

INFORMÁTICA

83



**XVI CONGRESSO
NACIONAL DE
INFORMÁTICA**

SÃO PAULO OUTUBRO 83

SIBER (SISTEMA INTEGRADO BASEADO NO MODELO E-R)

Rubens Nascimento Melo
Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
CEP 22453 - Rio de Janeiro - RJ

José Geraldo Silva
Departamento de Processamento de Dados
EMBRATEL - Empresa Brasileira de Telecomunicações S.A.
R. Senador Pompeu, nº 119/912 - DPD.1/4
CEP. 20221 - Rio de Janeiro - RJ

Paulo Sérgio Simões de Araujo
Departamento de Processamento de Dados
EMBRATEL - Empresa Brasileira de Telecomunicações S.A.
R. Senador Pompeu, nº 119/911 - DPD.1/5
CEP. 20221 - Rio de Janeiro - RJ

Palavras-chaves: Projeto-de-BD; SIBER; SDDPUC; Dicionário-de-Dados; INTEGRAÇÃO-DE-VISÕES.

Resumo

SIBER é um ambiente de desenvolvimento de Sistemas utilizando metodologia e ferramentas automatizadas baseadas no modelo de Entidades e relacionamentos. Nesse ambiente o processo de projeto de estrutura de Banco de Dados pode ser feito em 2 (dois) estágios.

No primeiro estágio é produzida uma visão macro do esquema conceitual utilizando-se o modelo de Entidades e Relacionamentos e suas extensões, recomendando-se o uso de facilidades para registro de informações sem preocupação com dependências de dados.

No segundo estágio os detalhes são preenchidos e verificações são efetuadas, utilizando-se uma representação gráfica da estrutura do Banco de Dados como ferramenta de apoio um sistema de dicionário de Dados Flexíveis.

1. Introdução

A análise da realidade do mundo do usuário é feita preliminarmente de maneira intuitiva e informal, e o emprego de ferramentas que possibilitem uma melhor interpretação e representação deste mundo real, é de fundamental importância na comunicação entre usuários e projetistas de Banco de Dados, tornando o processo de análise iterativo. Sem que isto ocorra, de pouco adiantaria a utilização de ferramentas formais, tão importantes para uma descrição consistente do Banco de Dados, a nível de Projeto Lógico. Um processo já comum de projeto de Banco de Dados é começar por uma descrição informal das entidades e relacionamentos relevantes no sistema a ser desenvolvido. Este processo depende muito da intuição e da maneira como os usuários percebem a realidade. O resultado pode ser esquematizado num diagrama de entidades e relacionamentos inicial. Esse modelo inicial precisa ser refinado de forma a proporcionar a eliminação de inconsistências cometidas por falha de interpretação dos dados inseridos num determinado contexto real. Para esse fim tem surgido uma série de proposições para desenvolvimento de sistemas de Banco de Dados. Algumas delas propõem o mapeamento do DER para um modelo relacional, permitindo a aplicação da teoria de normalização e de dependências funcionais para obter um conjunto de relações que adequadamente modelem o Banco de Dados desejado.

Outros enfatizam ferramentas gráficas às vezes não muito ricas nem formais para expressar certas informações importantes, e apelam à intuição dos projetistas para complementar as deficiências do método.

Um método que já tem sido assimilado pelos administradores de dados e de Banco de Dados de diversas organizações, e até mesmo já possui ferramentas automatizadas (DBDA) e (Dgn) é o método de estruturação canônica. Esse é um método do tipo acima mencionado. Entretanto, devido a sua facilidade de aceitação, seria interessante contribuir para melhorar o método. Uma das tentativas de formalização de certos passos que são intuitivos na representação

canônica é apresentada em Vetter (Vet) e aplicados em (JGS). Nesse artigo procuramos apresentar um enfoque de projeto de Banco de Dados que acrescente ao método simples da modelagem das entidades e relacionamentos e de estruturação canônica discutida por (RHu) e (JM), certos procedimentos sistemáticos sugeridos em (Vet) na obtenção de uma estrutura adequada para a base de dados.

Embora reconheçamos que há certos aspectos relevantes da especificação conceitual de um Banco de Dados, tais como as restrições de integridade estáticas e dinâmicas, bem como certas dependências mais sutis entre os dados do que as que utilizamos nesse método, acreditamos que esse trabalho possa contribuir na orientação dos projetos da estrutura de bases de dados que já vêm sendo desenvolvidos informalmente nas empresas, através do Modelo de Entidades e Relacionamentos e do método canônico.

Este artigo está assim dividido:

- Na seção 2 discutimos os aspectos de abordagem propostos e o posicionamento em relação ao ambiente de desenvolvimento de sistemas SIBER existente na PUC;
- Na seção 3, introduzimos o sistema de Dicionário de Dados Flexível, o SDDPUC, uma das ferramentas do SIBER que apoia o processo de análise e documentação;
- Na seção 4 mostramos a aplicação do método sistemático a um caso em estudo;
- Na seção 5, resumimos os procedimentos básicos que devam ser adotados no projeto de estruturas de Banco de Dados;
- Na seção 6, apresentamos as conclusões e sugestões para desenvolvimento de futuros trabalhos.

2. SIBER - Um Ambiente de Desenvolvimento de Sistemas

O SIBER (Sistema Integrado Baseado em Entidades e Relacionamentos) é um ambiente de desenvolvimento de sistemas em fase de crescimento. Este ambiente é composto de ferramentas e metodologias que auxiliam no projeto de Banco de Dados, e que atualmente podem ser vistos como elementos que capturam do mundo real, os dados necessários a serem implementados, apoiados no MER - Modelo de Entidades e Relacionamentos.

Propomos, através deste trabalho, que o enfoque de projeto de estrutura de Banco de Dados seja feito através de 2 (dois) métodos de abordagem: um que permite obter uma visão macro do esquema conceitual, e outro que reflita os detalhes e refinamentos necessários da visão macro obtida, objetivando alcançar o esquema conceitual mínimo.

Chamamos de 1º estágio, ao processo de obtenção da visão macro do mundo real a ser modelado. Para tal, utilizamos-nos do modelo de Entidades e relacionamentos e suas extensões (Web), (SNF), dentre outros.

Denominamos de 2º estágio, ao processo de otimização do MER, utilizando para tanto as formas de representação gráfica de «bolhas» (bubble-charts) e de algoritmos e de procedimentos para determinação do esquema conceitual mínimo, que possibilite a estruturação do Banco de Dados.

Tanto no 1º como no 2º estágio, utilizamos-nos de sistemas, subsistemas, ferramentas e metodologias já desenvolvidas e implantadas dentro do SIBER, destacando-se dentre estes o SDDPUC, que é um sistema de Dicionário de Dados Flexível da PUC.

A forma como o SDDPUC pode apoiar no processo de análise e documentação está descrita na próxima seção. Procuramos, através de um DFD (Diagrama de Fluxo de Dados) - (Gan) -, demonstrar como poderia ser o novo ambiente SIBER.

3. Sistema de Dicionário de Dados Flexível

3.1 A Utilização do SDDPUC - Sistema de Dicionário de Dados da PUC
Devido ao tamanho e complexidade dos DD, estes normalmente são gerenciados por sistemas especialmente desenvolvidos para estas funções, sendo conhecidos como Sistemas de Dicionário de Dados (SDD).

O SDD é um sistema manual ou automático, que propicia meios para manipular as informações armazenadas em um DD, centralizando e controlando as informações sobre os recursos de dados existentes, processados ou não em um ambiente de PD, descrevendo seus atributos e relacionamentos, indicando responsabilidades, funções, usos e origens envolvidos no fluxo de informações, suportando a descrição dos dados de interesse de forma global e centralizada. O SDD é uma ferramenta ambiciosa, que interfere em todas as etapas do ciclo de análise, projeto e implantação dos sistemas, apoiando na análise das

estruturas de dados, na formulação de um modelo conceitual, na análise funcional, no projeto lógico e físico do BD, na análise, implantação e manutenção dos sistemas de aplicação, no apoio à operação dos sistemas e na avaliação dos recursos de dados.

Os primeiros SDD foram desenvolvidos para atender as necessidades de outros softwares (SBGD's, compiladores, etc...), sendo conhecidos como Diretório de Dados, orientados para o nível interno da arquitetura ANSI/X3/SPARC. (SPARC) Os DD evoluíram para um enfoque gerencial de forma que pudessem suportar a parte do modelo conceitual referente aos dados armazenados em um ambiente de PD.

Com os Dicionários de Dados de Estrutura Flexível (DDF) é possível modelar todo o Sistema de Informações (SI), independente da localização da informação. Com o uso dos DDF modelando o nível externo, em conjunto com módulos adicionais acoplados ao SDD, pode-se desenvolver ferramentas de apoio ao desenvolvimento dos sistemas aplicativos.

O SDDPUC é um DDF, de uso primário, com enfoque gerencial, orientado para uso interativo, ativo nos componentes do sistema SIBER (Sistema Integrado Baseado em Entidades e Relacionamentos) em desenvolvimento no Departamento de Informática da PUC e usa uma linguagem quase natural semelhante ao Português.

A principal característica do sistema é a sua flexibilidade em admitir novos tipos de objetos, seus relacionamentos e atributos.

A definição dos novos tipos, relacionamentos e atributos é feita no Meta-Dicionário através do SMMD (Subsistema de Manipulação do Meta-Dicionário) através dos comandos: **AUTORIZAR** e **DES Autorizar**.

A manipulação do DD é feita através do SMDD (Subsistema de Manipulação do Dicionário de Dados) através dos comandos: **INCLUIR**, **EXCLUIR**, **LIGAR**, **DESLIGAR**, **DESCREVA** e **EXPANDIR**.

A recuperação das informações, por sua vez, é feita através do SEI (Subsistema de Extração de Informação) através dos comandos: **INFORME**, **EXAMINE**, **SELECIONE**, **RECUPERE** e **LISTE**.

Além destas facilidades o Sistema oferece um esquema de senhas (**MESTRE**, **DONO** e **USUÁRIO**) impondo restrições nos acessos ao Dicionário.

O modelo conceitual das informações do meta-dicionário pode ser observado no diagrama E-R apresentado na figura 3.1.

A disposição deste modelo na linguagem do sub-sistema de manipulação do meta-dicionário (SMMD) é a seguinte:

AUTORIZAR OBJETO ITEM;
AUTORIZAR RELACIONAMENTO ITEM ASSOCIA E ITEM ASSOCIADO;
AUTORIZAR ATRIBUTO VIEW;
AUTORIZAR ATRIBUTO TIPO VALIDANDO 1 M;
AUTORIZAR ATRIBUTO CHAVE VALIDANDO PRIM CONC;
AUTORIZAR ATRIBUTO PESO VALIDANDO 1-10;
AUTORIZAR ATRIBUTO CATEGORIA VALIDANDO ESSENCIAL;
AUTORIZAR LIGAÇÃO DE TIPO COM RELACIONAMENTO ITEM ASSOCIA ITEM;
AUTORIZAR LIGAÇÃO DE VIEW COM RELACIONAMENTO ITEM ASSOCIA ITEM CRITICANDO EXISTÊNCIA;
AUTORIZAR LIGAÇÃO DE PESO COM RELACIONAMENTO ITEM ASSOCIA ITEM;
AUTORIZAR LIGAÇÃO DE CHAVE COM RELACIONAMENTO ITEM ASSOCIA ITEM;
AUTORIZAR LIGAÇÃO DE CATEGORIA COM RELACIONAMENTO ITEM ASSOCIA ITEM.

Os algoritmos a serem implementados para determinação do fecho transitivo e redução das relações elementares redundantes podem ser feitas usando-se os procedimentos computacionais apresentados por Vetter [Vet] e utilizados por [JGS].

Como um "gráfico de bolhas", [HNC], nada mais é que um **grafo dirigido**, o mesmo pode ser representado por uma matriz binária de conectividade, cujos elementos indicam a presença ou não de uma ligação entre um nó-origem e um nó-destino.

Para que este programa possa acessar as informações das "user-views" armazenados no SDDPUC, devem ser utilizados os procedimentos indicados em [Lob].

3.2 Exemplo do Uso do SDDPUC no Projeto de estrutura de BD

O exemplo a seguir se baseia naquele apresentado por Raver [RHu] onde são apresentadas cinco "user-views" de uma empresa de transportes que carrega seus caminhões com produtos para despachá-los para vários clientes. Muitas viagens são feitas por dia e cada viagem é feita por um certo tipo de veículo. Para cada componente de um produto é dado número de embalagem. Numa viagem específica, todas as embalagens de um certo cliente são agrupadas em um único número de remessa. Partindo do pressuposto que um DER (Diagrama de Entidades e Relacionamentos) tivesse sido obtido como resultado da abordagem do 1º estágio, poderíamos obter no segundo estágio as "user-views".

A descrição destas "user-views" na linguagem de manipulação do Dicionário de Dados (SMDD) usando o modelo definido na figura 3.2 é o seguinte:

INCLUIR ITEM DATA;
INCLUIR ITEM VIAGEM;
INCLUIR ITEM VEÍCULO;
INCLUIR ITEM VOLUME;
INCLUIR ITEM PESO;
INCLUIR ITEM CLIENTE;
INCLUIR ITEM PRODUTO;
INCLUIR ITEM EMBALAGEM;
INCLUIR ITEM REMESSA;
INCLUIR ITEM MFG;
INCLUIR ITEM VOL-EMBALAGEM;
INCLUIR ITEM PESO-EMBALAGEM;
LIGAR DATA ASSOCIA VIAGEM COM TIPO=M VIEW=V1 CHAVE=INIC;
LIGAR VIAGEM ASSOCIA VEÍCULO COM VIEW=V1;
LIGAR VIAGEM ASSOCIA VOLUME COM VIEW=V1;
LIGAR VIAGEM ASSOCIA PESO COM VIEW=V1;

-
-
-

LIGAR EMBALAGEM ASSOCIA MFG COM VIEW=V5;
LIGAR EMBALAGEM ASSOCIA VOL-EMBALAGEM COM VIEW=V5;
LIGAR EMBALAGEM ASSOCIA PESO-EMBALAGEM COM VIEW=V5.

O resultado da integração destas cinco visões com indicações dos links que devem ser eliminados é apresentada na figura 3.3

O resultado desta integração em um esquema canônico para o projeto lógico do banco de dados da empresa de transporte pode ser observado na figura 3.4

A obtenção dessa estrutura final pode ser sistematizada utilizando os procedimentos descritos em [JGS]Cap. 3 - e tendo como ferramenta de apoio o SDDPUC, como veremos na próxima seção.

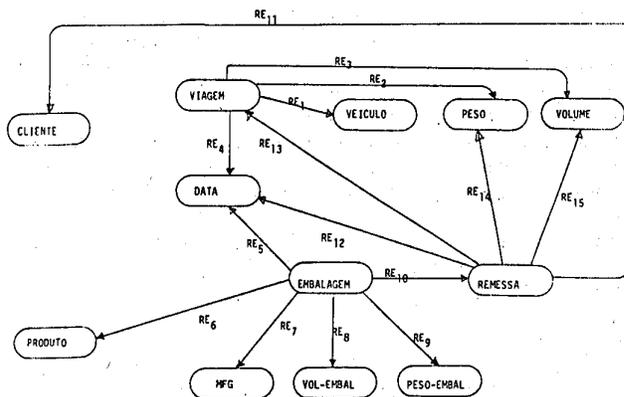
5 - Aplicação do Método Sistemático ao Caso em Estudo

Do dígrafo da fig. 3.3 que reúne as visões dos usuários, eliminaremos inicialmente as associações M:M que podem ser obtidas por outros caminhos. Obtemos assim o dígrafo contendo somente as relações elementares, i.e., relações irredutíveis (normalizadas).

Obs.: Ao eliminarmos as associações M:M, precisamos manter a integridade de nosso dicionário de dados desativando as ligações anteriormente feitas no SDDPUC. Esta desativação se faz pelo comando **DESLIGAR**.

No nosso caso-exemplo:

DESLIGAR DATA ASSOCIADO VIAGEM COM TIPO = M VIEW = V1;
DESLIGAR CLIENTE ASSOCIADO VIAGEM COM TIPO = M VIEW = V2;
DESLIGAR CLIENTE ASSOCIADO PRODUTO COM TIPO = M VIEW = V3;
DESLIGAR PRODUTO ASSOCIADO EMBALAGEM COM TIPO = M VIEW = V3;
DESLIGAR VIAGEM ASSOCIADO CLIENTE COM TIPO = M VIEW = V4;
DESLIGAR CLIENTE ASSOCIADO EMBALAGEM COM TIPO = M VIEW = V4
DESLIGAR REMESSA ASSOCIADO EMBALAGEM COM TIPO = M VIEW = V5.



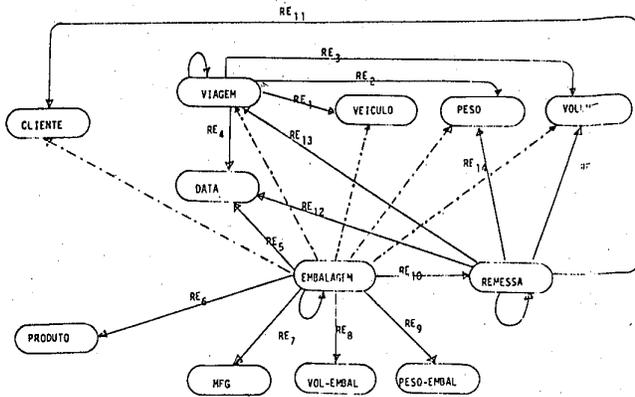
O dígrafo nos fornece a matriz de conectividade MC de dimensão 12x12

	CLIENTE	VIAGEM	DATA	VEICULO	PESO	VOLUME	PRODUTO	EMBAL.	REMESSA	MFG	VOL-EMB	PES-EMB
CLIENTE												
VIAGEM			1	1	1	1						
DATA												
VEICULO												
PESO												
VOLUME												
PRODUTO												
EMBAL.				1			1		1	1	1	1
REMESSA	1	1	1		1	1						
MFG												
VOL-EMB												
PES-EMB												

Aplicando os procedimentos descritos em [Vet] e [JGS] para determinação do "Fecho Transitivo", obtemos a matriz:

	CLIENTE	VIAGEM	DATA	VEICULO	PESO	VOLUME	PRODUTO	EMBALAGEM	REMESSA	MFG	VOL-EMBAL	PESO-EMBAL
CLIENTE												
VIAGEM		1	1	1	1							
DATA												
VEICULO												
PESO												
VOLUME												
PRODUTO												
EMBALAGEM	1	1	1	1	1	1	1	1	1	1	1	1
REMESSA	1	1	1	1	1	1						
MFG												
VOL-EMBAL												
PESO-EMBAL												

A partir desta matriz de Fecho Transitivo obtemos o digrafo:



O procedimento do "fecho transitivo" produz as dependências:

DEPENDENCIA		COMPOSICAO		SIGNIFICADO SEMANTICO
Nº	DE - PARA	Nº	OBTIDA ATRAVES DE:	
1	EMBAL - CLIENTE	1	EMBAL - REMESSA - CLIENTE	N
2	EMBAL - VIAGEM	2	EMBAL - REMESSA - VIAGEM	N
3	EMBAL - VIAGEM	3	EMBAL - REMESSA - VIAGEM - VEICULO	N
		4	EMBAL - VIAGEM - VEICULO	N
4	EMBAL - PESO	5	EMBAL - REMESSA - VIAGEM - PESO	N
		6	EMBAL - VIAGEM - PESO	N
		7	EMBAL - REMESSA - PESO	N
5	EMBAL - VOLUME	8	EMBAL - REMESSA - VIAGEM - VOLUME	N
		9	EMBAL - REMESSA - VOLUME	N
		10	EMBAL - VIAGEM - VOLUME	N
6	REMESSA - VEICULO	11	REMESSA - VIAGEM - VEICULO	N

Observação: A análise dos caminhos pode também ser feita através das ligações 1:1 feitas pelo comando LIGAR do SMDD (subsistema de Manipulação do Dicionário de Dados) do SDDPUC.

Ex.: A obtenção de EMBAL → PESO pode ser constatada por: "LIGAR EMBALAGEM ASSOCIA REMESSA COM VIEW=3" e "LIGAR REMESSA ASSOCIA PESO COM VIEW = 5"

Procedemos agora a análise semântica, para verificarmos a validade ou não de mantermos algumas dependências transitivas:

1º) Eliminamos de imediato as composições de composições 4,6 e 10;

2º) As demais dependências transitivas não tem significado semântico, podendo então serem eliminadas.

Do digrafo, temos as relações elementares

RE₁ (VIAGEM, VEICULO)

RE₂ (VIAGEM, PESO)

RE₃ (VIAGEM, VOLUME)

RE₄ (VIAGEM, DATA)

RE₅ (EMBAL, DATA)

RE₆ (EMBAL, PRODUTO)

RE₇ (EMBAL, MFG)

RE₈ (EMBAL, VOL-EMB)

RE₉ (EMBAL, PESO-EMB)

RE₁₀ (EMBAL, REMESSA)

RE₁₁ (REMESSA, CLIENTE)

RE₁₂ (REMESSA, DATA)

RE₁₃ (REMESSA, VIAGEM)

RE₁₄ (REMESSA, PESO)

RE₁₅ (REMESSA, VOLUME)

Obtemos agora, o conjunto de cobertura mínima do nosso exemplo:

$$S^{15} = \{RE_1, \dots, RE_{15}\}$$

Verificamos se deste conjunto, ainda existem relações que sejam compostas através de outras relações. Vejamos o quadro:

RELAÇÃO ELEMENTAR	COMPOSICAO	DISTANCIA d	CAMINHO(S)
RE ₁	-	1	-
RE ₂	-	1	-
RE ₃	-	1	-
RE ₄	-	1	-
RE ₅	C (RE ₁₀ , RE ₄)	3	RE ₁₀ , RE ₁₃ , RE ₄ (ou RE ₁₀ , RE ₁₂)
RE ₆	-	1	-
RE ₇	-	1	-
RE ₈	-	1	-
RE ₉	-	1	-
RE ₁₀	-	1	-
RE ₁₁	-	1	-
RE ₁₂	C (RE ₁₃ , RE ₄)	2	RE ₁₃ , RE ₄
RE ₁₃	-	1	-
RE ₁₄	C (RE ₁₃ , RE ₂)	2	RE ₁₃ , RE ₂
RE ₁₅	C (RE ₁₃ , RE ₃)	2	RE ₁₃ , RE ₃

Associando pesos às RE's de distância máxima maior que 1, procedemos à derivação do melhor conjunto de "coberturamínima". A atribuição dos pesos é feita de acordo com o grau de importância, da frequência de uso e recuperação da associação estabelecida, e da distância. Assim, os pesos são atribuídos quando da descrição das visões usando o SDDPUC. Poderíamos ter usado o comando LIGAR para expressar estas relações compostas, da seguinte forma:

Para RE₅:

LIGAR EMBALAGEM ASSOCIA DATA COM VIEW = V3 PESO = 8

Para RE₁₂:

LIGAR REMESSA ASSOCIA DATA COM VIEW = V5 PESO = 6

Para RE₁₄:

LIGAR REMESSA ASSOCIA PESO COM VIEW = V5 PESO = 5

Para RE₁₅:

LIGAR REMESSA ASSOCIA VOLUME COM VIEW = V5 PESO = 4

Adotando os procedimentos citados em [JGS] - capítulo 3 - chegamos às iterações:

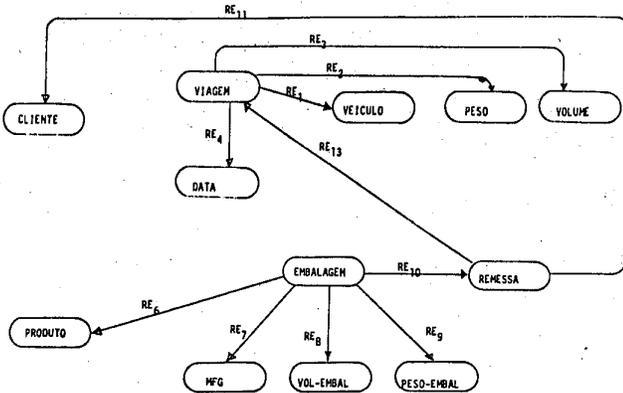
Iteração I₀

S _n ²	RE ^s	C (RE _i , RE _j)	DISTANCIA d _{ij}	PESO P _i	RE ^r REHOVIDA	S _{n-1} ²	CM
S ₁ ⁵	RE ₅	C (RE ₁₀ , RE ₄) C (RE ₁₀ , RE ₁₂)	3	8	RE ₅	S ₁₄	N X O
	RE ₁₂	C (RE ₁₃ , RE ₄)	2	6			
	RE ₁₄	C (RE ₁₃ , RE ₂)	2	5			
	RE ₁₅	C (RE ₁₃ , RE ₃)	2	4			
	RE ₁	-	1	-			
	RE ₂	-	1	-			
	RE ₃	-	1	-			
	RE ₄	-	1	-			
	RE ₆	-	1	-			
	RE ₇	-	1	-			
	RE ₈	-	1	-			
	RE ₉	-	1	-			
	RE ₁₀	-	1	-			
RE ₁₁	-	1	-				
RE ₁₃	-	1	-				

Na iteração I₁, eliminaríamos a RE₁₂; na iteração I₂ a RE₁₄; na iteração I₃ a RE₁₅; e o nosso melhor conjunto de "cobertura mínima" consiste de CM {S₁¹}

com S₁¹¹ = {RE₁, ..., RE₄, RE₆, ..., RE₁₁, RE₁₃}

que equivale ao dígrafo:



O "bubble-chart" assim obtido retrata a porção do mundo real modelado, fornecendo-nos as relações elementares:

- RE₁ (VIAGEM, VEICULO)
- RE₂ (VIAGEM, PESO)
- RE₃ (VIAGEM, VOLUME)
- RE₄ (VIAGEM, DATA)
- RE₆ (EMBAL, PRODUTO)
- RE₇ (EMBAL, MFG)
- RE₈ (EMBAL, VOL-EMB)
- RE₉ (EMBAL, PESO-EMB)
- RE₁₀ (EMBAL, REMESSA)
- RE₁₁ (REMESSA, CLIENTE)
- RE₁₃ (REMESSA, VIAGEM)

Podemos agora reduzir o número de relações elementares de acordo com o estabelecido em [JGS], no item 3.4.4 - Cap. 3. Criamos subconjuntos S_j com chaves K_j idênticas:

- S₁ = {RE₁, RE₂, RE₃, RE₄} com K₁ = VIAGEM
- S₂ = {RE₆, RE₇, RE₈, RE₉, RE₁₀} com K₂ = EMBAL
- S₃ = {RE₁₁, RE₁₃} com K₃ = REMESSA

A combinação das RE_i em cada subconjunto S_j produz a representação:

VIAGEM	DATA	VEICULO	PESO	VOLUME
--------	------	---------	------	--------

REMESSA	CLIENTE
---------	---------

EMBAL	MFG	VOL-EMB	PES-EMB	PRODUTO
-------	-----	---------	---------	---------

6 - Procedimentos Básicos para Projeto de Estruturas de Banco de Dados

Com base no que foi apresentado até aqui, resumimos uma série de procedimentos que os projetistas podem se utilizar quando da especificação da estrutura lógica de seus Bancos de Dados. Propomos que a abordagem ao problema seja feita em 2 (dois) estágios, permitindo que num 1º estágio se tenha uma visão macro do problema, e no 2º estágio, através de mecanismos mais refinados, possamos colher todos os detalhes - visão micro-necessários para obtenção da estrutura canônica dos dados.

a) Abordagem Primeiro Estágio - (APE)

a.1) Adotar os passos de nº 1 a 9, descritos em [JGS] - Cap. 2 - seção 2.3, para obter dos usuários uma visão global da realidade a ser modelada. Nesta fase, não deve haver preocupação com os problemas de dependências de dados;

a.2) Registrar todas as informações sobre as entidades, relacionamentos e seus atributos utilizando uma facilidade de entrada por telas do SDDPUC (MO), ou os documentos propostos, por [Melo].

b) Abordagem Segundo Estágio - (ASE)

• O objetivo principal deste estágio é obter uma estrutura lógica dos dados, a partir da qual possa ser implementada a base de dados. Para tanto, propomos os seguintes passos:

b.1) Definir os tipos de objetos, relacionamentos e atributos através do SMMD (Subsistema de Manipulação do Meta-Dicionário), usando os comandos AUTORIZAR e DESAUTORIZAR;

b.2) Com base no DER obtido da APE do 1º estágio e nas informações contidas nos formulários, construir um gráfico de bolhas (segundo as regras definidas em [JGS], Cap. 4) com as associações entre os dados ainda não-normalizados;

b.3) Registrar os dados no dicionário SDDPUC através dos comandos INCLUIR, EXCLUIR, LIGAR, DESLIGAR, DESCREVER e EXPANDIR, do subsistema SMDD (Subsistema de Manipulação do Dicionário de Dados);

b.4) Recuperar as informações contidas no Dicionário de Dados através dos comandos INFORME, EXAMINE, SELECIONE, RECUPERE e LISTE do subsistema de Extração de Informação - SEI. Isto permitirá que sejam reavaliadas todas as informações prestadas até aquele momento;

b.5) Proceder à análise das associações, selecionando as associações complexas, através dos comandos SELECIONE, RECUPERE e LISTE;

b.6) Proceder à normalização das relações a fim de obter as relações elementares;

b.7) Utilizar os comandos DESLIGAR (para desfazer as associações complexas M:M) e LIGAR (para ativar as associações entre itens de dados estabelecidos pelas relações normalizadas);

b.8) Aplicar os procedimentos descritos em [JGS] - Cap. 3 - seção 3.4.2, para determinação do conjunto máximo de RE's - relações elementares - deriváveis da coleção inicial de RE's (fecho transitivo);

b.9) Eliminar através do comando DESLIGAR, as composições inexpressivas semanticamente;

b.10) Registrar no SDDPUC as composições que tenham significado semântico através do comando LIGAR;

b.11) Eliminar todas as possíveis relações elementares redundantes, obtendo o que chamamos de "cobertura mínima" de acordo com os procedimentos descritos em [JGS] - no item 3.4.3 do Cap. 3;

b.12) Registrar o desligamento das relações redundantes utilizando-se do comando DESLIGAR do SMDD;

b.13) Recompôr as entidades a partir de relações elementares de mesma chave de acordo com os procedimentos descritos em [JGS] - item 3.4.4, no Cap. 3;

b.14) Representar o esquema canônico obtido sob a forma de uma estrutura relacional, ou hierárquica, ou CODASYL, ou quaisquer outras.

7 - Conclusões e Sugestões para Futuros Trabalhos

Neste trabalho discutimos os aspectos utilizados na obtenção de um esquema conceitual, no processo de projeto de Banco de Dados. Procuramos citar os procedimentos e as ferramentas que podem ser utilizadas

durante esta fase de projeto, mostrando a viabilidade de apoio destes instrumentos à análise e à documentação.
 É importante ressaltar que, durante o desenvolvimento de alguns exemplos estudados, surgiram alguns problemas quando da especificação dos dados e de seus relacionamentos. Observamos que a diagramação por gráfico de bolhas pelo processo intuitivo proposto por (JM) e (RHu), levou-nos a omitir não só algumas relações elementares dos dados, bem como a não eliminar dados redundantes e alguns tipos de dependências. Estas omissões puderam ser constatadas pelo emprego do método proposto por Vetter (Vet), e aplicado por (JGS), e que apesar de aparentemente conduzir o esquema conceitual mínimo final à sua representação inicial, mostra-se bastante válido ao evitar que tais omissões ocorram ao utilizar a diagramação por bolhas.

Verificamos que o mapeamento do modelo de entidades e relacionamentos para o modelo relacional resulta em esquemas de relações e dependências funcionais. Dependendo da aplicação em estudo, poderá resultar também em dependências de inclusão (Cas) e (Sev) que são um caso especial das dependências implicacionais embebidas estendidas (Fag2).

Procuraremos nos ater à validação do método mais simples mostrando como um sistema de Dicionário de Dados Flexível (no caso o SDDPUC) pode ser útil no processo de projeto e de documentação.

Concluindo, podemos dizer que outros trabalhos poderão ser desenvolvidos dentro do contexto de estrutura do SIBER (Sistema Integrado Baseado em Entidades e Relacionamentos), do qual faz parte o SDDPUC. Sugerimos que:

1ª) os procedimentos e algoritmos para determinação das transitividades e redundâncias venham a ser implementados no SIBER e incorporados ao SDDPUC;

2ª) a representação esquemática do Banco de Dados seja feita também através do SDDPUC, sob a forma de tabela ou de matriz, permitindo que o diagrama de estrutura de dados seja obtido mais facilmente.

Bibliografia

(Cas) - M.A. CASANOVA, R. Fagin, C.H. Papadimitriou "Inclusion Dependencies and their interaction with Functional Dependencies". PROC. OF THE ACM SIGACT - SIGMOD Conf. on Principles of Database Systems - MAR/82.
 (Chen1) - Chen, P.P.
 "The Entity-Relationship Model - Toward a Unified View of Data", ACM Transaction on DB Systems - Mar/76 - pág. 9 a 36.
 (DBDA) - Database design Aid, Designer's guide - GH20-1627, IBM Corporation, Data Processing Division, White Plains, NY.
 (Dgn) - "User's Guide of DATA DESIGNER" - Database Design, Inc. Jan/1980.
 (Fag2) - R. Fagin

"A Normal Form for Relational Databases that is based on Domain and keys".

ACM TODS 6.3 (Sept. 1981)

(Gan) - C. Gane & T. Sarson

"Structured Systems Analysis: Tools and Techniques", Impr. System Tech. - OCT/77.

(HNC) - F. Harary; R.Z. Norman; Cartwright

"Structured models: an introduction to the theory of directed graphs" - John Wiley - 1965.

(JGS) - José Geraldo Silva

"Um método sistemático de projeto de estrutura de Banco de Dados" - Dissertação de Mestrado - PUC/RJ - 1982

(JM) - James Martin

"Computer Database Organization" - 2nd.Ed. Prentice-Hall Inc. - 1977.

(Lob) - R.E. Lobel

"Uma contribuição para o desenvolvimento de estruturas flexíveis para Sistemas de Dicionário de Dados" - Dissertação de Mestrado, PUC/RJ - Jan/81.

(Melo) - R. N. Melo e C. Crisóstomo

"Administração de Dados e suas ferramentas" - I Conferência Latino-Americana, CARACAS/1980.

(MO) - Marly Oppenheimer

"Uma facilidade para o usuário final do SDDPUC" - Tese de Mestrado-PUC/RJ - 1982

(NgP) - P.A.Ng e Jean F. Paul

"A formal definition of entity-relationship models" Entity-Relationship Approach to Syst. Anal. and Design 1980 - North-Holland - pág. 211-230.

(RHu) - N. Raver, & G. Hubbard

"Automated logical data base design: Concepts and Applications", IBM Syst. J. Nº 3 - vol. 16 - 1977.

(Sev) - Severino P. L'Rego

"Uma teoria de dependências funcionais e dependências de inclusão sobre expressões relacionais" - Dissertação de Mestrado, PUC/RJ - ABRIL/1982.

(SNF) - C.S.Santos; E.J. Neuhold e A.L. Furtado

"A data type Approach to the E-R Model" - Entity - Relationship Approach to Syst. An. Desig. Chen, P. (Ed.) North Holland, Amsterdam-1980.

(SPARC) - ANSI/X3/SPARC - "Study group on Database Management Systems - Interim Report" - EDT - Bulletin of ACM - SIGMOD/75.

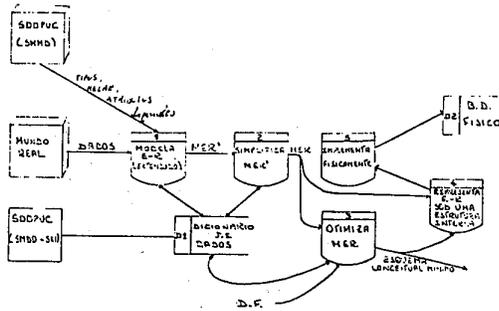
(Vet) - M. Vetter e R.N. Maddison

"Data Base Design Methodology", Prentice Hall Inc. 1981.

(Web) - Neil W. Webre

"An extended Entity-Relationship model and its use on a Defense Project" - in E-R approach to Inf. Modeling and Anlysis - P.P. Chen-1981 - pág. 175-195.

Fig. 2.2 - Ambiente SIBER (proposto)



3. 'Otimiza MER'

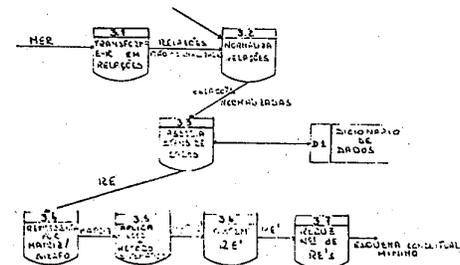
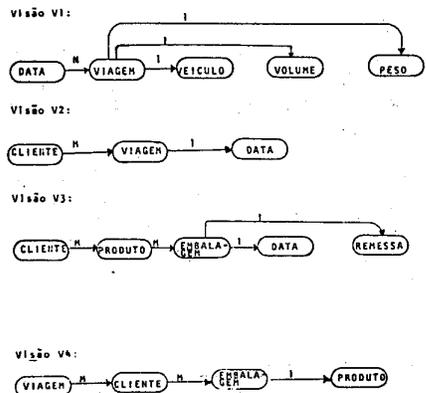


Figura 3.1. Modelo R-R das "User-Views"



Figura 3.2 - "Bubble-Chart" das User-Views do Exemplo



Visão V5:

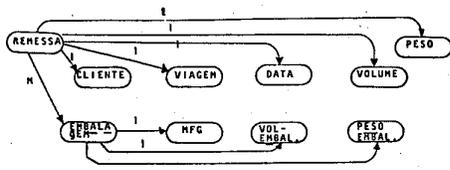


Figura 3.3 - Integração das 5 Visões do Exemplo

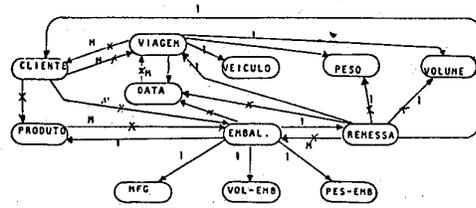


Figura 3.4 - Esquema Canônico do Exemplo

