CONSULTAS FLEXÍVEIS PARA INTELIGÊNCIA COMPETITIVA

Darlene Figueiredo Borges

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA CIVIL.

Aprovada por:

Prof. Nelson F. F. Ebecken, D.Sc.

Prof. Elton Fernandes, PhD

Profª Márcia P. B. Gottgtroy, D.Sc.

Profª Marta L. Queiros Mattoso, D.Sc.

Profª Beatriz S. L. Pires de Lima, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

NOVEMBRO DE 1999

BORGES, DARLENE FIGUEIREDO

Consultas Flexívies para Inteligência Competitiva [Rio de Janeiro] 1999

X, 183p. 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia Civil, 1999)

Tese – Universidade Federal do Rio de Janeiro, COPPE

1. Inteligência Competitiva 2. Data Warehouse 3. Data Mart 4. Data Mining 5. Tratamento de Incerteza

I. COPPE/UFRJ II.Título (série)

Aos meus pais, Oderlino e Darcy

À você, Vulmar

AGRADECIMENTOS

Ao Nelson, meu orientador, pelo apoio na definição do trabalho, orientação e estímulo, pela amizade, pelo carinho e especialmente pelo meu amadurecimento como aluna e pesquisadora.

Aos demais professores da COPPE/UFRJ que colaboraram diretamente para minha formação acadêmica no doutorado.

Aos meus pais, Oderlino e Darcy, pela formação, pela dedicação, pelo carinho, pelas orações e especialmente pelo amor.

A Vulmar, meu noivo, pelo carinho, pelo amor e por ter suportado os longos momentos de ausência.

Aos amigos sempre presente, Márcia e Paulo César, pela amizade, pelo incentivo e pelo carinho.

A Cléa Borges e Paulo Borges Júnior, pela amizade e pela tranqüilidade para a realização deste momento.

Aos amigos de verdade, Mônica, Padilha, Lúcia, Léa Mara, Bete, Guilherme e Myrian, pela amizade, presença e incentivo.

Aos companheiros da COPPE/CIVIL, pelo convívio, pelos ótimos momentos compartilhados e carinho durante a formação de minha vida profissional.

A Estela e Bete, pelo carinho e pelo apoio administrativo, sempre prestativas durante todas as fases deste trabalho.

Aos colegas do DCC/UFMT, em especial aos professores Einstein, Maria Eunice, Arlenes, Nelcileno e José Aírton, pela amizade e pela irrestrita colaboração para viabilizar a conclusão desta tese.

Ao CNPQ, pelo apoio financeiro durante o doutorado.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

CONSULTAS FLEXÍVEIS PARA INTELIGÊNCIA COMPETITIVA

Darlene Figueiredo Borges

Novembro/1999

Orientador: Nelson Francisco Favilla Ebecken

Programa: Engenharia Civil

Com a larga aplicabilidade computacional na ciência, na área empresarial e governamental, fica evidente que durante anos grandes volumes de informações vem sendo acumulados em diversos bancos de dados. A descoberta de interessantes conhecimentos, a partir desses bancos de dados, tem se tornado essencial para as mais variadas utilizações no mundo de negócios.

O presente trabalho desenvolve um ambiente central integrado. Na construção desse ambiente unificado é utilizado o Data Mining para realizar a transformação de dados sobre as informações a serem aí depositadas. Além disso, desenvolve também um conjunto de consultas flexíveis, cuja finalidade fundamental inclui a descoberta de conhecimento sobre os dados de bens patrimoniais de uma empresa de transporte, proporcionando novas maneiras de aumentar conhecimentos cruzados e mais sofisticados. Com base no resultado é possível gerar apoio para tomada de decisão e soluções estratégicas de mais alto nível de qualidade em tempo hábil que necessita um ambiente competitivo.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

FLEXIBLE QUERIES FOR COMPETITIVE INTELLIGENCE

Darlene Figueiredo Borges

November/1999

Advisors: Nelson Francisco Favilla Ebecken

Program: Civil Engineering

The large computational applicability in science for years, provide enough evidence that large quantities of information have been stored in many databases both in state and private areas. Based upon these databases, interesting knowledge has been discovered and has become an essential ingredient for a huge diversity of in the business world.

The present research develops a central integrated environment. To accomplish data transformation on the given information to be stored, Data Mining is used in the construction of this unified environment. Furthermore, it also develops a group of flexible queries whose fundamental purpose includes the discovery of particular knowledge about private goods data of transport enterprise, providing new ways to increase multi and more sophisticated knowledge. Based on this result it is possible to create decisions support systems and high quality levels of strategic solutions in a given time that needs a competitive environment.

ÍNDICE

Página

I. INTRODUÇÃO 1

II. DATA WAREHOUSE E DATA MINING 11

II.1. Introdução 11

II.2. Conhecendo o ambiente de Data Warehouse (DW) 13

II.2.1. Construindo um Data Warehouse 14

II.2.1.1. Aspectos fundamentais para construção de um Data Warehouse 16

II.2.1.2. Projeto de Data Warehouse 16

II.2.1.3. Modelo de dados para ambiente de Data Warehouse 19

II.2.1.4. Normalização/Denormalização 21

II.2.1.5. Metadados 22

II.2.2. Principais aspectos da transformação e integração dos dados .. 23

II.2.3. Tecnologia do Data Warehouse 25

II.2.4. Tipos de SGBDs e o Data Warehouse 26

II.2.5. SGBD Multidimensional e o Data Warehouse 27

II.2.6. Operação de atualização do Data Warehouse 28

II.3. Metadados 29

II.3.1. Conceitos e métodos para construir um ambiente integrado de dados 30

II.3.1.1. Refinamento e transformação de dados 30

II.3.1.2. Características do ambiente integrado de dados 31

II.3.2. Tipos de metadados 35

II.3.3. Padronização do metadados 36

II.4. Características dos dados em um ambiente de DW 37

II.5. Processo de Knowledge Discovery in Database (KDD) 40

II.5.1. Fases do processo de KDD 41

II.6. Data Mining (DM) 43

II.6.1. Árvore de Decisão 46

II.6.1.1. Algoritmos de Árvore de Decisão 47
II.6.1.2. Algoritmos de indução usados 49
II.6.2. Hardware e Software para DM 53
II.7. Conhecendo o ambiente de Data Mart 56
II.8. Integração do Data Mining (DM) com o Data Warehouse/Data Mart 58
II.9. Considerações finais 60
III. TRATAMENTO DE INCERTEZA 62
III.1. Introdução 62
III.2. Incerteza 63
III.3. Teoria de Conjunto fuzzy 65
III.3.1. Conceitos básicos 65
III.4. Conhecendo os Princípios e Características de Banco de Dados Fuzzy 72
III.4.1. Conceitos básicos de banco de dados 72
III.4.2. Projeto de Banco de Dados 73
III.4.3. Modelos de Banco de Dados 75
III.4.4. Incerteza em banco de dados convencional 76
III.4.5. Incerteza em banco de dados orientado a objetos 82
III.4.5.1. Modelo de banco de dados orientado a objetos 82
III.4.5.2. Modelo de banco de dados orientado a objetos fuzzy 85
III.4.6. Modelos alternativos para banco de dados fuzzy 90
III.4.6.1. Extensão fuzzy para o modelo E-R 90
III.4.6.2. Extensão fuzzy para o modelo de dados de rede 92
III.4.7. Evolução da incerteza nos banco de dados 93
III.5. Árvore de Decisão de Dados Fuzzy 94
III.6. Considerações finais 96
IV. FUZZY QUERIES 98
IV.1. Introdução 98
IV.2. Sistema de Gerenciamento de Dados (SGBD) Relacional 99
IV.3. Consultas fuzzy em banco de dados 101

IV.3.1. Consultas fuzzy baseadas no conceito de similaridade/proximidade 101
IV.3.1.1. Processo de avaliação de consulta 102
IV.3.1.2. Cálculo relacional 104
IV.3.1.3. Características fuzzy para banco de dados baseado em similaridade 104
IV.3.2. Consultas fuzzy baseadas no conceito de possibilidade 107
IV.3.2.1. Teoria de possibilidade 110
IV.3.2.2. Linguagem de consulta flexível 111
IV.3.2.3. SQLf 114
IV.3.2.4. Restrição de Integridade 115
IV.3.2.5. Aspecto de implementação e processamento de consultas 116
IV.4. Considerações finais 118
V. METODOLOGIA E ESTUDO DE CASO 119
V.1. Introdução 119
V.2. Definição do problema 120
V.3. Levantamento e análise dos dados disponíveis 122
V.3.1. Visualização dos dados através da tarefa de Data Mining (DM) 124
V.3.1.1. Classificação dos dados 125
V.4. Modelagem do problema utilizando a metodologia de Data Mart ... 131
V.4.1. Modelando os dados através do modelo dimensional 134
V.4.2. Metadados 135
V.5. Fuzzy Query 138
V.5.1. Construindo uma consulta 138
V.5.2. Resultados das consultas fuzzy 143
V.5.2.1. Principais consultas fuzzy 144
V. 5.2.1.1. Consulta que classifica a variação do valor das prestações dos bens patrimoniais da empresa 144
V. 5.2.1.2. Consulta que focaliza a venda dos bens patrimoniais da companhia 151

V.6. Metodologia CFIC (Consultas Flexíveis para Inteligência Competitiva) 158
V.7. Considerações finais 161
VI. CONCLUSÕES 163
REFERÊNCIAS BIBLIOGRÁFICAS 168

CAPÍTULO I

INTRODUÇÃO

À medida que o mundo avança para o próximo milênio, novos paradigmas surgem dentro da área de ciência da computação. Não só ocorrem mudanças, mas a uma velocidade cada vez maior. A tecnologia atual dá a última palavra como modeladora da subestrutura material da sociedade e também dos padrões do pensamento humano. Além da tecnologia, outras forças estão modelando a sociedade, como a exploração de conhecimento e sua gerência. Hoje em dia, o conhecimento é considerado um recurso extremamente importante principalmente relacionado com tomadas de complexas decisões e geração de estratégias dentro das empresas. Diversas empresas estão contando com inteligência competitiva para fornecer um suporte às tomadas de decisões nos negócios.

Inteligência competitiva pode ser definida como a seleção, coleção, interpretação e distribuição de importantes informações estratégicas. É também definida [123] como a informação que revela como a empresa é competitiva. Compreende a arena da competição, tornado-se hábil para predizer as intenções dos competidores e clientes, ações governamentais e assim por diante.

O principal objetivo da atividade de inteligência competitiva é entender, tanto quanto possível, a estratégia tecnológica dos competidores e/ou as tendências de mercado. Diante da necessidade da análise de inteligência competitiva, especialistas da área podem buscar inúmeras posições dentro da organização e ações governamentais. Tradicionalmente as empresas contam com a pesquisa de mercado, porém muitas delas estão investindo em inteligência competitiva como a mais nova fonte para fornecer suporte aos departamentos de vendas e marketing e também para atividades de planejamento estratégico.

A inteligência competitiva e a teoria de gerência do conhecimento associam-se com perfeita harmonia. A gerência do conhecimento está sendo implementada dentro de muitos departamentos das organizações e vem crescendo principalmente nas

atividades de planejamento. As empresas estão reconhecendo o valor em compartilhar todo tipo de informação sobre mercados, seus negócios e processos.

Nos ambientes de negócios competitivos é vital que a empresa saiba o que os seus competidores estão fazendo, e em que direção de mercado estão se movimentando. As questões mais importantes que modernas empresas vem colocando são: quais são as tendências de mercado; em que setor os competidores estão preparando novos produtos para serem colocados no mercado; quais são as tecnologias consideradas chaves para um certo setor, etc.

Pode-se dizer que umas das tendências para se encontrar essas repostas é buscar conhecimento nas montanhas de dados acumulados durante anos nos mais diversos bancos de dados. No entanto, a principal desvantagem desses bancos de dados são as suas dimensões, freqüentemente escritos em uma linguagem especializada que não pode ser convertida facilmente, tornando assim as informações contidas de difícil acesso. Para extrair essas informações, avançadas técnicas e ferramentas precisam ser usadas de forma automática, fácil e rápida para realizar os objetivos da inteligência competitiva.

Assim, a inteligência competitiva é um processo cuja base essencial é a captação de informações dispersas e a sua interpretação em sinais de alerta, os quais podem ser transformados em força motriz ou deflagradora de ações ou decisões importantes no dia-a-dia da empresa.

Atualmente, ainda é muito tímida a prática do uso de inteligência competitiva nas empresas, sendo que somente uma pequena parte delas têm se mostrado atenta. Entretanto, a lentidão da sua utilização não deixa de ter explicação. As maiores dificuldades têm sua origem na natureza das informações, na dificuldade de se trabalhar com informações imprecisas, e principalmente na fragmentação da informação, isto é, as informações só podem se apresentar sob a forma de trechos pacientemente coletados, reunidos, interpretados e validados. Isoladamente, cada informação é insignificante, entretanto, relacionada com outras, ela adquire progressivamente um significado.

As informações são então qualitativas, parciais, incertas, oriundas de fontes cujo acesso é pouco usual, com apresentação variável e de conteúdo ambíguo, para não mencionar os suportes multiformes, em se tratando de meios e plataformas

tecnológicas. Devido a estas características, o processo de informação que constitui a inteligência competitiva emerge dificuldades que a empresa não conhece quando envolvida com outros fluxos de informação, os quais são essencialmente fluxos de informações operacionais. Portanto, para dispor de informações estratégicas é necessário querer realmente buscá-las. De fato, a captação e o armazenamento das informações irão exigir aplicação de técnicas e métodos tecnológicos bastante avançados.

Na sua essência, a inteligência competitiva que envolve um trabalho direto com informação, trabalha com os seguintes processos:

1. Compreender o problema a ser analisado

Entender o problema do negócio, trabalhando junto com um especialista do domínio ou com um usuário final.

2. Definir a fonte dos dados e uma estratégia de pesquisa

Nos dias atuais, o problema de trabalhar com informação não é complicado, pois existe um volume de dados muito extenso. O mais complicado é recuperar a informação necessária dentro desse volume de dados.

A Internet é uma das ferramentas que permite que o usuário final obtenha informações de várias empresas, centros de pesquisas e instituições financeiras. Ela é caracterizada pelo baixo custo, contendo uma grande velocidade de transmissão de dados, um simples processo de conexão, e acima de tudo, sempre com novas inovações nos serviço on-line fornecidos para o usuário final. Os navegadores da Internet sabem que muitos dados e informações podem ser obtidas. A principal desvantagem é que esse "fluxo excessivo" de informação, é normalmente livre, entretanto, a informação não é certificada, ou seja, essa informação pode estar incorreta ou incompleta, portanto, esse tipo de dado não é normalmente utilizado na inteligência competitiva.

As informações advindas dos sistemas de banco de dados são as mais usadas. Geralmente possuem uma documentação mais organizada, normalmente dedicada a uma área específica.

E uma outra fonte de informação pode ser um banco de dados de origem particular. Porém, esses dados precisam estar bem modelados e analisados e também serem projetados com a máxima utilidade. Isso indica que o ambiente dos dados necessita possuir alguns requisitos, tais como: acessibilidade e grandes compartilhamentos dos dados; simplicidade nas interações das inter-aplicações; segurança quanto a padronização dos dados; etc.

Há algum tempo a tecnologia de banco de dados vem sofrendo mudanças com relação à perspectiva de dados. Sem dúvida o estilo de aplicação que mais prevalece era, e é, baseado em um paradigma de arquivo. As primeiras aplicações que surgiram foram os bancos de dados hierárquicos e os de rede, com estilos de organização física, e eram primariamente de navegação. Por aproximadamente mais de quinze anos essas aplicações tenderam a serem tratadas como padrões, no entanto, não tinham um forte fundamento teórico e não suportavam noções de independência de dados físicos e lógicos. De maneira a propiciar mais flexibilidade na organização de grandes bancos de dados e minorar alguns problemas dos primeiros bancos de dados, apareceu no início dos anos 70, o modelo de dados relacional e são os SGBD's mais populares até os dias de hoje.

Portanto, os banco de dados estão disponíveis há bastante tempo com o propósito de se fazer o processamento operacional, geralmente transacional. Nos últimos anos, surgiu um conceito de banco de dados mais sofisticado - que atende às necessidades informacionais ou analíticas, ou seja, que atende às necessidades relacionadas com tomada de decisões, chamado de Data Warehouse. A história do Data Warehouse começa com uma evolução durante a qual são examinadas as forças mais impactantes sobre as empresas.

O processamento analítico examina amplos espectros de dados para detectar tendências, assim muitos registros são acessados. Além disso, muito raramente os dados são atualizados no nível de registro individual. Os registros estão constantemente sendo acessados, e seus conteúdos são agrupados para análise, mas ocorre pouca ou nenhuma alteração dos registros individuais. Neste processamento, os requisitos de tempo de resposta são mais atenuados, normalmente alcançam de 30 minutos a 24 horas e os usuários são grupos bastante restritos.

O ambiente de Data Warehouse é mais granular, variável no tempo, totalmente integrado, baseado em negócios, e contém bastante nível de sumarização, sendo perfeitamente aplicável para muitas análises heurísticas.

Depois de definir a origem e o ambiente dos dados, uma estratégia de pesquisa da documentação deve ser definida e implementada para melhor identificar o mais importante conjunto de dados. Pesquisar informações é uma especialidade e um trabalho difícil. O sucesso ou insucesso da atividade de tomada de decisões, ambos do ponto de vista da informação e do ponto de vista econômico, baseia-se na capacidade de pesquisa no banco de dados.

3. Extração automática e interpretação dos resultados

Seleção de uma ferramenta correta para uma análise automática desse conjunto de dados para extração de conhecimento "desconhecido".

Atualmente existem vários métodos para analisar automaticamente o conjunto de dados, dentre eles destacam-se técnicas estatísticas e técnicas de Data Mining.

A estatística apresenta a distribuição dos dados e de suas variáveis, e permite detectar o fenômeno contido nos dados. A análise de freqüência contabiliza informações sobre o assunto, novas metodologias, e ruídos. A técnica de composição visualiza a conexão entre as informações e apresenta algumas decomposições. A técnica de análise de dados é utilizada para resolver as prováveis decomposições.

A técnica de Data Mining refere-se a aplicação de algoritmo de extração de padrões dos dados dentro do processo de descoberta de conhecimento sobre grandes volumes de dados, permitindo buscar informações que aparentemente, estão camufladas ou escondidas, possibilitando, com isso, agilidade nas tomadas de decisões. Essa técnica tem sido realizada nos últimos anos através de redes neurais, algoritmos genéticos, algoritmos de associações, árvore de decisão, etc.

Em suma, toda tomada de decisão e planejamento estratégico envolve um trabalho direto com informação, ou seja, trabalha com o processo de obter informação inseridas nas seguintes fases: a primeira está relacionada com a compreensão da

informação, a segunda trabalha com a integração da informação dentro de determinadas estratégias, e então a terceira mobiliza os recursos para executar um curso de ações.

Ainda que as informações sejam organizadas e preparadas para serem avaliadas, é preciso aceitar que as imprecisões se encontram nos conjuntos de idéias e mesmo nas informações, e então não podem simplesmente serem corrigidas pela utilização de instrumentos de observação mais acurados ou de tratamentos probabilísticos sofisticados. O tratamento das imprecisões se enquadram dentro da categoria de conceitos fuzzy (nebulosos).

Lógica fuzzy surgiu formalmente com a publicação de "*Fuzzy Sets*" no periódico "*Information and Control*" em 1965 pelo fisico-matemático Lofti Zadeh. Zadeh não inventou a lógica fuzzy do nada, existem várias histórias relacionadas com a exploração de alternativas para tratar a lógica booleana.

A preocupação em tratar a imprecisão é bem antiga e o raciocínio do desenvolvimento de métodos para auxiliar essa tarefa data de 1654, quando Pascal e Fermat propuseram a teoria clássica da probabilidade. Dessa data até hoje, esses métodos se multiplicaram; a própria probabilidade clássica evoluiu, gerando muitos ramos de pesquisa. Devido a complexidade da questão, a maioria das teorias e métodos é orientada a auxiliar no tratamento de alguns tipos de imprecisão, como por exemplo, a teoria clássica da probabilidade se propõe a tratar predição de eventos (probabilidade); a lógica não monotônica, as exceções; a lógica modal se propõe a lidar com a predição de eventos enfocando a possibilidade e com a precisão de conceitos e fatos; a lógica fuzzy, a precisão, a verdade ou a probabilidade de uma proposição, a abrangência e a descrição de uma condição ou situação [57].

A lógica fuzzy é uma lógica que tenta combinar a imprecisão associada a eventos naturais com a força computacional para produzir sistema de raciocínio flexível, altamente inteligente e robusto. A lógica fuzzy repousa sobre dois conceitos básicos: o da imprecisão e o da granulação.

Imprecisão é a condição de classes de objetos cuja fronteira não pode ser definida com precisão. Existem duas classes de objetos imprecisos. A primeira classe deriva do comportamento aleatório, como por exemplo, as vibrações das máquinas, que podem ser tratadas com instrumento estatístico bastante sofisticado. A segunda

classe pertence ao campo da informação, aos processos de pensamento, racionalização, cognição e percepção, como por exemplo, a dor. Pode-se sentir dor, mas não se pode comunicar com precisão a dor, ou seja, nenhum instrumento pode aferir a intensidade dessa dor.

A granulação é a repartição de um objeto em uma coleção de "grânulos", considerada como pertencente ao mesmo grupo, por similaridade, proximidade, funcionalidade ou por ser impossível a sua separação em classes distintas. O exemplo usado por Zadeh é o do corpo humano, que pode ter grânulos como: cabeça, queixo, estômago, sangue e pernas. O que une esses elementos tão distintos é pertencerem ao mesmo corpo. A granulação pode tender a ser firme ou fuzzy, densa ou esparsa. Mas as categorias da lógica fuzzy não são bivalentes como as das lógicas tradicionais.

Essa negação da bivalência não é nenhuma novidade. A lógica modal já incorpora conceitos como esses (para alguns, no lugar de, para todos, etc) desde a primeira metade do século. A novidade é trazer essas noções a partir do real e não a partir da abstração. O desconcertante no trabalho dos teóricos de lógica fuzzy, como os que trabalham na área de neurologia, é a demonstração de que os métodos de quantificação tem pouca utilidade porque o mundo é, em si, fuzzy. Isto é, as fronteiras entre os objetos, sejam eles reais, sejam ideais, são sempre imprecisas. Em suma, o que o mudou é que, enquanto a ciência vê o mundo em preto-e-branco e a estatística vê o mundo em gradações cinza, a lógica fuzzy considera que o mundo é mais do que isso. É preto, branco, cinza e exuberantemente colorido [115].

A lógica fuzzy evidencia três preceitos dentro da pesquisa: o primeiro é a esperança de que o mundo seja formado por unidades discretas identificáveis. O fenômeno é contínuo e muito difícil de se fixar em um conceito, mesmo em um conceito precário. O interessante é o fato de que a probabilidade é discreta, ou seja, pode-se calcular a probabilidade de uma iniciativa obter êxito ou fracassar. Mas não se tem como calcular a probabilidade de obter um êxito relativo ou de se constituir em quase fracasso, a menos que se defina o que esses termos significam, o que é claro, os tornariam imediatamente discretos.

O segundo é a fé na quantidade como ideal de precisão. Como Einstein disse uma vez "quanto mais as leis da matemática referem-se à realidade, menos precisas elas são. E quanto mais elas são precisas, menos se referem à realidade".

E o terceiro preceito evidencia o fato que não só a realidade é imprecisa como o pensamento é vago. Isto é, fora as raras ocasiões em que se dedica à reflexão abstrata, a mente trabalha exclusivamente com conceitos fuzzy.

Assim, aplicar o conhecimento da lógica fuzzy neste estudo é extremamente viável e simples, pois as categorias de tomadas de decisões e planejamentos estratégicos são vagas por natureza e não devido à ignorância. O conceito viabilidade de venda, por exemplo, é vago, não tem fronteiras definidas. Pertencem, inquestionavelmente, à segunda classe de objetos imprecisos que compreende os objetos vagos que não podem ser enquadrados pela teoria de estatística convencional. São conceitos perfeitamente operacionais ao nível de pensamento, mas impossíveis de se precisar.

O objetivo do presente trabalho é estabelecer uma metodologia que compreenda todas as etapas e características descritas, utilizando ferramentas que representem o "state-of-art", que possam ser consideradas comerciais e não simplesmente investigações acadêmicas e que produza um efeito diferencial e inovador na tomada de decisões em um ambiente competitvo.

Esta metodologia que envolve modelos de dados e data warehouse, descoberta de conhecimento e data mining, consultas flexíveis e fuzzy queries, deve considerar como enfatizado anteriormente, a complexidade e incerteza dos dados, a necessidade de se obter respostas em tempo hábil e o caracter primordialmente qualitativo do processo de descoberta do conhecimento em um ambiente competitivo.

O restante desta tese é organizada da seguinte forma: o capítulo II apresenta o ambiente de Data Warehouse, conceito de Data Mining e o beneficio da sua integração com Data Warehouse. Definições sobre a lógica fuzzy são apresentadas no capítulo III, focalizando inclusive a sua aplicabilidade nos diversos modelos de dados. No capítulo IV, é abordado o tema sobre uso da lógica fuzzy na formulação de consultas imprecisas sobre banco de dados. Uma metodologia que visa absorver esses paradigmas é proposta, implementada e testada. A abordagem sobre o ambiente unificado é discutida no capítulo V, e também o resultado de algumas consultas fuzzy sobre o conjunto de dados disponível. No capítulo VI são apresentadas as conclusões sobre o trabalho. Outros detalhes dos capítulos são apresentados a seguir.

O capítulo II explica o ambiente de Data Warehouse (DW) incluindo estudos relativos à sua construção e os aspectos fundamentais para essa construção, descreve o projeto de construção, apresenta a modelagem dimensional, aborda as características de integridade referencial, e apresenta um importante aspecto do ambiente de DW que diz respeito aos metadados. Apresenta os principais aspectos da transformação e integração dos dados, sendo esse um dos maiores problemas em ambientes de DW, pois normalmente tenta-se mover os dados de bancos de dados operacionais para um ambiente de DW, sem se preocupar com a sua integração. Aborda alguns requisitos tecnológicos relacionados com o DW, onde demanda um conjunto de características tecnológicas bastante simples. No DW não há atualização on-line; existe apenas exigências mínimas de bloqueio; uma interface de teleprocessamento muito básica e assim por diante. Apresenta ainda um breve comentário sobre os tipos de sistemas de gerenciamento de banco de dados e o DW. Apresenta também o conceito de processo de descobrir conhecimento em banco de dados, a técnica de Data Mining (DM) e considerações da integração de DM com DW.

No capítulo III, uma breve definição da teoria de lógica fuzzy é abordada. Algumas aplicabilidades dessa teoria sobre banco de dados são representadas, compreendendo os princípios e características de banco de dados fuzzy, passando pela representação de incerteza em banco de dados convencionais, em banco de dados orientado a objetos, em modelos alternativos em banco de dados, e finalmente visualizando a evolução da incerteza nos banco de dados.

O capítulo IV envolve a teoria de lógica fuzzy juntamente com as possíveis abordagens direcionadas na área de acesso às informações, começando pela abordagem de alguns dos principais conceitos do modelo relacional, focalizando a linguagem SQL e, portanto, introduz o uso da lógica fuzzy na construção de consultas flexíveis sobre banco de dados, mais especificamente sobre a linguagem SQL, apresentado-as através dos conceitos baseados em similaridade/proximidade e possibilidade.

O conceito para lidar com a complexidade inserida sobre os dados através de uma metodologia integrada em um ambiente unificado, utilizando os mais novos paradigmas, como DW/Data Mart, Data Mining e Fuzzy Query é representado no capítulo V. O ponto central deste capítulo é, portanto, apresentar um modelo que

envolva o planejamento e construção desse ambiente, possibilitando a captação, armazenamento, compartilhamento e análise do conjunto de dados disponíveis contendo informações sobre bens patrimoniais de uma empresa de engenharia de transporte de grande porte, sendo que estas informações possam produzir ações competitivas em tempo real.

As conclusões deste estudo encontram-se no capítulo VI. Algumas pesquisas futuras também são sugeridas.

CAPÍTULO II

DATA WAREHOUSE E DATA MINING

II.1. Introdução

A abordagem em Data Warehouse ainda é considerada por muitos como uma tecnologia que está surgindo ou até mesmo como um conjunto de tecnologias. Existe, portanto, a necessidade de ter uma visão mais clara do que é Data Warehouse? Quais são as suas principais características e quais são os seus componentes? Como é a sua arquitetura e qual é a sua estratégia de implementação? Em resumo, há um número significativo de questões que precisam ser esclarecidas para fortalecer este novo paradigma, bem como a sua completa utilização.

Essa nova tecnologia, Data Warehouse, tem como enfoque principal o armazenamento central de dados que normalmente foi extraído de um sistema de banco de dados operacional. Os dados extraídos são tratados e sofrem algumas transformações para eliminar inconsistências, são sumarizados se necessários e colocados dentro do Data Warehouse. O processo de transformação, criação de variáveis, sumarização e combinação de dados extraídos ajudam a criar o ambiente de acesso às informações. As informações no Data Warehouse são usadas somente quando são estáveis, pois a sua estabilidade é fundamental para análise e tomada de decisão.

No ambiente de Data Warehouse os usuários realizam tarefas pré-definidas, que geralmente requerem acesso a uma aplicação de banco de dados, e também é acessado por um conjunto limitado de usuários em um dado tempo. Geralmente esses usuários geram consultas com determinado grau de complexidade para encontrar um requisito complexo. Algumas vezes a resposta para uma consulta permite formular outra mais detalhada. Dados sumários representam o "o que" da situação e os dados detalhados permitem que os usuários construam uma imagem de "como" a situação foi deduzida.

Para construir um Data Warehouse é necessário que uma série de fases sejam desenvolvidas e implementadas num modelo padrão que permita um gradual e iterativo processo de implementação. Os dados num Data Warehouse não são voláteis e se constituem em um repositório de dados, entretanto, novos elementos podem ser adicionados. Um dos desafios de manutenção do Data Warehouse é planejar métodos para a identificação de novos ou de dados atualizados nos banco de dados operacionais.

Na implementação do Data Warehouse o desafio é a questão da transformação de dados. A transformação trabalha com a inconsistência nos formatos dos dados que existe quando múltiplos banco de dados contribuem para o Data Warehouse. Transformação de dados também lida com inconsistências no conteúdo dos dados. Uma vez que as transformações são feitas, regras são estabelecidas, definições podem ser criadas e incluídas nas rotinas de transformações. Um planejamento detalhado é necessário para transformar dados inconsistentes em dados consistentes.

Outro aspecto da arquitetura do Data Warehouse é criar apoio para os "metadados". Metadado é o coração do ambiente de Data Warehouse; é a informação sobre os dados que alimentam, ganham transformações, e o que existe no Data Warehouse. A criação de uma completa definição do metadado pode ser um processo de grande consumo de tempo, mas quanto melhor a definição, melhor será a compreensão do usuário, melhor será a gerência do software e melhor será a manutenção do mesmo no Data Warehouse. Metadado é um conceito genérico, mas cada implementação do metadado utiliza métodos e técnicas específicas. Esses métodos e técnicas dependem da necessidade de cada organização, das capacidades existentes, e da necessidade da interface do usuário.

Este capítulo está destinado a apresentar aspectos fundamentais que caracterizam esse novo paradigma denominado Data Warehouse. Para isso nos próximos itens são abordados: o conceito sobre o ambiente do Data Warehouse; as características fundamentais dos dados em um ambiente de Data Warehouse; conceito e estrutura do metadado; conceito de Data Mart considerado como uma extensão natural do Data Warehouse; conceito e características do processo de descoberta de conhecimento em

banco de dados; conceito de Data Mining e o benefício da sua integração com Data Warehouse.

II. 2. Conhecendo o ambiente do Data Warehouse (DW)

O Data Warehouse é um conjunto de dados baseado em assuntos, integrado, não-volátil e variável em relação ao tempo. Uma característica marcante do DW é o fato de que ele é integrado, sendo que de todos os seus aspectos, esse é um dos mais importante. As diversas aplicações já existentes aparecem sob mil formas diferentes, elas não apresentam coerência em termos de codificações, convenções de nomes dos atributos, atributos físicos, unidades de medidas dos atributos e assim por diante. O processo de introdução dos dados no DW é conduzido de forma que as muitas inconsistências das aplicações sejam desfeitas. As considerações sobre consistência são válidas para todas as questões de projeto de aplicações, como as convenções de atribuição de nomes, estruturas de chaves, unidades de medidas dos atributos e características físicas dos dados.

Outra característica importante de um DW consiste em que ele é não-volátil, mostrada na figura II.1. Os dados operacionais são acessados e atualizados registro por registro. Entretanto, os dados existentes no DW apresentam características distintas, eles são carregados (normalmente em grandes quantidades) e depois acessados.

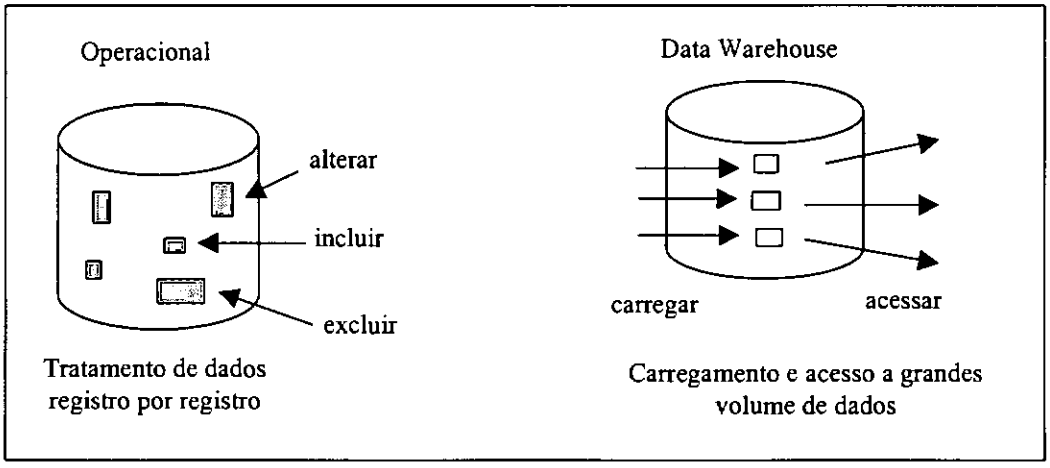


Figura II.1 – Representação da não-volatilidade

E uma outra característica significativa do DW diz respeito ao fato dele ser variável em relação ao tempo, ou seja, possui um horizonte de tempo de cinco a dez anos, dados instantâneos e a estrutura de chaves sempre contém um elemento de tempo.

II.2.1. Construindo um Data Warehouse

Os dados fluem do ambiente operacional para o DW. Na passagem do nível operacional para o nível de DW geralmente ocorre uma quantidade significativa de transformações sobre os dados.

Os DW não são construídos de uma só vez. Em vez disso, eles são projetados e povoados passo a passo, sendo, portanto, evolucionários. Os custos de construir um DW de uma vez, os recursos necessários e o transtorno causado ao ambiente, torna necessário a construção DW de forma ordenadamente iterativa, passo a passo.

II.2.1.1. Aspectos fundamentais para construção de um Data Warehouse

Um aspecto importante do projeto de DW é a questão da granularidade. A granularidade diz respeito ao nível de detalhe ou de resumo contido nas unidades de dados existentes no DW. Quanto mais detalhe, mais baixo o nível de granularidade. Quanto menos detalhe, mais alto o nível de granularidade.

A granularidade de dados tem se mantido como uma questão delicada de projeto e a razão pela qual a granularidade possui um papel de destaque é porque ela afeta o volume de dados que residem no DW e ao mesmo tempo, afeta o tipo de consulta que pode ser atendida. O volume de dados contidos no DW é balanceado de acordo com o nível de detalhe de uma consulta, um alto nível de granularidade consiste em um modo muito mais eficiente de representar os dados do que uma representação que utilize um nível baixo de granularidade.

No entanto, à medida que o nível de granularidade se eleva, há uma correspondente diminuição da possibilidade de utilização dos dados para atender a

consultas, isto é, com um nível muito baixo de granularidade, é possível responder a qualquer consulta. E, com um alto nível de granularidade, o número de questões a que os dados podem satisfazer é limitado. Porém, a procura por um único detalhe é um evento muito usado em DW.

Portanto, a definição do nível de granularidade é uma questão importante de projeto. Quando a granularidade de um DW é apropriadamente estabelecida, os demais aspectos de projeto e implementação fluem normalmente, porém quando ela não é estabelecida, todos os demais aspectos se complicam.

Uma outra importante questão do projeto dos dados contidos no DW (depois da granularidade) é o particionamento. O particionamento de dados diz respeito à repartição dos dados em unidades físicas separadas que podem ser tratadas independentemente, proporcionando muito mais flexibilidade no gerenciamento dos dados. Além disso, toda unidade de dados pertence a uma e somente uma partição. Quando os dados residem em unidades físicas de tamanho menor, eles podem ser reestruturados facilmente, indexados livremente, pesquisados seqüencialmente (se necessário), reorganizados, recuperados e monitorados facilmente, conseqüentemente possuem uma flexibilidade maior com relação ao acesso aos dados.

No ambiente de DW, as questões referentes ao particionamento de dados não enfocam a necessidade do particionamento ser feito ou não, mas como ele deve ser feito. Os critérios pelos quais é possível dividir dados, normalmente são feitos em função de datas, de área de negócio, de área geográfica, etc.

Uma das questões mais desafiadoras para o desenvolvedor do DW é a dúvida entre efetuar o particionamento no nível de sistema ou no nível de aplicação. O particionamento no nível de sistema é, até certo ponto, função do SGBD e do sistema operacional. O particionamento no nível de aplicação é feito pelo código da aplicação e é, única e exclusivamente, controlado pelo desenvolvedor.

Quando o particionamento é feito no nível de sistema, o SGBD inevitavelmente pressupõe uma definição de dados única. Quando o particionamento de dados é feito no nível de aplicação, o SGBD e o sistema operacional não tomam conhecimento de qualquer relação existente entre uma partição e outra. Como regra, faz sentido particionar os dados do DW no nível de aplicação, onde a razão mais importante para esse fato é que,

no nível de aplicação, é possível existir uma definição de dados diferente para cada ano. Pode existir a definição dos dados de 1998 e a definição dos dados de 1999, e elas podem ser iguais ou não. Os dados existentes no DW são caracterizados como um conjunto de dados que abrange um longo período de tempo.

II.2.1.2. Projeto de Data Warehouse

O projeto de DW começa pelos dados operacionais dos sistemas já existentes. Existem dois pontos relacionados com a construção do DW- o projeto da interface com os sistemas operacionais e o projeto do DW propriamente dito. De certa forma, projeto não é uma descrição exata do que acontece durante a construção do DW, uma vez que ele é construído de modo heurístico. Inicialmente são colocados alguns dados no DW, depois tais dados são alterados e/ou outros dados são adicionados ao DW, de acordo com o resultado da análise feita pelo usuário final. Essa análise acompanha toda a vida do DW, pois é um engano pensar que os enfoques de projeto que funcionaram no passado serão úteis na construção do DW. Os requisitos para a criação do DW não são conhecidos até que os dados estejam armazenados e usados. Portanto, ele não pode ser projetado do mesmo modo que são construídos os sistemas clássicos baseados em requisitos. Por outro lado, também constitui um engano pensar que não prever requisitos seja uma boa idéia. A realidade se encontra em algum ponto intermediário.

A idéia de que a construção de um DW consiste meramente em um processo de extração tem como erro básico o fato de que os dados existentes no ambiente operacional não são integrados. A figura II.2 mostra a representação da migração de dados do ambiente de sistemas já existentes para o DW, onde podemos verificar que várias aplicações irão contribuir para o DW, consequentemente existe uma falta de integração comum ao ambiente dos sistemas existentes.

Cada aplicação (poupança, empréstimos e seguros) possuía seu conjunto particular de requisitos e, durante o processo de desenvolvimento, as demais aplicações não eram consideradas, com isso, alguns dos mesmos dados existam em vários lugares com nomes diferentes. Tentar extrair dados dos diversos lugares em que eles existem é

um problema muito complexo, como mostra a figura II.3, pois existem mil detalhes a observar na programação para a retirada dos dados do ambiente operacional de forma correta.

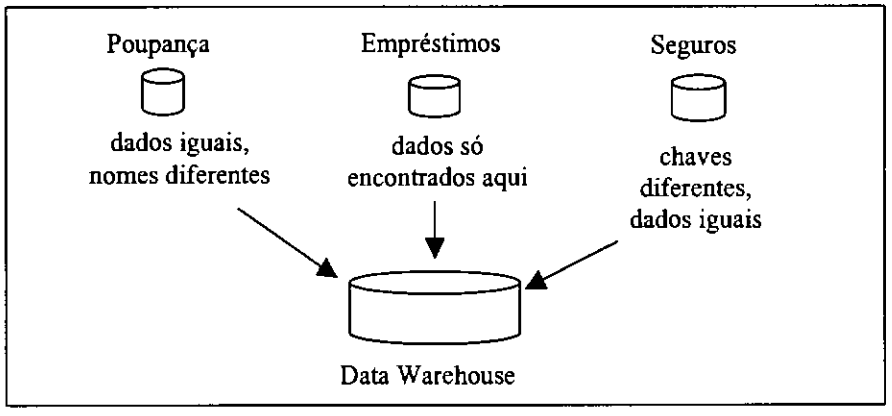


Figura II.2 – Representação da migração dos dados operacionais para o Data

Os exemplos da figura II.3 apenas dão uma leve idéia da questão da integração e não são, em si, complexos, mas quando são multiplicados pelos milhares de arquivos e sistemas existentes, a questão da integração se torna extremamente complexa e onerosa.

Entretanto, a falta de integração dos sistemas existentes não é a única dificuldade na passagem de dados do ambiente dos sistemas operacionais existentes para o ambiente de DW, há um outro problema importante que diz respeito ao acesso eficiente aos dados dos sistemas existentes. A carga dessa grande quantidade de dados é feita através do carregamento de dados históricos, que não é realizado com freqüência, do carregamento de dados de valor corrente no ambiente operacional, que é feito apenas uma vez e do carregamento de dados durante o processamento normal - enquanto são efetuadas alterações sobre o ambiente operacional, não é fácil realizar o rastreamento eficiente e o tratamento dessas alterações.

Existem diversas técnicas para limitar a quantidade de dados pesquisados, como por exemplo, a técnica que consiste em pesquisar dados que apresentem marcas de tempo. Quando uma aplicação registra o momento da última alteração ou atualização em

um registro, a procura no DW pode ser executada de forma bem eficiente, porque os dados que apresentarem datas diferentes das procuradas não precisarão ser tocados. Uma outra técnica consiste em pesquisar em um arquivo de auditoria ou de log. Porém, muitas vezes o departamento de operação protege os arquivos de log porque eles são necessários aos processos de recuperação e o departamento de operação não tem grande interesse em que seu arquivo de log seja utilizado com outros propósitos.

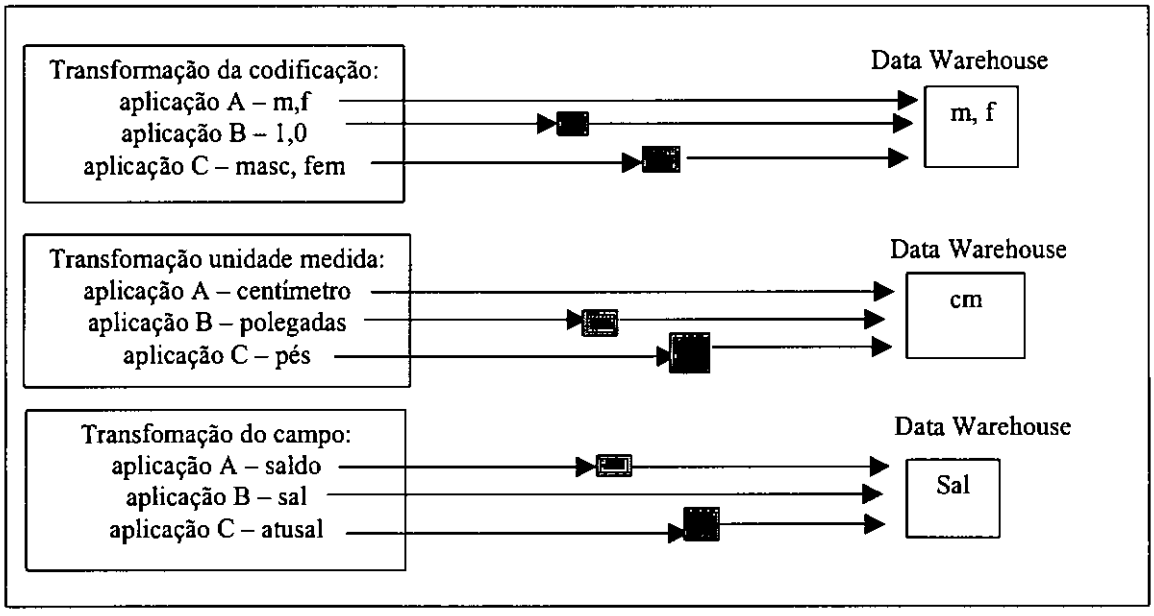


Figura II.3 – Exemplos da passagem de dados do ambiente de sistemas existentes para ambiente de DW

Porém, integração e performance não são as únicas discrepâncias fundamentais que impedem que o processo de extração simples seja utilizado para construir o DW. Uma terceira importante dificuldade está relacionada à alteração dos parâmetros de tempo, onde os dados operacionais existentes são, quase sempre, dados de valor corrente, cuja exatidão é válida para o momento do acesso e que podem ser atualizados. Esses dados devem apresentar um elemento de tempo vinculado. Portanto, à medida que os dados passam do ambiente de sistemas operacionais existentes para o ambiente do DW, é necessário que seja feita uma alteração significativa sobre eles.

Outra importante observação a ser feita durante a passagem dos dados do ambiente de sistemas operacionais existentes para o ambiente do DW é referente à

necessidade de gerenciar o volume de dados. É preciso efetuar a compactação, pois senão o DW logo ficará grande demais para ser controlado, portanto a compactação dos dados deve ser iniciada no momento da sua extração.

II.2.1.3. Modelo de Dados para ambiente de Data Warehouse

Segundo Inmon [73], existe três níveis de modelagem de dados: a modelagem de alto nível (diagrama de entidade-relacionamento - DER), a modelagem de nível intermediário (conjunto de itens de dados – Data Item Set, DIS) e a modelagem de baixo nível (modelo físico).

O diagrama de entidade-relacionamento apresenta as entidades e os relacionamentos. Essa representação encontra-se no nível mais alto de abstração, pertencendo ao escopo do modelo conhecido também como de escopo de integração, que estabelece as fronteiras do modelo de dados e deve ser definido antes do início do processo de modelagem. O escopo é combinado entre a equipe da modelagem e o usuário final do sistema. Caso o escopo não seja predeterminado, há uma grande possibilidade do processo de modelagem prolongar-se por um tempo excessivo

Depois de criado o modelo de dados de alto nível, é definido o modelo de nível intermediário, (conjunto de itens de dados - Data Item Set , DIS), como mostra a figura II.4. Para cada área de interesse, ou entidade identificada no modelo de dados de alto nível é criado um modelo de nível intermediário. O modelo de dados de alto nível identificou três entidades (áreas de interesse). Cada uma das áreas de interesse , ou entidade identificada no modelo de alto nível será, posteriormente, desenvolvida em seu próprio modelo de nível intermediário e depois é ampliado com mais informações. Há quatro elementos básicos referentes ao nível intermediário, em que o primeiro é o agrupamento primário de dados, que engloba os atributos que constam uma única vez em cada área de interesse. O segundo elemento é o agrupamento secundário de dados que engloba os atributos que podem aparecer várias vezes na mesma área de interesse. O

terceiro é o conector que relaciona os dados de um grupo aos do outro. E finalmente, o quarto elemento é o tipo de dados.

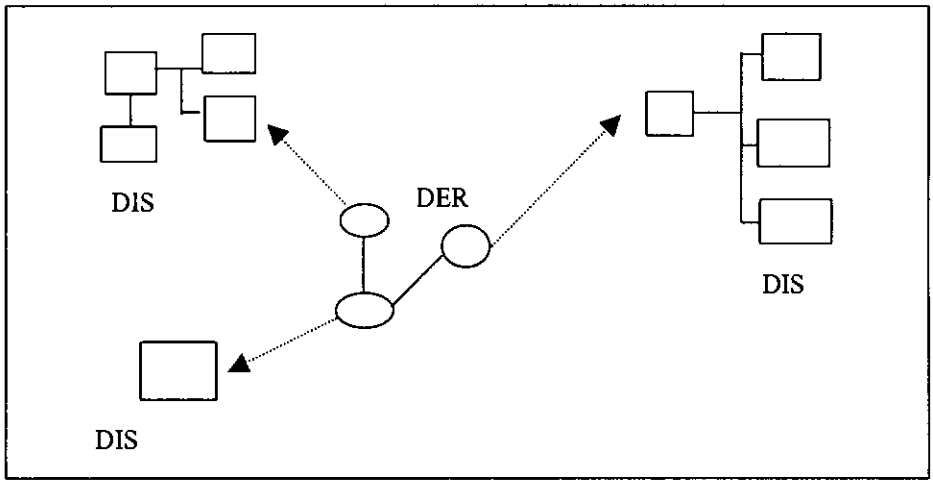


Figura II.4 – Relacionamento entre o DER e o DIS

O modelo físico é criado a partir do modelo de nível intermediário mediante a sua simples expansão de forma que passe a representar chaves, características físicas (como por exemplo, E/S - entrada/saída - física) e a característica de performance (granularidade e particionamento).

Um outro modelo considerado é o modelo dimensional. É uma metodologia de projeto top-down que lista dimensões e fatos relevantes a cada processo de negócio [79]. É também conhecido como "star join schema" (modelo estrela) porque o diagrama é semelhante a uma estrela com uma tabela grande no centro cercada por tabelas auxiliares exibidas em um padrão radial, conforme mostra o exemplo da figura II.5.

O modelo dimensional é muito assimétrico, há uma tabela dominante no centro do diagrama com múltiplas junções conectando-as às outras tabelas. Cada uma das tabelas secundárias possui apenas uma junção com a tabela central. A tabela central é denominada de tabela de fatos e as outras tabelas de tabelas de dimensão. A tabela de fatos armazena medições numéricas do negócio e as tabelas dimensionais armazenam as descrições textuais das dimensões do negócio.

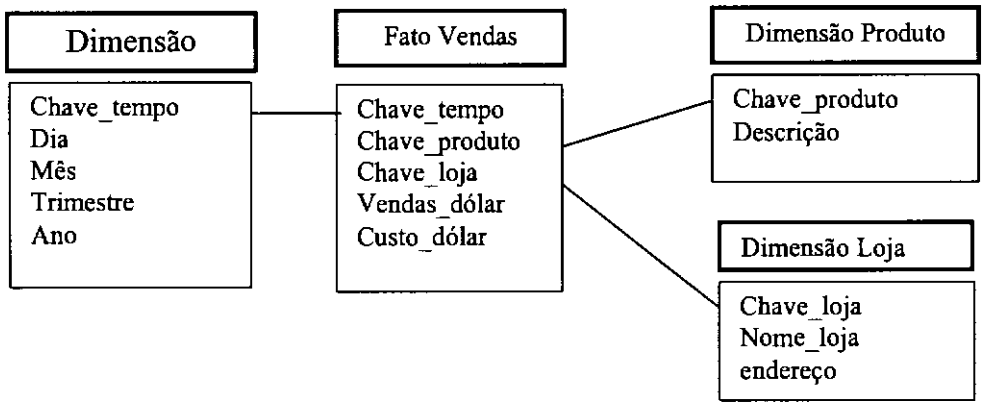


Figura II.5 – Modelo Dimensional

II.2.1.4. Normalização/Desnormalização

O resultado do processo de construção do modelo de dados consiste em uma série de tabelas, cada uma das quais contendo chaves e atributos. Esse processo normalmente resulta em muitas tabelas, cada uma contendo apenas um mínimo de dados, ocorrendo um problema de performance, pois surge um esforço enorme para interligar as tabelas dinamicamente, onde são consumidos recursos de E/S, tanto em termos de acesso aos dados, como em termos de acesso ao índice para a localização dos dados. Um enfoque mais racional consiste em intercalar as tabelas de forma que apenas um mínimo de E/S seja consumido para demandar menos E/S para realizar a mesma tarefa.

Todavia, a intercalação de tabelas é apenas uma das técnicas de projeto que podem poupar E/S. Outra técnica muito útil é a da criação de um array de dados, quando existe um número estável de ocorrências, em que os dados são acessados em seqüência, em que os dados são criados e/ou atualizados em uma seqüência estatisticamente padronizada, e assim por diante.

No DW, os dados são baseados em parâmetros de tempo, sempre apresentam relevância em relação a um determinado momento, e unidades de tempo ocorrem com grande regularidade. Em um DW, a criação de um array por mês, por exemplo, é algo muito natural.

Outra técnica importante de projeto relevante para o ambiente de DW consiste na introdução intencional de dados redundantes. Dessa maneira, o acesso aos dados é mais eficiente, mas a atualização destes não é otimizada. Porém, como os dados são estáveis, portanto existe poucas razões para a preocupação com a atualização.

Uma outra técnica útil é a separação de dados que apresentem probabilidades de acesso muito diferentes. Para tornar a atividade de E/S mais eficiente e armazenar os dados de modo mais compacto, faz sentido transformar a tabela em duas tabelas separadas.

Uma outra técnica de projeto é o gerenciamento da integridade referencial no ambiente de DW. Com relação ao volume de dados existente em um DW, o fato de o DW não ser atualizado, e o fato de o DW representar dados ao longo do tempo, e os relacionamentos não permanecerem iguais com a passagem do tempo, é necessário a utilização de um enfoque diferente para o tratamento da integridade referencial no ambiente de DW.

II.2.1.5. Metadados

Um importante aspecto do ambiente de DW diz respeito aos metadados, que na realidade são definidos como dados sobre dados. Os metadados fazem parte do meio de processamento de informações há tanto tempo quanto os programas e os dados. Contudo, é no mundo dos DW que os metadados assumem um novo nível de importância. E por meio dos metadados que a utilização mais produtiva do DW é alcançada.

Os metadados englobam o DW e mantêm informações sobre o que está aonde no DW. Praticamente, os aspectos sobre os quais os metadados mantêm informações são:

- a estrutura dos dados segundo a visão do programador,
- a estrutura dos dados segundo a visão do analista.
- a fonte de dados que alimenta o DW.
- a transformação sofrida pelos dados no momento de sua migração para o DW.
- o modelo de dados.

- o relacionamento entre o modelo de dados e o DW.
- o histórico de extrações.

II.2.2. Principais aspectos da transformação e integração dos dados

Um dos maiores problemas em ambientes de DW aparece com falta de integração de dados, ou seja, quando simplesmente tenta-se mover os dados de bancos de dados operacionais para um ambiente de DW, sem se preocupar com a sua integração. E uma outra questão está relacionada com o uso de técnicas não apropriadas para gerenciar múltiplos níveis de dados sumarizados. Dados sumarizados e armazenados para satisfazer uma necessidade específica, depois de utilizados geralmente são esquecidos e dificilmente serão usados novamente ou se forem usados, correm o risco de serem utilizados para o propósito errado porque o seu conteúdo possivelmente será desconhecido. Portanto, um ambiente de DW requer recursos de integração de dados e de conteúdo consistente.

Algumas das principais funcionalidades necessárias para ocorrer a passagem do ambiente operacional para o ambiente de DW são:

- a seleção de dados do ambiente operacional pode ser muito complexa. Para qualificar um registro para o processamento de extração, diversas pesquisas a outros registros existentes em uma variedade de arquivos devem ser efetuadas, demandando leituras indexadas, lógica de conexão e assim por diante.
- as chaves de entrada operacionais geralmente precisam ser reestruturadas antes de serem gravadas. Muito raramente uma chave de entrada permanece inalterada ao ser lida no ambiente operacional e gravada no ambiente de DW. Em casos simples, um elemento de tempo é acrescentado à estrutura de chave. Em casos complexos, toda a chave de entrada precisa ser reestruturada.
- os dados são reformatados. Como exemplo simples, dados de entrada sobre data são lidos como AA/MM/DD e são gravados no arquivo de saída como DD/MM/AA.

- Os dados passam por uma limpeza. Em alguns casos, um algoritmo simples é aplicado aos dados de entrada para corrigí-los. Em situações complexas, utiliza-se sub-rotinas de inteligência artificial para limpar os dados de entrada deixando-os em uma forma de saída aceitável.
- Quando há vários arquivos de entrada, a escolha das chaves deve ser feita antes que os arquivos sejam intercalados. Isso significa que se diferentes estruturas de chave são usadas nos diferentes arquivos de entrada, então a lógica que permite a escolha deve estar embutida no programa de intercalação.
- Pode haver vários resultados. Dados podem ser produzidos em diferentes níveis de resumo pelo mesmo programa de criação do DW.
- A alteração de nomes de elementos de dados durante a passagem do ambiente operacional para o ambiente de DW deve ser registrada. Quando um elemento de dados é movido, seu nome, geralmente, é alterado. A documentação dessa mudança precisa ser feita.
- A conversão do formato dos dados precisa ser feita. A conversão de EBCDIC para ASCII (ou vice-versa) precisa ocorrer.
- Volumes massivos de entrada devem ser levados em consideração. Quando há somente uma pequena quantidade de dados de entrada, diversas opções de projeto podem ser conciliadas. Mas quando há muitos registros de entrada, opções especiais de projeto (como cargas em paralelo ou leituras em paralelo) talvez tenham que ser empregadas.
- O DW espelha as informações históricas necessárias, ao passo que o ambiente operacional focaliza as informações correntes.

II.2.3. Tecnologia do Data Warehouse

O DW demanda um conjunto de características tecnológicas bastante simples. No DW não há atualização on-line; existe apenas exigências mínimas de bloqueio; uma

interface de teleprocessamento muito básica e assim por diante. Todavia, existe alguns requisitos tecnológicos relacionados ao DW, que são:

1) A capacidade de gerenciar grandes quantidades de dados de forma eficiente e econômica
2) A capacidade de gerenciar diversos meios de armazenamento, como por exemplo, sobre velocidade de acesso
3) Ter condições de indexar dados, como por exemplo, usar índice secundário, temporários, etc. E de monitorar dados, como por exemplo, determinar se uma reorganização precisa ser feita, determinar o espaço disponível, determinar se há grande quantidade de dados em overflow, etc.
4) Proporcionar interface, tanto ao recebimento de dados de quanto à passagem de dados para uma ampla variedade de tecnologias, como por exemplo, capacidade de funcionar em modo batch
5) Permitir que o projetista/desenvolvedor posicione os dados fisicamente, no nível de bloco/página, de forma otimizada
6) Gerenciar dados em paralelo
7) Possuir controle contínuo sobre os metadados. O DW funciona sob um ciclo de vida do desenvolvimento heurístico e iterativo. Para ser eficaz, o usuário do DW precisa ter acesso a metadados exatos e atualizados. Os metadados possuem estruturas das tabelas do DW, atributos das tabelas do DW, fonte de dados do DW (o sistema de registro), correspondência entre o sistema de registro e o DW, especificação do modelo de dados, logging de extrações e rotinas comuns de acesso a dados
8) Dispor de uma interface de linguagem rica para o DW, que deve possuir condições de acessar um conjunto de dados por vez, condições de acessar um registro de dados por vez, garantir especificamente que um ou mais índices serão usados para atender a uma chamada, dispor de uma interface SQL e condições de inserir, excluir ou atualizar dados.
9) Ter condições de fazer a carga eficiente do DW. Existem diversos métodos de carga com um registro por vez, por meio de uma interface de linguagem, ou em massa com ajuda de um utilitário.

10) Usar índice eficientemente, através da utilização de mapas de bits, do armazenamento de todo ou partes de um índice na memória principal, da compactação das entradas de índices quando a ordem dos dados que estiverem sendo indexados permitir tal compactação ou da criação de índices seletivos
11) Ter condições de armazenar dados compactados, possibilitando a utilização mínima de espaço
12) Suportar chaves compostas
13) Gerenciar efecientemente dados de tamanho variável
14) Ter condições de ativar e desativar o gerenciador de bloqueios quando for desejável. Oferecer condições para o controle explícito do gerenciador de bloqueio no nível do programador
15) Ter capacidade de efetuar processamento somente por índices
16) Ter capacidade de restaurar dados rápida e completamente a partir de um meio de armazenamento em massa
17) Outras características tecnológicas são: integridade de transações, uso de buffers de alta velocidade, bloqueio em nível de página, etc.

II.2.4. Tipos de SGBDS e o Data Warehouse

A espécie de processamento que ocorre em um ambiente de DW pode ser caracterizada como processamento de carga e acesso. Os dados são integrados, transformados e carregados para o DW a partir do ambiente operacional. Uma vez no DW, os dados são acessados e analisados. Depois que os dados são carregados, normalmente não é feita atualização no DW. Caso seja necessário efetuar ajustes ou correções nesse ambiente, isso é feito em horários livres, quando nenhuma análise está sendo feita sobre os dados do DW. Os DWs gerenciam grandes quantidades de dados porque eles possuem detalhes granulares ou atômicos, informações históricas e dados resumidos bem como dados detalhados. Como no ambiente de DW não ocorre atualização e é preciso otimizar o acesso aos dados, então há uma necessidade de

existirem muitos índices. Aliás, em DW é possível empregar uma estrutura de indexação muito robusta e sofisticada.

II.2.5. SGBD Multidimensional e o Data Warehouse

Uma das tecnologias discutidas, freqüentemente, no contexto do DW é a de sistemas de gerenciamento de bancos de dados multidimensionais. Sistemas de gerenciamento de bancos de dados multidimensionais proporcionam uma estrutura de sistemas de informação que permite o acesso flexível aos dados. O SGBD multidimensional oferece tanto flexibilidade quanto controle ao usuário final.

Há uma tendência em pensar que a tecnologia de SGBD multidimensional pode ser a tecnologia de bancos de dados para o DW. Porém, trata-se de num engano. Existe diferenças entre o SGBD multidimensional e o DW. O DW mantém quantidades massivas de dados; o SGBD multidimensional mantém pelo menos uma ordem de magnitude de dados a menos. O DW é direcionado para uma quantidade limitada de acesso flexível; o SGBD multidimensional é direcionado para acesso e análise muito pesados e imprevisíveis. O DW contém dados que apresentam um horizonte de tempo muito longo, de cinco a dez anos; o SGBD multidimensional mantém um horizonte de tempo de dados bem mais curto, doze a quinze meses. O DW permite que analistas acessem de forma limitada os dados nele contido; o SGBD multidimensional permite acesso ilimitado.

II.2.6. Operação de atualização do Data Warehouse

Uma vez que o DW é construído, a atenção passa da construção deste para suas operações diárias. O custo de operação e manutenção de um DW é bastante alto. O volume de dados contidos no DW aumenta rapidamente, mais do que o previsto. A utilização imprevisível e difusa do DW acarreta disputa no servidor que gerencia o DW.

Todavia, a maior despesa inesperada associada à operação do DW é a de atualização periódica dos dados herdados contidos no DW.

O primeiro passo para atualizar os dados do DW consiste na leitura direta dos bancos de dados herdados. Quando os dados precisam ser lidos em diferentes fontes herdadas para compor uma unidade a ser passada para o DW, a leitura direta dos dados herdados é praticamente a única opção para a renovação do DW. Ou, quando uma transação tem causado a atualização simultânea de diversos arquivos herdados, uma leitura direta dos dados herdados pode ser a única maneira de efetuar a atualização. Mas as leituras diretas e repetidas dos dados herdados constituem um método muito dispendioso, pois o SGBD herdado deve estar on-line e ativo durante o processo de leitura. E o outro fator é que os mesmos dados herdados são desnecessariamente passados várias vezes, aproximadamente num processo de 100% de um arquivo herdado, quando somente 1 ou 2% do arquivo herdado são efetivamente necessários. Portanto, esta estratégia apresenta utilidade e aplicabilidade limitadas.

Existem duas técnicas básicas para a captura de dados à medida que ocorre atualização no ambiente operacional herdado. Uma das técnicas é chamada de replicação de dados e a outra técnica é chamada de captura de dados alterados, onde as alterações que tiverem ocorrido são extraídas das fitas de log ou de diário que tiverem sido geradas durante o processo de atualização on-line.

A replicação requer que os dados a serem capturados sejam identificados antes da atualização. Depois, à medida que a atualização transcorre, os dados são capturados. É estabelecido um "gatilho" que faz com que a atividade de atualização seja capturada. Uma das vantagens da replicação é que o processo de captura pode ser controlado seletivamente. Apenas os dados efetivamente necessários são capturados. Uma outra vantagem da replicação é que o formato dos dados é "limpo" e bem definido. O conteúdo e a estrutura dos dados capturados são bem documentados e de fácil entendimento pelo programador. Como desvantagens da replicação citam-se os fatos de que E/S adicional resulta da captura dos dados e de que o sistema requer constante atenção às definições de parâmetros que controlam a captura no momento da atualização, em decorrência da natureza instável e mutável do DW. O total de E/S necessário, em geral, não é insignificante. Além do mais, a E/S que é consumida é extraída do meio do dia de alta

performance, no momento em que o sistema tem menos condições de atender a tal demanda.

O segundo método de atualização eficiente do ambiente de DW é por meio do que é chamado de "captura de dados alterados" (CDC - changed data capture). O CDC usa um arquivo contendo as operações realizadas como meio para capturar e identificar as alterações que tenham ocorrido durante o dia on-line.

O principal obstáculo existente no CDC encontra-se na leitura e compreensão desse arquivo. Porém, uma vez que esse obstáculo seja ultrapassado, há alguns benefícios muito atraentes na sua utilização para a renovação do DW. Ao contrário do que ocorre com a replicação, este processamento não requer E/S adicional, pois esse arquivo é gravado independentemente de vir, ou não, a ser usado para a renovação do DW. A segunda vantagem do CDC é que todo o processo de atualização é capturado, não sendo necessário redefinir parâmetros quando uma alteração é feita no DW.

II.3. Metadados

Em qualquer cenário do DW os metadados assumem um papel de grande destaque. Diante do armazenamento e gerenciamento dos dados discrepantes (dados heterogêneos, isto é, diferentes em tipo, qualidade, etc) os metadados possuem uma dimensão importante, porque é por meio deles que esses dados são registrados, acessados e controlados no ambiente de DW. Dessa forma, quando dados discrepantes estão envolvidos, a criação e manutenção adequada dos metadados são mais do que essencial para a operação do DW.

O problema real mais complexo da administração de dados é sobreviver a dados discrepantes, enquanto desenvolvem ambientes para a integração dos dados. O grande desafio é resolver quatro problemas básicos com dados discrepantes, que são: ter consciência dos dados existentes, entender os dados existentes, dados com alta

variabilidade e a existência de redundância de dados. Solucionando estes quatro problemas básicos ocorrerá sem dúvida uma redução do desenvolvimento contínuo de dados discrepantes, assim esse dados serão usados até que eles possam ser transformados em um ambiente de dados integrados.

II.3.1. Conceitos e métodos para construir um ambiente integrado de dados

II.3.1.1. Refinamento e transformação de dados

Refinamento de dados é o primeiro passo para o desenvolvimento de um ambiente de integração dentro de uma arquitetura comum. É um processo semântico para compreender o conteúdo e significado dos dados discrepantes sem alterar esses dados ou comprometer a operação do sistema de informação.

Para entender corretamente e gerenciar os dados discrepantes é preciso fazer o cruzamento de referência entre eles em uma arquitetura comum de dados, cujas redundâncias e variabilidades são removidas. Este processo é conhecido como refinamento dos dados. É uma fase do processo onde começa o desenvolvimento de integração dos dados em um ambiente integrado de dados como mostra a figura II.6.

Quando o cruzamento de referência de dados está completo, então a variabilidade de chaves primárias, características dos dados e código dos dados estão avaliadas, a integridade e precisão dos dados estão estabelecidas. Finalmente, dados são projetados para a preparação da integração físicas dos dados discrepantes.

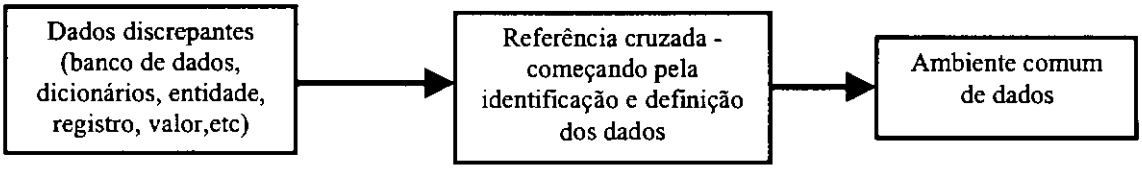


Figura II.6 – Refinamento dos dados

Dados discrepantes precisam ser formalmente transformados antes de fazerem parte do ambiente de dados integrados. Transformação de dados é um processo formal de dentro de uma arquitetura comum de dados, e não é uma tarefa trivial. Transformação de dados está baseado em uma perspectiva do ambiente de dados que podem existir em um ambiente de dados discrepante ou em um ambiente de dados integrado.

O processo de transformação de dados começa identificando e modelando o banco de dados. Depois os dados fonte necessários para desenvolver os dados são identificados. Este passo requer o processo de refinamento de dados se os dados fonte forem dados discrepantes. Os dados requeridos são então extraídos e colocados em um banco de dados provisório.

Qualquer dado que foi armazenado como valores de dados individuais é reconstruído em registros de dados. Depois os dados são traduzidos do banco de dados provisório para variações de dados, que também requerem o processo de refinamento de dados. Os dados são traduzidos do banco de dados provisório, são remodelados e reestruturados para satisfazer a exigência do ambiente de dados.

Finalmente, os dados transformados estão armazenados no banco de dados e revisados para assegurar que eles estão completos e corretos como esperado. Se os dados não estiverem completos e corretos, então os passos de transformação de dados são ajustados e os dados são retransformados. Este processo continua até os dados transformados se encontrarem completos e corretos.

II.3.1.2. Características do ambiente integrado de dados

Um ambiente de dados completamente integrado é desenvolvido através da engenharia de dados onde o seu principal objetivo é projetar, construir e manter uma

biblioteca de ambiente de dados para dar suporte à engenharia de informação e ao desenvolvimento de sistemas de informação como mostra a figura II.7.

É um processo estruturado que desenvolve o ambiente formal de dados e o ambiente integrado de dados em uma biblioteca de ambiente de dados. O ambiente formal de dados começa com o processo de entender e transformar dados discrepantes, transportando-os para um ambiente de dados completamente integrado, resolvendo dessa forma as suas variabilidades e redundâncias. Os dados discrepantes representam dados que foram identificados e que foram compreendidos dentro de um contexto comum.

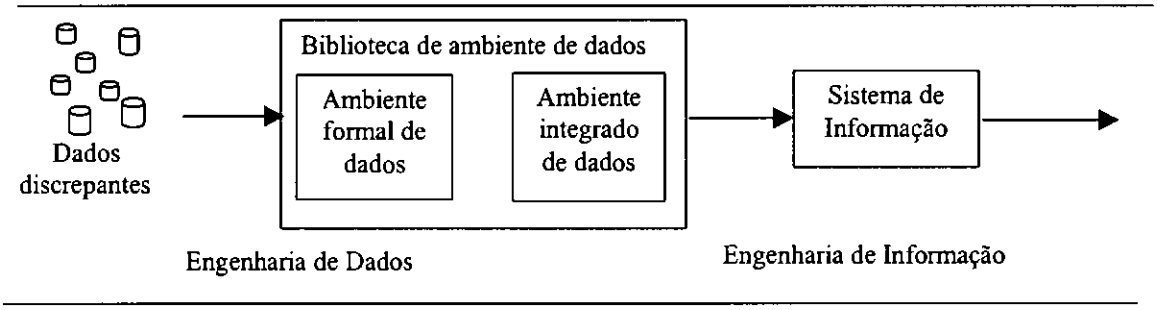


Figura II.7 – Engenharia de Dados

O desenvolvimento de um ambiente integrado de dados dentro de um contexto comum requer uma arquitetura de dados que engloba o conceito de uma infra-estrutura para a tecnologia de informação, a organização de um ambiente de dados dentro dessa infra-estrutura e a definição de uma arquitetura formal de dados. Uma arquitetura formal de dados compreende o conteúdo e o significado do dado, a estrutura do dado, a identificação e ajuste do dado e a definição do metadado.

Uma infra-estrutura para a tecnologia de informação descreve o relacionamento entre o ambiente de dados, a plataforma, as atividades de negócio e sistemas de informação. O ambiente de dados representa todos os dados disponíveis numa organização, já a plataforma corresponde ao hardware e sistema de software da organização. Os outros dois componentes como a atividade de negócio, representa toda a atividade da organização e o sistema de informação representa a implementação da atividade de negócio, usando o ambiente de dados e a plataforma.

Um fator importante está concentrado na descrição dos dados, pois compreender conteúdo e significado dos dados é fundamental para o efetivo e eficiente desenvolvimento e uso do ambiente de dados integrado. O processo para estabelecer um nome formal do dado e para compreender a sua definição é relativamente fácil se forem seguidas algumas diretrizes. Definição compreensiva do dado reflete a definição do objeto do mundo real, onde explica o conteúdo e o significado do dado, mas não propriamente o uso do dado.

Dentro da arquitetura de dados, a estrutura de dados é a representação do arranjo, do relacionamento e do conteúdo do assunto, da entidade e do arquivo. Uma estrutura de dados inclui os dados lógicos de um projeto e os dados físicos de um banco de dados, como mostra a figura II.8.

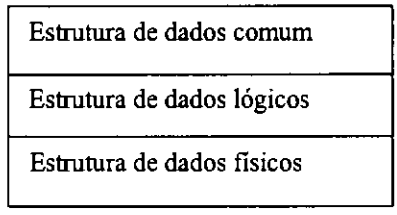


Figura II.8 –Níveis de estrutura de dados [17]

A estrutura de dados comum é uma estrutura de dados dentro de um modelo de dados comum que fornece toda compreensão de toda estrutura de dados discrepantes (definida como uma variedade de estrutura de dados diferentes que são baseadas em distintas perspectivas do mundo real, suportando aplicações que possuem diferentes necessidades e diferentes ambientes operacionais não possuindo a capacidade para todo o entendimento da representação de seu mundo) e múltiplas perspectivas do mundo real dessa estrutura de dados representada. Contém o universo do assunto e as características de todo projeto.

A estrutura de dados lógica é a estrutura de dados do modelo lógico. Ela representa um subconjunto de estrutura de dados comum que suportam um sistema de informação e representam normalmente uma perspectiva do mundo real. Ela contém entidades e atributos de dados necessários num sistema de informação.

A estrutura de dados físico é a estrutura de dados num modelo físico. Representa a implementação da estrutura de dados lógica de um sistema de informação num ambiente operacional específico. Ela contém arquivos de dados e itens de dados.

A qualidade de dados consiste em integridade de dados, precisão de dados e perfeição de dados. Integridade de dados trabalha com a definição formal de regras e aplicações consistentes dessas regras para garantir uma alta integridade de dados. Compreende técnicas que definem e mantém os valores corretos dos dados (como por exemplo, domínio dos dados, valores default dos dados, etc.), a integridade da própria estrutura de dados (como por exemplo, a cardinalidade, a integridade referencial, etc.), a retenção apropriada da integridade dos dados (como por exemplo, regras específicas de prevenção de perda de dados críticos através de atualizações ou exclusões) e a integridade dos dados derivados (como por exemplo, controle da redundância de dados).

Precisão de dados é o segundo componente de qualidade dos dados. Trabalha com a representação dos dados no mundo real e possui escopo muito extenso em relação às medidas para garantir a confiança nos dados.

Perfeição de dados é o terceiro componente de qualidade dos dados. É o processo para assegurar que todos os dados necessários estão disponíveis no ambiente dos dados para a demanda da informação.

Um aspecto importante de qualidade de dados é o aproveitamento constante dos dados através de análise de dados e da volatilidade de dados e ajuste da coleção de dados para a sua adequada representação do mundo real.

Atualmente a tendência esperada está baseada na construção e manutenção de um metadado, favorecendo totalmente a utilização de um integrado ambiente de dados. A falta de um bom metadado não só limita quanto a utilização do ambiente de dados como também conduz à coleção e criação de dados discrepantes que não são documentados corretamente permitindo assim um aumento de metados discrepantes.

Metadados discrepantes são dados que não são essencialmente semelhantes, ou seja, são diferentes em tipo, qualidade, ou caráter. Metadados normalmente não existem,

ou se existem, eles estão incompletos, obsoletos, mal definidos e de difícil entedimento. Geralmente estão espalhados por muitos documentos e com uma elevada dificuldade de serem localizados, ou então quando são localizados existe uma enorme dificuldade de entender o seu conteúdo, significado ou a qualidade dos dados, tornando difícil a sua integração com os dados.

II.3.2. Tipos de metadados

Hoje em dia a nova perspectiva sobre desenvolvimento e manutenção dos metadados incluem uma compreensão de diferentes tipos de metadados e o conceito de um metadado padrão. A definição atual usada numa arquitetura comum de dados é apresentada na figura II.9.

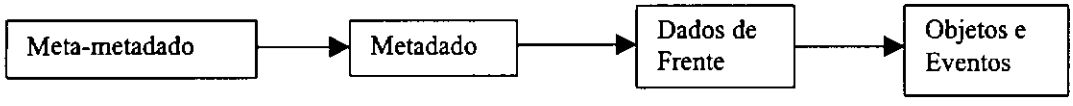


Figura II.9 – Tipos de metadados

Dados de frente são os dados que descrevem os objetos e eventos, normalmente são usados para localizar ou administrar objetos e eventos do real mundo, incluindo então a qualidade dos dados da representação desses objetos e eventos

- Exemplo: Maria Silva, Região Norte, 12/03/1991 e R$ 1.500,00 são os dados de frente; são os objetos e eventos do mundo real
- Exemplo: Somente pessoas empregadas pelo Estado e da Região Norte são dados de frente sobre a qualidade de dados representando objetos e eventos do mundo real

Metadados são os dados que descrevem os dados de frente, ou seja, fornecem as definições sobre os dados de frente.

- Exemplo: A definição do nome completo, aniversário e região são metadados descrevendo dados sobre os objetos e eventos.
-

Meta-metadados são os dados que descrevem o metadados, ou seja, oferecem um ambiente para desenvolver um metadado de alta qualidade.

- Exemplo: Definições de entidade de dados, característica de dados, variação da característica do dados e código do conjunto de dados.

II.3.3. Padronização do metadado

Muitas tentativas estão sendo feitas para desenvolver uma padronização comum de metadado, inclusive buscando o conhecimento de especialistas de dados para saber como os dados do mundo real devem ser gerenciados, partindo de um projeto lógico até o projeto físico. Diversas organizações vem desenvolvendo o seu próprio metadado, conhecido como padrão de dados, numa tentativa de solucionar a discrepância dos dados. Esses padrões geralmente tentam determinar como os dados devem ser projetados, construídos, denominados, estruturados e documentados. O problema com a padronização do metadado é que na maioria das vezes surgem nomes de dados que não são qualificados, dados que são abreviados de forma inconsistente e não possuem uma boa definição. Alguns metadados apresentam somente uma definição conceitual sem nenhum detalhe adicional e outros apresentam uma notação matemática que é de difícil aprendizado.

Segundo Brackett [17], uma das maneiras mais apropriadas para sua implementação e manutenção é realizada através do Metadado Warehouse, que contém um metadado comum e rotinas de pesquisa proporcionando um fácil entendimento com uma boa utilização do ambiente de dados, ou seja, toda utilização do ambiente dos dados. Representa um repositório para metadados comuns sobre dados tabulares e não tabulares, dados centralizados e distribuídos e dados automatizados e não automatizados. As rotinas de pesquisas são usadas para determinar quais dados estão disponíveis, o que eles representam, onde eles ficam situados e como eles podem ser acessados.

O objetivo fundamental do metadado é oferecer um manual do ambiente de dados, contendo as seguintes características:

- Um léxico de palavras comuns usadas para formar os nomes dos dados. Contém um conjunto de palavras abreviadas usadas para abreviar os nomes dos dados
- Uma arquitetura comum de dados para a descrição de dados, estrutura de dados e integridade de dados
- Uma arquitetura de dados contendo definições de vários esquemas de classificação
- Uma enciclopédia para identificar dados que existem dentro da arquitetura comum de dados
- Um glossário de palavras, termos e abreviações que suportam uso do abiente de dados
- Um inventário de dados mantido por várias organizações e referências cruzadas entre esses dados e a arquitetura comum de dados
- Um diretório de organizações que mantêm dados e contatos nessas organizações para obtenção de informação adicional sobre dados existentes
- Esquema de tradução de dados entre as variações de dados para apoiar dados compartilhados dentro e entre as organizações
- Um lugar disponível para conter fontes de dados, documentos inéditos e projetos que pertencem ao ambiente de dados
- Acesso rápido e amigável para o metadado

Um metadado pode ser ativo ou passivo. É dito passivo quando os novos dados que entrarem no ambiente de dados forem atualizados por operadores. É dito ativo quando novos dados entrarem no ambiente de dados e as suas atualizações acontecem de forma automática. O uso da abordagem passiva é melhor quando os dados discrepantes são identificados e compreendidos. A abordagem ativa é melhor quando um ambiente de dados integrado é desenvolvido e todos os dados são compreendidos e documentados dentro da arquitetura comum de dados.

II.4. Características dos dados em uma ambiente de DW

Um sistema de DW é normalmente um repositório consistente de dados históricos que podem ser facilmente acessados e manipulados para dar suporte à próxima geração de sistema de informação, sistema de tomada de decisão e processo analítico on-line (OLAP). Dados históricos são dados usados em processamento para avaliar tendências, projeções e alternativas. Podem ser dados tabulares ou não tabulares. São normalmente derivados de dados operacionais (dados usados no processamento operacional de transações diárias) e suportam diversos níveis de sumarização acima dos dados operacionais. Possuem as seguintes características:

a) Arquitetura dos dados – os dados precisam ser projetados, desenvolvidos e gerenciados dentro de uma arquitetura comum de dados, não pertencem a diferentes arquiteturas, onde é necessária uma notação para modelar e gerenciar a complexidade desses dados.
b) Múltipla dimensão dos dados - é a representação de um conjunto de objetos ou eventos do mundo real. É um dado que representa duas ou mais dimensões. On-line analytical processing (OLAP) para sistema de tomada de decisão é um processo de múltipla dimensão, também conhecido como análise de múltipla dimensão e é ainda referenciado como análise de dados estáticos porque os valores dos dados não sofrem atualização.
c) Perspectiva dos dados – é um conjunto de dados num ambiente de banco de dados histórico consistido de uma central de múltipla dimensão de dados cercada por vários dados de dimensão simples (dados operacionais) que formam o dado central ou o foco do dado.
d) Descrição dos dados – nome de dados formais e definições compreensivas dos dados são essenciais para a sua administração. Fornecem identificação única e completa informação sobre a existência, significado e localização dos dados históricos. Descrevem também a transformação e sumarização dos dados operacionais discrepantes para ambiente de dados integrados.

e) Estrutura de dados – contém toda a estrutura de dados dentro da arquitetura comum de dados, como por exemplo, chave primária, chave estrangeira, chave candidata, etc. A chave primária precisa ser significativa para a necessidade específica do assunto, incluindo uma ou mais características representando o tempo. A manutenção do componente estrutura de dados que estão sempre sendo atualizados, isto é, novas estruturas de dados estão sempre sendo definidas ou mesmo as já existentes estão em constante atualização e a sua documentação é ajustada adequadamente.

f) Integridade dos dados – contém regras importantes para manter a integridade de todos os dados numa arquitetura de dados comum, porém são difíceis de definir e documentar. A natureza estática dos valores dos dados, a natureza dinâmica das relações e os múltiplos níveis dos dados sumarizados requerem uma abordagem diferente para a integridade de dados. Esses dados são mais independentes de regras e geralmente possuem menos integridade referencial porque os dados não são atualizados e as relações são dinâmicas. Os dados não são totalmente normalizados, aumentando a capacidade de se fazer análise em múltiplas dimensões de dados sem muita navegação nas relações. Os dados são mais redundantes, mas o processamento é mais rápido.

   Os dados são sumarizados para obter uma visão generalizada sobre padrões e tendências, são desenvolvidos retirando as características das chaves primárias dos dados operacionais. Possuem um nível de sumarização que vai de uma leve sumarização para uma alta sumarização. A leve sumarização remove uma ou poucas característica da chave primária, conhecida como granularidade fina. E a alta sumarização remove muitas características dos dados da chave primária, conhecida como granularidade grossa.

g) Manutenção dos dados – são mantidos, apesar de seus dados não serem voláteis, através de constantes expansões de novos conjunto de dados com mais detalhes dentro do escopo existente e possíveis remoções de dados desnecessários podem ser ou não realizadas, como também constantes rederivações com novos dados que são adicionados para refletir a nova versão ou modificação.

II. 5. Processo de Knowledge Discovery in Database (KDD)

Em decorrência do aumento do volume de dados na área industrial, comercial, administrativa e outras áreas, existe atualmente a necessidade de se extrair conhecimentos dessa gama de informações. A técnica de KDD está voltada para estudar e analisar os grandes volumes de dados e depois extrair conhecimentos desse banco de dados para ser aplicada em tomadas de decisão, gerência de informação, processo de controle e muitas outras aplicações. As aplicações são desenvolvidas pelos programas que coletam, modificam, recuperam e excluem dados e por programas que usam, analisam, sumarizam, extraem ou manipulam os dados. A principal finalidade da técnica de KDD está na sua utilização para fazer a predição e/ou a descrição com uma aplicação de engenharia, onde a predição envolve campos do banco de dados para predizer conhecimentos, explorando informações ou valores de variáveis e a descrição se concentra em encontrar padrões interpretáveis descrevendo os dados.

O KDD é uma área multi-disciplinar [2] de pesquisa, que envolve: aprendizado de máquina, estatística, tecnologia de banco de dados, sistemas especialistas e visualizalização de dados (figura II.10). O sistema KDD é formado por métodos, algoritmos e técnicas para essas diversas áreas, tendo como principal objetivo extrair conhecimento dos dados no conteúdo dos grandes banco de dados.

Figura II.10 - KDD - área multi-disciplinar

II.5.1. Fases do processo de KDD

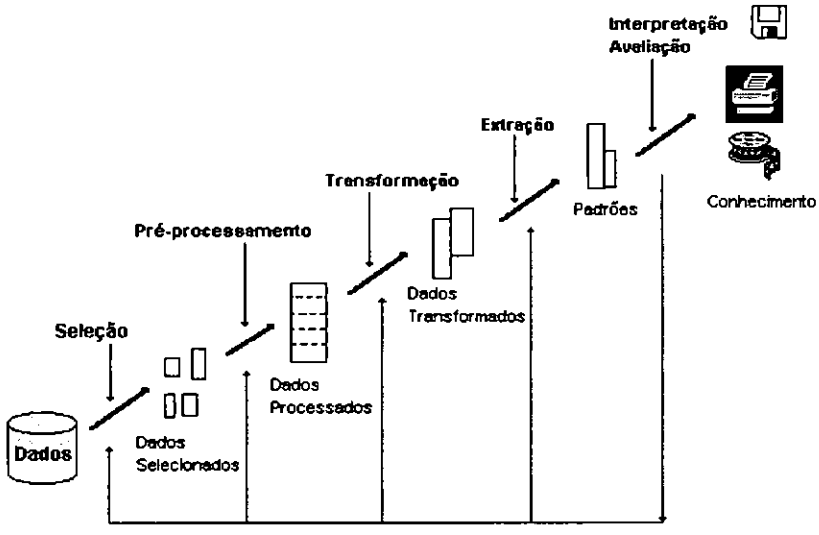


Figura II.11 – Passos do processo KDD segundo Fayyad [38]

O processo KDD (figura II.11) [38] é interativo e iterativo, envolvendo numerosos passos com muitas decisões sendo feitas pelo usuário. A metodologia apresentada não é linear, ou seja, quando se encontra na fase de pré-processamento ou na fase DM, pode-se chegar a conclusão que a fase de limpeza está incompleta, ou descobrir-se novos dados e usá-los para enriquecer o conjunto de dados já existente. Basicamente consiste de:

1 – Desenvolvimento de um domínio da aplicação, ou seja, a principal relevância e os objetivos do usuário final.

2 – Seleção de um conjunto de dados operacionais ou enfocar um subconjunto de variáveis ou exemplos de dados em que a performance pode ser encontrada.

3 - Limpeza dos dados e pré-processamento, ou seja, operações básicas tal como remoção de ruídos ou perda de consistência de domínio, colecionando a informação necessária para modelar, decidindo estratégias para manipular os dados perdidos, contabilizando informação do tempo, etc. O processo de limpeza de dados exige muita atenção no que se refere a remoção de registros duplicados, corrigindo erros gráficos, adicionando informação perdida, etc. Para implementar um processo KDD é preciso começar com um processo de refinamento de performance permanente e detalhamento dos dados. Os algoritmos de reconhecimento de padrões podem ser aplicados na limpeza dos dados. Enriquecimento dos dados e considerações de incertezas através da lógica fuzzy são algumas informações extras que certamente também poderão acontecer.

4 - Redução dos dados e projeção, ou seja, encontrar características para representar os dados dependendo do objetivo da tarefa. Usando redução ou método de transformação para diminuir o número de variáveis para encontrar representações dos dados.

5 - Escolhendo a tarefa Data Mining - nesta fase é preciso decidir se o objetivo do processo de KDD é de classificação, clustering, regressão, etc.

6 - Escolhendo o algoritmo de DM, ou seja, selecionando métodos para usar na pesquisa de padrão nos dados. Várias técnicas são usadas para diversas propostas: ferramenta de consulta, técnicas estatísticas, visualização, online analytical processing (OLAP), case_based learning (*k*-nearest neighbor), árvore de decisão, associação de regras, redes neurais, algorítmos genéticos

7 - Interpretação dos padrões, podendo também retornar aos passos 1-6 para futuras interações.

8 - Consolidação do conhecimento encontrado, ou seja, incorporação desse conhecimento ao sistema ou simplesmente documentá-lo em forma gráfica ou relatórios tornando o resultado do processo acessível.

II.6. Data Mining (DM)

O processo de KDD se refere a extração do conhecimento, ou seja, ao processo de encontrar e interpretar padrões dos dados, enquanto que DM refere a aplicação do algoritmo de extração de padrões dos dados dentro do processo KDD.

Data Mining é uma ferramenta que extrai padrões dos dados e aplica uma inferência de regras nos dados. Esses padrões e regras podem ser usados para fazer uma tomada de decisão e para prever o efeito dessas decisões, ou ainda fazer uma análise rápida enfatizando mais atenção nas informações mais importantes. DM permite que o usuário explore esses padrões com uma grande variação de possibilidades. Algumas técnicas de DM são apresentadas a seguir:

- Classificação é atualmente a mais comum e tem sido muito estudada pela comunidade de aprendizado de máquina. É uma função de aprendizado que mapeia um item de dados dentro de várias classes pré-definidas. Ela organiza padrões que descreve o grupo para qual um item pertence, fazendo um exame com os itens existentes que já foram classificados e inferindo um conjunto de regras. Classificação pode ajudar a descobrir características que levam a um modelo que pode ser usado para predizer quem eles são. Geralmente usadas em aplicações de diagnóstico médico, avaliação de risco de crédito e detecção de fraude.

- Clustering é uma tarefa comum de descrição onde busca identificar um conjunto finito de categorias ou "clusters" para descrever os dados. Está relacionado com classificação, mas difere porque os grupos ainda não estão definidos. É o processo de se agrupar objetos físicos ou abstratos em classes de objetos semelhantes. A análise de clustering ajuda encontrar diferentes grupos de objetos baseado na metodologia de dividir e conquistar, que decompõe um sistema de grande escala em componentes menores para simplificar o projeto e a implementação.

- Sumarização envolve métodos para encontrar uma descrição compacta para um subconjunto de dados. Um exemplo simples seria a tabulação do significado e desvio padrão para todos os campos. Técnicas de sumarização são freqüentemente aplicadas na análise exploratória e interativa dos dados ou na geração automática de relatórios.

- Modelar dependência consiste em encontrar um modelo que descreve dependências significativas entre variáveis. O modelo de dependência compreende dois níveis: o nível estrutural do modelo especifica quais são as variáveis localmente independentes umas das outras, e o nível quantitativo do modelo especifica a força das dependências utilizando uma escala numérica. Por exemplo, dependência probabilística usa independência condicional para especificar o aspecto estrutural do modelo e probabilidade ou correlações para especificar a força de dependência.

- Regras de associação acontecem quando as ocorrências estão juntas em um simples evento. O problema com regras de associação pode ser decomposto em duas partes:

  - Produzir todas as combinações de itens que uma transação suporta sobre um limite mínimo informado (conjunto de itens). Esta fase é considerada como a que consome maior tempo de tarefa do algoritmo.
  - Para todo conjunto de itens, gerar regras de associação para o transação do banco de dados, que é feito de uma maneira direta.

- Padrões seqüenciais se constitui numa metodologia que está sendo descoberta e incentivada pelas aplicações da indústria de vendas a varejo e pela área médica. Os dados de entrada são basicamente uma lista de transações seqüenciais e existe um tempo de transação associado com cada um deles. O objetivo primário é encontrar todos os padrões seqüenciais com suporte mínimo pré-definido, ou seja, o percentual de dados seqüenciais que contenha o padrão maior que um determinado

limite. Assim, algoritmos de padrões seqüenciais são usados para descobrir tendências nos dados.

- Visualização de dados é usado para descobrir padrões num conjunto de dados. Ela apresenta uma gráfico para que o usuário possa visualizar melhor a automatização do processo.

Vários são os métodos que podem ser utilizados para a realização das técnicas de Data Mining, dentre eles destacam-se:

- Redes neurais é uma coleção de nós conectados com entradas, saídas e o processamento de cada nó. A rede é capaz de aprender; ela tem um treinamento para um conjunto de dados de entrada que produz um conjunto conhecido de saída. Cada caso num conjunto treinado é comparado com um conjunto conhecido; se ele difere, uma correção é calculada e aplicada para o processamento nos nós na rede. Esses passos são repetidos até encontrar uma condição de parada. Rede neural é um processo cujo modelo resultante não possui uma clara interpretação. As redes neurais são aplicadas para casos mais gerais de reconhecimento de padrões, como por exemplo, interpretação de eletro-cardiograma.

- Árvores de decisão dividem os dados dentro de grupos baseados nos valores das variáveis. Elas usam uma metodologia em que o resultado é uma hierarquia de comandos "if-then" que classificam os dados. Existe um grande interesse nessa ferramenta, primeiro porque elas possuem um aprendizado mais eficiente do que as redes neurais, produzindo resultados com segurança semelhante. A desvantagem é que elas necessitam de extensivos dados para descobrir a complexa estrutura.

- Algoritmos genético se definem como uma estrutura de dados que admite todas as possíveis soluções para um problema, sendo que

solucionar problemas consiste em encontrar uma solução ótima. Algoritmo genético é na realidade um método de busca muito especial, onde as soluções dos problemas são capazes de reproduzirem entre si, combinando novas características e gerando novos resultados. Esses métodos são procedimentos baseados nos princípios de seleção natural, onde cada ciclo da seleção gera resultado que mais se aproxima do objetivo, eliminando o resto das soluções. As repostas selecionadas reproduzem entre si, permitindo de vez em quando alguma mutação ou modificação durante a reprodução.

Na metodologia proposta, optou-se por trabalhar com árvores de decisão por diversos motivos:

- implementação simples bastante conhecida, desde a construção, poda e extração de regras;
- por ser uma estrutura binária as regras são facilmente geradas, explicitando o conhecimento;
- o aprendizado é eficiente e suporta aplicações de grandes dimensões;
- pode ser utilizada em versão incremental, incorporando apenas as pequenas mudanças quando o banco de dados é atualizado;
- é utilizada na maior parte dos sistemas comerciais de data mining [82].

II.6.1. Árvore de decisão

Árvore de decisão é um dos métodos de DM que usa uma representação de ramificação para lógica, com pontos de decisão em cada nó e a ação para ser levada até o término de cada ramo. É chamada de árvore de decisão porque o modelo resultante é apresentado em forma de uma estrutura de árvore, sendo que a sua apresentação visual é fácil de entender e assimilar [29].

O nó de topo é chamado de nó raiz. Uma árvore de decisão cresce a partir do nó raiz, dividindo-se os dados em cada nível formando novos nós. A árvore resultante

inclui muitos nós conectados por ramos. Os nós que estão no final dos ramos são chamados de nó folhas.

Árvores de decisão podem usar simplesmente um único nó dividido e ter uma forma de representação simples, fazendo o modelo deduzido relativamente fácil de entender. Esta restrição para uma representação de uma árvore em particular pode restringir a forma funcional do modelo significativamente. Alternativamente, se ampliar o espaço do modelo para acomodar expressões mais gerais, o modelo torna-se mais poderoso, porém torna-se também mais complexo de entender.

Árvores de decisão classificam instâncias (registros ou vetores) pela ordenação da raiz da árvore para algum nó folha que provê a classificação da instância. Cada nó na árvore especifica um teste de algum atributo da instância e cada ramo descendente daquele nó corresponde a um dos possíveis valores para este atributo. Uma instância é classificada começando pelo nó raiz da árvore, testando o atributo especificado para este nó, movendo para baixo o ramo da árvore que corresponde ao valor do atributo. Este processo é então repetido para a sub-árvore considerada então "raiz" o novo nó.

II.6.1.1. Algoritmos de árvore de decisão

O algoritmo desenvolvido para árvore de decisão emprega uma pesquisa top-down através do espaço de possibilidades da árvore. Existe um número de específicos algoritmos, incluindo árvore de Classificação e Regressão (CART), Descoberta de Interação Automática Chi-square (CHAID), C4.5 e C5.0. Um dos algoritmos mais conhecidos que empregam essa técnica é o algoritmo ID3. O ID3 é um algoritmo baseado em domínio simbólico. ID3 assume discretos domínios com pequena cardinalidade, revelando uma grande vantagem, pois aumenta a compreensibilidade do conhecimento induzido, mas pode a priori requerer um desmembramento. Outra característica importante do ID3 é que cada atributo pode prover no máximo uma condição num determinado caminho. Isto também contribui para a compreensão do conhecimento resultante [41].

O resultado primário de um algoritmo de árvore de decisão é a própria árvore. O processo de treinamento que gera uma árvore de decisão normalmente é chamado

indução. Indução requer um pequeno número de passos através do treinamento do conjunto de dados. Para ser mais preciso, o número de passos necessários para construir uma árvore de decisão não é mais do que o número de níveis na árvore. Não existe limite predeterminado, embora a complexidade da árvore geralmente medida pela profundidade e largura da árvore aumenta com o incremento do número de variáveis independentes.

Uma árvore é dita binária quando cada divisão possui dois ramos, há vários algoritmos e produtos criam somente árvores binárias. Algoritmos que produzem árvores binárias podem no entanto modelar os mesmos relacionamentos com algoritmos que não são restritos às divisões binárias. A única diferença é que a árvore não binária será um pouco mais compacta porque terá menos níveis.

Árvore de Indução

Muitos algoritmos de árvore de decisão passam por duas fases: uma fase de crescimento da árvore (dividindo) seguido por uma fase de poda.

A fase de crescimento da árvore é um processo iterativo que envolve divisão dos dados em subconjuntos progressivamente menores. Cada iteração considera o dado em um só nó. A primeira iteração considera o nó raiz contendo todos os dados. Iterações subseqüentes trabalham em nós derivados que contém subconjuntos dos dados.

O algoritmo começa analisando os dados para encontrar a variável independente que quando usada como uma regra resultará em nós diferentes dos outros nós em relação à variável dependente. Existe várias alternativas para medir esta diferença. Algumas implementações possuem somente uma medida embutida, outras deixam o usuário escolher que medida usar. Essas medidas podem ser: entropia, informação mútua, taxa de ganho, e chi-square. [29].

A árvore deve crescer até quando se tem a idéia de que nenhuma divisão precisa ser feita, normalmente os algoritmos de construção de árvores possuem várias regras de parar. Essas regras são baseadas em diversos fatores incluindo profundidade máxima de árvore, número mínimo de elementos em um nó considerado para divisão,

ou seu próximo equivalente, o número mínimo de elementos que devem estar em um novo nó.

Depois que um DM produz uma árvore, então é preciso explorar o modelo. Os modelos de árvores explorados podem revelar nós ou sub-árvores indesejáveis por causa de um novo ajustamento, ou pode conter regras onde o especialista do domínio sente que são impróprios.

Poda é uma técnica comum usada em uma árvore. Podar significa remover divisões e as sub-árvores criadas. Em algumas implementações, podar é controlado por parâmetros configurados pelo usuário que causam divisão para ser podado porque, por exemplo, a diferença computada entre os nós resultantes descem a um limiar e é insignificante.

Algoritmos que constróem árvores a profundidade máxima, automaticamente invocam poda. Em alguns produtos, os usuários também possuem a habilidade de podar a árvore interativamente.

II.6.1.2. Algoritmos de indução usados

O algoritmo de indução utilizado [121] é binário, ou seja, cria-se dois caminhos na árvore em toda sua divisão. A seleção dos atributos no estágios de construção da árvore é feita de acordo com o conteúdo da informação de cada atributo em termos de classificação do grupo resultado. O atributo mais significativo é selecionado. Para atributos discretos (textos), os valores do grupo são separados entre duas ramificações a fim de maximizar o conteúdo da informação do atributo. Para atributos numéricos os dois caminhos são baseados num threshold que é derivado para aumentar ao máximo o conteúdo da informação do atributo. Quando o resultado é numérico, o desvio padrão do dado filtrado para as ramificações são usadas como base para a seleção do melhor atributo e melhor threshold.

Um dos parâmetros que precisa ser especificado antes do processo de indução começar é o número mínimo de exemplo em um recuo. Esse parâmetro oferece ao algoritmo de indução um critério de parada na criação de novas ramificações de

qualquer ponto na árvore se o número de exemplo de dados filtrados para esse ponto falhar antes do limite. Normalmente é atribuído um valor default.

Podar uma árvore é simplesmente não classificar todos os registros. A poda da árvore é controlada pelo número mínimo de exemplo em um recuo e o nível significante do F-test cut-off ou Chi-square definido antes da indução da árvore. Isso é denominado de poda anterior, dessa forma o crescimento da árvore é controlado durante o processo de indução. Um valor de 1% é recomendado para o F-test.

Algoritmo de regra de indução para resultados discretos

Regra de indução é o processo de produzir uma árvore generalizada a partir de um conjunto de dados. A árvore é gerada recursivamente dividindo o conjunto de dados de acordo com diferentes atributos até alcançar as folhas (ponto terminal) . A ordem da qual os atributos são usados na árvore depende de uma medida de classificação de cada atributo. Um critério de poda anterior é usado para decidir quando os pontos terminais são alcançados. Para a classificação da árvore, a seleção de atributos para dividir em todo o estágio na construção da árvore é feito de acordo com o conteúdo da informação de cada atributo em termos de classificação de resultado do grupo. Esse conteúdo da informação é medida pela entropia com máxima informação correspondendo a mínima entropia. O atributo com baixa entropia é selecionado em todo ponto da ramificação. Para atributos numéricos, os dois caminhos são baseados em threshold numérico que é derivado para minimizar a entropia do atributo. Entropia é calculada de acordo com a seguinte expressão:

$$\sum_{i=1}^{2} \frac{\sum_{j=1}^{c} f_{ij}}{R} \sum_{j=1}^{c} P(c_{ij}) L_n (P(c_{ij}))$$

Onde:

c é o número do valor resultado

f é a freqüência do resultado j na ramificação i

R é o total do número de registro nas ramificações

P (cij) é dado pela seguinte expressão:

$$\frac{f_{ij}}{\sum_{K=1}^{c} f_{ij}}$$

O critério de parada é alcançado quando não é mais possível produzir duas ramificações com um número de registros igual ou maior que o parâmetro Minimum Examples in a brabch.

Regra de indução para resultados numéricos

Neste processo, assim como no previamente definido, a árvore é gerada recursivamente dividindo o conjunto de dados de acordo com diferentes atributos até alcançar as folhas (ponto terminal). A ordem da qual os atributos são usados na árvore depende de uma medida de classificação de cada atributo. Um critério de poda anterior é usado para decidir quando os pontos terminais são alcançados.

Para a regressão da árvore (isto é, resultados numéricos), a seleção dos atributos para a divisão é feita de acordo com a habilidade de cada atributo produzir ramificações com pequenos desvios nos valores de seus resultados. Isso é medido pelo NSD (Normalised Standard Deviation) de uma divisão. Para atributos discretos, o valor dos grupos são divididos entre duas ramificações para minimizar o NSD. Para atributos numéricos, os dois caminhos divididos são baseados num threshold numérico que é derivado para minimizar o NSD. O NSD para qualquer divisão é calculado de acordo com a seguinte expressão

$$\sum_{k=1}^{2} \frac{R_k}{R} \sqrt{\frac{\sum^{Rk} V_{ik}^2 - \left\langle \sum^{Rk} V_{iK} \right\rangle^2}{R_K}}$$

Onde:

Vik é o valor do resultado numérico do registro I na ramificação k

Rk é o número de registro na ramificação k

R é o número total de registros em ambas as ramificações

O critério para parar é alcançado quando não é mais possível produzir duas ramificações com número de registro igual ou maior que o Lower Branching Limit. O critério de parada é também alcançado quando o percentual F-test para uma divisão é mais alto do que o valor quando solicita o processo de indução.

Montando as folhas de uma árvore

Quando é usado um resultado discreto, as folhas da árvore podem ser um conjunto para mostrar todos os valores do resultado, ou o valor dominante do resultado. O resultado dominante é definido como o resultado com a maior produção de freqüência de registro. Também apresenta em cada folha a probabilidade do resultado. A probabilidade do resultado é definida como:

$$P = \frac{C_d \cdot NF_d}{\sum_{i=1}^{n} C_i \cdot NF_i}$$

Onde:

n é o número de resultados

C_i e NF_i são as freqüências, peso e fator de normalização do resultado de i

C_d e NF_d são as freqüências, peso e fator de normalização do resultado apresentado

Regra de indução interativa

Quando se desenvolve uma árvore com resultados discretos interativamente, o sistema mostra uma lista de atributos para toda divisão. Esses atributos são sorteados

pela entropia ou Chi-square prescrita pelo usuário. Pode-se usar entropia ou chi-square como a base do atributo selecionado ou forçar qualquer outro atributo escolhido.

O valor da ramificação de limite numérico de um atributo pode ser também alterado onde é considerado melhor pelo algoritmo. Isso força uma reavaliação da entropia ou chi-square e se necessário rearruma os atributos . Também apresenta o nível significativo e o status da redução de erros de cada atributo depois da divisão. Essa informação pode ser usada para decidir quando se deve parar a árvore.

Limites do sistema utilizado

LIMITE	DESCRIÇÃO
2.000 milhões	Número de registros na tabela do data source
1000	Número de campos na tabela do data source
100	Operações por script
20	Conexões abertas por script
1000	Número de campos na tabela do data source, ProfileX lê os primeiros 1000 e ignora qualquer campo sobrando
40 caracteres	Nome do campo, ProfileX trunca nome de campos longos usado na tabela do data source e permite duplicidade. Campos podem então serem internamente renomeados
Sem limite	Valores discretos por campo. ProfileX seleciona as primeiras 254 freqüências e coloca qualquer sobra dentro de uma freqüência baixa num simples grupo
100	Número de atributos a serem formados em cada passo
100	Número de atributos incluídos para indução
32	Número de valores de resultados ou grupos de resultados

II.6.2. Hardware e Software para DM

A escolha do tipo de software e hardware depende muito da necessidade do usuário. A do hardware depende do volume de dados envolvidos e de técnica que se

deseja trabalhar; nem todos os SGBD's possuem todas as técnicas que são necessárias para o ambiente de DM.

Trabalhar com um ambiente cliente/servidor permite uma grande flexibilidade na escolha de softwares apropriados para o usuário, tendo em vista que cada usuário trabalharia numa workstation local. O elemento comum é o banco de dados que precisa ser totalmente otimizado, suprindo um número de funcionalidades para ter uma alta performance. Os banco de dados atuais podem também ter partes similares em outros bancos de dados. DM pode dividir o software em duas partes: a primeira trabalha com algoritmos no servidor do banco de dados e a segunda parte trabalha com os clientes; locais para gerar telas e relatórios para a visualização dos algoritmos.

Um ambiente de DM tem necessidade de hardware específicos. Certas tarefas de máquina de aprendizado envolvem a comparação de milhões de registros. Se um usuário precisa comparar um grande volume de registros por pequeno período de tempo, então o computador precisa ter a sua memória interna e todo o seu processamento numa única tarefa.

Como por exemplo quando se trabalha com algoritmos genéticos, é preciso comparar cada registro com outros milhões de registros dentro do banco de dados e encontrar um certo padrão dentro da base de dados, recalcular esses padrões, comparando-os constantemente com todos os registros, sendo que em alguns casos o usuário precisa de um resultado em curto espaço de tempo. Para obter um resultado satisfatório existe a possibilidade de mover o banco de dados para um sistema de multi-processamento. Em grandes bancos de dados, máquinas de multi-processamento são necessárias para o projeto de Data Mining.

Os dois tipos mais importantes de multi-processamento de máquinas são: o multi-processamento simétrico e o massivamente paralelo. O ambiente de máquina de multi-processamento simétrico trabalha com todos os processadores num único computador, todos são iguais e se comunicam via armazenamento compartilhado. Este sistema compartilha o mesmo disco rígido e a mesma memória interna.

O sistema massivamente paralelo corresponde a uma arquitetura de computador onde cada processador tem o seu próprio sistema de operação, sua memória e o seu disco rígido. Cada processador é independente com comunicação

entre os sistemas. Neste tipo de ambiente é permitido trabalhar com muitos processadores, mas cada parte do software é executado num processador específico, possuindo um bom tempo de resposta.

O banco de dados relacional Oracle desenvolveu um software para uma arquitetura hardware massivamente paralela junto com OLTP e suporte de decisão numa mesma imagem de banco de dados. As consultas e os processamentos de transação são representadas linearmente numa plataforma Massively Parallel Process (MPP).

A arquitetura de computador convencional de hardware e software disponíveis hoje incluem uniprocessadores, multiprocessamento simétrico (SMP) e clusters acoplados; não fornecendo capacidade de escalabilidade para suportar o armazenamento de dados e sistemas de recuperação.

Os MPP utilizam CPU como sua própria topologia conectiva (isto é, mesh, hipercubo) para criar sistema que escalam bem em torno das quatro dimensões escaláveis - CPU, memória, armazenagem de I/O e rede I/O.

Sistemas MPP tem penetrado timidamente nas aplicações de gerenciamento de dados. O objetivo é escrever software de banco de dados que possam receber vantagem do hardware do MPP. Essa tecnologia permite que múltiplos acessos independentes aos processadores compartilhem o banco de dados e garanta uma gerência quando dois processadores tentam simultaneamente acessar um mesmo item de dados.

A escalabilidade simples introduz uma grande vantagem, pois a aplicação de banco de dados pode ser escalada adicionando nós de processamento. Nenhuma aplicação precisa ser modificada e nenhum dado necessita ser redistribuído. Muitos nós podem ser também adicionados para as aplicações, permitindo a construção de um sistema de alta performance. Resulta numa fácil administração, pois existe somente um banco de dados para o processo de start-up, tune, monitor e back-up. Possui ainda uma grande acessibilidade, porque todos os nós tem acesso direto para entrar no banco de dados. Se um nó falha, outro nó ainda terá acesso a todos os dados e o processo de banco de dados pode continuar.

Hardware e DM possuem um estreito relacionamento. É devido as novas técnicas de hardware como MPP que foi possível introduzir as modernas técnicas

como a de DM. Com essas máquinas também é possível comparar milhões de registros em pequeno período de tempo. Atualmente modernos bancos de dados, como por exemplo, Informix, Sybase, DB/2 da IBM, Oracle, Tandem, etc., operam com computadores do tipo MPP.

II. 7. Conhecendo o ambiente de Data Mart

O Data Mart é um forma simples de Data Warehouse (DW) enfocando apenas uma área funcional. A diferença fundamental entre o DW e o Data Mart localiza-se no escopo da área abordada, pois ambos, em termos de dados, fornecem suporte para gerar conhecimento aos usuários com aceitáveis tempo de resposta. O DW compreende uma grande coleção de dados dentro de uma organização, envolvendo múltiplas áreas e o interrelacionamento entre elas, e necessariamente requer maiores recursos computacionais e um controle mais rígido. Já o Data Mart compreende uma coleção menor de dados, podendo ser um subconjunto dos dados existente no DW ou podendo ser também um conjunto de dados cobrindo apenas uma determinada área específica da organização

Os Data Marts são menores e menos complexos se comparados com o DW, e portanto, são fáceis de construir e de se manterem. Entretanto, não existe nada na definição básica que limita o tamanho de um Data Mart ou a complexidade de dados que ele contém. A tabela da figura II.12 apresenta um resumo da pequena diferença básica entre o DW e o Data Mart.

	DATA WAREHOUSE	**DATA MART**
Escopo	Toda organização	Parte de uma organização
Objetivo	Múltiplos objetivos	Apenas uma meta
Data Sources	Muitos	Poucos
Tamanho	100GB – TB+	<100GB
Tempo de Implementação	Meses - Anos	Meses

Figura II.12 – Diferença entre DW e Data Mart

Existem dois tipos básicos de Data Mart: o dependente e o independente. Data Mart dependente extrai os seus dados de um DW já existente, e o Data Mart independente extrai seus dados diretamente de um data source externo, conforme mostra figura II.13.

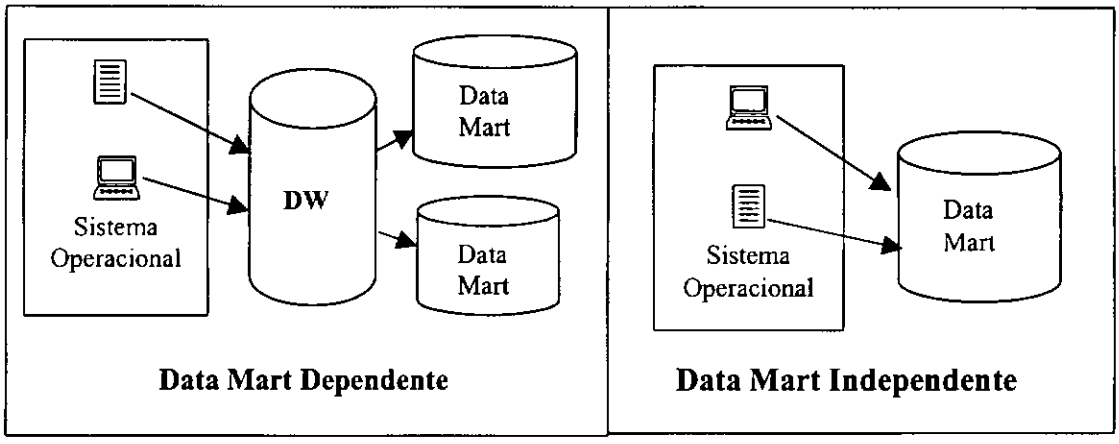


Figura II.13 – Tipos de Data Mart [32]

A diferença básica nesses dois tipos está baseada na maneira de como alimentar o Data Mart. Este processo envolve um conjunto de tarefas relacionadas com a extração dos dados de um sistema operacional, limpeza dos dados e armazenamento desses dados no Data Mart. No Data Mart dependente, a limpeza dos dados já foi feita dentro do DW, e aqui é uma questão simples de identificação na escolha do subconjunto de dados relevantes para copiar ou talvez para sumarizar o Data Mart. Já no Data Mart independente, é possível trabalhar com todos os aspectos do processo. Normalmente, o Data Mart dependente é construído para melhorar a performance e a acessibilidade, melhorar o controle e diminuir o custo de comunicação com apenas o acesso local a dados relevantes de determinadas partes da organização.

Os três maiores passos para a construção do Data Mart proposto são: alimentar os dados a partir da sua extração de um sistema operacional, armazenar e administrar os dados, e depois acessá-los para tomadas de decisões. A alimentação dos dados engloba todas as tarefas relacionadas com a identificação do data source e seleção de um subconjunto de dados relevantes; um projeto físico e lógico do Data Mart;

obtenção, limpeza e modificação dos dados, caso necessário, para um formato apropriado; seu carregamento inicial no Data Mart; e a criação e armazenamento de um metadado. O armazenamento de dados está relacionado com a construção de um Data Mart que fornece um acesso aos dados de forma rápida e eficiente e uma administração contendo diversas tarefas, tais como: acesso seguro, trabalhar com aumento no volume de dados e assegurar acessibilidade aos dados. E o uso do Data Mart está diretamente ligada a consulta aos dados, sua análise, criação de relatórios e gráficos, e sua publicação.

O DW contém dados históricos e integrados que são comuns para toda organização; já o Data Mart também contém dados históricos, porém esses dados fazem parte apenas de uma área específica da organização. Tanto o DW quanto o Data Mart possuem um metadado que descreve o conteúdo dos dados armazenados. Assim, vale ressaltar que o Data Mart é uma forte e natural extensão do DW.

II.8. Integração do Data Mining(DM) com Data Warehouse (DW)/Data Mart

Data Mining trabalha com grandes volumes de dados e esses dados são normalmente armazenados em banco de dados, tendo como objetivo retirar conhecimento através das informações armazenadas. Os bancos de dados tradicionais são usados para fornecer respostas para questões simples. Análise multidimencional, denominada online analytical processing (OLAP), permite que os usuários façam consultas mais complexas, mas mesmo assim são resultados para extrair valores ou uma agregação de valores. A ferramenta de Data Mining e a ferramenta de consulta são complementares. Uma ferramenta de DM não se coloca no lugar de uma ferramenta de consulta, mas ela pode dar ao usuário um conjunto adicional de possibilidades.

Atualmente aplicações de DM trabalham com alguns ambientes de dados, e cada um possui as suas vantagens:

- Os flat files são bastante usados como ambiente de dados para algoritmos de DM. Até mesmo quando os dados estão armazenados em

um sistema de banco de dados, parte deles são retirados e colocados em um flat files para serem utilizados pelo processo de DM, trabalhando somente com dados ajustados na memória, que limitam o total de informação a ser usada.

- Sistemas de gerenciamento de banco de dados (SGBD) são normalmente usados para processamento convencional de dados, porém, podem fornecer características valiosas quando se utiliza a tecnologia de DM. Como por exemplo, é possível trabalhar com conjunto de dados que são consideravelmente maiores que memória principal, assim o banco de dados é responsável em controlar informação, páginas e swapping quando necessário. Outros, como sistemas de banco de dados orientado a objetos oferecem habilidade para manipular imagem, vídeo e voz. Existe um grande potencial nessa área para explorar DM em tipos de dados complexos.

- Como já foi mencionado anteriormente o ambiente de DW/Data Mart tem como enfoque principal o armazenamento central de dados que foi retirado de um sistema de banco de dados operacional. Os dados extraídos são tratados e sofrem algumas transformações para eliminar inconsistências, sumarizados se necessários e colocados dentro do DW/Data Mart. As informações no DW/Data Mart são usados somente quando são estáveis, pois a sua estabilidade é fundamental para análise e tomada de decisão.

Não é necessário ter acesso ao DW/Data Mart para realizar um bom trabalho com Data Mining, mas na prática isso ajuda e muito. É possível trabalhar com a técnica de DM extraindo dos DW/Data Marts existentes a parte da informação que é de interesse do usuário final e também copiando os dados existentes para um computador específico, possibilitando um multi-processamento de máquina. A função do DW/Data Mart é fornecer as informações necessárias para fazer aquisição de decisão. Em alguns casos é possível usar a ferramenta padrão SQL para tomada de

decisão, mas para comparar milhões de registros e não conhecer exatamente o tipo de informação que está sendo requisitada, é então necessário o uso da técnica de DM.

A integração do DW/Data Mart com o DM é importante porque é impossível extrair dados de diferentes aplicações de bancos de dados com diferentes tipos de atributos e tipos de dados. Com DW/Data Mart esse problema não existe, pois toda a informação já foi transferida do banco de dados operacional para o DW/Data Mart, em muitos casos limpando os dados antes de começar o DM.

A informação no DW/Data Mart é subject-oriented, não volátil e possui uma natureza histórica, assim DW/Data Mart contém grandes conjuntos de dados. A combinação de DW/Data Mart, tomada de decisão e DM é uma inovação em gerência de informação.

II.9. Considerações Finais

De acordo com o comentado neste capítulo, os ambientes de DW/Data Marts reúnem grandes quantidades de dados e habilidade para resumir dados a um nível mais alto de generalização ou para encontrar detalhes em níveis mais baixos. Ambientes de DW/Data Mart oferecem suporte para processamento analítico on-line, sistemas de tomada de decisão e sistemas de informação.

Neste capítulo, ressaltou-se a importância do metadado dentro do ambiente de DW e conseqüentemente se estende para o Data Mart. É interessante observar que a visão futura do metadado é construir um ambiente de dados totalmente integrado onde possa entender e dispor rapidamente do seu conteúdo e significado dentro de um contexto comum para sua máxima utilização, oferecendo suporte para os sistemas de informação. Um ambiente integrado de dados, entretanto, não pode ser desenvolvido em apenas um passo e os dados discrepantes já existentes em outros sistemas também não podem ser transformados dentro de um ambiente integrado em um pequeno período de tempo. Como observado, desenvolver e construir um metadado é uma tarefa difícil, sendo tão difícil quanto desenvolver e gerenciar os dados. Porém, pode ser alcançada e se tornar mais fácil caso o desenvolvimento desse ambiente integrado de dados possuir técnicas e metodologias a serem seguidas.

Comentou-se também o ponto onde o dado se torna mais valioso - sua transformação em conhecimento. A estratégia para extrair dados de grandes quantidades de dados através do processo de descobrimento em banco de dados e mais especificamente usando DM, encontra sua utilização para fazer a predição e/ou a descrição. A predição envolve campos do banco de dados para predizer conhecimentos explorando informações ou valores de variáveis e a descrição se concentra em encontrar padrões interpretáveis descrevendo os dados.

Pelo que foi apresentado, pode-se perceber a importância e necessidade dessa ferramenta de projeto, que na verdade é um dos pontos fundamentais quando aplicado à gerenciamento de grandes massas de dados. O cumprimento desta etapa garante que a carga atual de dados novos seja um conjunto completo e consistente de dados, possibilitando o seu controle e qualidade. Sem dúvida, o planejamento e a construção de um DW/Data Mart é a base para o restante do processo.

CAPÍTULO III

TRATAMENTO DE INCERTEZA

III.1. Introdução

Na área de banco de dados tem ocorrido grandes avanços e o modelo dominante de banco de dados vem sofrendo modificações. Os bancos de dados são formas de modelar os aspectos do mundo real de maneira que todos os dados e relacionamentos sejam claros, precisos e possuam um valor, porém pouco se explorou para casos em que os dados são considerados incertos. A incerteza não está normalmente ligada a um ambiente de banco de dados. Isso contrasta com outras áreas de aplicações, tal como aplicação de engenharia, tomada de decisão, sistemas de controle, sistemas baseado em regras, etc., onde gerenciamento de incerteza tem sido incorporado e tem permitido novos e interessantes métodos. Essas aplicações tem em comum a ênfase na manipulação de dados, porém são menos exigentes em relação aos modelos de informação do que o ambiente de banco de dados.

As atuais pesquisas vem sendo desenvolvidas para entender os aspectos de incerteza no modelo de banco de dados e também propriamente nos dados. No entanto, são necessárias metodologias para fazer a manipulação da informação e ainda possuir uma representação inteligente. Assim, para realizar essa proposição a ferramenta de lógica fuzzy pode certamente ajudar.

A teoria de conjuntos fuzzy teve início em 1965 com o trabalho de L. A . Zadeh. Nas duas últimas décadas essa teoria também vem sendo desenvolvida e refinada, passando portanto a encontrar seu caminho dentro de um grande número de aplicações, em especial na área de controle. Mesmo assim, essa área ainda explora somente uma pequena parte da representação do conjunto fuzzy.

O conceito de lógica fuzzy, introduzida por L. A. Zadeh, é uma formulação de lógica que permite o tratamento de conceitos inerentes, nebulosos, vagos, usados por

nós, seres humanos, numa estrutura matemática rigorosa. A lógica fuzzy é um cálculo de compatibilidade [26].

Várias teorias foram desenvolvidas com o intuito de lidar com a incerteza. Estão incluídas aí: a probabilidade clássica, probabilidade Bayesiana, a teoria de Hartley nos conjuntos clássicos, a teoria de Shannon baseada na probabilidade, a teoria de Dempster-Shafer e a teoria fuzzy de Zadeh. E muitos métodos foram propostos no sentido de tratar computacionalmente esse conhecimento, incorporando-o aos outros tipos de conhecimento num processo de tomada de decisão, para permitir a escolha da melhor solução [57].

Neste capítulo será apresentado de maneira sucinta a teoria de conjuntos fuzzy e um breve comentário sobre a área de banco de dados, enfocando a integração da teoria de banco de dados com a teoria de modelagem de incerteza, sendo que combinando as vantagens de ambas teorias é permitido um projeto de banco de dados semanticamente muito mais forte e amigável que a teoria tradicional. É introduzida também uma abordagem de recentes pesquisas sobre árvore de decisão fuzzy.

III.2. Incerteza

É impossível capturar toda a informação semântica do mundo real, normalmente, a observação e o conhecimento sobre o mundo real são deficientes e, como conseqüência, sua modelagem resulta numa representação imperfeita, sendo tratada na literatura como incerteza, independente da área de aplicação.

Existe uma variedade de diferentes tipos de incerteza, a ciência tem ignorado muito a incerteza nos seus modelos, e prefere considerar a incerteza como falta (perda) de informação. Admitindo o uso de incerteza de informação é admitir levar em conta o conhecimento da informações dos indivíduos. Uma tradição na ciência é lutar

por uma objetividade e certeza absoluta, baseada em fatos e figuras de comum entendimento e pela avaliação e aprovação da comunidade. Com a chegada do terceiro milênio, essa tradição está mudando e a manipulação de incerteza está sendo reconhecida como ciência essencial.

Tipos de Incerteza

Focalizando na área de aplicação de banco de dados, é interessante ressaltar os diferentes tipos de incertezas que podem existir sobre dados, sobre a estrutura de dados, sobre os fatos envolvidos ou relacionados aos dados e sobre o significado dos dados. E também, os modelos de dados como também os modelos de banco de dados podem ser afetados pelos diversos tipos de incerteza.

Diante de uma classificação de incerteza com relação as aplicações de banco de dados e assumindo a incerteza do mundo real, utiliza-se o termo incerteza para referenciar a qualquer elemento de um modelo que não pode ser declarado com completa confiança, como mostra a tabela III.1. A existência de diferentes tipos de incerteza resulta em diversos caminhos para modelar incerteza.

TIPO	DESCRIÇÃO
- incerteza	quando não é possível determinar se sua declaração no modelo é verdadeira ou falsa
- imprecisão	quando a informação disponível no modelo não é específica como deveria ser
- vago (confuso)	quando o modelo inclui elementos que são basicamente vagos
- inconsistente	quando o modelo contém duas ou mais declarações que não podem ser verdadeiras ao mesmo tempo
- ambigüidade	quando alguns elementos do modelo perdem semântica, direção para possíveis interpretações

Tabela III.1 – Classificação de incerteza

III.3. Teoria de Conjuntos Fuzzy

III.3.1. Conceitos básicos

Conjuntos fuzzy

O ser humano raciocina em termos de símbolos fuzzy, não em termos de números ou símbolos discretos; esses termos fuzzy definem categorias gerais, que não são coleções rígidas ou fixas, mas *conjuntos elásticos,* os *conjuntos fuzzy*. Pode-se definir fuzziness (nebulosidade), como a medida de quão bem uma instância (valor) se adequa a um conceito semântico, descrevendo o grau *de pertinência a* um conjunto fuzzy . Esse grau de pertinência pode ser visto como o nível de compatibilidade entre uma instância do conjunto do domínio e o conceito que representa o conjunto – denomina-se essa medida de fuzziness por ser usada para acessar o grau de incerteza ligada a cada conjunto, algumas medições têm nebulosidade mínimas. A definição do que representa o conjunto fuzzy, depende do contexto em que está sendo usado [57].

A lógica fuzzy tem o suporte da teoria de conjuntos fuzzy, em que um conjunto fuzzy F num universo de discurso U é definido como um conjunto ordenado de pares (x,y), onde x é um elemento do universo de discurso e y é um elemento de um intervalo fechado [0,1]:

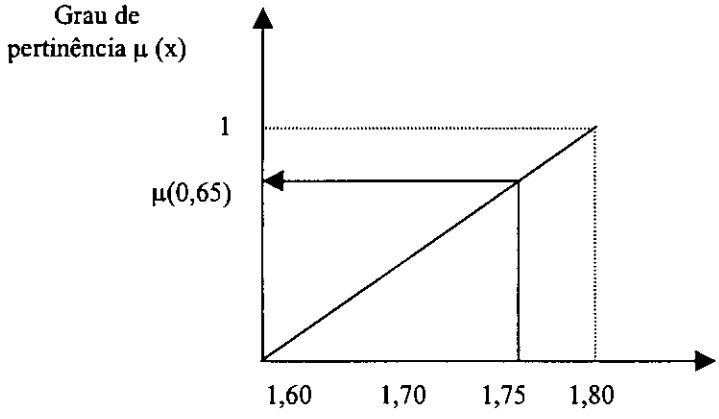
$$F = \{ (x,y) \mid x \in U, 0 \leq y \leq 1 \}$$

O valor y é chamado de grau de pertinência associado com x pelo conjunto fuzzy F. O grau de pertinência 1 indica uma pertinência fechada de x em F; o grau de pertinência 0 não indica pertinência em F, e *um* grau de pertinência $0<y<1$ indica um grau de pertinência parcial de x em F, ou seja, significa que o elemento de x pertence ao conjunto fuzzy somente por algum grau, que é indicado pelo grau de pertinência. Como um simples exemplo, considere a idéia de um indivíduo alto, como mostra a figura III.1. O conjunto fuzzy consiste de três componentes:

- uma linha horizontal apresentando números reais de forma crescente, que constitue a população do conjunto e que representa o domínio, (neste

caso, altura em metros) num intervalo considerado representativo do contexto do modelo em questão;

- uma linha vertical apresentando valores entre 0 e 1, que indica o grau de pertinência do conjunto, ou seja, o quanto cada número do eixo horizontal reflete – "ser alto";
- uma função, que faz a ligação das duas linhas (vertical e horizonta), no caso uma curva linear simples, mostrando que "ser alto" é diretamente proporcional à um valor, que representa a altura em metros.



Domínio: altura em metros
Figura III.1 – Exemplo conjunto fuzzy alto

A figura III.1 indica que abaixo de 1,60 m, inclusive, não significa ser alto; acima de 1,80 m significa ser alto e que 1,75 m significa ser moderadamente alto, sendo que esse conjunto está representando o conceito "alto" para a realidade brasileira [25].

Variável Lingüística

O ponto central da técnica de modelagem fuzzy é a idéia de variável lingüística, que é na verdade, o nome de um conjunto fuzzy. Variáveis lingüísticas também conduz em conceito de qualificadores ou reguladores (hedges) de conjunto

fuzzy, esses qualificadores modificam o corpo do conjunto fuzzy em seus atributos e funções.

As variáveis lingüísticas modelam o conjunto fuzzy para expressar o significado semântico usado pelos especialistas. Os qualificadores aplicados ao conjunto fuzzy expressam quantidades, freqüências, reduzindo a complexidade na descrição do dado em questão. Uma variável lingüística encapsula, como mostra a figura III.2, as propriedades de aproximação e conceitos de imprecisão na semântica e em caminhos computacionais. Reduz a aparente complexidade ao descrever um sistema em união com a semântica e o conceito, ela representa um espaço fuzzy.

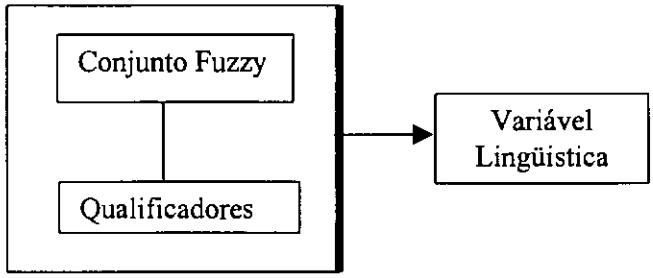


Figura III.2 – Estrutura da variável Lingüistica

Existem várias classes diferentes de qualificadores; qualificadores que intensificam as características de um conjunto fuzzy tornando-o menos fuzzy, como *muito, extremamente*; outros diluem a curva da função de pertinência, tornando o conjunto mais fuzzy, como *de alguma forma, quase*; os que geram o complemento de um conjunto, *não*, e os que aproximam uma região fuzzy ou convertem um escalar a um conjunto fuzzy, como *aproximadamente, perto*. Existe, portanto, uma série de considerações que pode ser feita a cerca da utilização dos qualificadores.

Funções de Pertinência

O conceito de função de pertinência representa numericamente o quanto um elemento pertence a um conjunto e representa o terceiro elemento de um conjunto

fuzzy: é a superfície do conjunto propriamente dito, que estabelece uma correspondência de um-para-um entre um elemento no domínio e um valor verdade que indica o grau de pertinência do elemento no conjunto. A função de pertinência é representada pela equação:

$$\mu_A(x) \leftarrow f(x \in A), \text{ onde A é o subconjunto fuzzy e } 0 \leq \mu_A \leq 1.$$

Essa função de pertinência não é simétrica, embora cada elemento do domínio só tenha um correspondente na função, cada grau da função de pertinência pode ser mapeado para mais de um ponto do domínio como em conjuntos fuzzy convexos que podem ser representados por triângulos, curvas beta, Gaussiana, etc.

Um conjunto fuzzy codifica a imprecisão associada a um fenômeno através de sua superfície, na verdade, o desenho da curva representa a semântica do conceito em questão.

Um exemplo simples de um conjunto fuzzy explicitado através de sua função de pertinência de alguns elementos de domínio, é apresentado da seguinte forma:

- universo de altura, definido por: U = (1,60; 1,70; 1,75; 1,80)
- subconjunto fuzzy alto, definido por: alto = (0.01/1,60; 0.5/1,70; 0.65/1,75; 1.0/1,80)

Tipos de Funções de Pertinência

Embora existam algumas curvas padrões, ocasionalmente elas podem não ser capazes de representar a semântica de um determinado conceito, isso é realmente verdade quando um conjunto fuzzy e seu complemento são usados para representar o estado de uma variável independentemente; assim é preciso traçar a curva que mais se aproxima do comportamento da variável em questão. Os principais tipos de representação de função de pertinência são:

- Representações lineares: a figura III.2 mostra uma representação linear crescente. É o tipo de representação mais simples e uma boa escolha para aproximar conceitos muito mal compreendidos, que não são números fuzzy;

- Representações em curva-s (sigmóide/logística): correspondem a superfícies não lineares crescentes e decrescentes Esse tipo de representação é bastante comum em muitos modelos. É utilizada em representações de frequência como *usualmente, a maioria de, próximo a, etc.*, onde o domínio representa um conjunto fuzzy de proporções, sendo esse conjunto fuzzy do tipo de conhecimento usado para expressar o pensamento no dia-a-dia; a base do senso comum. Uma curva-s é definida utilizando-se três parâmetros: o valor de pertinência para o limite mínimo, valor de pertinência para o limite máximo, e a inflexão, ponto em que o valor do domínio é 50% verdadeiro. A figura III.3 apresenta um exemplo dessa representação;

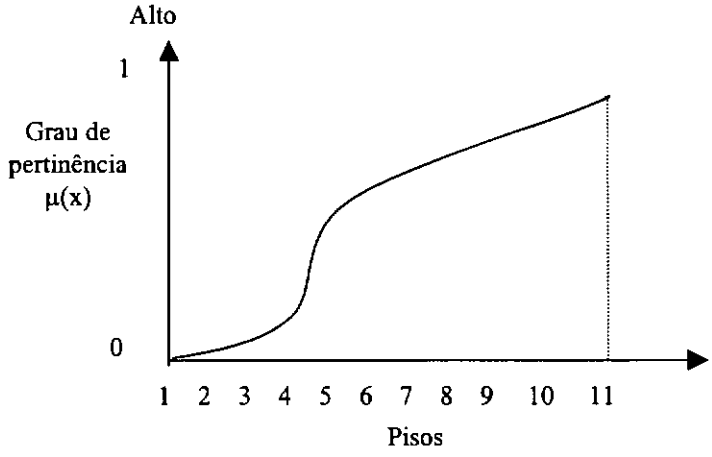


Figura III.3 – Conjunto fuzzy para um prédio alto com uma representação não linear

- Representações em curvas tipo sino: É uma classe importante de representações fuzzy, é baseada na representação das aproximações de um valor central, sendo graficamente visualizada como uma curva do tipo sino. Existem três importantes classes de curvas desse tipo: as curvas PI, Beta e

Gaussiana; a diferença está na abertura e nos valores dos pontos terminais. Essas curvas, ou melhor, conjuntos fuzzy, são usadas para representar números e espaços fuzzy. Um número fuzzy representa uma aproximação de algum valor e normalmente funciona melhor do que seu valor rígido; a figura III.4 ilustra essa afirmativa. Os números fuzzy também são muito importantes na definição de conceitos extremamente vagos como *algum, pouco*, que embora sejam altamente dependentes de contexto, são quantidades comparáveis. A noção de quantidade é importante na modelagem fuzzy e demonstra a característica de sobreposição dos conjuntos fuzzy , também ilustrado na figura III.4.

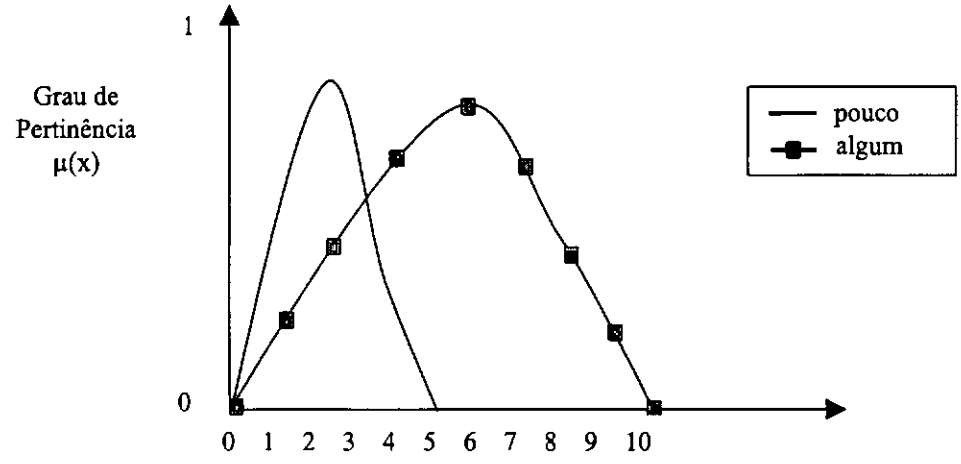


Figura III.4 – Quantidades fuzzy pouco e algum representado pela curva tipo sino

- Representações Triangulares, Trapezoidais e Patamares: São uma variação das curvas tipo sino, muito utilizadas nos processos da engenharia. Elas surgiram devido a busca de representações que permitissem ser especificadas com um mínimo de representação (memória e velocidade de processamento) e máximo aproveitamento: cada uma pode ser representada usando apenas 4 bytes; é representada em várias gradações,sendo cada uma por triângulos (trapézios ou sigmóides), geralmente com alto grau de sobreposição, a combinação final, representa então a variável em questão e

a união dos domínios de cada gradação representa o universo de discurso da variável; como mostra a figura III.5.

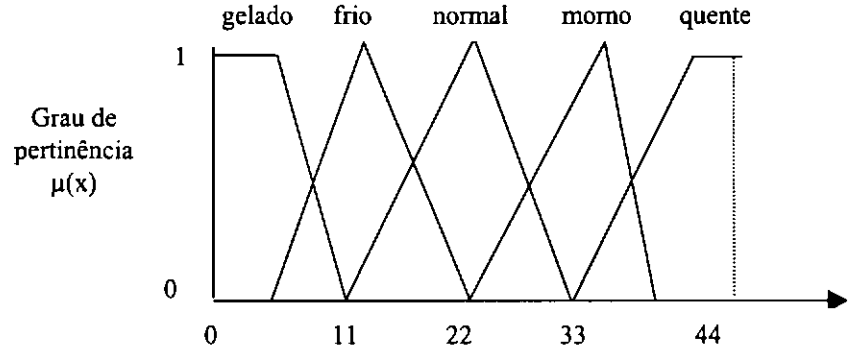


Figura III.5 – Representação triangular de temperatura ambiente

Operações Básicas de Zadeh sobre Conjuntos Fuzzy

Assim como na teoria de conjuntos convencionais, existem operações definidas especificamente para combinar e modificar conjuntos fuzzy. As operações básicas definidas por Zadeh são aplicadas ao nível da função de pertinência, elas não particionam os conjuntos, mas determinam a compatibilidade entre cada conjunto fuzzy e seu valor no domínio.

Tipos de operações:

A intersecção e a união correspondem e são realizadas pela aplicação dos operadores lógicos AND e OR respectivamente.

- Intersecção: $A \cap B \Leftrightarrow \mu_A(x) \cap \mu_B(x) = \min(\mu_A(x), \mu_B(x))$

- União: $A \cup B \Leftrightarrow \mu_A(x) \cup \mu_B(x) = \max(\mu_A(x), \mu_B(x))$

O complemento registra o quanto o valor de um elemento (x) é compatível com a assertiva x não é Y, onde x é um elemento do domínio e Y é uma região fuzzy.

- Complemento: $\sim A = 1-\mu_A(x)$

Pela caracteristica da função de pertinência (pertinência: $x \in A$), um elemento é membro de um conjunto fuzzy se:

a) ele está no domínio do conjunto;

b) seu valor de pertinência é maior do que zero;

c) seu valor de pertinência está acima do limiar alfa

III.4. Conhecendo os Princípios e Características de Banco de Dados Fuzzy

III.4.1. Conceitos básicos de Banco de Dados

A evolução de banco de dados começou com o uso de arquivos seqüenciais de dados para processar as informações, a partir daí, aplicações em banco de dados se multiplicaram e como conseqüência surgiram os conceitos e problemas básicos, dando início ao desenvolvimento do Sistema de Gerenciamento de Banco de Dados (SGDB) que desde a década de 70 estão disponíveis comercialmente.

O SGBD começou com o relacionamento entre a aplicação do usuário e a estrutura física do arquivo de dados (conjunto de registros). Essa estrutura possuía vários problemas com relação a um dispositivo de arquivo, organização, uso, etc. causando assim problemas nas aplicações dos programas. Depois surgiu um relacionamento entre a aplicação do usuário e o arquivo de dados com algumas facilidades em relação à de gerencia de dados, e portanto, a organização da estrutura na aplicação não precisava mais possuir a mesma organização física do arquivo de dados. Assim, passa a existir o grau de independência de dados, ou seja, quando existe necessidade de realizar uma alteração na estrutura de um arquivo, não ocorre mais a necessidade de se alterar a estrutura de dados na aplicação do programa. Surge, portanto, o verdadeiro ambiente de base de dados, isto é, a estrutura da aplicação pode acessar dados de qualquer organização de arquivo, caso o seu mapeamento possua formas disponíveis apropriadas, possibilitando um alto grau de independência de dados - descrevendo a estrutura de um completo SGBD.

A arquitetura para um SGBD é formada por três níveis, como mostra a figura III.6, conhecida como arquitetura ANSI/SPARC. O primeiro nível possui o esquema interno, que descreve fisicamente a estrutura armazenada do banco de dados, contendo detalhes dos dados armazenados e os métodos de acesso a eles; o segundo - nível lógico - descreve o modelo lógico que é uma visão da estrutura do banco de dados e de todos os usuários, trazendo detalhes dos usuários e se concentra nas descrições das entidades, tipos de dados, relacionamentos, operações dos usuários e

restrições; descrevendo as características particulares do banco de dados; e o terceiro - nível externo - inclui um número de esquemas externos ou visões de usuários, contendo geralmente a base de dados virtual de cada grupo de usuários.

Um SGBD oferece uma integração de inúmeras facilidades. Entre elas, encontra-se um suporte para a definição de uma estrutura lógica de entidades e relacionamentos, sendo especificado pela linguagem de definição de dados (LDD), e fornece também suporte para uma linguagem de consulta dentro da linguagem de manipulação de dados (LMD).

Os níveis oferecem somente descrições dos dados, pois os dados propriamente ditos estão armazenados num outro meio. Assim para obter respostas para as necessidades dos usuários do nível externo, a pergunta precisa passar por esses três níveis. O processo de transformar perguntas em resultados entre níveis são chamados de mapeamento. Esse mapeamento pode consumir um certo tempo, mas é a base para fornecer a independência lógica e física dos dados.

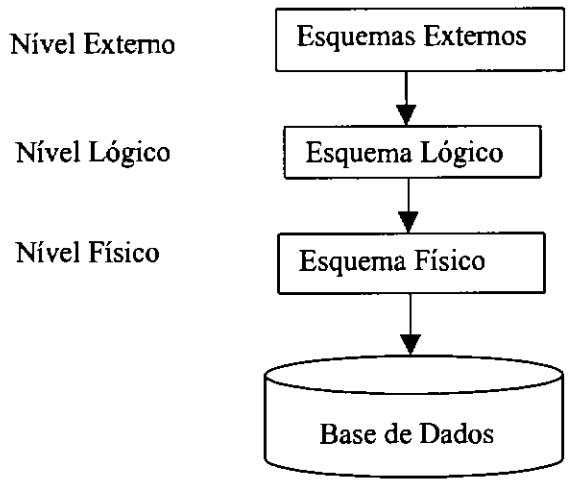


Figura III.6 – Arquitetura ANSI/SPARC

III.4.2. Projeto de Banco de Dados

O objetivo do processo de projeto de banco de dados é obter o aspecto essencial do mundo real e construir um banco de dados. A figura III.7 apresenta uma

descrição simplificada do processo de projeto de banco de dados. O primeiro passo abrange a coleção das necessidades e análise em que o projetista de banco de dados irá interagir com os usuários para entender e documentar os seus requisitos. O resultado desse passo é relatar todos os requisitos dos usuários. O próximo passo é criar um esquema conceitual para o banco de dados, usando o modelo de conceitual de dados. O esquema conceitual é uma descrição concisa de todas as necessidades dos usuários, incluindo detalhes dos tipos de dados, relacionamentos e restrições, o modelo entidade-relacionamento é um exemplo. Assim esse nível não apresenta detalhes da implementação, ele se concentra apenas em especificar.

O próximo passo é o projeto do modelo lógico, que pode ter como resultado uma modelagem relacional, uma modelagem orientado a objetos, etc. Essa é a representação do banco de dados para ilustrar uma aplicação particular .

Finalmente, o último passo é a fase do projeto de banco de dados físico que provavelmente terá certas escolhas para serem feitas em relação ao armazenamento da estrutura interna e organizações dos arquivos baseados em critérios de performance e necessidades de acesso.

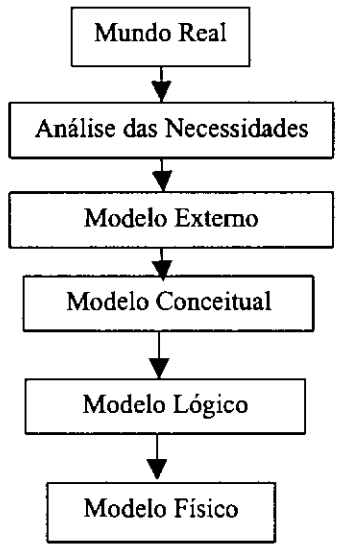


Figura III.7 – Processo de projeto de banco de dados

III.4.3. Modelos de Banco de Dados

Existe uma variedade de modelos de dados que são e estão sendo desenvolvidos desde modelo hierárquico ao relacional, como também o modelo de dados orientado a objetos.

a) Modelo Hierárquico e de Redes

Os primeiros modelos de banco de dados desenvolvidos foram os modelos hierárquico e os de redes. Os relacionamentos nos modelos hierárquico eram representados como uma estrutura de árvore hierárquica e nos modelos de redes como estruturas de ligaçào através de ponteiros. O modelo hierárquico utiliza a estrutura de árvore, onde cada hierarquia representa um conjunto de registros relacionados. O modelo de redes usa registros para representar dados e uma representação direta direcionada de um relacionamento de um-para-muitos.

b) Modelo Relacional

Essencialmente, banco de dados relacional consiste de uma ou mais relações em um formato bi-dimensional (linha e coluna). As linhas são chamadas de tuplas que corresponde ao registro e as colunas são chamadas de atributos e correspondem aos campos. Um ou mais atributos são considerados como atributo chave. É primordial manter a relação na terceira forma normal para evitar problemas de redundâncias e anomalias de armazenamento. Uma particularidade do modelo relacional é que as relações sempre estão na primeira forma normal, isto é, os valores de seus atributos são atômicos. Isso simplifica a teoria do modelo relacional, sua implementação e a linguagem de consulta que se baseia na álgebra relacional. Entretanto, a capacidade do modelo relacional é bastante restrita e isso é um dos motivos do grande desenvolvimento do modelo orientado a objetos.

c) Modelo Orientado a Objetos

Ultimamente os sucessores do modelo relacional são os modelos de dados orientado a objetos, representado pelo modelo orientado a objetos e por uma extensão do modelo relacional, também chamado de modelo relacional objeto. Eles foram desenvolvidos em resposta às complexas aplicações que não conseguiam ser representadas pelos modelos anteriormente apresentados. Tais aplicações incluem sistemas CAD/CAM, banco de dados de imagens, sistemas de informações geográficas, etc. O modelo de dados orientado a objetos é organizado em termos de objetos, suas propriedades e suas operações.

d) Outros modelos de dados

Um número razoável de outros modelos vem sendo desenvolvido para melhorar a representação do mundo real. Isso inclui o modelo de dados semânticos que introduziu a idéia de classe e subclasse dentro de uma modelagem de dados e os dedutivos ou banco de dados lógicos. O sistema dedutivo tem capacidade em especificar regras cujo mecanismo de inferência pode usar para derivar informações adicionais de dados armazenados nos banco de dados. Isso pode vir a influenciar no futuro no desenvolvimento e evolução de modelos banco de dados.

III.4.4. Incerteza em banco de dados convencional

Há uma variedade de dados incompletos e incertos representados através da introdução do conceito de valores nulos, sendo que a semântica de valores nulos compreende os valores desconhecidos, os valores não aplicáveis e os valores inexistentes. Entretanto, esses dois últimos valores não estão relacionados com incerteza.

A primeira extensão do modelo relacional que incorporou um conjunto de domínio não homogêneo não utilizou a teoria de conjunto fuzzy. Porém, ultimamente

a representação e recuperação de dados incertos e inexatos tem sido usado extensivamente na teoria de conjuntos e mapeamento do modelo relacional, permitido a representação e recuperação de dados imprecisos. Assuntos ligados à dependência funcional, considerações sobre implementação de segurança, e outros, também vem sendo estudado. Assim, a teoria de conjunto fuzzy vem sendo usada para representar imprecisão e incerteza em banco de dados relacional.

Nos últimos tempos o modelo relacional é considerado o modelo de banco de dados dominante e é perfeitamente natural a introdução da teoria de conjunto fuzzy no modelo de banco de dados relacional. Três abordagens vem sendo propostas para a introdução dessa teoria: a primeira utiliza o princípio de substituição da equivalência ordinária entre valor do domínio pela medida de proximidade tal como relacionamento de similaridade, relacionamento de proximidade e funções classificáveis (distinguível), e a segunda envolve uma variedade de estudo que utiliza distribuição de possibilidade dos valores de atributos. Existe também a combinação das duas propostas.

Um outra característica dessa abordagem é em relação a extensão do modelo relacional. Imprecisão e incerteza são estendidos para o modelo básico relacional não usando a primeira forma normal. Na primeira abordagem usando medidas aproximadas, a imprecisão do valor do dado é implícita, usando uma relação separada ou tabela de similaridade ou relacionamento de proximidade. Geralmente com o uso de distribuição de proximidade, muitas abordagens tem algumas descrições imprecisas dos dados explicitamente ou diretamente representada na base do valor do atributo da relação. Pode-se caracterizar essas abordagens como sendo ou representações homogêneas ou heterogêneas.

Uma característica representativa de um banco de dados relacional é a sua uniformidade ou homogeneidade de representação dos dados. Para cada domínio existe conjunto de valores prescritos para cada valor de domínio selecionado. Cada elemento do conjunto de domínio é da mesma estrutura, isto é, inteiro, real ou caracteres. Com o uso de similaridade ou da relação de proximidade, a imprecisão no valor do domínio é implícita e assim a representação permanece homogênea. Essas abordagens são usadas somente para modelos relacionais.

Para representar incerteza de maneira mais direta dentro de domínio de valores é necessário uma divergência da representação de homogeneidade. Os modelos baseados em teoria de possibilidade oferecem maiores capacidades para modelar outras formas de incerteza. Como esperado, nessa representação a implementação é mais complexa; a extensão mais complexa do modelo relacional básico leva a classifica-los como uma representação heterogênea.

Uma forma simples de um banco de dados fuzzy é a atribuição de um valor de pertinência (numérico ou lingüístico) para cada tupla. Isso permite a manutenção dos domínios de dados homogêneos e dos tipos de conjunto de dados, entretanto, o conteúdo semântico do domínio de pertinência é usado durante o processo de consulta

Teoria baseada em similaridade

A primeira abordagem em relação ao uso de relacionamento de similaridade num modelo relacional foi proposta por Buckles e Petry no final dos anos 70. Foi uma tentativa de generalizar o conceito de nulos e domínios multivalorados para uma implementação dentro de um ambiente operacional consistente com a álgebra relacional.

Para cada domínio, *j*, num banco de dados relacional, um conjunto base de domínio, Dj, é compreendido. Domínios para banco de dados relacional são ou escalares discretos, ou números discretos projetados para um conjunto finito ou infinito. Um exemplo de um domínio escalar é um conjunto de termos lingüísticos. O modelo fuzzy faz uso de um relacionamento de similaridade para permitir a comparação desses termos lingüísticos. O valor do domínio de uma tupla particular pode também ser um simples escalar ou números (incluindo valor nulo) ou uma seqüência de escalares ou números .

Definições básicas

Um aspecto importante de um banco de dados relacional fuzzy é que o valor do domínio não precisa ser atômico [96]. Um valor de domínio, d_i, onde i é o índice

do atributo na tupla, é definido para ser um subconjunto da base de seu domínio, D_i. Isto é, qualquer membro do conjunto pode ser um valor domínio, exceto nulo.

Uma relação fuzzy R é um subconjunto do produto do conjunto $\mathbf{P}(D_1)$ x $\mathbf{P}(D_2)$ x...x $\mathbf{P}(D_n)$. Pertinência em uma relação específica, r, é determinada pela semântica básica da relação. Como por exemplo:

> Se D_1 é o conjunto das maiores cidades e D_2 é o conjunto dos países
> Então (Paris, Bélgica) $\in$ $P(D_1)$ x $P(D_2)$, mas não é membro da relação (capital, país)

Uma tupla fuzzy, t, é qualquer membro de ambos r e $\mathbf{P}(D_1)$ x $\mathbf{P}(D_2)$ x...x $\mathbf{P}(D_n)$. Uma tupla arbitraria é: $ti = [d_{i1}, d_{i2}, \ldots, d_{im}]$ onde $d_{ij} \subseteq D_j$

Uma interpretação $\alpha = [a_1, a_2, \ldots, a_m]$ de uma tupla $ti = [d_{i1}, d_{i2}, \ldots, d_{im}]$ é qualquer valor atribuído tal que $a_j \in d_{ij}$ para todo j.

Resumindo, o espaço de interpretação é o produto do conjunto D_1 x D_2 x ... x D_n. Entretanto, para qualquer particular relação, o espaço é limitado pela validade da tupla do conjunto. Tuplas válidas são determinadas pela semântica da relação. Num banco de dados relacional, uma tupla é equivalente a sua interpretação. Relacionamentos de similaridade são usados para obter incerteza num valor de domínio, ambos lingüísticos e numéricos.

Uma operação fuzzy de álgebra relacional é igual a uma operação de álgebra relacional não fuzzy. Num banco de dados não fuzzy, a tupla é redundante se existir uma outra tupla igual. Qualquer operação sobre uma relação não fuzzy implica em remover redundância. Num banco de dados fuzzy, uma tupla é redundante se ela juntar-se com outra através da união do conjunto correspondendo ao valor do domínio. A união das tuplas, entretanto, está sujeita ao restrição na similaridade.

Num banco de dados fuzzy, cada tupla pode representar um grande número de interpretações. A perda de redundância num banco de dados não fuzzy é equivalente a ausência de múltiplas ocorrências da mesma interpretação. Portanto, dado uma interpretação de domínio, uma relação fuzzy poderá conter mais de uma tupla com essa interpretação.

Exemplo de múltiplas interpretações:

R1

Cor	Temperatura
Amarelo	Morna
Azul	Fresco
Preto	Frio
Vermelho	Quente

R2

Cor	Temperatura
{Amarelo, Vermelho}	{Morna, Quente}
{Azul, Preto, Amarelo}	{Fresco, Frio, Morna}

Teoria baseada em proximidade

Alguns aspectos da transitividade max-min numa similaridade podem causar dificuldades em modelar o relacionamento entre elementos do domínio. Pode ser difícil formular corretamente a propriedade transitiva do relacionamento. A característica essencial que produz a propriedade desejada e operação bem-definida é o particionamento do atributo do domínio pelo relacionamento de similaridade. O particionamento de um domínio escalar produz classes de equivalência ou clusters desarticulados. Esses clusters representam a informação não ideal armazenada no banco de dados. Entretanto, os clusters gerados não são fechados, sendo que nem todo elemento do cluster está relacionado com outro. Quando certos elementos das ligações são removidos do banco de dados, alguns clusters podem ser removidos, e o número

de clusters pode aumentar. Conseqüentemente, clusters removidos são variados, mas as tuplas também são introduzidas ou removidas do banco de dados. Ambos usos de relação de similaridade e de proximidade particionam um domínio escalar.

Teoria baseada em possibilidade

Já foi utilizado o conceito de similaridade ou proximidade, essa representação absorve a imprecisão em distintos elementos de conjunto de domínio dos atributos de uma relação.

No contexto do modelo de dados relacional, o uso da teoria de conjunto fuzzy tem sido defendido [96] por duas diferentes propostas (talvez complementares): para representar conceitos mal-definidos em termos de relação fuzzy (isto é, tuplas sobrecarregadas) e/ou informação incompleta dentro das tuplas.

Na teoria baseada em possibilidade, a informação disponível sobre o valor de um simples valor de atributo A para uma tupla t é representado pela distribuição de possibilidade $\pi_{A(t)}$ em $D \cup \{e\}$, onde D é o domínio do atributo A e *e* é um elemento extra que representa o caso quando o atributo não se aplica em t. A de distribuição de possibilidade $\pi_{A(t)}$ pode ser vista como uma restrição fuzzy de possíveis valores de A(t) e define um mapeamento de $D \cup \{e\}$ para [0,1].

Por exemplo, a informação João tem vasta experiência, é representada por:

$$\pi_{\text{EXPERIÊNCIA(João)}}(e) = 0, \text{ e } \pi_{\text{EXPERIÊNCIA(João)}}(d) = \mu_{\text{VASTA}}(d)$$

onde μ_{VASTA} é uma função de pertinência que representa o vago predicado VASTA no contexto. Os valores restritos à distribuição de possibilidade são considerados como exclusivos.

Muitos sistemas comerciais de banco de dados relacional assumem que qualquer relação para ser armazenada é preciso estar pelo menos na primeira forma normal, que significa que qualquer valor do atributo é atômico. Isso é feito para facilitar o armazenamento de dados dentro de colunas. Num ambiente de possibilidade para representar incerteza, um valor desconhecido é representado normalmente usando um conjunto. É importante distinguir entre a representação que

pode requerer vários valores e a noção de atomicidade que é conectada à semântica de um dado valor.

III.4.5. Incerteza em banco de dados orientado a objetos

A maioria dos estudos em banco de dados fuzzy trata de aspectos relativos a banco de dados relacional. As referências na literatura sobre incerteza no tratamento em aplicações de banco de dados são muito limitadas, principalmente no que diz respeito ao o problema de modelar fuzzy e incerteza no contexto de bancos de dados orientado a objetos. Porém, essa associação de fuzzy e incerteza com banco de dados orientado a objetos daria certamente uma expressividade semântica mais rica.

III.4.5.1. Modelo de banco de dados orientado a objetos

De uma maneira geral, há três tipos de necessidade de banco de dados: gerenciamento de dados, gerenciamento de objetos e gerenciamento de conhecimento. Somente gerenciamento de dados é atribuído ao sistema de gerencia de banco de dados relacional. Gerenciamento de objetos está ligado ao comportamento dos dados via métodos e é atribuída ao novo banco de dados orientado a objetos. Gerenciamento de conhecimento se refere ao escopo de derivação de informação sobre um domínio através de uma inferência.

O passo de um banco de dados relacional para orientado a objetos surgiu com o desenvolvimento de outros modelo de dados e de banco de dados, cada um oferecendo sua contribuição, mas nenhum deles se materializou num produto comercial. Por outro lado, numerosas extensões do modelo relacional seguiram para um caminho mais funcional e mais semântico. Entretanto nenhum dos novos modelos provou ser forte o bastante para substituir o modelo relacional como base para novos produtos.

Sistemas orientado a objetos possuiem uma grande vantagem em relação aos modelos anteriores, pois é um modelo de dados mais complexo que pode ser usado com um alto nível de abstração de dados [96].

Paradigma de orientação a objeto

O paradigma central da orientação-objeto é a noção de um objeto, que compreende o mundo real ou entidades abstratas e consistem de estruturas de dados encapsuladas, com possíveis complexidades, tendo capacidade comportamental. Baseado nessas propriedades de estrutura e comportamento, objetos pertencem a classes; as classes são organizadas dentro de uma hierarquia de super-classe e sub-classe. Cada classe possui operações que são procedimentos pré-definidos e são chamadas de métodos. Outra característica é o conceito de herança que permite a especificação de novas classes que herdam as suas operações e estruturas da sua super-classe. Portanto, sistemas de banco de dados orientado a objetos se valem da noção de objeto, classe e herança.

Neste modelo, cada objeto possui um único identificador que é usado para distinguir de outros objetos. Cada objeto é de um tipo particular, geralmente chamado de classe em linguagem de programação orientada a objetos. Assim o termo classe é usado para definir ambos objetos (estrutura e comportamento) e objeto extensão (conjunto de objetos). O termo classe tem também sido vagamente usado para referir a implementação de um tipo, assim um tipo pode ser mais do que uma especificação de implementação.

Alguns tipos conhecidos como tipos abstratos são colocados como supertipos de outros subtipos e podem não possuir objetos criados diretamente deles. Coleções de tipos são métodos de organização de dados pré-definidos tais como conjuntos, listas, arrays, etc. Objetos podem ser agrupados dentro de instâncias de uma dessas coleções.

O tipo objeto define a propriedade, ou seja, atributos e relacionamentos, e as operações que levam objetos desse tipo. Cada atributo tem um tipo. Atributos são classificados como simples ou complexos. Atributos simples levam a um valor literal. Atributos complexos são categorizados como atributos referência e coleção de atributos.

O outro tipo de propriedade, um relacionamento, define uma associação entre tipos de objetos. Um relacionamento é denominado como um atributo referência. O relacionamento especifica a cardinalidade, como pro exemplo, um relacionamento binário pode ser: 1-1, 1-n ou n-m.

Operações são funções que aplicadas nos objetos definem o comportamento do objeto. Uma operação é definida em duas partes: sua característica interna e externa, denominada a interface e sua implementação. São especificadas como um conjunto de operações e métodos. Uma operação define o nome da operação, o nome e tipo de qualquer parâmetro. O nome e o tipo retornam valores, e o nome de qualquer exceção que a operação pode causar. Um método de uma operação especifica a implementação da operação. Os métodos de um objeto são usados para encapsular a semântica do objeto.

Classe de objetos são ligados a outras classes de objetos em subclasse/superclasse hierárquico. A principal proposta dessa hierarquia é permitir a generalização que é também referenciada como herança, subclasse. Hierarquia pode ser considerada como uma especialização, especificação e classificação. Em especialização, as subclasses herdam as propriedade e operações da superclasse e podem escolher certas propriedades e operações da superclasse e/ou estender suas propriedades quando propriedades adicionais são necessárias pela subclasse. Especificação permite que um subtipo seja definido por um supertipo testando um predicado pré-definido sobre um conjunto de atributos. A distinção entre diferentes subtipos pode ser necessária. Em classificação, subtipos são simplesmente usados como conjunto de objetos para categorizar objetos. Especificação inclui classificação como um caso trivial. Assim um atributo específico pode ser definido indicando em qual subtipo ele pertence.

Herança múltipla ocorrem quando uma subclasse tem múltiplas superclasses. Heranças múltiplas são usadas quando existe mais que uma classificação ortogonal.

Base de dados orientado a objetos são persistentes, em contrastes com outros objetos que são transientes. O modelo objeto suporta multimídia e aplicações complexas e interagem com desenvolvedores de ferramentas baseadas em objetos.

III.4.5.2. Modelo orientado a objetos fuzzy

Muitas pesquisas e contribuições estão sendo realizadas na área de modelo de dados orientado a objetos fuzzy, algumas delas destacam aspectos de maneira mais particular, tais como: definição de classes fuzzy, herança fuzzy, esquemas fuzzy e outras estão centradas mais cuidadosamente no aspecto de implementação e em modelos de dados orientado a objetos fuzzy.

No modelo de dados orientado a objetos fuzzy existe uma proposta de um modelo gráfico objeto [11] suportando uma representação visual baseada em gráficos. As informações imprecisas, vagas e incertas são modeladas separadamente; elementos gráficos são definidos para representar e identificar informações vagas e incertas.

O modelo de dados baseado em gráfico fornece um caminho natural para manipulação de dados que aparecem nas aplicações tal como hipertexto ou sistema de informação multimídia.

Num modelo objeto baseado em gráfico o esquema de banco de dados e as instâncias podem ser vistas como gráficos com linguagem de manipulação apresentadas em forma de transformações gráficas. Laucarella e Zanzi propuseram um modelo objeto baseado em gráfico que vem sendo usado na modelagem conceitual e diretamente na manipulação (browsing, consultas e visões) dos objetos armazenados.

Apresenta-se um resumo da definição formal do modelo objeto baseado em gráfico, como mostra o exemplo da figura III.11, e uma extensão fuzzy desse modelo.

O esquema conceitual $\sum$ é definido com as seguintes quintuplas: {C, T, A, H, P). Onde:

- C - é um conjunto finito das classes
- T - é um conjunto finito de tipos

- A - é um conjunto de atributos

- H – é a herança parcial na relação

- P – é a propriedade da relação

O esquema conceitual pode ser representado como: $G(\Sigma) = (N,E)$. Onde:

- N – é o conjunto de nós

- E – é o conjunto de arco

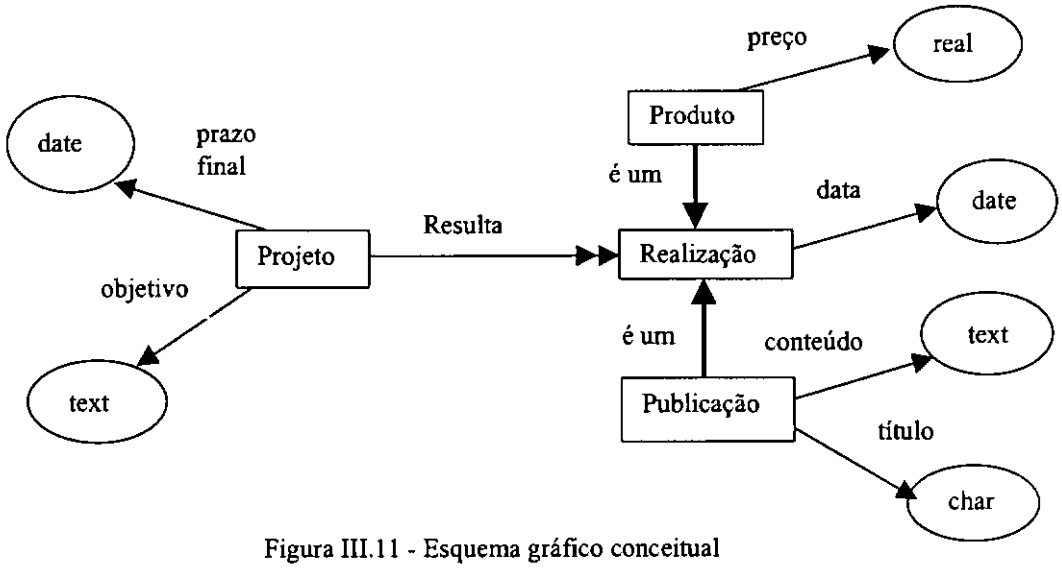


Figura III.11 - Esquema gráfico conceitual

Nesse exemplo (figura III.1.1) tem-se:

C={projeto, produto, realização, publicação}

T={real, date, text, char}

A={objetivo, prazo final, data, conteúdo, título}

H={(produto, realização), (publicação, realização)}

O modelo objeto baseado em gráfico tem sido projetado para manipular informações precisas, ou seja:

- valores de dados precisos;

- propriedades de objetos precisos;

- classificação de objetos precisos;

- hierarquia de classe precisas,

e manipulam somente consultas precisas, ou seja:

- seleção de condições precisas nos valores dos dados
- seleção de condições de agregação booleana

Este modelo não está disponível para manipular informações imperfeitas. Num banco de dados orientado a objetos, informação imperfeita sobre uma realidade considerada pode ser modelada em diferentes níveis (por exemplo, com os níveis de atributos, propriedades, associações de objetos com classes ou definição de classe hierárquica), dependendo da imperfeição da informação. Entretanto é importante distinguir o tipo de imperfeição para representar um contexto formal adequado.

Para o caso onde ocorre a modelagem de incerteza e vago ao mesmo tempo: esse caso pode ser realizado com o armazenamento de valores vagos dos atributos, por exemplo, a idade de José está entre 20 e 25 anos (jovem), que vem a ser um atributo fuzzy, como também pela associação de um grau de incerteza para a propriedade de José com sua idade. Portanto, quando a informação falhar, ela pode ser copiada introduzindo imprecisão nos valores dos dados ou incerteza na associação do valor para o dado, em geral tanto os valores dos dados vagos e quanto incerteza na associação do valor do dado pode coexistir.

É então necessário dispor o modelo separadamente:

- informação vaga e precisa;
- certeza e incerteza em informação precisa e vaga;

Assim esse modelo é baseado numa representação gráfica do banco de dados em que cada tipo de informação (isto é, objetos, valores de atributos, relações relação hierárquica) tem sua própria representação (retângulo, setas, etc.), oferecendo uma representação gráfica de incerteza.

Uma característica que tem sido introduzida é a associação de diferentes qualificadores com a propriedade relacionamento, ou seja, incluindo um grau lingüístico indicando o qualificador. Nesse caso o grau não expressa a incerteza na informação, mas o qualificador do relacionamento. Para evitar ambigüidade é

necessário expressar incerteza através de um grau numérico, considerando que o qualificador é expresso por variáveis lingüísticas, tais como: *alto, baixo, médio.*

Um qualificador pode ser associado também com a instância do relacionamento e a herança do relacionamento. Essas duas situações podem ser modeladas através da definição da classe fuzzy e hierarquias de classe fuzzy; esses dois conceitos estão interrelacionados; de fato, quando uma hierarquia fuzzy de classes fuzzy é definida, a pertinência de um objeto para uma subclasse é afetada pelo relacionamento fuzzy - "é um" - com a superclasse.

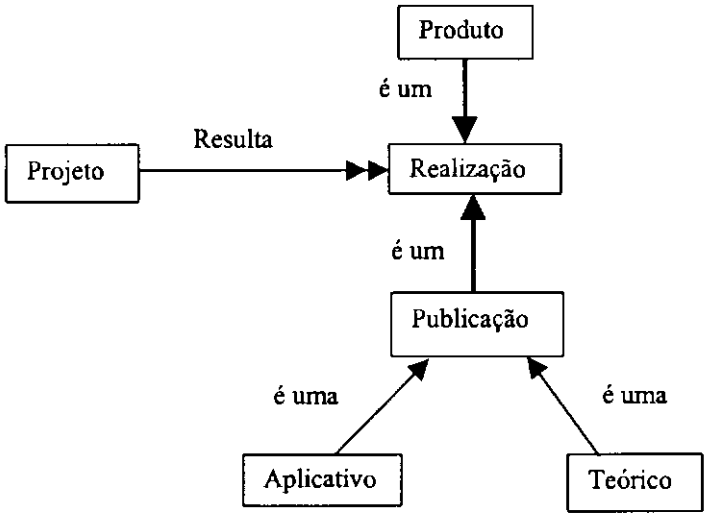


Figura III.12 - Exemplo de um esquema gráfico

A classificação das publicações do exemplo do esquema gráfico (figura III.12) como aplicativo ou teórico não é muito clara, é mais uma classificação fuzzy. No contexto de um artigo pode haver aspecto teórico ou aplicativo, e então pode possuir diferentes graus para a subclasse de publicação aplicativa e para a subclasse de publicação teórica. Esse grau de pertinência para as subclasses não depende da relação "é uma", mas depende de uma restrição fuzzy imposta no atual valor de atributo da publicação.

Outro caso de qualificador da associação de um objeto com uma classe ocorre quando deseja-se definir uma classe-subclasse hierárquica. Hierarquia fuzzy é planejada como significado para organizar (classificar) objetos em subclasses fuzzy que especializa ou generaliza em diferentes níveis o conceito de superclasse fuzzy. No caso de especialização a extensão dessa subclasse fuzzy é incluída na extensão de superclasse fuzzy e o contrário ocorre para o caso de generalização.

Valores de atributos vagos são representados pelo significado de possíveis distribuições π no atributo de domínio. Esses valores permitem modelar situações em que o valor preciso de um dado atributo é desconhecido.

Tv é o conjunto de nomes do tipo vago: Tipo vago são caracterizados por valores vagos ou imprecisos, que são representados pela distribuição de possibilidade. Para identificar o tipo vago no esquema gráfico estendido G($\sum$e) cada t$\in$Tv é representado por uma forma nebulosa de nó classificada de t. Te é então definido como Te=T$\cup$Tv um conjunto finito de tipo de nome. Cada t$\in$Te denota um tipo de coleção de objetos primitivos cujo valores são identificados por V(t).

Como no exemplo da figura III.11, os atributos podem ser vagos, assim o valor do tipo text, char, real, date $\in$ Tv foi introduzido para denotar o nome do domínio das propriedades de conteúdo, data, prazo final, etc. No caso de atributo com domínio de valor numérico tal como preço e data, os valores vagos podem ser expressos pelas variáveis lingüísticas: T(preço) = {muito baixo, baixo, muito alto, alto}, T(data) = {muito recente, recente, não muito recente}. Quando o domínio de um atributo não é numérico, como no caso de conteúdo, dados vagos são representados através de uma distribuição de possibilidade, mas não pode ser sintetizado por um valor lingüístico.

No modelo de dados gráfico orientado a objetos fuzzy, a linguagem de consulta poderá contar com a facilidade de uma navegação com visualização gráfica no banco de dados, especificando diretamente os passos que devem serem percorridos

na tela. Isso possibilita projetar uma linguagem visual de consulta baseada em conceitos de perspectiva e transformação gráfica.

III.4.6. Modelos alternativos para banco de dados fuzzy

III.4.6.1. Extensões fuzzy para o modelo E-R

O modelo Entidade-Relacionamento foi desenvolvido por Chen e vem sendo aceito para projetos de banco de dados como um modelo de ótimo potencial; vem sendo também estendido e modificado para um número de aplicações tais como ferramentas CASE e atualmente não existe uma padronização do modelo E-R, mas todos usam um conjunto básico de construtores.

Primeira extensão

No desenvolvimento de um modelo E-R tem-se considerado três níveis fuzzy. No primeiro nível, o modelo conceitual é fuzzy (o conjunto fuzzy é o conjunto de objetos semânticos). Isto é, um conjunto de entidades, um conjunto de relacionamentos ou um conjunto de atributos pode ser fuzzy. Isso pode se manifestar durante o projeto de banco de dados ou projeto de modificação do esquema conceitual de banco de dados. Se o conjunto de objetos semânticos é fuzzy, obtém-se conjuntos de entidades fuzzy, conjuntos de relacionamentos fuzzy e conjuntos de atributos fuzzy. Pertinência de um conjunto de entidade ou de um relacionamento no modelo é quando o grau de pertinência é menor que 1, existe uma possibilidade que é acompanhada por uma incerteza. No segundo nível, ocorrências específicas de entidade ou relacionamentos são atribuídas para ser fuzzy. Finalmente no terceiro nível, atributos de específicas entidades ou relacionamentos podem ser fuzzy. Com atributos fuzzy, a importância de incluir o domínio como parte do diagrama E-R é enfatizado. Esses três níveis são considerados independentes e qualquer combinação deles é possível.

Para capacitar manipulação de dados fuzzy foi proposta uma versão num modelo E-R e adaptada para manipular objetos fuzzy. Além disso generalizando os operadores convencionais (por exemplo, união, diferença e seleção) para o caso fuzzy, alguns novos operadores especificamente para manipulação de informação fuzzy foi proposto: eliminação fuzzy, distância fuzzy, cardinalidade fuzzy.

Segunda extensão

Esta outra extensão trabalha com a estrutura básica de E-R com representação de incerteza e imprecisão. Uma extensão de conjunto de entidades foi desenvolvida para permitir uso de subconjuntos do domínio da definição de certas entidades. Essa idéia de subconjuntos é usada como oposição a um possível conjunto de domínio.

Uma introdução explícita de qualificadores lógicos fornecem uma adicional capacidade de representação de conhecimento sobre o estado e o comportamento do sistema. Adicionalmente para aproximar, qualificadores fuzzy tal como: vários ou muito, mais operadores foram considerados. Isso inclui *exatamente um*, *exatamente n*, ou *entre m e n*. Eles indicam precisamente o número de objetos e valores de propriedades que são associados por um relacionamento particular.

Terceira extensão

O modelo de extensão fuzzy E-R é uma adaptação do modelo E-R incluindo subclasse e superclasse e conceitos de generalização e especialização. É baseado na lógica proposicional fuzzy usando o conceito fuzzy. Conjuntos de entidades e relacionamentos são representados como conjunto fuzzy generalizados. Um valor fuzzy é associado com a instância da entidade ou relacionamento que indica a proposição de pertinência de cada instância num conjunto.

Atributos ou componentes de atributos podem ser modelados pelo conjunto fuzzy. Relacionamentos tal como entre essas subclasses e superclasses trabalham com a representação de restrição no relacionamento também como proposição fuzzy.

III.4.6.2. Extensão fuzzy para o modelo de dados de rede

Modelo de dados de redes não tem recebido o nível de atenção como o do modelo relacional em pesquisas de banco de dados fuzzy. O modelo de rede não possui o foco da pesquisa quanto a imprecisão. Isso é devido a perda de formalismo na sua descrição que é particularmente evidente quando comparada com o modelo relacional.

A especificação mais expressiva de um modelo de rede é chamada de modelo de dados DBTG, publicado pela Data Base Task Group da Conference on Data System Languages (CODASYL) em 1971. O objetivo original foi, em geral, para especificar facilidades de um SGBD e, em particular, propor uma facilidade para esse SGBD em COBOL. O modelo de rede DBTG consiste em: um conjunto de tipo de registro $\{R_A, R_B, \ldots, R_N\}$, sendo que um tipo de registro é usado para representar entidades; um conjunto de nomes ligados $\{L_{IJ}\}$ conectando um registro do tipo R_I para zero ou mais tipo de registro R_J, em essas ligações são usadas para representar o relacionamento entre duas entidades; toda ligação é funcional numa única direção e nenhuma ligação sob a forma L_{II} é permitida. Uma das tentativas para utilizar a teoria de conjunto fuzzy em ambiente de rede foi atribuir ao DBTG um conjunto fuzzy.

Para resumir, os métodos mais diretos para aplicação da teoria de conjunto fuzzy são impedidos pela condição de funcionalidade. Duas alternativas podem ser propostas. Uma preserva a estrutura hierárquica expressando a mais complexa especificação de consulta e a outra dilui a estrutura hierárquica para manter uma simples especificação de consulta.

Assim o grande obstáculo para incorporar o conceito de conjunto fuzzy no modelo de dados DBTG é a condição de que a ligação pai-filho precisa funcionar somente em uma direção. Modelos mais gerais de redes que permitem relacionamento M:N permitiriam uma representação mais natural de imprecisão. Assim o modelo de dados de rede hierárquica modelando capacidades são amplamente substituídos (submetidos a uma outra idéia) pela atual abordagem de orientação objeto.

III.4.7. Evolução da incerteza nos bancos de dados

Uma necessidade do uso de lógica fuzzy em sistemas de informação vem sendo realizada na pesquisa de armazenamento de informação e recuperação de sistema. A maioria dos esforços em usar lógica fuzzy em banco de dados vem sendo enfocada nos casos de modelos de dados. Existem três aspectos de banco de dados que podem utilizar a lógica fuzzy: projeto de esquema, modelagem de dados e sistemas de consulta.

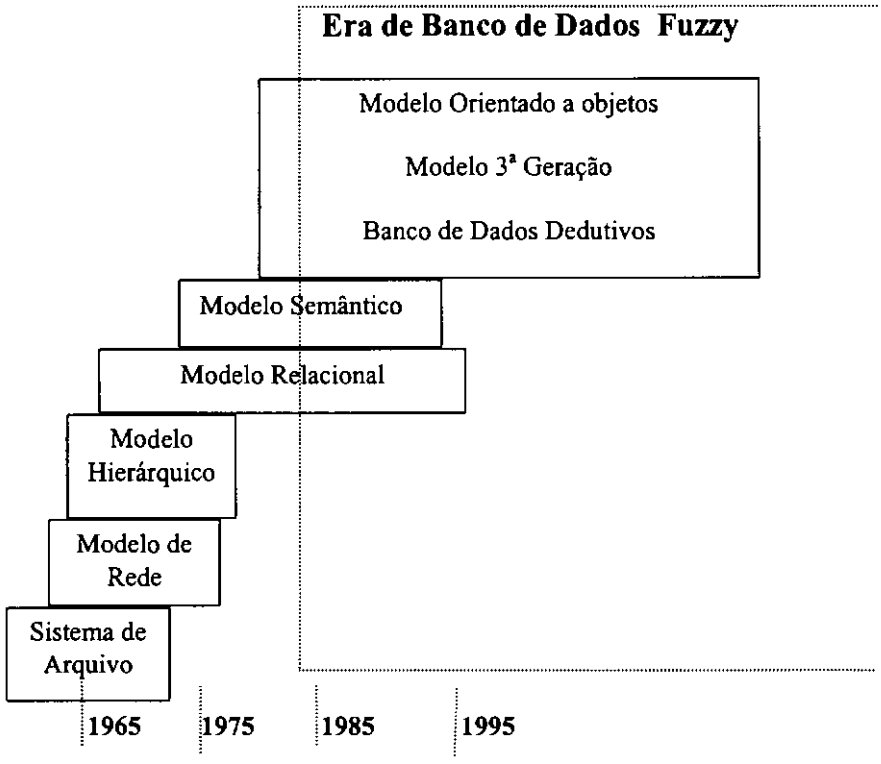


Figura III.13 - Evolução dos Modelos de Dados

Atualmente tem-se uma variedade de abordagens alternativas para banco de dados fuzzy, isso inclui modelos de banco de dados relacional em que a teoria do conjunto fuzzy tem sido aplicada para modelar incerteza, modelo E-R, modelo de rede e banco de dados orientado a objetos.

Banco de dados de rede foi de grande importância antes do modelo relacional, sendo dominante nos anos de 1980. Porém, introduzir incerteza de abordagem fuzzy

na sua estrutura causa certas restrições, pois o modelo de dados de redes é bastante limitado para ter uma abordagem fuzzy.

A abordagem de entidade-relacionamento é mais considerada como parte de um projeto de banco de dados, por outro lado pode ser considerada como um modelo conceitual. Não existe, no entanto, um SGBD usando diretamente um modelo E-R como modelo lógico, sua importância está mais ligada com a imprecisão da descrição.

A introdução de conceitos de incerteza em modelos orientado a objetos é a mais nova direção em SGBD. A desvantagem está relacionada com a inexistência de uma padronização desse modelo, pois a mesma ainda não foi formalizada e portanto é difícil desenvolver uma abordagem generalizada. Entretanto, existe muita ênfase em modelagem de orientação objeto fuzzy.

A figura III.13 apresenta a progressão dos modelos de banco de dados, mostrando a era de banco de dados fuzzy, indicando o relacionamento no tempo em que começou o desenvolvimento de banco de dados fuzzy [96].

III.5. Árvore de Decisões de Dados Fuzzy

Recentemente, tem-se investido grande esforço na construção de árvores de decisão de dados fuzzy, onde o principal [41] propósito é explorar o algoritmo de árvore de decisão juntamente com a habilidade da representação fuzzy possibilitando o aumento de representatividade do conhecimento. A árvore de decisão constitui uma das metodologias mais populares de aquisição de conhecimento simbólico, pois com um simples mecanismo de inferência é possível obter uma representação altamente compreensível, e a representação fuzzy trabalha com informações imprecisas e inexatas, porém essa combinação não é considerada muito trivial.

Uma árvore de decisão fuzzy difere de uma árvore de decisão tradicional em dois aspectos: o primeiro utiliza critérios de divisão baseados em restrições fuzzy e o segundo é que o procedimento de inferência é diferente.

Uma extensão do algoritmo ID3 foi realizada em recente pesquisa [41] para trabalhar com árvore de decisão fuzzy, utilizando conjunto fuzzy e raciocínio

aproximado para a construção da árvore e o procedimento de inferência. Um exemplo em ID3 (baseado em dados crisp) satisfaz exatamente uma das possíveis condições de saída de um nó. Em árvore de decisão fuzzy, um exemplo pode acoplar mais do que uma condição, pois essas condições são agora restrições fuzzy baseadas no conjunto fuzzy. Por isso, um dado exemplo pode falhar em muitos nós filhos de um nó pai. Isso pode acontecer em qualquer nível. Uma combinação fuzzy é qualquer número em [0,1], um nó folha pode conter um vasto número de exemplos, muitos dos quais podem ser encontrados em outros níveis também, com graus diversos em [0,1]. Esse fato é vantajoso, pois oferece um comportamento oportuno, especialmente quando se trabalha com informação imprecisa e vaga.

A rotina de construção de uma árvore permite que o algoritmo ID3 seja avaliado usando conjunto fuzzy, relacionamento, e método de raciocínio. Isso significa que um número de exemplos que falharem dentro de um nó são agora baseados em combinações fuzzy. Esses são os conjuntos fuzzy associados com variáveis lingüísticas usadas nessas condições, denominadas de restrições fuzzy. O procedimento de decisão de um nó folha pode ser descrito da seguinte forma:

> "SE restrição fuzzy permitir que um nó folha esteja satisfeito,
> ENTÃO inferir a classificação de treinamento do dado no nó folha".

As probabilidades são estimadas através de exemplos contabilizados em nós individuais. Em árvore de decisão fuzzy, o relacionamento do exemplo num dado nó é um número real [0,1], indicando a satisfação da restrição fuzzy permitida para esse nó. O nível de satisfação em árvore de decisão fuzzy cresce drasticamente. Assim os conflitos são controlados pelos operadores.

Múltiplos níveis podem ser combinados num exemplo causado outro nível de conflito. Esses conflitos resultam de uma sobreposição do conjunto fuzzy, e é agravado pelo conhecimento incompleto, etc. Por isso cada nível pode ser interpretado como uma regra fuzzy, assim múltiplos níveis possuem a interpretação disjuntiva. Muitas possibilidades de inferências ainda estão sendo pesquisadas, e as mesmas idéias podem levar a um número novo de inferência para árvore de decisão fuzzy.

Aparentemente a grande vantagem da tecnologia de árvore de decisão fuzzy se encontra na utilização do mesmo procedimento de árvore de decisão simbólica, acrescentando apenas a representação fuzzy. Isso permite a utilização da mesma estrutura de árvore para compreensão e verificação do conhecimento, e também possibilita ter um processamento mais robusto e com continuidade.

III.6. Considerações finais

Diariamente a informação é caracterizada por possuir algum tipo de imperfeição, incerteza e inconsistência. A imprecisão está relacionada com o conteúdo de uma proposição, mais precisamente trabalha com a granularidade do valor de alguns atributos envolvidos na proposição com relação ao universo do discurso. A incerteza está relacionada com a verdade da proposição, planejada com a conformidade da informação realizada pela proposição com a realidade considerada; expressões lingüísticas tal como *provavelmente, é possível*, são normalmente usadas para declarar a ignorância sobre a verdade do estado da informação na proposição. E a inconsistência vem de simultâneas situações presente na informação contraditória sobre a mesma realidade.

Já no primeiro momento do estudo de caso, após a visualização dos dados, pode-se perceber que além da complexidade e relevância dos dados disponíveis para análise a ser realizada, existe também situações em que as informações estão completamente afetadas pela incerteza, imprecisão e inconsistência. Levando em conta esse fato, foi então detectado a importância de um estudo do tratamento de conceitos inerentes, nebulosos, vagos, na tentativa de representá-los e tratá-los, tentando, no entanto, agrupar, identificar e explicar suas diversas formas e fontes através da teoria fuzzy.

Ressaltou-se ainda neste capítulo, a existência de uma variedade de abordagens alternativas para banco de dados fuzzy, incluindo os modelos de banco de dados como: o modelo de redes, o modelo E-R, o modelo relacional e o modelo orientado a objetos.

É de grande importância ainda a possibilidade de construir árvores de decisão diretamente dos dados incertos, pois esse novo paradigma ultrapassa o vazio entre os dados simbólicos e não-simbólicos através da associação qualitativa do termo lingüístico com dados quantitativos. Embora este enfoque ainda esteja em processo de pesquisa e estudos empíricos, vale ressaltar que essa representação compreenderá uma crescente aplicabilidade em contextos imprecisos, incertos, vagos, etc. Ao mesmo tempo, é possível proliferar uma estrutura tão compreensível quanto a estrutura de árvore simbólica, tendo como principal componente resultante o conhecimento.

CAPÍTULO IV

FUZZY QUERIES

IV.1. Introdução

Há algumas décadas, organizações das mais diversas áreas vem acumulando uma infinidade de informações. No começo dos anos 70, surgiu a tecnologia de banco de dados relacional, onde além de armazenar dados, tinha ainda como finalidade a administração dessa massa de dados e a produção de informações utilizáveis. A partir de então, vários métodos foram sendo projetados para trabalhar com a recuperação das informações dos bancos de dados. O mais conhecido desses métodos é a álgebra realcional, chamada de linguagem de consulta estruturada (SQL). E é sem dúvida até hoje, a linguagem mais usada para recuperar informação de dentro de banco de dados relacional.

Entretanto, a linguagem SQL possui algumas limitações com relação a tratamento de dados vagos, imprecisos, nebulosos, etc. Um SGBD relacional é projetado para processar consultas exatas, recuperando apenas os registros que possuem tal característica. Entretanto, se não existem registros na base de dados obedecendo a uma determinada restrição de consulta, então a recuperação desses registros passa a ser impossível.

Nos últimos anos, um grande número de pesquisadores da área vem estudando novos métodos para ultrapassar essas limitações da linguagem SQL. Um desses novos estudos vem sendo realizados na área de recuperação de dados, onde suas funcionalidades são estendidas para permitir consultas flexíveis.

Este capítulo destina-se a visualizar as possíveis abordagens direcionadas na área de acesso às informações, começando pela natureza e organização do sistema relacional, focalizando a linguagem SQL e, portanto, introduzindo os principais conceitos que envolve o uso da lógica fuzzy na formulação de consultas imprecisas sobre banco de dados, mais especificamente sobre a linguagem de SQL.

IV.2. Sistema de Gerenciamento de Dados (SGBD) Relacional

A maioria dos sistemas de banco de dados desenvolvidos nos últimos anos são relacionais. Assim, muitas das pesquisas atuais baseiam-se em conceitos relacionais. De fato, não há dúvida de que a abordagem em OR (object relational) representa atualmente a tendência dominante e que o modelo relacional representa o mercado dominante hoje e é simplesmente o desenvolvimento mais importante de toda história da área de banco de dados.

Base de dados relacional é visto como um conjunto de tabelas, sendo que as tabelas são compostas por linhas e as linhas são compostas por colunas. As linhas das tabelas são também chamadas de tuplas, como mostra a figura IV.1. As tabelas são manipuladas por um conjunto de operações, realizadas através do uso da álgebra relacional, não sendo procedurais, ou seja, manipulando conjuntos de uma só vez. Os relacionamentos entre as tabelas são estabelecidos dinamicamente pela conexão das tabelas.

O projeto de banco de dado relacional é centrado em torno do conceito de formas normatizadas, obedecendo a um conjunto específico de limitações. Como por exemplo, diz-se que uma tabela encontra-se na primeira forma normal se, e apenas se, satisfizer a limitação para que contenha somente valores atômico.

Linhas

CÓDIGO	NOME	TIPO	ANO
099-987	Vinho A	Tinto Suave	1981
012-345	Vinho B	Branco Seco	1977
090-909	Vinho C	Tinto Seco	1966
089-777	Vinho D	Branco Doce	1977
090-676	Vinho E	Tinto Suave	1977
090-679	Vinho F	Tinto Seco	1976

Coluna

Figura IV.1 – Tabela de Vinhos

Assim, um SGBD relacional contém muitas tabelas criadas e manipuladas de forma independente uma das outras. As tabelas são adicionadas ou removidas do banco de dados ou podem também ter suas estruturas alteradas. Em sistemas de banco de dados em produção existem outras facilidades que ajudam a melhorar a

performance. Essas facilidades incluem índices, domínios explícitos, controle de segurança, permissão de acesso, monitoramento e reorganização do banco de dados, facilidades de compressão, enfim inúmeros gerenciamento de controles.

A linguagem SQL oferece um conjunto de instruções para definir e manipular um banco de dados relacional. A definição e manutenção das tabelas são feitas através dos comando de CREATE, DELETE, INSERT e UPDATE, enquanto que a manipulação da informação é realizada através da utilização do comando SELECT.

A sintaxe completa do comando SELECT pode ser complexa e extensiva, envolvendo múltiplas tabelas, sub-consultas, funções, etc. A sintaxe básica da consulta é a seguinte:

```
SELECT col1, col2, ..., coln
        FROM tabela1, tabela2, .., tabelaj
        WHERE expressão1, [AND/OR] expressão2, ..., expressãoj
```

Onde:
SELECT: seleciona um conjunto de colunas de uma tabela específica a ser apresentada na consulta.

FROM: uma lista das tabelas.

WHERE: processo que expressa as restrições da consulta. Tem-se duas funções:

1) Seleciona linhas de uma tabela sob uma específica condição
2) Tabelas são conectadas baseadas numa específica comparação de condição

Exemplo de consulta da tabela de Vinhos:

```
SELECT nome, ano, tipo
        FROM Vinho
        WHERE ano = 1977;
```

NOME	ANO	TIPO
Vinho B	1977	Branco Seco
Vinho D	1977	Branco Doce
Vinho E	1977	Tinto Suave

A linguagem SQL é claramente considerada como uma linguagem de alto nível. Entretanto, ela pode perder na habilidade para recuperar a semântica da consulta, sendo que a consulta em SQL captura apenas uma coleção de registros crisp. Assim, métodos de acesso compartilham o mesmo tipo de restrição numa expressão e encontram limitações para recuperar informações imprecisas.

IV.3. Consultas flexíveis em banco de dados

Muitas vezes nas consultas de banco de dados deseja-se expressar uma proposição próxima de uma linguagem natural e assim distingui-las dentro de um contexto de valores aceitáveis. Para isso, várias soluções baseadas na representação de predicados vagos vem sendo sugeridas, através da teoria de conjuntos fuzzy. A consulta flexível possui a vantagem de sempre fornecer uma resposta e colocá-las de forma ordenada com as N melhores respostas. Atualmente, essas consultas ditas flexíveis, envolvendo critérios vagos são representadas pela lógica fuzzy, sendo que são definidas usando vários tipos de construtores flexíveis.

Os sistemas estruturados de forma relacional que vêm utilizando os conceitos da teoria de lógica fuzzy, possuem todo o seu desenvolvimento baseado na extensão da álgebra relacional. A sua sintaxe estrutural é modificada para compreender as especificações necessárias, pois o uso de cálculo relacional fornece uma especificação não procedural para uma consulta e pode ser estendida facilmente para uma linguagem de mais alto nível.

IV.3.1. Consultas fuzzy baseadas no conceito de similaridade/proximidade

Consultas baseadas em similaridade atuam em bancos de dados relacionais, através da realização de uma consulta contendo maior flexibilidade nos critérios de seleção dos dados sobre um conjunto vazio de linhas, transformando o conjunto em outro similar. Para isso, é utilizada a teoria de lógica fuzzy na criação da consulta.

Como já mencionado, um SGBD relacional é projetado para processar

consultas exatas, como por exemplo:

```
Select *
    From T
    Where nome = Maria,
```

recuperando apenas os registros que possui tal característica. Por outro lado, se não existir registros nesse banco de dados com atributos específicos, então não é possível recuperar os registros com atributos similares.

Com um SGBD relacional não é possível expressar consultas por similaridade. O principal objetivo de usar uma consulta por similaridade é obter resultados onde as consultas tradicionais falham. Normalmente uma consulta utilizando SQL obtêm dados com grau de pertinência igual a 1, enquanto uma consulta fuzzy por similaridade também considera dados com grau de pertinência menor do que 1.

A técnica que realiza uma consulta contendo maior flexibilidade nos critérios de seleção dos dados sobre um conjunto vazio de linhas é denominada α-cut e trabalha comparando o intervalo de busca definido (mínimo a máximo) e o domínio do atributo (valor mínimo a valor máximo). Para α-cut =1, o intervalo de busca é igual ao indicado pelo usuário. Para α-cut = 0, o intervalo de busca considerado é todo o intervalo de valores possíveis do atributo no banco de dados.

IV.3.1.1. Processo de avaliação de consulta

Uma consulta $C(a_i, a_h, \ldots, a_k)$ é uma expressão de um ou mais fatores combinados por operadores booleanos disjuntivos ou conjuntivos: V_i op V_h op ... op V_k. Tem-se de maneira ordenada para formar com a relação r, o conjunto de domínio $D_1, D_2, \ldots, D_m$, onde cada valor V_j precisa ser:

1) um elemento de domínio a, a $\in D_j$, onde D_j é um conjunto de domínio para r, ou;

2) um elemento de domínio modificado por uma ou mais variáveis lingüísticas (qualificadores), por exemplo: "Não", "Muito", "Mais ou Menos".

A relação r pode ser uma relação original do banco de dados ou pode ser obtida a partir do resultado de uma série operações de álgebra relacional fuzzy. A semântica fuzzy se aplica as operações e qualificadores. Como por exemplo:

"Mais ou menos" GRANDE e "Não muito" PESADO

onde: GRANDE é igual a abreviação do termo TAMANHO=GRANDE na relação tendo um domínio denominado TAMANHO. O valor PESADO também é uma abreviação.

Um valor de pertinência de uma tupla representado num relação r é atribuída de acordo com a possibilidade de seu acoplamento com uma especificação de consulta. Fazendo $a \in D_j$, ser um elemento arbitrário. O valor de pertinência $\mu_a(b)$, $b \in D_j$, é definido baseado na relação de similaridade, $s_j(a,b)$, sobre o domínio. A consulta $C(\bullet)$ induz a um valor de pertinência $\mu_C(t)$ para uma tupla t em resposta a r, da seguinte forma:

1) Cada interpretação $I = [a'_1, a'_2, \ldots, a'_m]$ de t determina um valor $\mu_{aj}(a'_j)$ para cada elemento de domínio aj, de C(aj, ah, ..., ak).

2) Avaliação dos qualificadores e operadores em $C(\bullet)$ sobre o valor de pertinência $\mu_{aj}(a'_j)$ produz $\mu_C(I)$, que é o valor de pertinência da interpretação com relação a consulta.

3) Finalmente, $\mu_C(t) = \max_{I \text{ de } t} \{\mu_C(I)\}$

Resumindo, o valor de pertinência de uma tupla representa a melhor interpretação. A relação resultante é então o conjunto de tuplas tendo um valor de pertinência diferente de zero. Na prática, isso pode ser real considerando somente a tupla com valor alto.

IV.3.1.2. Cálculo relacional

A abordagem em torno do cálculo relacional de um modelo de banco de dados de similaridade utiliza a formulação de cálculo para consultas que tendem a ser fechadas em uma linguagem natural. Uma tupla de formulação de cálculo contém ocorrências de diferentes tuplas variando com a condição join. O cálculo fuzzy permite a especificação de uma relação que pode ter um número infinito de tuplas; isso significa que não é protegido. O cálculo é garantido se cada sub-fórmula criada especificar uma relação finita.

Uma linguagem que possibilita simular o cálculo de tupla, ou equivalente, álgebra relacional, ou cálculo do domínio, é considerada uma linguagem completa. Numa sub-linguagem de dados, a perfeição é a capacidade de se obter todo relacionamento do banco de dados. Num banco de dados relacional isso proporciona uma habilidade para construir qualquer relação através das operações e das suas composições. Num banco de dados relacional fuzzy, o relacionamento de similaridade também possui a capacidade de especificar qualquer partição de um conjunto de domínio com perfeição, ou seja, é possível especificar fuzzy em toda qualquer relação resultante ou intermediária.

IV.3.1.3. Características fuzzy para banco de dados baseado em similaridade

a) Dependência funcional fuzzy

As abordagens relacionadas com bancos de dados relacionais fuzzy vêm desenvolvendo extensões de dependências funcionais. Existe dois tipos de dependência para similaridade: crisp e fuzzy. Para compreender melhor o conceito de dependência funcional em similaridade para banco de dados, primeiro é dado a definição para tupla-similaridade entre duas tuplas de uma relação de um certo domínio.

Tupla-similaridade

Para qualquer domínio Dk de uma relação R, pode-se definir tupla-similaridade para as tuplas t_i e t_j como:

$$Ts\ [\ Dk(t_i, t_j)] = Min\ (s(p,q)$$
$$p, q \in d_{ik}\ V\ dj_k$$

onde, a relação $d_{ik}(d_{jk})$ é o conjunto de valores presentes na tupla (t_i,t_j) para o domínio Dk. No caso de banco de dados relacional, tanto d_{ik} quanto o d_{jk} possuem o mesmo conjunto e a tupla-similaridade possui valor 1 ou 0. Para tupla-similaridade , o valor é igual a 1 para duas tuplas numa relação que possuem o mesmo valor de domínio, caso contrário, o valor é igual a 0.

Dependência crisp

Essa é uma extensão direta de dependência funcional de banco de dados relacional para banco de dados fuzzy. Se A e B são dois atributos de uma relação R, então $A \rightarrow B$ é satisfeita se:

$$Ts\ [A(t_i,t_j)] \geq Ts\ [B(t_i,t_j)],\ \forall\ t_i,\ t_j$$

Essa expressão pode ser também aplicada para qualquer banco de dados relacional, sendo que a expressão é satisfeita somente se, para qualquer duas tuplas, valores de domínio A e de domínio B para o mesmo par de tuplas são os mesmos. Conceitualmente, para um banco de dados fuzzy, A determina B, se duas tuplas na mesma relação apresentam valores de domínio B fechados para cada valor do domínio A.

Dependência fuzzy

Representa o relacionamento entre diferentes domínios da mesma relação existente, mas não é forte como uma dependência crisp. Dependência fuzzy entre o domínio A e B é representado como $A \rightarrow \partial B$, onde ∂ representa o fuzziness da

dependência. O valor dependente na variável lingüística pode ser usado para descrever a dependência. Pode-se ter A "aproximadamente" determina B, A "mais ou menos" determina B, etc. A variável lingüística determina a força da dependência que pode ser 0 ou 1. Quando a variável lingüística não é usada, o valor do ∂ é igual a 1 e a dependência fuzzy é a mesma da dependência crisp.

Proximidade

Dependências funcionais fuzzy são estendidas para relações baseadas em proximidade/similaridade. Qualquer α-cut de uma relação de similaridade define uma relação equivalente e conseqüentemente classes equivalentes constróem valores que são similares contendo um grau maior ou igual ao α-cut.

b) Medidas de informações-teóricas

Bancos de dados fuzzy são usados em aplicações que envolvem algumas imprecisões ou incertezas nos dados e na utilização de tomadas de decisão a partir desses dados. Para ajudar a entender o impacto de tais imprecisões, caracterizações em medidas em informações-teóricas estão sendo desenvolvidas, ou seja, são realizadas medidas de incertezas numa relação. Adicionalmente, uma variação de entropia tem sido usada para determinar como uma consulta fuzzy diferencia-se no meio de toda resposta em potencial. Entropia fuzzy pode ser medida como uma função de um valor de domínio ou como uma função de uma relação. Intuitivamente, a incerteza de um valor de domínio aumenta quando a sua cardinalidade aumenta ou quando a similaridade diminui. Assim se um valor de domínio num esquema relacional, consistindo de um simples elemento que representa a informação exata e múltiplos elementos são resultados de fuzziness, então essa incerteza pode ser representada pela entropia.

c) Segurança em banco de dados

Vários estudos vêm sendo realizados para modelos baseados em similaridade e para modelos baseado em proximidade com a tentativa de como representar proteção para a segurança dos dados num banco de dados.

Segurança é normalmente definida como a proteção de banco de dados contra o uso não autorizado, porém possui diversos aspectos. Salvaguardar a modificação ilegal ou destruição de dados é um aspecto de segurança, sendo que para essa discussão deve-se ater à proteção de visão não autorizada de dados .

Um banco de dados fuzzy permite acesso para algumas informações, caso a exata correlação dos itens de dados que permanecem desconhecidas são consideradas. Esse uso de segurança é similar ao uso de segurança em banco de dados estatístico. Por exemplo, informação estatística da média do salário de um grupo pode estar disponível, porém não pode estar disponível o salário de um determinado indivíduo.

Em banco de dados fuzzy cada tupla possui a potencialidade para representar um grande número de interpretações. Acoplamento de dados dentro de um conjunto, dependendo do nível dos valores, pode fornecer uma medida de segurança herdada num banco de dados fuzzy.

IV.3.2. Consultas fuzzy baseadas no conceito de possibilidade

A teoria de possibilidade e a variável lingüística vem sendo utilizada para gerenciar informação imperfeita numa única forma de contexto.

Em SGBD, a informação relacionada com uma aplicação é gerenciada em dois níveis distintos: o primeiro em relação a representação da informação e o segundo em relação a formulação da informação do usuário expressa através de consultas para selecionar dados de interesse que estão armazenado.

Em ambos os níveis a informação pode ser afetada pela imperfeição; quando tem-se a informação estruturada e armazenada num banco de dados, os dados precisos podem não estar disponíveis para seleção. O caso extremo de imperfeição em banco

de dados é dado por um valor nulo que tem sido extensivamente estudado em modelo de banco de dados relacional. Podem existir diferentes causas de valores nulos, como por exemplo: quando o valor do atributo de um objeto é desconhecido, ou quando nenhum valor no domínio é aplicado devido a inconsistência.

Várias abordagens e propostas vem sendo estudada e adotadas para representar dados incompletos, os dados são representados através de conjunto fuzzy interpretados como distribuição de possibilidade. A imperfeição relacionada com dados observados numa aplicação real é principalmente devido a falta de habilidade para selecionar um simples valor de um domínio de possíveis valores de dados ou para incerteza numa associação de um valor com um atributo. Um valor impreciso ou vago pode ser especificado como um conjunto fuzzy definido num domínio de valores de dados, que é interpretado como uma distribuição de possibilidade, nesse caso o grau de pertinência de cada valor num domínio expressa o grau de possibilidade que esse valor possui. Dentro da teoria de possibilidade um conjunto fuzzy tem uma interpretação disjunta.

Na figura IV.1, um valor impreciso do atributo idade de José é selecionado: *Idade(José)=entre 20 e 25.* Esse valor impreciso é representado através de uma distribuição de possibilidade $\mu_{idade(José)}(d)=1 \quad \forall d \in [20,25]$ e $\mu_{idade(José)}(d')=0$, $\forall d' \in [20,25]$.

Na figura IV.2, a função $\mu_{jovem}(d)$ é interpretada como a distribuição de possibilidade associada com o evento *Idade(José) = jovem*, corresponde com a restrição imposta pelo conjunto fuzzy jovem na idade numérica de José.

Dentro da teoria de possibilidade, a incerteza numa associação de um valor com um atributo pode ser também modelado. Na figura IV.3, a distribuição de possibilidade de valores de atributos vagos e incertos são representados. Nenhum valor do domínio é impossível; todo valor do domínio tem um grau de possibilidade diferente de 0 (zero). O valor $\in$ representa o grau de incerteza para a proposição *"José é jovem"*.

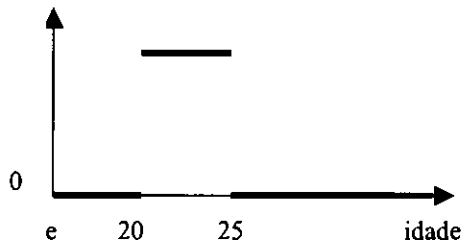


Figura IV.1 - Representação de dados

Idade (José) = entre 20 e 25 anos (valor impreciso)

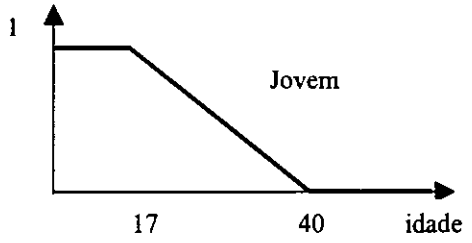


Figura IV.2 - Representação de dados

Idade (José) = jovem (valor vago)

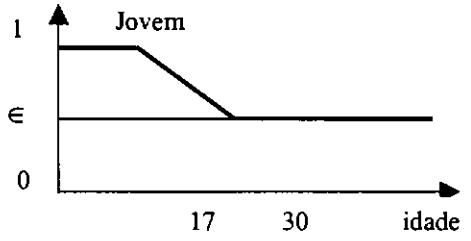


Figura IV.3 - Representação de dados incertos

Idade (José) = jovem

Como mencionado anteriormente, no modelo de dados fuzzy a presença de informação imperfeita pode ser apresentada em dois níveis: na representação de dados e na linguagem de consulta, ou até mesmo em ambas. Dependendo dessa representação, existem três situações que podem ser avaliadas pelo mecanismo de consulta:

a) dados precisos e consultas vagas (com predicados vagos);

b) dados incompletos e consultas precisas (com predicados booleanos);

c) dados incompletos e consultas vagas (com predicados vagos)

IV.3.2.1.Teoria de possibilidade

Na teoria baseada em possibilidade, a informação disponível sobre o valor de um simples valor de atributo A para uma tupla t é representado pela possibilidade distribuição $\pi_{A(t)}$ em $D \cup \{e\}$, onde D é o domínio do atributo A e e é um elemento extra que representa o caso quando o atributo não se aplica em t. A distribuição de possibilidade $\pi_{A(t)}$ pode ser vista como uma restrição fuzzy de possíveis valores de A(t) e define um mapeamento de $D \cup \{e\}$ para [0,1].

Por exemplo, a informação João tem vasta experiência, é representada por ($\forall$ $d \in D$):

$$\pi_{\text{EXPERIÊNCIA(João)}}(e) = 0, \text{ e } \pi_{\text{EXPERIÊNCIA(João)}}(d) = \mu_{\text{VASTA}}(d)$$

onde μ_{VASTA} é uma função de pertinência que representa o predicado vago VASTA no contexto. O valor restrito à distribuição de possibilidade é considerado como exclusivo.

Sistemas de banco de dados relacionais assumem que qualquer relação para ser armazenada é preciso estar pelo menos na primeira forma normal. Isso significa que qualquer valor do atributo é atômico. Isso é feito para facilitar o armazenamento de dados dentro de colunas, que é o padrão. Num ambiente de possibilidade para representar incerteza, um valor desconhecido é representado normalmente usando um conjunto.

É importante distinguir entre a representação que pode requerer vários valores e a noção de atomicidade que é conectada à semântica de um dado valor. Quando o

valor atual de um simples atributo é desconhecido, é necessário representar o conjunto de todos os prováveis, e esses valores também são atômicos.

Se dois valores a e b são descritos por suas respectivas distribuições de possibilidade πa e πb, então podem ser comparados de acordo com a expressão principal. Isso permite dois graus: expressando a extensão para que os valores possibilidade e necessidade satisfaçam a comparação da relação. Esses graus são dados por:

$$poss(a=b) = sup_{x,y}\ (min(\pi a(x), \pi b(y), \mu=(x,y)))$$

$$nec(a=b) = 1 - sup_{x,y}\ (min(\pi a(x), \pi b(y), \mu\neq(x,y)))$$
$$= inf_{x,y}(max(1-\pi a(x), 1-\pi b(y), \mu=(x,y)))$$

Quando a e b são conhecidos, esses dois graus sofrem um colapso (e levam seus valores em{0,1}), então não existe incerteza. Por outro lado, o fato de que dois valores de atributo (na mesma tupla ou em duas tupla distintas) são representadas pela mesma distribuição de possibilidade, não implica que esses valores precisam ser iguais.

IV.3.2.2. Linguagem de consulta flexível

Uma consulta envolvendo critérios vagos é representada pelo conjunto de lógica fuzzy e é definida usando vários tipos de construtores [96]. Predicados atômicos são definidos através do conjunto fuzzy sobre um ou vários domínios D_i, isto é, por uma função de pertinência de D_1 X... X Dn para [0,1], como por exemplo, tem-se valores de "alto", "novo", etc para predicados vagos. Na prática, funções com forma trapezoidal são normalmente utilizadas, claro que isso depende da escolha da representação do uso do predicado.

Qualificadores são representados pelas funções de [0,1] para [0,1] e são aplicados nas funções de pertinência para modelar o efeito da variável lingüistica, como por exemplo, "muito", "mais ou menos", "melhor", etc. As funções dos qualificadores mais comuns são:

1) $\mu_{mod}\, p(x) = (\mu p(x))^n$ (n>1 para um concentrador e n<1 para um dilatador)

2) $\mu_{mod}\, p(x) = o^n(\mu p(x))$

3) $\mu_{mod}\, p(x) = \mu p(x \pm a)$

Algumas consultas requerem as funções de máximo, mínimo, soma ou média dos valores, ou de uma forma geral, requerem a avaliação de qualquer função escalar sobre um conjunto de objetos especificados em fuzzy. Como por exemplo, *"Qual é a média do salário dos empregados jovens da empresa?"* Inúmeras consultas trabalham dessa maneira e são baseadas na definição de uma representação convencional para o conjunto fuzzy, em meio de α-cuts. Assim, um possível valor de f(A) onde f é uma função e A é um conjunto fuzzy de valores N, é definido por:

$$\mu_N(r) = \sup \{ \alpha \in [0,1] \mid f(A_\alpha) = r \}$$

onde: A_α é o α-cut de A

Ambiente baseado em possibilidade

As consultas de bancos de dados relacionais, onde alguns valores de atributos estão incompletos são representados pela distribuição de possibilidade. De acordo com a visão de possibilidade, quando uma condição é aplicada a um dado conhecido de maneira imperfeita, o resultado da avaliação da consulta pode não ser um simples valor. Assim os valores precisos de alguns atributos do mesmo item não

são conhecidos, o fato é que esses itens que satisfazem ou não a consulta, podem ser incertos.

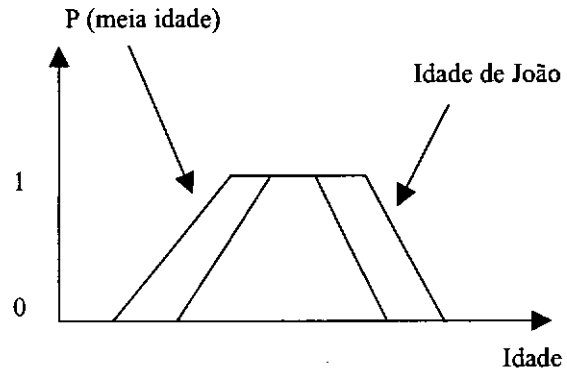


Figura IV.4 – Idade de João e o termo meia idade

Exemplo: Idade de João e o predicado fuzzy são representados de acordo com a figura IV.4, a avaliação da condição Idade de João = "meia idade" é baseado na computação dos valores:

$$\min(\pi_{\text{Idade de João}}(u), \mu_{\text{meia idade}}(u))$$

e

$$\max(1 - \pi_{\text{Idade de João}}(u), \mu_{\text{meia idade}}(u))$$

A figura IV.5. apresenta o caso, onde o grau de possibilidade é 1 e onde o grau de certeza é 0.17.

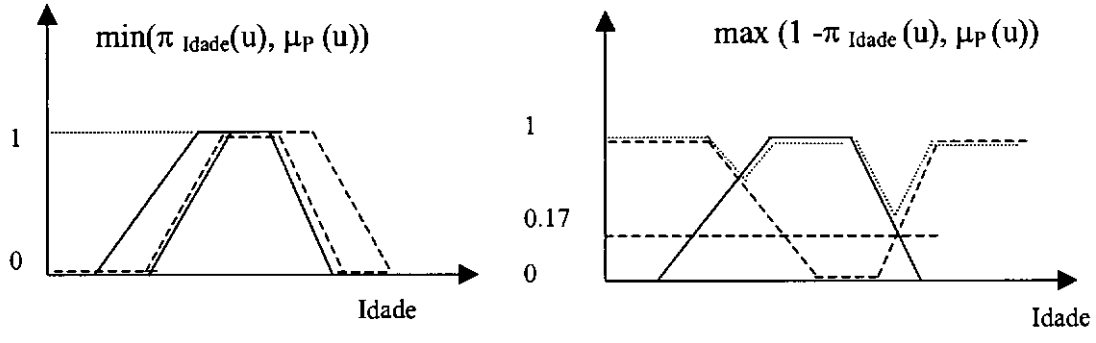


Figura IV.5 – Avaliação da Idade de João = meia idade

IV.3.2.3. SQLf

Na área de banco de dados relacional, a linguagem de SQL é considerada como linguagem padrão, sendo referência para qualquer linguagem de extensão de consulta. A linguagem conhecida como SQLf tem como objetivo principal suportar uma grande variedade de consultas fuzzy, ou seja, predicados fuzzy são introduzidos dentro dessa linguagem, e permanece com o espírito da linguagem original SQL, definida como uma linguagem forte. O princípio básico é a introdução de imprecisão no bloco da estrutura da linguagem SQL:

```
SELECT   (lista de atributos)
    FROM     (lista de relações)
    WHERE    (condição)
```

Inicialmente, uma nova característica é adicionada para fornecer ao usuário uma saída de mecanismo regular. Isso pode ser feito especificando o número de resposta desejada (N) e/ou um limite ($T \in [0,1]$) agindo como um α-cut na relação, resultando em uma seleção fuzzy. A nova formulação de estrutura seria:

```
SELECT   (N | T | N, T)  (atributos)
    FROM     (lista de relações)
    WHERE    (condição fuzzy)
```

e são interpretadas da seguinte forma:

1) aplicar o predicado (condição fuzzy) no produto cartesiano da relação envolvida
2) projetar a relação fuzzy resultante sobre a lista de atributos e retornar o melhor N e/ou as tuplas sobre o limite T

Basicamente, uma condição fuzzy aplicada a tupla individual é composta de predicados (booleano e fuzzy) e conectores. É também possível conectar duas relações pelo significado de um predicado de união fuzzy.

Exemplo: Considerar a relação de FUNCIONÁRIOS (Id_Func, Nome, Salário, Idade, Cidade, Id_depto) e DEPARTAMENTO (Id_depto, Nome, Orçamento, Localização). Consulta: "Encontrar os 5 melhores funcionários de SP que são bem remunerados e que trabalham num departamento que possui um alto orçamento". A linguagem de SQLf é representada da seguinte forma:

```
SELECT 5 F.Id_Func, D.Id_depto
        FROM FUNCIONÁRIOS F, DEPARTAMENTO D
        WHERE F.Cidade = "SP"AND F.Salário = "bem remunerado" AND
                D.Orçamento = "alto"  AND  F.Id_depto = D.Id_depto;
```

IV.3.2.4. Restrição de Integridade

A maior função de um SGBD é gerenciar as propriedades do dados, como por exemplo, tem-se um banco de dados que contém informações sobre clientes, compras e estoque, e deseja-se saber: "A quantidade de compras de um dado item é menor que a quantidade desse item no estoque ?", "Para comprar, um cliente precisa ser reconhecido como cliente ?", etc.

Quando um esquema de banco de dados é construído, uma característica importante está ligada a propriedade dos dados, normalmente chamada de restrição de integridade (IC). Segundo Bosc et al [13] os IC representam propriedades do mundo real e precisam ser testados quando atualizações são realizadas. Assim, se um IC indica que os salários de alguns funcionários são acima de mil reais, então a atualização ou a inserção de novos funcionários precisa ser testada. IC pode ser dividido dentro das seguintes categorias:

- restrições sobre uma tupla de uma relação (salário de funcionário acima de R$ 1200,00)
- restrição sobre uma relação (salário dos gerentes são maiores que a média do salário dos funcionários)
- restrição envolvendo várias relações (qualquer identificação do departamento usado na relação de funcionários precisa estar na relação do departamento)

Os IC são predicados booleanos mais ou menos complexos e indispensáveis em sistemas de banco de dados. Quando são violados, os SGBDs conseqüentemente rejeitam a atualização correspondente. Para resolver o problema de IC é necessário que o banco de dados possua dois tipos de dados: dados regulares e dados excepcionais, podendo violar um ou vários ICs. Essa visão pode ser representada como uma transformação de uma propriedade inicial crisp P. Tem-se então um tipo de restrição fuzzy e pode-se pensar sobre a generalização de possíveis restrições. Esse tópico ainda não está muito bem pesquisado, mas é uma tentativa de se ter uma restrição fuzzy.

Dependência funcional é apontada como uma forma de evitar redundância, facilitando a manutenção dos dados quando forem realizadas suas atualizações. Banco de dados relacionais trabalham com essa funcionalidade. No contexto de banco de dados fuzzy, existe imprecisão nos dados, pode-se então reconsiderar essa noção para expressar algumas restrições entre a similaridade dos valores de A e B na tupla. Esse assunto requer desenvolvimentos interessantes e algumas pesquisas estão sendo realizadas em dependência funcional.

IV.3.2.5. Aspecto de implementação e processamento de consultas

Na avaliação de consultas no contexto de linguagem declarativa (como relacional) persiste o problema devido à sua natureza combinatorial. Dado uma consulta, não existe nenhum plano de otimização (algoritmo) conhecido para o seu processamento, num tempo razoável. Isso se aplica também para consultas imprecisas,

especialmente para as mais complexas. Esse aumento em complexidade é em conseqüência às operações a serem realizadas dependendo do volume de dados, visto que seleção fuzzy, união, etc. produzem mais tuplas que no contexto booleano. Dois processos de consultas fuzzy podem ser previstos:

1) usar um SGBD convencional e desenvolver um plano adicional de regras para uma interface, ou
2) construir um novo sistema incluindo técnicas de processamento de consultas fuzzy no seu núcleo.

Desenvolvimentos limitados tem sido conduzidos ao longo dessa linha. Dois aspectos relacionados ao processamento de consulta devem ser ressaltados:

1) a extensão de mecanismo de indexação para predicados fuzzy (par um banco de dados convencional e fuzzy) e
2) a avaliação de consultas específicas (ou operadores) em banco de dados convencional, dando um nível de sofisticação para ser atendido.

As consultas fuzzy reduzem o risco de se obter repostas vazias com o uso de uma escala de discriminação [0,1]. Num conjunto baseado em interpretações fuzzy sempre tem-se a chance de um elemento ser considerado mais ou menos satisfatório, sendo que raras vezes pode acontecer que nenhum dos elementos satisfaçam a consulta do usuário.

Uma das abordagens sobre isso está relacionada com estudo sobre clustering em dados. Portanto, quando não existe dados que satisfaça uma consulta, o sistema apresenta quais são os tipos de dados que estão armazenados no banco de dados e fornece uma consulta fechada como uma alternativa. É aplicado um clustering para classificar os dados dentro de vários clusters fuzzy. Os clusters fuzzy são expressos pela classe lingüística definida pela função de pertinência. Quando a consulta do usuário falha, o sistema fornece ao usuário informações alternativas que consistem de três partes:

1) o dado do cluster é o próximo para que o usuário use,
2) dados alternativos no próximo cluster; e

3) condição para que o usuário possa modificar, caso não esteja satisfeito com a proposta do cluster ou do dado.

IV. 4. Considerações finais

Muitos raciocínios humanos se expressam através de variáveis cujos valores são incertos. As origens das incertezas são infindáveis e o mesmo conceito pode possuir diversos graus de imprecisão, em diferentes contextos ou tempos. No capítulo III foi apresentada a teoria de lógica fuzzy, onde pode-se definir como uma das principais formas para tratar a incerteza.

O uso dessa teoria representa um avanço no paradigma de análise e exploração de informações, pois permite que as incertezas sejam expressas através das variáveis lingüisticas, tornando as consultas mais flexíveis, mais confiáveis, menos sujeitas a erros e ambigüidades não detectáveis. O mais interessante é que essa afirmativa será vista mais adiante no estudo de caso proposto.

Neste capítulo, comentou-se as principais abordagens da utilização da teoria na formulação de consultas mais flexíveis, com o objetivo de obter informações dos banco de dados. Foi também apresentado dois principais tipos de consultas flexíveis: uma aborda consulta fuzzy baseada no conceito de similaridade/proximidade e a outra é baseada no conceito de possibilidade.

CAPÍTULO V

METODOLOGIA E ESTUDO DE CASO

V.1. Introdução

Como já foi dito, os métodos tradicionais de análise de dados, como por exemplo: planilhas eletrônicas e consultas ad-hoc, perderam a habilidade para interpretar e digerir grandes volumes de dados. Os métodos convencionais podem fornecer informações, mas não podem analisar os conteúdos dessas informações para focalizar um conhecimento importante. Existe, portanto, uma necessidade significativa para utilização de novas técnicas/ferramentas automatizadas e inteligentes que estão emergindo, com a capacidade de ajudar na analise dessa montanha de dados, transformando-as em conhecimento útil.

No entanto, essas grandes quantidades de informações são também consideradas imprecisas, vagas, nebulosas, inconsistentes, etc. Como se vê, essa análise não é simplesmente apenas descobrir conhecimentos. Existe ainda uma forte necessidade de se trabalhar com a proposta sobre o tratamento de incerteza, descrita no capítulo III.

De acordo com as abordagens dos capítulos anteriores, observou-se que uma das formas para lidar com essa complexidade instalada sobre os dados, é a utilização das novas ferramentas/técnicas que surgiram rapidamente: Data Mart, Data Mining e Fuzzy Query.

Neste capítulo é apresentado uma metodologia CFIC (Consulta Flexível para Inteligência Competitiva) que visa absorver esses novos paradigmas, dentro de um cenário que compreende a inteligência competitiva. O principal papel deste estudo é o planejamento de um ambiente unificado, gerando um modelo para compartilhar e analisar o conjunto de dados disponíveis, contendo informações sobre bens patrimoniais de uma empresa de engenharia de transporte. Essa análise tem como objetivo compor um processo analítico que irá usar essas informações para

desenvolver conhecimentos estratégicos relevantes e seguros sobre competidores, posições, performance, capacidades e intenções.

Assim, este estudo proposto a ser apresentado constituiu-se fundamentalmente na análise de uma unidade de informação, a fim de descrever seu perfil, organizar seu funcionamento, identificar suas forças e fraquezas para que possam ser exploradas em tempo hábil, e projetar uma metodologia que auxilie no gerenciamento dos dados e no desempenho de suas próprias funções.

V.2. Definição do problema

Uma empresa de grande porte da área de transporte de carga pesada com funcionamento em determinadas localidades do país – mais precisamente nos estados do Rio de Janeiro, Minas Gerais e Espirito Santos (parte da região Sudeste), encontra-se em vias de tomar complexas decisões e gerar grandiosas estratégias com relação ao destino da empresa.

O propósito é separar a empresa por região em três companhias de transporte pesado, onde a sua venda deverá ser efetuada através de vendas de ações colocadas no mercado financeiro. Certamente que isso gera um temor quanto à possíveis dificuldades previstas nos retornos dos investimentos. Muitas tomadas de decisões, entretanto, são desinclinadas para investir porque os riscos são percebidos como sendo maior do que as compensações.

O objetivo maior deste estudo é planejado apenas em uma fatia desta complexa tomada de decisão, ou seja, está relacionado somente com o potencial de análise das informações dos bens patrimoniais da empresa. A proposta é produzida dentro de um enfoque para traçar um perfil sobre as operações de negócios desses bens, identificando fatores envolvidos nas considerações que poderão definir a escolha de investir nessa área, como por exemplo, identificar os benefícios de realização da venda de determinados bens antes da divisão da companhia; verificar dentre as três companhias a serem criadas, quais delas se beneficiariam mais com a possibilidade de venda ; etc.

Com relação aos dados disponíveis armazenados, a empresa foi a primeira a admitir que os mesmos não eram muito claros. Em outras palavras, identificou-se de imediato a necessidade de efetuar a análise dentro de um processo contendo três passos. O primeiro passo compreendendo a informação, isto é, a visualização dos dados; o segundo passo integra a informação dentro de estratégias e o terceiro passo está vinculado à mobilização de recursos para análise, no sentido de oferecer mais um suporte que irá compor as ações de tomada de decisão dentro da nova realidade da companhia.

O princípio básico do processo de tratamento de informação como elemento de conteúdo estratégico é a transformação de um grande volume de informações, aparentemente desconexo ou sem relações fortes, de difícil compreensão, em informações úteis e com significado. Na verdade, a finalidade é de se extrair de uma massa de informações um valor adicional que não representará nada de totalmente inédito, mas que permitirá através da reunião do conjunto de informações analisadas, a redução da incerteza e a maior qualificação da tomada de decisão. Enfim, desenvolve um estudo, em profundidade, para extrair o essencial das informações e criar instrumentos de auxílio à decisão ou à ação estratégica.

O presente estudo de interesse comercial e de limitação à consulta à uma única base dados, como será descrito, passou pelas diferentes fases já relacionadas nos capítulos anteriores. Para que os dados fossem organizados e sintetizados, foi utilizado como técnica de tratamento da informação, o Data Mining, permitindo a elaboração de análises sobre um conjunto de informações, facilitando a observação da concentração e evolução de interesses, assim como a visualização de cenários. A legitimidade desse tipo de trabalho atinge não somente a área científica, mas também a tecnológica, e a comercial.

Esclarecida a definição do problema deste estudo, será então descrita a sua operacionalização, expondo-se os procedimentos utilizados para o tratamento das informações, o projeto e construção de um ambiente central e integrado dos dados – Data Mart e, posteriormente, a análise e interpretação dos resultados obtidos.

V.3. Levantamento e análise sobre os dados disponíveis

Dada a falta de documentação formal e compreensiva dos dados armazenados no sistema de banco de dados da empresa, necessidades e expectativas de um contexto mais apropriado para entender o verdadeiro conteúdo e significado dos dados e ainda compreender o que esses dados representam no mundo real foram levantadas, analisadas e organizadas dentro do seguinte escopo:

- Inicialmente foi avaliada toda documentação disponível do banco de dados, logo em seguida foi realizada entrevista utilizando o método e técnica de entrevista não estruturada que consiste em possibilitar que o entrevistado fale livremente sobre o escopo do banco de dados, revelando os objetos sobre os quais conhece, como estão relacionados ou organizados, quais são os seus domínios, quais os mais importantes e como são trabalhados;

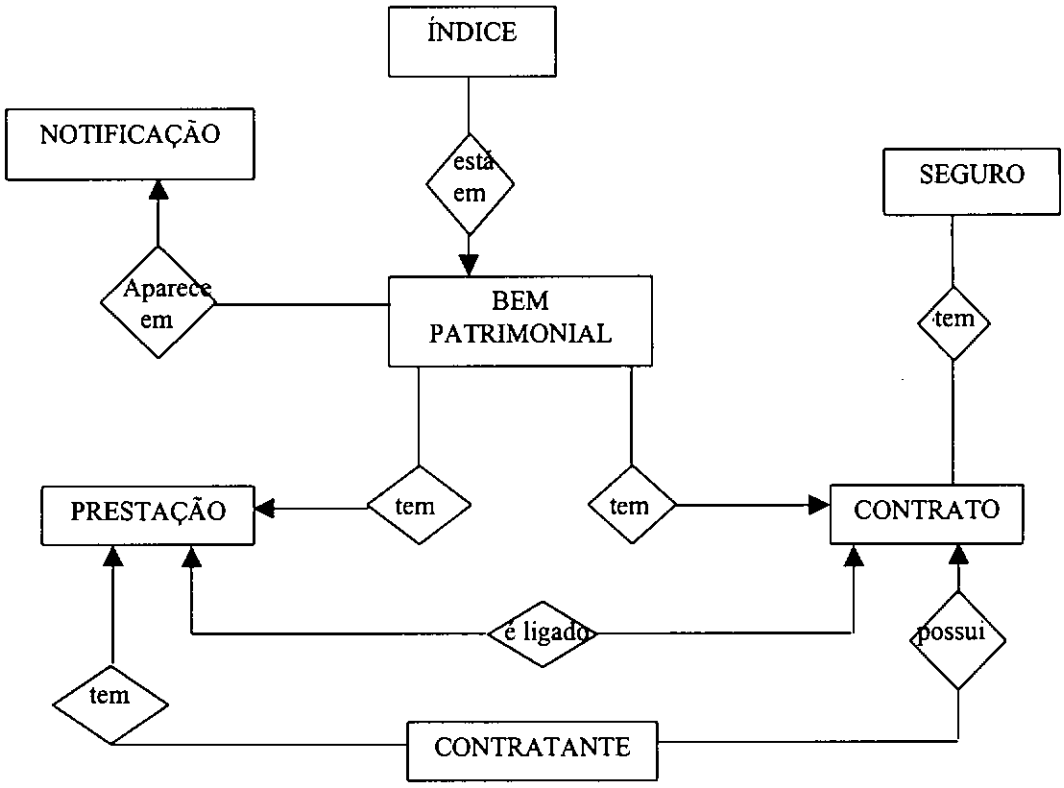
- Posteriormente foram feitas mais entrevistas e desta vez foi usado o método e técnica de entrevista estruturada, onde o entrevistado responde à colocação das questões focalizadas em casos particulares de detalhes, com o objetivo de se aprofundar no domínio do dado considerado relevante ou ainda sobre aqueles que possui pouca compreensão;

- Finalmente nesta fase foi desenvolvida uma limpeza nos dados, ou seja, operações básicas tais como: remoção de ruídos ou perda de consistência de domínio, remoção de registros duplicados, campos com valores inexistentes, etc. O processo de limpeza de dados foi realizado através de algoritmos usados na tarefa de Data Mining [121].

Antes do processo de levantamento, a documentação do sistema de banco de dados de bem patrimonial encontrava-se parcialmente completa e possuía algumas ambigüidades, tornando difícil capturar os conceitos existentes no domínio do

problema e a visualização dos relacionamentos existentes. Portanto, após este levantamento foi possível especificar e elaborar um Modelo de Entidade-Relacionamento descrevendo claramente o que realmente faz este banco de dados, ou seja, realiza o cadastro dos bens patrimoniais da empresa e o controle da exploração patrimonial desses bens englobando informações sobre as contratantes , os contratos, as prestações e as notificações, conforme é apresentado no diagrama de Entidade-Relacionamento.

Diagrama Entidade-Relacionamento



Notação Gráfica:

Entidade

Relacionamento

Cardinalidade 1

Cardinalidade N

Definição das chaves primárias e estrangeiras dos dados relacionados com os bens patrimoniais da empresa, os contratantes, os contratos, as prestações e as notificações, como mostra a figura V.1.

TABELAS	CHAVE PRIMARIA	CHAVE ESTRANGEIRA
Bem Patrimonial	COD-TR-BP + COD-BP + NUM-PCL-BP	(COD-IDF + DATA-IDF em relação a Índice)
Contrato	COD-TR-BP + NUM-CRL-CEBP + NUM-ADT+CEBP	(COD-TR-BP + COD-BP + NUM-PCL-BP em relação a Bem Patrimonial) (COD-TR-BP + NUM-CRL-CEBP + NUM-PEBP + NUM-PEBP-BCO em relação a Prestação) (NUM-CNBP em relação a Contratante)
Contratante	NUM-CNBP	
Prestação	COD-TR-BP + NUM-CRL-CEBP + NUM-PEBP + NUM-PEBP-BCO	(COD-TR-BP + COD-BP + NUM-PCL-BP em relação a Bem Patrimonial) (COD-TR-BP + NUM-CRL-CEBP + NUM-ADT em relação a Contrato) (NUM-CNBP em relação a Contratante)
Notificação	COD-TR-BP + COD-BP + COD-NTF-BP	(COD-TR-BP + COD-BP + NUM-PCL-BP em relação a Bem Patrimonial)
Índice	COD-IDF + DATA-IDF	
Seguro	COD-SGD	

Figura V.1 – Definição das chaves

V.3.1. Visualização dos dados através da tarefa de Data Mining (DM)

O termo DM é normalmente usado como uma aplicação de técnicas/algoritmos, por estatísticos, analistas de dados e administradores de

informações de sistemas, sob algumas aceitáveis limitações computacionais, que produzem uma quantidade de padrões sobre o banco de dados.

Aqui, o objetivo principal é estabelecer um conhecimento compreensivo, fazendo um estudo que possibilite a visualização dos dados, onde tenta-se obter uma representação, aumentando sua compreensão através de algoritmos de DM - árvore de decisão.

V.3.1.1. Classificação dos dados

Os dados de Bem Patrimonial utilizado neste estudo é um banco de dados Microsoft Access (Office 97) denominado BP.mdb. A sua análise é realizada para derivar conhecimentos (padrões e regras) dos dados. O software utilizado consiste dos seguintes de módulos:

1) Ambiente de desenvolvimento compreendendo uma definição de seqüência gráfica para trabalhar nas suas operações;

2) Mecanismo de engenharia – trabalha com tabelas e manipulação de campos e resultados;

3) Executa operações interativa - trabalha com um conjunto grande de ferramentas e facilidades para exploração dos dados;

4) Executa operações de associações sobre os dados;

5) Executa operações de associações sobre transações realizadas a partir dos dados.

Este banco de dados BP.mdb contém aproximadamente 60 mil linhas distribuídas dentro de 7 tabelas, como mostra a figura V.2.

TABELAS	QUANTIDADE DE REGISTROS	QUANTIDADE DE ATRIBUTOS
(BP) Bem Patrimonial	1.568	31
(CNBP) Contratante do bem patrimonial	15.686	31

(CEBP) Contrato de exploração patrimonial	4.518	73
(PEBP) Prestação de contrato de exploração	34.950	45
(NTF) Notificação	1.656	8
(SGO) Seguro	2	12
(IDF) Índice	681	5

Figura V.2 – Tabelas contidas no BP.mdb

O processo de Data Mining pode ser dividido da seguinte maneira: seleção de data sources, transformação, limpeza e preparação dos dados, informação e visualização dos dados, exploração dos padrões, e o desenvolvimento e descobrimento de padrões. A ferramenta aqui utilizada suporta todo processo, através de uma interface gráfica, onde o desenvolvedor do projeto define a seqüência de operações.

O papel predominante nesta etapa do estudo, já que os dados possuem uma certa complexidade, concentra-se nas operações de limpeza, transformação e visualização dos dados, onde o maior objetivo é tentar definir e compreender melhor as informações que deverão ser utilizadas no projeto do Modelo de Data Mart. É nesse ambiente que serão realizadas as análises para explorar conhecimento útil.

Deve-se trabalhar dentro das seguintes características: 1) definição de um Data Source que deve ser usado no projeto de Data Mining. É utilizado um ODBC (Open Database Connectivity) para a operação de entrada; 2) transformação dos dados, na maioria das vezes a entrada selecionada do data source precisa de uma seqüência de operações de transformações (manipulação de tabelas e campos) antes de serem lidos; 3) verificar o resultado de transformação, isto é usado para conferir que sua operação de transformação de dados está trabalhando dentro do esperado; 4) explorando os dados transformados, operações para explorar padrões precisam ser ligadas com um data source físico (com dados de entrada de um data source ou com uma operação de saída de um data source).

É muito raro ter um dado disponível para um Data Mining. Normalmente os dados são transformados (pré-processados) antes de serem explorados. Deve-se realizar extensas transformações de dados, por meio de manipulação das tabelas, envolvendo associação das tabelas, união das tabelas, ordenação das tabelas (de forma ascendentes ou descendentes), e ainda por meio de manipulação dos campos, envolvendo a geração de novos campos dos dados disponíveis.

O processo de operação para selecionar o data source, transformar, limpar e preparar os dados, visualizar os dados, explorar, desenvolver e descobrir os padrões deste banco de dados foi realizado dentro das seguintes fases:

a) Conexão com o banco de dados origem, conforme o exemplo da figura V.3. Sendo que para acessar o sistema de banco de dados é preciso ter um drive ODBC compatível com o banco de dados.

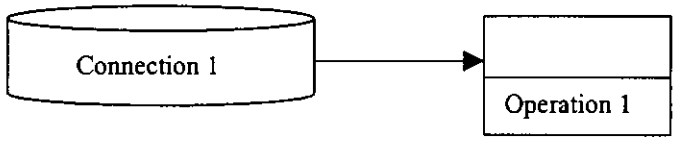


Figura V.3 – Conexão com o banco de dados fonte

b) Seleção do Edit - Insert - Data Input Connection para montar a entrada do dado de origem.

c) Seleção do diálogo Edit Connection para escolher o ODBC a ser utilizado.

d) Seleção do data source desejado – o arquivo BP.mdb, no caso o MS Access 97 Database, e logo em seguida selecionar a tabela a ser trabalhada., retornando para o diálogo do Edit Connection. Assim que pressionar o botão de OK, uma nova operação de conexão é criada.

e) Seleção do Edit – Insert – para adicionar uma nova operação

f) Examinar o resumo - na conexão do banco com os dados de origem é construído um resumo contendo informações estatísticas sobre os campos de entrada que são apresentados em uma tabela. Este processo pode levar minutos para um pequeno conjunto de dados ou até mesmo algumas horas para um grande conjunto de dados.

O programa detecta automaticamente o tipo de cada campo (numérico ou texto), ou seja, copia o data source para coletar informação dos campos de dentro da tabela. O primeiro passo para examinar a informação coletada é entender os dados. A Table View apresenta uma tabela com os campos; cada campo possui um nome, tipo, valores, grupos de valores, uso e classe. O nome do campo usado na tabela é o nome padrão, que pode ser renomeado. O tipo de cada campo é numérico ou discreto (isto é, texto). Para campos numéricos existe o valor mínimo, máximo e médio e o valor de desvio padrão. Para campos textos pode-se ver o número de valores discretos e a freqüência de cada valor.

Nesta etapa foram detectados campos totalmente em branco, ou seja, não possuíam nenhum valor e também campos que continham ruídos ou valores que nada conseguiam expressar. Certamente, todas as análises foram realizadas baseadas no resultado gerado a partir das entrevistas realizadas com o especialista. Após essa análise, chegou-se à seguinte conclusão com relação a limpeza dos dados, como apresenta a figura V.4.

TABELAS	QUANTIDADE DE REGISTROS	QUANTIDADE DE ATRIBUTOS
(BP) Bem Patrimonial	1.568	21
(CNBP) Contratante do bem patrimonial	15.686	30
(CEBP) Contrato de exploração patrimonial	4.518	65
(PEBP) Prestação de contrato de exploração	34.950	40
(NTF) Notificação	1.656	8
(SGO) Seguro	2	12
(IDF) Índice	681	5

Figura V.4 – Tabelas contendo as transformações dos dados

g) Seleção do campo de saída (resultado) – depois de estudar a informação básica em relação aos dados, é hora de decidir o campo resultado para a análise e também os campos atributos da análise para gerar uma árvore de classificação. Pode-se decidir também quais atributos serão excluídos da análise.

Normalmente todos os campos são considerados atributos. A proposta para análise é traçar o perfil do dado de acordo com um campo meta e o escolher para se tornar o campo resultado. Para realizar a exploração do conjunto de dados e gerar uma árvore é preciso selecionar um dos campos como resultado e também denominar os dois últimos campos de atributo, caso contrário, este algoritmo de DM não possui fundamento para começar.

Os dados disponíveis nesta análise incluem muitos atributos, sendo preciso excluir alguns atributos que não possuem relevância para predizer o modelo, aumentando com isso a velocidade de processamento para obter o resultado da árvore de indução. Entretanto, como nem sempre a exclusão dos atributos é muito óbvia, então as vezes se fez necessário a presença do especialista do domínio para ajudar na exclusão.

h) Indução da árvore de dados – do conjunto de informação contendo o total dos valores do campo selecionado como resultado. Deve-se selecionar a opção para gerar uma árvore e depois aceitar os parâmetros default, conforme mostra a figura V.5.

1		
reject accept	238 196	54.8% 45.2%
Total	434	100.0%

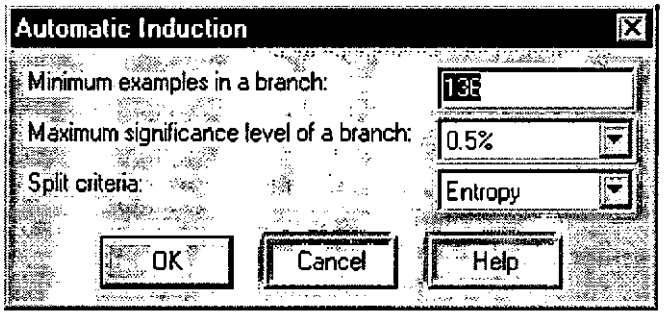


Figura V.5 – Parâmetros para gerar uma árvore

O processo de indução inicia com um status no diálogo indicando que o processo está sendo realizado. Entretanto, não se pode predizer como a árvore resultará – grande, complexa, etc.

i) Resultado da árvore de indução – é apresentado na janela podendo conter diferentes thresholds e valores, como mostra a figura V.6.

Atributo resultado (saída) = COD-SIT-CEBP

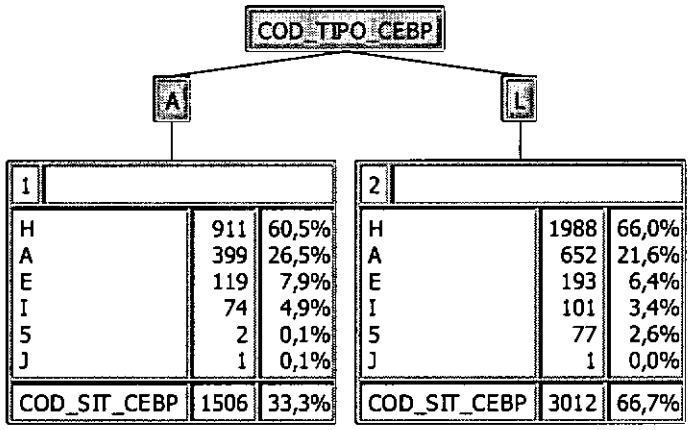


Figura V.6 – Resultado da árvore de indução

Para árvores muito grandes deve-se adicionar uma segunda apresentação em escala menor para ajudar na visualização da árvore. Para estudar o resultado da árvore é preciso seguir o caminho apresentado pela árvore do começo até o final para entender o relacionamento entre os atributos e o resultado.

j) Copiar a árvore atual como um arquivo gráfico de formato WMF (Windows Meta File). Pode-se ainda exportar a árvore para uma variedade de outros formatos, inclusive para o Microsoft Word, obtendo dessa forma mais flexibilidade e um melhor resultado final para visualização da árvore de decisão.

k) Testando a árvore com outros dados – para obter um resultado diferente é preciso definir um dado diferente como campo resultante para se ter um outro perfil em termos de resultado. Após a nova indução, a árvore existente é atualizada mostrando um novo resultado.

Encerramento da tarefa de DM

A partir da análise dos resultados das árvores, pode-se construir o novo modelo de dados – modelo dimensional, contendo as respectivas informações necessárias para integrar esse ambiente. Portanto, nessa fase é desenvolvido o contexto para a definição estratégia-estrutura do novo modelo, conforme apresenta a figura V.7.

TABELAS	QUANTIDADE DE REGISTROS	QUANTIDADE DE ATRIBUTOS
Bem Patrimonial	1.568	21
Contratante do bem patrimonial	15.686	30
Contrato de exploração patrimonial	4.518	60
Prestação de contrato de exploração	34.950	40

Figura V.7 - Composição final das informações a serem usadas no modelo dimensional:

V.4. Modelagem do problema utilizando a metodologia de Data Mart

Como abordado no capítulo I, normalmente os Data Marts são construídos e controlados por um departamento dentro de uma organização requerendo um menor recurso computacional, fornecendo aos usuários suporte para gerar informação a partir dos dados armazenados, com aceitáveis tempo de resposta. Em contraste, um Data Warehouse (DW) trabalha com múltiplas áreas e é implementado e controlado por uma única organização central, assim um DW congrega dados de múltiplos sistemas.

O estudo proposto possui um enfoque que recorre à centralização das informações de uma determinada área, trata-se de um contexto localizado apenas num setor da empresa. Dessa forma, sabe-se que é preciso considerar todos os aspectos sujeitos a regras e relacionamentos estabelecidas nas informações disponíveis e necessidades da análise.

A modelagem de um Data Mart do tipo independente é considerada a melhor opção, sendo que o motivo primordial desta escolha está estritamente relacionado com a extensão do escopo abordado, isto é, a precedência presente neste estudo para explorar as informações baseiam-se apenas em uma parte do conteúdo de toda informação da companhia, mapeando uma determinada área específica e não toda organização. Portanto, neste momento não se vislumbrou a necessidade de se construir um DW.

O Data Mart é dito independente porque a obtenção dos dados é realizada através da extração de um sistema operacional externo e não de um DW. O processo de alimentação dos dados engloba um conjunto de tarefas relacionadas com a extração dos dados de um sistema operacional, limpeza dos dados e armazenamento desses dados no Data Mart.

Criado esse contexto, os outros demais passos são conseqüência. O próximo passo para a construção do Data Mart envolve uma definição estratégica para a alimentação dos dados a partir de uma extração dos mesmos do sistema operacional, o seu armazenamento e administração, e a sua recuperação para dar suporte nas tomadas de decisões da empresa.

A alimentação dos dados engloba todas as tarefas relacionadas com: identificação e seleção de um conjunto de dados relevantes num data source; projeto lógico e físico do Data Mart; obtenção desses dados; limpeza e modificação dos mesmos, caso necessário, para um formato apropriado; criação e alimentação de um metadado; e a realização de um carregamento inicial dos dados para o Data Mart e a criação e armazenamento de um metadado. O armazenamento de dados está relacionado com a construção de um Data Mart que fornece um acesso aos dados de forma eficiente e uma administração contendo diversas tarefas, tais como: acesso seguro, trabalhar com aumento no volume de dados e assegurar acessibilidade aos dados. O uso do Data Mart está diretamente ligado com a consulta aos dados, sua análise, criação de relatórios e gráficos, e sua publicação.

Parte do processo de alimentação dos de dados, a limpeza dos dados, suas transformações e a identificação do subconjunto de dados apropriados baseado numa necessidade específica, foi realizado tendo como base a utilização do software de Data Mining. O Data Mart é uma aplicação específica contendo informações sobre bem

patrimonial que utiliza o software Microsoft Access 97 para o armazenamento dos seus dados.

O carregamento inicial dos dados foi feito em um ambiente operacional específico, totalmente dedicado para trabalhar com o Data Mart, ou seja, é um ambiente separado do sistema de banco de dados, onde serão realizadas todas as operações de consulta. O Data Mart permanecerá on-line, parcialmente ou exclusivamente em modo somente de leitura. O tempo previsto para as etapas de carregamento, indexação, controle de qualidade e liberação dos dados varia de duas a oito horas. Essas etapas são consideradas altamente previsíveis, portanto, estima-se um período em torno de doze em doze meses para realizar a atualização do Data Mart.

O Data Mart é variável em relação ao tempo, ou seja, a variação em relação ao tempo se manifesta de diversos modos, como por exemplo: o horizonte de tempo válido é significativamente maior do que o dos sistemas operacionais, variação de cinco a dez anos; os bancos de dados operacionais possuem dados de "valor corrente", isto é, dados cuja exatidão é válida para o momento de acesso, assim sendo, dados de valor corrente podem ser atualizados. Já os dados existentes no Data Mart não possuem essa característica, são dados capturados num determinado momento.

Portanto, a técnica utilizada para realizar a atualização do Data Mart é a técnica de replicação de dados. A técnica de replicação requer que os dados a serem capturados sejam identificados antes da atualização, ou seja, à medida que a atualização transcorre, os dados são capturados.

Uma grande vantagem da replicação é que o processo para obter os dados pode ser controlado seletivamente. Isto é, apenas os dados efetivamente necessários são capturados. Uma outra vantagem é que o formato dos dados é bem definido, sendo que o conteúdo e a estrutura dos dados capturados devem ser de fácil entendimento e bem documentados pelo desenvolvedor.

Como desvantagem pode-se dizer que existe uma E/S (entrada/saída) adicional, resultante da obtenção dos dados. Normalmente, a quantidade de E/S não é insignificante. Além do mais, essa E/S é consumida durante o meio do dia de alta performance, no momento em que o sistema tem menos condições de atender a tal demanda. Ainda uma outra desvantagem acontece devido a constante atenção que o sistema requer em relação às definições de parâmetros que controlam a obtenção dos

dados na hora da atualização, em função da natureza do Data Mart que pode vir a ser instável e mutável.

V.4.1. Modelando os dados através do Modelo Dimensional

Como exposto anteriormente no capítulo II, no modelo dimensional existe uma tabela dominante no centro do diagrama com múltiplas junções conectando-a às outras tabelas. Cada uma das tabelas secundárias, denominadas de tabelas de dimensão, possuem apenas uma junção com a tabela central, denominada de tabela de fatos.

A tabela de fatos é um meio que apresenta o amplo conjunto de relacionamentos muitos-para-muitos existente entre essas quatro dimensões. Geralmente, a tabela de fatos também armazena medições numéricas, onde cada uma das medições é obtida na interseção de todas as dimensões. Porém, uma peculiaridade acontece nesse diagrama (figura V.8), - não foi possível obter fatos mensuráveis. Isso significa que poderão ser fornecidas contagens de eventos de freqüência.

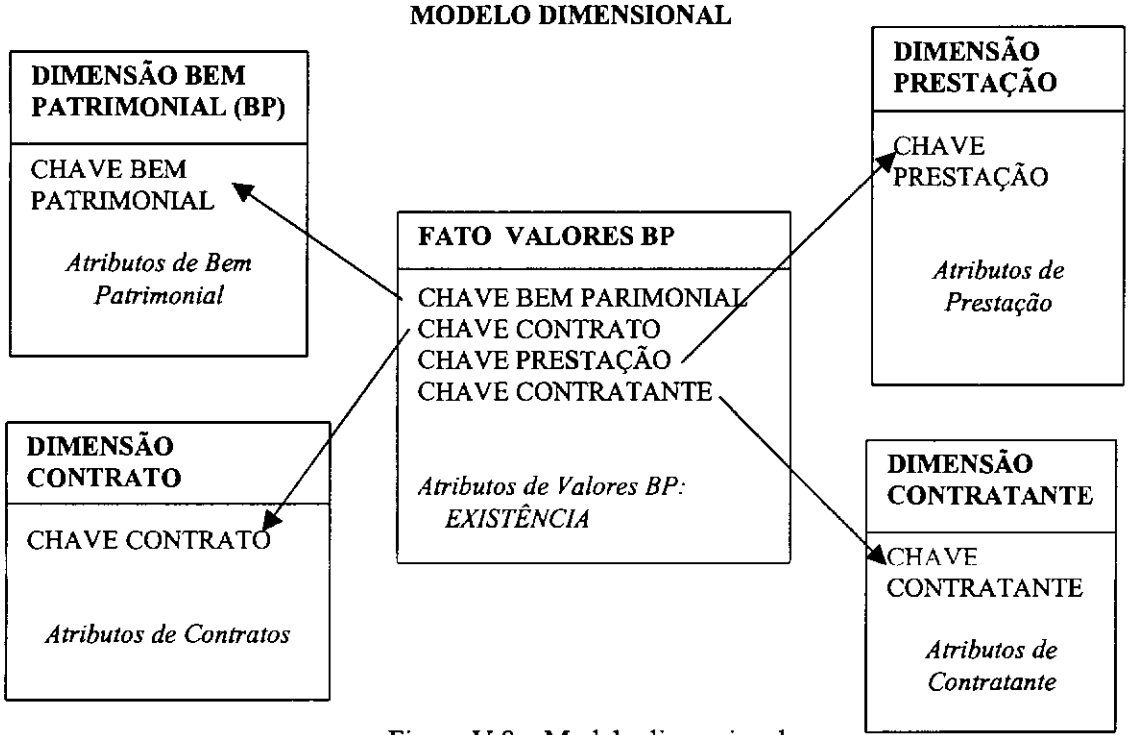


Figura V.8 – Modelo dimensional

V.4.2. Metadados

A representação do metadado é do tipo em que os novos dados quando entram no ambiente de dados são atualizados pelos operadores, seguindo uma definição que abrange a identificação e compreensão dos dados proporcionada pelo levantamento e pela visualização dos mesmos.

O metadado (figura V.9 e V.10) apresentado inclui uma tabela contendo descrição dos dados, o formato e definição em termos de sistemas e o exemplo de cada dado. E apresenta também a definição das chaves do modelo dimensional.

DADOS DE BEM PATRIMONIAL (BP)

ATRIBUTOS	DESCRIÇÃO	TIPO	TAMANHO	EXEMPLO
COD-TR-BP	CÓDIGO REGIONAL (NOT NULL)	NUMÉRICO	INTEGER	80
COD-BP	NUMERO DO BEM PATRIMONIAL (NOT NULL)	NUMÉRICO	DOUBLE	3002593
NUM-PCL-BP	NUMERO DAPARCELA DO BP (NOT NULL) EXEMPLO: UM TERRENO É DESMEMBRADO; O CÓDIGO DO BP PERMANECE E SÃO CRIADAS PARCELAS CORRESPONDENTES A CADA TERRENO, SENDO QUE A PARCELA ZERO ORIGINAL DEIXA DE EXISTIR	NUMÉRICO	INTEGER	20
DES-NLOGR-BP	ENDEREÇO DO BP (NOT NULL)	TEXTO	70	RUA NÓBREGA, Nº 62
DES-CLOGR-BP	COMPLEMENTO DO ENDERECO (NOT NULL)	TEXTO	30	CENTRO
NOME-MUNI-BP	NOME DO MUNICÍPIO (NOT NULL)		40	NITERÓI
CEP-BP	CEP DO BP (NOT NULL)	NUMÉRICO	LONG INTEGER	24220320

SIG-UF-BP	SIGLA DA UNIDADE FEDERAL DO BP (NOT NULL)	TEXTO	2	RJ
COD-NTZ-BP	CODIGO DA NATUREZA DO BP CADASTRAL (NOT NULL). O CÓDIGO PODE SER: BC-TERRENO/BENFEITORIA COMERCIAL BR-TERRENO/BENFEITORIA RESIDENCIAL FO-TELEFONE OB-OUTRAS BENFEITORIAS PP-PROPAGANDA TC-TERRENO COMERCIAL TR-TERRENO RESIDENCIAL	TEXTO	2	TR
DES-NTZ-BP	DESCRIÇÃO DA NATUREZA DO BP (NOT NULL)	TEXTO	35	TERRENO RESIDENCIAL
COD-SIT-BP	CODIGO DA SITUAÇÃO DO BP (NOT NULL) 'L' (LOCADO) 'A' (ALIENADO)	TEXTO	1	L
COD-IDF-CRCC-BP	CODIGO DO INDICE FINANCEIRO DE CORREÇÃO DO BP (NOT NULL) EX: 'UFIR'	TEXTO	10	UFIR
DATA-CRCC-BP	DATA DA ULTIMA CORREÇÃO DO BP PELO IDF (NOT NULL)	DATA		31/10/97
QTD-UNF-BP	QUANTIDADE DE UNIDADES FISCAIS DO BP (NOT NULL)	NUMERICO	DOUBLE	6703,7375
VAL-BP-CRCC	VALOR DO BP CORRIGIDO PELA UFIR (EM REAL) (NOT NULL)	NUMERICO	DOUBLE	5555,39
DDE-BP	DESCRIÇÃO DO BP (NOT NULL)	TEXTO	200	FAIXA DA LINHA 56609 M^2
COD-IGR-BP	COD DE INTEGRAÇÃO DO BP SE O BEM FOR UM IMOVEL QUE ESTA DENTRO DE UM TERRENO PERTENCENTE A EMPRESA: O BP DO TERRENO RECEBE O NUMERO DO BP DO IMOVEL NESTE CAMPO. O BP DO IMOVEL MANTEM ESTE CAMPO EM BRANCO.	NUMERICO	DOUBLE	3002593

NUM-IGR-BP	NUMERO-DE-INTEGRAÇÃO DO BP EQUIVALENTE AO NUMERO DA PARCELA DO BP (NO CASO INTEGRAÇÃO)	NUMERICO	INTEGER	0
TRE-OBS	OBSERVAÇÕES GERAIS	TEXTO	200	IMOVEL RUA SÃO PEDRO, Nº 154 SALA 303

Figura V.9 – Descrição dos dados de BP

Definição das chaves das dimensões

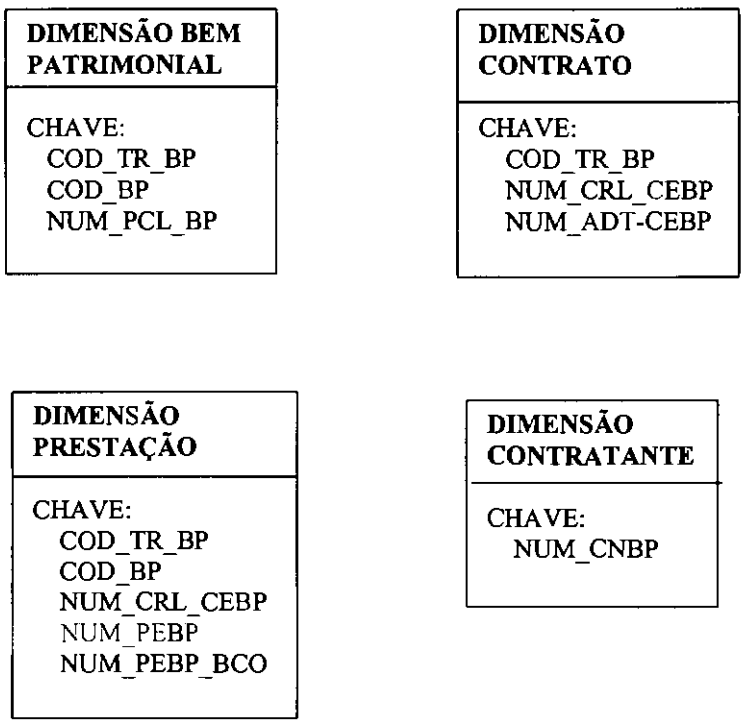


Figura V.10 – Definição das chaves das dimensões de BP

V.5. Fuzzy Query

Após a criação do novo modelo, uma coisa fica muito clara: é nesse novo cenário que se pretende realizar a recuperação das informações. Portanto, uma vez que o objetivo tenha sido definido, as consultas certas serão executadas com o auxílio da ferramenta Fuzzy Query [44]. A ferramenta escolhida para descobrir conhecimento do Data Mart está diretamente relacionada com a necessidade de se obter informações/conhecimentos que possuem uma enorme quantidade de incerteza.

V.5.1. Construindo uma consulta

a) No primeiro passo é preciso estabelecer um acesso aos dados. Esse acesso é feito por meio de uma conexão com um data source - onde os dados estão armazenados, no caso, com o arquivo BP_DMart .mdb da Microsoft Access 97, através de um ODBC.

b) Para começar a utilizar os dados, é necessário primeiro definir o universo de discurso de cada campo a ser usado na consulta. Como por exemplo, supondo que o campo (VAL-BP-CRCC) que se pretende usar na consulta possui o valor do bem patrimonial corrigido por uma moeda monetária (em real). Um universo razoável para esse campo contém um valor entre 100 reais e 55000 reais. Isso envolve inúmeros bens patrimoniais que estejam dentro desse universo, mas não é essencial que se cerque o extremo dos dados, pois todos podem ser tratados através da escolha dos limites do universo de discurso.

c) Uma vez definido o universo de discurso, então é preciso identificar e criar a semântica para cada subconjunto de dados do campo em questão. Como por exemplo (figura V.11), o campo VAL-BP-CRCC possui os seguintes subconjuntos:

Fuzzy Sets: VAL-BP-CRCC
Universe Discourse: 100 - 900000

FUZZY SETS: **PESSIMO**	ALPHA CUT = 0.3
	DOMAIN: 100 - 500
	FUZZY SET SHAPE: TRIANGLE
FUZZY SET: **RUIM**	ALPHA CUT = 0.3
	DOMAIN: 500 - 1000
	FUZZY SET SHAPE: TRIANGLE
FUZZY SET: **MEDIO**	ALPHA CUT = 0.3
	DOMAIN: 1000 - 5000
	FUZZY SET SHAPE: TRIANGLE
FUZZY SET: **BOM**	ALPHA CUT = 0.3
	DOMAIN: 5000 - 50000
	FUZZY SET SHAPE: TRIANGLE
FUZZY SET: **OTIMO**	ALPHA CUT = 0.3
	DOMAIN: 50000 - 900000
	FUZZY SET SHAPE: TRIANGLE

Figura V.11 – Exemplo da definição de subconjuntos fuzzy

Fuzzy Sets

É o nome dado ao subconjunto que pertence a um determinado universo de discurso, que pode ser RUIM, BOM, etc.

α-cut

É o parâmetro que especifica a pertinência mínima do subconjunto. Todos os valores de domínio que possuem valor menor do que este parâmetro são cortados. O parâmetro α-cut precisa ser um número entre 0 e 1 inclusive.

Qualquer conjunto fuzzy pode ser considerado como uma família de conjuntos fuzzy - este é um princípio conhecido também como um teorema de representação. Para explicar esta construção, é preciso definir uma noção de um α-cut de um conjunto fuzzy. O α-cut de A, A_α, é um conjunto fuzzy consistindo de elementos de X cujos valores de pertinência excedem o limite α-cut,

$$A_\alpha = \{ x \mid A(x) >= \alpha \}$$

Em outras palavras, A consiste de elementos de X que é identificado com A para um grau não menos que α. O nível mais alto nível, α = 1, determina um conjunto

X totalmente pertencendo a A. O nível mais baixo de α, é admitido para esse α-cut;

$$\text{Se } \alpha 1 > \alpha 2 \text{ então } A_{\alpha 1} \subset A_{\alpha 2}$$

O teorema de representação diz que qualquer conjunto fuzzy A pode estar decomposto em uma série de α-cut,

$$A = \cup_{\alpha \in [0,1]} (\alpha A_{\alpha})$$

Ou equivalente,

$$A(x) = \sup_{\alpha \in [0,1]} [(\alpha A_{\alpha}(x)]$$

Domínio

É o parâmetro que especifica o domínio máximo e mínimo do subconjunto. Esses valores precisam estar dentro dos valores do universo de discurso.

Forma da função de pertinência

É a representação do tipos de funções de pertinência, conforme já definido no capítulo III.

d) Depois da criação dos subconjuntos é necessário definir os critérios que se pretende utilizar para então construir as consultas. Como por exemplo (figura V.12 e V.13).

A: COD-NTZ-BP = TR
B: COD-NTZ-BP = BR
C: SIG-UF-BP = MG
D: SIG-UF-BP = ES
E: SIG-UF-BP = RJ

Figura V.12 - Para consultas Crisp

A: VAL-BP-CRCC is/are PESSIMO
B: VAL-BP-CRCC is/are RUIM
C: VAL-BP-CRCC is/are MEDIO
D: VAL-BP-CRCC is/are BOM
E: VAL-BP-CRCC is/are OTIMO
F: VAL-BP-CRCC is/are somewhat EXCELENTE
G: VAL-BP-CRCC is/are EXCELENTE
H: VAL-BP-CRCC is/are slightly EXCELENTE
I:VAL-BP-CRCC is/are slightly PESSIMO
J:VAL-BP-CRCC is/are slightly RUIM

Figura V.13 – Definição dos critérios de consulta fuzzy

e) O próximo passo é a construção da consulta fuzzy que pode ser ou não combinada com um ou mais critérios definidos previamente, usando os operadores AND e OR.

O construtor fuzzy possui dois tipos de lista: um para todo critério de consulta fuzzy e outro para todo critério de consulta crisp. Uma consulta consiste de duas partes: um comando qualificador crisp e um comando estimado fuzzy. Os critérios crisp são associados numa lista, podendo usar os operadores AND e OR para formar um comando crisp. De forma similar, os critérios fuzzy são associados em uma outra lista separada para formar um comando fuzzy.

O comando crisp é utilizado para qualificar um subconjunto de registros usando a linguagem de consulta padrão – o SQL. O comando crisp simplesmente ajuda a acelerar a velocidade de processamento da consulta, possibilitando a criação de um subconjunto de dados de interesse. Como por exemplo:

```
SELECT * FROM BP_DMART
        WHERE (SIG_UF_BP = "MG");
```

Nesse caso, o interesse da consulta se concentra somente no subconjunto de dados que pertence ao Estado de Minas Gerais. Após efetuada a construção, efetua-se a validade.

f) Como passo final, antes de executar a consulta, é preciso verificar alguns parâmetros na opção de controle da consulta, tais como:

1) Nome da tabela resultante – especificar o nome da tabela que irá conter o resultado da consulta.

2) Nome do campo QICX – especificar o nome da coluna do índice de compatibilidade de consulta, que irá ser incluído na tabela de resultado da consulta.

3) Limite QCIX – especifica o número de registros escritos na tabela de resultado, atribuindo um valor de compatibilidade mínimo. Esse limite é diferente do α-cut, pois esse limite é aplicado a uma pertinência resultante da composição da consulta estimada. O α-cut é aplicado para um conjunto fuzzy específico.

4) Campos da tabela resultado – especifica os campos da tabela resultado criada a partir da seleção dos campos do data source, e agregado ao campo do QCIX.

5) Consulta Fuzzy e opção AND – especifica os métodos de Min processing ou de Average processing. Dois tipos de métodos para processar o operador AND. Min processing obtém o mínimo dos valores do grupo da funcionalidade AND. Average processing retorna o valor mais significativo do grupode funcionalidade AND.

6) Opção Sort – especifica se apresenta a tabela de resultado com registros em ordem ascendente ou descendente baseado no campo QCIX.

g) E finalmente executa-se a consulta. Depois da execução da consulta, são apresentados os primeiros 1000 registros da tabela resultado. A tabela resultante é um arquivo com formato .csv.

V.5.2. Resultados das consultas Fuzzy

Na apresentação dos resultados, fica claro a importância de se obter o conhecimento de forma bem próxima do pensamento dos seres humanos, aproveitando a informação intrínseca nas incertezas para inferir e processar essas informações.

A consulta fuzzy é um processo bastante subjetivo e particular, ou seja, duas pessoas distintas podem modelar o mesmo problema de forma completamente diferente; depende muito da visão que cada um possui do problema, incluindo nesse caso, o objetivo específico que tem o modelo. Isso não é diferente para o estudo aqui abordado, principalmente por ser uma área bastante polêmica e relevante em quase todo o processo de tomada de decisão.

Como o escopo do problema definido é razoavelmente complexo e o processo de consulta fuzzy que envolve: a definição do universo de discurso, a criação dos subconjuntos fuzzy com seus respectivos domínios, α-cut, e tipos de pertinência, a definição de comandos de critério crisp e fuzzy, onde todos são elementos que requerem especificações muitas vezes estratégicas e delicadas, pode-se notar que não foi tão simples especificar uma consulta, haja visto que também foi preciso considerar as possíveis variáveis lingüísticas dentro da extensão do problema para melhor expressar a realidade. A classificação dos dados realizada através da tarefa de Data Mining e as entrevistas feitas com o especialista foram de fundamental importância para a especificação das consultas flexíveis.

Os parâmetros indicados para a realização das consultas fuzzy foram dimensionados de acordo com alguns aspectos, tais como: o domínio representa um conjunto fuzzy com um tipo de conhecimento usado para expressar o pensamento com base num senso comum - *algo, alguma coisa, usualmente, etc.*, portanto, a representação em curva-s que corresponde a superfícies não lineares crescente é bastante comum para esse tipo de representação, conforme visto no capítulo III. Outros parâmetros são o α-cut e QCIX, onde o limite QCIX especifica o número de

registros escritos na tabela de resultado, atribuindo um valor de compatibilidade mínima. Esse limite é diferente do α-cut, pois esse limite é aplicado a uma pertinência resultante da composição da consulta estimada. O α-cut é o parâmetro que especifica a pertinência mínima do subconjunto, possuindo um valor entre 0 e 1 inclusive. Todos os valores de domínio que possuem valor menor do que o α-cut são eliminados.

V.5.2.1. Principais consultas fuzzy

Conforme o objetivo geral deste trabalho, qual seja, o de estabelecer uma metodologia que permita analisar informações sobre bens patrimoniais de uma empresa de transporte, à luz da interpretação de dados armazenados em um Data Mart, serão apresentadas neste item as consultas flexíveis mais significativas. A interpretação dos resultados possibilita verificar a produção de um efeito diferencial e inovador na tomada de decisão em um ambiente competitivo.

V.5.2.1.1. Consulta que classifica a variação do valor das prestações dos bens patrimoniais da empresa

Embora não se identifique uma referência explícita, pode-se confirmar, através da análise dos dados e pela freqüência que a grande maioria dos bens patrimoniais estão locados (L) e alguns se encontram alienados (A), como indicado na figura V.14.

Sabe-se ainda que a classificação realizada de acordo com a natureza do bem patrimonial, apontam os bens de natureza BR (BP com benfeitoria residencial) e TR (terreno residencial) com um maior número de prestações, e logo a seguir estão os bens de natureza BC (BP com benfeitoria comercial) e TC (terreno comercial), como indicado na figura V.15.

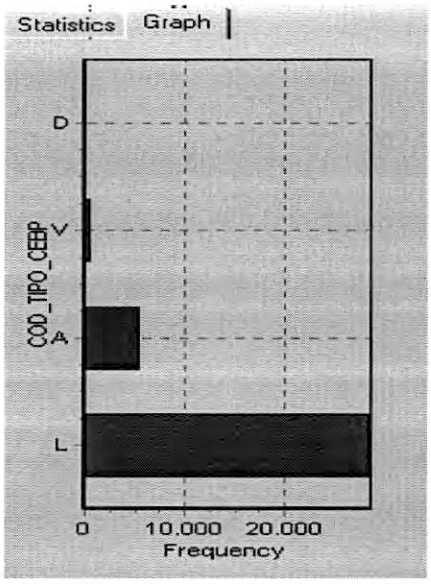


Figura V.14 – Representação gráfica da situação do atributo COD_TIPO_CEBP

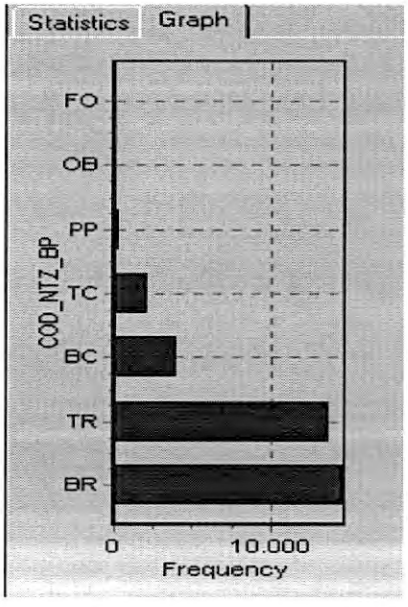


Figura V.15 – Representação gráfica da situação do atributo COD_NTZ_BP

a) Esta consulta permite investigar o valor total da prestação do bem patrimonial (VAL_TOT_PEBP) corrigido por uma moeda monetária (valor dado em real), onde a natureza do bem patrimonial considerada é o BR (BP com benfeitoria residencial) e TR (terreno residencial), dentro de um universo de discurso que varia entre 10 a 3500 reais. Os parâmetros utilizados são apresentados a seguir conforme as figuras V.16, V.17, V.18 e V.19.

FUZZY SETS: NADA ACEITAVEL	ALPHA CUT = 0.5
	DOMAIN: 10 – 200
	FUZZY SET SHAPE: GROWTH
FUZZY SET: POUCO ACEITAVEL	ALPHA CUT = 0.5
	DOMAIN: 200.99 – 300
	FUZZY SET SHAPE: GROWTH
FUZZY SET: MEDIO ACEITAVEL	ALPHA CUT = 0.5
	DOMAIN: 300.99 – 1000
	FUZZY SET SHAPE: GROWTH
FUZZY SET: ACEITAVEL	ALPHA CUT = 0.5
	DOMAIN: 1000.99 - 3000
	FUZZY SET SHAPE: GROWTH

Figura V.16 - Subconjuntos fuzzy

A: VAL-TOT-PEBP is/are somewhat NADA ACEITAVEL
B: VAL-TOT-PEBP is/are somewhat POUCO ACEITAVEL
C: VAL-TOT-PEBP is/are somewhat MEDIO ACEITAVEL
D: VAL-TOT-PEBP is/are somewhat ACEITAVEL

Figura V.17 - Critérios adotados

Consultas Fuzzy	**Parâmetro de controle/Resultado**
Fuzzy Query Evaluator Statement Workspace: A	QCIX: 0.1
	Fuzzy Query and Option: Min Processing
	Sort Option: Decreasing QCIX
	Resultado: 42% (com 14723 registros PRESTAÇÕES)
Fuzzy Query Evaluator Statement Workspace: B	QCIX: 0.1
	Fuzzy Query and Option: Min Processing
	Sort Option: Decreasing QCIX
	Resultado: 12% (com 4264 registros PRESTAÇÕES)

Fuzzy Query Evaluator Statement Workspace: C	QCIX: 0.1
	Fuzzy Query and Option: Min Processing
	Sort Option: Decreasing QCIX
	Resultado: 4% (com 1288 registros PRESTAÇÕES)
Fuzzy Query Evaluator Statement Workspace: D	QCIX: 0.1
	Fuzzy Query and Option: Min Processing
	Sort Option: Decreasing QCIX
	Resultado: 1% (com 386 registros PRESTAÇÕES)

Figura V.18 – Parâmetros de controle e Resultado da consulta fuzzy

Comandos para realizar a consulta fuzzy:

```
SELECT * FROM IR_ESCRJ_PREST
                WHERE (COD-NTZ-BP = "BR") OR
                        (COD-NTZ-BP = "TR");

Comando de consulta fuzzy usado para estimar registros:
        VAL-TOT-PEBP is/are somewhat ACEITAVEL

Comando SQL usado para criar tabela de resultado:
        CREATE TABLE Answer ('QCIX'Number, "COD-BP'Double,
                'NUM-CRL-CEBP' Short, 'NUM-PEBP' Short,
                'COD-NTZ-BP' Text(2), 'VAL-TOT-PEBP' Double
```

Figura V.19 – Exemplo dos comandos usados na consulta fuzzy

b) Esta consulta possui as mesmas características da consulta anterior, porém trata-se agora do bem patrimonial de natureza BC (BP com benfeitoria comercial) e TC (terreno comercial). A figura V.20 aponta os resultados desta consulta e a figura V.21 apresenta um exemplo dos comandos da consulta fuzzy realizada.

Consultas Fuzzy	**Resultado**
Fuzzy Query Evaluator Statement Workspace: A	67% (com 4060 registros BP)
Fuzzy Query Evaluator Statement Workspace: B	30% (com 1838 registros BP)
Fuzzy Query Evaluator Statement	10% (com 586 registros BP)

Workspace: C	
Fuzzy Query Evaluator Statement Workspace: D	3% (com 210 registros BP)

Figura V.20 – Reultado da consulta fuzzy

Comandos de uma consulta fuzzy:

```
SELECT * FROM IR_ESCRJ_PREST
        WHERE (COD-NTZ-BP = "BC")  OR
                (COD-NTZ-BP = "TC");

Comando de consulta fuzzy usado para estimar registros:
        VAL-TOT-PEBP is/are somewhat ACEITAVEL

Comando SQL usado para criar tabela de resultado:
        CREATE TABLE Answer ('QCIX'Number, "COD-BP'Double,
                'NUM-CRL-CEBP' Short, 'NUM-PEBP' Short,
                'COD-NTZ-BP' Text(2), 'VAL-TOT-PEBP' Double
```

Figura V.21 – Exemplo dos comandos usados na consulta fuzzy

c) Esta consulta também possui as mesmas características da anterior, isto é, possui os mesmos parâmetros, porém classifica a variação do valor das prestações dos bens patrimoniais da companhia de forma global, ou seja, considera todos as natureza dos bens (COD_NTZ_BP) numa mesma análise. A figura V.22 aponta os resultados desta consulta e a figura V.23 apresenta um exemplo dos comandos da consulta fuzzy realizada.

Consultas Fuzzy	**Parâmetro de controle**
Fuzzy Query Evaluator Statement Workspace: A	42% (com 14723 registros PRESTAÇÕES)
Fuzzy Query Evaluator Statement Workspace: B	12% (com 4264 registros PRESTAÇÕES)
Fuzzy Query Evaluator Statement Workspace: C	4% (com 1288 registros PRESTAÇÕES)
Fuzzy Query Evaluator Statement Workspace: D	1% (com 386 registros PRESTAÇÕES)

Figura V.22 – Reultado da consulta fuzzy

Comandos de uma consulta fuzzy:

SELECT * FROM IR_ESCRJ_PREST

Comando de consulta fuzzy usado para estimar registros:
VAL-TOT-PEBP is/are somewhat ACEITAVEL

Comando SQL usado para criar tabela de resultado:
CREATE TABLE Answer ('QCIX'Number, "COD-BP'Double,
'NUM-CRL-CEBP' Short, 'NUM-PEBP' Short,
'COD-NTZ-BP' Text(2), 'VAL-TOT-PEBP' Double

Figura V.23 – Exemplo dos comandos usados na consulta fuzzy

Através dessas consultas fuzzy (figuras V.24, V.25 e V.26), se tornou possível identificar o impacto caótico sobre o estado em que se encontra o patrimônio da companhia. Considerando o valor da natureza do BP de forma geral, tem-se: 42% das prestações recebidas são consideradas nada aceitáveis, 12% são pouco aceitáveis, 4% são medianamente aceitáveis e somente 1% dessas prestações são consideradas aceitáveis. Na verdade, esse impacto acontece se apenas levar em conta as prestações como medida de avaliação dos bens patrimoniais. É preciso prestar muita atenção para identificar onde a análise funciona e onde não. Todas as medidas para uma tomada de decisão são valiosas.

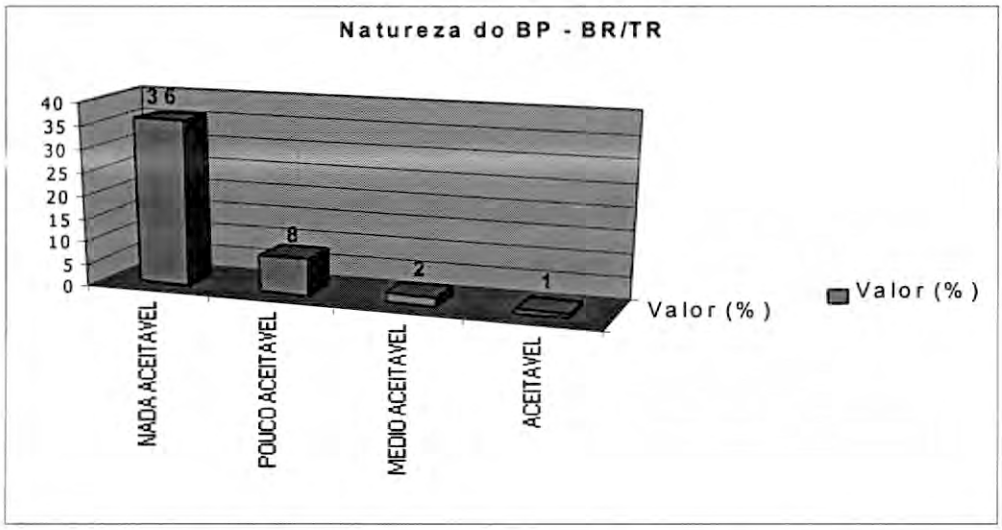


Figura V.24 - Resultado da análise fuzzy sobre VAL_TOT_PEBP

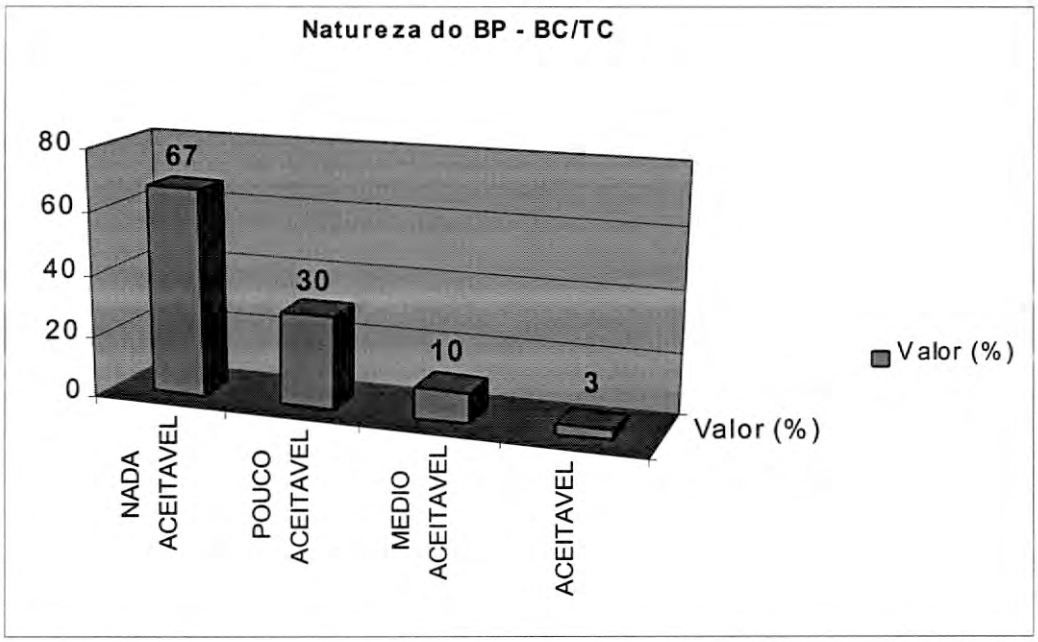


Figura V.25 – Resultado da análise fuzzy sobre os valores das prestações dos bens patrimoniais

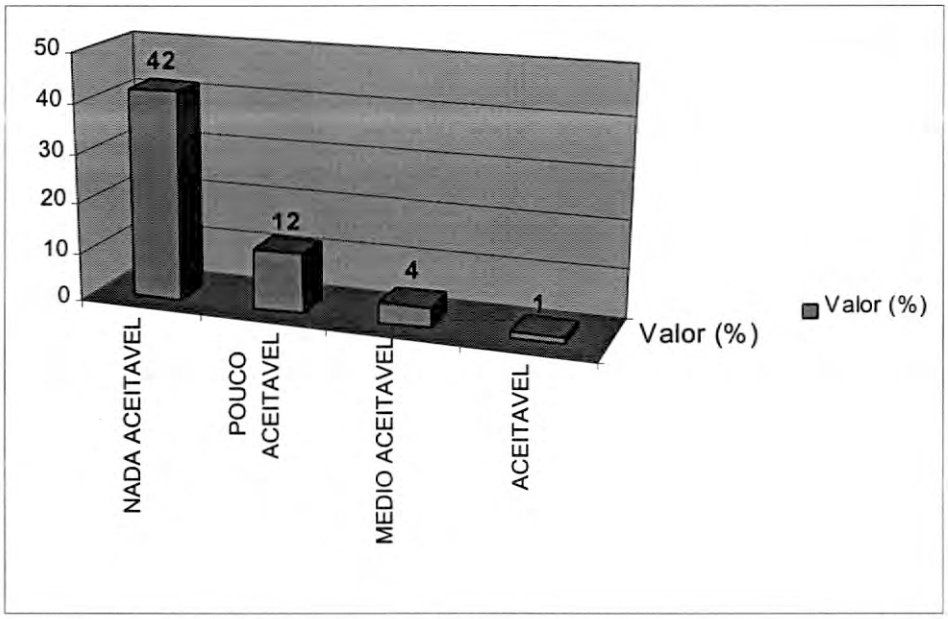


Figura V.26 - Resultado da análise fuzzy sobre os valores das prestações dos bens patrimoniais

V.5.2.1.2. Consulta que focaliza a venda dos bens patrimoniais da companhia

Existe duas considerações sobre a venda dos bens patrimoniais: a primeira considera que venda é realizada de forma distinta, ou seja, é realizada separadamente de acordo com os estados, como: RJ, MG e ES. A segunda considera que a venda é realizada pela companhia de forma global, isto é, considera todos os bens numa mesma análise. Sabe-se diante mão, através da classificação realizada, como mostra a figura V.27, que os bens patrimoniais se concentram mais no estado do Rio de Janeiro, seguido pelo de Minas Gerais e depois pelo do Espirito Santos.

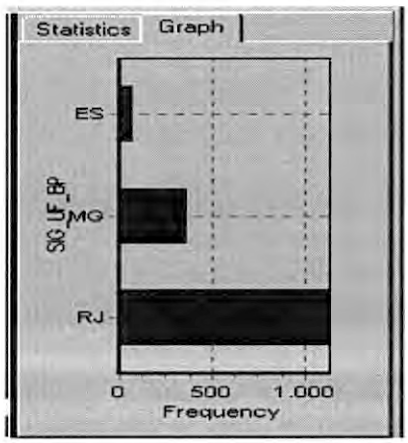


Figura V.27 - Representação gráfica da situação do atributo SIG_UF_BP

a) Esta consulta permite pesquisar o valor do bem patrimonial (VAL_BP_CRCC) corrigido por uma moeda monetária (valor dado em real), sendo que a localização considerada dos bens patrimoniais é o estado do Rio de Janeiro, dentro de um universo de discurso que varia entre 1000 a 900000 reais. Os parâmetros utilizados são apresentados a seguir conforme as figuras V.28, V.29, V.30 e V.31.

FUZZY SETS: MUITO RUIM	ALPHA CUT = 0.5
	DOMAIN: 1000 - 5000
	FUZZY SET SHAPE: GROWTH
FUZZY SET: RUIM	ALPHA CUT = 0.5
	DOMAIN: 5000.99 - 10000
	FUZZY SET SHAPE: GROWTH
FUZZY SET: MEDIO	ALPHA CUT = 0.5
	DOMAIN: 10000.99 - 15000
	FUZZY SET SHAPE: GROWTH

FUZZY SET: BOM	ALPHA CUT = 0.5
	DOMAIN: 15000.99 - 20000
	FUZZY SET SHAPE: GROWTH
FUZZY SET: BOM	ALPHA CUT = 0.5
	DOMAIN: 20000.99 - 40000
	FUZZY SET SHAPE: GROWTH
FUZZY SET: OTIMO	ALPHA CUT = 0.5
	DOMAIN: 40000.99 - 55000
	FUZZY SET SHAPE: GROWTH
FUZZY SET: EXCELENTE	ALPHA CUT = 0.5
	DOMAIN: 55000.99 - 900000
	FUZZY SET SHAPE: GROWTH

Figura V.28 - Subconjuntos fuzzy

A: VAL-BP-CRCC is/are somewhat MUITO RUIM
B: VAL-BP-CRCC is/are somewhat RUIM
C: VAL-BP-CRCC is/are somewhat MEDIO
D: VAL-BP-CRCC is/are somewhat BOM
E: VAL-BP-CRCC is/are somewhat MUITO BOM
F: VAL-BP-CRCC is/are somewhat OTIMO
G: VAL-BP-CRCC is/are EXCELENTE
H: VAL-BP-CRCC is/are VERY EXCELENTE

Figura V.29 - Critérios adotados

Consultas Fuzzy	**Parâmetro de controle**
Fuzzy Query Evaluator Statement Workspace: A	QCIX: 0.1
	Fuzzy Query and Option: Min Processing
	Sort Option: Decreasing QCIX
	Resultado: 19% (com 213 registros BP)
Fuzzy Query Evaluator Statement Workspace: B	QCIX: 0.1
	Fuzzy Query and Option: Min Processing
	Sort Option: Decreasing QCIX
	Resultado: 14% (com 163 registros BP)
Fuzzy Query Evaluator Statement Workspace: C	QCIX: 0.1
	Fuzzy Query and Option: Min Processing
	Sort Option: Decreasing QCIX
	Resultado: 10% (com 113 registros BP)
Fuzzy Query Evaluator Statement Workspace: D	QCIX: 0.1
	Fuzzy Query and Option: Min Processing
	Sort Option: Decreasing QCIX
	Resultado: 5% (com 59 registros BP)
Fuzzy Query Evaluator Statement	QCIX: 0.1

Workspace: E	
	Fuzzy Query and Option: Min Processing
	Sort Option: Decreasing QCIX
	Resultado: 3% (com 32 registros BP)
Fuzzy Query Evaluator Statement Workspace: F	QCIX: 0.1
	Fuzzy Query and Option: Min Processing
	Sort Option: Decreasing QCIX
	Resultado: 2% (com 17 registros BP)
Fuzzy Query Evaluator Statement Workspace: G	QCIX: 0.1
	Fuzzy Query and Option: Min Processing
	Sort Option: Decreasing QCIX
	Resultado: 0% (com 4 registros BP)
Fuzzy Query Evaluator Statement Workspace: H	QCIX: 0.1
	Fuzzy Query and Option: Min Processing
	Sort Option: Decreasing QCIX
	Resultado: 0% (com 4 registros BP)

Figura V.30 – Parâmetros de controle e Resultado da consulta fuzzy

Comandos da consulta fuzzy:

Comando SQL usado para obter o subconjunto de interesse:

```
SELECT * FROM IR_ESCRJ_BP
        WHERE (SIG-UF-BP = "RJ");
```

Comando de consulta fuzzy usado para estimar registros:

```
VAL-BP-CRCC is/are somewhat BOM
```

Comando SQL usado para criar tabela de resultado:

```
CREATE TABLE Answer ('QCIX'Number, "COD-BP'Double,
        'NUM-PCL-BP' Short, 'VAL-BP-CRCC' Double,
        'COD-NTZ-BP' Text(2), 'COD-SIT-BP' Text(1))
```

Figura V.31 – Exemplo dos comandos usados na consulta fuzzy

b) Esta consulta possui as mesmas características da consulta anterior, porém trata-se agora do Estado de Minas Gerais. A figura V.32 aponta os resultados desta consulta e a figura V.33 apresenta um exemplo dos comandos da consulta fuzzy realizada.

Consultas Fuzzy	Resultado
Fuzzy Query Evaluator Statement Workspace: A	36% (com 129 registros BP)
Fuzzy Query Evaluator Statement Workspace: B	30% (com 107 registros BP)
Fuzzy Query Evaluator Statement Workspace: C	22% (com 80 registros BP)
Fuzzy Query Evaluator Statement Workspace: D	16% (com 58 registros BP)
Fuzzy Query Evaluator Statement Workspace: E	8% (com 30 registros BP)
Fuzzy Query Evaluator Statement Workspace: F	4% (com 14 registros BP)
Fuzzy Query Evaluator Statement Workspace: G	0% (com 1 registros BP)
Fuzzy Query Evaluator Statement Workspace: H	0% (com 1 registros BP)

Figura V.32 - Reultado da consulta fuzzy

Comandos de uma consulta fuzzy:

Comando SQL usado para obter o subconjunto de interesse:

```
SELECT * FROM IR_ESCRJ_BP
        WHERE (SIG-UF-BP = "MG");
```

Comando de consulta fuzzy usado para estimar registros:

```
VAL-BP-CRCC is/are somewhat MÉDIO
```

Comando SQL usado para criar tabela de resultado:

```
CREATE TABLE Answer ('QCIX'Number, "COD-BP'Double,
        'NUM-PCL-BP' Short, 'VAL-BP-CRCC' Double,
        'COD-NTZ-BP' Text(2), 'COD-SIT-BP' Text(1))
```

Figura V.33 – Exemplo dos comandos usados na consulta fuzzy

c) Esta consulta possui as mesmas características da consulta anterior, porém trata-se agora do Estado de Espirito Santos. A figura V.34 aponta os resultados desta consulta e a figura V.35 apresenta um exemplo dos comandos da consulta fuzzy realizada.

Consultas Fuzzy	Resultado
Fuzzy Query Evaluator Statement Workspace: A	36% (com 28 registros BP)
Fuzzy Query Evaluator Statement Workspace: B	30% (com 23 registros BP)
Fuzzy Query Evaluator Statement Workspace: C	17% (com 13 registros BP)
Fuzzy Query Evaluator Statement Workspace: D	9% (com 7 registros BP)
Fuzzy Query Evaluator Statement Workspace: E	4% (com 3 registros BP)
Fuzzy Query Evaluator Statement Workspace: F	3% (com 2 registros BP)
Fuzzy Query Evaluator Statement Workspace: G	0% (com 0 registros BP)
Fuzzy Query Evaluator Statement Workspace: H	0% (com 0 registros BP)

Figura V.34 - Reultado da consulta fuzzy

Comandos de uma consulta fuzzy:

Comando SQL usado para obter o subconjunto de interesse:

```
SELECT * FROM IR_ESCRJ_BP
        WHERE (SIG-UF-BP = "ES");
```

Comando de consulta fuzzy usado para estimar registros:

```
VAL-BP-CRCC is/are somewhat ÓTIMO
```

Comando SQL usado para criar tabela de resultado:

```
CREATE TABLE Answer ('QCIX'Number, "COD-BP'Double,
        'NUM-PCL-BP' Short, 'VAL-BP-CRCC' Double,
        'COD-NTZ-BP' Text(2), 'COD-SIT-BP' Text(1))
```

Figura V.35 – Exemplo dos comandos usados na consulta fuzzy

d) Esta consulta também possui as mesmas características da anterior, isto é, possui os mesmos parâmetros, porém porém trata-se agora dos estados do Rio de Janeiro, Minas Gerais e Espírito Santo, ou seja, considera todos os estados numa mesma análise. A figura V.36 aponta os resultados desta consulta e a figura V.37 apresenta um exemplo dos comandos da consulta fuzzy realizada.

Consultas Fuzzy	Resultado
Fuzzy Query Evaluator Statement Workspace: A	24% (com 370 registros BP)
Fuzzy Query Evaluator Statement Workspace: B	19% (com 293 registros BP)
Fuzzy Query Evaluator Statement Workspace: C	13% (com 206 registros BP)
Fuzzy Query Evaluator Statement Workspace: D	8% (com 124 registros BP)
Fuzzy Query Evaluator Statement Workspace: E	4% (com 65 registros BP)
Fuzzy Query Evaluator Statement Workspace: F	1% (com 9 registros BP)
Fuzzy Query Evaluator Statement Workspace: G	0% (com 5 registros BP)
Fuzzy Query Evaluator Statement Workspace: H	0% (com 4 registros BP)

Figura V.36 - Reultado da consulta fuzzy

Comandos de uma consulta fuzzy:

SELECT * FROM IR_ESCRJ_BP

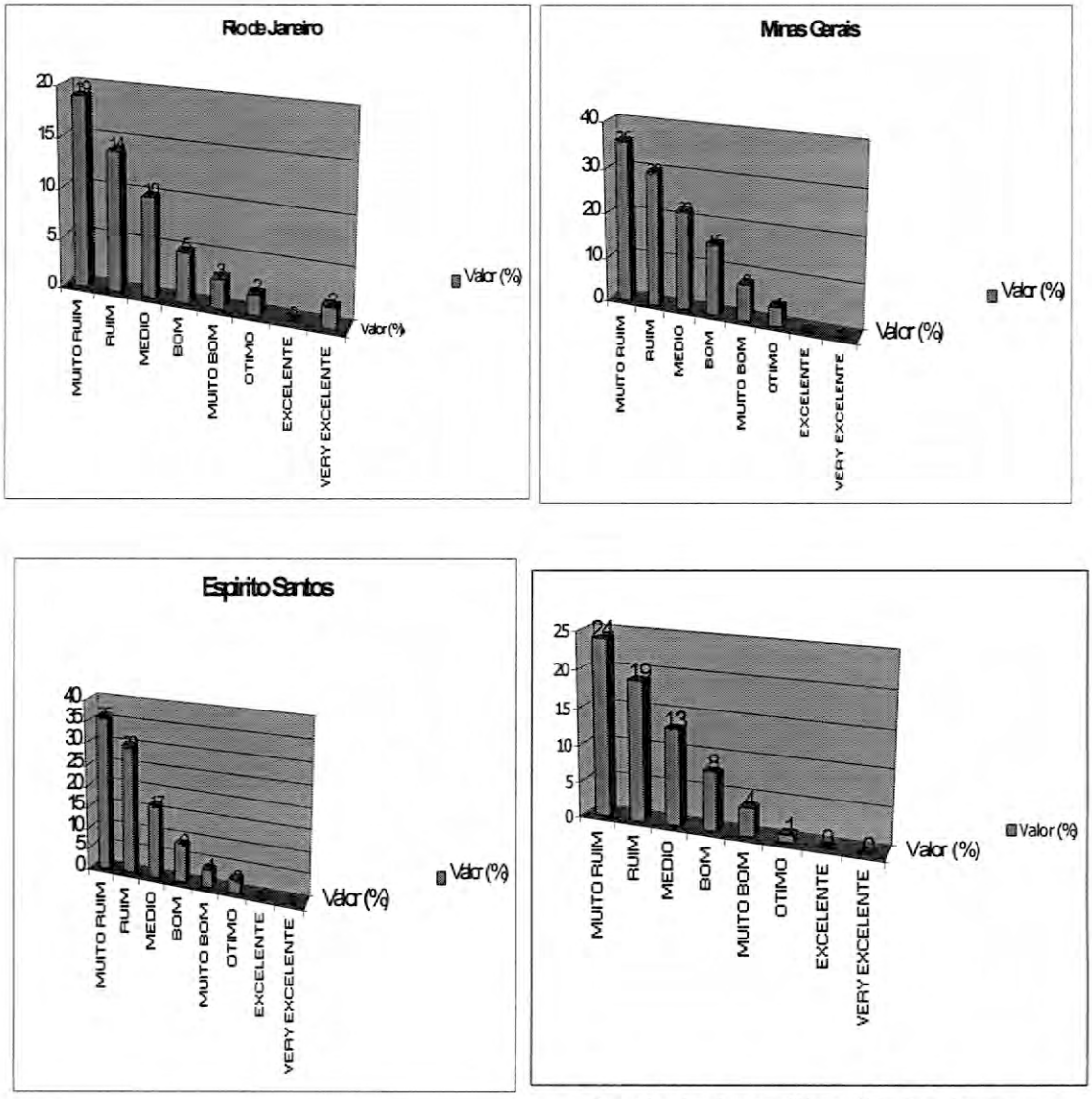
Comando de consulta fuzzy usado para estimar registros:
VAL-BP-CRCC is/are somewhat ÓTIMO

Comando SQL usado para criar tabela de resultado:
CREATE TABLE Answer ('QCIX'Number, "COD-BP'Double,
'NUM-PCL-BP' Short, 'VAL-BP-CRCC' Double,
'COD-NTZ-BP' Text(2), 'COD-SIT-BP' Text(1))

Figura V.37 – Exemplo dos comandos usados na consulta fuzzy

Considerando os resultados das análises fuzzy sobre o valor dos Bens Patrimoniais, conforme mostra a figura V.38, pode-se observar que a opção pela venda de forma integrada desses bens é a princípio a solução mais visível e apropriada. Embora a análise dessa consulta fuzzy seja subjetiva e pessoal, havendo inclusive a possibilidade de pensamentos diferentes, é importante salientar que uma

boa especificação sobre os valores lingüisticos compreende um resultado preciso e ainda admite maior flexibilidade do que os números.



Resultado das análises fuzzy sobre as vendas dos Bens Patrimoniais considerando todos os Estados

Figura V.38 – Resultado das análises fuzzy sobre as vendas dos Bens Patrimoniais

Antes de mais nada, vale ressaltar que a análise fuzzy realizada sobre o valor dos bens patrimoniais corrigidos monetariamente pode não corresponder ao valor de mercado, lembrando que o valor de mercado deve ser considerado posteriormente na

tomada de decisão, que certamente irá envolver todas as considerações contextuais, como por exemplo: valor patrimonial (bens físicos), valor da marca, potencial de mercado, etc.

V.6. Metodologia CFIC (Consultas Flexíveis para Inteligência Competitiva)

A metodologia CFIC procura responder a um momento atual, que gira em torno dos problemas operacionais que residem nas dificuldades em descobrir conhecimentos a partir de grandes volumes de dados armazenados e acumulados nos mais diversos sistemas de banco de dados. Propõe também a aplicação de técnicas para trabalhar fundamentalmente com a imprecisão instalada no conjunto de idéias de um assunto de negócio. E para isto, toma os principais paradigmas apresentados ao longo deste trabalho para a busca de conhecimentos comprovadamente coerentes, especulativos, intuitivos e de senso comum.

O estudo de caso apresentado neste capítulo é uma caracterização abrangente para registrar dados de um caso particular a fim de organizar uma experiência e avaliá-la, objetivando tomar decisões a seu respeito ou propor ações transformadoras para serem introduzidas numa gerência de estratégia em um ambiente de negócio competitivo.

A consolidação da metodologia deve-se a própria prática desse estudo de caso, onde pode privilegiar a verificação por meio de técnicas e ferramentas sofisticadas, alcançando um resultado bastante significativo. Essas técnicas foram descritas em capítulos anteriores, dinamizando a progressão da pesquisa e sua sistematização, explicando os meios mais eficientes para eliminar a dispersão e perda de informações.

Com esses pressupostos criou-se uma metodologia, através de técnicas, instrumentos e procedimentos de experimentação que foram aperfeiçoados para assegurar a objetividade do conhecimento, o critério e o rigor, a coerência e a relevância da aplicabilidade dessa metodologia.

A metodologia que envolve muitas variantes, trata do ambiente de dados, da exploração dos dados e da incertezas nas informações. As diferentes vias de

descoberta de conhecimentos podem ser condensadas em um modelo-padrão que inclui quatro fases, como mostra a figura V.39.

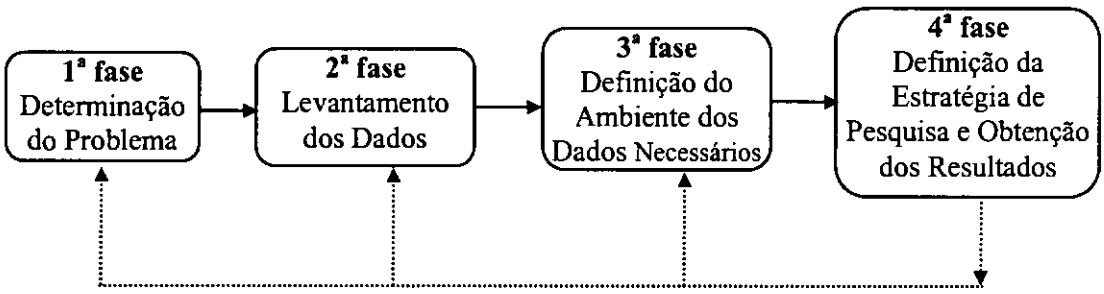


Figura V.39 - Representação das Fases da Metodologia CFIC

1ª fase: Determinação do problema

a) Selecionar o assunto – investigar o assunto, determinado em função de um interesse específico.

b) Definir e formular o problema – é uma etapa na qual se elege um núcleo particular, onde se pressupõe um inventário crítico sobre o problema de um assunto de negócio. A seleção e determinação do assunto de negócio são decisivas para a análise da situação estudada. É uma fase complexa e laboriosa de análise e sínteses, em que não existem regras fixas.

2ª fase: Levantamento dos dados

a) Realizar o levantamento de todo material. Selecionar a documentação mais importante, aproveitando-se do acervo de conhecimentos atualizados.

b) Realizar entrevistas com a finalidade de colher informações de forma livre, estruturada ou semi-estruturada. Todas as informações coletadas e observadas servem como indicadores e índices objetivos de variáveis que se pretende explorar.

c) Visualização e exploração dos dados através da técnica de Data Mining, como por exemplo - árvore de decisão, onde seu principal objetivo é propor uma classificação do conjunto de dados reunidos, ampliando as

possibilidades de correlação, comparação, compreensão e análise de uma gama considerável de dados.

3ª fase: Definição do ambiente dos dados necessários

A delimitação de um ambiente não é um resultado de uma afirmação simples e individual. Esse ambiente afigura-se acima de tudo como um contexto que abriga os dados relevantes de forma integrada, destinado apenas para fornecer as referências para as análises e interpretações sobre o assunto de negócio em um tempo bastante hábil e garantindo o objetivo da inteligência competitiva.

É importante salientar que no desenvolvimento, os dados colhidos são constantemente analisados e avaliados, ou seja, a coleta e seleção de dados é realizada de maneira iterativa. Assim, para escolha desse ambiente é preciso absorver um processo evolucionário, um ambiente integrado, um ambiente não-volátil, como é o caso do Data Mart.

4ª fase: Definição da estratégia de pesquisa e obtenção de resultados

Antes de se iniciar qualquer pesquisa é necessário prever os objetivos que se tem em vista, a área de abrangência e o conjunto de estratégias para identificar os problemas ou necessidades prioritárias e fazer um inventário-síntese das questões mais fluentes e das necessidades primordiais. Supõe-se assim elaborar previamente os objetivos e eleger as técnicas de pesquisa.

Após a elaboração dos objetivos que sustentam a formulação das necessidades é preciso então centrar na adoção de técnicas e instrumentos de pesquisa para posteriormente se efetivar a realização das análises dos resultados.

A técnica de pesquisa aqui utilizada e denominada de "fuzzy query", envolve o paradigma de consultas flexíveis que propõe retratar uma realidade, revelando a multiplicidade dos aspectos imprecisos, presentes em uma dada situação de negócio. O processo de consulta flexível não obedece a um padrão específico. Há diversas possibilidades na sua elaboração, vale muito o trabalho criativo e a reflexão sobre indagações nebulosas. O resultado das consultas flexíveis converge para um conjunto

de decisões que irá validar um conhecimento criado, a fim de eleger as estratégias de ação mais adequada à solução dos problemas.

V.7. Considerações finais

O ponto central deste estudo é a concepção da importância da integração das características de todos os paradigmas das ferramentas aqui apresentadas e utilizadas, pois a visualização da sua adequabilidade é bastante óbvia em vários pontos do processo. A modelagem do Data Mart representa todas as necessidades requeridas, expressando uma visão particular do universo do sistema, na qual tem-se um modelo dimensional que reproduz exatamente o que se deseja. Considerações sobre a complexidade do sistema e limites das suas necessidades, tratamento e análise das suas informações foram bastantes exploradas com o propósito de revelar um modelo o mais próximo possível da realidade proporcionando assim, uma capacidade de se organizar em torno de um interesse comum. A visualização dos dados foi inteiramente realizada com o auxílio da técnica de data mining, gerando um conforto significativo no que diz respeito a credibilidade do resultado do tratamento dos dados.

Vale ainda ressaltar que inicialmente a empresa decidiu fazer uma subdivisão das tarefas planejadas uma por uma de acordo com prioridades determinadas, onde a primeira está direcionada para a obtenção de um feedback excelente quanto ao destino dos seus bens patrimoniais. Tal meta tornou-se o fator preponderante para a opção pela técnica de Data Mart, pois além de ser considerada como menos dispendiosa, com mais facilidade para obter os dados necessários para o processamento, e consequentemente com uma implementação mais rápida, essa tarefa absorve somente uma parte das informações contidas no banco de dados da empresa, o que na realidade é o ponto fundamental para a escolha dessa técnica.

Além dessa representação, é preciso ir mais longe, ou seja, é preciso se ater para a existência de incertezas não só associadas aos dados, mas também associadas

ao universo das consultas que possuem indagações nebulosas, buscando conhecimento relacionados com raciocínios inexatos. Enfim, neste estudo existe diferentes e variados aspectos ligados às incertezas que foram dimensionados e tratados.

A análise dos dados, considerando essencialmente a forte necessidade de recuperar informações que possuem um certo grau de incompletude e compreensão, baseia-se na técnica de consultas fuzzy. A partir dessa técnica foi possível explorar e analisar as informações alcançando um objetivo muito próximo do ideal.

CAPÍTULO VI

CONCLUSÕES

Nas últimas três décadas, a informática galgou sucessos na área empresarial. A política, e até mesmo as posturas competitivas, também tiveram um papel significativo para esse sucesso. Na realidade algumas limitações tecnológicas inibiram um progresso até mais rápido. Mas com o passar do tempo surgem novas tecnologias, e assim o progresso é feito. Não existe fim para aplicação de tecnologia de computador, esta é a visão. O espectro do potencial das aplicações é muito grande, todo avanço tecnológico possui a força da atualização, acenando sempre para a próxima fase da tecnologia da computação.

A proximidade do século 21 tem despertado nos meios acadêmicos e empresariais um grande interesse sobre as mudanças que estão para ocorrer no novo milênio e seus impactos nos negócios e mercados. A busca da competitividade em um sentido mais amplo e abrangente está alterando os conceitos e técnicas tradicionais de gerenciamento e formatação nos negócios.

No entanto, o termo competitividade tem sido utilizado de forma generalizada e abrangente, dificultando a compreensão objetiva de seus aspectos e impactos nas estratégias empresariais.

A primeira vertente para gerar competitividade é a implementação de técnicas de gestão para melhorar a produtividade e eficiência nas operações internas das empresas. As técnicas mais utilizadas são a reengenharia, downsizing, tercerização, ISOs, just-in-time, entre outras tantas. Sua limitação é que elas criam diferenciais competitivos de curto prazo, pois são rapidamente reproduzidas pela concorrência.

Por outro lado, em um mundo sem fronteiras e com consumidores cada vez mais exigentes, não basta uma empresa ser eficiente e produtiva para sobreviver. Para ser competitiva ela necessita formular estratégias para aumentar sua inserção no mercado. Neste sentido, uma das principais estratégias inclui a descoberta de conhecimento. Muitas empresas descobriram novas maneiras de aumentar conhecimentos cruzados e mais sofisticados. Elas estão adquirindo esses

conhecimentos a partir de seus Data Warehouses ao aplicar técnicas novas e mais eficazes de exploração de dados.

A tecnologia da informação, que já é crítica para os processos básicos das empresas, está sendo mais disseminada e gerida pelas unidades de negócios individuais. Então, dentro de uma empresa todos, sem exceção, irão trabalhar com a tecnologia de informação, utilizando-as para atingir resultados tangíveis na empresa.

A partir desse prisma, um dos pontos centrais sobre o estudo efetuado nesta tese é a visão clara e precisa sobre a necessidade de se desenvolver uma metodologia para organizar os dados disponíveis dentro de um ambiente central integrado. O ambiente aqui gerado usou a tecnologia de Data Mart na sua construção. Ele possui um padrão binário de utilização, ou seja, o hardware está sendo totalmente usado ou não está, sendo que calcular um percentual médio relativo à sua utilização é uma tarefa sem sentido, por isso foi implementado numa máquina contendo somente o seu processamento.

Como resultado da aplicação do Data Mart, concluiu-se que a falta de documentação formal e compreensiva dos dados armazenados no sistema de banco de dados da empresa tem sua origem no volume muito grande de dados a serem manipulados, e que sem o controle sobre esses dados seria muito difícil projetar um ambiente central e integrado de informações. Assim, definiu-se como núcleo do sistema o projeto do Data Mart, e procedeu-se à classificação deles conforme a função, a caracterização dos campos e a identificação das relações entre os arquivos. A integração dos dados representa economia de recursos computacionais, segurança e confiabilidade da base, mas também significa, uma grande flexibilidade do sistema quanto ao número de possibilidades de recuperação de informações cruzadas, uma vez que inúmeras combinações de dados inevitavelmente surgirão nas pesquisas futuras.

O processo de introdução dos dados no ambiente do Data Mart foi realizado de forma consistente e independente da aplicação de origem. Na passagem do nível operacional para o nível do Data Mart ocorreu uma quantidade significativa de transformações sobre os dados.

Os dados extraídos passaram por um processo exaustivo de transformação, tendo em vista a precariedade da documentação de que dispunha o sistema e a quantidade significativa de ruídos instalada nesse banco de dados, eliminando as

inconsistências, sumarizando e povoando o ambiente de Data Mart. Assim o processo de transformação de dados desempenhou um papel bastante ativo, possibilitando mais fluidez, agilidade e criatividade no projeto e construção do ambiente.

Contudo, sabe-se que os melhores dados para qualquer empresa são os próprios registros que ela possui. A qualidade dos dados vem logo a seguir. Normalmente, um elemento de dados deveria ser bastante interessante se fosse de alta qualidade, mas não é a realidade, entretanto, este ambiente integrado não pode aprimorar dados de baixa qualidade, pois para aprimorá-los, foi preciso realizar uma reengenharia de dados disponíveis.

A necessidade de se ter qualidade dos dados – metadados, impulsionou a possibilidade de realizar um projeto de reengenharia de dados na empresa, e uma boa forma para justificar tal projeto foi publicar os dados incompletos e esperar que surja uma pressão natural na organização, pois assim que perceberem o quanto esses dados poderiam ser valiosos se fossem de melhor qualidade, então provavelmente os responsáveis pela coleta de dados e a gerência devem retornar à fonte dos dados com sistemas melhores, com gerenciamento melhor e com uma melhor visibilidade dos dados com valores adequados. Desse modo, a construção de um ambiente central integrado, pode desempenhar um papel-chave nos esforços para efetuar uma reengenharia de dados na empresa.

Especificamente esta pesquisa foi direcionada para trabalhar com o tratamento e manipulação de grandes massas de dados, com o tratamento e manipulação de massas de dados envolvendo muitos parâmetros relevantes, e também com a preparação dos dados e consideração de incertezas através da lógica "fuzzy".

Assim, um outro aspecto considerável pode ser observado, ou seja, pode-se concluir que uma ferramenta de Data Mining e uma ferramenta de consulta são perfeitamente complementares, ou seja, pode-se verificar que uma ferramenta de DM não se coloca no lugar de uma ferramenta de consulta, porém ela oferece um conjunto adicional de possibilidades.

Reforçando um aspecto já discutido no texto desta tese, sabe-se que a utilidade da inteligência competitiva está diretamente associada à melhor condição de gerenciamento da organização (ou de sua atividade) sob a pressão do dia-a-dia (e do tempo), num contexto de incertezas e num ambiente em constante mutação: para

reagir rápido e concorrentemente, deve-se ver as coisas a tempo e estar pronto no momento adequado e oportuno. Diante desse fato, para que se pudesse dar início aos procedimentos do tratamento de dados foi necessário consultar vários métodos utilizados na realização da técnica de Data Mining, o que serviu de suporte para a escolha dessa tecnologia. O método de árvore de decisão foi escolhido para realizar a tarefa de Data Mining, tendo como principal motivo o aprendizado eficiente e a produção de respostas qualitativas, seguras e essencialmente rápidas.

O Data Mining realizou duas tarefas importantes: a primeira está relacionada com a tarefa de transformação de dados, onde foram feitas as limpezas dados, incluindo a detecção dos campos totalmente em branco, ou seja, não possuíam nenhum valor e também dos campos que continham lixo ou valores que nada conseguiam expressar.

A segunda tarefa corresponde a extração de conhecimento das informações armazenadas no ambiente de Data Mart. Como já mencionado no capítulo V, com o escopo do problema razoavelmente complexo e o processo de consulta flexível, que envolve inúmeros parâmetros requerendo especificações muitas vezes estratégicas e delicadas, pode-se notar que a especificação da consulta não foi tão simples de se elaborar. Portanto, recorrer à classificação dos dados realizada através da ferramenta de Data Mining e também às entrevistas feitas com o especialista foram de fundamental importância para a especificação das consultas flexíveis.

A ferramenta selecionada para realizar a análise das informações dos bens patrimoniais de uma empresa de transporte, utilizou lógica fuzzy. Essa escolha está diretamente relacionada com o fato de que as informações/conhecimentos a serem obtidas se localizam dentro de um contexto que enfatizam uma enorme quantidade de imprecisão, refletindo na necessidade de se fazer consultas flexíveis.

As análises efetuadas compreendem apenas um indicador para compor o cenário da análise final que irá medir o escopo e a extensão do investimento em inteligência competitiva, possibilitando a tomada de decisão que irá gerar mudanças na estrutura da empresa.

É claro que não existe somente um caminho para o sucesso. Em vez de depender de apenas um diferencial ou impulso importante, a empresa precisa tecer sua

teia de estratégias. Lembrando que o benchmarking começa em casa, isto é, primeiro é preciso olhar para dentro da empresa para encontrar soluções.

É espantoso constatar que, quando o mundo da área de negócios parece em alguns momentos ter se aproximado da precisão característica das ciências exatas, essas começam a abandonar os métodos que, a partir do século 17, fizeram a sua glória. Hoje, é necessário se dar conta de que a quantificação, a previsibilidade, a precisão, apontam diretamente para o passado da ciência, não para seu futuro, nem sequer para a sua atualidade. Pois o que há de novo e revolucionário é a aplicação da imprecisão aos campos da física, da matemática e, por extensão, das engenharias. Portanto, a aplicabilidade da teoria de lógica fuzzy na área empresarial encerra um avanço significativo no conhecimento.

Permanecem, neste trabalho, questões que podem ser exploradas em estudos futuros. Para melhor aproveitamento da proposta apresentada, recomenda-se estudar o acoplamento dessa metodologia, ou seja, implementar um sistema onde se possa reunir as tecnologias de Data Mining, Data Mart e a ferramenta de análise flexível de forma integrada para que atendam às necessidades das empresas, no intuito de fornecer um amplo ambiente de oferta de informações para a tomada de decisões, obtendo ganhos de qualidade e produtividade.

Outra pesquisa sobre a imprecisão das informações poderia fazer parte do contexto, ou seja, uma sugestão para essa pesquisa é a aplicação de imprecisão nos dados de maneira explícita ou direta, através da atribuição de um valor de pertinência (numérico ou lingüístico) para cada tupla, ou aplicar a imprecisão no valor do dado de forma implícita, usando uma relação separada ou tabela de similaridade ou relacionamento de proximidade.

Ainda uma outra sugestão interessante seria aplicar o algoritmo de árvore fuzzy para realizar o tratamento de dados. Essa metodologia, como já mencionada, trabalha com conhecimento incompleto, porém não é considerada muito trivial.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] ADELMAN, S., "The Data Warehouse Database Explosion", http://www.sidadelman.com/, 1996.

[2] ADRIAANS, PIETRO, ZANTINGE, DOLF, *Data Mining*, Addison Wesley Longman, England, 1996.

[3] ADRIAANS, PIETER, KNOBBE, ARNO, HULST, MARC-PAUL, "Data Mining and Fuzzy Database", Syllogic, http://www.syllogic.nl/whipap1.htm, 1996.

[4] AGRAWAL, R., MANNILA, H., SRIKANT, R., TOIVONEM, H., VERKAMO, A. I., "Fast Discovery of Association Rules". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth, Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining*, pp 307-328, AAAI/MIT Press, 1996.

[5] ALBRECHT, J., GUNZEL, H., LEHNER W., "Na architecture for distributed OLAP" In: *PDPTA'98 International Conference*, vol. 3, 1482-1488, Las Vegas, Nevada, USA, 1998.

[6] BARTH, P., "Mining for profits in the Data Warehouse", In: *Barquin, R., Edelstein, H., Editor, Building, Using, and Managing The Data Warehouse*, chapter 8, pp. 137-149, Prentice Hall PTR, USA, 1997.

[7] BASSI, EDUARDO, "O marketing no novo milênio – o novo milênio já começou", *ESPM*, Vol. 6, ano 5, edição nº 3, maio/junho 1999.

[8] BERNDT, D. J., CLIFFORD, J., "Finding Patterns in Time Series: A Dynamic Programming Approach". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth, Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining*, pp 229-248. AAAI/MIT Press. 1996

[9] BERG, D. and HEAGELE, C., "Improving Data Quality: A Management Perspective and Model", In: *Barquin, R., Edelstein, H., Editor, Building, Using, and Managing The Data Warehouse*, chapter 5, pp. 85-99, Prentice Hall PTR, USA, 1997.

[10] BOAR, B., "Understanding Data Warehousing: A Strategic Perspective", In: *Barquin, R., Edelstein, H., Editor, Building, Using, and Managing The Data Warehouse*, chapter 13, pp. 277-299, Prentice Hall PTR, USA, 1997.

[11] BORDOGNA, G., PASI, G. AND LUCARELLA, D., "Extending a graph-based data model to manege fuzzy and uncertain information", In: *Fuzzy and Uncertain Object-Oriented Databases – Concepts and Models*, vol. 13, chapter 4, pp.97-122, University of Ghent, Belgium, 1997.

[12] BORDOGNA, G., PASI, G., "A Fuzzy Information Retrieval System Handling User's Preferences on Documention Sections". In: Dubois, Didier, Prade, Henri, Yager, Ronald, (eds), *Fuzzy Information Engineering - A Guide Tour of Applications*, chapter 17, pp.265-281, Wiley Computer Publishing, s/d.

[13] BOSC, P., PIVERT, O., "Imprecise Data Management and Flexible Querying in Databases". In: *Yager, Ronald, Zader, L.A.,(eds), Fuzzy Sets, Neural Networks and Software Computing*. Van Nostrand Reinhold, chapter 19, pp 368-395, NY, 1994.

[14] BOSC PATRICK, PIVERT, OLIVIER, "On Comparasion of Imprecise Values in Fuzzy Databases", *Fuzzy IEEE*, pp. 707-712, 1997.

[15] BOSC, PATRICK, PIVERT, OLIVIER, "Extending SQL Rrtrieval Features for the Handling of Flexible Queries". In: *Dubois, Didier, Prade, Henri, Yager, Ronald, editors, Fuzzy Information Engineering - A Guide Tour of Applications*, chapter 15, pp.233-251, Wiley Computer Publishing, s/d.

[16] BRACHMAN, RONALD J., "The Process of Knowledge Discovery: in Database". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth, Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining*, pp 37-57. AAAI/MIT Press, 1996.

[17] BRACKETT, M. H., *The Data Warehouse Challenge – Taming Data Chaos*, John Wiley & Sons, Inc., USA, 1996.

[18] BRIGGS, F.A., HWANG, K., *Computer architecture and parallel processing*, McGraw-Hill, Singapore, 1998.

[19] BUNTINE, WRAY, "Graphical Models for Discovery Knowledge". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth, Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining*, pp 59-82. AAAI/MIT Press, 1996.

[20] BURCH, G., "Building High Data Quality Into Your Data Warehouse", In: *Barquin, R., Edelstein, H., Editor, Building, Using, and Managing The Data Warehouse*, chapter 4, pp. 69-83, Prentice Hall PTR, USA, 1997.

[21] CATTELL, R.G.G., *Object Data Management: Object-Oriented and Extended relational Database Systems*, Addison-Wesley Publishing Company, USA, 1991.

[22] CALUWE, DE R., GYSEGHEM, VAN N. AND CROSS, V., "Basic notions and rationale of the integration of uncertainty management and object-oriented databases", In: *Fuzzy and Uncertain Object-Oriented Databases – Concepts and Models*, vol. 13, pp. 1-20, University of Ghent, Belgium, 1997.

[23] CHEESEMAN, PETER, STUTZ, JOHN, "Bayesian Classification (AutoClass): Theory and Results". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth,*

Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining, pp 153-180. AAAI/MIT Press, 1996.

[24] CHEN, MING-SYAN, HAN, JIAWEI, YU, PHILIP S., "Data Mining: An Overview from Database Perspective", s/l, s/ed, s/d.

[25] COX, EARL, *The Fuzzy Systems Handbook: a practitioner's guide to building, using, and maintaining fuzzy systems*, Academic Press, USA, 1994.

[26] COX, EARL, *Fuzzy logic – for business and industry*, s/ed, USA, s/d.

[27] Data Warehouse Overview, http://www.dwo.bull.com/dwtecha.htm, 1996.

[28] DATE, C. J., *Introdução a sistemas de banco de dados*, 4ª ed., Editora Campus, Rio de Janeiro, 1990.

[29] DBMS, Data Mining, http://www.dbmsmag.com/9807m05.html, 1999.

[30] DECKER, KARSTEN M., FOCARDI, SERGIA, "Technology Overview: A Report on Data Mining", Swiss Scientific Computing Center, 1995.

[31] DEMAREST, MARC, "Building The Data Mart", *DBMS Magazine*, vol.7, nº 8, pp44, 1994.

[32] DUNBAR, VERONICA, et al, *Oracle Data Mart Suit Cookbook*, Oralce Corporation, Release 1.0, 1997.

[33] DZEROSKI, SASO, "Inductive Logic Programming and Discovery Knowledge in Database". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth, Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining*, pp 117-152. AAAI/MIT Press, 1996.

[34] EBECKEN, N., MATTOSO, M., SOUZA, M., "Data mining on parallel database systems" *PDPTA'98 International Conference*, vol. 3, 1147-1154, Las Vegas, Nevada, USA, 1998.

[35] EDELSTEIN, HERB, "Technology How To: Mining Data Warehouses", Euclid Associates, http://techweb.cmp.com/iwk, 1996.

[36] ELDER IV, JOHN F., PREGIBON, DARYL, "A Statistical Perspective on Discovery Knowledge in Database". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth, Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining*, pp 82-113. AAAI/MIT Press, 1996.

[37] FAYYAD, USAMA M., PIATESKY-SHAPIRO, GREGORY, SMYTH, PADHRAIC, "From Data Minig to Knowledge Discovery: An Overview". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth, Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining*, pp 1-31. AAAI/MIT Press, 1996.

[38] FAYYAD, USAMA, UTHURUSAMY, R.AMASAMY, "Data Mining and Knowledge Discovery inDatabases", *Comunication of the ACM*, Vol. 39, N° 1, pp 24-26, 1996.

[39] FAYYAD, USAMA, PIATETSKY-SHAPIRO, SMYTH, PADHRAIC, "The KDD for Extracting from Volumes of Data", *Comunication of the ACM,* Vol. 39, N° 1, pp 27-34, 1996.

[40] FIELD, TOM, "Em busca de super-heróis", *HSM Management*, n° 14, pp 14-17, ano 3, maio/junho 1999.

[41] FID: Fuzzy Decision Tree, http://www2.cs.umsl.edu/~janikow/fid, 1998.

[42] FREEMAN, MILLER, "Data Mining Solutions Supplement", http://www.datamssmg.com, 1998.

[43] FU, YONGJIAN, *Discovery of Multiple-Level Rules from Large Databases.* Doctor of Philosophy, Simon Fraser University, 1996.

[44] Fuzzy Query Product Information Page, Sonalysts Inc., Siftware Development, http://www.sonalysts.com/fq.html, 1999.

[45] GAINES, BRAIN R., "Transforming Rules and Trees", In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth, Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining,* pp 205-226. AAAI/MIT Press, 1996.

[46] GALINDO, J., MEDINA, J., VILA, A., PONS, O., "Fuzzy comparators for flexible queries to databases", In: *International Conference on Artificial Intelligence,* Lisboa, Portugal, 1998.

[47] GALINDO, J., MEDINA, J., CUBERO,J., PONS, O., "A server for fuzzy SQL queries", In: *International Conference on Flexible Query Answery Systems, FQAS'98,* 1998.

[48] GANTI, N., "Managing and Staffing the Data Warehouse", In: *Barquin, R., Edelstein, H., Editor, Building, Using, and Managing The Data Warehouse,* chapter 11, pp. 213-262, Prentice Hall PTR, USA, 1997.

[49] GAZZOTTI, D., PIANCASTELLI, L., SARTORI, C., BENEVENTANO, D., "FuzzyBase: a Fuzzy Logic Aid for Relational Database Queries", In: *International Conference on Flexible Query Answering Systems, FQAS'98,* 1998.

[50] GEORGE, R., YAZICI, A., PETRY, F.E. AND BUCKELES, "Modeling impreciseness and uncertainty in the object-oriented data model – a similarity based approach", *Fuzzy and Uncertain Object-Oriented Databases – Concepts and Models*, vol. 13, chapter 3, pp. 63-96, University of Ghent, Belgium, 1997.

[51] GEORGE, R., SRIKANTH, R., BUCKLES, B.P., PETRY, F.E., "An Approach to Modelling Impreciseness and Uncertainty in the Object-Oriented Data Model". In: *Dubois, Didier, Prade, Henri, Yager, Ronald, editors, Fuzzy Information Engineering - A Guide Tour of Applications*, chapter 21, pp.325-346, Wiley Computer Publishing, s/d.

[52] GHOSHAL, SUMANTRA, "A empresa individualizada", *HSM Management*, nº 14, pp 20-24, ano 3, maio/junho 1999.

[53] GLASSEY-EDHOLM, K., "Bringing user perspective to Data Warehouses: Twenty-one points of consideration", In: *Barquin, R., Edelstein, H., Editor, Building, Using, and Managing The Data Warehouse*, chapter 6, pp. 103-120, Prentice Hall PTR, USA, 1997.

[54] GLYMOUR, CLARK, MANDIGAN, D., PREGIBON, D., SMYTH P., "Statistical Inference and Data Mining", *Comunication of the ACM*, Vol. 39, Nº 1, pp 35-41, 1996.

[55] GOIL, S., CHOUDHARY, A., "On scalable parallel computation of multidimensional data cube" *PDPTA'98 International Conference*, vol. 3, 1155-1162, Las Vegas, Nevada, USA, 1998.

[56] GOTTGTROY, MÁRCIA DE PAIVA BASTO, *O Processo de Aquisição do Conhecimento na Construção de Sistemas Especialistas*, Tese de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1990.

[57] GOTTGTROY, MÁRCIA DE PAIVA BASTO, *Aplicação de Técnicas da Engenharia do Conhecimento para a Avaliação das Condições de Segurança de Sistemas Estruturais*, Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Barsil, 1996.

[58] GUYON, ISABELLE, MATIÉ, NADA, VAPNIK, VLADIMIR, "Discovering Informative Patterns and Data Cleaning". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth, Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining*, pp 181-203. AAAI/MIT Press, 1996.

[59] GYSEGHEM, N. VAN, CALUWE, R., "The UFO database model: dealing with imperfect information", In: *Fuzzy and Uncertain Object-Oriented Databases – Concepts and Models*, vol. 13, chapter 1, pp. 123-186, University of Ghent, Belgium, 1997.

[60] HACKNEY, DOUGLAS, "Data Warehouse Delivery – What is a Data Mart", *Data Management Review Magazine*, 1997.

[61] HALE, JOHN, FINNERTY SCOTT, SHENOI SUJEET, "Analyzing Inference in Fuzzy Database Systems", 0-7803-1896-X/94 IEEE, 1994.

[62] HAMMER, K., "Migrating Data from Legacy Systems: Challenges and Solutions", In: *Barquin, R., Edelstein, H., Editor, Building, Using, and Managing The Data Warehouse*, chapter 2, pp. 27-40, Prentice Hall PTR, USA, 1997.

[63] HAN, JIAWEI, FU, YONGJIAN, "Attribute-Oriented Induction in Data Mining". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth, Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining*, pp 399-421. AAAI/MIT Press, 1996.

[64] HAN, JIAWEI, FU, YONGJIAN, "Advances of the DBLearn for Knowledge Discovery in Large Databases", School of Computing Science, Simon Fraser University, Burnaby, BC, Canada V5A 1S6, 1996.

[65] HANDY, CHARLES, "No reino das oportunidades", *HSM Management*, nº 14, pp 06-12, ano 3, maio/junho 1999.

[66] HAWKES, L. W., DERRY, S., KANDEL, A., "FUZZY EXPERT SYSTEMS FOR AN INTELLIGENT COMPUTER BASED TUTOR", s/l, s/d.

[67] HECKERMAN, DAVID, "Bayesian Networks for Knowledge Discovery". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth, Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining*, pp 273-305. AAAI/MIT Press, 1996.

[68] HILMETZ, S. A. and BRIDGE, S. R., "Gauging the returns on investments in competitive intelligence: a three-step analysis for executive decision makers", *Competitive Intelligence Review*, Vol. 10(1), pp 4-11, 1999.

[69] HSU, CHUN-NAN, KNOBLOCK, CRAIG A., "Using Inductive Learning to Generate Rules for Semantic Query Optimization". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth, Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining*, pp 425-445. AAAI/MIT Press, 1996.

[70] IBM, http://www.rs6000.ibm.com/sitemap.html

[71] IMIELINSKI, TOMASZ, MANNILA, H., *Comunication of the ACM*, Vol. 39, Nº 1, pp 58-64, 1996.

[72] INMON, W.H., "Excerpts from What is a Data Mart?", Data to Knowledge, http://www.d2k.com/library2.htm, 1999.

[73] INMON, W.H., *Como Construir Data Warehouse*, Editora Campus, Rio de Janeiro, 1997.

[74] ISOMOTO, YUKUO, YOSHINE, K., ISHII, N., NAKATANI, H., "Data Model and Fuzzy Information Retrival for Theoretical Extension of a Traditional Crisp Model to Fuzzy Model". In: *Dubois, Didier, Prade, Henri, Yager, Ronald, editors, Fuzzy Information Engineering - A Guide Tour of Applications*, chapter 18, pp.283-289, Wiley Computer Publishing, s/d.

[75] JONES, MARTYN RICHARD, "Information: It's architecture and management", Knowledge Asset Management and Corporate Memory - martyn@ibm.net., 1997

[76] KACPRZYK, JANUSZ, ZADROZNY, SLAWONIR, "Fuzzy Queries in Microsoft Access V.2". In: *Dubois, Didier, Prade, Henri, Yager, Ronald, (eds), Fuzzy Information Engineering, - A Guide Tour of Applications*, chapter 14, pp.223-232, Wiley Computer Publishing, s/d.

[77] KHOSHAFIAN, SETRAG, *Banco de Dados Orientado Objeto*, Editora Infobook S.A., Rio de Janeiro, 1994.

[78] KIMBALL, RALPH, "Mastering Data Extraction", DBMS online - Data Warehouse Architect - http://www.rkimball.com, 1996.

[79] KIMBALL, RALPH, *Data Warehouse Toolkit: Técnicas para Construção de Data Warehouses Dimensionais*, Makron Books do Brasil, São Paulo, 1998.

[80] KIMBALL, RALPH, "Dangerous Preconceptions", DBMS online - Data Warehouse Architect - http://www.rkimball.com., 1996.

[81] KLOSGEN, WILLI, "Explora: A Multipattern and Multistrategy Discovery Assistant". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth,*

Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining, pp 249-271. AAAI/MIT Press, 1996.

[82] Knowledge Discovery Nuggets, http://www.kdnuggets.com

[83] KOTLER, PHILIP, "Como criar, conquistar e dominar mercados", *HSM Management*, nº 14, pp 33-40, ano 3, maio/junho 1999.

[84] LAN, MING-SHONG, TAKENAGA, HIROSHI, ABE SHIGEO, "Character Recognition using Fuzzy Rules Extracted from Data", 0-7803-1896-X/94 IEEE, 1994.

[85] LARSEN, HENRIK LEGIND, YAGER, RONALD, "Query Fuzzification for Internet Information Retrival". In: *Dubois, Didier, Prade, Henri, Yager, Ronald, (eds), Fuzzy Information Engineering - A Guide Tour of Applications*, chapter 19, pp.291-310, Wiley Computer Publishing, s/d.

[86] LEITNER, PETER, "Além dos números", *HSM Management*, nº 14, pp 114-120, ano 3, maio/junho 1999.

[87] LIENERT, ANITA, "Semeando para um futuro incerto", *HSM Management*, nº 14, pp 26-30, ano 3, maio/junho 1999.

[88] LINDER, BENJAMIN, "Oracle Parallel RDBMS om Massively Parallel Systems", 0-8186-3330-1/93 IEEE, 1993.

[89] MARTINEZ, T. L., GARCIA, A. F., FONDON, D. M., GUITERREZ, A. D., LOVELLE, C. J. M., "A computational model for an object-oriented operating system" *PDPTA'98 International Conference*, vol. 1, 143-150, Las Vegas, Nevada, USA, 1998.

[90] MARTINEZ, T. L., GARCIA, A. F., FONDON, D. M., GUITERREZ, A. D., LOVELLE, C. J. M., "AGRA: the object distribution subsystem of the SO4 object-oriented operating system" *PDPTA'98 International Conference*, vol. 1, 255-259, Las Vegas, Nevada, USA, 1998.

[91] MATOSO, MARTA, 'Banco de Dados – Tópicos avançados", http://www.cos.ufrj.br/~marta, 1998.

[92] MELO, RUBENS NASCIMENTO, *Banco de Dados não Convencionais: a Tecnologia do BD e suas novas Áreas de Aplicação*, INUCAMP, IMECC, Campinas, SP, 1988.

[93] MENNINGER, D., "Building Object-Oriented OLAP Applications: not just any object-oriented toll will do", In: *Barquin, R., Edelstein, H., Editor, Building, Using, and Managing The Data Warehouse*, chapter 7, pp. 121-136, Prentice Hall PTR, USA, 1997.

[94] Oracle, http://www.oracle.com/plataformas/

[95] PEDRYEZ, WITOLD, *Fuzzy sets engineering*, CRC Press. Inc., USA, 1995.

[96] PETRY, F. E., *Fuzzy Databases: Principles and Applications*, Kluwer Academic Publishers, USA, 1996.

[97] PINHEIRO, C. A. R., "Events Specification in Active Database Management System". In: *Ebecken, Nelson F.F., (eds), Data Mining*, pp. 413-431. WIT Press, 1998.

[98] PRICE, ROBERTO TOM, GOLENDZINER, LIA GOLDSTEIN, *Banco de Dados para Aplicações não Convencionais*, Editora Kapelusz S.A., Argentina, 1989.

[99] RADEN, NEIL, "Data Everywhere", Information Week, narden@ix.netcom.com., 1995.

[100] RASMUSSEN, DAN, YAGER, RONALD, "A Fuzzy SQL Summary Language for Data Discovery". In: *Dubois, Didier, Prade, Henri, Yager, Ronald, (eds), Fuzzy Information Engineering - A Guide Tour of Applications*, chapter 16, pp.253-264, Wiley Computer Publishing, s/d.

[101] ROSSAZZA, JEAN-PAUL, "A hierarchical model of fuzzy classes", *Fuzzy and Uncertain Object-Oriented Databases – Concepts and Models*, vol. 13, chapter 2, pp. 21-61, University of Ghent, Belgium, 1997.

[102] RUMBAUGH, JAMES, et al., *Modelagem e projetos baseados em Objetos*, Editora Campus, Rio de Janeiro, RJ, 1997.

[103] SELEE, N. A., SEOG PARK, "Fuzzy object-oriented data model and fuzzy associaation algebra", *Fuzzy and Uncertain Object-Oriented Databases – Concepts and Models*, vol. 13, chapter 6, pp. 187-206, University of Ghent, Belgium, 1997.

[104] SETZER, VALDEMAR W., *Banco de Dados - Conceitos, Modelos, Gerenciadores, Projeto Lógico e Físico*, Editora Edgard Blucher Ltda., São Paulo, 1986.

[105] SHAH ARVIND D., MILSTEIN BARNEY M., "Data Warehousing: Pratical Tips for Successful Implementation", Managing The Data Warehouse Commitment, s/l, s/d.

[106] SHEN, WEI-MIN, ONG, KAYLIANG, MITBANDER, BHARAT, ZANIOLO, CARLO, "Metaqueries for Data Mining". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth, Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining*, pp 374-398. AAAI/MIT Press, 1996.

[107] SIDHU, MANJIT, "Data Warehouse Discussion - Data Warehouse vs Data Mart", s/l, 1998

[108] SILVA, E. B., XEXÉO, G. B., "Constrcting Data Mining Functionalities in a DBMS". In: *Ebecken, Nelson F.F., (eds), Data Mining*, pp. 367-381. WIT Press, 1998.

[109] SILVERSTON, L., INMON, W. H., GRAZIANO, K., *The Data Model Resource Book: A Library of Logical Data Models and Data Warehouse Designs*, John Wiley & Sons, Inc., USA, 1997.

[110] SIMOUDIS, EVANGELOS, LIVEZEY, BRAIN, KERBER, RANDY, "Integrating Inductive and Deductive Reasoning for Data Mining". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth, Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining*, pp 353-373. AAAI/MIT Press, 1996.

[111] SOUZA, M..S., MATOSO, M.L.Q., EBECKEN, "Data Mining: A Database Perspective". In: *Ebecken, Nelson F.F., (eds), Data Mining*, pp. 413-431. WIT Press, 1998.

[112] STALLINGS, WILLIAM, *Computer organization and architecture*, Mcmillan Inc., USA., 1993.

[113] SULAIMAN, A., SOUZA, J., "A Decision Support System that Reverse Engineers Abstract Database Transactions – The Conceptual Model". In: *Ebecken, Nelson F.F., (eds), Data Mining*, pp. 401-411. WIT Press, 1998.

[114] SWEIGER, M., "Optimal Machine Architectures for Parallel Query Scability", In: *Barquin, R., Edelstein, H., (eds), Building, Using, and Managing The Data Warehouse*, chapter 3, pp. 41-68, Prentice Hall PTR, USA, 1997.

[115] THIRY-CHERQUES, H.R., "O caos e a confusão", *ESPM*, Vol. 6, ano 5, edição nº 3, maio/junho 1999.

[116] VANDENBERGHE, R., CALUWE, R. DE, "Fuzzy Database Techniques: An Application to Criminal Investigation". In: *Dubois, Didier, Prade, Henri, Yager, Ronald, editors, Fuzzy Information Engineering - A Guide Tour of Applications*, chapter 20, pp.311-323, Wiley Computer Publishing, s/d.

[117] YOURDON, EDWARD, *Análise Estruturada Moderna*, Editora Campus, Rio de Janeiro, RJ, 1992.

[118] WATTERSON, KAREN, "Attention, Data-Mart Shoppers", www.byte.com, 1997

[119] WELCH, J. D., "Updating Data Warehouse", In: *Barquin, R., Edelstein, H., (eds), Building, Using, and Managing The Data Warehouse*, chapter 10, pp. 173-210, Prentice Hall PTR, USA, 1997.

[120] WILLIAMS, JOSEPH, "Sagent Data Mart Solution", DBMS online - Sagent Technology Inc. - http://www.sagenttech..com., 1997.

[121] XpertRule Miner, Knowledge from Data, Attar Software Limited, Release 1.1, http://www.attar.com, 1999.

[122] ZAGELOW, G., "Data Warehousing – Client/Server for the Rest of the Decade", In: *Barquin, R., Edelstein, H., Editor, Building, Using, and Managing The Data Warehouse*, chapter 1, pp. 3-23, Prentice Hall PTR, USA, 1997.

[123] ZANASI, ALESSANDRO, "Competitive Intelligence through Data Mining public sources", *Competitive Intelligence Review*, Vol. 9(1), pp 44-54, 1998.

[124] ZEMBOWICZ, ROBERT, ZYTKOW, JAN M., "From Contingency Tables to Various Forms of Knowledge". In: *Fayyad, Usama M., Piatesky-Shapiro, Gregory, Smyth, Padhraic, Uthurusamy, Ramasamy, (eds), Advances in Knowledge Discovery and Data Mining*, pp 329-349. AAAI/MIT Press, 1996.