



Universidade Estadual de Campinas  
Faculdade de Engenharia Elétrica e de Computação  
Departamento de Engenharia de Computação e Automação Industrial

## Ferramentas para o gerenciamento de informação nebulosa em bancos de dados.

**Marta Ines Velazco Fontova**

Orientador: Prof. Dr. Paulo César Bezerra  
(UNICAMP/FEEC/DCA)

Este exemplar corresponde a redação final da tese  
defendida por MARTA INES VELAZCO FON-  
TOVA e aprovada pela Comissão  
Julgada em 03/03/1998  
*Paulo Cesar Bezerra*  
Orientador

Dissertação apresentada à Faculdade  
de Engenharia Elétrica e de  
Computação da Universidade  
Estadual de Campinas como parte  
dos requisitos exigidos para a  
obtenção do título de Mestre em  
Engenharia Elétrica

Campinas, 3 Fevereiro de 1998.

V541f

33539/BC



UNIDADE	BC
N.º ORÇAMADA:	Unicamp
V.	591f
TOTAL	33539
PREÇO	395/98
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00
DATA	17/04/98
N.º OPD	

CM-00109740-5

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

V541f

Velazco Fontova, Marta Ines

Ferramentas para o gerenciamento de informação  
nebulosa em banco de dados. / Marta Ines Velazco  
Fontova .--Campinas, SP: [s.n.], 1998.

Orientador: Paulo César Bezerra

Dissertação (mestrado) - Universidade Estadual de  
Campinas, Faculdade de Engenharia Elétrica e de  
Computação.

1. Banco de dados relacionais. 2. Conjuntos difusos  
3. SQL (Linguagem de programação de computador). I.  
Bezerra, Paulo César. II. Universidade Estadual de  
Campinas. Faculdade de Engenharia Elétrica e de  
Computação. III. Título.

*A mis padres*

# *Agradecimentos*

---

*A Paulo César Bezerra pela ajuda durante estes anos,*

*A Luisma, meu companheiro, amigo e professor,*

*Á CAPES pelo suporte financeiro do projeto,*

*A meus amigos da comunidade cubana que representam minha Cuba do Brasil,*

*A meus amigos brasileiros, Anderson, Pedrito, Ronaldo e Ana.*

# *Resumo*

---

Esta dissertação, apresenta o uso da lógica nebulosa para a extensão do modelo relacional de bancos de dados, com o objetivo de permitir a representação e manipulação de informação incompleta e imprecisa, não permitida nos modelos de bancos de dados convencionais.

Também é apresentado um estudo das diferentes abordagens em bancos de dados nebulosos, é proposto um novo modelo que integra tais abordagens e adiciona novos recursos de manipulação de dados ao gerenciador. Completando o conjunto, é apresentada uma arquitetura com o objetivo de implementar o modelo proposto usando sistemas gerenciadores de bancos de dados convencionais existente no mercado. No capítulo 5, o modelo é implementado usando o Borland Delphi 2.0 como gerenciador de suporte.

Vale realçar que, como todas as funções lógicas são armazenadas em relações, os operadores lógicos podem ser redefinidos no instante de uma sessão de consultas. Sendo assim, uma ferramenta importante para aqueles que experimentam lógicas diversas usando o intervalo  $[0,1]$ .

# *Abstract*

---

This dissertation concerns about the use of fuzzy logic for extending the relational database model in order to represent and manipulate incomplete and inaccurate information, which is impossible in conventional models.

It's also presents in this work an study of the different approaches of fuzzy database, a model that integrates them and adds new recourses for data manipulation. An architecture of the model is defined to permit a Database Management System (DBMS) in the market, to be the host for the implementation. In chapter 5, this new model is implemented on Borland Delphi 2.0.

It's important to emphasize that, as all logical functions are stored in relations, the logical operators can be redefined in a query section. Obviously, they are important tools for the people who experiment non-conventional logic in  $[0,1]$  interval.

# Sumário

---

<b>Capítulo 1</b> Introdução.	1
<b>Capítulo 2</b> Princípios de Bancos de Dados e a Lógica Nebulosa.	5
2.1 Modelo Relacional de Bancos de Dados.	5
2.1.1 Modelo Relacional.	6
2.1.2 Domínio, Atributo, Tupla e Relação.	6
2.1.3 Manipulação dos dados. A Álgebra Relacional e O Calculo Relacional.	9
2.1.3.1 A Álgebra relacional. Operações.	9
2.1.3.2 O Calculo Relacional.	15
2.1.4 Tratamento da informação incompleta em bancos de dados relacionais.	16
2.1.4.1 Abordagens existentes para o tratamento do valor nulo.	16
2.2 Conjuntos Nebulosos.	17
2.2.1 Conceito de Conjunto Nebuloso.	18
2.2.2 Operações em Conjuntos Nebulosos.	19
2.2.3 Princípio de extensão.	21
2.2.4 Números nebulosos.	22
2.2.5 Variáveis lingüísticas.	23
2.2.6 Teoria de possibilidade.	24
2.2.6.1 Distribuição de possibilidade. Definição.	25
2.2.6.2 Medidas de Possibilidade e Necessidade.	25
2.2.7 Similaridade.	26
<b>Capítulo 3</b> Modelos de Bancos de Dados Nebulosos.	28
3.1 Apresentação do problema.	28
3.2 Modelo de Unificação a partir de relações de similaridade.	30
3.2.1 Álgebra relacional nebulosa.	35
3.2.2 Análise Crítica.	41
3.3 Modelo Relacional Nebuloso.	42
3.3.1 Análise Crítica.	43
3.4 Modelo Relacional Possibilístico.	43
3.4.1 Modelo de Prade-Testemale.	44
3.4.2 Modelo de Zemankova-Kandel.	49
3.4.3 Modelo de Umano.	50
3.4.4 Análise Crítica.	54
3.5 Comparação entre as abordagens.	54

<b>Capítulo 4</b> O Modelo Integrado de Bancos de Dados Nebulosos.	56
4.1 GEFRED. Modelo generalizado de bancos de dados nebulosos.	56
4.1.1 Domínio Nebuloso Generalizado.	57
4.1.2 Relação Nebulosa Generalizada.	57
4.1.3 Álgebra Nebulosa Generalizada.	59
4.1.4 Estudo do GEFRED comparado com outras abordagens.	61
4.1.4.1 GEFRED e o Modelo Relacional Nebuloso.	61
4.1.4.2 GEFRED e o Modelo de unificação a partir de relações de similaridade.	61
4.1.4.3 GEFRED e o Modelo Relacional Possibilístico.	62
4.1.4.3.1 GEFRED e o Modelo de Prade-Testemale.	62
4.1.4.3.2 GEFRED e o Modelo de Zemankova-Kandel.	63
4.1.4.3.3 GEFRED e o Modelo de Umano.	63
4.2 O Modelo Integrado de Bancos de Dados Nebulosos.	64
4.2.1 Domínio Integrado Nebuloso.	64
4.2.2 Relação Integrada Nebulosa.	66
4.2.3 Atributos com valores definidos como conceitos complexos.	68
4.2.4 Álgebra Integrada Nebulosa.	69
4.2.5 Proposta de SQL-Nebuloso.	73
4.2.5.1 Definição sintática da linguagem.	74
<b>Capítulo 5</b> Aspectos da Implementação do Modelo Proposto.	77
5.1 Arquitetura do SGBD Nebuloso.	77
5.1.1 Extensão da Linguagem de Definição de dados.	79
5.1.1.1 Dados nebulosos sobre Domínios Numéricos.	81
5.1.1.2 Dados Nebulosos sobre Domínios de Escalares.	86
5.1.1.3 Dados Nebulosos sobre Domínios de Escalares com Proximidade.	88
5.1.1.4 Dados Nebulosos sobre Domínios de conceitos complexos.	89
5.1.2 Extensão da linguagem de Consultas.	90
5.2 Implementação em Delphi.	91
5.2.1 Implementação dos tipos de atributos.	94
5.2.2 Implementação do meta-conhecimento.	96
<b>Capítulo 6</b> Exemplo de aplicação.	99
6.1 Definição das relações para solução do problema.	100
6.2 Definição dos domínios dos atributos.	101
6.3 Preenchimento dos valores das tabelas.	106
6.4 Armazenamento do meta-conhecimento.	109
6.5 Representação física dos atributos nebulosos nas tabelas.	113
6.6 Exemplo de consulta.	104
<b>Capítulo 7</b> Conclusões.	117
<b>Bibliografia.</b>	119

# Lista de Figuras

---

Figura 2.1. Aplicação do Princípio de Extensão.	22
Figura 2.2. Função Convexa.	23
Figura 2.3. Função Não Convexa.	23
Figura 2.4. Árvore de partições para a relação $s$ .	24
Figura 3.1. Definição dos conjuntos nebulosos definidos sobre o atributo <i>Altura</i> .	35
Figura 3.2. Definição dos conjuntos nebulosos definidos sobre o atributo <i>Peso</i> .	35
Figura 3.3. Aplicação da fórmula 1 no conjunto <i>baixo</i> .	40
Figura 3.4. Aplicação da fórmula 1 no conjunto <i>mediano</i> .	40
Figura 3.5. Aplicação da fórmula 1 no conjunto <i>alto</i> .	40
Figura 5.1. Modelo de Implementação.	78
Figura 5.2. Tipos de dados do SGBD Nebuloso.	79
Figura 5.3. Função Triangular.	82
Figura 5.4. Distribuição Intervalar.	82
Figura 5.5. Componentes do Interpretador de Consultas nebulosas.	91
Figura 5.6. Hierarquia do Delphi para bancos de dados.	91
Figura 5.7. Extensão da hierarquia para o tratamento de Relações Nebulosas.	93
Figura 5.8. Hierarquia para a extensão dos tipos de atributos nebulosos.	95
Figura 5.9. Hierarquia dos valores nebulosos.	96
Figura 5.10. Hierarquia das tabelas do meta-conhecimento.	97
Figura 6.1. Valores Lingüísticos do domínio <i>Idade</i> .	102
Figura 6.2. Valores Lingüísticos do domínio <i>Quantidades</i> .	104
Figura 6.3. Valores Lingüísticos do domínio <i>Profundidade</i> .	105
Figura 6.4. Valores Lingüísticos do domínio <i>Área</i> da relação <b>Quartos</b> .	106
Figura 6.5. Cálculo do $c_{area_1} = 0,77$ .	115
Figura 6.6. Cálculo do $c_{area_3} = 1$ .	116

# Lista de Tabelas

---

Tabela 2.1. Relação <i>Estudante_Mestrado</i>	8
Tabela 2.2. Relação <i>Estudante_Doutorado</i> .	8
Tabela 2.3. Relação <i>Estudante_Mestrado</i> $\cup$ <i>Estudante_Doutorado</i> .	10
Tabela 2.4. Relação <i>Estudante_Mestrado_Idade</i> $\geq 20$	13
Tabela 2.5. Projeção sobre <i>Estudante_Mestrado</i> .	14
Tabela 2.6. Relação Universidades.	15
Tabela 2.7. Relação resultante da junção	15
Tabela 2.8. Relação de similaridade, <i>s</i> .	27
Tabela 3.1. Relação <i>Físico</i> .	32
Tabela 3.2. Relação de similaridade sobre o domínio <i>Cor do Cabelo</i> .	34
Tabela 3.3. Relação de similaridade sobre o domínio <i>Cor do Cabelo</i> .	34
Tabela 3.4. Relação Resposta	39
Tabela 3.5. Pertinência da tupla à relação resposta	39
Tabela 3.6. Relação Resposta	40
Tabela 3.7. Pertinência da tupla à Relação Resposta	40
Tabela 3.8. Representação da relação físico no Modelo Relacional Nebuloso.	43
Tabela 3.9. Representação da relação físico no Modelo de Prade-Testemale.	46
Tabela 3.10. Representação da relação <i>físico</i> no Modelo de Umano.	51
Tabela 3.11. Relação resposta à consulta.	54
Tabela 3.12. Comparação entre as abordagens.	55
Tabela 4.1. Tipos de dados representados pelo modelo.	58
Tabela 4.2. Nova representação proposta.	65
Tabela 5.1. Tabela das relações e atributos nebulosos.	80
Tabela 5.2. Relação dos domínios nebulosos.	81
Tabela 5.3. Representação física de um atributo de tipo numérico.	84
Tabela 5.4. Relação das funções.	85
Tabela 5.5. Relação das variáveis lingüísticas.	85
Tabela 5.6. Representação física do tipo escalar.	87
Tabela 5.7. Relações de proximidade.	88
Tabela 5.8. Relação dos pares AND-OR.	89
Tabela 5.9. Relação que define os conceitos complexos.	90
Tabela 6.1. Definição dos conceitos complexos: <i>Péssimo, Regular, Boa</i> .	102
Tabela 6.2. Relação de proximidade entre os valores do domínio <i>Estado</i> .	103

Tabela 6.3. Relação <b>Imóvel</b> .	107
Tabela 6.4. Relação <b>Proprietário</b> .	108
Tabela 6.5. Relação <b>Acabamento</b> .	108
Tabela 6.6. Relação <b>Quartos</b> .	108
Tabela 6.7. Relação <b>Piscina</b> .	108
Tabela 6.8. Tabela para definição dos atributos nebulosos.	109
Tabela 6.9. Tabela para a definição dos domínios nebulosos.	110
Tabela 6.10. Tabela para a definição das funções.	110
Tabela 6.11. Tabela para a definição dos rótulos em domínios numéricos.	111
Tabela 6.12. Tabela para a definição das relações de similaridade.	112
Tabela 6.13. Tabela para a definição dos conceitos complexos.	112
Tabela 6.14. Representação física dos atributos nebulosos da relação <b>Imóvel</b> .	113
Tabela 6.15. Representação física dos atributos nebulosos da relação <b>Acabamento</b> .	114
Tabela 6.16. Representação física dos atributos nebulosos da relação <b>Quartos</b> .	114
Tabela 6.17. Representação física dos atributos nebulosos da relação <b>Piscina</b> .	114
Tabela 6.18. Relação Resposta.	116

# *Capítulo 1*

## *Introdução*

---

O advento do papel, em torno do ano 100 da era cristã, é, sem dúvida, um marco importante para o arquivo e consulta de informações. É interessante notar que, naquela época, a teoria dos números engatinhava, fazendo com que os dados numéricos fossem de pouca expressão.

Somente com a sedimentação da notação posicional e o aparecimento do zero no século XIV é que aconteceu a popularização da aritmética, incluindo as frações decimais, isto em torno do século XV. Obviamente, a popularização da aritmética deu um impulso ao comércio marítimo da época, o que nos induz a pensar que os livros de contabilidade dos negócios podem ter sido os primeiros bancos de dados da “época numérica”.

A análise de outras invenções e desenvolvimentos que contribuíram para a memorização e consultas de informações, numéricas e não numéricas, poderia se estender por muito mais. Entretanto, a contribuição do computador digital é a que mais importa neste trabalho.

Durante os anos 60 do século XX, os computadores digitais já permitiam a implementação de bancos de dados de grande porte que, na sua maioria, mantinham fidelidade as estruturas hierárquicas, típicas de organizações comerciais.

Em 1970 Codd [Co70] apresentou o modelo relacional que tem como principal característica a representação de relações através de tabelas, onde as linhas são os

“*records*” e as colunas os atributos. Além disso, o modelo permite operações lógicas entre tabelas.

Muitos modelos foram apresentados até hoje estendendo o modelo relacional. Todos eles modelam a informação a partir de termos precisos e com isto alguns problemas começam a aparecer. O primeiro problema, é o significado da informação não existente, a não definida e a incompleta. Muitas abordagens são apresentadas para resolver este problema [Ga77, Lip79].

Em 1965, Zadeh [Za65] propôs a teoria sobre a lógica nebulosa, expandindo a teoria dos conjuntos binários. Especificamente, estende o conceito de pertinência ou não (1 ou 0) de um elemento a um conjunto ao grau de pertinência de 0 a 1. Com esta extensão, foi possível modelar a informação imprecisa e termos como *alto* e *poucos*, ficaram facilmente representáveis.

Com o desenvolvimento da lógica nebulosa foram também aparecendo as primeiras abordagens sobre bancos de dados nebulosos, definidos a partir de bancos de dados relacionais, onde as relações são estendidas a relações nebulosas, e é possível a representação da imprecisão e da incerteza.

Ao longo do tempo, os modelos de bancos de dados nebulosos foram diferenciados, em três abordagens: Modelo Relacional Nebuloso, Modelo de Unificação a partir de Relações de Similaridade, Modelo Relacional Possibilístico. Cada um destes enfoques tratando a incerteza de uma maneira diferente.

Depois que estes modelos foram definidos começaram a aparecer algumas versões modificadas. Por exemplo, Shenoit [SM89] estendeu as relações de similaridade a relações de proximidade. Juntamente, Liu [Li90] apresentou uma aplicação sobre Prolog onde utilizava a abordagem de Umano [Um83]. Outro trabalho importante nesta área, é a tese de Doutorado de Medina [Me94], onde ele mostra a implementação de um modelo de banco de dados nebulosos sobre ORACLE.

A análise comparativa dos modelos Relacional Nebuloso, de Unificação a partir de Relações de Similaridade e Relacional Possibilístico foi o que gerou a motivação

deste trabalho de tese de mestrado. O resultado da análise, mostrou que havia um espaço a ser coberto por um novo modelo, que não era atendido por nenhum daqueles outros. Assim, elegeram-se as seguintes diretrizes, na definição de um novo modelo:

- Incluir em uma única representação a possibilidade de definir os aspectos mais relevantes das abordagens anteriores,
- Definir uma álgebra para a manipulação desta informação.

Com base do modelo proposto é desenvolvida uma especificação para o desenvolvimento e implementação de um sistema gerenciador de bancos de dados. Por outro lado, esta especificação deve também permitir que sejam usados gerenciadores de Bancos de Dados como suporte. Em decorrência da especificação, foi possível implementar um conjunto de hierarquias de classes do Delphi 2.0, estendendo-o ao tratamento de dados nebulosos.

Formado por sete capítulos este trabalho apresenta no capítulo 2 os conceitos básicos do modelo relacional de bancos de dados e as noções fundamentais sobre a teoria de conjuntos nebulosos. Tais conceitos, servem como base para a definição dos modelos de bancos de dados nebulosos.

O terceiro capítulo, apresenta as três abordagens principais sobre bancos de dados nebulosos, onde cada enfoque é analisado a partir de um exemplo comum. Finalmente, apresenta uma comparação entre as abordagens.

No capítulo 4 é proposto um modelo de banco de dados nebulosos que é uma extensão do modelo de Medina [Me94a].

O capítulo 5, começa apresentando de forma geral, como deve ser implementado o modelo do capítulo 4 e termina com a hierarquia de classes que constituem as ferramentas para o desenvolvimento de aplicações modeladas a partir de um banco de dados nebulosos.

O poder de representação do modelo proposto é apresentado no capítulo 6, através de um exemplo de aplicação. Adicionalmente neste capítulo, são mostradas as tabelas utilizadas na implementação, que formam o banco de meta-conhecimento.

O último capítulo trata as conclusões gerais deste trabalho e as perspectivas com relação a futuras extensões.

# Capítulo 2

## *Princípios de Bancos de Dados e a Lógica Nebulosa*

---

Os conceitos básicos que constituem o ponto de partida desta dissertação são apresentados neste capítulo. Inicialmente são abordados os conceitos básicos do modelo relacional de bancos de dados e em seguida as noções fundamentais sobre a teoria de conjuntos nebulosos

### **2.1 Modelo Relacional de Bancos de Dados.**

A tecnologia de bancos de dados contribui fortemente para o uso crescente de computadores que, de modo inverso, possibilita o seu desenvolvimento. É inquestionável que os bancos de dados desempenham um importante papel na maior parte das áreas onde os computadores são usados, incluindo negócios, engenharia, medicina, leis, educação, entre outras. Com o objetivo de criar um vocabulário claro ao longo de todo este trabalho, a seção seguinte apresenta um subconjunto de definições e enunciados de conceitos, relativos a bancos de dados relacionais.

Define-se **Banco de Dados** [EN94] como uma coleção de dados relacionados que representam aspectos do mundo real. São projetados, construídos e povoados com dados para um propósito específico. Podem ser de qualquer tamanho e de complexidade variável.

Um grande número de trabalhos sobre bancos de dados relacionais foram publicados [UI82, EN94] desde que Codd [Co70] apresentou o primeiro trabalho, em 1970, onde definia o Modelo Relacional. Posteriormente foram aparecendo diversas extensões e modificações do mesmo [Ga77, Lip79, Pe96, Me94, Po94, Li90].

### 2.1.1 Modelo Relacional.

O modelo relacional foi enunciado por Codd e apresenta os bancos de dados como relações. Cada relação é representada como uma tabela onde cada linha representa uma coleção de valores relacionados. Estes valores podem ser interpretados como fatos que descrevem o mundo real. O nome da tabela e das colunas é usado para ajudar na interpretação do significado de cada linha na tabela. Todos os valores na mesma coluna são do mesmo tipo de dado.

No modelo relacional as linhas são chamadas de **tuplas**, o cabeçalho da coluna é chamado de **atributo** e uma tabela é chamada de **relação**. Um tipo de dado que descreve os tipos de valores que podem aparecer em cada coluna é chamado de **domínio**.

Para ilustrar os conceitos será utilizado um exemplo, assim enunciado: suponha a secretaria de pós-graduação de uma faculdade qualquer. Para armazenar os dados dos estudantes que estão matriculados no cursos de mestrado ou doutorado optou-se por um banco de dados. Para cada estudante são armazenados os seguintes dados: nome completo, idade, endereço, telefone, registro acadêmico (RA), RG, nacionalidade, área e código da universidade da graduação.

### 2.1.2 Domínio, Atributo, Tupla e Relação.

Um domínio  $D$  define-se como, um conjunto de valores atômicos para os quais é especificado o tipo de dado e o formato. Entende-se como atômico que cada valor no domínio é indivisível.

Um **esquema de relação**  $R$  é usado para descrever uma relação. Denotado por

$$R(A_1, A_2, \dots, A_n),$$

sendo formado por um nome de relação  $R$  e um conjunto de **atributos**. Cada atributo  $A_i$  toma os valores no domínio  $D_i$ , denotado por  $dom(A_i)$ . O **grau da relação** é o número de atributos que formam o esquema de relação.

No exemplo, podemos definir dois esquemas de relação de grau 9, como seguem:

*Estudante\_Mestrado*(Nome, Idade, Endereço, fone, Registro Acadêmico (RA), RG, Nacional., Área, Graduação),

*Estudante\_Doutorado*(Nome, Idade, Endereço, fone, Registro Acadêmico (RA), RG, Nacional., Área, Graduação).

Cada esquema apresentado define uma relação de 9 atributos. Podem ser especificados os seguintes domínios para alguns dos atributos definidos,  $dom(Nome)=Nomes$ ;  $dom(Idade)=Idades$ ;  $dom(fone)=Telefones_Locais$ ;  $dom(Área)=Área_facultade$ .

A partir das definições anteriores uma **relação**  $r$  do esquema  $R(A_1, A_2, \dots, A_n)$ , denotada por  $r(R)$ , pode ser definida como um subconjunto do produto cartesiano dos domínios de  $R$ :

$$r(R) \subseteq (dom(A_1) \times dom(A_2), \dots, dom(A_n))$$

Com isto,  $r$  define um subconjunto de **n-tuplas**  $r = \{t_1, t_2, \dots, t_n\}$ . Cada **n-tupla**  $t$  é uma lista ordenada de  $n$  valores  $t = \langle v_1, v_2, \dots, v_n \rangle$ , onde cada valor  $v_i$ ,  $1 \leq i \leq n$ , é um elemento do  $dom(A_i)$  ou o valor especial **nulo**. De acordo com esta definição, uma **tupla** pode ser considerada como um conjunto de pares  $(\langle atributo \rangle, \langle valor \rangle)$ , onde cada par define um mapeamento de um atributo  $A_i$  ao valor  $v_i$  do  $dom(A_i)$ .

As tabelas 2.1 e 2.2 representam as relações *Estudante\_Mestrado* e *Estudante\_Doutorado*, que correspondem a os esquemas que foram apresentados. Cada tupla nas tabelas, mostrada como linha, representa uma entidade de estudante de mestrado ou estudante de doutorado respectivamente. Os atributos representados com o cabeçalho de cada coluna indicam a interpretação dos valores.

Nome	Idade	Endereço	Fone	RA	RG	Nacional.	Área	Graduação
Vladimir Velazco	26	Sta. Cruz 315, Cambuí	Nulo	956698	V167555-Q	Cuba	Automação	01
Ronaldo Ruiz	23	Ave. Anchieta 245, Centro.	2357589	967690	H123745-O	Brasil	Telemática	02
José Gómez	19	Ave. Brasil, 222, Guanabara.	2444687	973424	V171543-U	Peru	Controle	03

Tabela 2.1. Relação *Estudante\_Mestrado*.

Nome	Idade	Endereço	Fone	RA	RG	Nacional.	Área	Graduação
Luis Del Val	34	Sta. Cruz 315, Cambuí	Nulo	956450	V167345-Q	Cuba	Automação	01
Lino Rosell	27	Ave. Anchieta 345, Centro.	2395646	953466	H162362-O	Brasil	Telemática	02
Luiz Diaz	19	Ave. Brasil, 234, Guanabara.	2434587	963424	V172543-U	Cuba	Controle	04

Tabela 2.2. Relação *Estudante\_Doutorado*.

Como tinha-se definido que uma relação é um conjunto de tuplas e, por definição, todos os elementos de um conjunto são distintos, então todas as tuplas de uma relação devem ser diferentes. Assim sendo, é definida como **chave primária** de uma relação o atributo ou conjunto de atributos que identificam unicamente uma tupla na relação. Na relação *Estudante\_Mestrado*, pode-se tomar como chave primária qualquer dos atributos *Registro Acadêmico* ou *RG*, para o exemplo se considerará o atributo *Registro Acadêmico*.

### 2.1.3 Manipulação dos dados. A Álgebra Relacional e O Calculo Relacional.

Outra parte importante no modelo relacional é a coleção de operadores de que dispõe o usuário para consultar os dados armazenados, que geram novas relações a partir das antigas. Decorre então a definição de duas linguagens formais equivalentes. **Álgebra Relacional e O Calculo Relacional.**

A **álgebra relacional** é um conjunto de operadores que são usados para manipular as relações, especificando *o como* deve ser recuperada a informação. O **calculo relacional** especifica *o quê* deve ser recuperado. Ambas linguagens de certa forma são equivalentes, qualquer recuperação feita com a álgebra relacional pode também ser feita com o calculo relacional e vice versa. Isto é, o poder de expressão de ambas as duas é idêntico.

#### 2.1.3.1 A Álgebra relacional. Operações.

O conjunto de operadores que define a álgebra relacional esta dividido em dois grupos. Um grupo inclui as operações básicas da teoria matemática de conjuntos, as quais são aplicáveis porque cada relação é definida como um conjunto de tuplas. Estas operações são: **União, Interseção, Diferença, e Produto Cartesiano.** O outro grupo consiste em operações específicas sobre relações: **Seleção, Projeção e Junção.**

As operações de união, interseção e diferença não podem ser aplicadas sobre quaisquer relações. Restrições são introduzidas através do conceito de compatibilidade pela união. Duas relações  $R(A_1, A_2, \dots, A_n)$  e  $S(B_1, B_2, \dots, B_n)$ ,  $R$  e  $S$  são compatíveis pela união se tem o mesmo grau  $n$ , e  $dom(A_i) = dom(B_i)$  para  $1 \leq i \leq n$ . Isto é, as relações  $R$  e  $S$  tem o mesmo número de atributos e cada par dos correspondentes atributos pertencem ao mesmo domínio.

As duas relações definidas na seção anterior *Estudante\_Mestrado* e *Estudante\_Doutorado* são compatíveis pela união, pois cumprem com a condição anterior. Assim, serão utilizadas para exemplificar as operações.

Operador União (**Union**).

A união de duas relações  $R$  e  $S$ , denotado por

$$R \cup S,$$

é a relação que inclui todas as tuplas que estão na relação  $R$  ou na relação  $S$ . As tuplas duplicadas são eliminadas.

A união das relações *Estudante\_Mestrado* e *Estudante\_Doutorado*,

$$\textit{Estudante\_Mestrado} \cup \textit{Estudante\_Doutorado}$$

é a relação formada pelas tuplas que pertencem à relação *Estudante\_Mestrado* ou a relação *Estudante\_Doutorado*, como mostra a tabela 2.3.

Nome	Idade	Endereço	Fone	RA	RG	Nacional.	Área	Graduação
Vladimir Velazco	26	Sta. Cruz 315, Cambuí	Nulo	956698	V167555-Q	Cuba	Automação	01
Ronaldo Ruiz	23	Ave. Anchieta 245, Centro.	2357589	967690	H123745-O	Brasil	Telematica	02
José Gómez	19	Ave. Brasil, 222, Guanabara.	2444687	973424	V171543-U	Peru	Controle	03
Luis Del Val	34	Sta. Cruz 315, Cambuí	nulo	956450	V167345-Q	Cuba	Automação	01
Lino Rosell	27	Ave. Anchieta 345, Centro.	2395646	953466	H162362-O	Brasil	Telematica	02
Liuz Diaz	19	Ave. Brasil, 234, Guanabara.	2434587	963424	V172543-U	Cuba	Controle	04

Tabela 2.3. Relação *Estudante\_Merstrado*  $\cup$  *Estudante\_Doutorado*.

Operador Interseção (**Intersection**).

A interseção de duas relações  $R$  e  $S$ , denotada por

$$R \cap S,$$

é a relação que inclui as tuplas que estão na relação  $R$  e na relação  $S$ .

A relação formada pela interseção das relações *Estudante\_Mestrado* e *Estudante\_Doutorado*,

$$\textit{Estudante\_Mestrado} \cap \textit{Estudante\_Doutorado}$$

é a relação formada pelas tuplas que pertencem à relação *Estudante\_Mestrado* e a relação *Estudante\_Doutorado*. Neste exemplo não existe nenhuma tupla que pertença as duas relações,

$$\textit{Estudante\_Mestrado} \cap \textit{Estudante\_Doutorado} = \phi,$$

portanto a relação obtida é vazia.

Operador Diferença (**Difference**).

A diferença de duas relações  $R$  e  $S$ , denotado por

$$R - S,$$

é a relação que inclui todas as tuplas que estão na relação  $R$  mas não estão incluídas na relação  $S$ .

A relação formada pela diferença das relações *Estudante\_Mestrado* e *Estudante\_Doutorado*,

$$\textit{Estudante\_Mestrado} - \textit{Estudante\_Doutorado}$$

é a relação formada pelas tuplas que pertencem à relação *Estudante\_Mestrado* e não pertencem à relação *Estudante\_Doutorado*. Neste exemplo a relação obtida é igual à relação *Estudante\_Mestrado*,

$$\textit{Estudante\_Mestrado} - \textit{Estudante\_Doutorado} = \textit{Estudante\_Mestrado}.$$

Operador Produto Cartesiano (**Cartesian Product**).

O produto cartesiano entre duas relações  $R$  e  $S$ , denotado por

$$R \times S,$$

é a relação que se forma da combinação de cada tupla de  $R$  com cada uma das tuplas da relação  $S$ . O esquema relacional resultante esta formado por todos os atributos de  $R$  e de  $S$ .

Operação Seleção (**Select**).

A operação de seleção é usada para selecionar um subconjunto de tuplas que satisfazem uma condição em uma relação. É denotada por:

$$\sigma_{\langle \textit{condição de seleção} \rangle}(\langle \textit{nome da relação} \rangle),$$

onde  $\sigma$  é o símbolo usado para denotar o operador de seleção sobre uma relação, indicada pelo nome. A condição de seleção, que pode ser atômica ou composta, é uma expressão lógica especificada sobre os atributos da relação. Em geral, o resultado da operação de seleção pode ser determinada como segue. A  $\langle \textit{condição de seleção} \rangle$  é aplicada independentemente a cada tupla na relação especificada por  $\langle \textit{nome da relação} \rangle$ . No caso de que avaliação da condição seja verdadeira a tupla é selecionada.

Para ilustrar melhor esta operação, definamos a seleção de um subconjunto de tuplas da relação *Estudante\_Mestrado*, tal que, a *Idade* dos estudante seja maior ou igual que 20 anos. A partir desta consulta podemos formar a seguinte condição de seleção,

$$Idade \geq 20;$$

e o operador de seleção estaria expressado como:

$$\sigma_{(Idade \geq 20)}(\langle \langle Estudante\_Mestrado \rangle \rangle).$$

Ao realizar esta operação obtemos a nova relação *Estudante\_Mestrado\_Idade ≥ 20* definida como segue:

Nome	Idade	Endereço	Fone	RA	RG	Nacional.	Área	Graduação
Vladimir Velazco	26	Sta. Cruz 315, Cambuí	Nulo	956698	V167555-Q	Cuba	automação	01
Ronaldo Ruis	23	Ave. Anchieta 245, Centro.	2357589	967690	H123745-O	Brasil	Telemática	02

Tabela 2.4. Relação *Estudante\_Mestrado\_Idade ≥ 20*

Operação Projeção (**Project**).

A operação de projeção, seleciona um conjunto de atributos de uma relação. Em geral é denotada por:

$$\pi_{\langle \text{lista de atributos} \rangle}(\langle \langle \text{nome da relação} \rangle \rangle),$$

onde  $\pi$  é o símbolo usado para denotar o operador de projeção. Esta operação projeta uma  $\langle \text{lista de atributos} \rangle$  sobre uma relação especificada por  $\langle \text{nome da relação} \rangle$ . Esta operação elimina implicitamente as tuplas duplicadas na relação resultante se na lista de atributos foram incluídos só atributos não chaves.

Por exemplo, sobre a relação *Estudante\_Mestrado* podem ser projetados os atributos *RA*, *nome* e *Nacionalidade*. A partir disto, o operador de projeção é expressado como

$$\pi_{\langle RA, nome, nacional \rangle}(\langle \langle Estudante\_Mestrado \rangle \rangle)$$

O resultado da operação é a relação que mostra a tabela 2.5.

Nome	RA	Nacional.
Vladimir Velazco	956698	Cuba
Ronaldo Ruiz	967690	Brasil
José Gómez	973424	Peru

Tabela 2.5. Projeção sobre *Estudante\_Mestrado*.

Na relação obtida não foi preciso eliminar tuplas redundantes pois a chave primaria da relação estava incluída na lista de atributos que foram projetados.

Operação Junção (**Join**).

A operação junção é usada para combinar duas tuplas de duas relações em uma única relação. Em geral a forma da operação junção sobre duas relações  $R(A_1, A_2, \dots, A_n)$  e  $S(B_1, B_2, \dots, B_n)$  é denotada por:

$$R \triangleright \triangleleft \langle \text{Condição de junção} \rangle S,$$

onde  $\langle \text{Condição de junção} \rangle$  representa a condição para que as tuplas das relações  $R$  e  $S$  se combinarem. A junção constrói uma relação composta de todas as possíveis combinações de pares de tuplas das relações  $R$  e  $S$ , que satisfazem uma condição.

Para mostrar como esta operação funciona, considera-se a existência de uma relação que representa a informação das universidades que estão catalogadas. A relação está formada por três atributos, *Código da Universidade*, *País*, *Nome da Universidade*. A tabela 2.6 representa a relação *Universidades*.

Código	País	Universidade
01	Cuba	Universidade da Havana
02	Brasil	Universidade Estadual de Campinas
03	Perú	Universidade Católica de Lima
04	Cuba	Universidade De Stg. de Cuba

Tabela 2.6. Relação Universidades.

No exemplo, será realizada a junção das relações *Estudante\_Mestrado* e *Universidades*, tal que os códigos das Universidades sejam os mesmos. A relação da tabela 2.7 é o resultado da consulta. O operador de junção está expressado como segue:

$$Estudante\_Mestrado \triangleright \triangleleft_{Graduação=Código} Universidades,$$

Nome	...	Graduação	Código	País	Universidade
Vladimir Velazco	...	01	01	Cuba	Universidade da Havana
Ronaldo Ruiz	...	02	02	Brasil	Universidade Estadual De Campinas
José Gómez	...	03	03	Perú	Universidade Católica de Lima

Tabela 2.7. Relação resultante da junção

### 2.1.3.2 O Calculo Relacional.

O calculo relacional é uma linguagem formal, fundamentada no ramo da lógica matemática chamado de calculo de predicados. Existem duas formas conhecidas na qual o calculo de predicados pode ser adaptado em uma linguagem para bancos de dados relacionais: o calculo relacional de tuplas e o calculo relacional de domínios. Ambas as duas são adaptações do calculo de predicados de primeira ordem. Nos trabalhos de Ullman [UI82] e Elmasri [EN94] é mostrada uma explicação ampla sobre este tema.

### 2.1.4 Tratamento da informação incompleta em bancos de dados relacionais.

Define-se como informação incompleta todos os casos em que não podemos associar um valor para um atributo ou, não podemos assumi-lo em forma precisa, seja qual fosse o motivo pelo qual não possuímos o valor. No modelo relacional a forma para representar esta informação é o valor nulo. Existem varias interpretações para os valores nulos em bancos de dados relacionais:

1. O atributo não é aplicável a esta tupla. Por exemplo na relação da tabela 2.1 o atributo *Fone* pode não ser aplicável ao objeto Vladimir Velazco, pois ele não possui telefone.
2. O valor do atributo para esta tupla é desconhecido. Para o mesmo caso pode ser que se saiba que Vladimir Velazco tem telefone mas não se conhece o número.
3. Não se sabe se o valor é indefinido ou se é desconhecido. Tomando novamente o mesmo exemplo, pode acontecer que não se saiba se ele tem telefone ou não.

Adicionalmente, se muitos dos atributos não são aplicáveis às tuplas de uma relação, vão existir muito valores nulos na relação. Este tipo de valores ocupam espaço e levam a problemas na interpretação do significado dos atributos. Além disto, gera problemas adicionais na realização de uma consulta sobre a relação, especificamente nos operadores binários de comparação.

#### **2.1.4.1 Abordagens existentes para o tratamento do valor nulo.**

Nesta seção serão apresentadas algumas das abordagens que existem para o tratamento do valor nulo em bancos de dados e portanto para o tratamento da informação imprecisa.

O primeiro trabalho para o tratamento do valor nulo foi apresentado por Codd[Co79], que utiliza uma lógica de três valores, onde os valores verdades são: 1(verdade),  $\frac{1}{2}$ (desconhecido), 0(falso). Este trabalho não resultou completo pois não incluía todos os possíveis tipos de valores nulos e nem sempre eram obtidos os resultados desejáveis nas consultas.

Por outro lado, Grant[Ga77] critica o tratamento que Codd faz dos valores nulos numa linguagem para a atualização e recuperação dos dados de uma relação. Também propõe um novo tratamento para este tipo de valores.

Na proposta de Grant, a cada ocorrência do valor nulo substitui-se por um valor possível ( um membro do correspondente domínio). Se o valor da expressão é TRUE ( respetivamente FALSE ) para todas as possíveis substituições assinala o valor TRUE ( respetivamente FALSE ) para a expressão, de outra maneira assinala o valor UNKNOWN.

Com relação ao tipo de valor nulo que indica que o domínio não é aplicável à tupla faz uma outra proposta. A sugestão é que nesses caso seja usada a lógica binaria para que quando apareça uma ocorrência em uma formula, deva ser assinalado o valor FALSE.

Uma abordagem mais geral é proposta por Lipski [Lip79]. Ele define um modelo matemático de um banco de dados com informação incompleta, que é chamado de *Sistemas de Informação*.

Basicamente, estes sistemas armazenam informação relacionada com as propriedades dos objetos. A informação pode ser incompleta do ponto de vista de que pode ser conhecido ou não se o objeto possui a propriedade. Lipski também descreve uma linguagem para consultar tais sistemas.

## **2.2 Conjuntos Nebulosos.**

A matemática discreta é sem duvida uma ferramenta básica no tratamento da informação e da computação. Na área de bancos de dados, a teoria de conjuntos é fundamental na formalização do modelo relacional. Uma grande contribuição à teoria de conjuntos foi dada por Zadeh [Za65], quando expandiu o conceito binário de pertinência de um elemento a um conjunto para um conceito de multivalor, dando origem à lógica nebulosa. Seguidamente ao trabalho de Zadeh, uma grande quantidade de trabalhos foram apresentados na mesma linha [Ka75, WF97, YF94]. Devido ao fato do presente trabalho se referir a bancos de dados nebulosos e não

especificamente a lógica nebulosa, serão apresentados aqui somente conceitos e assuntos específicos aquela aplicação de bancos de dados.

### 2.2.1 Conceito de Conjunto Nebuloso.

Seja  $U$  um universo de discurso discreto ou contínuo, que descreve uma coleção de objetos  $\{u\}$ , onde  $u$  representa os elementos genéricos de  $U$ .

Um **Conjunto Nebuloso**  $F$  em um universo  $U$  está caracterizado por uma função de pertinência  $\mu_F$ , a qual toma valores no intervalo  $[0,1]$ , isto é,  $\mu_F:U \rightarrow [0,1]$ . Um conjunto nebuloso pode ser visto como uma generalização do conceito de conjunto cuja função de pertinência toma dois valores,  $\{0,1\}$ . Desta maneira, um conjunto nebuloso  $F$  em um universo  $U$  é representado como um conjunto de pares ordenados de elementos genéricos  $u$  e o seu grau de pertinência:  $F = \{(u, \mu_F(u)) \mid u \in U\}$ .

O **suporte** de um conjunto de nebuloso  $F$ , é um subconjunto do conjunto universo  $U$  tal que

$$Supp(F) = \{u \mid u \in U \text{ e } \mu_F(u) > 0\}.$$

Em particular, o elemento  $u \in U$  tal que  $\mu_F(u) = 0.5$  é chamado de **ponto de inflexão**.

Outra definição relacionada ao suporte é o conceito de  $\alpha$ -cut. O  $\alpha$ -cut,  $F_\alpha$  de um conjunto nebuloso  $F$  é um subconjunto de  $F$  cujos elementos tem como função de pertinência um valor maior ou igual ao valor  $\alpha$ :

$$F_\alpha = \{u \mid \mu_F(u) \geq \alpha\} \text{ com } 0 \leq \alpha \leq 1.$$

O conjunto nebuloso  $F$  é dito **normalizado** se:

$$\exists u \in U, \mu_F(u) = 1,$$

isto é, se existe um elemento  $u$  no conjunto tal que o valor da função de pertinência para esse valor é 1.

### 2.2.2 Operações em Conjuntos Nebulosos.

As operações sobre conjuntos nebulosos estão definidas através das funções de pertinência que as descrevem. Sejam  $F$  e  $G$  dois conjuntos nebulosos definidos sobre um universo  $U$ , com funções de pertinência  $\mu_F$  e  $\mu_G$  respectivamente.

**Igualdade** de conjunto:

$$F = G \Leftrightarrow \mu_F(u) = \mu_G(u) \quad \forall u \in U,$$

$F$  e  $G$  são iguais se e somente se, os valores das funções de pertinência forem iguais para cada elemento  $u$  dos conjuntos.

**Inclusão** de conjuntos:

$$F \subseteq G \Leftrightarrow \mu_F(u) \leq \mu_G(u) \quad \forall u \in U,$$

O conjunto  $F$  está incluído no conjunto  $G$  se e somente se, o valor da função de pertinência para todo elemento  $u$  em  $F$  é menor ou igual a valor da função de pertinência para o mesmo elemento em  $G$ .

**Complemento** de um conjunto:

$$\bar{F} = \{u / (1 - \mu_F(u))\},$$

O complemento de um conjunto nebuloso  $F$  é o conjunto formado pelos elementos  $u$  tais que, o valor da função de pertinência para estes elementos é definido como o complemento do valor da função de pertinência do elemento em  $F$  ( $1 - \mu_F(u)$ ).

**União de conjuntos:**

A função de pertinência da união de dois conjuntos nebulosos  $F$  e  $G$ , denotada por  $F \cup G$ , esta definida como:

$$\mu_{F \cup G} = f(\mu_F(u), \mu_G(u)), u \in U,$$

onde  $f$  é uma S - norma [WF97].

Uma S - norma é uma operação binária  $s:[0,1]^2 \rightarrow [0,1]$ , conhecida também como co-norma, que satisfaz as seguintes regras:

Comutatividade	$x s y = y s x,$
Associatividade	$x s (y s z) = (x s y) s z,$
Monotonicidade	Se $x \leq y$ e $w \leq z$ então $x s w \leq y s z,$
Condição de fronteira	$x s 0 = x \quad x s 1 = 1.$

O operador  $max$  é uma s-norma.

**Interseção de conjuntos:**

A função de pertinência da interseção de dois conjuntos nebulosos  $F$  e  $G$ , denotada por  $F \cap G$ , esta definida como:

$$\mu_{F \cap G} = g(\mu_F(u), \mu_G(u)), u \in U,$$

onde  $g$  é uma T - norma [WF97].

Uma T- norma é uma operação binária  $s:[0,1]^2 \rightarrow [0,1]$ , que satisfaz os regras abaixo:

comutatividade	$x t y = y t x,$
----------------	------------------

associatividade  $xt(ytz) = (xty)tz,$

monotonicidade Se  $x \leq y$  e  $w \leq z$  então  $xtw \leq ytz,$

Condição de fronteira  $0tx = 0$   $1tx = x.$

O operador *min* é uma t-norma.

### 2.2.3 Princípio de extensão.

O princípio de extensão foi apresentado por primeira vez por Zadeh em 1975 [Za75a]. Este princípio é uma identidade básica que permite aos domínios de definição de funções ou relações, serem estendidos de pontos de um domínio  $U$  para subconjuntos nebulosos do domínio  $U$ . Suponha que  $f$  é uma função,  $f:U \rightarrow V$  e  $F$  um subconjunto nebuloso de  $U$  expressado como

$$F = ((u_1, \mu_1), (u_2, \mu_2), \dots, (u_n, \mu_n)),$$

O princípio de extensão expressa que, podemos obter um conjunto nebuloso  $G$  definido em  $V$  a partir do mapeamento de  $f$  em  $F$ :

$$G = f(F) = f((u_1, \mu_1), (u_2, \mu_2), \dots, (u_n, \mu_n)) = \sup_u F(u),$$

isto é, a imagem de  $F$  sobre  $f$  pode ser deduzida do conhecimento das imagens de  $u_1, u_2, \dots, u_n$  sobre  $f$ . Para ilustrar melhor este conceito vamos a mostrar um exemplo. Suponha os domínios  $U$  e  $V$

$$U = (1,2,3,4,5,6,7) \text{ e } V(0,1,2,4,5,6,8,9,10,11,12,13,14,15),$$

e seja  $f:U \rightarrow V$  uma função definida como,  $f = 2*x$ . Tomemos um subconjunto nebuloso  $F$  definido sobre o domínio  $U$  como segue

$$F = ((0,0)(1,0.25), (2,0.5), (3,0.75), (4,1), (5,1), (6,1), (7,1))$$

Quando é aplicado o princípio de extensão, é obtido o conjunto nebulosos  $G$ :

$$G = f(F) = ((0,0), (2,0.25), (4,0.5), (6,0.75), (8,1), (10,1), (12,1), (14,1)),$$

como mostra a figura 2.1.

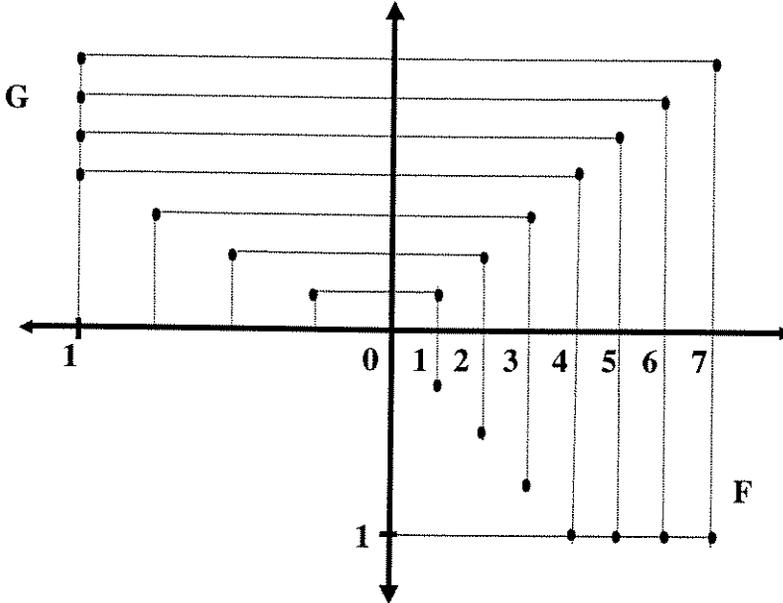


Figura 2.1. Aplicação do Princípio de Extensão.

### 2.2.4 Números nebulosos.

Em muitas situações, a caracterização da informação numérica só pode ser feita de maneira imprecisa e termos como, *perto de cinco*, *mais ou menos 10* são amplamente usados. Estes termos são exemplos dos chamados **números nebulosos**.

Define-se como **número nebuloso** sobre um universo contínuo  $U$ , um conjunto nebuloso  $F$  em  $U$  que é normalizado e convexo (figuras 2.2 e 2.3), isto é,  $F$  cumpre que :

$$\max_{u \in U} \mu_F(u) = 1, \text{ ( Normalizado ) e}$$

$$\mu_F(\lambda u_1 + (1 - \lambda)u_2) \geq \min(\mu_F(u_1), \mu_F(u_2)), \quad u_1, u_2 \in U, \lambda \in [0, 1], \text{ (Convexo).}$$

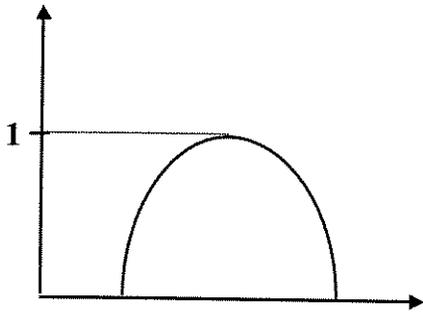


Figura 2.2. Função Convexa

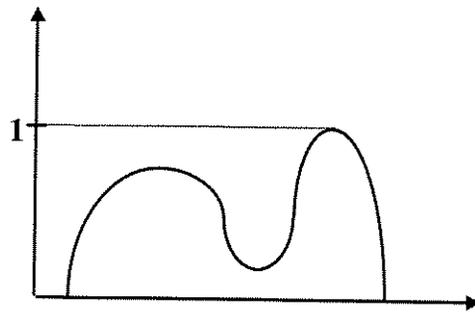


Figura 2.3. Função Não Convexa.

### 2.2.5 Variáveis lingüísticas.

Zadeh em [Za75a, Za75c] faz um estudo sobre as variáveis lingüísticas e a sua aplicação no raciocínio aproximado. Uma variável lingüística pode ser considerada como uma variável cujos valores são números nebulosos. Mas, especificamente, é definida como segue:

**Definição:** Uma variável lingüística é caracterizada por uma quintupla

$$(x, T(x), U, G, M)$$

onde  $x$  é o nome da variável;  $T(x)$  é um conjunto de termos de  $x$ , isto é, um conjunto de nome de valores lingüísticos de  $x$  onde cada valor é um número nebuloso definido sobre  $U$ ;  $G$  é uma gramática para gerar os nome de  $x$ ; e  $M$  é uma regra semântica que associa cada termo com seu significado, isto é, um conjunto nebuloso sobre  $U$ .

Para apresentar melhor este conceito pode-se tomar a variável *altura* como uma variável lingüística. Com isto, o conjunto de termos  $T(\text{altura})$  pode ser:

$$T(\text{altura}) = \left\{ \begin{array}{l} \text{muito alto, alto, não muito alto, mediano,} \\ \text{não muito baixo, baixo, muito baixo} \end{array} \right\}$$

onde cada termo é um valor lingüístico da variável *altura*. As regras semânticas de  $M$  definem os conjuntos nebulosos sobre  $U = [50, 250]$  que caraterizam cada termo.

### 2.2.6 Teoria de possibilidade.

Seja  $X$  uma variável que toma valores no universo  $U$ , e  $u$  um elemento genérico de  $U$ ;

$$X = u$$

significando que a  $X$  é assinalado o valor  $u$ ,  $u \in U$ .

Seja  $F$  um subconjunto nebuloso de  $U$  caracterizado pela função de pertinência  $\mu_F$ .

Define-se  $F$  como uma restrição nebulosa de  $X$  se,  $F$  atua como restrição dos valores que devem ser assinalados a  $X$ , isto, é expressado como:

$$X = u: \mu_F(u)$$

onde  $\mu_F(u)$  é interpretado como o grau de restrição, representado por  $F$ , satisfeito quando  $u$  é assinalado a  $X$ .

Para ilustrar este conceito, consideremos a proposição  $X$  é  $F$ . A variável  $X$  é o nome do objeto, da variável ou da proposição; e  $F$ , é o nome do conjunto nebuloso de  $U$ . Por exemplo “*Luís é alto*” define *Luís* como o objeto e *alto* como o subconjunto nebuloso.

Esta proposição é expressada como:

$$R(A(X)) = F$$

onde  $A(X)$ , é o atributo de  $X$  implicado na restrição que toma valores sobre  $U$ . No exemplo o atributo implicado do objeto *Luís* seria sua *altura*:

$$Luís \text{ é } alto \rightarrow R(altura(Luís)) = alto$$

Isto significa que a proposição  $X$  é  $F$  tem o efeito de assinalar  $F$  à restrição nebulosa associada a  $A(X)$ .

Quando  $X = u$ ,  $\mu_F(u)$  pode ser interpretado como o grau de compatibilidade de  $u$  a  $F$ . Com isto define-se que a proposição  $X \text{ é } F$  converte o significado de  $\mu_F(u)$ , (grau de compatibilidade de  $u$  com  $F$ ), no grau de possibilidade para que  $X = u$  dê a proposição  $X \text{ é } F$ . Com isto são relacionados os conceitos de restrição nebulosa e distribuição de possibilidade, que a seguir será apresentado. Para simplificar assumiremos que  $X = A(X)$ .

### 2.2.6.1 Distribuição de possibilidade. Definição.

Seja  $F$  um subconjunto do universo de discurso  $U$  caracterizado pela função de pertinência  $\mu_F$ , com grau de pertinência  $\mu_F(u)$  interpretado como a compatibilidade de  $u$  com  $F$ . Seja  $X$  uma variável definida sobre  $U$  e seja  $F$  uma restrição nebulosa,  $R(X)$ , associada a  $X$ . Então a proposição  $X \text{ é } F$ , é interpretada como:

$$R(X) = F ,$$

associa a *distribuição de possibilidade* [Za78],  $\Pi_X$ , com  $X$  definida igual a  $R(X)$

$$\Pi_X = R(X).$$

A função de distribuição de possibilidade associada com  $X$  é denotada por  $\pi_X$  e é definida numericamente igual à função de pertinência de  $F$ ,

$$\pi_X = \mu_F .$$

### 2.2.6.2 Medidas de Possibilidade e Necessidade.

A medida de possibilidade do conceito associado com o conjunto nebuloso  $F$  é

$$Poss(X \text{ é } F) \equiv \Pi_X(F) = \sup_{\forall u \in U} (\min\{\mu_F(u), \pi_X(u)\}).$$

A medida de necessidade é

$$Nec(X \text{ é } F) \equiv N_X(F) = \inf_{\forall u \in U} \left( \max\{\mu_F(u), 1 - \pi_X(u)\} \right).$$

A necessidade de um evento corresponde com a impossibilidade do evento oposto. Esta é a relação entre as duas medidas.

### 2.2.7 Similaridade.

Uma relação de similaridade [Za71] define uma relação de equivalência  $s(x, y)$ , sobre um domínio  $U$ . A relação  $s$  é o mapeamento de cada par de elementos do domínio sobre o intervalo  $[0, 1]$ , tal que  $s$  deve ser reflexiva, simétrica e transitiva. Isto é,

1. Reflexiva:  $s_U(x, x) = 1$ .
2. Simétrica:  $s_U(x, y) = s_U(y, x)$ .
3. Transitiva:  $s_U(x, z) \geq \max(\text{oper}[s_U(x, y), s_U(y, z)])$ .

O símbolo *oper* substitui os operadores *min* ou *\** (produto), chamadas formas *max-min* e *max-produto* respetivamente.

Um relação de similaridade  $s$ , pode ser representada como uma família aninhada de seus  $\alpha$ -cuts,  $s_\alpha$ . Cada  $\alpha$ -cut é uma relação de equivalência sobre  $U$

$$s_\alpha = \{(x, y) | s(x, y) \geq \alpha\},$$

e constitui uma partição nebulosa sobre  $U$ .

Os  $\alpha$ -cuts ou níveis de refinamento de uma relação de similaridade são representadas em forma de árvore de partições.

Para ilustrar as relações de similaridade considere-se a relação,  $s$ , da tabela 2.8. No exemplo se define o universo  $U = \{w, x, y, z\}$ .

$s$	w	x	y	z
w	1	0.2	0.2	0.2
x	0.2	1	0.5	0.3
y	0.2	0.5	1	0.3
z	0.2	0.3	0.3	1

Tabela 2.8. Relação de similaridade,  $s$ .

Da relação  $s$ , podem ser obtidas as seguintes classes de equivalência:

$$\begin{aligned} & \{\{w\}, \{x\}, \{y\}, \{z\}\}: s_1 \\ & \{\{w\}, \{x, y\}, \{z\}\}: s_{0.5} \\ & \{\{w\}, \{x, y, z\}\}: s_{0.3} \\ & \{\{w, x, y, z\}\}: s_{0.2} \end{aligned}$$

A árvore de partições desta relação é apresentada na figura 2.4. Como mostra a figura com o incremento de  $\alpha$  as partições diminuem o seu tamanho.

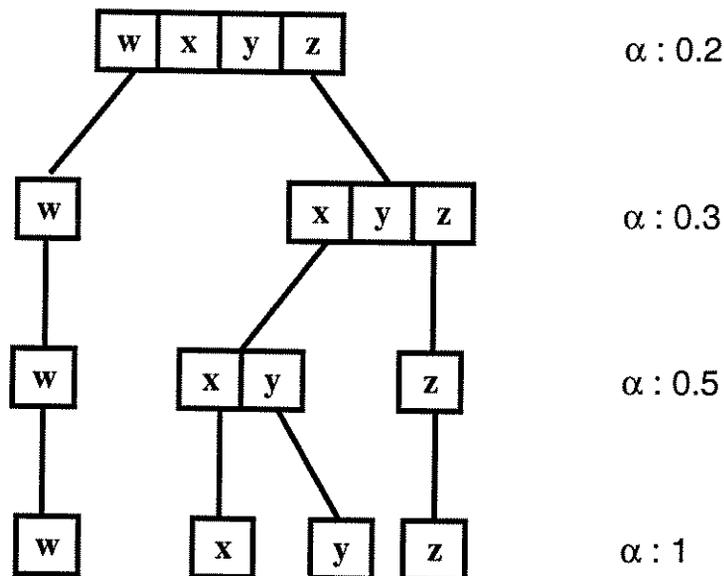


Figura 2.4. Árvore de partições para a relação  $s$ .

# Capítulo 3

## *Modelos de Bancos de Dados Nebulosos*

---

No capítulo anterior foi apresentado o modelo de bancos de dados relacionais. Tal modelo, embora bastante completo, não permite representar a informação como é percebida pelo homem, que se apresenta de maneira imprecisa e incerta. Outro problema que também não resolve é a geração da informação imprecisa a partir daquelas armazenadas em forma precisa.

Neste capítulo são analisadas as principais abordagens encontradas na literatura para a solução destes problemas que tem como base a utilização da teoria da lógica nebulosa para a representação da informação imprecisa e da incerteza. Para facilitar o entendimento das abordagens é apresentado um exemplo onde, a partir de um problema comum de armazenamento de informação é introduzida a imprecisão e a incerteza.

Os itens 3.2, 3.3, e 3.4 apresentam as abordagens com uma análise das suas vantagens e desvantagens. No item 3.5 elas são comparadas entre si.

### **3.1 Apresentação do problema.**

Suponha que se quer armazenar algumas das características físicas de uma conjunto de pessoas e que os dados a serem armazenados sejam: *nome, idade, peso, sexo, altura, cor da pele e cor do cabelo*. Para se formatar esta informação visando a criação de um banco de dados relacional pode se criar uma relação definida como segue:

*Físico ( nome, idade, peso, sexo, altura, cor\_pele, cor\_cabelo )*

Suponha que os fatos conhecidos para preencher esta tabela sejam os seguintes:

Luiz é um jovem alto de 70kg e pele branca e cabelo preto.

Maria é uma mulher gorda de 30 anos de 1.70 m de altura e pele morena e cabelo entre loiro e castanho .

Tenho quase certeza que o Carlos é um velho de 80 anos, baixo, preto, sem cabelo.

Pedro é um rapaz jovem de aproximadamente 80 Kg, alto de pele branca e cabelo loiro.

Eu acho que a Marta é uma mocinha de altura mediana, magra, branca e cabelo moreno.

A partir dos fatos acima, uma definição para os domínios dos atributos pode ser a seguinte:

$Dom(nome) = Nomes,$

$Dom(Idades) = \{[0,100], velho, jovem, maduro, adolescente\},$

$Dom(Peso) = \{[4,100], gordo, médio, magro\},$

$Dom(Sexo) = \{F, M\}$

$Dom(altura) = \{[0.40, 2.05], baixa, mediana, alta\},$

$Dom(Cor_Pele) = \{preto, moreno, branco, amarelo, marrom\},$

$Dom(Cor_Cabelo) = \{loiro, ruivo, preto, marrom, vermelho, castanho\}.$

Mas, a definição destes possíveis domínios não é permitida em bancos de dados relacionais, pois os domínios incluem valores com imprecisão, por exemplo no atributo *Peso*, o valor *magro*. Segundo foi analisado no capítulo 2, o tratamento da

informação imprecisa em bancos de dados relacionais somente é possível através do valor nulo. Com isto, a informação armazenada seria incompleta pela impossibilidade do seu armazenamento.

Para solucionar este problema foram criados novos modelos de bancos de dados: os bancos de dados nebulosos. Estes modelos utilizam a lógica nebulosa para a representação e manipulação da informação.

Basicamente os modelo que usam lógica nebulosa podem ser classificados segundo três enfoques diferentes: (a) modelo relacional nebuloso, (b) modelo de unificação a partir de relações de similaridade, (c) modelo relacional possibilístico. Existem alguns exemplos de aplicações feitas sobre banco de dados nebulosos baseadas nestes modelos [DM89, Li90, Va89]. A seguir é apresentada cada uma das abordagens, assim como uma análise crítica das mesmas, utilizando as idéias apresentadas por Medina e Pons em suas Teses Doutorais respectivamente [Me94, Po94] e alguns pontos comparativos importantes que Motro apresentada no seu artigo [Mo90].

### **3.2 Modelo de Unificação a partir de relações de similaridade.**

A definição proposta por Buckles e Petry [BP82, BP83, BP84, BP89, Pe96], difere em três aspectos da definição de bancos de dados convencionais:

1. Os valores dos atributos para cada tupla não precisam ser valores atômicos, isto é, o valor do atributo pode ser um conjunto de valores do domínio de definição.
2. A relação de identidade é substituída por uma relação de similaridade, onde a relação de identidade é um caso especial.
3. Na consulta, cada atributo possui um limiar de similaridade que representa o grau máximo de satisfação para cada atributo, na condição de seleção.

Neste modelo um banco de dados [BP82] é definido como um conjunto de relações nebulosas,

$$BD = \{R_1, R_2, \dots, R_n\}$$

onde cada relação nebulosa  $R_i$  é definida como subconjunto do produto cartesiano

$$2^{D_1} \times 2^{D_2} \times \dots \times 2^{D_m},$$

onde  $m$  delimita o número de atributos da relação e cada  $2^{D_j}$  ( $j = 1, \dots, m$ ) é um elemento qualquer do conjunto potência do domínio base  $D_j$ ,  $P(D_j) \cap \emptyset$  (o elemento nulo não pertence ao domínio).

Os domínios base,  $D_j$ , são os conjuntos de definição de cada atributo, onde estão incluídos todos os valores possíveis que um atributo  $j$  de uma relação pode tomar. O valor do atributo  $j$  para a tupla  $i$  é um conjunto de valores  $d_{ij}$ , não necessariamente atômico, que é chamado de domínio de valores da tupla  $i$  para o atributo  $j$ .

Segundo a definição deste modelo o elemento nulo é excluído do conjunto base. No modelo o elemento nulo de banco de dados convencionais é tratado como o conjunto de todos os elementos do conjunto base, representando a maior incerteza possível na informação.

Os domínios de valores permitidos para os atributos são:

1. Conjunto finito de escalares. Ex. DE = { feio, bonito, belo }.
2. Conjunto finito de números. Ex. DN = { 15, 16, 17 }.
3. Conjunto de números nebulosos. Ex. DNN = { alto, baixo, meio } [BP84].

A relação *Físico* apresentada na introdução deste capítulo, segundo este modelo, pode ser definida como segue:

Nome	Idade	Peso	Sexo	Altura	Cor_Pele	Cor_Cabelo
Luiz	jovem	70	M	alto	branca	preto
Maria	30	gorda	F	1.70	morena	{loiro, castanho }
Carlos	80	{...}	M	baixo	preto	{...}
Pedro	jovem	aprox. 80	M	alto	branca	loiro
Marta	mocinha	{...}	F	mediana	branca	moreno

Tabela 3.1. Relação *Físico*.

Na tabela 3.1, a constante {...} simboliza que o valor para o atributo é todo o domínio.

A partir da definição da relação *Físico* podem-se induzir três problemas:

1. Quando a informação é desconhecida ou indefinida, o valor do atributo é todo o domínio. Esta definição traz, como consequência, que o tratamento a este tipo de informação não seja correta. Por exemplo, na tupla das descrições de Carlos, o peso é desconhecido e o valor da cor do cabelo não existe, mas os dois recebem o mesmo tratamento.
2. O valor do atributo *Cor\_cabelo* na tupla 2 não é atômico. Não satisfaz a primeira forma normal exigida em bancos de dados relacionais.
3. No inclui um espaço, na definição de relação, para representar o grau de incerteza ao nível de tuplas, que existe nas tuplas 3 e 5.

Neste modelo a incerteza é tratada através da substituição da relação de identidade por relações de similaridade. Esta relação representa a medida em que os valores do domínio são parecidos segundo a visão do observador.

A relação de similaridade é definida como uma função sobre cada par de elemento do domínio base,  $D_j$ , ao intervalo  $[0,1]$ ,

$$s: (x, y) \rightarrow [0, 1],$$

onde  $x, y \in D_j$ . A relação de similaridade,  $s$ , deve ser

1. Reflexiva:  $s(x, x) = 1$ ,
2. Simétrica:  $s(x, y) = s(y, x)$  e
3. Transitiva:

$$T1: \quad s(x, z) \geq \max_{y \in D_j} \{ \min [s(x, y), s(y, z)] \}$$

$$T2: \quad s(x, z) \geq \max_{y \in D_j} \{ s(x, y) * s(y, z) \}$$

onde  $*$  é a multiplicação aritmética.

Seja  $t_i$  uma tupla, que pertence à relação  $R_l$ . Como o valor do atributo  $j$  para a tupla  $i$  é um conjunto, então, para cada tupla,  $t_i$ , define-se um conjunto de interpretações. Uma interpretação  $\alpha = (a_1, a_2, \dots, a_m)$  da tupla  $t_i$ , é qualquer atribuição de valores tal que  $a_j \in d_{i,j} \forall j$  com  $j = 1, \dots, m$ . Note-se que uma interpretação é equivalente a uma tupla em bancos de dados convencionais.

Para cada domínio,  $D_j$ , é definido um limiar de similaridade,  $THRES(D_j)$ , como a mínima similaridade sobre todo o domínio de valores:

$$THRES(D_j) = \min_{\forall i} \left\{ \min_{x, y \in d_{ij}} [s(x, y)] \right\}$$

Em banco de dados relacionais convencionais a cardinalidade de  $d_{ij} = 1$  e  $s(x, x) = 1$  consequentemente:

$$THRES(D_j) = 1, \forall j.$$

A partir das relações de similaridade são definidas classes de equivalência que representam o grau em que os elementos de uma mesma classe são não distinguíveis para um limiar.

No exemplo podem-se definir duas relações de similaridade nos atributos *Cor\_pele* e *Cor\_cabelo* como representam as tabelas 3.2 e 3.3:

<i>S<sub>Cor_pele</sub></i>	branca	morena	preta
branca	1	0.5	0
morena	0.5	1	0.5
preta	0	0.5	1

Tabela 3.2. Relação de similaridade sobre o domínio *Cor da pele*.

<i>S<sub>Cor_Cabelo</sub></i>	loiro	preto	castanho	moreno
loiro	1	0.6	0.4	0
preto	0,6	1	0.5	0.1
castanho	0.4	0.5	1	0.8
moreno	0	0.1	0.8	1

Tabela 3.3. Relação de similaridade sobre o domínio *Cor do Cabelo*.

As relações de similaridade para os atributos *nome* e *sexo* são definidas da mesma maneira que a relação de identidade.

As figuras 3.1 e 3.2 representam a definição dos números nebulosos que pertencem aos domínios *Peso* e *Altura*.

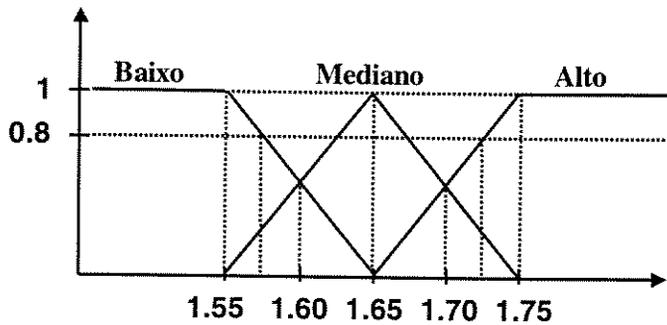


Figura 3.1. Definição dos conjuntos nebulosos definidos sobre o atributo *Altura*.

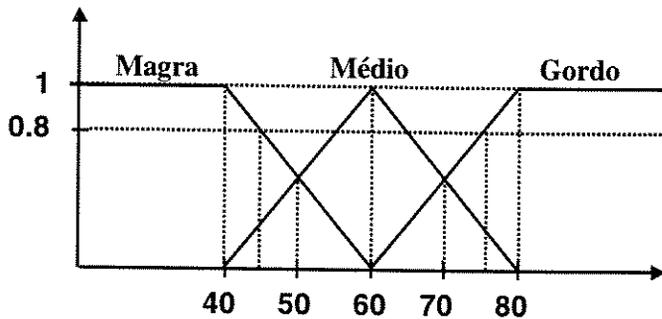


Figura 3.2. Definição dos conjuntos nebulosos definidos sobre o atributo *Peso*.

### 3.2.1 Álgebra Relacional Nebulosa.

As operações da álgebra relacional nebulosa [BP89] são definidas similarmente às operações da álgebra relacional, definidas para bancos de dados convencionais, com a adição de uma cláusula que define o limiar de similaridade mínimo para cada atributo.

O limiar de similaridade significa o grau de similaridade mínimo que devem cumprir os valores do domínio. Este limiar é definido para cada atributo da relação no momento da consulta e no caso de não aparecer será assumido o valor 1. Isto significa que, para o resto dos valores a similaridade se traduz em igualdade.

Em bancos de dados relacionais, uma tupla é definida como redundante se é exatamente igual a outra tupla. Toda operação da álgebra relacional de bancos de dados relacionais considera a eliminação das tuplas redundantes. Segundo o modelo de Buckles e Petry, uma tupla é redundante se pode ser fundida com outra tupla através da união dos seus correspondentes domínios de valores, sem violar o limiar de similaridade definido para o domínio.

### Tuplas Redundantes

Duas tuplas  $t_i = (d_{i1}, d_{i2}, \dots, d_{im})$  e  $t_k = (d_{k1}, d_{k2}, \dots, d_{km})$ ,  $i \neq k$ , são redundantes se:

$$THRES(D_j) \leq \min_{x,y \in d_{ij} \cup d_{kj}} [s(x,y)]$$

para  $j = 1, \dots, m$  e  $THRES(D_j)$  dado com anterioridade.

Em bancos de dados relacionais convencionais a relação de similaridade é a de identidade pelo que duas tuplas são redundantes se são iguais,  $s(x,x) = 1$ . Assim uma das tuplas é eliminada; as tuplas não redundantes  $s(x,y) = 0$  pelo que não são eliminadas.

Quando o domínio de valores é definido como números nebulosos  $q_i$ , a relação de similaridade utilizada para conjuntos discretos não pode ser estendida para conjuntos contínuos. Para este tipo de domínios é definida uma nova relação de cercania chamada  $\alpha$ -similaridade o  $\alpha$ -proximidade.

**Definição:**  $q_i$  e  $q_j$  são  $\alpha$ -similares,  $q_i S_\alpha q_j$ , se dados  $\beta \in [0,1]$ ,  $x \in (q_i \cup q_j)_\alpha$ ,  $y \in (q_i \cup q_j)_\alpha$ , e  $z = \beta x + (1-\beta)y$ , então  $z \in (q_i \cup q_j)_\alpha$ .

Isto é, o conjunto  $q_i \cup q_j$  tem que ser convexo para esse  $\alpha$ -cut definido e o  $\alpha$  máximo que pode ser escolhido é a possibilidade entre os dois conjuntos:

$$0 \leq \alpha \leq \text{poss}(q_i, q_j)$$

Se o  $\alpha$  escolhido é maior que a fronteira definida na equação anterior então  $q_i$  e  $q_j$  não são  $\alpha$ -similares.

**Definição:**  $q_i$  e  $q_j$  são  $\alpha$ -próximas,  $q_i S_\alpha^+ q_j$ , se existem zero ou mais números nebulosos,  $q_h, q_k, \dots, q_p$ , tal que  $q_i S_\alpha q_h S_\alpha q_k S_\alpha \dots S_\alpha q_p S_\alpha q_j$ .

Com isto a definição de tuplas redundantes é ampliada. As tuplas  $t_i$  e  $t_j$  são redundantes [BP84] se simultaneamente :

1.  $d_{ij} \cup d_{kj}$  não viola o limiar de similaridade para qualquer domínio discreto  $D_j$ .
2.  $q_{ij}$  e  $q_{kj}$  são  $\alpha_j$ -similares para qualquer domínio de números nebulosos  $q_j$ ,

As tuplas redundantes são eliminadas pela fusão (via união de conjuntos ) dos respectivos domínios das tuplas originais.

Os autores deste modelo definiram o valor de pertinência das tuplas obtida a partir de uma consulta á relação resposta [BP83]. Mas o modelo que definirão não inclui mecanismos para que essa pertinência seja representável como parte do conceito de relação. A seguir é mostrado como este valor é calculado.

A consulta  $Q(.)$  produz o valor de pertinência  $\mu_Q(t)$  para a tupla  $t$ , da relação resposta  $R_l$ , como segue:

Seja  $a \in D_j$  um elemento qualquer e seja o valor de pertinência ao domínio  $\mu_a(b)$ , onde  $b \in D_j$  definido igual a  $s_j(a, b)$ . Os passos para calcular o valor são:

1. Cada interpretação  $I = (a_1, a_2, \dots, a_n)$  de  $t$  determina um valor  $\mu_{a_j}(a_j)$  para cada elemento  $a_j$  do domínio de  $Q(a_i, a_h, \dots, a_k)$ .
2. A avaliação dos modificadores e os operadores em  $Q(.)$  sobre os valores de pertinência  $\mu_{a_j}(a_j)$ , produz  $\mu_Q(I)$ , o valor de pertinência da interpretação com respeito à consulta.
3. Finalmente,  $\mu_Q(t) = \max_{I \text{ det}} \{ \mu_Q(I) \}$ .

Resumindo, o valor de pertinência da tupla representa a melhor interpretação dos casamentos. A relação resposta é então, o conjunto de tuplas que possuem um valor de pertinência maior que zero, mas na pratica é mais realista considerar somente as tuplas com maior valor.

No exemplo serão analisadas duas consultas:

*Obter os nomes das pessoas de pele branca com similaridade 0.5.*

*Obter as pessoas com altura muito próxima 1.73m*

A primeira consulta poderia ser definida como segue em um SQL-estendido:

**Select** Nome

**From** fisico

**Where** pele= branca

**With**  $THRES(pele) \geq 0.5$

Os umbrais de similaridade para o resto dos atributos se definiria igual a 1,  $THRES(D_j) = 1 \quad \forall j \neq Cor\_pele$ .

Como primeiro passo para a execução desta consulta, são criadas as classes de equivalência sendo obtida a relação da tabela 3.4. A tabela 3.5 representa o valor de pertinência obtida para cada tupla depois da consulta.

Nome	Cor_Pele
Luiz	Branca
Maria	Morena
Pedro	Branca
Marta	Branca

Tabela 3.4. Relação Resposta

Certeza
1
0.5
1
1

Tabela 3.5. Pertinência da tupla à relação resposta

Para a solução da segunda consulta é feita a suposição de que a expressão “*muito próxima*” significa que o limiar de similaridade para o atributo altura é 0.8. A partir disto a consulta pode ser definida:

**Select** Nome

**From** fisico

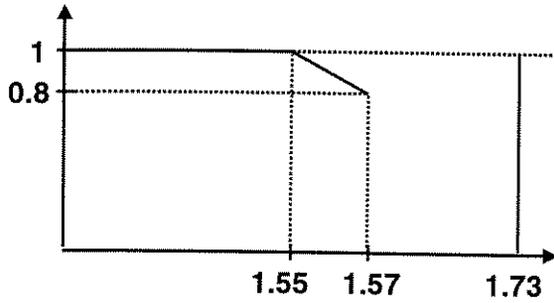
**Where** altura= 1.73

**With**  $THRES(altura) \geq 0.8$

Para a determinação das classes de equivalência é preciso comprovar-se:

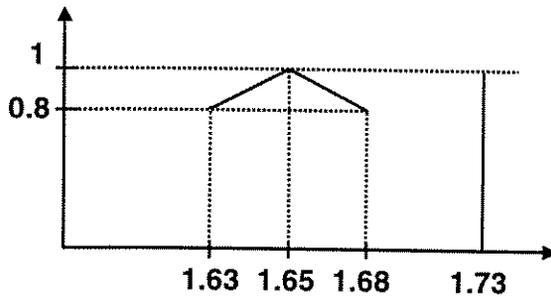
$$(q_{altura})_{0.8} \cap (1.73/1) \neq \emptyset \quad (1).$$

As figuras a seguir representam a aplicação desta expressão para cada conjunto nebuloso definido sobre o domínio *Altura* e o resultado obtido.



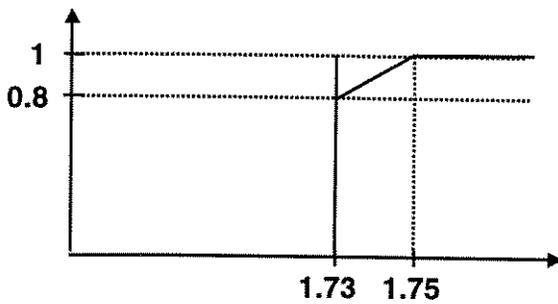
$$(\text{baixo})_{0.8} \cap (1.73/1) = \emptyset$$

Figura 3.3. Aplicação da fórmula 1 no conjunto *baixo*.



$$(\text{mediano})_{0.8} \cap (1.73/1) = \emptyset$$

Figura 3.4. Aplicação da fórmula 1 no conjunto *mediano*.



$$(\text{alto})_{0.8} \cap (1.73/1) = \{1.73/0.8\}$$

Figura 3.5. Aplicação da fórmula 1 no conjunto *alto*.

A relação resposta da consulta define-se na tabela 3.6.

Nome	Altura
Luiz	Alto
Pedro	Alto

Tabela 3.6. Relação Resposta

Certeza
0.8
0.8

Tabela 3.7. Pertinência da tupla à Relação Resposta

Em definições posteriores Shenoi e Melton [SM89, SM90] estenderam esta abordagem utilizando relações de proximidade as quais substituem as relações de similaridade utilizadas na definição de Buckles e Petry.

### 3.2.2 Análise Crítica.

#### Vantagens:

1. Este modelo é utilizado quando não é possível a formulação precisa dos valores dos atributos a partir de uma definição matemática, e a incerteza dos parâmetros do problema não é possibilística, portanto oferece uma ferramenta para representar a imprecisão de conceitos do tipo qualitativo,

Exemplo: { *loiro, preto, marrom* }

2. Introduce o uso de umbrais de forma independente para cada atributo.

#### Desvantagens:

1. Não satisfaz a primeira forma normal exigida em bancos de dados relacionais convencionais. Isto é, em banco de dados relacionais convencionais se exige que o valor de um atributo deve ser indivisível, mas neste modelo o valor dos atributos pode ser um conjunto de valores.

2. Este modelo não realiza um bom gerenciamento da informação desconhecida ou nula.

3. Não permite modelar bem todos os aspectos nebulosos da informação imprecisa, como é o caso dos termos: *jovem, alto, grande*.

4. Não permite incluir no modelo a incerteza ao nível de tuplas.

### 3.3 Modelo Relacional Nebuloso.

No modelo relacional introduzido por Codd [Co70], um banco de dados é definido como um conjunto de relações, onde cada uma é um subconjunto do produto cartesiano dos domínios que a compõem.

Esta abordagem encontrada nos trabalhos [Li90, BZ84, Um83], define uma relação como um subconjunto nebuloso do produto cartesiano dos domínios que a formam. Com isto, o conceito de pertinência TRUE (1) ou FALSE (0) de uma tupla à relação é substituído por valores verdade entre zero (0) e um (1) ou distribuições de possibilidade sobre  $[0,1]$ . Estes valores verdade caracterizam o grau de pertinência ou compatibilidade de uma tupla com a relação nebulosa, assim como o grau de incerteza na informação que essa tupla representa.

Com isto, um banco de dados nebuloso é definido como um conjunto de relações nebulosas. Cada tupla é caracterizada por uma função de pertinência:

$$BD = \{R_1, R_2, \dots, R_m\}$$

$$\mu_{R_j} : D_1 \times D_2 \times \dots \times D_n \rightarrow P([0,1]) \quad j = 1, \dots, m$$

$D_i$  : Domínio do  $i$ -ésimo atributo da relação  $R_j$  e  $\times$  define o produto cartesiano.

Este modelo é utilizado por Baldwin [BZ84] para a definição de uma linguagem de inferência. Também é incluído na definição de bancos de dados nebulosos utilizada por Li et al. [Li90] na implementação de um sistema de banco de dados nebulosos sobre Prolog. O modelo proposto por Umano [Um83] também utiliza esta abordagem.

Seguindo o modelo em discurso a relação *físico* estaria definida como mostra a tabela 3.8.

Nome	Idade	Peso	Sexo	Altura	Cor_Pele	Cor_Cabelo	Certeza
Luiz	Nulo	70	M	Nulo	branca	preto	1
Maria	30	Nulo	F	1.70	morena	Nulo	1
Carlos	80	Nulo	M	Nulo	preto	Nulo	0.8
Pedro	Nulo	Nulo	M	Nulo	branca	loiro	1
Marta	Nulo	Nulo	F	Nulo	branca	moreno	0.7

Tabela 3.8. Representação da relação físico no Modelo Relacional Nebuloso.

Na tabela, foi considerado que o termo “quase certeza” significa que a informação é 0.8 certa, e o termo “acho que” significa que é 0.7 certa.

No modelo, não é possível representar a incerteza para cada um dos atributos da relação.

### 3.3.1 Análise Crítica.

#### Vantagens:

Define um meio para representar incerteza no tratamento da informação, com a introdução do valor de função de pertinência para cada tupla da relação.

#### Desvantagens:

Cada tupla assume seu caráter nebuloso de forma global, sem que possa ser representado o caráter nebuloso de cada atributo da relação.

### 3.4 Modelo Relacional Possibilístico.

Sobre o modelo Relacional Possibilístico existem varias abordagens propostas, uma por Umano [UM89, Um83], outra por Prade-Testemale [PL84, PT84] e outra Zemankova [ZK85, Ze89]. Todas representam a imprecisão dos dados por meio do

uso da distribuição de possibilidade definida por Zadeh [Za78] e diferenciam-se basicamente em:

1. Representação da distribuição de possibilidade e gerenciamento da informação desconhecida e nula; apresentado em banco de dados convencionais nos trabalhos de Lipske [Li79] e de Grant [Ga77],
2. Resolução da consulta.
3. Eleição dos operadores de seleção para avaliar as condições impostas na consulta

Nestas abordagens um banco de dados é definido como um conjunto de relações

$$BD = \{R_1, R_2, \dots, R_m\},$$

no qual, cada relação esta definida como:

$$R_j: P(D_1) \times P(D_2) \times \dots \times P(D_n) \quad j = 1, \dots, m,$$

onde  $P(D_i)$  representa a família de todas distribuições de possibilidade sobre o domínio  $D_i$  e  $\times$  define o produto cartesiano.

### 3.4.1 Modelo de Prade-Testemale.

O conhecimento disponível, concernente aos valores do atributo  $A$  do objeto  $x$ , é representado pela distribuição de possibilidade  $\pi_{A(x)}$  sobre  $D \cup \{\varepsilon\}$ . O elemento extra  $\varepsilon$ , define o caso em que existe uma possibilidade diferente de 0 de que o atributo não é aplicável ao objeto  $x$ , isto é, um valor indefinido para o objeto  $x$  [PT84].

Junto de um exemplo, utilizando a definição anterior, se analisará os tipos de conhecimentos sobre um fato. Especificamente é modelado o conhecimento sobre “A idade do carro de Luís”, tomando  $D$  como o conjunto das possíveis idades.

1. “Não conhecemos se Luís tem carro ou não”. A idade do carro de Luís pode ser qualquer elemento do domínio  $D \cup \{\varepsilon\}$

$$\pi(d) = 1 \quad \forall d \in D \cup \{\varepsilon\}$$

2. “A possibilidade de que Luís não tenha carro é  $\lambda$ , mas se ele tem carro, é novo”.

$$\pi(\varepsilon) = \lambda, \quad \lambda > 0; \quad \pi(d) = \mu_{\text{novo}}(d) \quad \forall d \in D$$

3. “Luís não tem carro”.

$$\pi(\varepsilon) = 1, \quad \pi(d) = 0 \quad \forall d \in D$$

4. “Luís tem carro, mais não conhecemos a sua idade”.

$$\pi(\varepsilon) = 0, \quad \pi(d) = 1 \quad \forall d \in D$$

5. “O carro de Luís tem entre 2 e 4 anos”.

$$\pi(\varepsilon) = 0, \quad \pi(d) = \begin{cases} 1 & \forall d \in [2,4] \subseteq D \\ 0 & \text{outro caso.} \end{cases}$$

6. “O carro de Luís é novo”.

$$\pi(\varepsilon) = 0, \quad \pi(d) = \mu_{\text{novo}}(d) \quad \forall d \in D$$

7. “O carro de Luís tem dois anos”.

$$\pi(\varepsilon) = 0, \quad \pi(d) = \begin{cases} 1 & d = 2 \\ 0 & \text{outro caso} \end{cases}$$

Neste modelo, uma *relação nebulosa* é uma relação ordinária sobre o produto cartesiano dos domínios de discurso. Estes domínios são distribuições de possibilidade sobre  $D \cup \{\varepsilon\}$ .

A tabela 3.9 representa a relação *físico* segundo a definição de relação nebulosa de Prade e Testemale.

Nome	Idade	Peso	Sexo	Altura	Cor_Pele	Cor_Cabelo
Luiz	jovem	70	M	alto	branca	preto
Maria	30	<i>Null</i>	F	1.70	morena	{ <i>loiro/1, castanho/1</i> }
Carlos	80	<i>Unknown</i>	M	baixo	preto	<i>Undefined</i>
Pedro	jovem	aprox. 80	M	alto	branca	loiro
Marta	mocinha	<i>Unknown</i>	F	mediana	branca	moreno

Tabela 3.9. Representação da relação físico no Modelo de Prade-Testemale.

Segundo as definições de Prade e Testemale os valores nulos (*Null*, *Unknown* e *Undefined*) estariam definidos como seguem:

Atributo *Peso*: Segundo este modelo o conhecimento disponível sobre este atributo define-se sobre o domínio dos pesos  $D_p$ . O domínio  $D_p$ , por sua vez é igual à união do domínio dos reais que descrevem o peso,  $D$ , e do valor  $\epsilon$  que significa o valor indefinido,  $D_p = D \cup \epsilon$ . Com isto o valor *Unknown*, *Undefined* e *Null* definem-se como:

$$Unknown = \{1/d, 0/\epsilon; \quad d \in D\}$$

$$Undefined = \{0/d, 1/\epsilon; \quad d \in D\}$$

$$Null = \{1/Unknown, 1/Undefined\}.$$

Atributo *Cor Cabelo*: Neste caso, define-se como o domínio das cores do cabelo como a união do conjunto das possíveis cores de cabelo  $D$  e do valor  $\epsilon$ . Com isto define-se o valor *Undefined* como:

$$Undefined = \left\{ 0/d, 1/\varepsilon; \quad d \in D \right\}.$$

O modelo de Prade e Testemale constroi as operações de comparação a partir de relações nebulosas sobre o domínio de discurso. Oferece duas versões para estes operadores, uma fundamentada na medida de possibilidade e a outra na medida de necessidade.

A possibilidade de que o valor do atributo  $A$  da tupla  $x$  e o valor do atributo  $B$  se encontrem relacionados a partir do operador  $\Theta$ ,  $A \Theta B$ , é expressado como:

$$\Pi(\Theta|(A(x), B(x))) = \sup_{(d, d') \in D \times D} \min(\mu_{\Theta}(d, d'), \pi_{A(x)}(d), \pi_{B(x)}(d'))$$

onde  $\pi_{A(x)}(d)$  e  $\pi_{B(x)}(d')$  são as distribuições de possibilidade definidas sobre  $A(x)$  e  $B(x)$  respetivamente, e  $\mu_{\Theta}(d, d')$  é a função de pertinência que descreve o operador relacional  $\Theta$ .

Da mesma forma, a necessidade de que dois atributos estejam relacionados nos mesmos termos é expressada como:

$$N(\Theta|(A(x), B(x))) = \inf_{(d, d') \in D \times D} \max(\mu_{\Theta}(d, d'), 1 - \pi_{A(x)}(d), 1 - \pi_{B(x)}(d'))$$

Com a definição destas medidas, a resposta de uma consulta se constitui de dois conjuntos: (1) o conjunto de tuplas que possivelmente satisfazem o operador  $\Theta$  e (2) o conjuntos de tuplas que necessariamente satisfazem o operador  $\Theta$ .

O operador  $\Theta$  representa os operadores relacionais,  $\langle =, \neq, >, \geq, <, \leq \rangle$ . Cada um deles está caracterizado por uma função de pertinência.

Para resolver a consulta “*Obter as pessoas com altura muito próxima a 1.73m*”, deve ser definido como primeiro passo o operador relacional de comparação nebuloso “*muito próxima*”:

$$\mu_{\text{muito\_próxima}}(x, y) = \begin{cases} \frac{-0.02 - (x - y)}{0.02} & \text{se } (x - y) \in [-0.02, 0] \\ \frac{0.02 - (x - y)}{0.02} & \text{se } (x - y) \in [0, 0.02] \\ 0 & \text{se } (x - y) < -0.02 \text{ ou } (x - y) > 0.02 \end{cases}$$

A partir da definição do operador, são calculados os conjuntos de possibilidade e necessidade. No exemplo, a comparação é feita com um *crisp* e portanto a distribuição é  $\mu(173) = 1$ . O conjunto que possivelmente satisfaz a condição e o conjunto que necessariamente satisfaz a condição são definidos a seguir ( $\Theta = \text{Muito\_próxima}$ ):

$$\Pi(\Theta | (A(x), 173)) = \sup_{(d, d') \in \text{Altura} \times \text{Altura}} \min(\mu_{\Theta}(d, d'), \pi_{A(x)}(d), \pi_{173}(d'))$$

$$N(\Theta | (A(x), 173)) = \inf_{(d, d') \in \text{Altura} \times \text{Altura}} \max(\mu_{\Theta}(d, d'), 1 - \pi_{A(x)}(d), 1 - \pi_{173}(d'))$$

Primeiro o modelo resolve a composição do operador relacional nebuloso “*muito próxima*”, com a distribuição de possibilidade do conjunto com o qual se compara. Desta forma, é obtido o conjunto:

$$A \circ R = \left\{ \max_{d \in \text{Altura}} (\mu_A(d) \wedge \mu_R(d, d')) / d' : d' \in \text{Altura} \right\} = \left\{ \max_{d \in \text{Altura}} (\mu_{173}(d) \wedge \mu_{\text{Muito\_Próxima}}(d, d')) / d' : d' \in \text{Altura} \right\} = \{173/1\}$$

Depois, são recuperadas as tuplas que satisfazem a consulta, agrupadas em dois conjuntos obtidos pelas medidas de possibilidade e necessidade. A medida de possibilidade esta expressada como:

$$\Pi(A \circ R | F) = \left\{ \sup_{d \in \text{Altura}} \{ \mu_{A \circ R}(d) \wedge \pi_F(d) \} \right\} = \{ \text{Luiz}/0.8, \text{Pedro}/0.8 \},$$

onde  $F$  representa os conjuntos nebulosos definidos sobre o domínio *Altura*.

### 3.4.2 Modelo de Zemankova-Kandel.

O modelo de Zemankova e Kandel [ZK85, Ze89], compõe-se de três partes: (a) uma base de dados de valores (VDB) onde se organizam os dados em forma similar aos outros modelos propostos, (b) uma base de dados explicativa onde estão as definições para os subconjuntos e relações nebulosas, e (c) um conjunto de regras de tradução.

Se constroem sobre domínios que são classificados nos seguintes tipos:

1. Escalares discretos,
2. Conjuntos de números finitos ou infinitos.
3. O intervalo unidade.

A recuperação na consulta é similar ao modelo de Prade-Testemale. A partir da expressão  $A \Theta F$ , a medida de possibilidade utilizada para procurar a compatibilidade do subconjunto  $F$  com o valor  $x$  do atributo  $A$  para cada tupla na relação esta expressada como:

$$p_A(F) = \sup_{u \in U} \{ \mu_F(u) \cdot \pi_{A(x)}(u) \}$$

e a medida de certeza

$$c_A(F) = \max_{u \in D} \{ 0, \inf \{ \mu_F(u) \cdot \pi_A(u) \} > 0 \}$$

é usada no lugar da medida de necessidade de Prade-Testemale. Esta medida é interpretada como o grau de certeza com o qual  $u$  pertence a  $F$  ou é descrito por  $F$ .

Na consulta os operadores relacionais  $\Theta$  são definidos a partir de uma relação de similaridade,  $s$ , sobre  $D \times D$ . Com estas relações se constroem os operadores.

Por exemplo, o operador *maior\_que* poderia definir-se como segue:

$$\mu_{\text{maior\_que}}(x, y) = \begin{cases} 1 - 0.5 \cdot s(x, y) & \text{se } x \geq y \\ 0.5 \cdot s(x, y) & \text{se } x < y \end{cases}$$

A partir desta definição a medida de possibilidade para condições do tipo  $A\Theta f$  se define como:

$$\Pi(\Theta(A(x), f)) = \sup_{d \in D} \{\mu_{\Theta}(d, f) \cdot \pi_A(d)\},$$

onde  $f$  é um crisp.

### 3.4.3 Modelo de Umano.

Umano [Um83] define um banco de dados nebulosos  $BD$ , como um conjunto de relações nebulosas  $R_j$  com  $j = 1, \dots, m$

$$BD = \{R_1, R_2, \dots, R_m\},$$

onde cada relação nebulosa é definida como:

$$R_j: P(D_1) \times P(D_2) \times \dots \times P(D_n) \rightarrow P([0, 1]) \quad j = 1, \dots, m,$$

$P(D_i)$  representa a família de todas distribuições de possibilidade sobre o domínio  $D_i$  e  $\times$  define o produto cartesiano. O termo  $P([0, 1])$  representa que o espaço de pertinência é estendido como uma coleção de todas as distribuições de possibilidade sobre  $[0, 1]$ .

A representação da relação *físico* segundo este modelo é apresentada na tabela 3.10.

Nome	Idade	Peso	Sexo	Altura	Cor_Pele	Cor_Cabelo	Certeza
Luiz	jovem	70	M	alto	branca	preto	1
Maria	30	Nulo	F	1.70	morena	{loiro/1, castanho/1}	1
Carlos	80	Unknown	M	baixo	preto	Undefined	0.8
Pedro	jovem	aprox. 80	M	alto	branca	loiro	1
Marta	mocinha	Unknown	F	mediana	branca	moreno	0.7

Tabela 3.10. Representação da relação *físico* no Modelo de Umano.

As operações da álgebra relacional são estendidas por Umano para o modelo relacional nebuloso de distribuições de possibilidades. Para isto, considere-se duas relações  $n$ -árias  $R_1$  e  $R_2$  no universo  $D = D_1 \times D_2 \times \dots \times D_n$  e sejam  $\Pi_{t_i}, \Pi_{t_j}$  distribuições de possibilidade definidas sobre as tuplas  $t_i$  e  $t_j$  das relações  $R_1$  e  $R_2$ .

A notação  $\{\dots\}_p$  é usada para representar a distribuição de possibilidade e  $\{\dots\}$  para representar conjuntos nebulosos.

As operações da álgebra se definem em seguida:

União ( $R_1 \cup R_2$ )

$$\Pi_{t_i} \vee \Pi_{t_j} = \Pi_{t_i \cup t_j} = \left\{ \pi_{t_i}(v_1) \wedge \pi_{t_j}(v_2) / v_1 \vee v_2 : v_1, v_2 \in [0, 1] \right\}_p,$$

onde  $\vee$  denota o máximo e  $\wedge$  denota o mínimo.

Interseção ( $R_1 \cap R_2$ )

$$\Pi_{t_i} \wedge \Pi_{t_j} = \left\{ \pi_{t_i}(v_1) \wedge \pi_{t_j}(v_2) / v_1 \wedge v_2 : v_1 \in [0, 1] \right\}_p$$

### Diferença ( $R_1 - R_2$ )

$$\Pi_{t_i} - \Pi_{t_j} = \left\{ \pi_{t_i}(v_1) \wedge \pi_{t_j}(v_2) / v_1 - v_2 : v_1 \in [0,1] \right\} p,$$

onde - denota a subtração convencional ou diferença-limitada

### Produto Cartesiano Estendido ( $R_1 \times R_2$ )

Suponha-se  $R_1$  definida como uma relação m-ária em  $U = U_1 \times U_2 \times \dots \times U_m$  e  $\Pi_{t_i}$  a distribuição de possibilidade definida sobre a tupla  $t_i$ . O produto cartesiano é uma relação  $R_3$  (m+n)-ária onde a distribuição de possibilidade  $\Pi_{t_m}$  sobre a tupla  $t_m$  é obtida da seguinte forma:

$$\Pi_{t_m} = \Pi_{t_i} \times \Pi_{t_j} = \left\{ \pi_{t_i}(v_1) \wedge \pi_{t_j}(v_2) / v_1 \wedge v_2 : v_1, v_2 \in [0,1] \right\}$$

### Projeção ( $\pi_{A_{i_1}, A_{i_2}, \dots, A_{i_k}}(R_1)$ )

Seja  $R_1$  formada pelos atributos  $d_1, d_2, \dots, d_n$  e  $t(d_{i_1}, d_{i_2}, \dots, d_{i_k})$  uma tupla que contem somente os valores dos atributos  $d_{i_1}, d_{i_2}, \dots, d_{i_k}$ . A projeção da relação nebulosa  $R_1$  sobre os atributos  $d_{i_1}, d_{i_2}, \dots, d_{i_k}$  é a relação  $R_3$  definida como

$$\Pi_{t_m} = \Pi_{t_i}$$

isto é, o valor da distribuição de possibilidade sobre as tuplas da nova relação, é igual ao valor das respectivas tuplas da relação  $R_1$ . No caso que existam tuplas repetidas, o novo valor de distribuição de possibilidade é calculado a partir da união das tuplas.

### Seleção

A seleção sobre uma relação nebulosa  $R(A_1, A_2, \dots, A_n)$ , a partir de uma predicado  $\theta$  com respeito a os atributo  $A_{i_1}, A_{i_2}, \dots, A_{i_k}$  é definido como:

$$R[\theta(A_{i_1}, A_{i_2}, \dots, A_{i_k})] = \left\{ \mu/r : \mu_R(r) / r \in R, \mu = t_\theta(r(A_{i_1}), r(A_{i_2}), \dots, r(A_{i_k})) \wedge \mu_R(r) \right\}$$

onde  $t_\theta$  é a função do valor verdade do predicado nebuloso  $\theta$ .

O valor verdade do predicado nebuloso é definido a seguir.

Seja  $\Pi_{x_i}$  uma distribuição de possibilidade no Universo  $U_i$ . Então, o valor verdade do predicado nebuloso k-ário  $\theta(x_1, x_2, \dots, x_k)$ , caracterizado pela função verdade  $t_\theta(u_1, u_2, \dots, u_k)$ , é definido como uma distribuição de possibilidade:

$$T(\theta(\Pi_{x_1}, \Pi_{x_2}, \dots, \Pi_{x_k})) = \left\{ \frac{\pi_{x_1}(u_1) \wedge \dots \wedge \pi_{x_k}(u_k)}{t_\theta(u_1, u_2, \dots, u_k)} \right\}_P \\ u_i \in U_i \quad (i = 1, 2, \dots, k)$$

As outras operações da álgebra são definidas da mesma forma, a partir das distribuições de possibilidades.

Para resolver a consulta “*Obter as pessoas com altura muito próxima 1.73m*”, deve-se definir como primeiro passo o operador relacional de comparação nebuloso, sendo assumida a definição feita na seção 3.4.1.

Para calcular a distribuição de possibilidade sobre cada tupla da relação procede-se com segue:

$$\Pi_{luiz} = t_\theta(alto) \wedge 1 = t_\theta(alto) = \left\{ alto / \mu_{muito\_próxima}(173, d) \right\}_P = \\ \left\{ \frac{d - 1.65}{0.1} / \mu_{muito\_próxima}(173, d) \right\}_P \quad d \in [1.71, 1.75]$$

$$\Pi_{Pedro} = t_\theta(alto) \wedge 1 = t_\theta(alto) = \left\{ alto / \mu_{muito\_próxima}(173, d) \right\}_P = \\ \left\{ \frac{d - 1.65}{0.1} / \mu_{muito\_próxima}(173, d) \right\}_P \quad d \in [1.71, 1.75]$$

$$\Pi_{Maria} = t_\theta(170) \wedge 1 = t_\theta(170) = \left\{ 1 / \mu_{muito\_próxima}(173, 170) \right\}_P = 0$$

$$\Pi_{Carlos} = t_\theta(baixo) \wedge 0.8 = \left\{ baixo / \mu_{muito\_próxima}(173, d) \right\}_P \wedge 0.8 = 0 \wedge 0.8 = 0$$

$$\Pi_{Marta} = t_{\theta}(\text{mediana}) \wedge 0.7 = \left\{ \text{mediana} / \mu_{\text{muito\_próxima}}(173, d) \right\} p \wedge 0.7 = 0 \wedge 0.7 = 0$$

A relação resposta é a relação da tabela 3.11.

Nome	Altura	Certeza
Luiz	alto	$\left\{ \frac{d - 165}{0.1} / \mu_{\text{muito\_}}(173, d) \right\} p \quad d \in [171, 175]$
Pedro	alto	$\left\{ \frac{d - 165}{0.1} / \mu_{\text{muito\_}}(173, d) \right\} p \quad d \in [171, 175]$

Tabela 3.11. Relação resposta à consulta.

### 3.4.4 Análise Crítica.

Os modelo relacionais possibilísticos, a partir da distribuição de possibilidade, permitem capturar o significado de diversos conceitos nebulosos definidos sobre domínios numéricos. Mas, por outro lado, não fazem um bom tratamento da incerteza nos atributos nebulosos qualitativos, definidos sobre domínios não numéricos.

O uso de distribuições de possibilidade permite o tratamento da informação imprecisa a nível dos atributos nos bancos de dados. Estes modelo permite também, representar e manipular todos os tipos de informação incompleta: a informação desconhecida e a indefinida.

### 3.5 Comparação entre as abordagens.

As abordagens que foram apresentadas, tratam de uma forma ou outra, os tipos de incerteza e imprecisão mais encontrados na linguagem natural. Contudo, possuem como principal problema não incluírem no modelo um espaço para o armazenamento de todos os tipos de imprecisão. A tabela 3.12 apresenta alguns pontos que diferenciam os três modelos de bancos de dados nebulosos.

Modelos/Asp. Comparativos	Buckles & Petry	Modelo Relacional Nebuloso	Modelo Relacional Possibilístico
Informação Incompleta e nula	A informação incompleta recebe um único tratamento. Não são diferenciados os tipos de valores nulos.	Não inclui mecanismos para tratar e representar este tipo de incerteza.	Existem dois tratamentos principais. O tratamento de Prade-Testamale é mais completo por tratar um tipo a mais de informação nula.
Incerteza em domínios qualitativos não numéricos.	É tratada com o uso de relações de similaridade entre os elementos do domínio.	Não inclui mecanismos para tratar e representar este tipo de incerteza.	Só o modelo de Zemankova-Kandel que definem relações de similaridade e proximidade entre os elementos deste tipo de domínios para a consulta. O resto dos modelos não incluem nenhuma consideração.
Incerteza na informação geral	Calcula esta incerteza na consulta, mas não inclui um marco para representa-la como parte do modelo.	Inclui este tipo de incerteza e introduz mecanismos para a sua manipulação.	Só o Umano no seu modelo apresenta os mecanismos para a representação deste tipo de incerteza e introduz mecanismos para sua manipulação.

Tabela 3.12. Comparação entre as abordagens.

# Capítulo 4

## *O Modelo Integrado de Bancos de Dados Nebulosos*

---

O *Modelo Integrado de Bancos de Dados Nebulosos* que toma como base para sua definição o Modelo generalizado de bancos de dados nebulosos ou GEFRED (A Generalized model for Fuzzy Relational Databases) é apresentado neste capítulo. Nas primeiras seções é analisado o GEFRED e é comparado com as outras abordagens apresentadas no capítulo 3. Nas seções que seguem é exposto o novo modelo com a definição da álgebra para a manipulações da informação e a linguagem de interrogação.

### **4.1 GEFRED. Modelo generalizado de bancos de dados nebulosos.**

Antes de introduzir o *Modelo Integrado de Bancos de Dados Nebulosos* será feita uma análise sobre o modelo GEFRED ou Modelo Generalizado de Banco de Dados Nebulosos porque apresenta muitas características desejáveis, discutidas no capítulo anterior e serviu como base para o desenvolvimento de uma proposta de modelo, que é o objetivo deste trabalho

O modelo GEFRED é proposto por Medina [Me94, Me94a, Me95]. Nele são incluídos aspectos positivos das abordagens discutidas no capítulo anterior e são eliminados os inconvenientes e as limitações que apresentavam. Nas primeiras seções deste capítulo o GEFRED é apresentado de forma geral e nas seções que seguem é comparado com os enfoques anteriores.

#### 4.1.1 Domínio Nebuloso Generalizado.

O objetivo do modelo é proporcionar uma ampla cobertura, no que se refere à representação e tratamento da informação nebulosa de todo tipo. A tabela 4.1 mostra todos os tipos de dados representáveis neste modelo.

Um domínio nebuloso generalizado,  $D_G$ , representa o elemento sobre o qual se definem os valores da tabela 4.1.

**Definição:** Seja  $D$  um domínio de discurso,  $\tilde{P}(D)$  o conjunto de todas possíveis distribuições de possibilidade sobre  $D$ , incluídas as que definem os tipos *Unknown* e *Undefined* e seja *Null* o elemento definido na tabela 4.1, chamaremos **Domínio Nebuloso Generalizado**,  $D_G$ , a  $D_G \subseteq \tilde{P}(D) \cup \text{Null}$ .

#### 4.1.2 Relação Nebulosa Generalizada.

Define-se como *Relação Nebulosa Generalizada*,  $R_{FG}$ , um par de conjuntos cabeçalho, (H), e corpo, (B),  $R_{FG} = (H, B)$ , definidos da seguinte maneira:

- A *cabeçalho* é um conjunto fixo de ternas *atributo-domínio-compatibilidade*,

$$H = \left\{ \left( A_{G_1} : D_{G_1}, [C_{A_{G_1}}] \right), \left( A_{G_2} : D_{G_2}, [C_{A_{G_2}}] \right), \dots, \left( A_{G_n} : D_{G_n}, [C_{A_{G_n}}] \right) \right\},$$

onde para cada atributo  $A_{G_j}$  existe um domínio não necessariamente distinto,  $D_{G_j}$  com  $j = 1, \dots, n$ , e  $C_{A_{G_j}}$  corresponde a um atributo de compatibilidade definido no intervalo  $[0, 1]$ .

- O *corpo* é um conjunto de tuplas nebulosas generalizadas distintas, onde cada uma delas é composta por ternas *atributo-valor-grau*,

$$B = \left\{ \left( A_{G_1} : \tilde{d}_{i_1}, [C_{i_1}] \right), \left( A_{G_2} : \tilde{d}_{i_2}, [C_{i_2}] \right), \dots, \left( A_{G_n} : \tilde{d}_{i_n}, [C_{i_n}] \right) \right\},$$

onde  $\tilde{d}_{ij}$  representa o valor de domínio que toma o atributo  $A_{G_j}$  na tupla  $i$  e  $C_{ij}$  o grau de compatibilidade associado a este valor ( $i = 1, \dots, m$ , onde  $m$  representa o número de tuplas da relação).

1. Um escalar simples (Ex. <i>Atitude</i> = <i>boa</i> , representado pela distribuição de possibilidade $1/boa$ ).
2. Um número simples (Ex. <i>Idade</i> = 28, representado pela distribuição de possibilidade $1/28$ ).
3. Um conjunto de possíveis valores de escalares ( Ex. <i>Atitude</i> = { <i>boa</i> , <i>ruim</i> }, expressada $\{1/boa, 1/ruim\}$ ).
4. Um conjunto de possíveis valores de números (Ex. <i>Idade</i> = {20, 21}, expressada $\{1/20, 1/21\}$ ).
5. Uma distribuição de possibilidade no domínio dos escalares (Ex. <i>Atitude</i> ={0.2/ <i>boa</i> , 0.8/ <i>ruim</i> }).
6. Uma distribuição de possibilidade no domínio dos números (Ex. <i>Idade</i> ={0.4/23, 0.6/25, 0.8/27}).
7. Um número real representando grau de satisfação (Ex. <i>Qualidade</i> =0.9).
8. Um valor indefinido <i>Undefined</i> representado pela distribuição de possibilidade, $Undefined = \{0/d : d \in D\}$ .
9. Um valor desconhecido <i>Unknown</i> representado pela distribuição de possibilidade, $Unknown = \{1/d : d \in D\}$ .
10. Um valor <i>Null</i> representado pela distribuição de possibilidade $Null = \{1/Unknown, 1/Undefined\}$ .

Tabela 4.1. Tipos de dados representados pelo modelo.

A partir da definição de *Relação Nebulosa Generalizada*, é introduzida a definição de *Chave Primaria Generalizada*, que garante a acessibilidade a cada uma das tuplas da relação:

**Definição:** Seja  $R_{FG}$  uma *Relação Nebulosa Generalizada* representada por:

$$R_{FG} = \begin{cases} H & = \left\{ \left( A_{G_1} : D_{G_1} [ , C_{A_{G_1}} ] \right), \dots, \left( A_{G_n} : D_{G_n} [ , C_{A_{G_n}} ] \right) \right\} \\ B & = \left\{ \left( A_{G_1} : \tilde{d}_{r_1} [ , c_{r_1} ] \right), \dots, \left( A_{G_n} : \tilde{d}_{r_n} [ , c_{r_n} ] \right) \right\} \end{cases}$$

onde  $r = 1, \dots, m$ , sendo  $m$  o número de tuplas da relação. Define-se *Chave Primaria Generalizada*,  $K_G$ , a um subconjunto de  $H$  expressado como:

$$K_G \subseteq H, K_G = \left\{ \left( A_{G_s} : D_{G_s} \right) \mid s \in S \subseteq \{1, \dots, n\} \right\}$$

que cumpre:

1.  $\forall s \in S, D_{G_s}$  é um domínio numérico o escalar simples, tipos 1 e 2 da tabela 4.1.
2.  $\forall i, i' \in \{1, \dots, m\}, \exists s \in S: \left( A_{G_s} : d_{i_s} \right) \neq \left( A_{G_s} : d_{i'_s} \right)$ , onde o operador  $\neq$  representa a desigualdade clássica.

### 4.1.3 Álgebra Nebulosa Generalizada.

O modelo GEFRED contempla a manipulação das relações  $R_{FG}$ , mediante a utilização da *Álgebra Relacional Nebulosa Generalizada*. As operações clássicas, *União, Interseção, Diferença, Produto Cartesiano, Projeção, Seleção e Junção*, são estendidas a fim de que possam operar sobre as  $R_{FG}$  em forma coerente.

A operações de *Seleção* e *Junção* baseiam o seu funcionamento no emprego de operadores relacionais, construídos com a utilização das medidas de compatibilidade encontradas na literatura.

O modelo faz uma definição geral, para os operadores de comparação, a partir do significado das operações definidas sobre o domínio de discurso, estendendo-as para o tratamento de valores do *Domínio Nebuloso Generalizado*.

As operações de *União, Interseção* e *Diferença* possuem como restrição para sua aplicação, a compatibilidade com respeito à união de duas *Relações Nebulosas Generalizada*:

**Definição:** Duas *Relações Nebulosas Generalizadas*  $R_{FG}$  e  $R'_{FG}$  se dizem *compatíveis respeito à união* se são compatíveis (no sentido clássico) os respectivos domínios dos seus atributos.

A seguir, define-se somente a *União nebulosa Generalizada*,  $\cup_G$ , já que as outras operações são definidas similarmente.

**Definição:** Sejam  $R_{FG}$  e  $R'_{FG}$  duas *Relações Nebulosas Generalizadas* representadas por:

$$R_{FG} = \begin{cases} H & = \left\{ \left( A_{G_1} : D_{G_1} [ , C_{A_{G_1}} ] \right), \dots, \left( A_{G_n} : D_{G_n} [ , C_{A_{G_n}} ] \right) \right\} \\ B & = \left\{ \left( A_{G_1} : \tilde{d}_{i_1} [ , c_{i_1} ] \right), \dots, \left( A_{G_n} : \tilde{d}_{i_n} [ , c_{i_n} ] \right) \right\} \end{cases}$$

$$R'_{FG} = \begin{cases} H' & = \left\{ \left( A'_{G_1} : D_{G_1} [ , C_{A'_{G_1}} ] \right), \dots, \left( A'_{G_n} : D_{G_n} [ , C_{A'_{G_n}} ] \right) \right\} \\ B' & = \left\{ \left( A'_{G_1} : \tilde{d}'_{k_1} [ , c'_{k_1} ] \right), \dots, \left( A'_{G_n} : \tilde{d}'_{k_n} [ , c'_{k_n} ] \right) \right\} \end{cases}$$

com  $i = 1, \dots, m$  e  $k = 1, \dots, m'$ , sendo  $m$  e  $m'$  as respectivas cardinalidade, então, a *União Nebulosa Generalizada* de  $R_{FG}$  e  $R'_{FG}$  está definida por:

$$R_{FG} \cup R'_{FG} = \begin{cases} H_{\cup_G} & = \left\{ \left( A_{G_l} : D_{G_l} [ , C_{A_{G_l}} ] \right), \dots, \left( A_{G_n} : D_{G_n} [ , C_{A_{G_n}} ] \right) \right\} \\ B_{\cup_G} & = \begin{cases} B_{\cup_G}^v & = B^v \cup B'^v \\ B_{\cup_G}^c & = \left\{ [c_{l_j}^v], \dots, [c_{l_n}^v] \right\} \end{cases} \end{cases} \quad l = 1, \dots, m''$$

onde  $m''$  é a cardinalidade da união e

$$c_{l_j}'' = \begin{cases} \max\{c_{l_j}, c'_{l_j}\} & \text{se } \exists c_{l_j} \text{ e } \exists c'_{l_j} \\ c_{l_j} & \text{se } \exists c_{l_j} \text{ e } \nexists c'_{l_j} \\ c'_{l_j} & \text{se } \exists c'_{l_j} \text{ e } \nexists c_{l_j} \\ 0 & \text{se } \nexists c_{l_j}, \nexists c'_{l_j} \text{ e, } \exists C_{l_j} \text{ ou } \exists C'_{l_j} \end{cases}$$

As demais operações da *Álgebra Relacional Nebulosa* são definidas similarmente.

#### 4.1.4 Estudo do GEFRED comparado com outras abordagens.

Para mostrar a capacidade do GEFRED, esta seção é dedicada à comparação deste modelo com as outras abordagens anteriormente apresentadas. A análise está encaminhada no estudo dos aspectos diferentes de cada abordagem com relação ao GEFRED.

##### 4.1.4.1 GEFRED e o Modelo Relacional Nebuloso.

O GEFRED não contém o modelo Relacional Nebuloso na sua definição. A incerteza ao nível de tupla refere-se à ambigüidade entre os valores. Esta pode ser resolvida com um grau de pertinência da tupla à relação.

##### 4.1.4.2 GEFRED e o Modelo de unificação a partir de relações de similaridade.

O modelo de unificação a partir de relações de similaridade trabalha sobre conjuntos finito de escalares ou números ao igual que o GEFRED. Mas existe um aspecto da representação que não é incluído nos mesmos termos: a interpretação não excludente do conceito de conjunto para o valor de um atributo, frente a excludente utilizada no modelo GEFRED. Este aspecto refere-se a que no modelo de Buckles-Petry para representar a expressão “*Luíz tem dois filhos: Pedro e Raul*” se utiliza somente uma tupla (isto é, o atributo *filho = { Pedro, Raul }*), informação esta representada no GEFRED em duas tuplas. Esta forma de representação ( Buckles-Petry ) vai contra ao princípio de atomicidade da informação exigida no modelo de bancos de dados relacionais convencionais.

O modelo de Buckles-Petry define o conceito de relação nebulosa como um subconjunto do produto cartesiano  $2^{D_1} \times 2^{D_2} \times \dots \times 2^{D_m}$ , onde  $2^{D_j}$  é um subconjunto do domínio de discurso  $D_j$ . A partir desta definição, com a diferença anteriormente citada, GEFRED pode incluir o conceito de relação nebulosa como um caso particular da definição de *Relação Nebulosa Generalizada*.

Para o domínios de escalares, o GEFRED utiliza a definição de uma função de similaridade como a definida por Buckles-Petry, e assim é relaxado o conceito de igualdade entre os valores do domínio. Igualmente, o GEFRED, permite estabelecer limiares de consulta para cada atributo envolvido na mesma. Mas GEFRED vai mais longe, ao poder armazenar o grau com que cada atributo de cada tupla satisfaz a consulta dentro do seu modelo.

Em geral o GEFRED absorve os aspectos positivos do modelo de Buckles-Petry e proporciona mecanismos alternativos para resolver alguns dos problemas que o modelo de unificação a partir de relações de similaridade apresenta.

#### **4.1.4.3 GEFRED e o Modelo Relacional Possibilístico.**

Nas seções que seguem, o GEFRED é comparado com os três modelo relacionais possibilísticos. Cada modelo é comparado do ponto de vista do GEFRED, e são analisadas as principais diferenças que comprovam as vantagens do GEFRED sobre estes modelos.

##### **4.1.4.3.1 GEFRED e o Modelo de Prade-Testemale.**

Os aspectos diferenciadores do modelo de Prade-Testemale estão nas medidas de necessidade e possibilidade utilizadas para obter diferentes respostas, a uma condição, em uma consulta feita.

No modelo de Prade-Testemale uma relação é definida como o produto cartesiano dos domínios de discurso que são distribuições de possibilidade sobre  $D \cup \{\varepsilon\}$ . Com esta definição, é permitida a definição de um tipo de valor que o GEFRED não inclui. Mas, com esta exceção a definição de relação, é representável pelo GEFRED, assim como os tipos de valores com os que o modelo de Prade-Testemale trabalha.

Os operadores relacionais utilizados no modelo de Prade são representáveis pelo *comparador estendido* utilizado por GEFRED no seu modelo.

Nas suas representações, as distribuições de possibilidade devem se normalizadas, aspecto que pode não ser sempre alcançado em conjunto de escalares. No GEFRED, esta exigência é eliminada.

#### **4.1.4.3.2 GEFRED e o Modelo de Zemankova-Kandel.**

Os aspectos diferenciadores mais importantes residem na utilização das medidas para avaliar uma determinada condição,  $F$ , com os valores do domínio das tuplas

O modelo de Zemankova-Kandel se constroi sobre domínios classificados nos seguintes tipos: Escalares discretos, conjuntos de números finitos ou infinitos, o intervalo unidade. Sobre eles, são definidas distribuições de possibilidade com as que opera. A partir desta análises podemos concluir que toda a informação tratada pelo modelo é tratada pelo GEFRED através do domínio nebuloso generalizado.

Uma relação no modelo de Zemankova-Kandel é o produto cartesiano dos conjunto de todas as distribuições de possibilidade definidas sobre os tipos de domínios antes mencionados. Esta definição é incluída na definição adotada pelo GEFRED.

Igualmente que o GEFRED, este modelo exige que alguns dos atributos tenham informação precisa para poder definir chave primaria.

#### **4.1.4.3.3 GEFRED e o Modelo de Umano.**

A partir da análises de ambas as duas abordagens sugerem-se os seguintes pontos conclusivos:

1. O domínio considerado no Modelo de Umano é um subconjunto do considerado no GEFRED. A relação definida também pode-se formular nos termos do GEFRED.
2. O modelo do Umano inclui a incerteza ao nível de tuplas, tratamento que não faz o GEFRED.

## 4.2 O Modelo Integrado de Bancos de Dados Nebulosos.

Considerando-se o conteúdo da tabela 3.12 onde pode-se notar as diferenças entre as abordagens mais conhecidas e que, nenhuma delas detêm recursos para acolher todo tipo de incerteza; considerando-se também que o modelo GEFRED já apresenta soluções parciais para estas situações não cobertas, procura-se propor um modelo mais atual e abrangente. Este modelo que apresentado nas seguintes seções deste capítulo.

### 4.2.1 Domínio Integrado Nebuloso.

**Definição:** Seja  $D$  um domínio de discurso e seja  $Undefined$  o elemento que descreve o caso de atributo não aplicável sobre o objeto  $x$ . Seja  $\tilde{P}(D \cup \{Undefined\})$  o conjunto de todas as possíveis distribuições de possibilidade sobre  $D \cup \{Undefined\}$ , define-se *Domínio Integrado Nebuloso*,  $D_G$ , a  $D_G \subseteq \tilde{P}(D \cup \{Undefined\})$ .

A tabela a 4.2 representa as modificações do novo modelo na representação de alguns dos valores que podem incluir um domínio. Esta representação é consequência direta da definição de *Domínio Integrado Nebuloso*.

<p>1. Um valor indefinido <i>Undefined</i>, definido sobre qualquer tipo de domínio, é representado pela distribuição de possibilidade,</p> $Undefined = \left\{ \frac{1}{undefined}, \frac{0}{d} : d \in D \right\}.$ <p>O valor do atributo para essa tupla está indefinido, só o valor do domínio <i>Undefined</i> é aplicável. Suponha no banco de dados o atributo cor do cabelo, mais se a pessoa é careca o valor do atributo estará indefinido. No caso será <math>cor\_cabelo = Undefined</math></p>
<p>2. Um valor desconhecido <i>Unknown</i> é representado pela distribuição de possibilidade,</p> $Unknown = \left\{ \frac{0}{Undefined}, \frac{1}{d} : d \in D \right\}.$ <p>Representa o desconhecimento sobre os valores que toma o atributo, mas é conhecida a sua definição. Pode estar definido sobre domínios discretos ou domínios contínuos ordenados. No valor do atributo altura ponderai-se dizer: <math>altura = Unknown</math>, isto é, a altura da pessoa é desconhecida</p>
<p>3. Um valor <i>Null</i> é representado pela distribuição de possibilidade</p> $Null = \left\{ \frac{1}{d} : d \in D \cup \{Undefined\} \right\}.$ <p>É assinalado esse valor por dois motivos: (1) é desconhecido se existe o valor (<i>Unknown</i>), ou (2) porque é desconhecido se é aplicável o atributo a essa tupla (<i>Undefined</i>). Pode estar definido sobre domínios discretos ou domínios contínuos ordenados. Por exemplo para o mesmo atributo do exemplo anterior, suponha que se tem a seguinte informação: não se sabe, se a pessoa é careca ; e no caso que não seja, é desconhecido a cor do cabelo. No banco de dados ficaria <math>cor\_cabelo = Null</math>.</p>
<p>4. O valor <i>distribuição de possibilidade sobre <math>D \cup \{Undefined\}</math></i> está definido sobre qualquer tipo de domínio. Este tipo de valor representa o tipo de informação quando existe uma possibilidade maior que 0 de que o atributo não é aplicável. O domínio pode ser numérico ou escalar sobre os que existem distribuições de possibilidade ou relações de proximidade segundo o caso. Por exemplo suponha que existe uma possibilidade de 0,6 que a pessoa seja careca, mas se não é, a cor do cabelo é branco. A distribuição de possibilidade ficaria expressada desta maneira:</p> $\left\{ \frac{0,6}{Undefined}, \frac{1}{Branco} \right\}.$

Tabela 4.2. Nova representação proposta.

Observações:

1. No modelo proposto por Medina, a abordagem utilizada para a representação dos tipos de dados coincide com a de Umano, na nossa representação a abordagem é a proposta por Prade-Testemale pelo fato de que pode ser representado o tipo de incerteza do item 11, informação não representável com a outra abordagem.
2. O item 11 representa o tipo de informação onde existe possibilidade de indefinição e no caso de definição existe uma distribuição de possibilidade definida sobre o conjunto  $D$ .

#### 4.2.2 Relação Integrada Nebulosa.

A definição de *Relação Nebulosa Generalizada* é igualmente estendida para introduzir a incerteza ao nível de tuplas no modelo. Com isto é apresentado o conceito de *Relação Integrada Nebulosa* que constitui o elemento principal através do qual é organizada a informação em todas as suas formas.

Define-se uma *Relação Integrada Nebulosa*,  $R_N$ , como um par de conjuntos cabeçalho, (H), e corpo, (B),  $R_N = (H, B)$ , definidos da seguinte maneira:

- A *cabeçalho* é um conjunto fixo de ternas *atributo-domínio-compatibilidade* mais o atributo de compatibilidade da tupla:

$$H = \left\{ \left( A_{N_1} : D_{N_1} [C_{A_{N_1}}] \right), \left( A_{N_2} : D_{N_2} [C_{A_{N_2}}] \right), \dots, \left( A_{N_n} : D_{N_n} [C_{A_{N_n}}] \right), C_N \right\}$$

onde para cada atributo  $A_{N_j}$  existe um domínio não necessariamente distinto,  $D_{N_j}$  com  $j = 1, \dots, n$ ,  $C_{A_{N_j}}$  e  $C_N$  correspondem com atributos de compatibilidade definidos no intervalo  $[0,1]$ .

- O *corpo* é um conjunto de tuplas nebulosas distintas onde cada uma delas esta compostas pelas ternas *atributo-valor-grau*, mais o atributo de compatibilidade da tupla.

$$B = \left\{ \left( A_{N_1} : \tilde{d}_{i_1} [ , C_{i_1} ] \right), \left( A_{N_2} : \tilde{d}_{i_2} [ , C_{i_2} ] \right), \dots, \left( A_{N_n} : \tilde{d}_{i_n} [ , C_{i_n} ] \right), C_i \right\}$$

onde  $\tilde{d}_{i_j}$  representa o valor de domínio que toma o atributo  $A_{N_j}$  na tupla  $i$ ,  $C_{i_j}$  o grau de compatibilidade associado a este valor ( $i = 1, \dots, m$ ,  $m$  é o número de tuplas da relação). O termo  $C_i$  representa o atributo de compatibilidade da tupla  $i$  com a relação.

Observações:

1. No modelo de Medina não é tratada a incerteza ao nível de tupla, só a nível de atributos, no modelo proposto são tratados aos dois níveis. Este atributo permite que nas consultas ao banco de dados possa ser definido um limiar de interrogação para a tupla, permitindo um refinamento iterativo da solução.
2. Esta definição contém o modelo de bancos de dados convencionais como caso particular assim como o resto dos modelo definidos para o tratamento da informação nebulosa.
3. O atributo de compatibilidade da tupla à relação é de definição obrigatória, no caso que não está explicitamente colocado, tem valor 1, significando a total certeza na informação representada pela tupla.
4. O *Modelo Integrado de Bancos de Dados Nebulosos*, permite a definição de relações de proximidade entre os elementos de domínios de escalares [SM89]. As relações de proximidade não exigem a propriedade de transitividade exigida nas relações de similaridade.

É importante em todo modelo de Banco de Dados poder reconhecer as tuplas redundantes, com isto, a seguir é apresentado o conceito de *tuplas redundantes* a partir da definição de uma *Relação Integrada Nebulosa*:

Sejam  $t_i = (a_{i1}, a_{i2}, \dots, a_{in})$  e  $t_j = (a_{j1}, a_{j2}, \dots, a_{jn})$ , onde  $i \neq j$ , duas tuplas de uma relação  $R_N$ . As tuplas  $t_i$  e  $t_j$  são redundantes e a tupla  $t_i$  é eliminada se  $a_{im} \subseteq a_{jm}$ ,  $m = 1, \dots, n$ . Os atributos de compatibilidade da tupla  $t_j$  tomarão o valor máximo dos respectivos atributos das tuplas comparadas.

### 4.2.3 Atributos com valores definidos como conceitos complexos.

Em bancos de dados convencionais existe um tipo de relação que não necessariamente existe em uma tabela física, isto é uma tabela virtual. Esta relação é chamada de visão [EN94]. Um visão é um tabela obtida como resposta da seleção de tuplas e atributos de tabelas físicas o outras visões a partir de uma condição. O novo modelo não trata este tipo de relações, no entanto, algumas facilidades deste tipo são introduzidas através dos **conceitos complexos**.

Maia em [Ma81] faz uma análise dos conceitos complexos que a seguir serão definidos.

**Conceitos complexos:** Define-se como uma expressão lógica nebulosa

$$V_1 \text{ op } V_2 \text{ op } \dots \text{ op } V_n,$$

onde os  $V_i$  representam fatores e  $op$  representa um operador nebuloso de conjunto de disjunção ou conjunção. Cada fator  $V_i$  está definido como segue :

$$\text{exp} / \text{not} \langle \text{exp} \rangle,$$

onde o termo  $\langle \text{exp} \rangle$  esta definido

$$\langle \text{atributo} \rangle \langle \text{operador} \rangle \langle \text{constante} \rangle.$$

O termo  $\langle \text{constante} \rangle$  define uma constante que pertence ao domínio de definição do atributo. O termo  $\langle \text{operador} \rangle$  representa os operadores relacionais nebulosos de comparação:

$$\langle =, \neq, <, \leq, \geq, >, \approx \rangle.$$

Os atributos que formam a expressão nebulosa estão definidos em uma relação que pertence ao banco de dados nebulosos com que se está trabalhando.

A partir da definição deste conceito se define um tipo de valor que contém no seu domínio de definição conceitos complexos. Do ponto de vista lógico este atributo pode ser tratado como outro dos tipos de atributos apresentados, embora não possam ser modificados seus valores diretamente. Isto é, o atributo é virtual. Como o atributo é definido a partir de uma expressão o valor é calculado cada vez que precisar. Este tipo de atributo pode ser construído em banco de dados convencionais a partir do conceito de visão.

Com a introdução deste conceito o modelo trata um novo tipo de valor e é ampliado ao seu domínio de discurso:

5. Um valor definido a partir de um conceito complexo.

#### 4.2.4 Álgebra Integrada Nebulosa.

Nesta seção são apresentadas as operações da *Álgebra Integrada Nebulosa* para a manipulação dos dados do novo modelo. Todas as operações são estendidas segundo o nosso objetivo.

Para apresentar as operações de União, Interseção e Diferença é preciso definir o conceito de compatibilidade com respeito à União. Considerando-se a similaridade entre este modelo e o de Medina é empregado o mesmo conceito de compatibilidade.

A seguir, são definidas algumas operações da *Álgebra Integrada Nebulosa*, para exemplificar como é a manipulação da informação no modelo.

A *União Integrada Nebulosa*,  $\cup_N$ , é definida como segue:

**Definição:** Sejam  $R_N$  e  $R'_N$  duas *Relações Integradas Nebulosas* representadas por:

$$R_N = \begin{cases} H & = \left\{ \left( A_{N_1} : D_{N_1} [C_{A_{N_1}}] \right), \dots, \left( A_{N_n} : D_{N_n} [C_{A_{N_n}}] \right), C_N \right\}, \quad i = 1, \dots, m \\ B & = \left\{ \left( A_{N_1} : \tilde{d}_{i_1} [c_{i_1}] \right), \dots, \left( A_{N_n} : \tilde{d}_{i_n} [c_{i_n}] \right), c_i \right\} \end{cases}$$

$$R'_N = \begin{cases} H' & = \left\{ \left( A'_{N_1} : D_{N_1} [C_{A'_{N_1}}] \right), \dots, \left( A'_{N_n} : D_{N_n} [C_{A'_{N_n}}] \right), C_N \right\}, \quad i = 1, \dots, m' \\ B' & = \left\{ \left( A'_{N_1} : \tilde{d}'_{k_1} [c'_{k_1}] \right), \dots, \left( A'_{N_n} : \tilde{d}'_{k_n} [c'_{k_n}] \right), c_i \right\} \end{cases}$$

onde  $m$  e  $m'$  são as respectivas cardinalidade, então, a *União Integrada Nebulosa* de  $R_N$  e  $R'_N$  está definida por:

$$R_N \cup R'_N = \begin{cases} H_{\cup_N} & = \left\{ \left( A_{N_1} : D_{N_1} [C_{A_{N_1}}] \right), \dots, \left( A_{N_n} : D_{N_n} [C_{A_{N_n}}] \right), C_N \right\} \\ B_{\cup_N} & = \begin{cases} B_{\cup_N}^v & = B^v \cup B'^v \\ B_{\cup_N}^c & = \left\{ [c''_{i_1}], \dots, [c''_{i_n}], c_i \right\} \end{cases}, \quad i = 1, \dots, m'' \end{cases}$$

onde  $m''$  é a cardinalidade da união e os atributos de compatibilidade são definidos como

$$c''_{i_j} = \begin{cases} s(c_{i_j}, c'_{i_j}) & \text{se } \exists c_{i_j} \text{ e } \exists c'_{i_j} \\ c_{i_j} & \text{se } \exists c_{i_j} \text{ e } \nexists c'_{i_j} \\ c'_{i_j} & \text{se } \exists c'_{i_j} \text{ e } \nexists c_{i_j} \end{cases}$$

$$c''_i = s(c_{i_1}, c'_{i_1})$$

o operador  $s$  representa uma  $s$ -norma qualquer.

O *Produto Cartesiano Integrado Nebuloso*,  $\times_N$ , é definido como segue

**Definição:** Sejam  $R_N$  e  $R'_N$  duas *Relações Integradas Nebulosas* representadas por:

$$R_N = \begin{cases} H & = \left\{ \left( A_{N_1} : D_{N_1} [C_{A_{N_1}}] \right), \dots, \left( A_{N_n} : D_{N_n} [C_{A_{N_n}}] \right), C_N \right\}, \quad i = 1, \dots, m \\ B & = \left\{ \left( A_{N_1} : \tilde{d}_{i_1} [c_{i_1}] \right), \dots, \left( A_{N_n} : \tilde{d}_{i_n} [c_{i_n}] \right), c_i \right\} \end{cases}$$

$$R'_N = \begin{cases} H' & = \left\{ \left( A'_{N_1} : D_{N_1} [C_{A'_{N_1}}] \right), \dots, \left( A'_{N_n} : D_{N_n} [C_{A'_{N_n}}] \right), C_N \right\}, \\ B' & = \left\{ \left( A'_{N_1} : \tilde{d}'_{k_1} [c'_{k_1}] \right), \dots, \left( A'_{N_n} : \tilde{d}'_{k_n} [c'_{k_n}] \right), c_i \right\} \end{cases}, \quad i = 1, \dots, m'$$

onde  $m$  e  $m'$  são as respectivas cardinalidade, e  $n$  e  $n'$  os respectivos graus, então, o *Produto Cartesiano Integrado Nebuloso* de  $R_N$  e  $R'_N$  está definido por:

$$R_N \times R'_N = \begin{cases} H_{\times N} & = H \times H' \\ B_{\times N} & = B \times B' \\ C_{\times N} & = \min(C, C') \end{cases}.$$

A *Seleção Integrada Nebulosa de tuplas*,  $\sigma_N$ , é assim definida:

**Definição:** Seja  $R_N$  uma *Relação Integrada Nebulosa* representada por:

$$R_N = \begin{cases} H & = \left\{ \left( A_{N_1} : D_{N_1} [C_{A_{N_1}}] \right), \dots, \left( A_{N_n} : D_{N_n} [C_{A_{N_n}}] \right), C_N \right\}, \\ B & = \left\{ \left( A_{N_1} : \tilde{d}_{i_1} [c_{i_1}] \right), \dots, \left( A_{N_n} : \tilde{d}_{i_n} [c_{i_n}] \right), c_i \right\} \end{cases},$$

sejam  $\tilde{a} \in D_N$  uma constante,  $\Theta$  um operador relacional nebuloso definido a partir de uma função de pertinência,  $\mu_\Theta(d, d')$ ,  $\lambda$  o grau de satisfação para o atributo implicado e  $\gamma$  o grau de satisfação para a tupla. A *Seleção Integrada Nebulosa* de tuplas da relação  $R_N$  a partir da condição imposta pelo operador  $\Theta$  e a constante  $\tilde{a}$  sobre o atributo  $A_{N_j}$ ,  $\sigma_N(R_N; \Theta(A_{N_j}, \tilde{a}) \geq \lambda) \geq \gamma$ , é a relação  $R_N^\sigma$  definida como:

$$R_N^\sigma = \begin{cases} H & = \left\{ \left( A_{N_1} : D_{N_1} [C_{A_{N_1}}] \right), \dots, \left( A_{N_n} : D_{N_n} [C_{A_{N_n}}] \right), C_N \right\} \\ B & = \left\{ \left( A_{N_1} : \tilde{d}_{i_1} [c_{i_1}] \right), \dots, A_{N_j} : \tilde{d}_{i_j}, c'_{i_j}, \left( A_{N_n} : \tilde{d}_{i_n} [c_{i_n}] \right), c'_i \right\} \end{cases},$$

com

$$c_\sigma = \sup_{(d \times d') \in D_N \times D_N} \min(\mu_\Theta(d, d'), \mu_{\tilde{a}}(d), \pi_{A_{N_j}}(d'))$$

$$c'_{i_j} = \begin{cases} c_\sigma & \text{se } \exists c_{i_j} \\ \min(c_\sigma, c_{i_j}) & \text{se } \exists c_{i_j} \end{cases}$$

$$c'_i = \min(c_i, c'_{i_j})$$

Para consulta mais complexas formadas por mais de uma condição combinadas com os operadores de negação, conjunção e disjunção é utilizada a seguinte definição:

O *Complemento da seleção* é definido como:

$$\neg \sigma_N(R_N; \Theta(A_{N_j}, \tilde{a}) \geq \lambda) \geq \gamma = \sigma_N(R_N; \neg \Theta(A_{N_j}, \tilde{a}) \geq \lambda) \geq \gamma,$$

onde

$$c'_{i_j} = \begin{cases} 1 - c_\sigma & \text{se } \exists c_{i_j} \\ 1 - \min(c_\sigma, c_{i_j}) & \text{se } \exists c_{i_j} \end{cases}$$

$$c'_i = \min(c_i, c'_{i_j})$$

A *Conjunção de seleções* define-se utilizando t - normas como mostra a seguinte equação:

$$t_j(\sigma_N(R_N; \Theta_j(A_{N_j}, \tilde{a}_j) \geq \lambda_j)) \geq \gamma = \bigcap_{j_N} \sigma_N(R_N; \Theta_j(A_{N_j}, \tilde{a}_j) \geq \lambda_j) \geq \gamma,$$

A *Disjunção de seleções* define-se utilizando s - normas como segue:

$$s_j(\sigma_N(R_N; \Theta_j(A_{N_j}, \tilde{a}_j) \geq \lambda_j)) \geq \gamma = \bigcap_{j_N} \sigma_N(R_N; \Theta_j(A_{N_j}, \tilde{a}_j) \geq \lambda_j) \geq \gamma$$

onde

$\lambda_j$ , são um conjunto de grau de satisfação para os atributos,

$\Theta_j$ , são um conjunto de operadores relacionais nebulosos,

$\tilde{a}_j \in D_{N_j}$ , conjunto de constantes e

$A_{N_j}$ , conjunto de atributos não necessariamente distintos; com  $j = 1, \dots, n$ .

#### 4.2.5 Proposta do SQL-Nebuloso.

Nesta seção é apresentada uma proposta de extensão do SQL [UI82, EN94], com o objetivo de gerar uma ferramenta que proporcione consultas as relações integradas nebulosas definidas. Esta linguagem de consulta diferencia-se sintática e semanticamente do SQL convencional, como será visto em seguida.

O SQL-Nebuloso permite a manipulação dos atributos nebulosos, através de operadores relacionais nebulosos e a manipulação das tuplas através de operadores de conjuntos definidos através de t - normas e s - normas.

A linguagem, na sua definição, inclui duas novas componentes que correspondem aos atributos de compatibilidade referentes aos atributos e a relação. Estes atributos funcionam como restrição ao grau de satisfação na aplicação de uma expressão nebulosa em um atributo e uma tupla. Para cada expressão nebulosa simples existira um termo **with** <certeza> e o mesmo termo é adicionado para toda a consulta. De forma geral a linguagem estaria definida como

```

select <atributos>

from <relações>

where <expressão nebulosa simples with <certeza>>

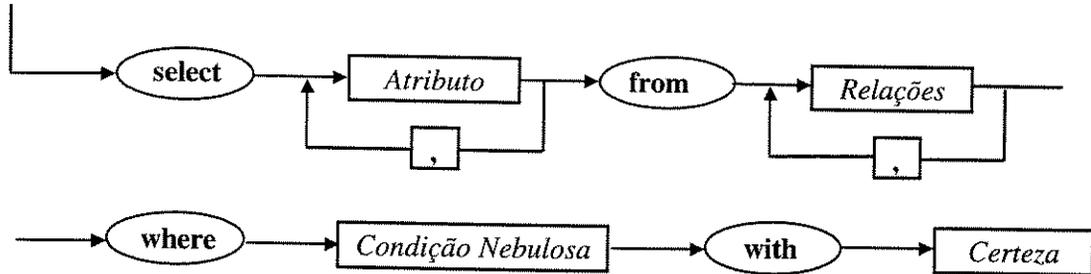
with <certeza> .

```

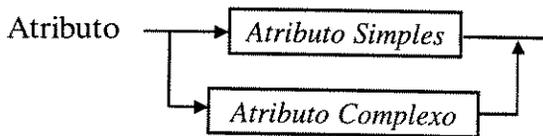
A seguir é definida a linguagem de forma mais detalhada.

### 4.2.5.1 Definição sintática da linguagem.

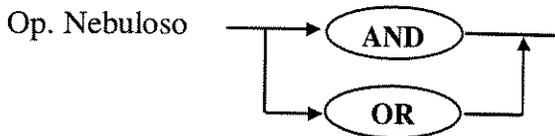
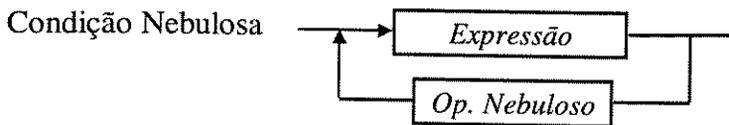
Operador de Seleção, *select*.



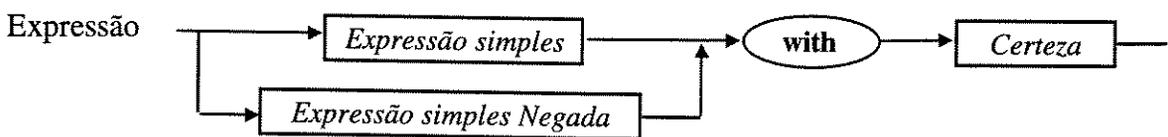
O termo *Certeza* representa o valor de certeza com a qual a consulta deve ser obtida.



Um *atributo simples*, é um atributo qualquer que pertence à relações definidas no termo *Relações*. O atributo complexo é um atributo definido como um conceito complexo que virtualmente pertence à mesma relação.



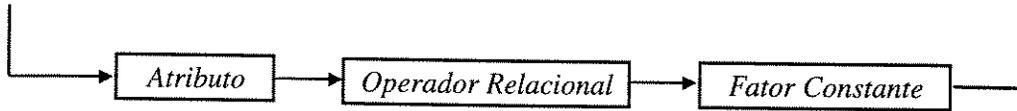
Os operadores de união (OR) e interseção (AND) são definidos como S-normas e T-normas respectivamente.



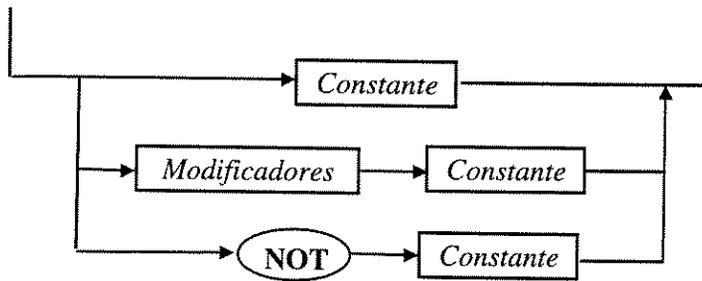
Expressão Simples Negada



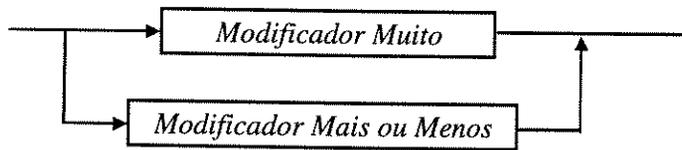
Expressão simples



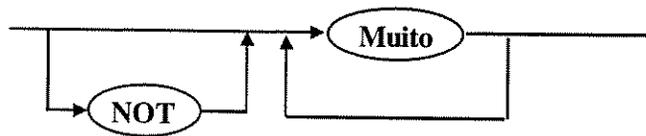
Fator Constante



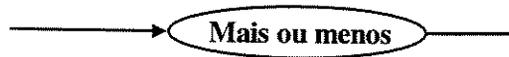
Modificadores



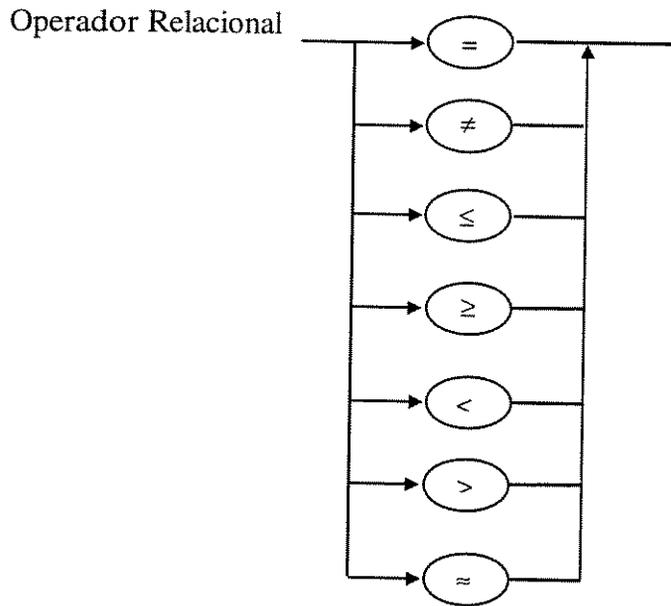
Modificador Muito



Modificador Mais ou Menos



Os modificadores são definidos como operadores de concentração e dilatação sobre conjuntos nebulosos respectivamente.



Os operadores relacionais são definidos a partir de funções de pertinência.

Algumas versões sobre SQL nebuloso foram definidas [Bo88, Ta77, Ta91], e elaboram mecanismos de consultas através de linguagens já existente ou linguagens novas. Todas elas com o objetivo de obter de forma imprecisa a informação armazenada em bancos de dados relacionais.

# Capítulo 5

## *Aspectos da Implementação do Modelo Proposto*

---

Uma arquitetura para a implementação do modelo integrado de bancos de dados nebulosos utilizando um sistema de banco de dados convencional é apresentada neste capítulo. Tal arquitetura é fundamentada no modelo de Medina [Me94] apresentado no capítulo anterior. Inicialmente é analisado como os diferentes tipos de valores nebulosos que o modelo manipula, podem ser representados sobre um banco de dados convencional sendo, em seguida definida uma proposta de implementação do meta-conhecimento e das consultas. Com estes elementos é definida uma arquitetura genérica para implementar um banco de dados nebuloso, a ser usada numa implementação do modelo sobre o DataBase Engine do sistema Delphi V 2.0 1996, para Windows 95 e Windows NT. A implementação é feita a partir da extensão da hierarquia de classes que o Delphi oferece para o tratamento de bancos de dados relacionais.

### **5.1 Arquitetura do SGBD Nebuloso.**

A proposta de implementação deste capítulo se baseia fundamentalmente na extensão de um Sistema Gerenciador de Bancos de Dados (SGBD) Convencionais, visando a manipulação de dados nebulosos. Para atingir este objetivo, é preciso estender os seguintes pontos:

- Extensão do conjunto de atributos permitidos, para incluir aqueles com tratamento nebuloso;
- Tipos de domínios para o tratamento de valores nebulosos;

- Linguagem de consulta.

Com a definição destes pontos apresenta-se a seguinte arquitetura

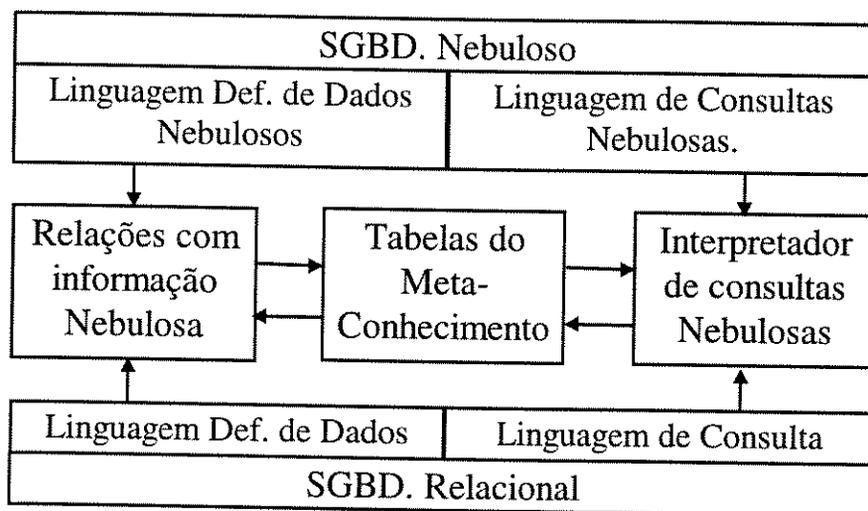


Figura 5.1. Modelo de Implementação

No modelo que se propõe na figura 5.1, identifica-se três componentes que formam o núcleo da implementação de um sistema de banco de dados nebulosos, sobre o SGBD Convencional. Tais componentes são:

- As Relações com informação Nebulosa correspondem com as tabelas convencionais que brinda o SGBD, sobre as quais são armazenadas as relações nebulosas. Os campos de tipo nebuloso são armazenados em campos destas relações de longitude variável.
- As Tabelas do Meta-Conhecimento armazenam a informação adicional necessária para o trabalho com relações, atributos e valores de caráter nebuloso.
- O Interpretador de Consultas Nebulosas transforma uma consulta do SQL-Nebuloso (ver Capítulo 4) a uma consulta do SQL Convencional, fornecida pelo SGBD suporte.

### 5.1.1 Extensão da Linguagem de Definição de dados.

A extensão da linguagem de definição de dados esta especificamente direcionada à adição de recursos para o tratamento de relações, atributos e valores nebulosos. Para isto, a partir de um SGBD Convencional é necessária a geração de tabelas adicionais (Tabelas do Meta-Conhecimento). Nesta seção, são apresentados os novos tipos de atributos que o modelo introduz assim como a representação física sobre o SGBD suporte e a meta-informação necessária para sua definição. Esta meta-informação é armazenada como relações. A figura 5.2 representa os tipos de dados que a nova linguagem de definição de dados permite.

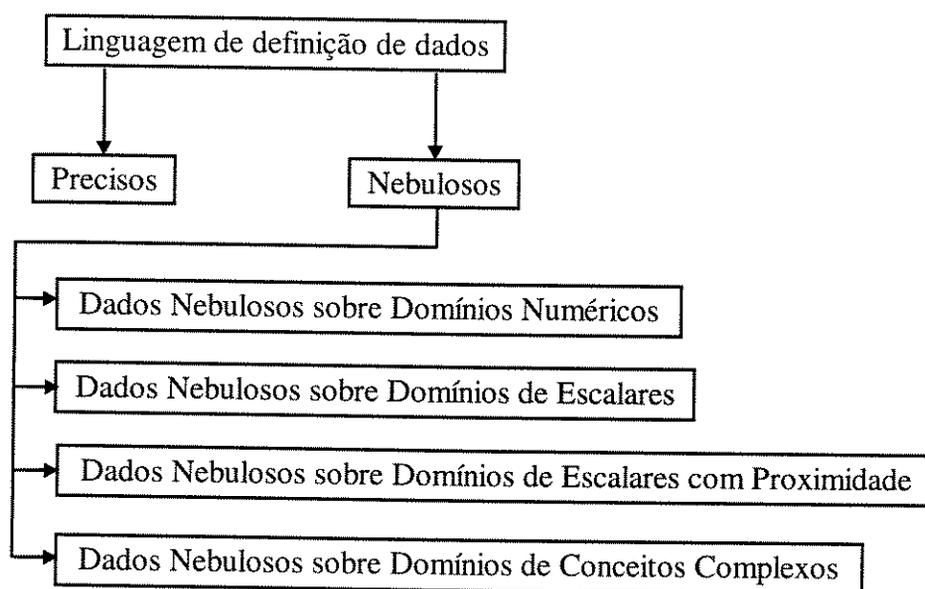


Figura 5.2. Tipos de dados do SGBD Nebuloso.

O tipo *Preciso* é o tipo inerente a um SGBD Convencional qualquer e refere-se aos tipos 1 e 2 da tabela 4.1 (capítulo 4). Para seu armazenamento, utiliza os mecanismos do SGBD Convencional sobre o qual esta suportado o modelo. Isto é, os formatos para alfanuméricos, valores numéricos, datas, horas, etc.

Os tipos de *Dados Nebulosos* são diferenciados no domínio base sobre os quais são definidos os tipos de valores que permitem armazenar. Cada tipo de dados é caracterizado de maneira geral como segue:

- *Dados Nebulosos sobre Domínios Numéricos:* Sobre um domínio numérico são definidas distribuições de possibilidade que podem ser, desde variáveis lingüísticas até um intervalo.
- *Dados Nebulosos sobre Domínios de Escalares:* Sobre um domínio de valores lingüísticos de escalares são definidas distribuições de possibilidade.
- *Dados Nebulosos sobre Domínios de Escalares com Proximidade:* Sobre um domínio de valores lingüísticos de escalares existe uma relação de proximidade e são definidas distribuições de possibilidade.
- *Dados Nebulosos sobre Domínios de Conceitos Complexos:* Define um domínio de valores de conceitos complexos. Estes conceitos foram definidos no capítulo 4.

Um SGBD Convencional possui mecanismos para reconhecer os tipos de domínios dos atributos das tabelas que formam o banco de dados. Com a adição de novos tipos de domínios não reconhecidos pelo SGBD suporte é preciso criar uma relação que armazene as tabelas, como também os atributos dentro delas que possuem tratamento nebuloso. A tabela 5.1 representa a estrutura geral da relação que armazena esta meta-informação.

TableName	FieldName	Domain
-----------	-----------	--------

Tabela 5.1. Tabela das relações e atributos nebulosos.

Os campos da relação são:

*TableName:* Nome da tabela onde está o atributo nebuloso,

*FieldName:* Nome do atributo nebuloso,

*Domain:* Domínio nebuloso que ao qual pertence o atributo.

Adicionalmente é criada uma tabela que descreve cada um dos domínios dos tipos de dados nebulosos. Esta tabela permite que atributos de tabelas diferentes possam

definir-se sobre o mesmo domínio, o que facilita a consulta. A tabela 5.2 representa a relação para armazenar os domínios.

Domain	DomainType	IniDomain	EndDomain	Jump
--------	------------	-----------	-----------	------

Tabela 5.2. Relação dos domínios nebulosos.

Os campos da relação são:

*Domain*: Domínio nebuloso,

*DomainType*: Tipo de tratamento nebuloso que recebe os valores que pertencem ao domínio.

*IniDomain*, *EndDomain* : Representam os limites de definição dos domínios numéricos.

*Jump* : Representa o passo de discretização dos domínios numéricos

Nas seções seguintes são apresentados cada um dos novos tipos de dados nebulosos, onde para cada um deles são definidos os tipos de valores que permitidos, a representação física no campo de longitude variável que o armazena, e a informação adicional que também deve ser armazenada.

#### 5.1.1.1 Dados nebulosos sobre Domínios Numéricos.

Este tipo de domínio foi chamado de **tipo 1** e se caracteriza por estar definido sobre domínios numéricos. Sobre eles são especificadas distribuições de possibilidade que diferenciam tipos de valores.

#### Tipos de valores

- Valores *Unknown*, *Undefined*, *Null*
- Valor *distribuição de possibilidade sobre Undefined*.
- Valor *Função*. Este tipo de valor é uma função de pertinência que descreve uma distribuição de possibilidade. O valor vai conter os parâmetros que descrevem a

função assim como o tipo de função. Por exemplo para o atributo altura o seu valor é  $altura = triangular (a=1.20m, m=n=1.50m, b=1.80m)$ , figura 5.3.

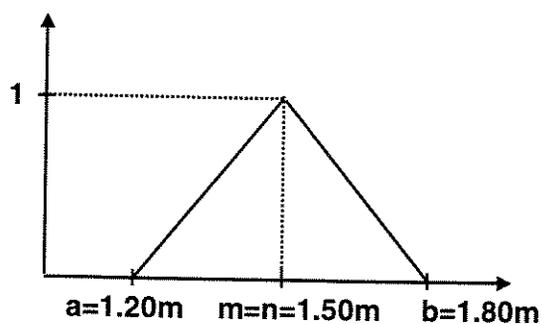


Figura 5.3. Função Triangular

- Valor *Rótulo*. São variáveis lingüísticas que definem uma distribuição de possibilidade sobre o domínio. Fazem referencia a um conceito nebuloso representado por uma função de pertinência. Por exemplo para o atributo altura o seu valor é  $altura = alto$ .
- Valores *aproximados*. Dado um valor,  $n$ , pode ser representado o conceito impreciso “aproximadamente  $n$ ”. Serão utilizadas funções de pertinência triangulares. Por exemplo, para o atributo altura pode ser definido  $altura = aproximadamente 1.70m$ .
- Valor *Intervalo*. Representado pela função da figura 5.4. É tratada como uma função trapezoidal

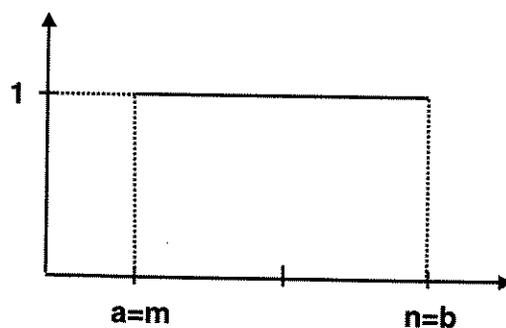


Figura 5.4. Distribuição Intervalar

### **Parâmetros adicionais para cada tipo.**

Por causa da variedade de tipos de valores que os atributos de **Tipo 1** permitem definir, é preciso que o atributo leve junto ao valor, o tipo de valor que pode ser lido. Adicionalmente, cada valor se complementa com um conjunto de parâmetros que devem ser armazenados. Com isto, um atributo de **tipo 1** é formado por duas partes: o tipo de valor e os parâmetros do valor. A seguir são definidos os tipos de valores e os parâmetros, que segundo o caso, devem ser armazenados no atributo.

Os **tipos de valores** são: *Valor Unknown*, *Valor Undefined*, *Valor Null*, *Crisp*, *Rótulo*, *Intervalo*, *Valor Aproximado*, *Função Nebulosa* e *Distribuição de possibilidade sobre Undefined*.

O conjunto de parâmetros que complementam a definição do valor são:

Os tipo de valores *Unknown*, *Undefined* e *Null* não precisam informação adicional além da especificação do tipo.

*Valor Crisp*: O valor.

*Valor Rótulo*: O string que representa a variável lingüística.

*Valor Intervalo*: Os valores inicial e final do intervalo.

*Valor Aproximado*: O valor aproximado e o suporte da função triangular utilizada para representar o conceito aproximado.

*Valor Função*: O identificador da função e os valores dos parâmetros que descrevem a função.

*Valor Distribuição de possibilidade sobre Undefined*: O valor de possibilidade definido sobre *Undefined*, o tipo de valor definido sobre o resto dos valores do domínio (não podem ser os tipos que representam os valores *Unknown*, *Undefined*, *Null* ou o próprio) e em dependência do valor do campo anterior os parâmetros que descrevem esse tipo.

**Representação física do atributo.**

Este tipo de valor deve ser armazenado em um campo de longitude variável. Esta representação divide-se em duas partes: tipo de valor e os parâmetros do tipo. .A tabela 5.3 mostra as treze formas de estrutura de um atributo de **Tipo 1**.

Tipo 1								
T. Unknown								
T. Undefined								
T. Null								
T. Crisp	valor							
T. Rótulo	Rótulo							
T. Intervalo	Valor Inicial	Valor Final						
T. Aproximado	valor Maximo	Suporte						
T. Função	tipo de função	a	m	n	b	k		
T. Dist. Poss. s/ Undefined.	Poss. s/ Undefined	T. Crisp	value					
		T. Rótulo	Rótulo					
		T. Intervalo	Valor Inicial	Valor Final				
		T. Aproximado	Valor Máximo	suporte				
		T. Função	Tipo de função	a	m	n	b	k

Tabela 5.3. Representação física de um atributo de tipo numérico.

**Tabelas do meta-conhecimento relacionados ao tipo de dado.**

Os atributos de **tipo 1** precisam da definição de duas tabelas adicionais que permitam armazenar a meta-informação. Estas tabelas completam a definição dos tipos função e Rótulo.

Tabela para a definição das funções.

Armazena a representação matemática da função, tais funções são utilizadas na definição do tipo de valor *Rótulo*, *aproximado*, *intervalar* e *função*. Adicionalmente, estas funções são utilizadas para definir os operadores relacionais e os operadores de conjunto.

As funções permitidas para a definição dos tipos de valores são as parametrizadas [WF97] que são as mais usadas. Dentre elas pode-se citar a função triangular, função- $\Gamma$  e a função gaussiana.

Estrutura:

FuncName	FuncType	Exp_Mat	Polonesa
----------	----------	---------	----------

Tabela 5.4. Relação das funções.

Os campos da relação são:

*FuncName*: Nome da função.

*FuncType*: Tipo da função que está armazenada.

*Exp\_Mat*: Expressão matemática que define a função.

*Polonesa*: Notação polonesa que descreve a expressão matemática da função.

Tabela para a definição de rótulos em domínios numéricos.

Armazena os dados das funções de pertinência que descrevem os rótulos para domínios numéricos.

Estrutura:

Domain	Label	Funcid	a	m	n	b	k
--------	-------	--------	---	---	---	---	---

Tabela 5.5. Relação das variáveis lingüísticas.

Os campos da relação são:

*Domain*: Domínio de definição do rótulo.

*Label*: Identificador do valor.

*FuncId*: Identificador da função de pertinência que descreve o atributo.

*a, m, n, b, k* : Parâmetros que definem a função que descreve o rótulo.

### 5.1.1.2 Dados Nebulosos sobre Domínios de Escalares.

Este tipo de domínio, chamado de **tipo 2**, é construído sobre domínios discretos de escalares, sobre os quais são definidas distribuições de possibilidade.

#### Tipos de valores

- Valores *Unknown, Undefined, Null*
- Valor *distribuição de possibilidade sobre todo o domínio, Undefined*.

Valores *Escalares simples*. Utilizam para o seu armazenamento os mecanismos oferecidos pelo sistema de banco de dados que o suporta. Por exemplo: suponha o valor **preto** para o atributo *cor*. Para armazenar este dado é utilizado o tipo *string* de qualquer sistema.

- Valor *distribuição de possibilidade sobre discretos*. Devem ser armazenados os valores do domínio com os valores de possibilidade associados  $((v_1, p_1), \dots, (v_n, p_n))$ . Por exemplo  $((preto, 0.5), (moreno, 0.8))$ .

#### Parâmetros adicionais para cada tipo.

O atributo, deve armazenar o tipo de valor que vai ser definido e os parâmetros que descrevem o valor. Com isto, para cada valor de atributo, é armazenado o tipo de valor e os parâmetros adicionais que complementam a definição de cada tipo.

Os **tipos de valores** são: *Valor Unknown*, *Valor Undefined*, *Valor Null*, *Valor Crisp*, *Conjunto de escalares com uma distribuição de possibilidade* e *Distribuição de possibilidade sobre Undefined*.

A seguir é apresentado o conjunto de parâmetros que descreve cada tipo de valor:

*Crisp*: O valor propriamente.

*Conjunto de escalares com uma distribuição de possibilidade*: São armazenados os pares valor e possibilidade do valor. A quantidade máxima de pares a ser armazenada é um valor definido como constante na implementação.

*Valor Distribuição de possibilidade sobre Undefined*: O valor de possibilidade sobre o valor *undefined*, o tipo do valor definido sobre o resto dos valores do domínio e dependendo do tipo do valor do campo anterior, os parâmetros que descrevem esse tipo. O tipo do valor definido sobre o resto dos valores do domínio, não podem ser os tipos que representam os valores *Unknown*, *Undefined*, *Null* ou o próprio.

### Representação física do atributo.

A partir dos tipos de valores que podem definir os atributos de **tipo 2**, pode-se concluir na tabela 5.6 as sete possíveis formas de estrutura que pode ter um atributo deste tipo.

Tipo de dado						
T. <i>Unknown</i>						
T. <i>Undefined</i>						
T. <i>Null</i>						
T. <i>Crisp</i>	Valor					
T. Distr. Poss.	Valor	poss.	...	valor	poss.	
T. Poss. s/ <i>Undefined</i>	T. <i>Crisp</i>	valor				
	T. Dist. Poss.	valor	poss.	...	valor	poss.

Tabela 5.6. Representação física do tipo escalar.

### **Tabelas do meta-conhecimento relacionadas ao tipo de dado.**

Os atributos de **tipo 2** não precisam informação adicional para armazenar o seu significado.

#### **5.1.1.3 Dados Nebulosos sobre Domínios de Escalares com Proximidade.**

Este tipo de domínio é igualmente construído sobre domínios discretos de escalares. Mas, entre os valores que pertencem ao domínio, existe uma relação de proximidade. Por esta razão, define-se um novo tipo de atributo, o **tipo 3**.

#### **Tipos de valores**

Os tipos de valores são os mesmos que o tipo anterior, o **tipo 2**.

#### **Parâmetros adicionais para cada tipo.**

Os parâmetros adicionais para cada tipo de valor foram apresentados no **tipo 2**.

#### **Representação física do atributo.**

A representação física deste tipo de atributo é igual à do **tipo 2**.

### **Tabelas do meta-conhecimento relacionados ao tipo de dado.**

Adicionalmente, para este tipo de atributo, deve ser armazenada a relação de proximidade entre os elementos do domínio. A proximidade entre os elementos é armazenada numa relação com estrutura igual à da tabela 5.7.

#### Tabela de definição das relações de proximidade.

Estrutura

Domain	value1	value2	grade
--------	--------	--------	-------

Tabela 5.7. Relações de proximidade.

Os campos da relação são:

*Domain*: Domínio ao qual pertencem os valores.

*Value1*: Identificador do primeiro valor do par a relacionar.

*Value2*: Identificador do segundo valor do par a relacionar.

*Grade*: Contem o grau de proximidade entre os dois valores.

#### 5.1.1.4 Dados Nebulosos sobre Domínios de conceitos complexos.

Este tipo de valor não pertence fisicamente à relação, sendo calculado cada vez que é preciso, a partir de uma expressão lógica nebulosa. Os atributos que compõem esta expressão estão definidos em outra relação que pertence ao mesmo banco de dados. Mas, pelo tratamento que este tipo de dados recebe, define um novo tipo de atributo, o **tipo 4**. Este atributo só trabalha com um tipo de valor que é definido em uma tabela adicional.

#### Tabelas do meta-conhecimento relacionados ao tipo de dado.

Para os atributos de **tipo 4**, é preciso armazenar em relações adicionais a definição do conceito complexo e alguma informação adicional relacionada ao conceito. As tabelas 5.8 e 5.9 representam, a primeira, um conjunto de pares de funções AND e OR que são utilizadas nas expressões e a segunda a própria definição do conceito.

##### Tabela para a definição dos pares AND e OR.

Estrutura:

PairName	FuncAnd	FuncOr
----------	---------	--------

Tabela 5.8. Relação dos pares AND-OR.

Os campos da relação são:

*PairName* : Nome do par que o identifica.

*FuncAnd*: Identificador da função de pertinência que descreve a função AND.

*FuncOr*: Identificador da função de pertinência que descreve a função OR.

Tabela de definição de conceitos complexos.

Estrutura:

TableName	Domain	Value	Expr	Polonesa
-----------	--------	-------	------	----------

Tabela 5.9. Relação que define os conceitos complexos.

Os campos da relação são:

*TableName*: Armazena o nome da relação onde estão os atributos que formam a expressão.

*Domain*: Domínio ao que pertence o conceito.

*Value*: Valor definido como conceito complexo.

*Exp*: Expressão lógica nebulosa que descreve o conceito.

*Polonesa*: Expressão em notação polonesa que representa a expressão lógica nebulosa que descreve o conceito.

Um atributo do **tipo 4**, utiliza o interpretador de conceitos nebulosos para obter o valor final. A expressão lógica nebulosa é aplicada sobre a relação a partir do operador **select**.

### 5.1.2 Extensão da linguagem de Consultas.

A extensão da linguagem de consultas esta fundamentada na introdução de um interpretador de consultas nebulosas. Este elemento deve realizar as seguintes funções:

- A análise sintática de uma consulta na linguagem,

- A análise semântica.

A figura 5.5 mostra a descrição do interpretador em duas componentes: Analisador sintático e Analisador semântico. Cada componente realiza as respectivas operações antes definidas.

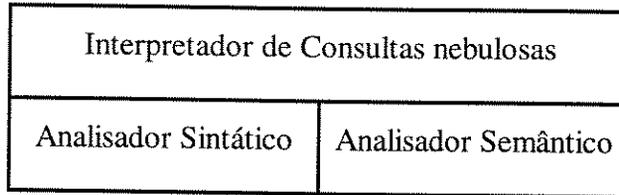


Figura 5.5. Componentes do Interpretador de Consultas nebulosas.

## 5.2 Implementação em Delphi.

As propostas para a implementação de um SGBD Nebuloso feitas na seção anterior são utilizadas para implementação de um SGBD Nebuloso sobre o Delphi. O Delphi permite o gerenciamento e manipulação de bancos de dados através dos seus componentes que são apresentados numa hierarquia de classes na figura 5.6.

A notação OMT[Ru95a,Ru95b] é usada na descrição de classes ao longo de todas as seções deste capítulo

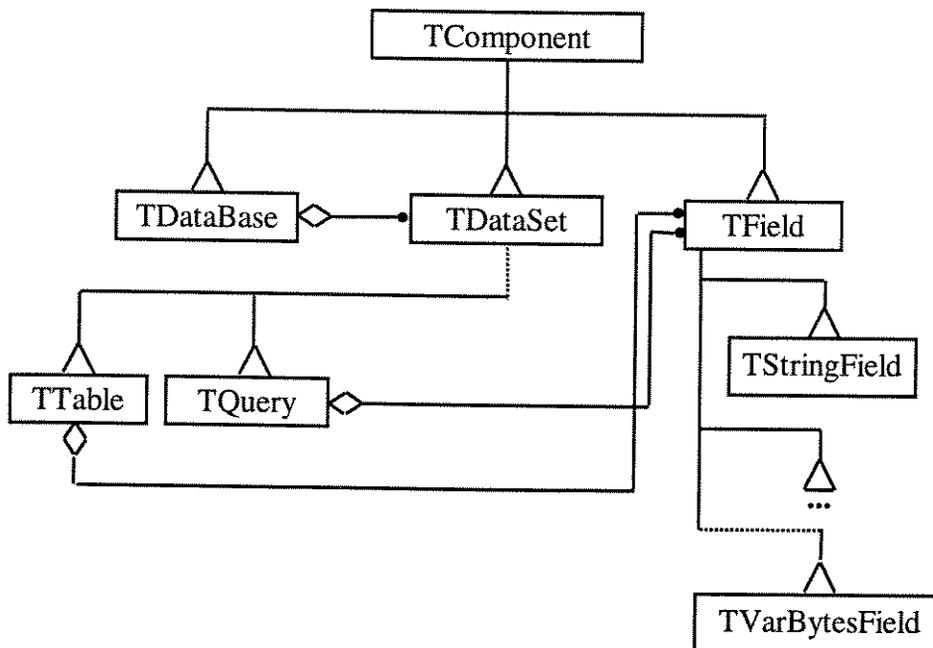


Figura 5.6. Hierarquia do Delphi para bancos de dados.

As classes que descrevem esta hierarquia são definidas a seguir:

**TComponent:** É uma super-classe á partir da qual herdam todas as classes que permitem construir uma aplicação sobre o Delphi [DA95].

**TDataBase:** Encapsula a conexão a um banco de dados dentro de uma aplicação. Contém uma referencia a todas as instancias de TDataSet pertencentes ao banco de dados [DA95].

**TDataSet:** Armazena um conjunto de dados organizados em tuplas que podem estar em uma tabela ou ser resultado de uma consulta [DA95].

**TTable:** Encapsula a funcionalidade sobre as tabelas. Contem um conjunto de instancias da classe **TField** que representam a estrutura da tabela [DA95].

**TQuery:** Encapsula instruções em SQL que permitem a recuperação de um conjunto de dados [DA95].

**TField:** É uma super-classe, a partir da qual herdam todas as classes para a definição dos tipos de atributos. Na figura 5.6 são representadas duas classes que exemplificam esta herança. A classe **TStringField** é utilizada para representar os atributos de tipo *string*. A outra classe, **TVarByteField**, armazena um atributo de longitude variável.

O objetivo da implementação realizada neste trabalho é a extensão desta hierarquia, criando componentes com funcionalidades similares no contexto de Bancos de Dados Nebulosos. A figura 5.7 representa esta extensão, onde diferenciam-se quatro classes que constituem o núcleo da implementação realizada: **TFuzzyDataBase**, **TFuzzyTable**, **TFuzzyMetaKnowledge** e **TFuzzyField**.

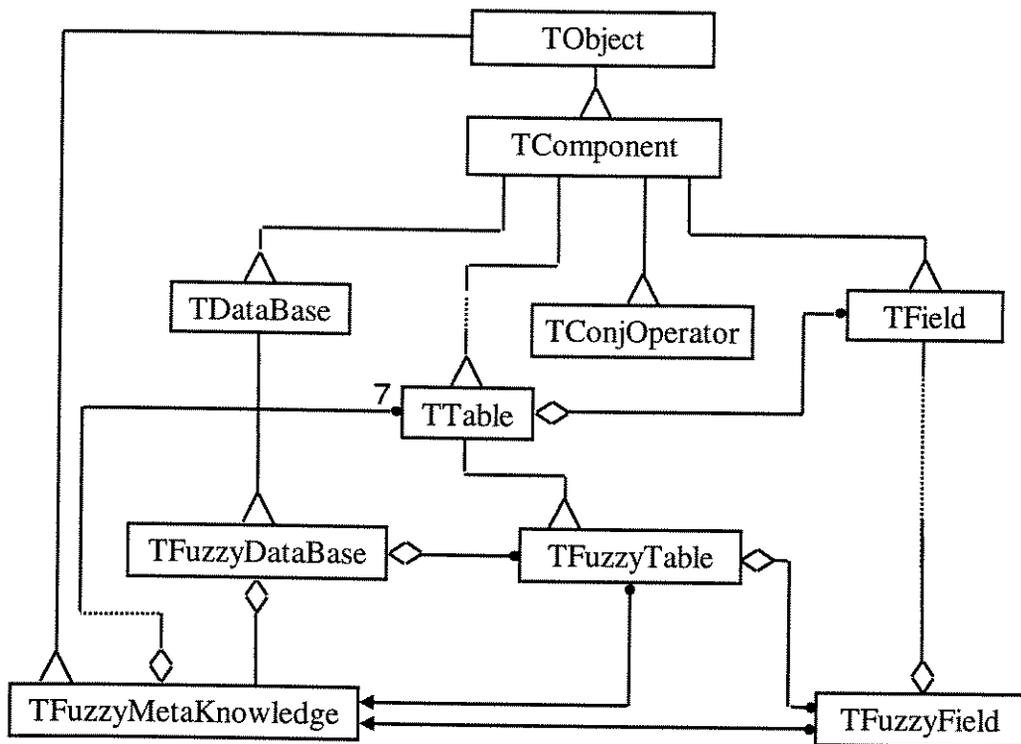


Figura 5.7. Extensão da hierarquia para o tratamento de Relações Nebulosas.

**TFuzzyDataBase:** Herda de **TDataBase**. Quando esta classe é instanciada, cria um objeto da classe **TFuzzyMetaKnowledge** e uma lista para armazenar as tabelas nebulosas criadas na aplicação que pertencem ao banco de dados. A **TFuzzyDataBase** através dos seus métodos permite: conhecer as relações nebulosas que formam o banco de dados e o acesso à classe do meta-conhecimento.

**TFuzzyMetaKnowledge:** Encapsula a funcionalidade sobre as tabelas do meta-conhecimento. Quando é criada uma instancia de **TFuzzyMetaKnowledge** são geradas estruturas para as tabelas de meta-conhecimento. Estas tabelas são manipuladas como instâncias de **TTable**

**TFuzzyTable:** Representa as relações nebulosas. As tabelas nebulosas são implementadas sobre tabelas convencionais. Herdeira de **TTable**, esta classe é responsável pelo gerenciamento dos atributos nebulosos que são referenciados por uma lista que é mantida nesta classe. **TFuzzyTable:** também fornece operadores básicos sobre uma tabela, tais como: criar, apagar, abrir ou fechar uma tabela

nebulosa, conhecer e atualizar a estrutura de uma tabela, operadores para o acesso e manipulação das tuplas e dos atributos da relação.

**TFuzzyField**: Representa um atributo nebuloso genérico. Possui um atributo de tipo **TField** que armazena o campo definido em **TTable**. A **TFuzzyField** representa a super-classe de uma hierarquia com os diferentes tipo de atributos nebulosos apresentados na seção que segue. As classes desta hierarquia são: **TScalarFuzzyField**, **TAnalogyFuzzyField**, **TComplexFuzzyField** e **TNumericFuzzyField** como é apresentado na figura 5.8.

**TConjOperator**: Classe que define os operadores de conjunto utilizadas para a manipulação dos dados. Estes operadores são definidos a partir de funções de pertinência que são passadas como parâmetros nos métodos que descrevem os operadores.

### 5.2.1 Implementação dos tipos de atributos.

A hierarquia que permite o trabalho com os tipos de atributos nebulosos é descrita nesta seção. Para cada tipo de atributo nebuloso (Dados nebulosos sobre domínios numéricos, domínios de escalares, domínios escalares com proximidade e domínios de conceitos complexos) foi criada uma classe que representa aquele tipo. Também foi criada uma hierarquia que permite o trabalho com os tipos de valores que podem ser representados em cada tipo de atributo, como mostra a figura 5.9.

A figura 5.8 mostra as quatro classes para o trabalho com os atributos nebulosos. A mesma figura, apresenta a classe **TFuzzyValue**, representando uma super-classe que define os métodos comuns a todos os tipos de valores nebulosos que podem ser definidos em cada tipo de atributo. Dela, herdam quatro classes (figura 5.9) que por sua vez, definem hierarquias de classes. Cada uma destas hierarquias descreve os diferentes tipos de valores que podem ser definidos para cada um dos atributos representados pelas classes, **TNumericFuzzyField**, **TScalarFuzzyField**, **TAnalogyFuzzyField**. Por exemplo da classe **TScalarValue** herdam quatro classe: **TListDistPoss**, **TCrisp**, **TStrPossUnd**, **TSUndefined**, **TSUnknown**, **TSNull**. Cada

uma delas representa os tipos de valores, *distribuição e possibilidade sobre conjunto de escalares, crisp, distribuição de possibilidade sobre Undefined*, o valor *Undefined*, o valor *Unknown* e o valor *Null* respectivamente.

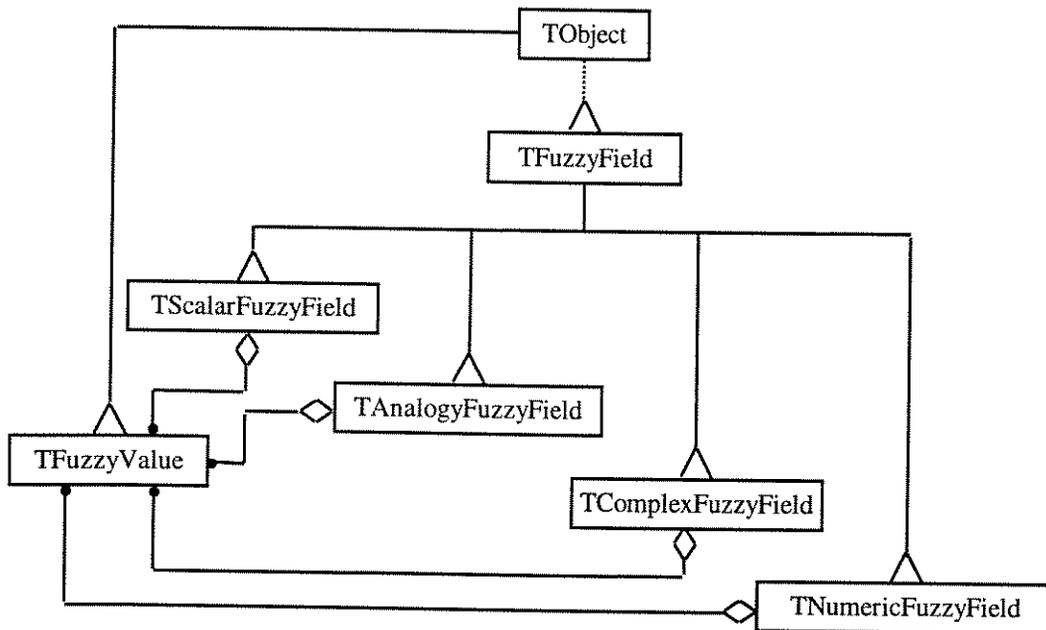


Figura 5.8. Hierarquia para a extensão dos tipos de atributos nebulosos.

As classes **TNumericFuzzyField**, **TScalarFuzzyField**, **TAnalogyFuzzyField**, funcionam geralmante da seguinte forma: cada classe possui um método que especifica o tipo de valor nebuloso que o atributo armazena na tupla atual. Além disto, as classes possuem um método para cada tipo de valor que tal atributo possa definir. O usuário, para conhecer o valor armazenado, deve primeiro conhecer o tipo de valor e em dependência disto executa o método para ler o valor de maneira apropriada. Existe uma verificação de erro que não permite ler outro tipo de valor que não seja o armazenado.

**TNumericFuzzyField**: Representa o tipo de atributo de **tipo 1** (seção 5.1.2.1 **TNumericFuzzyField** oferece métodos para o trabalho com os diferentes tipos de valores que ela pode definir.

**TScalarFuzzyField**: Representa os tipos de atributos de **tipo 2** (seção 5.1.2.2).

**TAnalogyFuzzyField**: Representa os tipos de atributos de **tipo 3** (seção 5.1.2.3).

**TComplexFuzzyField**: Representa os tipos de atributos de **tipo 3** (seção 5.1.2.4).

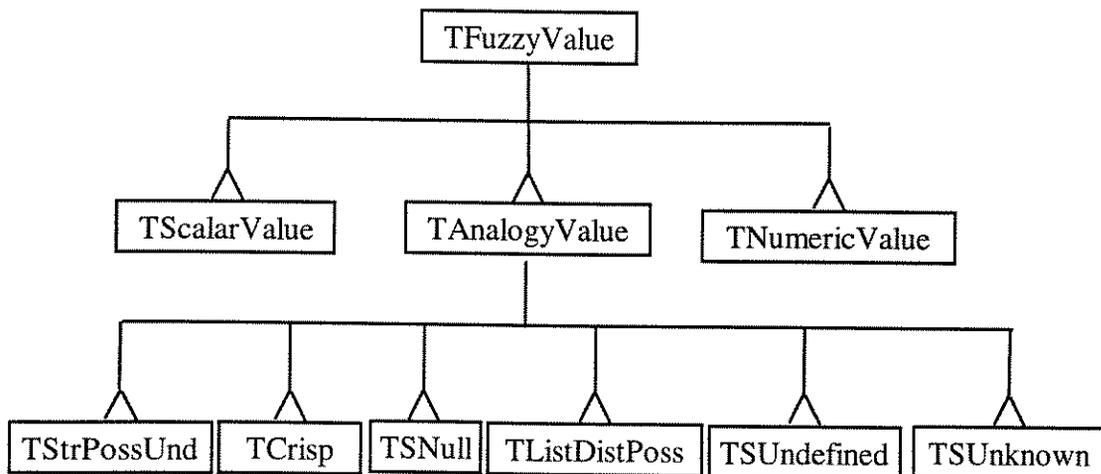


Figura 5.9. Hierarquia dos valores nebulosos.

A classe **TFuzzyValue** define os métodos comuns para o trabalho com os operadores relacionais sobre cada tipo de valor. Em cada classe, de quaisquer das hierarquias, estes métodos são redefinidos seguindo a semântica do tipo de valor. Por exemplo, o operador de igualdade para um valor de um atributo definido sobre um domínio com proximidade entre seus elementos (**TAnalogyValue**) e diferente para um valor de um atributo de tipo numérico (**TNumericValue**). Adicionalmente, cada uma destas classes implementa os operadores para o acesso ao valor que representa.

### 5.2.2 Implementação do meta-conhecimento.

Para o trabalho com as tabelas, os atributos e os valores nebulosos é preciso armazenar a meta-informação que completa a definição de cada um destes componentes.

A classe **TFuzzyMetaKnowledge** encapsula toda a funcionalidade para o trabalho com a meta informação. Na **TFuzzyMetaKnowledge** são instanciadas todas as classes representadas na hierarquia da figura 5.10. Esta classe adicionalmente oferece um conjunto de métodos para o trabalho com múltiplas tabelas.

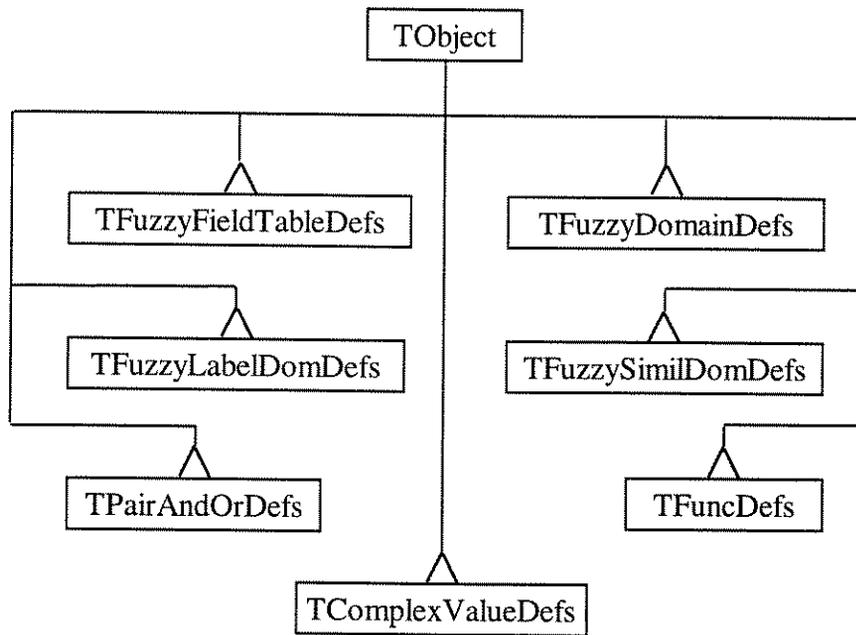


Figura 5.10. Hierarquia das tabelas do meta-conhecimento.

Na hierarquia existe uma classe para cada tabela do meta-conhecimento que foram apresentadas .ao início do capítulo.

**TFuzzyFieldTableDefs:** Representa a tabela das relações e os atributos com caracter nebuloso. e oferece métodos para adicionar ou apagar um atributo nebuloso e atualizar a estrutura do atributo.

**TFuzzyDomainDefs:** Representa a tabela para a definição dos domínios. e permite adicionar ou apagar um domínio e atualizar a definição dele.

**TFuzzyLabelDomDefs:** Representa a tabela das variáveis lingüísticas definidas sobre os domínios de **tipo 1**. Permite adicionar ou apagar um Rótulo e atualizar a sua definição.

**TFuzzySimilDomDefs:** Representa a tabela das relações de proximidade. Oferece métodos para adicionar ou apagar um valor e modificar a similaridade definida entre eles.

**TPairAndOrDefs:** Representa a tabela de definição dos pares de funções AND e OR. Oferece métodos para adicionar ou apagar um par de funções e actualizar a definição deles.

**TFuncDefs:** Representa a tabela da definição das funções. Oferece métodos que permitem adicionar ou apagar uma função e actualizar a definição delas.

**TComplexValueDefs:** Representa a tabela de definição dos conceitos complexos. Oferece métodos para adicionar ou apagar um conceito e actualizar a sua definição.

# Capítulo 6

## *Exemplo de aplicação*

---

Neste capítulo é enunciado um problema cujas variáveis apresentam incerteza na informação, para que a sua solução seja encaminhada de acordo com o modelo proposto no capítulo 4. Primeiramente, são mostradas as relações nebulosas que modelam o problema e em seguida, é descrita a representação destas relações a partir da implementação desenvolvida no capítulo 5.

Uma outra questão também apresentada aqui, é a da consulta ao banco de dados seguindo as definições do capítulo 4.

Um problema pratico envolvendo incerteza é assim enunciado: seja uma imobiliária e os seus diversos imóveis para alugar. Cada um deles, caracterizado por um subconjunto de propriedades cuja descrição pode ser imprecisa. Por exemplo, a área de um imóvel pode não ser conhecida de forma precisa mas pode ser caracterizada como: *grande*, *pequena* ou *mediana*.

É importante frisar, que os valores associados aos atributos imprecisos, refletem os conceitos próprios desenvolvidos pelos administradores da imobiliária. Assim, o preenchimento de uma mesma tabela pode ser bastante diferente de uma imobiliária para outra.

Outro aspecto importante, é lembrar que o que foi proposto ao longo deste trabalho, e implementado como descrito no capítulo 5, é um subconjunto de ferramentas que proporcionam a um analista, por exemplo, a resolver problemas como o enunciado

acima. Particularmente, no que diz respeito a implementação de regras conceituais sobre variáveis imprecisas, como a área de imóvel, citada no exemplo da imobiliária.

As relações que formam o banco de dados da imobiliária incluem todos os tipos de imprecisão e incerteza que o modelo trata. As consultas a serem feitas pelos potenciais locatários tem como restrição os domínios dos atributos predefinidos.

É fundamental, no processo de implementação de um banco de dados com informações imprecisas, que os responsáveis pela implementação interajam com os administradores da empresa a ser mapeada para que possam entender a base conceitual e transcrever suas regras.

### 6.1 Definição das relações para solução do problema.

Tendo em conta as atividades que se realizam em uma Imobiliária, no que diz respeito a locação de imóveis, foi criado o banco de dados *Imóveis*, formada pelas seguintes relações.

A relação **Imóvel** armazena as características dos imóveis a serem alugados.

*Imóvel*(*Id\_Im*, *Endereço*, *Aluguel*, *Acabamento*, *Tipo*, *Idade*, *Elevadores*, *Área*, *Estado*, *Quartos*, *Id\_Prop*, *Piscina*, *banheiro*).

A relação **Proprietário** armazena os dados pessoais dos proprietários com os quais a imobiliária trabalha.

*Proprietário*( *Id\_Prop*, *Nome*);

A relação **Acabamento** armazena as características do acabamento dos imóveis.

*Acabamento*( *Id\_Im*, *Piso*, *Paredes*, *Ins\_Elec*, *Ins\_Hid* )

A relação **Piscina** armazena as características das piscinas dos imóveis, se eles tiverem.

*Piscina*( *Id\_Im*, *Profundidade*, *Área* )

A relação **Quartos** armazena as características dos quartos dos imóveis.

**Quartos**( *Id\_Iml, Num\_Quarto, Área*)

## 6.2 Definição dos domínios dos atributos.

Nesta seção é apresentado o significado de cada atributo junto com o seu domínio de definição. Para cada atributo, é definido o conjunto de valores permitidos e o tipo de dado ao qual ele pertence, segundo a classificação feita no capítulo 5. Os atributos nebulosos utilizam para sua representação o modelo definido no capítulo 5 e os atributos precisos, os recursos brindados pelo Delphi.

### Relação Imóvel.

*Id\_Im*: Atributo que identifica o imóvel. Constitui a chave primaria da relação Imóvel. Define-se  $Dom(Id\_Im) = \{Identificadores\}$ . Atributo de **tipo preciso**.

*Endereço*: Atributo que armazena o endereço do imóvel. Define-se  $Dom(Endereço) = \{Endereços\}$ . Atributo de **tipo preciso**.

*Aluguel*: Atributo que armazena o valor do aluguel do imóvel. Define-se  $Dom(Aluguel) = \{Reais\}$ . Atributo de **tipo preciso**.

*Acabamento*: Atributo que define a qualidade do acabamento do imóvel. Define-se  $Dom(acabamento) = \{ bom, ruim, excelente, péssimo, Unknown \}$ . Atributo nebuloso do **tipo 4**.

Por ser de tipo 4 (capítulo 5), o atributo *Acabamento*, não está definido fisicamente como parte da relação e seus valores são definidos como conceitos complexos. O valor deste atributo é calculado a partir das expressões lógicas definidas na tabela 6.1. Os atributos implicados na expressão estão definidos na relação **Acabamento**.

Valor	Expressão lógica
Péssimo	$(Paredes = umidas \text{ AND } Ins\_Elec = pessima \text{ AND } Ins\_Hid = pessima) \geq 0.8$
Regular	$(Ins\_Elec = bom \text{ AND } Ins\_Hid = regular) \geq 0.8$
Boa	$(Paredes = pintadas \text{ AND } Ins\_Elec = bom \text{ AND } Ins\_Hid = bom) \geq 0.8$

Tabela 6.1. Definição dos conceitos complexos: *Péssimo*, *Regular*, *Boa*.

Os operadores relacionais e os operadores de conjunto que formam expressão nebulosa são definidos no momento da avaliação do conceito para procurar o valor.

*Tipo*: Atributo que descreve o tipo do imóvel que se oferece. Define-se  $Dom(Tipo) = \{Casa, Apartamento, Kitnet, Unknown\}$ . Atributo nebuloso de **tipo 2**.

*Idade*: Atributo que armazena a idade do apartamento. Define-se  $Dom(Idade) = \{novo, mediana, velho, \text{números inteiros}, Unknown, \text{valores aproximados}, intervalos\}$ . Atributo nebuloso de **tipo 1**.

A figura 6.1 mostra a definição dos valores lingüísticos: *novo*, *mediana*, *velho*, usados no atributo *Idade*.

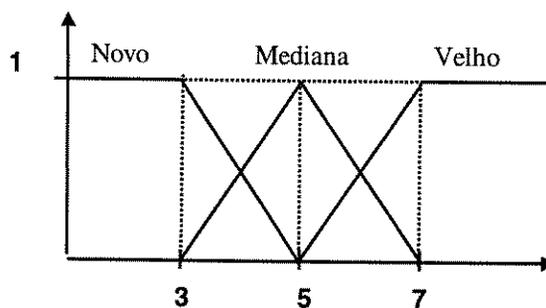


Figura 6.1. Valores Lingüísticos do domínio *Idade*.

*Elevadores*: Armazena o número de elevadores do imóvel, seja ele um apartamento ou uma residência. Define-se  $Dom(Elevadores) = \{\text{números inteiros}, Unknown, Undefined, Null\}$ . Atributo nebuloso de **tipo 1**.

*Área*: Atributo que armazena a área do imóvel. Define-se  $Dom(\textit{Área})=\{\text{números reais, Unknown, intervalos, valores aproximados}\}$ . Atributo nebuloso de **tipo 1**.

*Estado*: Atributo que armazena o estado geral do imóvel. Define-se  $Dom(\textit{Estado})=\textit{Estado}$ . O domínio *Estado* caracteriza o estado físico de um objeto sendo formado pelos valores *Bom*, *regular*, *excelente*, *péssimo*, *Unknown*. Atributo nebuloso de **tipo 3**. Entre os valores do domínio existe uma relação de proximidade representada na tabela 6.2.

<i>Relação</i>	<b>Bom</b>	<b>Regular</b>	<b>Excelente</b>	<b>Péssimo</b>
<b>Bom</b>	1	0.2	0.8	0.1
<b>Regular</b>	0.2	1	0.1	0.8
<b>excelente</b>	0.8	0.1	1	0
<b>Péssimo</b>	0.1	0.8	0	1

Tabela 6.2. Relação de proximidade entre os valores do domínio *Estado*.

*Quartos*: Atributo que armazena o número de quartos do imóvel. Define-se  $Dom(\textit{quartos})=\textit{Quantidades}$ . Atributo nebuloso de **tipo 1**.

*Id\_Prop*: Atributo que armazena o código do proprietário do imóvel. Define-se  $Dom(\textit{Id\_Prop})=\{\text{Identificadores de proprietários}\}$ . Atributo de **tipo preciso**.

*Piscina*: Atributo que armazena se o imóvel tem piscina ou não. Define-se  $Dom(\textit{Piscina})=\{\text{verdadeiro, Falso, Unknown}\}$ . Atributo nebuloso de **tipo 2**.

*Banheiro*: Atributo que armazena o número de banheiros do imóvel. Define-se  $Dom(\textit{Banheiros})=\textit{Quantidades}$ . Atributo nebuloso de **tipo 1**.

O domínio *Quantidades* está definido sobre os números inteiros. Os valores que podem ser definidos sobre ele são: *números inteiros*, *Unknown*, *Undefined*, *Null*, *muitos*, *poucos*, *intervalos*. A figura 6.2 representa as funções que caracterizam os valores lingüísticos.

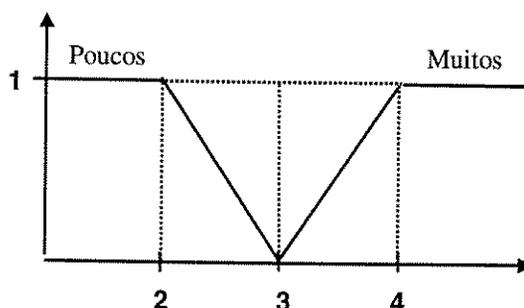


Figura 6.2. Valores Linguísticos do domínio *Quantidades*.

### Relação Proprietário.

*Idt\_Prop*: Atributo que identifica o proprietário. Constitui a chave primaria da relação proprietário. Define-se  $Dom(Id\_Prop) = \{identificadores\}$ . Atributo de **tipo preciso**.

*Nome*: Atributo que armazena o nome do proprietário. Define-se  $Dom(Nome) = \{Nomes\}$ . Atributo de **tipo preciso**.

### Relação Acabamento.

*Id\_Im*: Atributo que armazena o identificador do imóvel. Este identificador define a chave primaria da relação. Define-se  $Dom(Id\_Im) = \{Identificadores\ de\ imóveis\}$ . Atributo de **tipo preciso**.

*Piso*: Atributo que armazena as características do piso do imóvel. Define-se  $Dom(Piso) = \{granito, carpete, mármore, madeira, Unknown\}$ . Atributo nebuloso de **tipo 2**.

*Paredes*: Atributo que armazena as características das Paredes do imóvel. Define-se  $Dom(Paredes) = \{pintado, despintando, úmido, Unknwon\}$ . Atributo nebuloso de **tipo 2**.

*Ins\_Elec*: Atributo que armazena um critério sobre o estado de conservação da instalação elétrica do imóvel. Define-se  $Dom(Ins\_Elec) = Estado$ . Atributo nebuloso de **tipo 3**.

*Ins\_Hid*: Atributo que armazena um critério sobre o estado de conservação da instalação *hidráulica* do imóvel. Define-se  $Dom(Ins\_Hid)=Estado$ . Atributo nebuloso de **tipo 3**.

Relação Piscina.

*Id\_Iml*: Atributo que armazena o identificador do imóvel. Este identificador define a chave primaria da relação. Define-se  $Dom(Id\_Imóvel)=\{Identificadores\ de\ imóveis\}$ .

*Profundidade*: Atributo que armazena a profundidade media da piscina. Define-se  $Dom(Profundidade)=\{profunda, media, baixa, Unknown, números\ reais, intervalos, valores\ aproximados\}$ . Atributo nebuloso de **tipo 1**. A figura 6.3 representa a caracterização dos valores lingüísticos deste domínio.

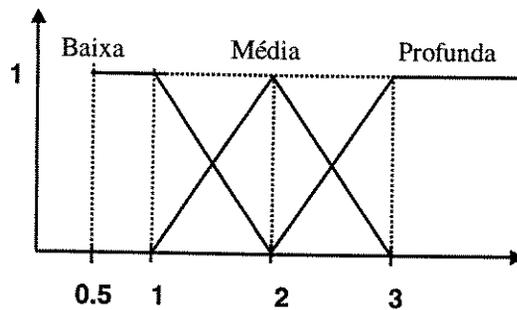


Figura 6.3. Valores Lingüísticos do domínio *Profundidade*.

*Área*: Atributo que armazena a área da piscina. Define-se  $Dom(Área)=\{grande, pequena, Unknown, números\ reais, intervalos, valores\ aproximados\}$ . Atributo nebuloso de **tipo 1**.

### Relação Quartos.

*Id\_Iml*: Atributo que armazena o identificador do imóvel. Este identificador define a chave primaria da relação junto com o atributo *Id\_Quartos*. Define-se  $Dom(Id\_Iml)=\{Identificadores\ de\ imóveis\}$ . Atributo de **tipo preciso**.

*Id\_Quartos*: Atributo que identifica o quarto. Este identificador define a chave primaria da relação junto com o atributo *Id\_Im*. Define-se  $Dom(Id\_Quartos)=\{Identificadores\ de\ quartos\}$ . Atributo de **tipo preciso**.

*Área*: Atributo que armazena a área do quarto. Define-se  $Dom(Área)=\{grande, pequena, Unknown, números\ reais, intervalos, valores\ aproximados\}$ . Atributo nebuloso de **tipo 1**. A figura 6.4 representa a caracterização dos valores lingüísticos deste domínio.

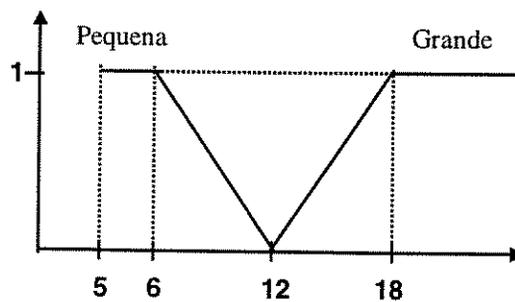


Figura 6.4. Valores Lingüísticos do domínio *Área* da relação **Quartos**.

### **6.3 Preenchimento dos valores das tabelas.**

Nesta seção são apresentadas as tabelas que formam o banco de dados *Imóveis* preenchidas de forma a gerar um exemplo que trabalhe com informações de caracter impreciso.

Id_Im	Id_Prop	Endereço	Aluguel	Acabamento	Tipo	...
01	03	Rua Santa Cruz 400, apto. 401, Cambuí.	400	Bom	Apartamento	...
02	02	Rua Itú 306, apto. 101, Bosque.	250	Péssimo	Kitnet	...
03	01	Rua Santo Antonio 209, apto. 204, Centro.	600	Regular	Unknown	...

...	Idade	Piscina	Banheiro	Elevadores	Área	Estado	Quartos
...	novo	True	2	2	80	Bom	1
...	aprox 10.	False	1	{0.8/Undefined, 1}	56	Péssimo	Undefined
...	[5,10]	Null	3	2	150	Bom	Unknown

Tabela 6.3. Relação Imóvel.

<b>Id_Prop</b>	<b>Nome</b>
01	Eduardo Huerta Yero
02	Sahudy Montenegro
03	Mercedez García Ruíz

Tabela 6.4. Relação **Proprietário**.

<b>Id_Im</b>	<b>Piso</b>	<b>Paredes</b>	<b>Ins_Elec</b>	<b>Ins_Hid</b>
01	carpete	pintado	bom	bom
02	madeira	úmido	regular	péssimo
03	carpete	pintado	bom	regular

Tabela 6.5. Relação **Acabamento**.

<b>Id_Im</b>	<b>Id_Quartos</b>	<b>Área</b>
01	01	aprox. 16
03	01	grande
03	02	aprox. 25

Tabela 6.6. Relação **Quartos**.

<b>Id_Im</b>	<b>Profundidade</b>	<b>Área</b>
01	profunda	Grande

Tabela 6.7. Relação **Piscina**.

#### 6.4 Armazenamento do meta-conhecimento.

A tabela 6.8 armazena todos os atributos do banco de dados *Imóveis* que possuem tratamento nebuloso. Na tabela para cada atributo é armazenado a relação a que pertence e o domínio sobre o qual esta definido.

TableName	FieldName	Domain
Imóvel	acabamento	Acabamento
Imóvel	Tipo	Tipo
Imóvel	Idade	Idades
Imóvel	Piscina	Piscina
Imóvel	Elevadores	Quan_Elevador
Imóvel	Área	Área_Imóvel
Imóvel	Estado Geral	Estados
Imóvel	Quartos	Quantidades
Imóvel	Banheiro	Quantidades
Acabamento	Piso	Piso
Acabamento	Paredes	Paredea
Acabamento	Ins_Electrica	Estados
Acabamento	Ins_Hidraulica	Estados
Piscina	Profundidade	Profundidades
Piscina	Área	area_Piscina
Quartos	Área	area_Quarto

Tabela 6.8. Tabela para definição dos atributos nebulosos.

A tabela 6.9 armazena o tipo de tratamento nebuloso que recebe cada domínio definido no banco de dados.

Domain	DomainType	IniDomain	EndDomain	Jump
Idades	Tipo 1	0	400	1
Acabamento	Tipo 4	Nulo	Nulo	Nulo
Quantidades	Tipo 1	0	20	1
Quan_Elevador	Tipo 1	1	10	1
Área_Imóvel	Tipo 1	7	1000	1
Estados	Tipo 3	Nulo	Nulo	Nulo
Profundidades	Tipo 1	0.5	15	0.5
area_Piscina	Tipo 1	7	100	1
area_Quarto	Tipo 1	5	100	1
Tipo	Tipo 2	Nulo	Nulo	Nulo
Piscina	Tipo 2	Nulo	Nulo	Nulo
Piso	Tipo 2	Nulo	Nulo	Nulo
Paredes	Tipo 2	Nulo	Nulo	Nulo

Tabela 6.9. Tabela para a definição dos domínios nebulosos.

A tabela 6.10 armazena a definição das funções usadas para descrever os valores lingüísticos, os operadores relacionais e os operadores de conjunto.

Func-Name	FuncType	Exp_Mat	Parser
trapezio	Rótulo	$\begin{cases} x - \frac{a}{m-a} & \text{se } x \in [a, m] \\ 1 & \text{se } x \in [m, n] \\ b - \frac{x}{b-n} & \text{se } x \in [n, b] \end{cases}$	$x, a, -, m, a, -, /, 1, b, x, -, b, n, -, /, \min, 0, \max$

Tabela 6.10. Tabela para a definição das funções.

Á tabela para a definição dos pares de funções AND e OR não foi preciso adicionar nenhuma informação. As funções para estes operadores serão as definidas por default, min e max, assim como para os operadores relacionais.

A tabela 6.11 armazena os parâmetros que descrevem as funções de pertinência dos valores lingüísticos definidos. Para cada rótulo é armazenado o domínio ao qual pertencem, o tipo da função que o descreve e os valores dos parâmetros da função.

A tabela 6.12 armazena as relações de proximidade definidas nos domínios de tipo 3.

Domain	Label	FuncId	a	m	n	b	k
Idades	novo	trapezio	0	0	3	5	
Idades	mediana	trapezio	3	5	5	7	
Idades	velho	trapezio	5	7	400	400	
Quantidades	poucos	trapezio	0	0	2	3	
Quantidades	muitos	trapezio	3	4	20	20	
Profundidades	baixa	trapezio	0.5	0.5	1	2	
Profundidades	média	trapezio	1	2	2	3	
Profundidades	profunda	trapezio	2	3	15	15	
area_Piscina	grande	trapezio	50	90	100	100	
area_Piscina	pequena	trapezio	7	7	10	50	
area_Quarto	grande	trapezio	12	18	50	50	
area_Quarto	pequena	trapezio	5	5	6	12	

Tabela 6.11. Tabela para a definição dos rótulos em domínios numéricos.

Domain	Value1	Value2	Grade
Estados	Bom	Excelente	0.8
Estados	Bom	regular	0.2
Estados	Bom	Péssimo	0.1
Estados	Excelente	Regular	0.1
Estados	Excelente	Péssimo	0
Estados	Regular	Péssimo	0.8

Tabela 6.12. Tabela para a definição das relações de similaridade.

TableName	Domain	Value	Expr	...
Acabamento	Acabamento	Bom	...	...
Acabamento	Acabamento	Regular	...	...
Acabamento	Acabamento	péssimo	...	...

...	<b>Parser</b>
...	<i>Paredes,umidas,=,Ins_Elec,pessima,=,AND,Ins_Hid,pessima,=,AND,0.8,≥</i>
...	<i>Ins_Elec,bom,=,AND,Ins_Hid,regular,=,AND,0.8,≥</i>
...	<i>Paredes,pintadas,=,Ins_Elec,bom,=,AND,Ins_Hid,bom,=,AND,≥,0.8</i>

Tabela 6.13. Tabela para a definição dos conceitos complexos.

Na tabela 6.13, está representada a definição dos conceitos complexos a partir de expressões lógicas nebulosas. Para cada valor, é armazenado o domínio a que pertence, a relação onde se definem os atributos que formam a expressão e a

expressão lógica que o caracteriza. O atributo *Expr* armazena a expressão lógica nebulosa mostrada na tabela 6.1.

### 6.5 Representação física dos atributos nebulosos nas tabelas.

Na relação **Imóvel**, existem nove atributos com tratamento impreciso, mas só oito são representados fisicamente, o que é feito na tabela 6.14.

...	Tipo	Idade	Piscina	Banheiro	Elevadores	...
...	Crisp, Apartamento	Label, novo	Crisp, true	Crisp, 2	Crisp, 2	...
...	Crisp, Kitnet	aprox, 10, 6	Crisp, false	Crisp, 1	PossUndNum, 0.8, Crisp, 1	...
...	Unknown	interval, 5, 10	Null	Crisp, 3	Crisp, 2	...

...	Área	Estado	Quartos
...	Crisp, 80	Cirsp, Bom	Crisp, 1
...	Crisp, 56	Cirsp, Ruim	Undefined
...	Crisp, 150	Cirsp, Bom	Unknown

Tabela 6.14. Representação física dos atributos nebulosos da relação **Imóvel**.

Na relação **Acabamento**, existem quatro atributos nebulosos e são armazenados como mostra a tabela 6.15.

...	Piso	Paredes	Ins_Elec	Ins_Hid
...	Crisp, carpete	Crisp, pintado	Crisp, bom	Crisp, bom
...	Crisp, madeira	Crisp, úmido	Crisp, regular	Crisp, péssimo
...	Crisp, carpete	Crisp, pintado	Crisp, bom	Crisp, regular

Tabela 6.15. Representação física dos atributos nebulosos da relação **Acabamento**.

Na relação **Quartos**, o atributo *Área* possui tratamento impreciso e a tabela 6.16 mostra como foram armazenados os valores.

...	Área	...
...	aprox, 16, 6	...
...	Label, grande	...
...	aprox, 25, 8	...

Tabela 6.16. Representação física dos atributos nebulosos da relação **Quartos**.

Na relação **Piscina** existem dois atributos com tratamento impreciso como mostra a tabela 6.17.

...	Profundidade	Área
...	Label, profunda	Label, Grande

Tabela 6.17. Representação física dos atributos nebulosos da relação **Piscina**.

## 6.6 Exemplo de consulta.

A linguagem de consulta, proposta no capítulo 4, é utilizada nesta seção sobre o banco de dados *Imóveis*.

Suponha-se que quer obter todos os imóveis com quartos grandes com grau de satisfação 0.7. A partir desta consulta o SQL-Nebuloso estaria definido como:

Select Id\_Im, Id\_Quartos

From Quartos

Where area = grande With 0.7

Para procurar as tuplas que cumpram esta condição deve ser definido o operador =. Para o exemplo será o clássico.

$$\mu_{=} (d, d') = \begin{cases} 1 & d = d' \\ 0 & \text{outro caso} \end{cases}$$

Com isto, é calculado o grau de compatibilidade do atributo *área*,  $c_{area_j}$ , e da tupla,  $c_j$  com  $j = 1..3$ , obtido a partir da aplicação da expressão a cada tupla. Este valores são calculados segundo a expressão 1.

$$c_{area_j} = \sup_{(d \times d') \in area \times area} \left\{ \min(\mu_{=} (d, d'), \mu_{grande} (d), \pi_{area_j} (d')) \right\} \quad (1)$$

$$c_j = c_{area_j}$$

O atributo de compatibilidade do atributo *área* na primeira tupla é calculado como mostra a equação a seguir:

$$c_{area_1} = \sup_{d \times d' \in area \times area} \left\{ \min(\mu_{grande} (d), \pi_{aprox.16} (d')) \right\}$$

A figura 6.5 representa este cálculo graficamente.

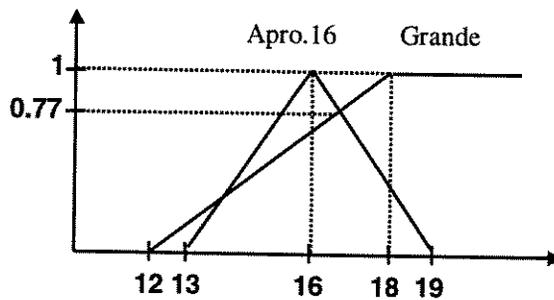


Figura 6.5. Cálculo do  $c_{area_1} = 0.77$ .

Na segunda tupla  $c_{area_2} = 1$ .

A figura 6.6 representa o cálculo dos grau de satisfação na terceira tupla.

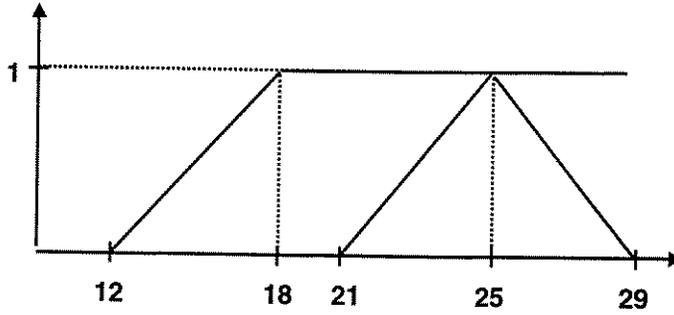


Figura 6.6. Cálculo do  $c_{area_3} = 1$

A tabela 6.18 representa a relação obtida como resposta da consulta.

Id_Im	Id_Quartos	Área	$C_{area}$	C
01	01	aprox.16	0.77	0.77
03	01	grande	1	1
03	02	aprox.25	1	1

Tabela 6.18. Relação Resposta.

# Capítulo 7

## Conclusões

---

Ao longo do desenvolvimento deste trabalho, foram examinados varios modelos de bancos de dados nebulosos e feitas comparações entre ele. Finalmente, optou-se por um modelo que:

- Estendesse o Modelo Relacional de Bancos de Dados.
- Permitisse representar a incerteza a todos os níveis que ela se manifesta.
- Considerasse na sua definição os modelos de bancos de dados já definidos.

A opção de “Estender o Modelo Relacional de Bancos de Dados” ao invés de propor um modelo próprio, sem restrições, permitiu a definição de uma estrutura adequada aos gerenciadores convencionais existente no mercado. Isto foi importantíssimo, porque tornou viável o trabalho de implementação, com todos os recursos visuais do Delphi 2.0. É importante notar que, a definição de classes feitas para expandir o Database Engine do Delphi, é facilmente comparável ao C-Builder da Borland, que tem como linguagem de suporte o C orientado a objeto.

“Permitir a incerteza a todos os níveis que ela se manifesta”, foi uma opção levada até o ponto em que os operadores das operações lógicas pudessem ser redefinidos a cada sessão de consultas ao banco de dados. Sem dúvida, este recurso é extremadamente

importante para aqueles pesquisadores que fazem análise de sensibilidade das suas consultas ao banco de dados, em função dos operadores lógicos usados nas consultas.

“Considerar na sua definição os modelos de bancos de dados já definidos”, foi uma decisão decorrente da análise comparativa entre os modelos relacional possibilístico, relacional nebuloso e de unificação a partir de relações de similaridade. Isto resultou num modelo que incorpora todos os tipo de dados nebulosos propostos por aqueles modelos, e trata a incerteza ao nível de atributo e de tuplas. Obviamente, isto representa uma oferta maior de recursos numa única ferramenta.

Consideramos que as opções adotadas no que se refere as características principais do modelo, bem como as opções de implementação, compõem um conjunto coerente, tornam o simples e de implementação viável, para uma tese de mestrado.

### **Extensões do trabalho.**

Como futuras extensões do trabalho propõe-se:

- Extensão da linguagem de SQL-Nebuloso para incluir todas as instruções de um SQL-Convencional redefinidas para o *Modelo Integrado Bancos de Dados Nebulosos*.
- Extensão da hierarquia de Delphi para incluir dentro da sua funcionalidade instruções do SQL-Nebuloso definido no capítulo 4.
- Incluir a possibilidade de executar consultas nebulosas sobre dados precisos armazenados em relações convencionais.
- Implementação de classes para a visualização dos tipos de atributos para com isto incluir os tipos nebulosos.

# *Bibliografia*

---

- [Bo88] P. Bosc, M. Galibourg, G. Hamon, Fuzzy querying with SQL: Extensions and implementation aspects, *Fuzzy Sets and Systems* 28, 1988, 333-349.
- [BP82] B. P. Buckles and F. E. Petry, A fuzzy representation of data for relational databases, *Fuzzy Sets and Systems* 7(3), 1982, 213-226.
- [BP83] B. P. Buckles and F. E. Petry, Information-theoretical characterization of fuzzy relational databases, *IEEE Trans. on SMC* SMC-13(1), 1983, 74-77.
- [BP84] B. P. Buckles and F. E. Petry, Extending the fuzzy databases with fuzzy numbers, *Inf. Sci.* 34, 1984, 145-155.
- [BP89] B.P. Buckles and F. E. Petry, A Domain Calculus for Fuzzy Relational Databases, *Fuzzy Sets and Systems* 29, 1989, 327-340.
- [BZ84] J.F. Baldwin, S.Q. Zhou, A fuzzy relational inference language, *Fuzzy Sets and Systems* 14, 1984, 155-174.
- [Co70] E. F. Codd, A Relational Model of Data for large shared Data Banks, *Commun ACM* 13, 1970, 377-387.
- [Co79] E. Codd, Extending the Database Relational model to capture more meaning, *ACM Trans. On Database Sys.*, 4, 156-174, 1979.
- [DA95] Database Application developer's Guide, Borland Delphi for Windows 95 & Windows NT, Borland International, Inc., 1996.

- [DM89] J. T. Dockery, E. Murray, Fuzzy linguistic databases: an application, *Information Systems* 14 (6), 1989, 501-505.
- [EN94] R. Elmasri, S. B. Navathe, *Fundamentals of Database Systems Second Edition*, The Benjamin/Cummings Publishing Company, Inc. 1994.
- [Ga77] J. Grant, Null values in a relational database, *Information Processing Letters* 6 (5), 1977, 156-157.
- [Ka75] A Kaufmann, *Introduction to the Theory of Fuzzy Subsets*, Academic Press, 1975.
- [Lip79] W. Lipski, On semantic issues connected with incomplete information databases, *ACM Transactions on databases Systems* 4 (3), 1979, 262-296.
- [Li90] D. Li, D. Liu, *A Fuzzy Prolog Databases Systems*, Jonh Wiley & Sons. Inc., 1990.
- [Ma81] L. F. J. Maia, *Caracterização e Reconhecimento de conceitos*, Tese de Doutorado, Faculdade de Engenharia Elétrica e Computação, UNICAMP, 1981.
- [Me94] J. M. Medina, *Bases de datos relacionales difusas: Modelo teórico y astectos de su implementacion*, Ph. Thesis. University of Granada, 1994.
- [Me94a] J. M. Medina, O. Pons, M. A. Vila, GEFRED. A generalized model of fuzzy relational databases. Ver. 1.1, *Information Science* 76, 1994, 87-109.
- [Me95] M. Medina, M. A. Vila, J.C. Cubero, O. Pons, Toward the implementation of a generalized fuzzy relational databases model, *Fuzzy sets and systems* 75, 1995, 273-289.
- [Mo90] A. Motro, Imprecision and incompleteness in relational databases: survey, *Information and software technology* 32 (9), 1990, 579-588.

- [Pe96] F. E. Petry, *Fuzzy Databases Principles and Applications*, Klumer Academic Publishers, 1996.
- [PL84] H. Prade, Lipski's approach to incomplete information databases restated and generalized in the setting of Zadeh's possibility theory, *Inf. Systems* 9(1), 1984, 27-42.
- [Po94] O. Pons, *Representación lógica de bases de datos difusas. Fundamentos teóricos e implementación*, PH. Thesis. University of Granada, 1994.
- [PT84] H. Prade, C. Testemale, Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries, *Inf. Sci.* 34(2), 1984, 107-141.
- [Ru95a] J. Rumbaugh, OMT: The object model, *JOOP*, January 1995, 21-28.
- [Ru95b] J. Rumbaugh, OMT: The dynamic model, *JOOP*, February 1995, 6-29.
- [SM89] S. Shenoj and A. Melton, Proximity relations in the fuzzy relational database model, *Fuzzy Sets and Systems* 31, 1989, 285-296.
- [SM90] S. Shenoj and A. Melton, An extended version of fuzzy relational database model, *Inf. Sci.* 52, 1990, 35-52.
- [Ta77] V. Tahini, A conceptual framework for fuzzy query processing - A step toward very intelligent database systems, *Information processing & Management* 13, 1977, 289-303.
- [Ta91] Y. Takahashi, A fuzzy query language for Relational Databases, *IEEE Trans. on SMC* 21(6), 1991, 1576-1579.
- [UI82] J. D. Ullman, *Principles of Database systems Second Edition*, Computer Science Press, Inc. 1982.
- [Um83] M. Umamo, Retrieval from fuzzy database by fuzzy relational algebra, *Knowledge Information Systems, Medical Applications*, 1, 1983.

- [UM89] M. Umamo, M. Mizumoto, FSTDS System: A fuzzy-set manipulation System, *Information Science* 14, 1989, 115-159.
- [Va89] R. Vandenberghe, A. V. Schooten, R. De Caluwe, E.E. Kerre, Some practical aspects of fuzzy databases techniques: an example, *Information Systems* 14 (6), 1989, 465-472.
- [WF97] W. Pedrycz, F. Gomide, *An Introduction to Fuzzy Sets: Analysis And Design*, 1997 (a ser publicado).
- [YF94] R. R. Yager, D. P. Filev, *Essentials of Fuzzy Modeling and Control*, John Wiley & Sons, Inc, 1994.
- [Za65] L.Zadeh, Fuzzy Sets, *Information and Control* 8, 1965, 338-353.
- [Za75a] L. Zadeh, The Concept of a Linguistic Variable and its Application to Approximate Reasoning - I, *Information Sciences* 8, 1975, 199-249.
- [Za75c] L. Zadeh, The Concept of a Linguistic Variable and its Application to Approximate Reasoning - III, *Information Sciences* 9, 1975, 43-80.
- [Za71] L. Zadeh, Similarity relations and fuzzy ordering, *Information Science* 3, 1971, 177-200.
- [Za78] L. Zadeh, Fuzzy sets as basis for a theory of possibility, *Fuzzy Sets and Systems* 1, 1978, 3-28.
- [Ze89] M. Zemankova, FILIP: A fuzzy intelligent information system with learning capabilities, *Information Science* 14 (6), 1989, 473-486.
- [ZK85] M. Zemankova, A. Kandel, Implementing imprecision in information systems, *Inf. Sci.* 37, 1985, 107-141.