração de estatísticas, pode consultar-se a vista **USER_TABLES**, ou a **DBA_TABLES**, no sentido de recolher valores de blocos ocupados, número de registos ou tamanhos médios de registos. Sabendo o tamanho padrão dos blocos da BD, é fácil calcular o espaço físico ocupado por cada tabela. Aplicando o mesmo raciocínio aos índices, que são mais numerosos que as tabelas, é também possível calcular os recursos por eles alocados. Através de consultas às vistas **USER_INDEXES** ou **DBA_INDEXES**, o utilizador pode saber o número de blocos-folha do índice e os níveis de blocos intermédios até à raiz, podendo então avaliar o espaço ocupado pelos índices, sabendo de antemão o tamanho padrão dos blocos da BD.

Depois de calculado o espaço total da BD, este pode ser constantemente analisado, a fim de avaliar potenciais variações no tamanho da BD e planear para a capacidade do sistema.

2.4.6 Tamanho exacto dos dados das colunas

Esta técnica envolve a recolha dos comprimentos dos registos de todas as colunas em todas as tabelas e índices da BD. Este método será um dos mais precisos, embora se apresente como uma séria ameaça à *performance* e um pouco moroso. Desta forma, as melhores situações para a aplicação deste método caracterizam-se pela existência de uma BD relativamente pequena, em que a recolha das informações necessárias seria feita com mais rapidez. Também numa situação de migração de BD de um sistema não-Oracle justifica este método, para o planeamento ser mais rigoroso [GAVIN02].

Inicialmente, o utilizador deverá consultar as vistas **USER_TAB_COLUMNS** ou **DBA_TAB_COLUMNS**, que contém informação sobre o tamanho máximo dos campos de cada coluna. Para manter a simplicidade do método, apenas se devem considerar os tipos de dados simples, por terem um tamanho fixo; no entanto, destes, destaca-se o **VARCHAR2**, de tamanho variável, pelo que será o único a necessitar do uso do comando **LENGTH**, quer determinará o tamanho real ocupado por este campo, sendo por esta razão uma ameaça à *performance*, pois terá que analisar este campo em todos os registos.

Com estes tamanhos recolhidos, o cálculo do tamanho de cada tabela resume-se à multiplicação do número total de registos pelo tamanho total de cada um deles. Somando-se estes valores, consegue-se o tamanho total da BD. Quanto aos índices, o mesmo processo é aplicado, multiplicando o número de registos pelo tamanho do campo guardado pelo índice. No entanto, este método poderá conduzir a resultados desviados da realidade, uma vez que existem valores nulos que não são guardados, o que resulta num número menor de registos em relação à tabela correspondente. No seguimento dos métodos anteriores, depois de determinado o tamanho total da BD e feitas algumas análises à evolução deste valor, pode fazer-se um planeamento de capacidade para o sistema para suportar o potencial crescimento do volume de dados a armazenar.

2.5 Tecnologias de redundância de dados

Em ambientes empresariais, a informação tem um carácter mais crítico, devendo estar sempre acessível e segura. A perda de dados pode trazer consequências muito graves ao negócio, seja a nível financeiro, ou mesmo a nível de reputação e imagem passada para o exterior. Desta forma, torna-se imperativo a implementação de soluções que garantam a segurança da informação e que resistam a qualquer tipo de falha que possa afectar o sistema, no sentido de salvaguardar todo o conteúdo de uma BD.

Dada esta necessidade de segurar a informação, ganha importância a tecnologia RAID² para implementação em servidores de bases de dados. Esta tecnologia, criada na Universidade de Berkeley, Califórnia [WIKI09b], consiste na combinação de vários discos rígidos, formando uma única unidade lógica, sendo os dados repartidos entre todos. Consequentemente, quando um dos discos falhar, os outros continuam a assegurar o acesso aos dados, até que a reparação ou substituição do disco faltoso seja efectuada, não havendo perda de dados, nem períodos de inoperância do sistema. Esta distribuição de dados pelos discos traz também vantagens ao nível da velocidade de leitura e de escrita, face a uma implementação clássica, de discos singulares e independentes. Existem diversas implementações desta tecnologia, que diferem na forma como distribuem os dados pelos discos. Estas implementações são denominadas de “níveis”, apresentando-se em seguida os mais comuns actualmente, por cobrirem todos os requisitos esperados.

2.5.1 RAID 0

Este nível não é propriamente uma implementação RAID, na medida em que não garante a redundância dos dados, não garantindo, por isso, a segurança dos mesmos. No entanto, partilha as características da tecnologia, por utilizar diversos discos combinados.

O seu funcionamento consiste na divisão dos blocos de dados pelos discos existentes, obrigatoriamente em número igual ou superior a 2. Com esta divisão, a leitura e escrita dos dados é mais rápida, uma vez que os diversos discos são utilizados em paralelo. A sua implementação é fácil, sendo muito comum em computadores residenciais. A sua grande desvantagem reside então na ausência de redundância de dados, que não proporciona qualquer segurança contra falhas, havendo perda de todos os dados nestes casos.

2.5.2 RAID 1

Esta técnica consiste na replicação exacta do conteúdo de um disco para outro, havendo assim redundância de dados, podendo um disco falhar e o sistema continuar em actividade, pois todos os dados estão disponíveis no outro.

² RAID é um acrónimo inglês que significa *Redundant Array of Independent Disks*.

Aquando da escrita dos dados, o controlador cria uma cópia exacta dos pacotes e escreve-os na outra unidade, garantindo uma redundância total dos dados. Devido a estas características, esta implementação requer 2 ou mais discos para funcionar. Apresenta como principal desvantagem o facto de a capacidade de um dos discos não ser aproveitada, uma vez que apenas serve como espelho de outro.

2.5.3 RAID 5

Nesta implementação, os dados são repartidos em blocos pelos discos existentes, existindo um bloco de paridade associado. Em caso de falha de algum dos discos – note-se que esta técnica apenas garante os dados em caso de falha de apenas um dos discos – este bloco de paridade é utilizado por um algoritmo do controlador, permitindo a reconstrução dos dados em falta. A velocidade de leitura dos dados é melhorada, uma vez que permite o acesso a múltiplos discos em simultâneo; já a velocidade de escrita é um pouco afectada, devido à necessidade de criação e escrita dos pacotes de paridade, que consomem algum tempo em cada transacção, traduzindo-se assim na desvantagem principal desta implementação. Em relação às outras mencionadas nesta secção, há que mencionar a implicação de um terceiro disco para escrita dos blocos de paridade, sendo então necessários 3 ou mais discos para esta técnica.

Esta apresenta-se como a implementação mais comum em servidores de bases de dados, servidores de ficheiros ou de aplicações, considerando-se o nível RAID mais versátil [DELL].

2.6 Resumo e conclusões

Neste capítulo foi feita uma revisão bibliográfica dos conceitos e tecnologias que fazem parte do projecto. É possível ficar a conhecer o conceito de uma base de dados, e perceber as diferentes estruturas e processos de gestão que constituem uma BD Oracle, que será alvo de estudo neste projecto. Na secção de armazenamento, uma chamada de atenção para o funcionamento das estruturas *B-tree*, que são o suporte lógico dos índices das bases de dados.

A pesquisa efectuada sobre técnicas de dimensionamento de bases de dados retornou as metodologias apresentadas, todas elas perseguindo o mesmo objectivo, usando diferentes meios para o atingir. Esta pesquisa, além de revelar o trabalho feito na área, permitiu chegar ao método utilizado na resolução do problema, através da combinação de algumas técnicas, método este que será apresentado em maior detalhe numa fase posterior deste documento. A fechar, incluiu-se uma apresentação das diferentes implementações da tecnologia RAID, que se conclui ser de extrema importância nos servidores de BD, uma vez que a segurança dos dados é vital e imperativa, sendo que a sua perda acarreta sérios riscos e consequências para o negócio.

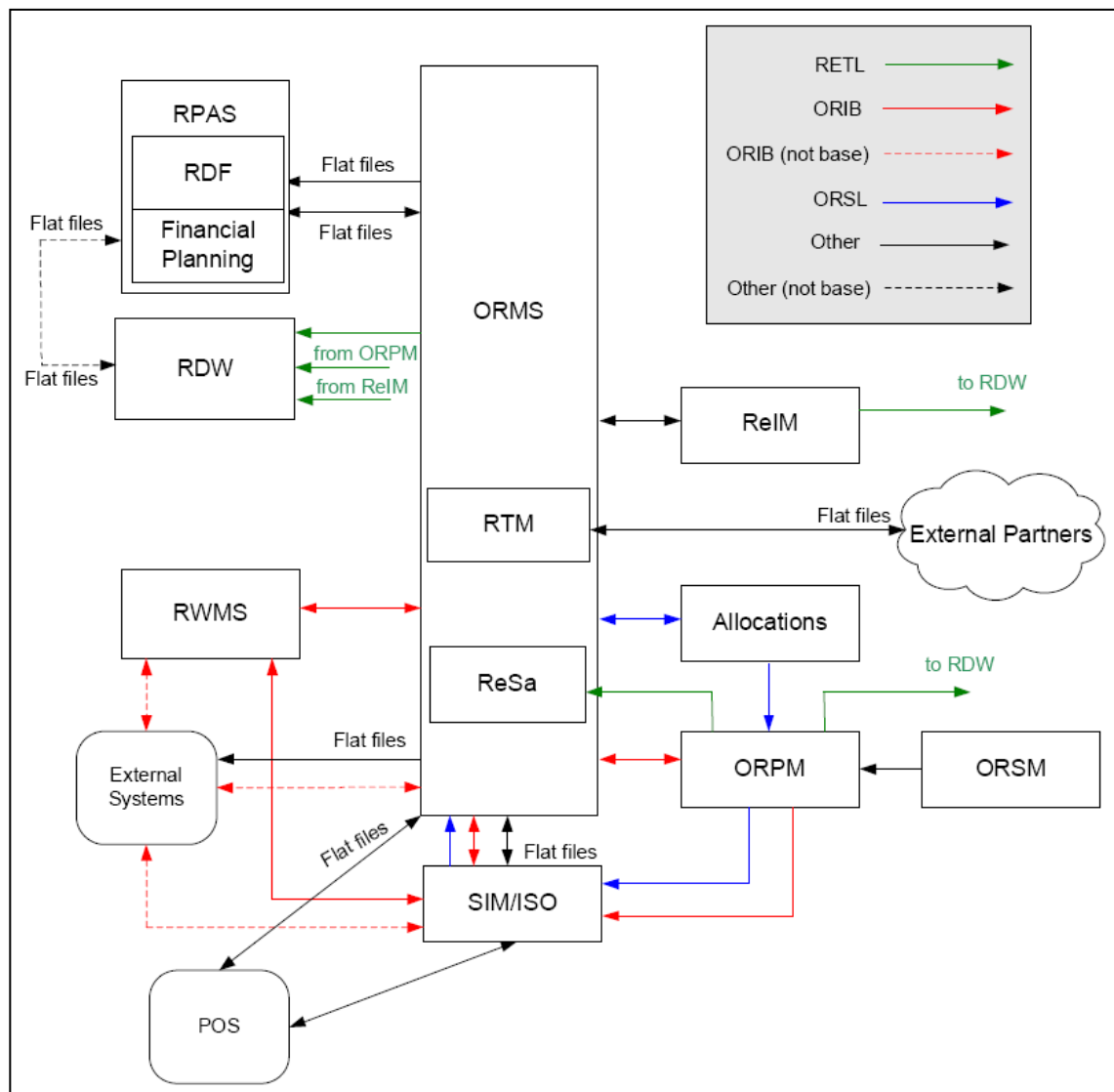
3 Caracterização do problema

Neste capítulo apresenta-se a suite Oracle Retail, para que os destinatários deste documento fiquem com uma ideia do que realmente é esta carteira de produtos, quais as suas potencialidades e mais-valias para o cliente, justificando-se a necessidade de fazer este projecto relacionado com a mesma. É ainda feita uma descrição do projecto, explicitando o seu âmbito e os seus detalhes, seguido da enumeração dos objectivos principais. Posteriormente, apresenta-se alguns dos aspectos essenciais à compreensão do projecto, tal como o sistema ORMS, o seu modelo de dados e os casos práticos analisados.

3.1 Oracle Retail

A carteira de produtos Oracle Retail consiste em diversos módulos de *software* para vendas a retalho, desenvolvidos especificamente para cada área funcional ligada ao negócio, que são responsáveis por uma transformação e reorganização dos processos de negócio dos clientes que se decidem pela sua implementação na sua organização. Este sistema resulta de uma estratégia da Oracle em adquirir as melhores aplicações desenvolvidas no sector, integrando-as e continuando a trabalhar sobre elas, investigando novas funcionalidades, sempre sem descurar as evoluções dos modelos de negócio nas vendas a retalho. Uma implementação deste tipo permite aos retalhistas obter uma melhor integração de todas as unidades funcionais da sua empresa, ao mesmo tempo que cria valor e vantagens competitivas, ao agilizar processos e reduzir custos, permitindo responder com mais eficiência e eficácia às necessidades dos clientes que todos os dias visitam as suas lojas, sejam elas físicas ou virtuais. Segundo a Oracle, os 20 maiores retalhistas globais têm produtos Oracle Retail implementados [ORAC09].

Os diversos módulos que constituem a suite de produtos podem ser consultados no diagrama que se segue, acompanhados de indicações sobre a forma como comunicam entre si.



Os módulos que constam do diagrama acima estão relacionados com as mais diversas actividades inerentes ao tipo de negócio em causa. A figura que se segue enumera todas essas áreas funcionais, sendo depois apresentada uma relação entre os módulos e as unidades funcionais dos retalhistas.

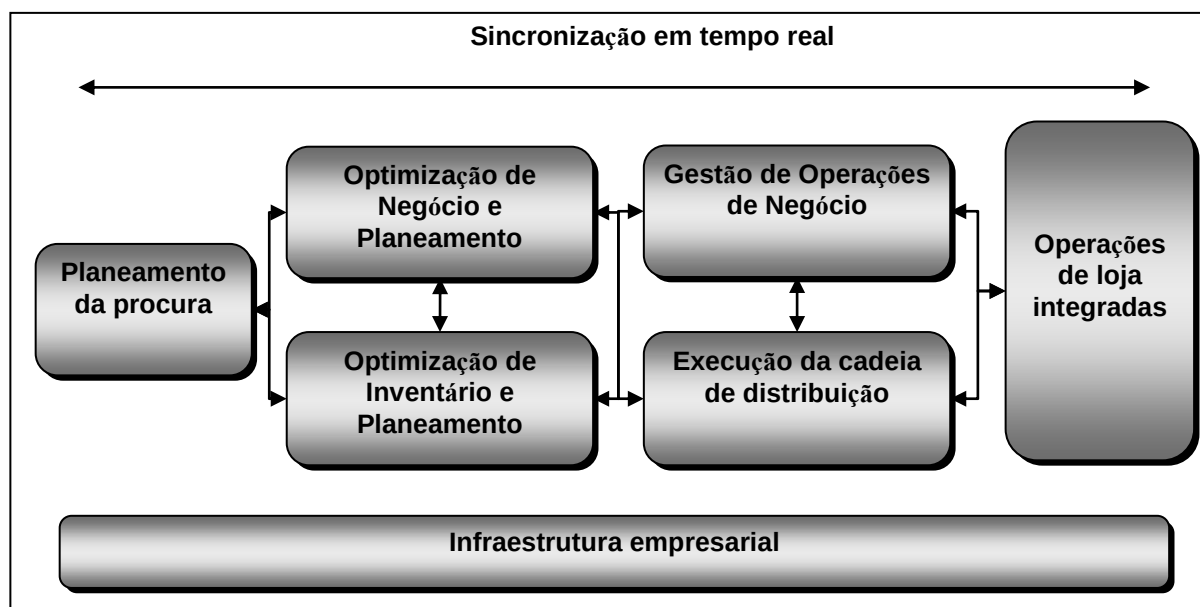


Figura 10 – Diagrama representativo das diversas áreas funcionais do negócio

- **Planeamento da Procura** – O planeamento da procura é conseguido com o uso de uma ferramenta poderosa de análise dos processos em tempo real, reunindo informação sobre vendas e *stocks*, fazendo previsões sobre os níveis futuros da procura dos produtos. O Oracle Retail Demand Forecast (RDF) faz parte do modulo Oracle Retail Predictive Application Server (RPAS), e é o responsável por esta area, integrando-se com as outras soluções.
- **Optimização de negócio e planeamento** – Nesta área funcional, é necessário criar estimativas de vendas, planear encomendas e maximizar o lucro, ao mesmo tempo que se diminuem as perdas relativas a promoções forçadas ou falhas de *stock*. O Oracle Retail Predictive Application Server (RPAS) tem funções nesta área, com capacidades para gestão e planeamento de finanças e operações de negócio.
- **Gestão de operações de negócio** – Estas operações são asseguradas por diferentes módulos completamente integrados que se podem implementar de uma maneira independente. Estes conseguem coordenar os processos de negócio e manter a informação sincronizada e disponível para todas as unidades funcionais da organização. Oracle Retail Merchandising System (ORMS), Oracle Retail Sales Audit (OReSA), Oracle Retail Trade Management (ORTM), Oracle Retail Invoice Matching (OReIM) e Oracle Retail Price Management (ORPM) são os módulos que se incluem neste domínio.
- **Optimização de inventário e planeamento** – Aqui considera-se a procura, a oferta e algumas restrições características da cadeia de distribuição. Os módulos responsáveis por esta área asseguram a gestão dos inventários e a distribuição dos produtos pelas lojas, minimizando riscos e custos, estando focada na qualidade do serviço e no lucro. Inclui-se o módulo Allocations, responsável pela distribuição dos produtos e pela organização de encomendas.
- **Execução da cadeia de distribuição** – As aplicações nesta área planeiam e optimizam a reposição, através da gestão eficaz de todos os aspectos relacionados com os armazéns, onde os produtos estão em quantidades mais elevadas, esperando as necessidades das lojas. Esta gestão justifica-se pela elevada gama de produtos encontrada neste negócio, os níveis de vendas que escoam os produtos rapidamente,

não esquecendo a complexa rede de distribuição que engloba centenas ou milhares de lojas e armazéns, sendo vital a correcta ligação e interacção entre todos estes locais. O Oracle Retail Warehouse Management System (ORWMS) é o responsável por esta gestão de armazéns e distribuição.

- **Operações de loja integradas** – Estas operações são extremamente importantes, uma vez que é a este nível que se dão as vendas, a fonte de lucro para a organização. A sua integração e gestão permite oferecer um melhor serviço aos clientes, permitindo ao mesmo tempo conduzir o negócio através de diferentes canais, como a Internet ou os telefones. Aqui têm funções o módulo de Point-of-Sale (POS) e o Oracle Retail Store Inventory Management (ORSIM).

3.2 Descrição e objectivos principais

Este projecto desempenha uma função importante no planeamento de uma implementação de um sistema Oracle Retail (OR), mais propriamente o módulo Oracle Retail Merchandising System (ORMS). Este *software*, que será apresentado posteriormente, é o cerne da suite OR, concentrando em si as funcionalidades mais principais, ao mesmo tempo que armazena informação crucial para o desempenho da organização.

O dimensionamento da base de dados que suporta o sistema, e que guarda toda a informação é um passo importante no planeamento, pois dita o equipamento que deve ser adquirido para a implementação do ORMS. Este planeamento justifica a sua importância devido às necessidades normalmente apresentadas por uma implementação deste tipo serem acima da média, levando a investimentos avultados por parte dos clientes, pelo que há uma preocupação dos mesmos no sentido de fazer diminuir o montante a investir em Tecnologias de Informação ao verdadeiramente requerido, sem entrar em excessos inúteis. Este dimensionamento consiste numa primeira abordagem ao cliente, no sentido de perceber as características do seu ambiente de negócio e consequentemente, as suas necessidades. Para isso, é-lhe pedido que preencha um questionário, onde são feitas perguntas relativas à sua organização, desde o número de lojas e artigos a comercializar, passando por estimativas de vendas ou de artigos devolvidos. Este questionário foi desenvolvido pela Oracle, e contém partes que se mostraram irrelevantes para o exercício de dimensionamento, ou que os clientes não se sentiram confortáveis a responder, por se desviarem um pouco do negócio e se aproximarem das funcionalidades do produto que, numa fase tão prematura, ainda não conhecem em detalhe.

Outra das questões relevantes deste projecto é a necessidade da Wipro adoptar uma postura crítica face a um exercício de dimensionamento da Oracle, pois conhece mais de perto as necessidades de cada cliente, ao contrário da Oracle, que se baseia exclusivamente no inquérito. Nesta linha de pensamento, a Wipro criou este projecto, para conseguir perceber a mecânica dum exercício de dimensionamento, poder fazer a sua própria proposta e comparar com os resultados obtidos pela Oracle, decidindo depois qual a melhor solução para o problema.

Assim, os objectivos primários deste projecto são os seguintes:

- Estudar e perceber o ORMS, desde a sua composição até ao seu funcionamento;
- Analisar o questionário da Oracle, classificando as perguntas e percebendo a sua relação com o sistema;
- Estudar o modelo de dados do ORMS;
- Estudar o *batch* nocturno e os processos que o compõem;
- Elaborar ferramenta de dimensionamento.

Para atingir estes objectivos, será necessário utilizar o sistema ORMS, analisar documentação oficial produzida pela Oracle, estudar técnicas de dimensionamento de bases de dados e analisar casos práticos de ambientes de produção.

3.3 Oracle Retail Merchandising System

O Oracle Retail Merchandising System (ORMS) é o sistema que armazena e controla todos os dados numa organização retalhista e assegura a integridade da mesma em todos os sistemas integrados. Este inclui como principais funções o controlo de inventários, compras, contratos e negócios com fornecedores, gestão de custos, reposição de artigos e gestão financeira. Com o ORMS, o acesso à informação crucial para as actividades diárias da empresa é facilitado, ao passo que as decisões chave são facilitadas, ajudando a atingir os objectivos de vendas e lucros previstos.

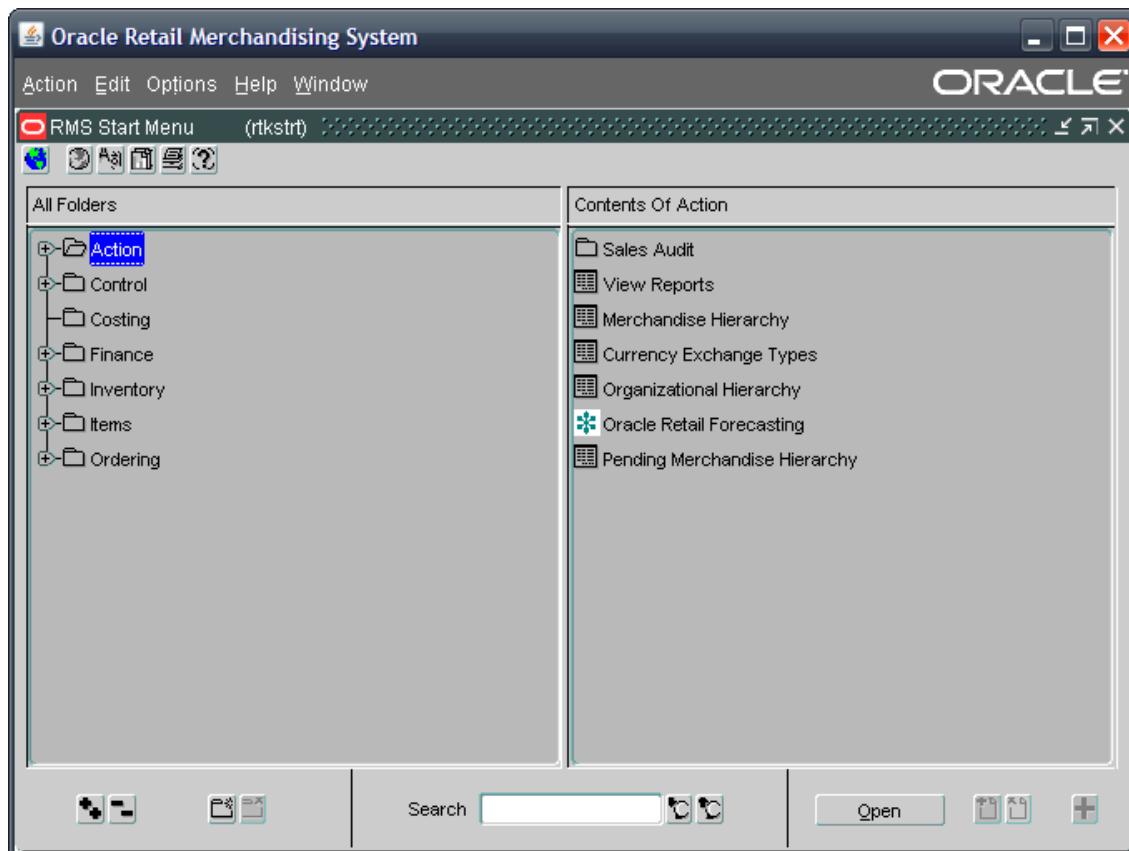


Figura 11 – Ecrã inicial do sistema

Com a utilização do ORMS, os retalhistas encontram todos os sistemas de negócio unidos num só, ao longo de toda a cadeia de retalho, melhorando o serviço prestado ao cliente final. Por ser desenhado para escalabilidade e por ser *web-based*³ permite suportar os elevados volumes de dados e transacções gerados por um negócio desta área em particular, facilitando as actividades dos retalhistas que podem assim concentrar-se em outras operações de negócio.

3.3.1 Dados fundamentais

A *Foundation Data* é o conjunto de dados que define os conceitos centrais do negócio, tais como:

- **Produtos** – o sistema permite a criação de artigos simples ou complexos. Os artigos complexos são criados pela definição de algumas variações do mesmo, conhecidas como *Item Levels*. Um *Diff* dá a possibilidade de distinguir os itens por características, como cores, tamanhos ou até mesmo aromas, facilitando a

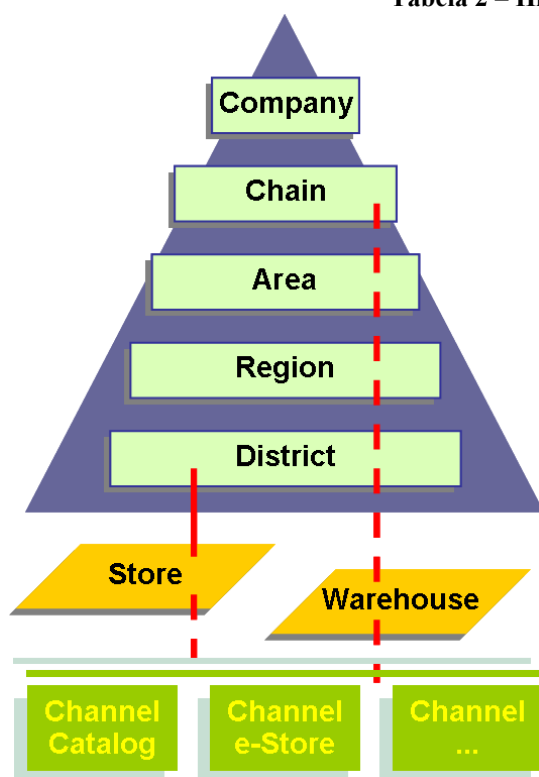
³ Sistema distribuído, com interface acedida através de um *browser* de Internet.

classificação dos artigos. Existe ainda outra forma de organizar os artigos, que passa pela criação de atributos definidos pelo utilizador (UDA), que promovem a classificação de itens mais específicos.

- **Lojas e armazéns (entrepósitos)** – as lojas são os pontos de venda físicos, sendo a este nível que todas as transacções de negócio são processadas, o que inclui vendas, recebimentos, ajustes e contagens de *stock*. Os armazéns são os pontos físicos de armazenamento dos produtos, servindo também como ponto de distribuição para todas as lojas. Tem todas as capacidades transaccionais de uma loja, excepto as vendas e a definição de preços. Estes podem ser divididos em vários armazéns virtuais, quando o ambiente de distribuição é multi facetado e inclui diferentes canais.
- **Fornecedores** – para cada fornecedor, o ORMS permite guardar múltiplos endereços, como de facturação, de retorno ou de correspondência. Oferece ainda a gestão integrada de termos de pagamento, unidades monetárias, ciclos de reposição, quantidades mínimas de encomendas ou atributos das mesmas, entre outros dados. É feito o registo de condições de retorno de produtos, perfis de despesas, atributos de importações ou definição de agendas de entrega, onde se especificam os melhores dias para a entrega dos produtos em determinadas localizações.

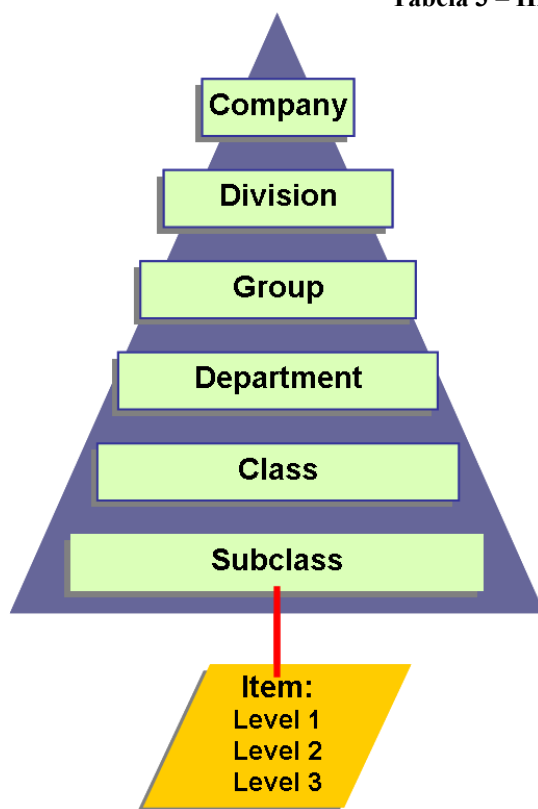
Além destes, a *Foundation Data* também é composta pela hierarquia organizacional, e pela hierarquia mercadológica, apresentadas nas figuras seguintes.

Tabela 2 – Hierarquia organizacional



- **Company** – é a unidade mais alta na hierarquia organizacional. Apenas uma pode ser definida no sistema.
- **Chain** – usada para agrupar diferentes formatos de loja e locais geográficos dentro da organização.
- **Area** – define um grupo geográfico no sistema. Uma *area* corresponde a uma única *chain*.
- **Region** – é também usada para agrupar localizações geográficas, sendo que uma *region* apenas pertence a uma *area*.
- **District** – representa uma localização geográfica pertencente a uma *region*, não podendo pertencer a mais nenhuma.
- **Store** – pontos de venda físicos, onde ocorrem as transacções de negócio.
- **Warehouse** – representa um armazém ou entreposto, não estando ligado a nenhum nível hierárquico.

Tabela 3 – Hierarquia mercadológica



- **Company** – o nível mais elevado na hierarquia, partilhado da hierarquia organizacional. Apenas uma pode ser definida.
- **Division** – representa o tipo de produtos.
- **Group** – este é um tipo de agrupamento dos produtos a um nível mais baixo, pertencendo a uma única *division*.
- **Department** – apresenta-se como o nível onde a gestão dos produtos é feita, produzindo-se relatórios, orçamentos, inventários e lucros.
- **Class** – grupos de produtos dentro de um departamento.
- **Subclass** – agrupamento de classes em sub-categorias.
- **Item** – este é o nível ao qual os artigos são mantidos.

3.3.2 Controlo de inventário

Esta funcionalidade permite controlar todas as transacções de produtos a nível interno. Estas transacções envolvem a movimentação de *stocks* entre diferentes localizações geográficas da organização. Em seguida, lista-se as diferentes operações controladas pelo sistema, na área de inventários:

- **Shipping** – esta funcionalidade controla todas as expedições de mercadoria, permitindo guardar e consultar dados de expedição de todas as ordens de compra e todas as transferências de *stock* feitas internamente.
- **Transfer** – aqui se armazenam os dados relativos a transferências. Uma transferência é uma movimentação de *stocks* entre duas localizações na empresa, tais como devoluções, circulação de produtos entre armazéns ou lojas, ou entre armazéns e lojas.
- **Inventory adjustments** – o sistema permite também controlar os ajustes de inventário, registando as alterações feitas e as razões pelas quais o ajuste foi realizado. Estes ajustes são feitos ao nível do *stock on hand*, que é o montante de produtos disponíveis para venda, e ao nível do inventário indisponível, que por diversas razões, está reservado e não pode ser vendido. Estes valores de *stock* são determinados pelas *stock counts*, que são as contagens de inventário, feitas pela organização e controladas pelo ORMS.

3.3.3 Compras

O ORMS possibilita ao utilizador a gestão de todas as compras efectuadas pela organização, mantendo o registo de orçamentos, origens e destinos dos produtos, pessoas responsáveis e custos associados às transacções.

- **Buyers** – estas pessoas são registadas no sistema como responsáveis pelos níveis de *stock* em cada loja, no sentido de cumprir com a procura dos clientes.
- **Open to buy** – este é o valor em dinheiro que a pessoa responsável pelas compras tem disponível para adquirir produtos para a loja, podendo ser ajustado a qualquer momento.
- **Purchase orders** – uma das funcionalidades mais importantes do sistema é a gestão das ordens de compra. O sistema permite a criação de ordens, a sua submissão e aprovação, quer automática ou manualmente. Após a aprovação da ordem de compra, é ainda possível determinar a distribuição dos produtos pelos locais pretendidos, segundo múltiplos critérios. Quando a mercadoria é recebida, o sistema compara as quantidades recebidas com as quantidades pedidas, avisando o utilizador no caso de haver diferenças, facilitando a emissão de notas de crédito para os fornecedores.

3.3.4 Contratos e negócios

Nesta área funcional, é oferecida a possibilidade de gerir os contratos e os negócios do retalhista com os seus fornecedores. Um contrato é um acordo legal entre as duas partes, para a venda de um certo tipo de produtos a um preço previamente negociado. A utilidade do registo dos contratos reside na capacidade do sistema em escolher o melhor contrato no momento de efectuar um pedido a um fornecedor, garantindo assim o melhor preço disponível.

Um negócio é também um acordo entre as duas partes, onde se fixam obrigações para as mesmas, estabelecendo-se algumas metas de quantidades ou de montantes de compras que, quando atingidas, resultam em benefícios, como preços diminuídos, ou pagamento de apenas uma parte da encomenda.

3.3.5 Controlo de custos

Em termos de controlo de custos, o ORMS permite definir informações sobre alteração de custos, submeter e aprovar alterações de custos, estimar custos totais de transporte de encomendas e despesas associadas a transferências. Esta funcionalidade possibilita a gestão dos custos associados a cada produto, influenciando os preços praticados pelo retalhista. Quando os custos se alteram, o utilizador deverá actualizar o sistema, para reflectir essa alteração nos dados armazenados.

3.3.6 Reposição

A reposição no ORMS consiste na constante monitorização dos níveis de *stock* existentes em cada localização, resultando na criação automática de ordens de compra ou transferências, para cumprir com as necessidades dos clientes. Os dados relativos aos fornecedores são tidos em conta nestas operações. Com esta funcionalidade, é possível efectuar também *Investment Buys*, que são compras de ocasião, onde o retalhista encontra os produtos que quer a um preço menor, proporcionando-lhe um retorno do investimento maior. O ORMS possui algoritmos de análise das oportunidades, de níveis de inventário e de custos da encomenda, decidindo sobre a quantidade óptima a encomendar para garantir benefício.

3.3.7 Gestão financeira

No ORMS, a gestão financeira é efectuada ao nível de calendários e orçamentos, performance financeira e impostos associados às vendas.

Os calendários podem ser normais ou de retalho⁴. Os orçamentos são mantidos a nível semestral e mensal. Pela introdução de percentagens e montantes monetários, é possível definir metas, usadas para monitorizar a performance do negócio. O *open to buy*, referido anteriormente, é normalmente gerido por um módulo externo, pelo que o ORMS mantém apenas o seu registo, actualiza-o e permite alterações manuais.

O *stock ledger* é uma parte do ORMS responsável por verificar a performance financeira do retalhista, incorporando todas as transacções financeiras, incluindo vendas, compras ou transferências.

Os impostos, nomeadamente o Imposto de Valor Acrescentado, são mantidos pelo sistema, por estarem muitas vezes associados aos preços dos produtos. É possível definir no ORMS códigos de taxas, e a percentagem associada. Há também a possibilidade de criar zonas de impostos, determinando as taxas a aplicar aos produtos das lojas que se incluem numa zona em particular. As taxas são aplicadas a nível do preço a praticar, bem como a nível de custos associados ao produto.

3.4 Modelo de dados do ORMS

Nesta secção, pretende dar-se ao leitor uma visão geral do modelo de dados do ORMS. Para isso, foi feita uma divisão em áreas funcionais, sendo apresentadas as tabelas mais importantes da base de dados, para cada uma destas áreas. Em anexo, pode consultar-se uma versão um pouco mais detalhada do modelo de dados, sendo que a consulta da sua totalidade só será possível mediante a leitura do documento oficial da Oracle. Desta forma, foram definidas 9 áreas funcionais [WIPR09], que constam das próximas sub-secções.

3.4.1 Dados fundamentais

Esta área funcional, identificada no sistema por *Foundation Data*, compreende toda a informação sobre a organização, utilizada para qualquer operação de negócio. O sistema promove o seu armazenamento numa forma centralizada, para que possa ser partilhada por todas as áreas que dela necessitam.

De todas as tabelas agrupadas por esta área, destacam-se as tabelas que armazenam dados dos diferentes níveis de hierarquia, como a CHAIN, AREA, REGION e DISTRICT, e as tabelas dos locais, a STORE e a WH. É também aqui que se trata a informação relativa à hierarquia mercadológica. Assim, deve evidenciar-se as tabelas DIVISION, GROUPS, DEPS, CLASS e SUBCLASS.

Ainda na área da *Foundation Data*, temos a informação referente aos fornecedores, as entidades que providenciam os produtos para venda. Estes dados são armazenados nas tabelas SUPS e SUP_ATTRIBUTES. A relação entre os fornecedores e os artigos é assegurada nos registos das tabelas ITEM_SUPPLIER e ITEM_SUPP_COUNTRY.

Para finalizar, existem outros dados importantes, como as taxas de imposto a aplicar, ou as unidades monetárias usadas no sistema, sendo esta informação guardada nas tabelas VAT_CODE_RATES e CURRENCIES.

3.4.2 Manutenção de itens

A manutenção dos artigos, ou itens, é também uma operação central do ORMS. Os artigos têm que ser definidos no sistema antes de o retalhista os poder comprar e vender. As tabelas

⁴ Este calendário designa-se por 4-5-4, sendo dividido em meses com 4 semanas, 5 semanas e 4 semanas, ciclicamente.

criadas para armazenar a informação relativa aos artigos são as que têm um maior potencial de crescimento, uma vez que a manutenção dos itens gera imensos registos, dada a variedade que um retalhista normalmente oferece aos seus clientes.

Das tabelas relacionadas com artigos, há a realçar a importância da ITEM_MASTER, que tem um registo por cada artigo definido no sistema, a ITEM_LOC e a ITEM_LOC_SOH, que têm informação relativa ao estado dos produtos nas diversas lojas, e a ITEM_LOC_HIST, que contém um histórico relativo às vendas dos artigos por cada local. As tabelas DEPT_SALES_HIST, CLASS_SALES_HIST e SUBCLASS_SALES_HIST são também importantes, pois contêm histórico de vendas agrupado por *department*, *class* e *subclass*, respectivamente. As operações de reclassificação são definidas e guardadas para consulta futura nas tabelas RECLASS_HEAD e RECLASS_ITEM.

Relativamente aos *differentiators*, guarda-se informação base desta funcionalidade nas tabelas DIFF_TYPE e DIFF_IDS. O cruzamento de dados entre artigos, *differentiators* e vendas está na tabela ITEM_DIFF_LOC_HIST.

Os *packs* são conjuntos de artigos vendidos num único pacote, que pode conter artigos de uma ou de variadas categorias. A informação relativa a este tipo de produto é guardada em tabelas diferentes das dos outros artigos, pois os seus atributos são um pouco diferentes. Assim, existe na base de dados as tabelas PACKITEM e PACKITEM_BREAKOUT, que ajudam no processamento e classificação da informação gerada por estes itens.

Os UDA, atributos definidos pelo utilizador, são também parte integrante desta área funcional, pois estão directamente relacionados com os artigos, ao consistirem em atributos adicionais associados a estes. Os dados mais importantes são guardados nas tabelas UDA e UDA_VALUES, existindo depois outras tabelas para cada tipo de UDA.

A classificação dos artigos é auxiliada por diversos componentes do sistema ORMS. Um deles é a possibilidade de criar listas de artigos e locais, que têm características comuns, facilitando a realização de operações em massa. Estas listas estão armazenadas em tabelas como a SKULIST_HEAD, para os itens, e a LOC_LIST_HEAD, para os locais. Outras tabelas existem para armazenar mais detalhes relativos a estas listas.

3.4.3 Manutenção de custos

O custo inicial dos artigos é definido no processo de introdução de um novo artigo no sistema, ao nível do artigo, do fornecedor e do país ou localização de origem. Todas as alterações posteriores a estes valores deverão ser feitas com recurso ao processo de alteração de custos. Estes eventos são armazenados na base de dados, para efeitos estatísticos e de cálculo de resultados, estimativa de custos e fixação de preços.

Das tabelas que contêm estes dados, destacam-se as COST_SUSP_SUP_HEAD e a COST_SUSP_SUP_DETAIL. As diversas razões que justificam um evento deste tipo são armazenadas na tabela COST_CHG_REASON.

3.4.4 Manutenção de negócios

Um negócio é um acordo feito entre o retalhista e o fornecedor, firmando benefícios e obrigações entre si, durante um período determinado. Normalmente resulta num desconto dado pelo fornecedor ao retalhista, resultando num aumento das vendas do fornecedor e numa relação com o retalhista mais longa. Estes descontos influenciam o cálculo dos custos associados aos produtos, que por sua vez têm impacto na definição dos preços a praticar e das margens de lucro a praticar pelos retalhistas.

Os dados relativos a negócios são guardados em tabelas como a FIXED_DEAL, para negócios em que o retalhista não recebe descontos, mas um pagamento por promover

determinado produto do fornecedor em questão, a DEAL_HEAD e a DEAL_DETAIL, para todos os negócios, e a DEAL_ITEMLOC, para os artigos envolvidos em negócios. A associação com as ordens de compra está patente nos registos das tabelas ORDLOC_DISCOUNT e ORDHEAD_DISCOUNT.

3.4.5 Ordens de compra

As ordens de compra são criadas quer manualmente, pelos utilizadores, quer automaticamente, pelas funcionalidades ligadas à reposição. O objectivo é manter os níveis de *stock* dos produtos para conseguir cumprir com os níveis de procura nas lojas.

Esta funcionalidade é suportada pela base de dados através de tabelas como a ORDHEAD, que contém informação base de cada ordem de compra, a ORDSKU, com os artigos que constam de cada ordem lançada no sistema, e a ORDLOC, com informação sobre a combinação entre ordens, artigos e lojas ou armazéns. Relacionada com esta área, temos a área de expedições, pois cada ordem aprovada dá origem a *shipments*. Estes eventos estão nas tabelas SHIPMENT e SHIPSKU.

3.4.6 Gestão de inventário

Na gestão de inventário estão envolvidas diversas operações, como a realização de transferências de artigos entre locais da organização, recolha de produtos para retornar ao fornecedor, ajustes de inventário e contagens de *stocks*.

Para os eventos de transferências, são guardados dados nas tabelas TSFHEAD e TSFDETAIL. Estes eventos, tal como as ordens de compra, geram também registos nas tabelas SHIPMENT e SHIPSKU. São ainda armazenadas as entidades que podem fazer parte de transferências, na tabela TSF_ENTITY.

Os eventos de retorno de mercadoria aos vendedores geram informação que é guardada nas tabelas RTV_HEAD e RTV_DETAIL. Os ajustes de inventário estão todos nas tabelas INV_ADJ.

As contagens de *stock*, indispensáveis para se saber as quantidades reais de cada produto em cada local, representam uma quantidade considerável de informação associada, por serem relativamente numerosas e acontecerem em diversos locais. Assim, espera-se que tabelas como a STAKE_SKU_LOC tenha um número elevado de registos, pois armazena informação relativa a cada artigo de cada localização que participa numa dada contagem de *stock*. Todos os outros dados relativos a esta operação estão em outras tabelas, nomeadamente na STAKE_HEAD, STAKE_LOCATION e STAKE_PRODUCT. O agendamento destas tarefas consulta-se na tabela STAKE_SCHEDULE.

3.4.7 Reposições

O sistema de reposição automática é responsável por monitorizar constantemente os níveis de quantidade dos produtos em cada loja e desencadear operações no sentido de encomendar novos produtos assim que os níveis de *stock* caiam para valores que possam não responder às necessidades da procura dos consumidores e dar origem a prateleiras desagradavelmente vazias. Os indicadores que suportam esta actividade podem ser encontrados nos dados relativos aos fornecedores, às opções de sistema, aos artigos em cada local e à combinação entre artigos, fornecedores e país de origem, para medir o tempo máximo que uma entrega demorará.

A informação gerada por esta funcionalidade é guardada nas tabelas REPL_DAY, que diz os dias em que cada artigo deve ser revisto, REPL_ITEM_LOC, com informação sobre cada artigo em cada local, e REPL_RESULTS, onde se encontram os resultados de cada evento de reposição. Quando um artigo não se encontra disponível para encomendar ou transferir, o

sistema consulta as tabelas SUB_ITEMS_HEAD e SUB_ITEMS_DETAIL, para encontrar um produto equivalente que esteja disponível.

3.4.8 Stock Ledger

O *Stock Ledger* dá acesso a informações financeiras relativas a transacções, incluindo compras, transferências, reduções de preços e vendas nas lojas. Estas informações são mantidas a nível diário, semanal e mensal, organizadas por *subclass* e local. O *Stock Ledger* é a ponte entre o ORMS e sistemas externos de gestão financeira, processando os dados e tornando-os acessíveis a estes sistemas.

As tabelas relativas a esta área são das mais volumosas da base de dados, pois esta funcionalidade gera um número elevado de registos. Assim, a TRAN_DATA fica com todos os dados relativos a transacções que ocorrem durante o dia, sendo depois truncada quando os seus registos passam para a tabela de histórico de transacções, a TRAN_DATA_HISTORY. A DAILY_DATA oferece um histórico, diário, de transacções a nível de vendas que é utilizado para cálculo de estatísticas de final do mês, comparação com períodos iguais em anos anteriores e cálculo de níveis de *stock*. A WEEK_DATA contém estes dados agrupados por semana e *subclass*, sendo construída com base na DAILY_DATA. A MONTH_DATA contém um histórico de transacções mensal, agrupado por *department* e por local, com base nos dados da DAILY_DATA. Os orçamentos são guardados nas tabelas HALF_DATA_BUDGET e MONTH_DATA_BUDGET.

3.4.9 Variáveis de sistema

Esta área refere-se à definição de parâmetros do sistema que são usados por todas as funcionalidades, para controlo. As tabelas que constam desta área são controladas pelo administrador da base de dados, que insere os registos necessários durante a instalação do produto. Daqui destaca-se a importância da SYSTEM_OPTIONS, que contém diversos parâmetros definidos pelo utilizador, e a PERIOD, que armazena datas para uso nos processos do *batch*, na produção de relatórios e outros programas.

3.5 Processamentos de retaguarda

Os processamentos de retaguarda têm associado o conceito de *batch*. O *batch* consiste na execução de um grupo de programas em bloco. Estes programas correm sem a intervenção do utilizador, sendo utilizados para diversas operações, como o processamento de elevadas quantidades de informação, interacção e troca de dados com sistemas externos e também intervenções a nível interno, como actualização de dados ou outras tarefas de manutenção.

Estes programas conseguem tratar grandes quantidades de informação de uma forma rápida e eficiente, levando a cabo tarefas que, embora pudessem ser feitas através de transacções *online*, não têm um impacto tão elevado na performance do sistema, se feitas por este grupo de programas. Dado que negociam com uma carga elevada de dados e necessitam de alocar uma quantidade considerável de recursos, estes programas são agendados para execução em alturas de menor actividade do sistema, sendo que, no caso em concreto dos retalhistas, essa altura coincide com o período em que as lojas estão fechadas ao público, normalmente durante a noite.

O agendamento destes programas é feito de forma automática, usando *software* específico para o efeito. Este deve ser feito de acordo com as especificações dos programas que constituem o *batch*, pois existem processos que dependem de outros, havendo uma ordem específica a seguir.

3.5.1 Tipos de programas

Os programas que constam do *batch* do Oracle Retail (OR) são de diversos tipos, com funções variadas:

- Programas de envio de dados submetem dados de sistemas externos e armazenam-nos na base de dados do OR.
- Programas de extracção de dados recolhem dados do OR e formatam-nos para enviar aos sistemas externos. Como exemplo, o programa *posupld*, que faz parte do *batch* nocturno do ORMS, reúne informações sobre artigos e locais e envia-os para os sistemas que se encontram nas lojas.
- Programas de manutenção do sistema são responsáveis por actualizar o sistema, como por exemplo incrementar a data no final de cada ciclo de *batch*, entre outras operações deste tipo.
- Programas de manutenção funcional processam dados referentes a áreas funcionais. Como exemplo, o programa *storeadd* actualiza tabelas que contêm informação relativa a lojas, com o objectivo de adicionar uma nova loja ao sistema.

3.5.2 Janela de execução

Como referido anteriormente, a execução dos programas de *batch* consome uma boa parte dos recursos do sistema, sendo poucas vezes viável conduzir as suas operações durante as horas em que o sistema está em uso, isto é, as lojas estão abertas ao público. Desta forma, existe uma janela de oportunidade, normalmente durante a noite, em que as lojas estão encerradas, no qual os programas são lançados para o processamento de todos os dados. Esta janela é o período de execução do *batch*. Para dar um exemplo, nos Estados Unidos, um retalhista que tenha lojas espalhadas pelo país, obedece a diferentes fusos horários. Se necessitar de ter os seus sistemas em plenas funções desde as 8h da manhã GMT-5 até às 21h GMT-8, o *batch* terá uma janela de execução de 8 horas. Consoante as regiões onde as lojas estejam, os retalhistas poderão ter mais ou menos tempo para executar as operações de *batch*.

3.5.3 Agendamento e fases

A ordenação e a definição de prioridades são críticas quando se aborda a execução dos programas do *batch*. De facto, existem alguns processos que, para realizarem as suas tarefas, necessitam de informação produzida por outros processos, que idealmente já acabaram de executar as suas operações. Por esta razão, torna-se indispensável a criação de um agendamento do lançamento dos programas, para assegurar a ordem de execução de cada um deles, garantindo que a informação requerida por cada um está sempre disponível quando este se prepara para arrancar.

O agendamento do *batch* do ORMS está dividido em fases, sendo que estas seguem uma ordem rígida, não sendo possível iniciar processos de uma fase sem que os processos da fase que a antecede tenham terminado. Dentro de cada fase, os diversos programas estão também ordenados, para garantir a satisfação de todas as dependências. Assim sendo, temos as seguintes fases e o tipo de tarefas realizadas em cada uma delas:

- **Fase 0** – nesta fase são feitas operações de manutenção da informação nas tabelas, nomeadamente a limpeza diária de dados obsoletos, actualização de taxas de câmbio e actualização de taxas de impostos;
- **Fase 1** – os programas desta fase preparam as tabelas para a interacção com os sistemas externos, que ocorre na Fase 2. Entre outros, o processo de variação de

stock executa actualizações sobre os dados referentes a níveis de *stock* dos produtos;

- **Fase 2** – durante esta fase, é recebida informação de diversos sistemas externos, sendo de realçar o envio de dados por parte dos sistemas dos pontos de venda;
- **Fase 3** – esta fase contém programas responsáveis por processar informação relativa a áreas centrais do ORMS, como compras, criação de encomendas, gestão financeira, negócios com fornecedores e reposição de produtos;
- **Fase 4** – aqui são enviados dados para fontes externas. A informação do sistema que foi alterada é reconstruída e processada, sendo também nesta fase que são enviados dados para os sistemas de previsão e planeamento.
- **Ad Hoc** – nesta fase particular, os programas podem ser lançados a qualquer altura, sem preocupação com dependências, uma vez que não utilizam nem disponibilizam dados de e para outros processos.

3.5.4 Processamento paralelo

O conceito de *multithreading*, ou processamento paralelo, associado à execução de programas do *batch* tem um significado diferente do conceito associado aos sistemas operativos. Na verdade, consiste no lançamento de várias instâncias, ou *threads*, do mesmo processo simultaneamente, com o intuito de acelerar o processamento de grandes quantidades de dados, permitindo operações de reinício ou de recuperação a nível de cada instância e assumindo que os dados estão distribuídos de forma equivalente.

Os dados são divididos pelas instâncias do processo, agrupados por um campo não-nulo, denominado de *Driver*. Desta forma, são criadas unidades de trabalho, ou *Units Of Work* (UOW), que consistem em pedaços de informação a ser processada pela instância do processo. Em caso de falha, apenas este pedaço precisa de voltar a ser processado, salvando-se o resto dos dados. Os registos referentes aos diversos *drivers* definidos para cada programa, bem como o número máximo de *threads* que podem ser lançadas para cada um deles, estão armazenados numa vista do sistema, a *restart_control*, onde o ORMS verifica as características definidas para cada processo na altura do seu lançamento.

3.6 Análise de casos de estudo

Para o exercício de dimensionamento foram recolhidas estatísticas de bases de dados de sistemas já implementados e em fase de produção. Estas estatísticas têm o principal objectivo de permitir a avaliação dos volumes de dados normalmente encontrados para cada tabela, identificando-se as mais importantes e que representam um maior impacto no tamanho total da BD.

Assim, foi criado um pequeno *script* em SQL, que foi executado em 4 diferentes clientes da Wipro Retail, com dimensões e actividades principais distintas. Essa consulta à base de dados revelou valores sobre o número de registos de cada tabela, blocos ocupados por cada tabela, tamanho médio do registo e tamanho da tabela, por multiplicação dos registos pelo seu tamanho médio. A análise destes casos de estudo proporcionou o primeiro contacto com números reais de casos práticos de implementações de sistemas Oracle Retail. Para o projecto foram apenas considerados os dados relativos ao ORMS, por ser esse o objecto de estudo.

3.6.1 Caso 1

O primeiro caso analisado trata-se de um retalhista do Médio Oriente, com cerca de 95 lojas e mais de 250.000 artigos definidos no sistema, entre licores e tabacos, electrodomésticos, relógios, perfumes e cosméticos, roupas ou brinquedos. Na tabela seguinte apresenta-se alguns números relativos a este caso

Tabela 4 – Dados relativos ao caso 1

Descrição	Valor
Número de tabelas	1775
Total de registos (milhões)	539
Tamanho das tabelas (GB)	36,8
Número de índices	2260
Total de registos indexados (milhões)	259
Tamanho dos índices (GB)	10

3.6.2 Caso 2

O caso 2 refere-se a uma organização do Médio Oriente, que opera em diversos canais de distribuição, com cerca de 500 lojas espalhadas pela região. Segundo os valores apresentados, oferece mais de 130.000 artigos, desde moda para as diversas idades, até aos cosméticos, passando por uma marca dedicada exclusivamente a brinquedos. Os dados recolhidos estão representados na tabela abaixo.

Tabela 5 – Dados relativos ao caso 2

Descrição	Valor
Número de tabelas	1122
Total de registos (milhões)	352
Tamanho das tabelas (GB)	25,4
Número de índices	1963
Total de registos indexados (milhões)	896
Tamanho dos índices (GB)	36,9

3.6.3 Caso 3

No caso 3, foi estudado um retalhista internacional, da área de moda e acessórios, que conta com cerca de 640 lojas e 12.000 pontos de venda espalhados por mais de 40 países. No sistema estão registados mais de 1.000.000 de artigos, o que representa uma dimensão considerável para este cliente.

Tabela 6 – Dados relativos ao caso 3

Descrição	Valor
Número de tabelas	945
Total de registos (milhões)	570
Tamanho das tabelas (GB)	35,7
Número de índices	1700
Total de registos indexados (milhões)	1555
Tamanho dos índices (GB)	63,5

3.6.4 Caso 4

Este caso é referente a um cliente sediado em França, com cerca de 300 lojas registadas no sistema. Ainda olhando aos dados resultantes da consulta, comercializa mais de 290.000 artigos na área de moda e acessórios. Quanto à base de dados do sistema, apresenta os seguintes valores:

Tabela 7 – Dados relativos ao caso 4

Descrição	Valor
Número de tabelas	1456
Total de registos (milhões)	925
Tamanho das tabelas (GB)	61,8
Número de índices	1719
Total de registos indexados (milhões)	2287
Tamanho dos índices (GB)	92,6

3.7 Resumo e conclusões

Neste capítulo foi caracterizado o problema, apresentando-se uma descrição do que realmente é o Oracle Retail, seguido de uma apresentação do problema proposto. Em todo o resto do capítulo são relatados com detalhe todos os aspectos que o problema envolve, como o sistema ORMS e todas as componentes a si associadas e que constituíram objecto de estudo para a construção da solução, como o modelo de dados e o *batch*. A análise dos casos práticos ofereceu o primeiro contacto com números reais de implementações do ORMS, para se ter uma ideia da dimensão das bases de dados a serem planeadas com recurso à ferramenta a ser produzida como resultado do projecto em causa.

4 Desenvolvimento da solução

Neste capítulo, apresenta-se em detalhe a solução encontrada para resolver o problema encontrado. Inicialmente faz-se uma explicação e justificação da metodologia adoptada, seguido duma cronologia das fases seguidas para a elaboração da ferramenta de dimensionamento. Esta ferramenta é então apresentada e analisada, finalizando-se o capítulo com uma análise de sensibilidade.

4.1 Metodologia de dimensionamento adoptada

No capítulo 2, de revisão bibliográfica, foram apresentadas algumas metodologias de dimensionamento de bases de dados, recolhidas após uma pesquisa efectuada nos meandros deste domínio. Estas técnicas são, na sua maioria, focadas na determinação do espaço ocupado pela base de dados num dado momento, e na sua posterior monitorização, a fim de estimar um factor de crescimento do volume de dados e ajustar as capacidades físicas do sistema em concordância. Para o problema em causa, estas metodologias não se aplicam totalmente, uma vez que o propósito principal é elaborar uma estimativa do volume de dados, partindo de uma base de dados completamente vazia, e de indicações quanto aos volumes de dados a serem introduzidos.

4.1.1 Armazenamento físico

Os modelos teóricos podem apresentar desvios significativos relativamente à implementação, uma vez que são baseados em situações extremas, por não se conhecerem os dados concretos. Consequentemente, e em bases de dados muito grandes, o ideal é seguir uma política de amostragem de dados, para suprir estes desvios relativamente às necessidades reais.

Desta forma, optou-se pela análise de alguns casos práticos, de sistemas já implementados e em ambiente de produção, para perceber os volumes envolvidos no processo, e também os tamanhos de cada tabela da base de dados. Indo de encontro a uma das metodologias descritas, a de recolha de estatísticas, procedeu-se à recolha dos tamanhos médios de cada registo, o número de registos armazenados e o número de blocos ocupados. Estas estatísticas foram recolhidas para cada uma de todas as tabelas que constituem uma instalação genérica do sistema, pois o projecto visa estimar as necessidades deste tipo de implementações típicas. Os sistemas podem ser configurados à medida dos pedidos dos clientes, sendo que nesse caso as necessidades são

avaliadas recorrendo a outros métodos e a testes de carga, o que não se enquadra no âmbito deste projecto.

Para os índices, foram recolhidas estatísticas referentes ao número de registos, ao número de nós-folha, número de blocos e tamanho ocupado pelo índice, em *bytes*. O objectivo é determinar o tamanho teórico ocupado pelo índice, pela análise dos nós-folha, e o tamanho efectivo, pela análise do número de blocos e os *bytes* ocupados. Com o cruzamento destes valores, é possível determinar um factor de crescimento dos índices no normal decorrer da utilização do sistema.

4.1.2 Carga de processador

Para avaliar a carga de processamento requerida por uma instalação, e calcular as necessidades do sistema em termos de número de processadores a incluir nos servidores, procedeu-se à análise da configuração do *batch* nos diversos casos. Consultando uma das vistas da base de dados, a *restart_control*, é possível visualizar o número de *threads* lançadas em paralelo por cada um dos processos. Estudando a informação recolhida sobre os processos que compõem o *batch*, estabelece-se uma relação entre o tipo de áreas funcionais sobre as quais estes operam, nomeadamente as tabelas envolvidas e o seu volume de dados, e o número de *threads* configurado.

PROGRAM_NAME	PROGRAM_DESC	DRIVER_NAME	NUM_THREADS	UPDATE_ALLOWED	PROCESS_FLAG	COMMIT_MAX_CTR
auditprg		NONE	1	Y	T	1000
auditsys		NONE	1	Y	T	1000
ccprg		NONE	1	Y	T	1000
cednid		NONE	1	Y	T	1000
cmpprg		NONE	1	Y	T	10000
cmpupld		NONE	1	N	F	10000
cntrmain		NONE	1	N	T	1000
cntrordb		CONTRACT_NO	1	Y	T	1
cntrprss		DEPT	1	Y	T	1
costcalc	calculate deal discounts into future_cost	SUPPLIER	5	Y	T	1000
cremhierdly		NONE	1	Y	T	1000
dealact	Calculates actuals information	NONE	1	Y	T	1000
dealcls	close deals that have reached close_date	NONE	1	Y	T	1000
dealday	Sums up to deal day	DEPT	5	Y	T	1000

Figura 12 – Excerto da vista *restart_control*

Desta forma, traduz-se a quantidade de informação envolvida em cada processo no número de *threads* a serem lançadas em paralelo. Definindo-se um quociente de 2 *threads* por cada *core*, calcula-se o número estimado de processadores a serem instalados no sistema, partindo dos processos com maiores necessidades.

4.1.3 Memória

Na estimativa das necessidades de memória do sistema, torna-se importante analisar o número médio de utilizadores que estão ligados em simultâneo, avaliar os processos em execução aquando do estado *online* e, principalmente, as tabelas que são mais vezes acedidas e manipuladas.

A memória principal terá que ser suficientemente grande para manter em *cache* os dados que são mais vezes acedidos, com o objectivo de melhorar a performance do sistema, reduzindo o número de operações de E/S nos discos, que são responsáveis pelo aumento do tempo de resposta. Os processos utilizam espaços da memória para tarefas de ordenação, para cálculos intermédios e armazenamentos temporários, pelo que devem também ser estudados, pois contribuem para a estimativa final do tamanho total da memória principal.

4.2 Etapas de elaboração da solução

Esta secção pretende explicar em detalhe todos os passos seguidos na elaboração da ferramenta de dimensionamento. Para facilitar a explicação, optou-se por utilizar um diagrama que representa as diversas actividades e as suas precedências, para que se tenha uma ideia da cronologia dos eventos, que serão depois explicados em maior detalhe.

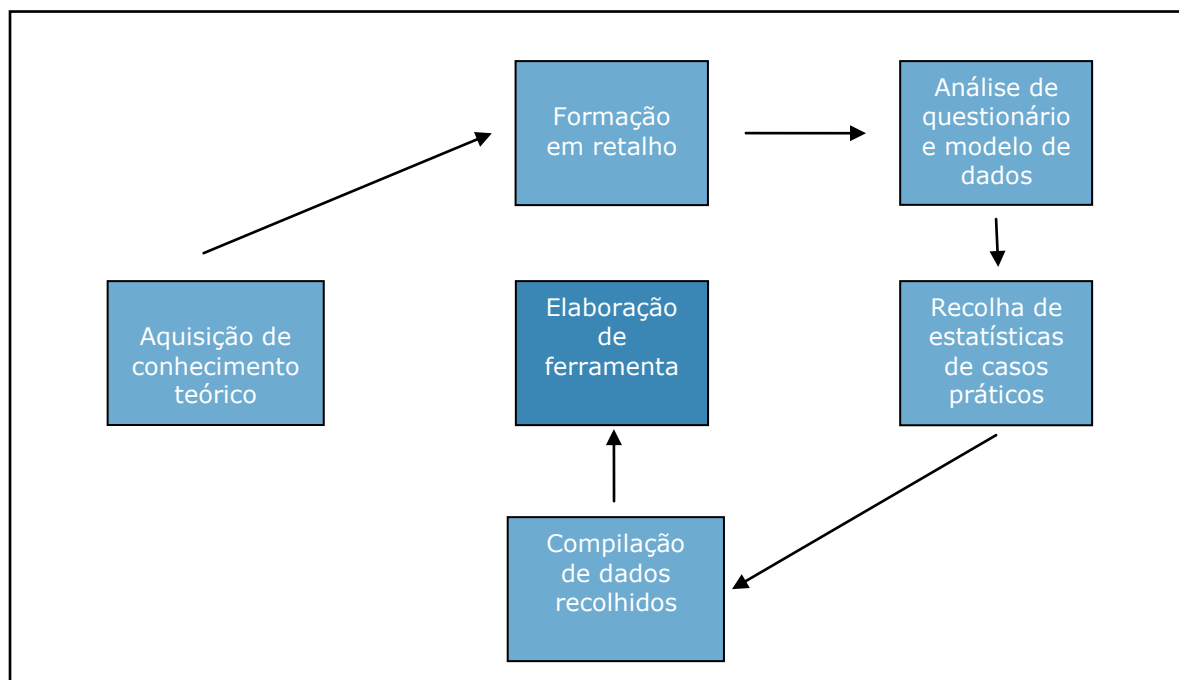


Figura 13 – Etapas de desenvolvimento da solução

4.2.1 Aquisição de conhecimento teórico

Para iniciar a abordagem ao problema, optou-se por estudar a fundo as bases de dados, nomeadamente as da Oracle. Foi feita uma análise da documentação existente sobre o tema, com o intuito de perceber a sua mecânica, os processos envolvidos em cada operação feita sobre a informação armazenada e os componentes que dela fazem parte. Este estudo foi importante, pois permitiu entender todo o funcionamento das bases de dados, a maneira como a informação é armazenada nas diversas estruturas lógicas e físicas, o que resultou numa familiarização com conceitos do domínio, que proporcionaram uma mais fácil interacção com os dados recolhidos dos casos práticos, e uma conceptualização dos processos a aplicar na obtenção da solução para o problema.

Foi também levada a cabo uma pesquisa sobre métodos de dimensionamento de bases de dados e planeamentos de capacidades normalmente aplicados na área. A análise das práticas encontradas permitiu a formulação da metodologia seguida para a resolução do problema, ao introduzir conceitos e técnicas desconhecidas, que devidamente associadas resultaram num processo adequado à elaboração da solução pretendida.

Para a estimativa dos tamanhos ocupados pelos índices, foi utilizada uma técnica que resultou de uma pesquisa sobre métodos de gestão do espaço em bases de dados. Esta técnica, utilizada normalmente para avaliar a necessidade de efectuar operações de reconstrução de índices, revelou-se útil para determinar factores de crescimento dos índices, que se traduzem numa diferença entre o tamanho teórico que estes deviam apresentar e o tamanho que

efectivamente têm. O método e a sua utilidade são explicados com maior detalhe numa posterior altura neste documento.

Com o objectivo de aprimorar o resultado do dimensionamento, fez-se uma recolha das diferentes tecnologias de redundância de dados, de extrema importância em ambientes deste tipo, onde a informação é crítica e tem que estar sempre acessível e segura, sob pena de o negócio parar, caso esta seja de alguma forma perdida ou danificada, o que resulta em danos irreparáveis para o retalhista. Ao mesmo tempo, estas tecnologias permitem também ganhar em termos de performance, pelo que a sua aplicação deve ser estudada de acordo com os dados a armazenar.

O resultado deste estudo pode encontrar-se no capítulo 2 deste documento, sendo que os principais conceitos a reter residem na parte relativa ao armazenamento das bases de dados.

4.2.2 Formação em retalho

O modelo de negócio, as operações mais vitais, os conceitos e o *modus operandi* do mundo da venda a retalho são matérias fulcrais para o desenvolvimento da solução, pois o material que serviu de base à sua elaboração contém, obviamente, diversas referências a este domínio. Assim, foi necessário adquirir conhecimentos sobre este negócio, para conseguir reter e processar toda a informação que consta dos materiais utilizados.

A formação recebida facilitou a posterior aproximação ao sistema ORMS, à base de dados e seu modelo de dados e ao questionário de dimensionamento da Oracle.

4.2.3 Análise de questionário e modelo de dados

Depois de consolidado o conhecimento teórico, deu-se lugar à análise do modelo de dados do ORMS e do questionário de dimensionamento utilizado pela Oracle nestes exercícios. Este questionário, que se pode encontrar na figura abaixo, em excerto, e em anexo na versão integral, contém cerca de 400 perguntas, agrupadas por áreas funcionais do sistema, com o intuito de avaliar as necessidades de negócio do cliente, recolhendo informações relativas a volumes de transacções, número de lojas e armazéns, artigos a serem registados, históricos de vendas, entre outros.

Oracle Retail Sizing Estimator Product: RMS, RPM Product version: 12.0.x Estimator version: 1.6 Estimator version date: 23-Feb-06 					
Question #	Area	Sample	Req'd?	Impact	Question
1000	Sizing Admin	Large Retailer	Y	N/A	Client Name
1001	Sizing Admin	J. Doe	N	N/A	Client Business Lead
1002	Sizing Admin	J. Doe	N	N/A	Oracle Retail Business Consultant
1003	Sizing Admin		N	N/A	Oracle Retail Technical Architect
1004	Sizing Admin	J. Doe	N	N/A	Oracle Retail Project Manager
1005	Sizing Admin	2	Y	N/A	Sizing Iteration
1006	Sizing Admin	30-Nov-2004	Y	N/A	Sizing Submission Date
1007	Sizing Admin	36	Y	High	Time Horizon for sizing, in months
1010	Org Hierarchy	3	Y	Low	How many chains will be defined in RMS?
1011	Org Hierarchy	141	Y	Low	How many areas will be defined in RMS?
1012	Org Hierarchy	388	Y	Low	How many regions will be defined in RMS?
1013	Org Hierarchy	388	Y	Low	How many districts will be defined in RMS?
1014	Org Hierarchy	22	Y	High	How many physical warehouses will be defined in RMS?
1015	Org Hierarchy	1.010	Y	High	How many stores will be defined in RMS?
1016	Org Hierarchy	100.0%	Y	Low	What percentage of stores will attributes defined?
1017	Org Hierarchy	100.0%	Y	Low	What percentage of warehouses will have attributes defined?

Figura 14 – Excerto do questionário de dimensionamento

O cruzamento do modelo de dados com o questionário é um dos exercícios mais importantes e significativos de todo este projecto. Consistiu na elaboração de um mapeamento entre as questões apresentadas pela Oracle e as tabelas da base de dados.

Para levar a cabo esta tarefa, cada pergunta foi cuidadosamente analisada, assinalando-se a área funcional em que se inseria, e o seu objectivo.

Por consulta da área correspondente no modelo de dados, reúnem-se todas as tabelas que estão relacionadas com a resposta dada à pergunta, quer por fazerem parte da mesma área funcional, quer por conterem na sua estrutura registos que se identificam com o objecto da pergunta. Como forma de suportar este mapeamento, verificou-se ao mesmo tempo os formulários do ORMS que se relacionam de alguma forma com a área funcional e o objecto da pergunta, observando-se os campos a preencher para efectuar a operação em causa e confrontando com as tabelas identificadas, com a finalidade de validar o trabalho efectuado e eliminar ambiguidades ou más escolhas.

Oracle Retail Sizing Estimator					
Product: RMS, RPM					
Product version: 12.0.x					
Estimator version: 1.6					
Estimator version date: 23-Feb-06					
Area	Req'd?	Impact	Question	Tabelas envolvidas	
Item Loc	Y	High	What is the average number of tran level items per store including pack items?	ITEM_LOC, ITEM_LOC_SOH, ITEM_LOC_HIST, ITEM_LOC_HIST_MTH	
Item Loc	Y	High	What is the average number of tran level items per warehouse (or virtual wh, if multi-channel) including pack items?		
Item Loc	Y	High	What is the average number of departments with inventory per store?		
Item Loc	Y	High	What is the average number of classes with inventory per store?		
Item Loc	Y	High	What is the average number of subcls sets with inventory per store?		

Figura 15 – Exemplo de exercício de mapeamento

Para melhor se perceber a dimensão deste exercício, explica-se em maior detalhe a sua mecânica, dividindo-se o questionário nas diversas áreas funcionais e realçando as principais dificuldades e pormenores encontrados ao longo desta etapa.

4.2.3.1 Hierarquia Organizacional

O conjunto de perguntas nesta área visa recolher as dimensões da organização do retalhista, questionando-o sobre o número de *chains*, *areas*, *regions*, lojas ou armazéns. Deste conjunto, destacam-se as perguntas sobre o número de lojas e armazéns, cujo valor integra diversas tabelas, particularmente as chaves primárias de muitas delas, revestindo-se de um carácter sensível na determinação do número de registos em cada uma destas tabelas.

4.2.3.2 Multi-canal

Nesta área há um particular destaque para o número médio de armazéns virtuais por cada armazém físico a ser definido, pois este afecta não só as tabelas relativas aos armazéns, mas também as tabelas relativas a *stocks* e manuseamento de artigos, pelo que o seu impacto é moderado.

4.2.3.3 Graduação de lojas, Características de locais, Listas de locais

Estas três áreas aparecem aqui agrupadas, uma vez que os valores previstos de registos são baixos, sendo pouco influentes para o resultado final da estimativa do dimensionamento. Estes correspondem a funcionalidades de classificação e agregação de lojas e armazéns, para efeitos de produção de estatísticas e relatórios pelo sistema, não apresentando impacto em mais tabelas do que as destinadas à retenção destes dados.

4.2.3.4 Hierarquia mercadológica

Os dados mais importantes a processar do conjunto de perguntas nesta área são o número de *departments*, *classes* e *subclasses*. Estes entram nas chaves primárias de diversas tabelas, nomeadamente as de histórico de transacções que, devido ao elevado montante de informação gerado pelas inúmeras transacções que acontecem derivadas da normal actividade do retalhista,

atingem tamanhos consideráveis, figurando entre as tabelas com maior número de registos. Assim, estes valores afectam tabelas como DEPS, STORE_DEPT_AREA, DEPT_SALES_FORECAST, DEPT_SALES_HIST, CLASS, CLASS_SALES_FORECAST, CLASS_SALES_HIST, SUBCLASS, SUBCLASS_SALES_FORECAST, SUBCLASS_SALES_HIST ou outras, menos evidentes, como a DAILY_DATA, MONTH_DATA ou HALF_DATA. Esta identificação resulta de uma análise exaustiva de todo o modelo de dados, na procura de tabelas que incluam colunas para estes valores. As outras perguntas, quanto ao número de responsáveis de vendas ou de compras, têm pouco impacto no resultado final, pois são dados que não figuram em outras tabelas, sem ser a BUYER, ou a MERCHANT, que pela análise dos casos práticos se pode verificar não atingirem valores consideráveis de registos introduzidos.

4.2.3.5 Fornecedores, Características de fornecedores, Parceiros

Optou-se por agregar estas duas áreas nesta explicação, na medida em que são relacionadas, por se referirem aos fornecedores. Das 7 perguntas que são feitas ao cliente, a mais importante é, sem margem para dúvida, a do número de fornecedores com que trabalha. A informação relativa aos fornecedores afecta áreas como ordens de compra, devoluções, reposição de produtos, seja automática ou manual, Electronic Data Interchange⁵, contratos e negócios.

Todos os artigos têm um ou mais fornecedores associados, pelo que todas as tabelas de artigos e de informação relativa a *stocks* armazenam dados relativos aos fornecedores. Estes aparecem também diversas vezes associados a lojas e armazéns, e podem ter múltiplos países de origem, o que resulta na geração de um elevado número de registos. Em suma, este é um dos valores mais importantes a considerar no dimensionamento, por ter um impacto elevado no sistema e em grande parte das tabelas da base de dados. As características dos fornecedores, ou *supplier traits*, são utilizadas para fins de agrupamento de fornecedores com características semelhantes, pelo que o seu impacto é baixo, pois apenas servem para efeitos estatísticos e de produção de relatórios, não atingindo volumes de dados relevantes.

Nas perguntas relativas a *partners*, os parceiros do retalhista, destaca-se a que se refere à quantidade de parceiros que têm níveis de *stock* associado, por serem transformadores de produtos ou outra actividade semelhante. Estes devem ser tidos em conta na altura de estimar tamanhos para tabelas de *stocks* e artigos, pois são influenciadas pela quantidade deste tipo de parceiros.

4.2.3.6 Artigos

Os artigos são a base do negócio do retalhista, pois são os bens comprados e vendidos, e que geram lucro. Desta forma, todas as perguntas que constam desta área foram analisadas com a máxima atenção, percorrendo-se todo o modelo de dados e identificando as tabelas que contêm colunas com informação relativa a estes.

Existem diversas tabelas que associam a informação relativa aos artigos com os dados mais variados, como os níveis de *stock* actuais, os locais físicos onde se encontram exemplares dos mesmos, os seus fornecedores, o histórico de vendas, as transacções efectuadas, as compras realizadas, devoluções, entre muitos outros. As maiores tabelas da base de dados, como a ITEM_SUPP_COUNTRY_LOC, a ITEM_MASTER, a ITEM_LOC ou TRAN_DATA_HISTORY, entre outras, têm referências a dados dos artigos, muitas vezes na combinação de registos que constituem a chave principal, o que faz com que os valores introduzidos pelo cliente quanto ao número de artigos que transacciona seja dos mais sensíveis, senão o mais sensível, para este exercício de dimensionamento, apresentando um impacto brutal

⁵ Interface electrónica que permite a entidades externas interagir com o sistema.

no tamanho final da base de dados, podendo esperar-se variações elevadas associadas a modificações destes valores.

4.2.3.7 Packs

Os *packs* são também a origem de muitas transacções no sistema. Na verdade, são artigos que se encontram em conjunto, podendo incluir uma ou mais variedades de itens no mesmo pacote, sendo comprados e vendidos à semelhança dos artigos singulares. No entanto, as perguntas relativas a esta área têm um carácter de menor importância, pois estão vocacionadas para as tabelas `PACKITEM`, `PACKITEM_BREAKOUT`, `PACK_TMPL_HEAD`, `PACK_TMPL_DETAIL` e `SIMPLE_PACK_TEMP`, que apenas armazenam informações relativas às características de cada *pack*, ou armazenam dados sobre modelos de *packs*, que facilitam a tarefa do utilizador na definição dos mesmos no sistema. A influência destes artigos no resto do sistema é avaliada por outras perguntas, que serão vistas mais à frente nesta explicação, pelo que a importância das respostas aqui encontradas é relativamente diminuída.

4.2.3.8 Localização de artigos

As tabelas que se encontram associadas a esta área funcional são responsáveis pelo cruzamento de informação relativa a artigos e aos locais onde estes se encontram, e em que quantidades. São também produzidos históricos baseados na combinação destes dados, classificados por diferentes critérios, como por *department*, *class* ou *subclass*, o que origina tabelas de tamanho considerável, e com um peso relevante na estimativa do espaço total ocupado pela base de dados.

São tabelas como a `ITEM_LOC`, `ITEM_LOC_SOH`, `ITEM_LOC_HIST` e `ITEM_LOC_HIST_MTH`, que são directamente influenciadas pelas perguntas feitas nesta área, onde se pretende respostas sobre números médios de artigos transaccionáveis por loja e por armazém, seguido do número médio de *departments*, *classes* e *subclasses* com inventário guardado nas lojas.

4.2.3.9 Fornecedores e artigos

As perguntas relativas a esta área funcional pretendem obter dados no que diz respeito à média de fornecedores existente por cada artigo, e o número médio de países de origem de cada fornecedor. O principal intuito destas perguntas é ajudar a prever o número de registos das tabelas `ITEM_SUPPLIER`, `ITEM_SUPP_COUNTRY`, `ITEM_SUPP_COUNTRY_LOC`, `ITEM_SUPP_COUNTRY_DIM` e `ITEM_SUPP_COUNTRY_BRACKET_COST`, que, como visto anteriormente, é influenciada por outros valores, nomeadamente os de artigos e locais físicos.

O impacto destes valores é moderado, por participarem nestas tabelas de volume considerável, pelo que a sua contribuição para o dimensionamento é importante.

4.2.3.10 Listas de artigos e atributos de artigos

As listas de artigos, ou *item lists*, são utilizadas como meio de classificação e ordenação dos artigos, ao agregarem itens que partilham algumas características, para facilitar a sua identificação. Estas desempenham um papel secundário no sistema, não sendo muito relevante a sua contribuição para o tamanho da base de dados. As questões apresentadas pretendem saber se a funcionalidade será utilizada, e quantas listas de artigos serão criadas, e com quantos artigos em média.

O ORMS permite definir atributos adicionais dos artigos, para colmatar algum tipo de exigências dos retalhistas, pelo que existem as tabelas ITEM_ATTRIBUTES e ITEM_IMPORT_ATTRIBUTES, utilizadas para armazenar estes dados. Dado que as funcionalidades de definição dos artigos oferecem suficientes opções para a classificação dos artigos, é raro ser necessária atribuir qualificadores adicionais, pelo que os valores aqui pedidos ao cliente não são relevantes para o exercício de dimensionamento.

4.2.3.11 Diferenciadores

Os *differentiators* são características dos produtos que possibilitam a sua classificação e agrupamento. Como exemplo tem-se cores, tamanhos, aromas ou cherios, entre outros atributos semelhantes.

Esta funcionalidade do sistema permite uma mais fácil ordenação dos produtos e a produção de estatísticas e relatórios mais elaborados e organizados. É também importante nas compras de produtos aos fornecedores, pois ao agrupar-se artigos segundo determinada característica comum, pode fazer-se encomendas com maior rapidez e facilidade, indo de encontro às necessidades da procura, que pode centrar-se em algum destes atributos.

No entanto, tal como grande parte das funcionalidades destinadas a classificação de artigos e produção de relatórios e estatísticas, o seu contributo apresenta um impacto pouco relevante no cálculo do tamanho a ocupar pela base de dados. As perguntas que constam desta área têm um objectivo mais funcional, no sentido de saber se a funcionalidade será utilizada ou não, e que atributos serão definidos no sistema. As tabelas directamente afectadas por estes valores resumem-se às que estão relacionadas com *differentiators*. Embora existam registos de outras tabelas com dados relativos a estes atributos, não integram a chave principal de nenhuma delas, pelo que não estão revestidos de um grau de importância considerável.

4.2.3.12 Atributos definidos pelo utilizador

Os *User-Defined Attributes* (UDA) são atributos adicionais definidos pelos retalhistas para ajudar na classificação dos seus produtos, facilitando a produção de relatórios e estatísticas. Estes são valores específicos, como marcas, país de origem, tipo de materiais de que é composto, entre outros, que ficam à escolha do utilizador, optimizando ainda a realização de operações em massa sobre os produtos registados no sistema.

As perguntas aqui encontradas pretendem avaliar o número de atributos de cada tipo a serem definidos, para estimar o número de registos a constar das tabelas com o prefixo “UDA_”. Estes atributos são encontrados em outras tabelas relativas a artigos e a transacções, mas não são de carácter obrigatório, pelo que o impacto das respostas nesta área funcional é de baixo impacto para o dimensionamento.

4.2.3.13 Transacções e histórico

Esta é a área do ORMS que gera mais tráfego de informação e, conseqüentemente, novos registos nas tabelas, expandindo as suas dimensões consideravelmente.

Aqui encontram-se perguntas relativas à percentagem de lojas abertas 7 dias por semana, à média de dias em que as outras lojas estão abertas, ao número de artigos vendidos por dia e por semana e à média de devoluções feitas pelos clientes. Tais valores permitem prever uma parte dos registos que se espera que as tabelas de transacções, nomeadamente a TRAN_DATA_A e TRAN_DATA_B, e as de auditação de vendas, como a SA_TRAN_ITEM, venham a armazenar.

É necessário dar particular atenção a esta área, uma vez que diversas operações no ORMS geram transacções, e conseqüentes registos nestas tabelas, não existindo aqui perguntas relativas

a essas operações. Assim, foi vital a utilização do sistema, para perceber que outras tarefas implicam a criação de registos de transacções, estando descritas nas próximas sub-secções do documento.

O histórico de transacções apresenta volumes elevados de informação, ocupando diversos *bytes* nos suportes de armazenamento. As perguntas aqui encontradas tencionam avaliar o número de dias de histórico de transacções a serem retidos, o que influencia directamente a tabela `TRAN_DATA_HISTORY`, uma das maiores em toda a base de dados, e o número de meses de histórico de vendas, questionando ainda a maneira como as vendas serão classificadas, se por *class*, *subclass*, *department* ou *differentiator*.

4.2.3.14 Calendário

Esta área contém apenas uma pergunta, que se refere ao número de anos a serem definidos no calendário do sistema, sendo de impacto extremamente baixo para o dimensionamento, embora tenha alguma relevância a nível funcional.

4.2.3.15 Stock Ledger

Na área de *Stock Ledger*, as perguntas encontradas pretendem saber se o sistema suportará a definição de várias unidades monetárias, o que terá um impacto significativo a nível operacional, mas a nível de volume de dados, cinge-se aos registos das tabelas `CURRENCIES` e `CURRENCY_RATES`, que não se espera que apresentem tamanhos consideráveis.

As perguntas restantes são importantes e sensíveis, pois questionam o cliente sobre a maneira como trata a informação desta área: se por dia, semana, mês ou semestre. Estas opções referem-se às tabelas `DAILY_DATA`, `WEEK_DATA`, `MONTH_DATA` e `HALF_DATA`, que contêm dados relativos a transacções, agrupados pelos períodos especificados, resultando em tabelas volumosas, devido ao elevado número de dados gerados pelas transacções. Assim, estas opções devem ser analisadas, para se saber se estas tabelas irão realmente armazenar estes dados, dado que a sua presença ou ausência afecta o resultado final do dimensionamento.

4.2.3.16 Imposto de Valor Acrescentado

O ORMS oferece a possibilidade de gerir a informação relativa ao IVA, cuja sigla inglesa é VAT, permitindo a definição de diversas taxas e regiões, associando-as para o cálculo correcto deste imposto.

Das perguntas feitas ao cliente nesta área, destaca-se a que se refere ao *stock ledger*, onde se pretende saber se os preços lá definidos contêm este imposto aplicado. Esta informação é importante para estimar os registos da tabela `VAT_HISTORY`, que guarda as transacções diárias de artigos que têm esta taxa aplicada, pelo que pode atingir volumes consideráveis. As restantes perguntas referem-se a recolher informações básicas sobre esta funcionalidade, como os códigos de taxas a definir, a média de valores de imposto por cada código e o número de regiões de imposto diferentes, sendo um conjunto de dados que em pouco altera o tamanho total de uma base de dados de uma instalação do ORMS.

4.2.3.17 Open-to-Buy

Esta área é pouco relevante no que toca ao dimensionamento, uma vez que a utilização da funcionalidade em causa em pouco altera o volume de dados das tabelas. Nas perguntas apresentadas, apenas se pretende saber se a funcionalidade é de facto utilizada ou não, sendo pedidos valores associados a tempo de histórico a manter, não sendo esperado um número considerável de registos gerados.

4.2.3.18 Ordens de compra

O mapeamento das perguntas desta área com as tabelas da base de dados tornou-se um pouco mais complexo, na medida em que envolve outras áreas, e requer um conhecimento do processo de elaboração e aprovação de ordens de compra.

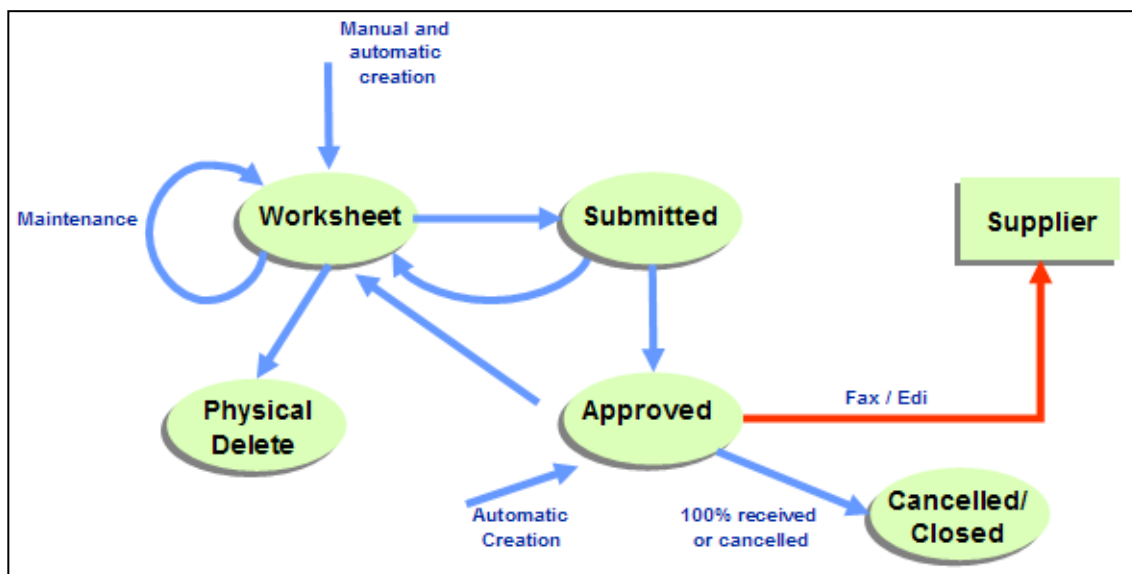


Figura 16 – Operações possíveis sobre ordens de compra

Quando uma ordem é criada, insere-se um registo nas tabelas ORDHEAD e ORDCUST, sendo que nesta última isto acontece apenas para encomendas dos consumidores. Como este é um processo central na actividade do retalhista, a pergunta que se refere ao número previsto de ordens de compra criadas manualmente por mês, reveste-se de um carácter sensível, prevendo-se um número considerável de registos nas tabelas associadas a estas operações.

A pergunta seguinte pretende avaliar a percentagem de ordens criadas que são efectivamente aprovadas. Quando uma ordem é aprovada, os seus detalhes são publicados em ORDER_DETAILS_PUBLISHED e ORDER_PUB_INFO. As tabelas ORDSKU e ORDLOC recebem os dados relativos aos artigos e aos locais que a encomenda envolve, respectivamente. É necessário criar um evento de envio de mercadoria, pelo que as tabelas SHIPMENT e SHIPSKU recebem novos registos e a ORDLOC_EXP passa a conter um registo com as despesas associadas a esta ordem de compra. Ainda olhando às tabelas ORDSKU e ORDLOC, são pedidos no questionário valores referentes à média de artigos e de locais que constam das ordens de compra, para estimar o volume de informação a armazenar nestas tabelas.

Posteriormente, ao cliente é pedido que indique a percentagem de ordens que são revistas, e a média de revisões efectuadas, o que envolve as tabelas ORDHEAD_REV, REV_ORDERS, ORDLOC_REV e ORDSKU_REV, onde não são esperados muitos registos, pois normalmente as operações de revisão não são numerosas. Finalmente, o cliente deverá indicar o período de histórico a manter, que influenciará todas as tabelas referidas, pois estas retêm a informação relativa às ordens de compra até serem limpas pelos programas de manutenção, no final do período de histórico definido.

4.2.3.19 Contratos

Os contratos feitos com os fornecedores são introduzidos no sistema, permitindo-lhe escolher a melhor fonte de artigos, na altura de gerar ordens de compra.

As perguntas aqui efectuadas têm pouco impacto no dimensionamento, uma vez que apenas envolvem as tabelas referentes aos contratos, com o prefixo "CONTRACT_". As

primeiras perguntas pretendem saber o número previsto de contratos a registar no sistema, envolvendo a tabela `CONTRACT_HEADER`. Em seguida, questiona-se o cliente quanto à média de artigos prevista por tipo de contrato, sendo as suas respostas a base para a estimativa dos registos nas tabelas `CONTRACT_DETAIL`, `CONTRACT_COST`, `CONTRACT_ORDSKU` e `CONTRACT_MATRIX_TEMP`.

Estas tabelas são ainda influenciadas pela pergunta que se refere à percentagem de contratos efectivamente aprovados, acrescentando-se à lista as tabelas `CONTRACT_ORDHEAD` e `CONTRACT_ORDLOC`, pois a aprovação de contratos desencadeia a criação de ordens de compra.

A funcionalidade, no entanto, não gera normalmente grandes níveis de informação, pelo que a sua influência no tamanho final da base de dados é relativamente pequena.

4.2.3.20 Negócios e provas de desempenho

Os negócios são armazenados no sistema, com o mesmo propósito dos contratos: conseguir sempre o melhor fornecedor na altura de comprar novos artigos para venda.

As perguntas que constam desta área têm um carácter mais funcional, não sendo as suas respostas tão relevantes para o exercício de dimensionamento. Embora os dados introduzidos no sistema a este nível influenciem diversas áreas, como as reposições ou a criação de ordens de compra, a informação gerada é diminuída, resumindo-se às tabelas referentes a negócios, sendo armazenados dados relativos a detalhes.

Ao cliente são pedidos números relativos à quantidade de negócios a definir no sistema, dos diversos tipos aceites. Para contribuir na estimativa, pede-se ainda as quantidades médias de artigos e número de locais referenciados em cada negócio, tudo com o propósito de avaliar o nível de informação a armazenar potencialmente nas tabelas referentes a *deals*.

As provas de performance ajudam o utilizador a monitorizar as estatísticas referentes ao negócio, para avaliar a sua proximidade do objectivo fixado, que lhe trará os benefícios firmados no acordo de negócio com o fornecedor. Esta funcionalidade não tem um impacto significativo no dimensionamento da base de dados, pois não gera volumes consideráveis de dados. As perguntas encontradas pretendem avaliar a percentagem de negócios que terá esta funcionalidade associada, finalizando com o número médio de termos de cada prova de performance.

4.2.3.21 Alterações de custos

Quando um artigo é definido no seu sistema, introduz-se o valor relativo ao seu custo. Todas as alterações subsequentes a este valor serão realizadas através desta funcionalidade.

As perguntas aqui encontradas pretendem saber o número de alterações previstas por mês, a média de produtos afectados e de locais onde a alteração será aplicada. Os dados aqui introduzidos são mais relevantes a nível funcional, influenciando a duração do *batch*, pois é nesta altura que todas as alterações são realizadas. Como as operações maioritárias causadas por esta funcionalidade são *updates*, não são de esperar grandes níveis de informação nas tabelas da base de dados.

4.2.3.22 Custo de desembarque estimado

O ORMS permite estimar o custo total de desembarque dos produtos, ao analisar todas as despesas associadas a um evento de transporte de artigos.

As perguntas encontradas nesta área têm um carácter mais funcional, pois o nível de informação gerado por esta funcionalidade é reduzido, sendo gerido com base em tabelas temporárias, que se esvaziam depois de completado o processo.

Assim, as respostas a recolher referem-se a números de zonas criadas para o cálculo dos custos, artigos que contêm informação relativa a custos adicionais de transferência e outros dados relativos à criação de perfis de despesas, como a média de artigos associados a estes perfis, ou locais de carregamento e descarregamento de encomendas.

4.2.3.23 Transferências de retorno em massa

Estas transferências, identificadas como *Mass Return Transfer* (MRT), ocorrem em duas situações distintas. Ambas consistem na reunião de um conjunto de artigos em diversas lojas e na sua transferência para um único armazém, sendo que depois podem ou ser redistribuídos pelas lojas, ou retornados ao vendedor.

As perguntas desta área têm a principal finalidade de obter dados relativos a esta funcionalidade do sistema, tentando obter respostas quanto ao número de MRT criadas por mês, valores médios de locais e artigos a incluir, percentagem de aprovações e período de histórico a reter.

A informação gerada por esta funcionalidade estende-se principalmente às tabelas de transacções, *shipments* e manutenção de artigos, não sendo, no entanto, de elevada quantidade, pelo que não se mostra muito importante para o exercício de dimensionamento.

4.2.3.24 Transferências

As transferências constituem uma área sensível do ORMS. Todos os meses são criadas centenas ou milhares de transferências, que consistem na movimentação de produtos de um sítio para outro, dentro da rede organizacional. Estas podem ser de diversos tipos, seja entre lojas, entre armazéns ou entre os dois, para enumerar alguns.

Desta forma, o volume de dados gerado é considerável, pelo que as respostas às perguntas desta área devem ser tratadas com cuidado. As mais importantes referem-se ao número esperado de transferências criadas manualmente por mês, à média de artigos definidos em cada uma, à percentagem de aprovações e ao período de histórico a manter.

Além das tabelas referentes a transacções, *shipments* e manutenção de artigos, as tabelas TSFHEAD, TSFDETAIL e TSF_ITEM_COST são influenciadas por estes dados, prevendo-se um número considerável de registos gerado em todas elas.

4.2.3.25 Ordens de trabalhos

As *work orders* são documentos que contêm um conjunto de tarefas a serem executadas sobre os artigos que fazem parte de uma transferência. O impacto dos dados recolhidos nas respostas às perguntas desta área é diminuído, uma vez que o volume de informação gerado é diminuto. Isto deve-se ao facto de as *work orders* não se associarem a um elevado número de transferências. Aliás, a pergunta mais importante no conjunto é a que se refere precisamente à percentagem de transferências que têm uma ordem de trabalhos associada, no sentido de prever os registos a armazenar nas tabelas TSF_WO_HEAD e TSF_WO_DETAIL.

4.2.3.26 Devoluções ao fornecedor

As devoluções aos fornecedores, denominadas de *Return To Vendor* (RTV), são um tipo de transferências que podem ser definidas singularmente, ou podem resultar da criação de MRT.

Estas operações desencadeiam registos nas tabelas de eventos de transporte de artigos, alterações nas tabelas de manutenção de produtos e novos registos nas tabelas correspondentes a esta funcionalidade, que têm o prefixo “RTV_”. Aqui as perguntas têm um carácter funcional, questionando o cliente sobre o número de RTV criadas por mês, médias de artigos e locais referidos em cada uma delas e percentagem de aprovações. A pergunta mais importante refere-se ao período de histórico a reter para esta funcionalidade, afectando directamente as tabelas RTV_HEAD e RTV_DETAIL. A informação gerada nestas operações não apresenta uma quantidade considerável de registos para o cálculo do tamanho final da base de dados.

4.2.3.27 Expedições

Esta área refere-se à expedição de artigos, sendo todos os eventos associados a esta actividade registados na tabela SHIPMENT. Esta tabela recebe um novo registo de cada vez que uma transferência ou uma ordem de compra são aprovadas. Quando os artigos chegam dos fornecedores, outro registo é inserido na tabela, para dar conta da operação de recebimento.

É precisamente sobre as operações de recebimento que recaem as perguntas desta área. Pretende saber-se qual a média de recebimentos por cada ordem de compra e por cada transferência, no sentido de prever parte do número de registos a conter na tabela. A outra parte advém das outras áreas funcionais que originam este tipo de eventos, que foram devidamente identificadas.

4.2.3.28 Ajustes de inventário

O ORMS permite ajustar os valores de *stock* dos produtos, a nível de disponíveis e indisponíveis, para todos os locais onde existem artigos armazenados.

Esta é uma funcionalidade que não gera muita informação, pois consiste na alteração de valores nas tabelas. No entanto, existem 5 tabelas associadas, que armazenam dados relativos a eventos de ajuste de inventário, às razões que os originaram, aos códigos de estado dos inventários e quantidades de *stock* nesses estados. As perguntas feitas ao retalhista sobre esta funcionalidade visam prever os registos a inserir nestas tabelas, embora estes sejam pouco significantes. É de realçar o facto de cada ajuste de inventário gerar um novo registo nas tabelas de transacções, pois os ajustes são processados durante o *batch*.

4.2.3.29 Planeamento de actividade

Esta funcionalidade é praticamente irrelevante no que toca a volume de dados gerado, uma vez que os registos são apresentados em pouco número e tamanho. As perguntas nesta área pretendem saber o número de vezes em que toda a companhia fica inactiva, o número de locais fechados em determinadas datas e a média de datas diferentes por cada local inoperante. Estes dados não afectam outras tabelas, apenas as que se referem a esta funcionalidade, nomeadamente as tabelas COMPANY_CLOSED e LOCATION_CLOSED.

4.2.3.30 Reposição

A reposição automática é uma funcionalidade que monitoriza constantemente as condições do inventário e baseado nestas condições cria ordens de compra ou transferências para atender as necessidades de procura nas lojas.

Das perguntas apresentadas, as mais importantes referem-se à percentagem de artigos a incluir na funcionalidade, à média de revisões semanais feitas aos artigos, ao número médio de locais por cada artigo a ser repostos e à percentagem de aprovações. Os valores das respostas afectam directamente as tabelas REPL_ITEM_LOC, REPL_ITEM_LOC_UPDATES,

REPL_RESULTS, REPL_RESULTS_TEMP, REPL_DAY e REPL_DAY_UPDATE, ao passo que as aprovações envolvem, além destas, as tabelas de *shipping*, transacções, transferências e ordens de compra.

Esta funcionalidade gera um nível considerável de informação, uma vez que as tabelas REPL_RESULTS e REPL_ITEM_LOC contém registos de reposições ao nível das combinações de artigos e locais, sendo que esta combinação, como visto em áreas anteriores, atinge um tamanho relativamente elevado. Assim, os dados resultantes das perguntas referidas anteriormente devem ser tratados com alguma sensibilidade.

4.2.3.31 Contagem de stock

As contagens físicas de *stocks* utilizam-se para se saber as quantidades reais existentes de cada artigo em cada local da organização. Estas podem ser agendadas pelo retalhista, podendo ser executadas em diferentes alturas, para diferentes locais.

As perguntas nesta área são importantes, uma vez que esta funcionalidade é responsável por um elevado número de registos inseridos nas tabelas. As mais críticas são as que se referem à média de locais, de *subclasses* e de artigos por local em cada contagem de *stock*. O número previsto de contagens é também pedido ao cliente. Associado aos valores mencionados anteriormente, permite prever o número de registos das tabelas STAKE_HEAD, STAKE_LOCATION, STAKE_PRODUCT, STAKE_PROD_LOC e STAKE_SKU_LOC.

Esta última tabela espera-se que armazene uma elevada quantidade de registos, uma vez que a sua chave primária é constituída pela combinação entre contagens de *stock*, artigos, tipos de locais e identificação de locais. Assim, é importante analisar os valores recolhidos nesta área, pois são significantes para o cálculo do tamanho total da base de dados.

4.2.3.32 Taxas de vendas

Esta funcionalidade do ORMS permite associar regiões geográficas a valores de taxas a aplicar nas vendas. Os dados gerados são reduzidos, baseando-se em regiões, códigos e valores de taxa a aplicar, pelo que o seu contributo para o tamanho total da base de dados é insignificante.

O questionário avalia a utilização desta funcionalidade e diversa informação básica associada, como número de regiões a definir, quer sejam países, cidades ou estados, percentagem de lojas associadas a códigos e quantidade de códigos a introduzir no sistema.

4.2.3.33 Cronogramas

O ORMS permite a definição de cronogramas, ou *timelines*, que consistem na determinação de etapas associadas a tarefas e a uma duração específica. Esta funcionalidade tem um carácter secundário e não é responsável pela geração de um número de registos que deva ser tido em atenção. É pedida ao cliente informação base, como o número de *timelines* a definir no sistema, média de etapas para cada uma e média de duração total, em dias, de cada *timeline*.

Esta informação apenas se traduz em registos nas tabelas TIMELINE, TIMELINE_HEAD, TIMELINE_STEPS e TIMELINE_STEP_COMP, não se prevendo um tamanho relevante para estas tabelas.

4.2.3.34 Documentos

No sistema é possível guardar tipos de documentos e associar-lhes pedaços de texto predefinido, representando informação padronizada e importante para os mesmos. Apenas o número previsto de documentos é pedido ao cliente, que se traduz no número de registos a esperar para a tabela DOC, sendo que não serão importantes relativamente ao volume de dados.

4.2.3.35 Etiquetagem

Esta funcionalidade está associada às etiquetas a incluir nos artigos, que podem ser criadas e geridas no sistema.

São pedidos ao cliente valores relativos a tipos diferentes de etiquetas, média de atributos definidos por cada um dos tipos e a percentagem de artigos com uma etiqueta associada. Estes valores influenciam as tabelas TICKET_TYPE_HEAD, TICKET_TYPE_DETAIL e TICKET_REQUEST. A percentagem de artigos com etiquetagem traduz-se no número de registos previsto para a tabela ITEM_TICKET. No entanto esta tabela não tem um número elevado de colunas, pelo que os seus registos não reclamam muito espaço, o que confere a esta funcionalidade um carácter pouco relevante para o tamanho total da base de dados.

4.2.3.36 Regiões

O ORMS permite definir locais, fornecedores e/ou *departments* específicos que um dado grupo de utilizadores tem responsabilidades associadas. As tabelas relativas a esta funcionalidade não estão referenciadas no sistema, sendo usadas internamente para produção de relatórios e estatísticas.

As perguntas nesta área pretendem saber o número de combinações entre *groups* e /ou locais, no sentido de prever o número de registos nas tabelas que suportam a funcionalidade, como o caso da REGIONALITY_MATRIX, REGIONALITY_HEAD ou REGIONALITY_SUP. As influências desta funcionalidade não se estendem para além destas tabelas, sendo de desprezar o seu contributo para o exercício de dimensionamento.

4.2.3.37 Utilizadores e navegação

O registo dos utilizadores acreditados para utilização do sistema é feito em tabelas criadas para o efeito, com o objectivo de controlar os acessos e as operações que cada um pode fazer no ORMS, tentando evitar a execução de tarefas não autorizadas, ou para prevenção de erros, por utilização de funcionalidades para as quais o utilizador não está habilitado.

As tabelas USER_ATTRIB, SEC_USER_GROUP, SEC_USER_ZONE_MATRIX e USER_DOMAIN serão influenciadas pela resposta à pergunta mais importante, que se refere ao número de utilizadores a definir no sistema. As 2 seguintes são de carácter mais funcional, para a fase de planeamento de capacidade, pois referem-se aos números médio e máximo de utilizadores concorrentes no sistema.

Na área de navegação, apenas se pretende avaliar a quantidade de papéis, ou *roles*, de navegação, não tendo este valor qualquer impacto de maior na base de dados.

4.2.3.38 Segurança de locais, produtos e zonas

Estas três áreas são semelhantes, pois dizem respeito a segurança, diferindo apenas no seu alvo principal. A funcionalidade relacionada com esta segurança permite associar utilizadores a grupos de segurança, com o objectivo de controlar os acessos e as operações que estes estão autorizados a efectuar.

A informação gerada por esta funcionalidade é extremamente reduzida, não se esperando que tenha um impacto significativo no sistema. As tabelas envolvidas, nomeadamente as SEC_GROUP, SEC_USER_GROUP e SEC_GROUP_ZONE_MATRIX, armazenam registos na ordem das centenas, que apresentam poucos campos, não reclamando muito espaço físico.

4.2.3.39 Configuração de pontos de venda

Os pontos de venda, ou *point of sale* (POS), são os sistemas encontrados nas lojas que registam as transacções e processam os pagamentos, devoluções e outras operações inerentes à actividade dos retalhistas. A sua configuração pode ser feita através do ORMS, que oferece a possibilidade de associar botões a tarefas, criar cupões, restrições de produtos, controlo de dinheiro de caixa, entre outros.

Estas configurações são descarregadas para os sistemas nas lojas, não tendo informação de grande volume associada, pelo que não é esperado que resultem em tabelas de tamanho a considerar no exercício de dimensionamento.

4.2.3.40 Retail Price Management

O *Retail Price Management* (RPM) é um módulo do Oracle Retail responsável pela gestão e processamento dos preços dos produtos. Este módulo, embora seja autónomo, está integrado no ORMS, fazendo parte de quase todas as implementações do sistema. Desta forma, faz sentido estender o exercício de dimensionamento às tabelas que suportam este módulo.

Informação básica

Nesta área encontram-se diversas perguntas relativas a informação base do RPM, bem como a algumas funcionalidades básicas, como o guia de preços, calendário, códigos, regras e restrições de fixação de preços e definição de áreas diferenciais, para agrupamento de áreas semelhantes, facilitando alterações em massa. As respostas a estas perguntas não têm um grande impacto no sistema, pois não são responsáveis por um elevado volume de informação gerada.

Seguidamente, aparecem umas questões mais sensíveis, relativas a estratégias de preços. Aqui pretende saber-se quantas estratégias de liquidação, margem de lucro ou competitividade serão criadas no sistema, o número de lojas concorrentes com informação guardada, e a média de vezes em que são efectuadas revisões a estas estratégias. As tabelas RPM_STRATEGY e RPM_STRATEGY_DETAIL são as mais afectadas por estes valores.

São também pedidos ao cliente valores referentes a alterações de preços dos produtos, a nível de quantidade de alterações previstas mensalmente e número médio de produtos e locais afectados por estas mudanças, influenciando as tabelas RPM_PRICE_CHANGE e RPM_PRICE_CHG_PAYLOAD. A tabela PRICE_HIST do ORMS é também afectada por estas mudanças, sendo uma tabela com um volume considerável de dados.

As informações recolhidas nesta parte do questionário relativas ao RPM resultam numa razoável quantidade de registos. No entanto, as tabelas envolvidas não apresentam muitos campos, pelo que estes são de tamanho reduzido, não se esperando que as tabelas atinjam valores elevados no que toca a espaço de armazenamento.

Liquidações

Tal como na parte de alterações de preço, aqui pede-se nomeadamente a quantidade de liquidações prevista mensalmente, o número médio de artigos e locais afectados e o período de histórico a reter. As tabelas influenciadas directamente são a RPM_CLEARANCE e a RPM_CLEARANCE_PAYLOAD. Dado este evento não ser tão frequente quanto as alterações de preços, espera-se um volume de informação gerada ainda menor do que o referido anteriormente.

Promoções

As promoções provocam alterações nos preços dos produtos, nos *stocks*, pois existem quantidades reservadas para estes eventos e desencadeiam um número mais elevado de transacções. As informações aqui recolhidas são referentes ao número previsto de promoções mensais, médias de produtos afectados, de produtos oferecidos, de locais afectados e percentagens relativas aos tipos de promoções, sejam elas simples, de compra/oferta, ou baseadas em montantes de compras.

A tabela mais significativa nesta área é a RPM_PROMO, onde se armazenam todas as promoções realizadas na organização, pelas diversas lojas e artigos. Ainda que estes eventos sejam frequentes, os registos gerados não têm tamanho significativo, sendo o seu impacto mais a nível funcional e operacional do que em termos de volume de dados.

O principal objectivo deste exercício foi assim traduzir as respostas dadas às perguntas do questionário em números previstos de registos nas tabelas, para cálculo da estimativa final de espaço físico. Findo este exercício, pode então proceder-se à utilização dos casos práticos para completar a operação de dimensionamento da base de dados.

4.2.4 Recolha de estatísticas de casos práticos

Com a identificação das tabelas principais concluída, partiu-se para a recolha de estatísticas dos casos práticos, apresentadas no capítulo anterior. O objecto principal de estudo foi o valor médio do tamanho do registo de cada tabela, para todos os casos. Com este valor recolhido em todos os casos, calculou-se a média aritmética do mesmo, atribuindo-se a cada tabela. O tamanho efectivo da tabela será estimado através da multiplicação do número previsto de registos por este valor, chegando-se ao montante, em *bytes*, de espaço físico requerido para o armazenamento da tabela em questão. No decorrer desta tarefa, verificou-se que algumas funcionalidades do sistema são raramente utilizadas pelos clientes, o que origina tabelas na base de dados completamente vazias e, conseqüentemente, sem valores estatísticos. Para estes casos, foi determinado o comprimento máximo do registo, sendo este utilizado da mesma forma que o valor médio do registo.

OWNER	TABLE_NAME	NUM_ROWS	BLOCKS	AVG_ROW_LEN	SIZE
RMSP	ADDR	15191	237	97	1473527
RMSP	ADD_TYPE	6	1	13	78
RMSP	ADD_TYPE_MODULE	16	1	11	176
RMSP	ALC_ALLOC	26	5	71	1846
RMSP	ALC_APPROVAL_QUANTITY	0	0	0	0
RMSP	ALC_AUTO_QUANTITY_LIMITS	0	0	0	0
RMSP	ALC_CODE_DETAIL	5	1	33	165
RMSP	ALC_CODE_HEAD	2	1	20	40
RMSP	ALC_COMP_LOC	0	0	0	0
RMSP	ALC_CORPORATE_RULE_DETAIL	0	0	0	0
RMSP	ALC_CORPORATE_RULE_HEAD	0	0	0	0
RMSP	ALC_FLEXIBLE_COLUMNS	201	7	51	10251
RMSP	ALC_FREIGHT_COST	0	0	0	0
RMSP	ALC_HEAD	0	0	0	0
RMSP	ALC_HEAD_TEMP	0	0	0	0
RMSP	ALC_IDEAL_WEEKS_OF_SUPPLY	0	0	0	0
RMSP	ALC_ITEM_LOC	432	12	106	45792
RMSP	ALC_ITEM_LOC_EXCLUSION	1126	20	85	95710
RMSP	ALC_ITEM_SOURCE	755	12	66	49830
RMSP	ALC_ITEM_SOURCE_SHIP_SCHEDULE	0	0	0	0
RMSP	ALC_LOCATION	171	5	38	6498
RMSP	ALC_LOC_GROUP	27	5	32	864
RMSP	ALC_LOC_GROUP_DETAIL	27	5	36	972

Figura 17 – Exemplo de estatísticas recolhidas para tabelas

OWNER	SEGMENT_NAME	SEGMENT_TYPE	TABLE_NAME	BYTES	BLOCKS	NUM_ROWS	BLEVEL	LEAF_BLOCKS
RMSP	ADDR_I1	INDEX	ADDR	524288	64	15191	1	54
RMSP	ADDR_I2	INDEX	ADDR	65536	8	94	0	1
RMSP	ADDR_I3	INDEX	ADDR	655360	80	15191	1	70
RMSP	ADDR_I4	INDEX	ADDR	524288	64	15191	1	50
RMSP	ALC_AUTO_QUANTITY_LIMITS_I1	INDEX	ALC_AUTO_QUANTITY_LIMITS	65536	8	0	0	0
RMSP	ALC_AUTO_QUANTITY_LIMITS_I2	INDEX	ALC_AUTO_QUANTITY_LIMITS	65536	8	0	0	0
RMSP	ALC_COMP_LOC_I1	INDEX	ALC_COMP_LOC	65536	8	0	0	0
RMSP	ALC_CORPORATE_RULE_DETAIL_I1	INDEX	ALC_CORPORATE_RULE_DETAIL	65536	8	0	0	0
RMSP	ALC_FLEXIBLE_COLUMNS_I1	INDEX	ALC_FLEXIBLE_COLUMNS	65536	8	201	0	1
RMSP	ALC_FLEXIBLE_COLUMNS_I2	INDEX	ALC_FLEXIBLE_COLUMNS	65536	8	201	0	1
RMSP	ALC_HEAD_I1	INDEX	ALC_HEAD	65536	8	0	0	0
RMSP	ALC_HEAD_I2	INDEX	ALC_HEAD	65536	8	0	0	0
RMSP	ALC_HEAD_I3	INDEX	ALC_HEAD	65536	8	0	0	0
RMSP	ALC_HEAD_I4	INDEX	ALC_HEAD	65536	8	0	0	0
RMSP	ALC_IDEAL_WEEKS_OF_SUPPLY_I1	INDEX	ALC_IDEAL_WEEKS_OF_SUPPLY	65536	8	0	0	0
RMSP	ALC_ITEM_LOC_EXCLUSION_I1	INDEX	ALC_ITEM_LOC_EXCLUSION	65536	8	1126	1	3
RMSP	ALC_ITEM_LOC_I1	INDEX	ALC_ITEM_LOC	65536	8	432	1	5
RMSP	ALC_ITEM_LOC_I2	INDEX	ALC_ITEM_LOC	65536	8	432	1	2
RMSP	ALC_ITEM_SOURCE_I1	INDEX	ALC_ITEM_SOURCE	65536	8	755	1	2
RMSP	ALC_LOCATION_I1	INDEX	ALC_LOCATION	65536	8	171	0	1
RMSP	ALC_LOC_GROUP_DETAIL_I1	INDEX	ALC_LOC_GROUP_DETAIL	65536	8	27	0	1
RMSP	ALC_LOC_GROUP_I1	INDEX	ALC_LOC_GROUP	65536	8	27	0	1
RMSP	ALC_LOC_GROUP_I2	INDEX	ALC_LOC_GROUP	65536	8	0	0	0
RMSP	ALC_PLAN_I1	INDEX	ALC_PLAN	65536	8	0	0	0
RMSP	ALC_PLAN_I2	INDEX	ALC_PLAN	65536	8	0	0	0
RMSP	ALC_PLAN_I3	INDEX	ALC_PLAN	65536	8	0	0	0
RMSP	ALC_PREPACK_CALC_RESULTS_I1	INDEX	ALC_PREPACK_CALC_RESULTS	65536	8	0	0	0

Figura 18 – Exemplo de estatísticas recolhidas para os índices

No caso dos índices, determinar as suas necessidades em termos de espaço físico é um processo um pouco mais complexo. De facto, com a normal utilização do sistema, e as constantes operações sobre as tabelas e registos da base de dados, os índices também vão sendo modificados. Quando um registo é alterado e passa a ocupar mais espaço e deixa de caber no bloco onde reside, é movido para outro bloco, onde haja espaço livre. Este fenómeno origina um crescimento do tamanho total de um índice, com uma taxa de aproveitamento dos blocos cada vez mais diminuída.

Para estimar este crescimento, recorreu-se a um método de cálculo do espaço teórico a ocupar pelo índice, baseado na estrutura *B-tree*, usando-se o valor de nós-folha que consta das estatísticas recolhidas nos casos práticos. O diagrama seguinte ilustra o algoritmo em que esta técnica assenta.

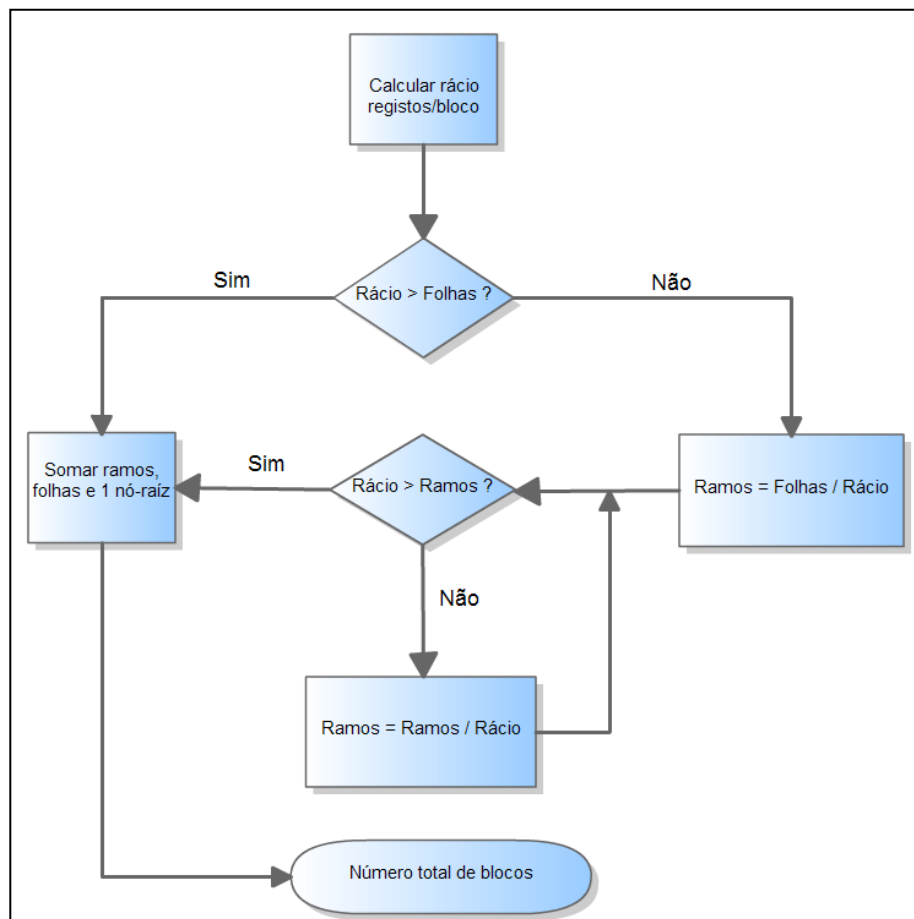


Figura 19 – Diagrama de fluxo representativo do algoritmo

Inicialmente, determina-se o rácio de registos por cada bloco, por divisão do valor total de registos pelo valor total de folhas. Se este valor for superior ao número de folhas, significa que existe um nível de ramos acima, que tem que ser endereçado. Nesse caso, divide-se o número de folhas pelo rácio calculado, registando-se o resultado como o número de ramos do nível acima.

Caso este número de ramos seja superior ao valor de rácio calculado, significa que existe outro nível de ramos, que se calcula dividindo o valor actual de ramos pelo rácio. Esta operação executa-se ciclicamente, até que o valor de ramos calculado numa iteração seja inferior ao rácio.

Finalmente, somam-se todas as folhas e ramos calculados e acrescenta-se um nó, como sendo a raiz da árvore [DONB03]. Pela comparação do resultado obtido, com o valor de blocos efectivamente ocupado pelo índice, chega-se a um valor percentual de diferença entre o valor teórico e o valor prático.

Para estimar o espaço a ocupar pelos índices, segue-se uma lógica idêntica. Inicialmente divide-se o tamanho padrão dos blocos pelo tamanho do registo a indexar, resultando no rácio de registos por bloco. Em seguida, determina-se o número de folhas da árvore, ao dividir o número previsto de registos, idêntico ao das tabelas a que estão associados, pelo rácio encontrado anteriormente. O resultado traduz-se no número de folhas a serem preenchidas, aplicando-se então o mesmo método para calcular o número total de blocos ocupados, adicionando-se no final a percentagem de crescimento determinada anteriormente. A avaliação do espaço efectivamente ocupado pelo índice, em *bytes*, consegue-se com a multiplicação do número de blocos estimado pelo tamanho padrão dos blocos, definido pelo administrador da

base de dados no acto de instalação do sistema. A título informativo, verificou-se que este valor é igual a 8192 *bytes* em todos os casos analisados.

4.3 Caracterização e apresentação da ferramenta desenvolvida

A ferramenta desenvolvida consiste numa folha de cálculo, que engloba todas as metodologias seguidas em cada etapa do desenvolvimento da solução, apresentando-se como o culminar de todo este processo, após todo o trabalho de pesquisa e identificação dos factores mais influentes ter sido efectuado. Contém as perguntas do questionário com os campos para introdução dos valores de cada resposta. Os dados inseridos nestes campos são utilizados em fórmulas de cálculo desenvolvidas para estimar os registos de cada tabela, tendo em conta as combinações de registos referidas anteriormente, em que a chave primária de cada tabela dita os valores a incluir no cálculo da estimativa.

Depois de apurados estes dados, procede-se à estimativa do tamanho físico a ocupar por cada uma das tabelas, por multiplicação destes valores pelas médias determinadas na análise das estatísticas dos casos práticos, resultando no valor previsto de *bytes* de armazenamento a serem necessários para armazenar a tabela. A soma de todos estes valores dita o espaço total requerido para albergar a informação prevista na base de dados.

Área	Observações	Tabela	Valores			
			Registos	Tam máx registo	Tam médio registo	Tamanho
Org Hierarchy		CHAIN	2		23,8	47,6
		AREA	100		25,4	2540
		REGION	200		25,8	5160
		DISTRICT	500		31,8	15900
		WH	50		100,2	5010
	WH * DEPS	WH_DEPT	15000	202		3030000
	WH * DEPS	WH_DEPT_EXPL	15000	259		3885000
		WH_STORE_ASSIGN	8000	52		416000
		WH_MFQUEUE			682,5	0
		WH_PUB_INFO	50		11	550
		STORE	8000		162,2	1297600
	STORE * DEPS	STORE_DEPT_AREA	2400000		19	45600000
		WALK_THROUGH_STORE	8000	44		352000
		STORE_FORMAT			15,25	0
		STORE_PUB_INFO	8000		8,75	70000
		STORE_HIERARCHY	8000		22,8	182400
		STORE_ORDERS		83		0
		STORE_ATTRIBUTES	6400		19,25	123200
		WH_ATTRIBUTES	40		15,5	620

Figura 20 – Excerto da ferramenta desenvolvida, cálculo das tabelas

Para os índices, os números de registos previstos são também utilizados para o cálculo do tamanho a ocupar, através de fórmulas de cálculo baseadas no método descrito. A soma de todos os valores dita as necessidades de armazenamento físico para os índices. Como se pode notar pela imagem seguinte, alguns índices não apresentam o valor de percentagem de crescimento. Isto deve-se ao facto de não terem sido encontrados nos casos práticos valores que permitissem efectuar este cálculo. Outra particularidade deste exercício foi o facto de os índices que não apresentam qualquer registo, ocuparem sempre 8 blocos, o que foi incluído na estimativa.

Área	Tabela	Índice	Registos	Blocos teóricos	% Crescimento	Blocos reais
Merch Hierarchy	DIVISION	DIVISION_I1	0	0		8
		DIVISION_I2	0	0		8
		PK_DIVISION	16	2	400	8
	GROUPS	GROUPS_I1	0	0		8
		GROUPS_I2	0	0		8
		GROUPS_I3	33	2	400	8
	DEPS	PK_GROUPS	33	2	400	8
		DEPS_I1	0	0		8
		DEPS_I2	0	0		8
	DEPT_SALES_FORECAST	DEPS_I3	33	2	400	8
		PK_DEPS	33	2	400	8
		DEPT_SALES_FORECAST_I1	0	0		8
	DEPT_SALES_HIST	DEPT_SALES_HIST_I1	37376	128	200	256
		PK_DEPT_SALES_HIST	37376	187	137	256
	CLASS	CLASS	3488	13	123	16
		PK_CLASS				
		CLASS_SALES_FORECAST_I1	0	0		8
	SUBCLASS	CLASS_SALES_HIST_I1	643086	4195	101	4224
		PK_CLASS_SALES_HIST	643086	3773	102	3840
		PK_SUBCLASS	6193	27	119	32
	SUBCLASS_SALES_FORECAST	SUBCLASS_SALES_FORECAST_I1	0	0		8
		PK_SUBCLASS_SALES_HIST	850722	5445	103	5632
		SUBCLASS_SALES_HIST_I1	301571	5336	98	5248
	MERCH_HIER_DEFAULT	MERCH_HIER_DEFAULT_I1	0	0		8
		MERCH_HIER_DEFAULT_I2	0	0		8
		MERCH_HIER_DEFAULT_I3	5660	29	110	32
	PEND_MERCH_HIER	PK_MERCH_HIER_DEFAULT	5660	11	145	16
		PEND_MERCH_HIER_I1	0	0		8
		PEND_MERCH_HIER_I2	0	0		8
	DYNAMIC_HIER_CODE	PEND_MERCH_HIER_I3	0	0		8
		UK_PEND_MERCH_HIER	0	0		8
		PK_DYNAMIC_HIER_CODE	26	2	400	8
	BUYER	MERCHANT	3	2	400	8
		PK_BUYER	35	2	400	8

Figura 21 – Excerto da ferramenta desenvolvida, parte dos índices

Embora todo o trabalho de pesquisa de fundamentos teóricos sobre o *batch* nocturno e as necessidades de memória tenha sido efectuado, bem como todo o estudo da sua mecânica e utilização pelo sistema, a ferramenta não ficou dotada da funcionalidade de cálculo destas necessidades, resumindo-se a sua utilidade ao cálculo de requisitos de armazenamento.

4.4 Resumo e conclusões

Neste capítulo apresenta-se a solução encontrada para o problema em questão, detalhando-se todas as etapas em que consistiu a sua formulação e elaboração.

Pela leitura do capítulo pode concluir-se que a técnica de dimensionamento utilizada, de recolha de estatísticas, foi a mais adequada à situação, uma vez que existem já casos práticos de sistemas implementados e em pleno ambiente de produção, fornecendo dados realísticos e de confiança.

A etapa mais crucial do desenvolvimento da solução foi então a criação da associação entre as perguntas do questionário, que avaliam as necessidades de negócio dos clientes, e as tabelas da base de dados, resultando numa tradução das respostas recolhidas em volumes de dados a prever na base de dados.

A ferramenta apresentada é então um culminar de todo este trabalho, onde se procede a um cálculo automático da estimativa do tamanho da base de dados, com base no mapeamento efectuado anteriormente e atentando em todas as combinações de registos, restrições e influências menos directas entre áreas funcionais identificadas nesse mesmo exercício.

5 Conclusões e Trabalho Futuro

Neste capítulo final, são apresentadas as conclusões do projecto. É feita uma retrospectiva do mesmo, realçando-se os aspectos mais essenciais que marcaram todo o processo de construção da solução. Em seguida, analisam-se os objectivos propostos inicialmente e confronta-se com a solução desenvolvida, enumerando as metas atingidas e as que não o foram. Finalmente, abre-se uma perspectiva de algum trabalho que poderá ser realizado no futuro, tendo em conta um melhoramento do projecto.

5.1 Retrospectiva

A área de negócio em que este projecto se inseriu, a venda a retalho, representa um papel muito importante nas sociedades actuais, uma vez que as pessoas têm todos os dias necessidades materiais que conseguem colmatar com os produtos que os retalhistas têm disponíveis. Num só lugar, é possível encontrar diversos artigos, o que se torna extremamente cómodo para o consumidor médio, que assim vê a sua vida facilitada, pois não terá que deslocar-se para mais sítios, poupando tempo e dinheiro. Este é um negócio que envolve quantidades elevadas de informação, e com o crescimento que tem sofrido ao longo dos tempos, esta informação ultrapassou os limites aceitáveis a uma gestão puramente humana, baseada em livros e pastas.

Desta forma, a implementação de soluções tecnológicas, ligadas à área dos Sistemas de Informação, revelou-se indispensável para a sobrevivência dos retalhistas, ao mesmo tempo que lhes proporcionou uma oportunidade para melhorar o seu serviço aos clientes, reduzir custos e otimizar processos internos, tudo isto resultando em mais clientes, redução de custos e aumento de lucros. Dos sistemas criados para este tipo de actividade, destaca-se o Oracle Retail, referenciado como o escolhido pelos maiores retalhistas mundiais [ORAC09], cuja implementação é a especialidade da Wipro Retail, empresa onde este projecto foi realizado.

Assim, o projecto centrou-se no módulo mais central do Oracle Retail, o Oracle Retail Merchandising System, que é o cerne de toda a suite, onde se concentram as funcionalidades mais importantes e a informação mais crítica de todo o negócio. O planeamento duma implementação deste sistema contém uma etapa destinada ao dimensionamento da base de dados e ao planeamento de capacidade, que constituem um exercício sensível de ajuda à decisão sobre o equipamento de *hardware* a adquirir, resultando normalmente em investimentos avultados por parte dos clientes, dado que este sistema requer equipamentos de elevada capacidade, por negociar com volumes de dados acima dos níveis vulgares. Sendo este processo da responsabilidade do fabricante, neste caso a Oracle, nasceu a vontade de estudá-lo e percebê-

lo, efectuando uma aproximação à realização deste exercício de uma forma autónoma para poder confrontar os resultados obtidos pelo fabricante e pela Wipro.

A solução foi conseguida através de um estudo de alguns métodos de dimensionamento existentes, tendo sido feita uma adaptação ao problema em causa. Foram recolhidos dados estatísticos de casos práticos de sistemas implementados e em fase de produção, que permitiram dotar a ferramenta produzida da capacidade de estimar volumes de dados de uma base de dados, baseando-se também nas respostas a um questionário, dadas pelo cliente, onde este expõe as suas necessidades de negócio.

A nível pessoal, a experiência foi enriquecedora. Começando pela integração numa empresa de tamanha envergadura, passando pelo contacto com diferentes pessoas e métodos. A nível de conhecimento adquirido, foi importante analisar situações práticas e reais, de sistemas que criam valor para as organizações e, consequentemente, para a sociedade em geral. O contacto com bases de dados da dimensão das encontradas constituiu um salto dos conhecimentos teóricos e trabalhos académicos para o que realmente acontece na área das Tecnologias de Informação.

5.2 Satisfação de objectivos

Olhando aos objectivos definidos para este projecto e ao trabalho conseguido, o saldo é positivo. As tarefas principais foram cumpridas, desde a análise do questionário da Oracle, passando pela criação de uma ponte entre as perguntas e as tabelas envolvidas do modelo de dados, o estudo dos processos de *batch* e a elaboração de uma ferramenta de dimensionamento.

No entanto, a solução desenvolvida apresenta algumas lacunas. De facto, as suas funcionalidades não compreendem o planeamento de capacidade de processamento. Embora todo o estudo teórico referente a esta questão tenha sido efectuado, não foi possível dotar a ferramenta desta capacidade, à altura de conclusão deste documento.

5.3 Trabalho Futuro

Considerando a lacuna na ferramenta identificada anteriormente, uma das oportunidades para melhorar o produto deste projecto passa por dotar a ferramenta desta funcionalidade para estimar requisitos de capacidade de processamento, baseada nas estimativas de volume de dados e no estudo efectuado sobre os processos de *batch*.

Posteriormente, o exercício de mapeamento entre as perguntas do questionário e as tabelas do modelo de dados poderá ser aprimorado, recorrendo a ferramentas de auditoria de bases de dados e, mediante a operação do sistema ORMS, registar todas as alterações efectuadas às tabelas da base de dados, permitindo perceber com maior rigor os registos e tabelas envolvidos em cada tarefa realizada no sistema.

Anexo A: Modelo de dados do ORMS

Neste capítulo anexo apresenta-se as tabelas mais importantes da base de dados do ORMS, acompanhadas de uma breve descrição. Para uma melhor compreensão, foi feita uma divisão em 9 áreas funcionais.

A.1 Dados fundamentais

Tabela	Descrição
COMPHEAD	Contém um registo, com o nome da companhia.
CHAIN	Contém uma linha por cada <i>chain</i> e/ou formato de loja.
AREA	Contém uma linha por cada <i>area</i> pertencente a uma <i>chain</i> .
REGION	Contém uma linha por cada <i>region</i> definida no sistema.
DISTRICT	Contém um registo por cada <i>district</i> .
CHANNELS	Contém os diversos canais da organização, sendo apenas usada num ambiente multi-canal.
STORE	Contém os dados relativos às lojas.
WH	Esta tabela contém dados relativos aos armazéns.
STORE_DEPT_AREA	Contém as <i>areas</i> de cada <i>department</i> associado a uma loja.
STORE_ATTRIBUTES	Contém atributos adicionais relativos a lojas.
WH_ATTRIBUTES	Contém atributos adicionais relativos a armazéns.
WH_STORE_ASSIGN	Cada registo determina uma associação entre um armazém e uma loja, numa dada data.
LOCATION_CLOSED	Aqui registam-se períodos em que certas localizações estão indisponíveis para determinadas funções.
COMPANY_CLOSED	Contém registos de períodos de inactividade de toda a

	organização.
COMPANY_CLOSED_EXCEP	Contém registos de locais que operam em períodos de inactividade de toda a organização.
SOURCE_DLVRY_SCHED	Mantém registos de lojas e locais.
SOURCE_DLVRY_SCHED_DAYS	Mantém calendarização de lojas e locais.
SOURCE_DLVRY_SCHED_EXC	Contém excepções de artigos para fornecedores e armazéns.
STOP_SHIP	Contém datas onde determinada loja não recebe determinado tipo de artigo.
STORE_HIERARCHY	Contém um registo por cada loja, mais a sua organização hierárquica.
WH_DEPT_EXPL	Contém informação relativa a <i>investment buys</i> ao nível dos armazéns e departamentos.
WH_DEPT	Contém dados associados a armazéns ou a uma combinação armazém/departamento
BANNER	Contém um registo por cada <i>banner</i> , que é o nome de um ou mais canais.
COMPHEAD	Contém um registo, com o nome da companhia.
DIVISION	Contém uma linha por cada <i>division</i> existente.
GROUPS	Contém um registo por cada <i>group</i> existente.
DEPS	Contém todos os <i>departments</i> do sistema.
CLASS	Contém uma linha por cada combinação <i>department/class</i> .
SUBCLASS	Contém um registo por cada combinação <i>department/class/subclass</i> .
BUYER	Contém todos os responsáveis de compras definidos no sistema.
MERCHANT	Contém os responsáveis pelas vendas definidos no sistema.
STORE_DEPT_AREA	Contém as <i>areas</i> de cada <i>department</i> associado a uma loja.
DYNAMIC_HIER_CODE	Contém um registo por cada referência às hierarquias organizacional ou mercadológica.
MERCH_HIER_DEFAULT	Contém registos sobre informação por defeito relativa à hierarquia mercadológica.
SUPS	Contém uma linha por cada fornecedor/localização de fornecedor.
SUP_ATTRIBUTES	Contém atributos adicionais dos fornecedores.
FREIGHT_TERMS	Nesta tabela guarda-se os termos de transporte aceites pela companhia. Esta tabela é mantida pelo administrador da base de dados.
TERMS	Nesta tabela guarda-se os termos dos fornecedores.

	Esta tabela é mantida pelo administrador da base de dados.
ITEM_SUPPLIER	Contém um registo por cada combinação artigo/fornecedor
ITEM_SUPP_COUNTRY	Contém um registo por cada país de origem de uma combinação artigo/fornecedor
SUPP_PREISSUE	Contém informação para gerar números prévios de ordens para os fornecedores
ORD_PREISSUE	Contém números de ordens pré-gerados.
SUP_DATA	Contém todas as transacções feitas com os fornecedores para o mês corrente, sendo usada para preencher a tabela SUP_MONTH
SUP_MONTH	Esta tabela contém um histórico por cada mês das transacções efectuadas com os fornecedores.
PARTNER	Contém os parceiros da companhia, como bancos ou agentes.
CURRENCIES	Contém as diversas unidades monetárias definidas no sistema.
CURRENCIES_RATES	Contém as taxas de câmbio de diversos pares de unidades monetárias
VAT_CODES	Contém um registo por cada código de IVA, que determina a taxa a aplicar a cada artigo
VAT_CODE_RATES	Contém as taxas associadas a cada código num determinado período.
VAT_REGION	Contém regiões referentes a uma dada taxa de imposto aplicável
VAT_DEPS	Contém um registo por cada combinação entre um <i>department</i> e uma região de imposto

A.2 Manutenção de artigos

Tabela	Descrição
ITEM_MASTER	Esta tabela contém toda a informação sobre cada artigo disponível no sistema, incluindo <i>packs</i>
ITEM_ATTRIBUTES	Contém atributos adicionais para os artigos
ITEM_IMAGE	Contém cópias de imagens e documentação associada a cada artigo
ITEM_TICKET	Contém um registo para cada combinação artigo/tipo de etiqueta

ITEM_XFORM_HEAD	Contém dados de cabeçalho para itens transformados.
ITEM_XFORM_DETAIL	Contém detalhes associados aos itens transformados que constam da tabela anterior.
ITEM_LOC	Contém um registo por cada item em <i>stock</i> em cada local
ITEM_LOC_SOH	Contém informação sobre <i>stock on hand</i> para cada item em <i>stock</i> em cada local
ITEM_LOC_HIST	Contém um histórico baseado numa combinação artigo/local/semana/tipo de venda
ITEM_PARENT_LOC_HIST	Esta tabela guarda histórico de vendas para itens “pai”, que estão no topo da hierarquia.
ITEM_SEASONS	Contém todas as épocas ou fases para cada item.
POS_MODS	Contém alterações aos dados que constam dos sistemas nos pontos de venda. Estes sistemas são actualizados com dados desta tabela.
DEPT_SALES_HIST	Contém histórico semanal de vendas por <i>department</i>
CLASS_SALES_HIST	Contém histórico semanal de vendas por <i>class</i>
SUBCLASS_SALES_HIST	Contém histórico semanal de vendas por <i>subclass</i>
ITEM_SUPPLIER	Contém um registo por cada combinação artigo/fornecedor
ITEM_SUPP_COUNTRY	Contém um registo por cada país de origem de uma combinação artigo/fornecedor
ITEM_SUPP_COUNTRY_LOC	Contém um registo por cada local associado a uma combinação artigo/fornecedor/país de origem
ITEM_SUPP_COUNTRY_DIM	Contém as dimensões de cada combinação artigo/país de origem/fornecedor
DIFF_TYPE	Contém os diversos tipos de <i>diffs</i> definidos no sistema.
DIFF_IDS	Contém os tamanhos, cores, cheiros, sabores, entre outros, com o código NRF[5] associado
DIFF_GROUP_HEAD	Contém uma linha por cada grupo diferenciador. Um grupo destes contém <i>diffs</i> usados em conjunto, como, por exemplo, todos os tamanhos de camisas de homem.
DIFF_GROUP_DETAIL	Contém os <i>diffs</i> que fazem parte de um dado grupo.
DIFF_RANGE_HEAD	Contém gamas diferenciadoras, que agregam diversos <i>diffs</i> . Funcionam como subconjuntos dos grupos diferenciadores.

DIFF_RANGE_DETAIL	Contém os <i>diffs</i> que fazem parte de uma dada gama.
ITEM_DIFF_LOC_HIST	Contém um histórico de vendas para todos os artigos marcados com <i>diffs</i> .
PACKITEM	Esta tabela contém uma linha por cada pacote, ou <i>pack</i> , criado no sistema, guardando-se informação base sobre cada artigo que consitui o <i>pack</i> .
PACKITEM_BREAKOUT	Esta tabela facilita o processamento dos <i>packs</i> , ao guardar informação relativa a componentes e quantidades ao nível dos artigos.
SIMPLE_PACK_TEMP	Esta é uma tabela temporária que armazena informação durante a criação de <i>packs</i> simples, que contém apenas um tipo de artigo.
PACK_TMPL_HEAD	Contém informação relativa a modelos de <i>packs</i> definidos no sistema. Estes modelos facilitam a criação de <i>packs</i> no sistema.
PACK_TMPL_DETAIL	Contém detalhes sobre os modelos definidos na tabela anterior.
RECLASS_HEAD	Contém um registo por cada reclassificação de artigos definida no sistema.
RECLASS_ITEM	Contém um registo por cada artigo a ser reclassificado, numa dada operação de reclassificação.
UDA	Contém um registo para cada atributo definido pelo utilizador (UDA).
UDA_VALUES	Contém os valores associados a um UDA
UDA_ITEM_LOV	Contém uma linha para cada combinação artigo/atributo para UDA do tipo lista de valores
UDA_ITEM_DATE	Contém uma linha para cada combinação artigo/atributo para UDA do tipo data
UDA_ITEM_FF	Contém uma linha para cada combinação artigo/atributo para UDA do tipo texto livre
SKULIST_HEAD	Contém informação base sobre listas de artigos
SKULIST_DETAIL	Contém um registo por cada artigo que conste de uma dada lista
SKULIST_DEPT	Contém todos os <i>departments</i> nos quais a lista se baseia
LOC_LIST_HEAD	Contém informação base sobre listas de localizações, quer sejam lojas ou armazéns
LOC_LIST_DETAIL	Contém uma linha por cada localização que conste de uma dada lista

A.3 Manutenção de custos

Tabela	Descrição
COST_SUSP_SUP_HEAD	Contém um registo por cada evento de custos criado no sistema.
COST_SUSP_SUP_DETAIL	Contém a combinação artigo/fornecedor associada a um evento de custos
COST_SUSP_SUP_DETAIL_LOC	Contém os locais aos quais se aplica uma alteração de custos ao nível da combinação artigo/fornecedor/país
COST_CHG_REASON	Contém as razões válidas para efectuar uma alteração de custos
COST_CHANGE_TEMP	Esta é uma tabela temporária, alimentando a COST_SUSP_SUP_DETAIL
COST_CHANGE_LOC_TEMP	Esta é uma tabela temporária, alimentando a COST_SUSP_SUP_DETAIL_LOC

A.4 Manutenção de negócios

Tabela	Descrição
FIXED_DEAL	Contém informação básica sobre negócios fixos. Um negócio fixo resulta no pagamento de um valor do fornecedor ao retalhista, por este promover os seus produtos.
FIXED_DEAL_MERCH	Aqui encontram-se os níveis mercadológicos associados a um negócio fixo.
FIXED_DEAL_MERCH_LOC	Contém as percentagens de contribuição de cada local para o negócio
DEAL_HEAD	Contém informação básica sobre negócios, como datas e fornecedor
DEAL_DETAIL	Contém detalhes de cada negócio
DEAL_THRESHOLD	Contém os limiares de cada negócio, quer sejam quantidades ou montantes comprados ou vendidos, a partir dos quais os benefícios são concedidos
DEAL_COMP_TYPE	Aqui guardam-se componentes dos negócios definidos pelo utilizador
DEAL_ITEMLOC	Contém os artigos que fazem parte de um dado negócio
DEAL_QUEUE	Esta tabela guarda todos os negócios criados no sistema, durante um dia

DEAL_CALC_QUEUE	Aqui encontram-se as ordens de compra associadas a negócios
DEAL_SKU_TEMP	Esta tabela temporária tem dados relativos a itens, fornecedores, países de origem e locais relacionados com os negócios da DEAL_QUEUE.
FUTURE_COST	Aqui guardam-se todos os itens afectados por negócios e os seus vários custos
ORDLOC_DISCOUNT	Nesta tabela estão todas as componentes dos negócios que afectam artigos que constam de ordens de compra
ORDHEAD_DISCOUNT	Contém somas de descontos totais de negócios e suas componentes, numa ordem de compra
ORDLOC	Informação relativa a itens e locais. Os descontos serão guardados aqui também.

A.5 Ordens de Compra

Tabela	Descrição
ORDHEAD	Contém um registo para cada ordem de compra criada no sistema
ORDCUST	Armazena ordens de compra criadas para clientes
ORDSKU	Nesta tabela guardam-se todos os artigos que constam de cada ordem de compra
ORDLOC	Aqui guardam-se dados relativos à combinação ordem/artigo/loja ou armazém
ORDLOC_EXP	Esta tabela contém as despesas associadas a cada item numa ordem e num local
SHIPMENT	Todos os dados relativos a expedições são guardados nesta tabela
SHIPSKU	Contém um registo por cada artigo que faz parte de uma expedição.
REV_ORDERS	Armazena dados relativos a modificações feitas a ordens de compra
CONTRACT_HEADER	É nesta tabela que estão todos os contratos feitos pelo retalhista com os seus fornecedores

A.6 Gestão de inventário

Tabela	Descrição
TSFHEAD	Contém uma linha por cada transferência efectuada
TSFDETAIL	Aqui armazenam-se outros detalhes referentes às transferências
TSF_ENTITY	Nesta tabela estão todos os detalhes sobre as entidades que intervêm nas transferências

RTV_HEAD	Contém um registo por cada transferência de retorno ao vendedor
RTV_DETAIL	É nesta tabela que se armazenam detalhes adicionais de transferências de retorno ao vendedor
INV_ADJ	Esta tabela contém dados relativos aos ajustes de inventário
STAKE_HEAD	Contém informação base sobre contagens de <i>stocks</i>
STAKE_LOCATION	Esta tabela apresenta dados referentes aos locais onde decorrem eventos de contagem de <i>stock</i>
STAKE_PRODUCT	Contém informação relativa a <i>departments</i> , sendo apenas usada para contagens de <i>stock</i> físico
STAKE_SKU_LOC	Nesta tabela guarda-se uma linha por cada artigo em cada local que tenha participado de uma contagem de <i>stock</i>
STAKE_SCHEDULE	Aqui guardam-se todos os horários definidos para contagens de <i>stock</i> nos mais diversos locais

A.7 Reposição

Tabela	Descrição
REPL_DAY	Indica os dias da semana em que as quantidades de um artigo devem ser revistas
REPL_ITEM_LOC	Esta tabela é utilizada para armazenar atributos de reposição dos artigos a nível dos locais
STORE_ORDERS	Nesta tabela estão as ordens de compra feitas pelas lojas para reposição
REPL_RESULTS	Aqui estão os resultados do cálculo da reposição, indicando-se as quantidades a enviar de cada artigo para cada local
SUPS_MIN_FAIL	Contém dados sobre ordens de compra que não foram executadas por não cumprirem quantidades mínimas de entrega
SUB_ITEMS_HEAD	Contém informação relativa a produtos substitutos
SUB_ITEMS_DETAIL	Esta tabela tem cada artigo que pode substituir um outro artigo

A.8 Stock Ledger

Tabela	Descrição
TRAN_DATA	Esta tabela armazena toda a informação gerada por cada transacção registada no sistema. É depois utilizada para construir a TRAN DATA HISTORY

TRAN_DATA_CODES	Aqui definem-se todos os códigos que distinguem os diversos tipos de transacções
DAILY_DATA	Esta tabela contém um histórico diário de todas as transacções efectuadas em cada local
WEEK_DATA	Esta tabela tem um histórico semanal de transacções, obtido através da tabela DAILY_DATA
MONTH_DATA	Os registos desta tabela representam um histórico mensal de transacções, obtido pela selecção de dados da DAILY_DATA
HALF_DATA	Oferece um histórico semestral de transacções.
HALF	Contém uma linha por cada semestre definido no sistema
HALF_DATA_BUDGET	Contém dados sazonais para efeitos de previsão de margens brutas
MONTH_DATA_BUDGET	Contém dados mensais que permitem a previsão de orçamentos
TRAN_DATA_HISTORY	Esta tabela contém um registo por cada transacção efectuada na organização

A.9 Variáveis de sistema

Tabela	Descrição
SYSTEM_OPTIONS	Esta é uma tabela de controlo que contém todos os parâmetros definidos pelo utilizador
UNIT_OPTIONS	Tal como a anterior, aqui também se encontram alguns parâmetros definidos pelo utilizador
SYSTEM_VARIABLES	Nesta tabela de controlo, armazenam-se as colunas que são regularmente actualizadas
PERIOD	Esta tabela é utilizada para armazenar datas do sistema para todos os processos e relatórios

Anexo B: Questionário da Oracle de dimensionamento

Neste capítulo anexo apresenta-se o questionário que visa perceber as necessidades de negócio de um retalhista. As perguntas encontram-se em Inglês.

Area	Question
Org Hierarchy	How many chains will be defined in RMS?
	How many areas will be defined in RMS?
	How many regions will be defined in RMS?
	How many districts will be defined in RMS?
	How many physical warehouses will be defined in RMS?
	How many stores will be defined in RMS?
	What percentage of stores will attributes defined?
	What percentage of warehouses will have attributes defined?
Multi-Channel	Will multi-channel functionality be used in RMS?
	How many banners will be defined in RMS?
	How many channels will be defined in RMS?
	What is the average number of virtual warehouses per physical warehouse?
Store Grade	Will store grade functionality be used in RMS?
	How many store grade groups will be defined in RMS?
	How many store grades will be defined in RMS?
	What is the average number of stores per store grade?
Loc Traits	Will location traits functionality be used in RMS?
	How many location traits will be defined?
	How many stores will have location traits defined?
	For stores with location traits defined, what is the average number of location traits per store?
Loc List	Will location list functionality be used in RMS?
	How many location lists will be defined?
	What is the average number of locations per location list?

Merch Hierarchy	How many divisions will be defined in RMS? (department)
	How many groups will be defined in RMS? (sub-department)
	How many departments will be defined in RMS? (category)
	How many classes will be defined in RMS? (sub-category)
	How many subclasses will be defined in RMS? (segment)
	Will merchandise hierarchy default functionality be used in RMS?
	What percentage of subclasses will have merchandise hierarchy default information defined?
	For subclasses with merch hier default info defined, what is the average number of defaults/required values per subclass?
	Will dynamic hierarchy codes be defined for merchandise and org hierarchy levels?
Supplier	How many suppliers will be defined in RMS?
	What percentage of suppliers will have attributes assigned, excluding import attributes?
	What percentage of suppliers will have import attributes assigned?
Supplier Traits	
	Will supplier traits functionality be used in RMS?
	How many supplier traits will be defined?
	What percentage of suppliers will have supplier traits assigned?
Partner	How many partners will be defined in RMS?
	How many partners will be stockholding locations (i.e., external finishers)?
Item	What is the name for level 1 items?
	What is the name for level 2 items?
	What is the name for level 3 items?
	How many level 1 items will be defined in RMS? (excluding packs)
	How many level 2 items will be defined in RMS? (excluding ref items for packs)
	How many level 3 items will be defined in RMS?
	What percentage of level 1 items are transaction level items? (excluding packs)
	What percentage of level 2 items are transaction level items?
	What percentage of level 2 items are reference items (below transaction level)?
Pack	What percentage of level 3 items are transaction level items?
	How many packs will be defined in RMS?
	What percentage of packs are simple packs?
	What is the average number of component items per complex (non-simple) pack?
Item Loc	What percentage of all packs have reference items (e.g., UPCs)?
	What percentage of complex packs are sellable only (not orderable) packs?
Item Loc	What is the average number of tran level items per store including pack items?
	What is the average number of tran level items per warehouse (or virtual wh, if multi-channel) including pack items?

Item Loc	What is the average number of departments with inventory per store?
	What is the average number of classes with inventory per store?
	What is the average number of subclasses with inventory per store?
Item Supplier	What is the average number of suppliers per item?
	What is the average number of countries per supplier?
Item List	Will item list functionality be used in RMS?
	How many items lists will be defined in RMS?
	What is the average number of items per item list?
	What is the average number of criteria per item list?
Item Attributes	What percentage of items will have attributes assigned, excluding import attributes?
	What percentage of items will have import attributes assigned?
Differentiators	Will differentiators be used in RMS?
	How many differentiator types will be defined in RMS?
	How many differentiators will be defined in RMS?
	How many differentiator groups will be defined in RMS?
	What is the average number of differentiators per differentiator group?
	How many differentiator ranges will be defined in RMS?
	What percentage of differentiator ranges will consist of two differentiator groups?
	What percentage of differentiator ranges will consist of three differentiator groups?
	How many differentiator ratios will be defined for ordering?
	What percentage of differentiator ratios will consist of two differentiator groups?
	What percentage of differentiator ratios will consist of three differentiator groups?
UDA	Will User Defined Attribute (UDA) functionality be used in RMS?
	How many user-defined attributes (UDAs) will be defined in RMS with a display type of date?
	How many UDAs will be defined in RMS with a display type of free form?
	How many UDAs will be defined in RMS with a display type of list of values (LOV)?
	For UDAs with a display type of LOV, what is the average number of values per UDA?
	What percentage of items will have required date UDAs?
	For items with date UDAs, what is the average number of UDAs per item?
	What percentage of items will have required free form UDAs?
	For items with free form UDAs, what is the average number of free form UDAs per item?
	What percentage of items will have required LOV UDAs?
	For items with LOV UDAs, what is the average number of LOV UDAs per item?
	What percentage of UDAs will have default values?
	For UDAs with default values, what is the average number of defaults (e.g., dept/class/subclass) per UDA?
Transactions	What percentage of all stores will hold inventory and have transactions during the sizing time horizon (i.e., open stores)?
	What percentage of stores are open 7 days per week?
	For stores that are open fewer than 7 days, what is the average number of open days per week?
	How many times will POS data will be loaded into RMS for each store per day? (min 1)
	What is the average number of unique tran-level items with sales per store per day?
	What is the average number of unique tran-level items with sales per store per week?
	What is the average number of unique tran-level items with returns by customers per store per day?
	What percentage sales will have discounts (promotion, emp discount, but not clearance)?
	What is the average number of unique subclasses with sales or returns per store per day?

	What is the average number of unique subclasses with sales per store per week?
	What is the average number of unique classes with sales per store per week?
	What is the average number of unique departments with sales per store per week?
Tran History	How many days of detail transaction data will be retained?
	How many months of sales history will be retained?
	Will sales history be rolled up by week to the subclass, class, and department level in RMS? (hstbld)
	Will sales history be rolled up by month to the subclass, class, and department level in RMS? (hstbldmth)
	Will sales history be rolled up by differentiator in RMS? (hstbld_diff)
Calendar	How many years will be defined in the RMS calendar?
Stock Ledger	Will RMS be configured to support multiple currencies?
	Will stock ledger data be maintained at the day level in RMS? (saldly)
	Will stock ledger data be maintained at the week level in RMS? (salweek)
	Will stock ledger data be maintained at the month level in RMS? (salmth)
	Will stock ledger data be maintained at the half-year level in RMS? (saleoh)
VAT	Will VAT functionality be used in RMS?
	How many VAT codes will be defined?
	What is the average number of tax rates per VAT code?
	How many VAT regions will be defined?
	Will retail values in the stock ledger include VAT?
OTB	Will Open to Buy (OTB) functionality be used in RMS?
	How many months of OTB information will be maintained?
	How many months of OTB forward limit information will be maintained?
Purchase Orders	How many orders will be created manually in RMS per month (excluding orders generated via replenishment, investment buy, VMI, or Allocations)?
	What percentage of manually created orders will be approved?
	What is the average number of items per order?
	What is the average number of locations per order?
	What percentage of items on an order will have order quantities for every location on the order?
	What percentage of orders will be revised after initial order approval?
	For orders revised after initial order approval, what is the average number of revisions per order?
	What percentage of orders will have applicable deals?
	What percentage of orders will contain customer information?
	What is the average number of days between purchase order creation and purchase order approval?
	What is the average number of days between purchase order approval and purchase order completion (PO closed)?
	How many months of order history will be retained?
	How many days of order revision history will be retained?
Contracts	Will contract functionality be used in RMS?
	How many Type A (plan/availability) contracts will be created and submitted per month?
	How many Type B (plan/no availability) contracts will be created and submitted per month?
	How many Type C (no plan/no availability) contracts will be created and submitted per month?
	How many Type D (no plan/availability) contracts will be created and submitted per month?

	What is the average number of items per contract for contract types A or B (plan)?
	What is the average number of items per contract for contract types C or D (no plan)?
	What percentage of contracts will be approved?
	What is the average number of days between contract creation date and contract start date?
	What is the average number of days between contract start date and contract end date?
Deals	Will complex deals functionality be used in RMS?
	How many annual off-invoice deals will be defined?
	How many promotional (non-annual) off-invoice deals will be created per month?
	How many bill-back deals will be created per month?
	How many fixed deals will be created per month?
	How many bracket deals will be created per month?
	How many PO-specific deals will be created per month?
	What percentage of non-annual deals will be approved?
	What is the average number of days between deal creation and deal active date for non-annual deals?
	What is the average number of days between deal active date and deal close date for non-annual deals?
	What is the average number of deal components per deal for non-annual deals?
	What is the average number of deal components per deal for annual off-invoice deals?
	What percentage of deal components will have thresholds (e.g., volume discounts) defined?
	What is the average number of items per deal for all deal types excluding fixed deals?
	What is the average number of locations per deal for all deal types?
	For bill-back deals, what is the average number of bill-back periods per deal?
	For fixed deals, what is the average number of collection periods per deal?
	How many months of deal history will be retained?
Deals POP	Will deals proof of performance functionality be used in RMS?
	What percentage of non-annual, non-fixed deals will have proof of performance terms defined?
	What percentage of fixed deals will have proof of performance terms defined?
	For deals with proof of performance terms defined, what is the average number of terms per deal?
Cost Change	How many manual cost changes will be submitted per month?
	What percentage of manual cost changes will be approved?
	What is the average number of items per manual cost change?
	What percentage of manual cost changes are by item/location (vs. by item only)?
	What is the average number of locations per manual item/location cost change?
	What is the average number of days between approval date and effective date for manual cost changes?
EDI Cost Change	How many EDI cost changes will be submitted per month?
	What percentage of EDI cost changes will be approved?
	What is the average number of items per EDI cost change?
	What percentage of EDI cost changes are by item/location (vs. by item only)?
	What is the average number of locations per EDI item/location cost change?
	What is the average number of days between approval date and effective date for EDI cost

	changes?
	How many days of accepted/rejected EDI cost change information will be retained?
ELC	Will Estimated Landed Cost (ELC) functionality be used in RMS?
	How many location-level cost zone groups will be defined (max 1)?
	How many zone-level cost zone groups will be defined?
	What is the average number of zones per zone-level cost zone group?
	What is the average number of locations per zone-level cost zone?
	How many Computation Value Bases (CVB) will be defined?
	What is the average number of components per CVB?
	How many outside locations will be defined (e.g., entry ports, discharge ports, lading ports)
	Will upcharge functionality be used in RMS?
	What percentage of departments will have up charges defined?
	For departments with upcharges defined, what is the average number of from locations?
	For departments with upcharges defined, what is the average number of to locations?
	What is the average number of up charge components per dept upcharge?
	What percentage of items will have upcharges defined (including items inheriting upcharge info from dept settings)?
	For items with upcharges defined, what is the average number of from locations?
	For items with upcharges defined, what is the average number of to locations?
	What is the average number of up charge components per item upcharge?
	Will expense profile functionality be used in RMS?
	How many country expense profiles will be defined?
	For country expense profiles, what is the average number of lading port / discharge port combinations?
	How many supplier/cost zone expense profiles will be defined (including partners)?
	How many supplier/country expense profiles will be defined (including partners)?
	For supplier/country expense profiles, what is the average number of lading port / discharge port combinations?
	What is the average number of components per expense profile?
	What percentage of items (tran-level and above) will have expense profiles?
	What is the average number of expense profiles per item/supplier combination (incl. country, zone, and backhaul)?
MRT	Will Mass Return Transfer (MRT) functionality be used in RMS?
	How many mass return transfers will be created per month for consolidation to reallocate to other stores?
	How many mass return transfers will be created per month for consolidation to return to vendor?
	What is the average number of source locations per MRT?
	What is the average number of items per MRT?
	What percentage of MRTs will be approved?
	What is the average number of days between MRT creation and approval?
	What is the average number of days between MRT approval and completion?
	How many days of MRT history will be retained?
Transfers	Will transfer functionality be used in RMS?
	How many transfer entities will be defined?
	How many transfer zones will be defined?
	How many manual transfers will be created per month (excluding transfers generated or combined by replenishment or MRT)?

	What is the average number of items per transfer?
	What percentage of transfers will meet to/from upcharge conditions?
	What percentage of transfers will be approved?
	What is the average number of days between transfer creation and transfer approval?
	What is the average number of days between transfer approval and delivery date for transfers without work orders?
	How many months of transfer history will be retained?
Work Order	Will work order functionality be used in RMS?
	How many transfer work order activities will be defined?
	How many work order templates will be defined?
	What is the average number of activities per work order template?
	What percentage of transfers will have work orders attached?
	What is the average number of days between transfer approval and delivery date for transfers including work orders?
	What percentage of work orders will require finishing?
	For finishing work orders, what is the average number of activities per item?
	What percentage of work orders will require packing?
	For packing work orders, what is the average number of items per pack exploded or per pack built?
	What percentage of work orders will require item transformation?
RTV	How many return to vendor (RTV) orders will be created or interfaced to RMS per month, excluding MRT RTV?
	What percentage of RTV orders will be approved?
	What is the average number of items per RTV order?
	What is the average number of locations returned from per RTV order?
	What is the average number of days between RTV order approval and RTV order completion?
	How many months of RTV order history will be retained?
Shipping	What is the average number of receipts per purchase order?
	What is the average number of receipts per transfer?
Inv Adjustment	Will inventory adjustment functionality be used in RMS?
	How many inventory adjustments by item will be entered manually in RMS per month?
	What is the average number of locations per inventory adjustment by item?
	How many inventory adjustments by location will be entered manually in RMS per month?
	What is the average number of items per inventory adjustment by location?
	How many item/location inventory adjustments will be generated per DAY from external systems (e.g., RWMS, SIM)?
	What percentage of inventory adjustments (both manual & external) will adjust inventory to non-salable status?
	How many months of inventory adjustment history will be retained?
Activity Schedule	Will activity schedule information for stores and ware houses be defined in RMS?
	How many company closed dates will be defined?
	How many locations will remain open on company closed dates?
	How many locations will have closed dates defined excluding company closed dates (i.e. company open, individual store closed)?
	What is the average number of closed dates per location with closed dates defined?

Replenishment	Will replenishment functionality be used in RMS?
	What percentage of stores will have delivery schedules defined?
	What percentage of warehouses will have delivery schedules defined?
	What percentage of suppliers will have delivery schedules defined?
	What percentage of suppliers will have inventory management constraints defined?
	What is the average number of departments per supplier for inventory management constraint definitions?
	What percentage of items will be replenished?
	For items on replenishment, what is the average number of review days per item per week?
	What is the average number of locations per item on replenishment?
	What percentage of items on replenishment will have substitute items defined?
	What percentage of replenishment results are expected to generated orders with zero quantity?
	Will replenishment results be generated even if the recommended order quantity is zero?
	What percentage of replenishment orders will be approved?
	How many days of replenishment history should be retained for generated orders in worksheet status?
	How many days of replenishment results history will be retained?
Stock Counts	Will stock count functionality be used in RMS?
	How many unit and dollar stock counts will be scheduled in RMS per month?
	What is the average number of locations per unit & dollar stock count?
	What is the average number of subclasses per unit & dollar stock count?
	What is the average number of days between stock count request and stock count start date?
	What is the average number of days between stock count start and end dates?
	How many unit stock counts will be generated per month?
	What is the average number of item/location combinations per unit stock count?
Sales Tax	Will geographic sales tax information be maintained in RMS?
	How many country geocodes will be defined for sales tax in RMS?
	How many state geocodes will be defined for sales tax in RMS?
	How many county geocodes will be defined for sales tax in RMS?
	How many city geocodes will be defined for sales tax in RMS?
	How many district geocodes will be defined for sales tax in RMS?
	What percentage of stores will have geocodes assigned?
	What is the average number of geocodes assigned per store?
	How many tax codes will be defined?
	What is the average number of tax rates per tax code?
Timelines	Will timeline functionality be used in RMS?
	How many timelines will be defined per month?
	What is the average number of timeline steps per timeline?
	What is the average number of days to completion for all steps on a timeline?
Documents	Will document functionality be used in RMS?
	How many documents will be defined?
Ticketing	Will ticketing functionality be used in RMS?
	How many ticket types will be defined?
	What is the average number of attributes per ticket type?
	What percentage of items will have a ticket assigned?
Regionality	Will regionality functionality be used in RMS?

	How many group/location combinations will be defined?
	What is the average number of elements per group/location combination?
Users	How many users will be defined in RMS?
	What is the maximum number of concurrent users expected in RMS?
	What is the average number of concurrent users expected in RMS?
Navigation	How many navigation roles will be defined (minimum 1)?
Location Security	Will location security functionality be used?
	How many location security groups will be defined?
	What is the average number of locations per location security group?
	What is the average number of users per location security group?
Product Security	Will product security functionality be used?
	How many product security groups will be defined?
	What is the average number of subclasses per product security group?
	What is the average number of users per product security group?
Zone Security	Will zone security functionality be used?
	How many zone security groups will be defined?
	What is the average number of zones per zone security group?
	What is the average number of users per zone security group?
POS Config	Will POS configurations be defined in RMS?
	How many coupon configurations will be created per month?
	How many money order configurations will be created per month?
	How many pay in / pay out configurations will be created per month?
	How many product restriction configurations will be created per month?
	How many tender type configurations will be created per month?
	How many touch button configurations will be created per month?
	What is the average number of days between POS configuration creation and configuration effective date?
	What is the average number of days between POS configuration effective date and expiration date?
	What is the average number of stores per POS configuration?
	For coupon configurations, what is the average number of items per configuration?
	For product restriction configurations, what is the average number of items per configuration?
RPM	How many zone groups will be defined in RPM?
	What is the average number of zones per zone group?
	What is the average number of locations per zone?
	Will retail prices be managed at warehouse locations in RPM?
	How many RPM users will be defined?
	What is the maximum number of concurrent RPM users?
	What is the average number of concurrent RPM users?
	How many calendars will be defined in RPM?
	What is the average number of periods per calendar?
	How many corporate-level price guides will be defined?
	How many department-level price guides will be defined?

	What is the average number of departments per department-level price guide?
	What is the average number of price intervals per price guide?
	How many reason codes will be defined in RPM?
	How many item link codes will be defined in RPM?
	How many market basket codes will be defined in RPM?
	How many candidate rules will be defined?
	What is the average number of conditions per candidate rule?
	Will candidate rule conditions vary by department?
	What percentage of candidate rule conditions will vary by department?
	For candidate rule conditions that vary by department, what is the average number of department variances per condition?
	Will area differential functionality be used in RPM?
	How many primary area differentials will be defined at the zone level?
	How many primary area differentials will be defined at the location level?
	For area differentials defined at the location level, what is the average number of locations per differential?
	What percentage of area differentials will contain excluded items?
	For area differentials with excluded items, what is the average number of items excluded?
	Will pricing strategy functionality be used?
	How many clearance pricing strategies will be defined?
	For clearance pricing strategies, what is the average number of step markdowns to be defined?
	How many margin pricing strategies will be defined?
	How many competitive pricing strategies will be defined?
	For competitive pricing strategies, what is the average number of competitor stores per strategy?
	What is the average review period in days for all pricing strategies?
	During each review cycle, what is the average number of departments reviewed?
	How many manual price changes will be approved per month?
	How many system-generated price changes will be approved per month?
	What is the average number of days between price change creation and price change approval for manual price changes?
	What is the average number of days between price change approval and price change effective date?
	What is the average number of items per price change?
	What is the average number of locations per price change?
RPM Clearance	How many manual clearance price changes will be approved per month?
	How many system-generated clearance price changes will be approved per month?
	What is the average number of days between clearance creation and clearance approval for manual clearances?
	What is the average number of days between clearance approval and clearance effective date?
	What is the average number of items per clearance?
	What is the average number of locations per clearance?
	What percentage of clearances will have price reset dates defined?
	For clearances with reset dates, what is the average number of days between clearance effective date and reset date?
	How many months of clearance history will be retained?

RPM Promo	Will you use promotion events?
	How many promotion events will be defined per year?
	How many promotions will be created per month?
	What is the average number of locations per promotion?
	What is the average number of days between promotion creation and promotion change effective date?
	What is the average number of components per promotion?
	What percentage of promotions will be rejected?
	How many days of rejected promotion history will be retained?
	How many months of promotion history will be retained?
	What percentage of promotions are simple promotions?
	For simple promotions, what is the average number of items per promotion?
	What percentage of promotions are buy/get promotions?
	For buy/get promotions, what is the average number of buy items?
	For buy/get promotions, what is the average number of get items?
	What percentage of promotions are threshold promotions?
	What is the average number of items per threshold?
	What is the average number of threshold intervals per threshold promotion?
	What percentage of promotions will have applicable deals?
	For promotions with applicable deals, what is the average number of deals per promotion?

Referências

- [DELL] Dell. Enterprise Support Tutorials. <http://support.dell.com/support/topics/global.aspx/support/entvideos/raid?c=us&l=en&s=gen/> (acedido em Junho de 2009).
- [DONB03] Donald K. Burleson. *Oracle Space Management Handbook*, 2003, BMC Software e DBAZine.
- [WIKI09a] Wikipedia. Retailing, 2009. <http://en.wikipedia.org/wiki/Retailing> (acedido em Maio de 2009).
- [WIKI09b] Wikipedia. RAID, 2009. <http://en.wikipedia.org/wiki/RAID> (acedido Maio de 2009).
- [WIPR06] Shubba Krishna Murthy. Database Capacity Planning. Documento Interno, Wipro Technologies, Março 2006.
- [WIPR09] Pedro Ferreira. ORMS Datamodel. Documento Interno, Wipro Technologies, Março 2009.
- [SCHO08] Scholarpedia. B-tree and UB-tree, 2008. http://www.scholarpedia.org/article/B-tree_and_UB-tree (acedido em Maio de 2009).
- [DOUG79] Douglas Comer. The Ubiquitous B-Tree. Computer Science Department, Purdue University, Indiana. Junho 1979.
- [ORAC08] Immanuel Chan. *Oracle Database Performance Tuning Guide, 11g Release 1 (11.1)*, Julho 2008, Oracle.
- [ORAC05] Michele Cyran. *Oracle Database Concepts, 10g Release 2 (10.2)*, Outubro 2005, Oracle .
- [ORAC09] Oracle. Retail Applications, 2009. <http://www.oracle.com/industries/retail/index.html> (acedido em Junho de 2009).
- [GAVIN02] Gavin JT Powell, Capacity Planning with Oracle, Outubro 2002, OracleDBAExpert.com .