ANDREA RODACKI

# APLICAÇÃO DE ESTRATÉGIAS DE INTEGRAÇÃO DE BANCOS DE DADOS: UM ESTUDO DE CASO

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Curso de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Marcos Sfair Sunye

CURITIBA
2000

Ministério da Educação
UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE CIÊNCIAS EXATAS

# PARECER

Nós, abaixo assinados, membros da Comissão Examinadora da defesa de Dissertação de Mestrado em Informática, da aluna Andrea Cristina Rodacki, avaliamos o trabalho intitulado **"Aplicação de Estratégias de Integração de Bancos de Dados: Um Estudo de Caso"**, cuja defesa foi realizada no dia 26 de Janeiro de 2000. Após a Avaliação, decidimos pela Aprovação da Candidata.

Curitiba, 26 de janeiro de 2000.

Prof. Dr. Marcos Sfair Sunye
Presidente

Profª. Dra. Wanda Maria Maia da Rocha Paranhos

Prof. Dr. Alexandre Ibrahim Direne

Esta dissertação é dedicada aos meus pais Ugo e Cristina que sempre me apoiaram e que me deram toda a base educacional para que eu pudesse realizar esta pesquisa.

Também dedico este trabalho em especial ao meu "mestre" e incentivador Marcos Sunye, pois sem seu crédito, atenção e competência, este meu objetivo não seria realizado.

## AGRADECIMENTOS

Inicialmente gostaria de agradecer aos amigos que fiz durante o desenrolar deste curso: Alexandre Manoel dos Santos, Cristiani Batata, Denis Rezende, Fabio Araujo, Jaylson Teixeira, Patricia Bassi e demais colegas de aula.

Ao coordenador executivo, Roberto Almeida, aos gerentes, Rui Krelling e Guilherme Lorenzi e a toda a equipe do CITS – Centro Internacional de Tecnologia de Software, pela ajuda constante e compreensão nas atividades do cotidiano.

Aos graduandos Renato Katsuragawa e Kemmel da Silva Scopim pela ajuda no desenvolvimento desta pesquisa.

Aos funcionários do Centro de Computação Eletrônica, Denise Lobo e Custódio pela cooperação na integração do Sistema de Controle de Pesquisa e Pós Graduação.

A bibliotecária Angela P. F. Mengatto pelas informações sobre as regras de formatação a serem empregadas neste documento.

A Universidade Federal do Paraná e a todo Departamento de Informática, professores e funcionários.

Aso meus amigos que sempre me deram apoio nas horas de estudo e compreenderam os momentos que tive me dedicar a este trabalho e não a eles.

E finalmente, gostaria de agradecer a Deus pela suas energias que inspiram minha serenidade e consciência.

SUMÁRIO

## LISTA DE FIGURAS

## LISTA DE QUADROS

## LISTA DE TABELAS

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| ACS | ABSTRACT CONCEPTUAL SCHEME |
| BERDI | FEDERATED DATABASE TOOL |
| CASE | COMPUTER AIDED SOFTWARE ENGINEERING |
| CLT | CONSOLIDAÇÃO DAS LEIS TRABALHISTAS |
| CODASYL | CONFERENCE ON DATA SYSTEMS LANGUAGES |
| DBA | DATABASE ADMINISTRATOR |
| DDBS | DISTRIBUTED DATABASE SYSTEMS |
| ER | ENTIDADE RELACIONAMENTO |
| ERC+ | ENTIDADE RELACIONAMENTO COMPLEXO ESTENDIDO |
| GDM | GENERIC DATA MODEL |
| HOSQL | HETEROGENEOUS OBJECT STRUCTURED QUERY LANGUAGE |
| IMS | INFORMATION MANAGEMENT SYSTEM |
| MUVIS | MULTI-USER VIEW INTEGRATION SYSTEM |
| OLAP | ON-LINE ANALYTICAL PROCESSING |
| RWS | REAL WORLD STATE |
| SAGU | SISTEMA DE APOIO AO GERENCIAMENTO UNIVERSITÁRIO |
| SGBD | SISTEMA GERENCIADOR DE BANCO DE DADOS |
| SIBI | SISTEMA DE BIBLIOTECAS |
| SIM | SCHEMA INTEGRATION METHODOLOGY |
| SIS | SCHEMA INTEGRATION SYSTEM |
| SQL | STRUCTURED QUERY LANGUAGE |
| UFPR | UNIVERSIDADE FEDERAL DO PARANÁ |

| | |
|---|---|
| UoD | UNIVERSO DO DISCURSO |
| VODAK | OPEN OBJECT ORIENTED DATABASE SYSTEMS |

# RESUMO

Este trabalho é dedicado à integração de bancos de dados heterogêneos, a qual podemos definir como sendo o processo que, através de uma entrada de um conjunto de bancos de dados, produz, como saída, uma descrição unificada do esquema inicial, chamado esquema integrado, e a informação de mapeamento que apoia o acesso aos dados armazenados no esquema integrado. Apresentamos uma aplicação prática da utilização de duas metodologias de integração de bancos de dados heterogêneos, a metodologia de SPACCAPIETRA; PARENT e DUPONT (1992), a qual nos deu embasamento no processo de integração, e a de BATINI e LENZERINI (1984) que foi utilizada em alguns casos particulares. Esta aplicação utilizou, como modelo de dados padrão, o modelo ERC+, Entidade Relacionamento Complexo Estendido, visto que é um modelo semântico e por ser empregado na principal metodologia de integração utilizada neste trabalho. Como o escopo deste trabalho envolve a aplicação de metodologias de integração e como a integração de bancos de dados heterogêneos é um problema complexo, que demanda tempo de análise e desenvolvimento, este trabalho apresenta uma primeira versão do protótipo do esquema integrado completo, a qual foi gerada utilizando os seguintes sistemas da Universidade Federal do Paraná: Sistema de Automação Universitária - Administração e Pessoal, Sistema de Automação Universitária - Controle Acadêmico, SIBI - Sistema de Bibliotecas e o Sistema de Controle de Pesquisa e Pós-Graduação. A implementação do protótipo foi realizada em três fases: pré integração; identificação das correspondências e verificação da conformidade dos esquemas; e integração. Como benefícios gerados no processo de integração podemos destacar: disseminação do conhecimento de aplicações distribuídas e heterogêneas; visão unificada dos dados; possibilidade de construção de sistemas de apoio à decisão e *data warehouses* a partir do esquema integrado. Metas futuras desta pesquisa são apresentadas com base na elaboração do mapeamento de consultas partindo do esquema integrado para os esquemas iniciais.

## ABSTRACT

This work is dedicated to the integration of heterogeneous databases, which we can define as the process that, through as input a set of databases, produces, as output, an unified description of the initial scheme, called integrated scheme, and the mapping information that supports the access to the data stored in the integrated scheme. We present a practical application of the use of two methodologies of heterogeneous databases integration, the methodology of SPACCAPIETRA; PARENT and DUPONT (1992), which gave us the basis for the integration process, and the methodology of BATINI and LENZERINI (1984) that was used in some particular cases. The ERC+, Extended Entity Relationship Complex model was used, as a pattern data model, because it is a semantic model and for being used in the main integration methodology of this work. Due to the scope of this work involving the application of integration methodologies and as the integration of heterogeneous databases is a complex problem, that demands time for analysis and development, this work presents a first version of the prototype of the integrated scheme, which was generated using the following systems of the Federal University of Paraná: System of University Automation - Administration and Personal, System of University Automation - Academic Controls, Library System and the System of Research and Masters Degree Control. The implementation of the prototype was accomplished in three phases: pre integration; identification of the correspondences and verification of the conformity of the schemes; and integration. As benefits generated in the integration process we can highlight: dissemination of the knowledge of distributed and heterogeneous applications; unified vision of the data; possibility of construction of decision support systems and data warehouses starting from the integrated scheme. Future goals of this research are presented based in the elaboration of the query mappings starting from the integrated scheme to the initial schemes.

# 1. INTRODUÇÃO

Empresas e organizações, em todo o mundo, armazenam as informações referentes aos seus negócios, em uma miríade de bancos de dados distribuídos em plataformas diferentes, incluindo computadores de grande porte, como os *mainframes*, estações de trabalho, arquiteturas cliente-servidor, *intranet* e *internet*.

Historicamente, os bancos de dados foram concebidos para suprir as necessidades de várias empresas sem que houvesse requisitos únicos a serem cumpridos.

Este fato levou à proliferação dos sistemas gerenciadores de bancos de dados que obedecem a diferentes conjuntos de requisitos para modelar objetos que possuem o mesmo significado no mundo real.

Em muitos casos, por causa da grande variedade de informações existentes nas organizações, os usuários criam seus sistemas isolados do sistema principal da empresa, porém estes dados já estão armazenados em um banco de dados central.

Devido à necessidade cada vez maior de acesso às informações dentro das organizações, através de seus colaboradores e dirigentes e, fora delas, pelos seus clientes, a pressão para prover informações distribuídas em diferentes plataformas é cada vez mais freqüente.

Devido ao avanço dos estudos nas tecnologias de redes e nos sistemas de informações distribuídas, combinados ao alto grau de conectividade existente hoje em dia nas redes de computadores, o estudo de aplicações de bancos de dados distribuídos tornou-se uma realidade no mundo da computação atual.

Os bancos de dados atuais são administrados por diferentes sistemas gerenciadores de bancos de dados (SGBD's) baseados em plataformas heterogêneas.

O grande desafio da área de estudos de bancos de dados distribuídos é dispor aos usuários, as informações distribuídas, como se as mesmas estivessem localizadas em um único banco de dados, preservando a integridade e os investimentos realizados na construção dos bancos de dados iniciais (BRODIE e STONEBRAKER, 1995).

Nestes sistemas, é necessário, além de fornecer acesso às informações distribuídas de uma maneira transparente, conceder aos bancos de dados, técnicas particulares de troca e compartilhamento de dados, de uma maneira sincronizada.

Um usuário final de um sistema de bancos de dados heterogêneos deve ter a possibilidade de acessar os múltiplos bancos de dados existentes e coordenar suas transações de uma maneira transparente. Estes sistemas devem funcionar independentes do computador, sistema operacional, plataforma de hardware ou tipo de dado em que estão baseados os sistemas iniciais.

Considerando todos os itens dispostos acima, para que uma solução de arquitetura de sistemas de bancos de dados heterogêneos seja realizada com sucesso, devemos preservar a autonomia e a heterogeneidade dos bancos de dados envolvidos através da utilização de metodologias de integração de bancos de dados.

Existem diferentes tipos de heterogeneidade e autonomia. As primeiras pesquisas na área de bancos de dados distribuídos ignoraram os problemas de autonomia e enfatizaram a solução da heterogeneidade (HURSON; BRIGHT e PAKZAD, 1994). A questão da autonomia foi ignorada até o início do ano de 1980 devido às condições pouco favorecidas de conexão de rede existentes na época.

O problema da heterogeneidade ocorre em todos os segmentos de estudo de sistemas gerenciadores de bancos de dados distribuídos. Algumas pesquisas focaram seus estudos nos esquemas dos bancos de dados.

As técnicas tradicionais de bancos de dados distribuídos levaram em consideração outros aspectos da heterogeneidade, como o controle da concorrência dos dados. Como resultado, a primeira pesquisa baseada na integração, ou fusão, de componentes dos esquemas conceituais dos bancos de dados está descrita em BATINI; LENZERINI e NAVATHE em 1986.

Uma outra área de pesquisa, com um foco maior na autonomia e flexibilidade dos bancos de dados, pode ser verificada em HEIMBIGNER e McLEOD (1985) e denomina este caso de bancos de dados federados.

Porém, o que podemos constatar, é que a linha principal das pesquisas de BATINI; LENZERINI e NAVATHE (1986) e de HEIMBIGNER e McLEOD (1985) é a mesma. Inicialmente, a integração parcial é realizada, gerando um esquema integrado. À medida que existe a necessidade de integração de um novo sistema, o mesmo é acoplado no esquema integrado.

Após este passo, os administradores dos bancos de dados, em conjunto com os desenvolvedores dos sistemas, processam a integração manualmente, fazendo deste processo uma seqüência exaustiva e repetitiva.

Outro estudo, mais recente, que favorece mais a autonomia do que a heterogeneidade, é a pesquisa em multi banco de dados (HURSON; BRIGHT e PAKZAD, 1994). Neste caso, os usuários são responsáveis por integrar os esquemas que eles precisam na aplicação.

O suporte é dado por uma multi linguagem de banco de dados que contém toda a sintaxe necessária para o acesso e manipulação dos dados. Este caso tira a responsabilidade do administrador de banco de dados e a coloca sobre os usuários, que têm o ônus de localizar e escolher os esquemas necessários no processo de integração.

Este trabalho é dedicado à integração de banco de dados, a qual podemos definir como sendo o processo que, através de uma entrada de um conjunto de bancos de dados, produz, como saída, uma descrição unificada do esquema inicial, chamado esquema integrado ou esquema global, e a informação de mapeamento que apoia o acesso aos dados armazenados no esquema integrado.

O estudo da integração de bancos de dados surgiu na década de 1970, onde iniciaram-se os trabalhos na integração de visões. A integração de visões foi definida como um passo da modelagem de um esquema conceitual global a partir de um conjunto de visões formalmente definidas pelos usuários.

A integração de visões é o processo de geração de um único esquema integrado a partir de múltiplas visões de usuários e é tipicamente utilizada na criação de um novo esquema de banco de dados.

As metodologias de integração de visões são normalmente utilizadas na modelagem de bancos de dados complexos, que possuem um esquema conceitual com um número expressivo de objetos.

Habitualmente, bancos de dados deste grau de grandeza são destinados a vários tipos de usuários, como por exemplo, em um banco de dados corporativo, onde existem os seguintes tipos de usuário: administradores, diretores, gerentes e secretárias.

Esta complexidade, de tamanho e de tipos de usuários, torna mais adequada a criação de vários esquemas específicos, ou visões, uma para cada tipo de usuário.

Estas visões correspondem a partes específicas do banco de dados que cada usuário deve ter acesso.

Devido a este fato, o método de integração de visões inicia-se através da análise das múltiplas visões dos usuários, gerando o esquema integrado correspondente a estas visões e, após esta fase, gera-se o modelo do banco de dados que corresponde a este esquema. Este modelo é gerado a partir de visões que foram criadas utilizando um modelo único de dados.

A integração de esquemas é empregada para integrar bancos de dados existentes. Como a integração de esquemas possibilita a integração de bancos de dados heterogêneos, diferentes modelos de dados podem ser utilizados para representar os esquemas a serem integrados, porém esta não é a melhor alternativa.

Devemos manter os esquemas de todos os bancos de dados representados em um único modelo para facilitar a especificação do esquema integrado, mesmo que os bancos de dados iniciais estejam baseados em modelos de dados diferentes.

Outra importante distinção verificada entre a integração de esquemas e a integração de visões é que na última, as visões do usuário não refletem dados que existem, diretamente, na base.

Por exemplo, uma visão de um banco de dados de uma universidade faz o acesso a uma única tabela (virtual) para consultar os dados dos estudantes, porém, no banco de dados, estes dados estão distribuídos em duas tabelas (físicas), uma de alunos de graduação e outra de alunos de pós graduação.

Isto já não acontece na integração de esquemas, onde integramos dados que estão representados exatamente como estão armazenados no banco de dados. No caso do exemplo acima, as tabelas físicas.

Esta é uma consideração importante, visto que o esquema gerado pela integração de esquemas não viola a semântica dos bancos de dados iniciais. Na integração de visões, considerando-se que as visões representam uma abstração dos objetos, existe uma maior flexibilidade na interpretação da semântica.

A integração de esquemas é um problema complexo que demanda tempo de análise e de desenvolvimento, principalmente porque a maioria das representações dos esquemas não conseguem representar a semântica dos bancos de dados como um todo.

Por este motivo, o processo de integração de esquemas exige uma freqüente interação com os desenvolvedores e administradores dos bancos de dados que irão formar o esquema integrado, assegurando que o mesmo possua todo o entendimento semântico e não viole as regras dos bancos de dados iniciais.

O termo integração de esquemas tem sido muito utilizado na literatura referindo-se a metodologias que facilitam a integração de esquemas da mesma forma que na integração de visões. Este fato decorre de que muitas das técnicas aplicadas no contexto de integração de esquemas são também utilizadas na integração de visões e vice-versa.

A integração de esquemas, usualmente, envolve um processo de tradução dos esquemas iniciais que pode ser realizado manualmente, automaticamente, ou pelos dois métodos. Este processo abrange a tradução das linguagens de consulta ou o mapeamento de um modelo de dados para outro. O modelo final é normalmente um modelo canônico (BATINI; LENZERINI e NAVATHE, 1986; ATZENI e TORLONE, 1993; SPACCAPIETRA e PARENT, 1994).

Como a integração de bancos de dados envolve um processo no qual o conhecimento da semântica de dados é necessário, a maioria das metodologias atuais utiliza uma modelagem de dados semântica.

Em particular, o modelo Entidade Relacionamento (ER), em suas várias formas estendidas, é o mais utilizado. Isto é consistente com o atual estado dos negócios, nos quais o modelo ER age como um padrão na área de métodos de projetos conceituais e ferramentas de bancos de dados.

Quando usamos um modelo de dados semântico, é possível que diferentes desenvolvedores modelem o mesmo objeto do mundo real de formas diferentes. Isto pode acontecer porque os mesmos dados podem ser modelados de maneiras equivalentes ou porque os desenvolvedores têm percepções diferentes daquela realidade.

Este fato é chamado de *semântica* relativista que é a multiplicidade de possíveis representações de um determinado objeto no mundo real.

O modelo escolhido, neste trabalho, para realizar a integração de bancos de dados heterogêneos, é o modelo ERC+ (Entidade Relacionamento Complexo Estendido), o qual suporta a semântica relativista e, por esta razão, expressa, com

maior riqueza, a relação existente entre o significado do objeto no mundo real e a sua representação no banco de dados.

O modelo ERC+ é um modelo ER estendido que foi definido para apoiar a descrição de objeto complexo e sua manipulação. Este modelo é chamado ERC+ (SPACCAPIETRA e PARENT, 1992) devido ao seu significado: ER para objetos Complexos (o + denota o enriquecimento do modelo ERC básico).

Fizemos uso deste modelo porque o mesmo é utilizado na principal metodologia de integração a ser empregada neste trabalho.

Utilizamos a metodologia de SPACCAPIETRA; PARENT e DUPONT (1992) pois a consideramos a mais abrangente e completa, tanto na questão da heterogeneidade dos modelos de bancos de dados que suporta, quanto na abrangência de bancos de dados que podem ser integrados.

Esta metodologia interage tanto com os problemas estruturais quanto com os de heterogeneidade semântica. O esquema é visualizado como um grafo com ângulos e vértices. As relações existentes entre dois objetos são especificadas através de regras de correspondências, que podem ser consideradas como uma extensão dos conceitos de equivalência de objetos e de relacionamentos apresentados em LARSON; NAVATHE e EL-MASRI (1989).

Como esta metodologia aborda essencialmente a problemática da integração de esquemas, em alguns casos particulares, encontrados durante o andamento do projeto, fomos levados a aplicar soluções propostas na metodologia de BATINI e LENZERINI (1984).

Nossa meta, neste trabalho, é relacionar a utilização das metodologias com a implementação prática do processo de integração, contribuindo, assim, para um melhor conhecimento deste processo.

Existem muitas metodologias de integração de bancos de dados heterogêneos porém, poucos trabalhos na literatura citam aplicações práticas das mesmas. Na demonstração da execução das metodologias são utilizados exemplos hipotéticos que não nos permitem verificar o grau de abrangência das regras demonstradas.

Este trabalho apresenta um protótipo da integração dos sistemas utilizados na Universidade Federal do Paraná (UFPR), baseado em vários bancos de dados

diferentes. Temos, através do ambiente dos sistemas existentes na UFPR, uma problemática adequada para a utilização das metodologias de integração.

Os sistemas envolvidos, conforme iremos explanar e demonstrar em capítulo específico deste trabalho, estão baseados em três modelos de dados diferentes. Podemos citar, dentre eles, o Sistema de Automação Universitária que fundamenta-se no modelo hierárquico; o Sistema de Gerenciamento de Usuários, o Sistema de Controle de Pesquisa e Pós-Graduação e o Sistema de Bibliotecas, que são relacionais e o Repositório de Eventos Clínicos do Hospital de Clínicas de Curitiba, entidade vinculada à UFPR, que se baseia no modelo objeto relacional.

Como podemos verificar, existe um alto grau de heterogeneidade para a aplicação das metodologias, pois os sistemas que serão envolvidos no processo de integração estão baseados em três modelos de dados diferentes, além de que os mesmos são administrados por diferentes SGDB's, tais como, o DMS-II no Sistema de Automação Universitária; o MICROSOFT SQL SERVER no Sistema de Gerenciamento de Usuários; o ORACLE no Sistema de Controle de Pesquisa e Pós-Graduação; o ACCESS no Sistema de Bibliotecas; e o ORACLE versão 8i no Repositório de Eventos Clínicos.

Com a elaboração do esquema integrado, por meio desta aplicação prática das metodologias de integração de bancos de dados heterogêneos, poderemos comprovar o custo deste processo e as melhorias realizadas, tanto para os usuários finais, quanto para os desenvolvedores e administradores dos sistemas envolvidos.

O resultado desta integração também poderá ser utilizado pela UFPR como base para o desenvolvimento de sistemas de *data warehouse* com a utilização de técnicas de análise (*data minings*), visto que os sistemas integrados armazenam informações estratégicas e de suporte à decisão da universidade.

A integração de informações de diferentes fontes visa a melhoria da qualidade das informações disseminadas.

As principais características da solução demonstrada neste estudo são:

- Integração de visões e esquemas, contemplando assim tanto as bases já existentes, quanto a construção de novos sistemas.
- A alteração da estrutura das bases operacionais, a partir do esquema integrado, não é contemplada.

As fases de desenvolvimento deste estudo envolveram pesquisas em metodologias de integração, definição do modelo comum a ser utilizado para diagramar o esquema integrado, identificação dos casos de uso, assim como dos conflitos existentes entre eles e, finalmente, geração do protótipo, através da geração do esquema integrado ou esquema global.

Nossa abordagem de integração utiliza os seguintes passos para desenvolver o esquema integrado:

- Pré integração, onde os esquemas a serem integrados são adequados para que fiquem mais homogêneos, tanto sintática quanto semanticamente. Nesta fase todos os casos a serem utilizados na integração são representados através do modelo ERC+.
- Identificação das correspondências e verificação da conformidade dos esquemas. Este passo dedica-se à comparação dos esquemas visando identificar e descrever os objetos que são representados de maneiras diferentes porém possuem o mesmo conceito no mundo real. Detectamos nesta fase os casos de conflito.
- Integração, último passo a ser desenvolvido e que resolve os conflitos encontrados nas interseções dos esquemas a serem integrados, unindo os esquemas com o objetivo de obter o esquema integrado que corresponde ao resultado final deste estudo.

Este trabalho está organizado da seguinte maneira: o capítulo 2 explanará sobre o modelo de dados ERC+, explicando sua notação e principais características; o capítulo 3 fará uma introdução das definições de bancos de dados distribuídos e a classificação utilizada; no capítulo 4 discorreremos sobre uma visão geral e história da integração de bancos de dados; o capítulo 5 discorre sobre o estado da arte das metodologias de integração; o capítulo 6 descreve a metodologia de SPACCAPIETRA; PARENT e DUPONT (1992), a qual nos deu embasamento neste trabalho e de BATINI e LENZERINI (1984), que nos apoiou no processo de integração; o capítulo 7 apresentará o protótipo e as estratégias de integração que utilizamos para implementar esta pesquisa, descrevendo os sistemas que compõem o protótipo, a sua execução e a solução adotada; e, finalmente, no capítulo 8, iremos concluir, indicar os benefícios deste estudo de caso e os trabalhos futuros a serem realizados.

## 2. O MODELO DE DADOS

Segundo ELMASRI e NAVATHE (1994) pode-se definir modelo de dados como sendo um conjunto de conceitos que podem ser usados para descrever a estrutura de um banco de dados. Pode-se categorizar estes modelos de acordo com os conceitos que utilizam para descrever a estrutura de um banco de dados.

Os modelos de dados conceituais, ou de alto nível, utilizam conceitos que são próximos da maneira de como o usuário visualiza aquele dado no mundo real, enquanto que os modelos físicos, ou de baixo nível, provêem conceitos que descrevem os detalhes de como os dados são armazenados no banco de dados.

Estes conceitos são compreendidos somente por pessoal especializado da área de computação, enquanto que os modelos conceituais, podem ser entendidos pelos usuários finais dos bancos de dados.

Entre estes dois extremos existe uma outra classificação chamada de representacional (ou implementacional), que fornece conceitos que podem ser compreendidos pelo usuário final porém possuem uma aproximação maior da representação de como os dados são armazenados no banco de dados.

Os modelos de dados de alto nível usam conceitos como entidades, atributos e relacionamentos. Temos como exemplo o modelo Entidade Relacionamento (ER) que mostra um diagrama gráfico dos dados do SGBD. Normalmente considera-se que um esquema ER será traduzido para um outro modelo de dados, relacional, por exemplo, na implementação de um banco de dados.

Os modelos representacionais ou implementacionais são mais utilizados nos SGBD's comerciais, e incluem os três modelos de dados mais utilizados na implementação de bancos de dados: relacional, em rede e hierárquico. Os dados são representados utilizando a estrutura dos registros, sendo que estes modelos também podem ser chamados de modelos de dados baseados nos registros.

O modelo relacional foi introduzido em 1970 por Codd e é baseado em uma estrutura de dados simples e uniforme – as relações – e tem uma sólida fundamentação teórica.

Por este motivo, o modelo relacional é o mais utilizado na implementação dos SGDB's comerciais, tais como, Microsoft SQL Server, SYBASE e ORACLE. Os dados são representados como um conjunto de relações, as tabelas, cada linha da

tabela é denominada uma tupla e cada coluna um atributo. Os tipos de dados que podem popular uma coluna são chamados domínios.

O modelo em rede foi apresentando em 1971 e foi definido pelo comitê CODASYL (*Conference on Data Systems Languages*), por este motivo é também conhecido como modelo em rede CODASYL.

Os comandos básicos de manipulação dos dados de um sistema baseado no modelo em rede são realizados através de linguagens de programação como o PASCAL ou o COBOL. Existem duas estruturas básicas neste modelo: os registros e os conjuntos. Os dados são armazenados em registros, cada registro consiste de um conjunto de valores de dados que possuem uma relação entre si.

O modelo hierárquico, desenvolvido no final da década de 1960, modela os muitos tipos de organizações hierárquicas que existem no mundo real. Este modelo representa os dados hierárquicos de uma organização de uma maneira direta e natural e pode ser a melhor opção em alguns casos, como a árvore genealógica de uma família. Porém, existem vários problemas quando precisamos representar situações que não se enquadram em dados hierárquicos, que é a estrutura utilizada para representar os dados. Podemos citar como exemplo de *software* que utiliza este modelo de dados o IMS (*Information Management System*) da IBM.

Podemos classificar os modelos orientados a objetos como uma nova família dentro dos modelos de alto nível, pois a sua representação é muita próxima da apresentada nos modelos conceituais.

O modelo de dados orientado a objeto foi proposto na década de 1980, com o objetivo de representar dados não convencionais, como imagens e objetos multimídia. A principal característica deste modelo é o poder que dá ao desenvolvedor de especificar a estrutura dos objetos complexos e as operações que aplicam-se aos mesmos. A versão 8i do SGDB da ORACLE pode ser enquadrada como objeto-relacional pois utiliza alguns conceitos de orientação a objeto.

O modelo Entidade Relacionamento (ER) foi desenvolvido em 1976 por Peter Chen como uma ferramenta para a modelagem de dados. O Modelo ER trabalha com três conceitos básicos: entidade (objetos do negócio), relacionamentos (associação entre os objetos) e atributos (propriedades dos objetos e dos relacionamentos) (CHEN, 1976). Este método demonstra a independência dos

dados e é baseado na teoria dos conjuntos e na teoria das relações (ABITEBOUL; HULL e VIANU, 1995).

O modelo ERC+, que é o modelo utilizado neste estudo para diagramar os dados dos esquemas iniciais e do esquema integrado, é uma extensão do modelo Entidade Relacionamento, especialmente desenvolvido para suportar objetos complexos e suas identidades. Seu desenvolvimento começou em 1983, como parte de um projeto de sistemas de bancos de dados heterogêneos distribuídos (SPACCAPIETRA et al., 1983).

A modelagem do objeto complexo e seu gerenciamento é uma das metas principais dos modelos de dados de hoje. Através de um objeto complexo, queremos dizer que um objeto é representado por uma coleção de informações, seus componentes, tal que cada um destes componentes pode ser representado por uma coleção de informações, e assim por diante.

O suporte a objetos complexos não contradiz a distinção básica que o modelo ER faz entre entidades e atributos: esta distinção está baseada em considerações semânticas (que são os objetos primários de interesse), não em propriedades sintáticas (sendo atômico ou não).

O modelo ERC+ permite, especificamente, esta descrição interativa de um objeto, até um número arbitrário de níveis. A estrutura resultante é uma árvore de atributo cuja raiz é o objeto. Além disso, qualquer nó na árvore pode levar a um valor de atributo sem igual, ou um multi conjunto de valores de atributo.

Esboçaremos, brevemente, nos próximos parágrafos, as características do modelo ERC+.

A estrutura de uma entidade consiste em um conjunto de um ou mais atributos.

Os relacionamentos podem unir qualquer número de entidade. É dito que eles são cíclicos se a mesma entidade participa, mais de uma vez, no relacionamento.

Um papel é associado a cada participação de uma entidade em um relacionamento. Este papel é caracterizado por seu mínimo e máximo de cardinalidades, especificadas como 0-1, 0-n, 1-1 ou 1-n de acordo com o vínculo da entidade para o relacionamento.

Atributos podem ser:

- Obrigatórios ou opcionais: uma instância de um atributo opcional pode estar vazia (nenhum valor), para um atributo obrigatório um valor deve ser definido, em cada instância do atributo.
- Monovalorado ou multivalorado: uma instância de um atributo multivalorado pode incluir vários valores, enquanto uma instância de um atributo monovalorado é composta de um único valor.
- Simples ou complexos: se simples, o atributo possui domínio definido e indica que o mesmo é o último componente da hierarquia de um grafo, ou seja, uma folha. Se complexo, o atributo é composto de um conjunto de outros atributos que são os componentes daquele atributo. Os atributos componentes podem ser simples ou complexos. Agrupando-se esta definição, pode-se proceder a qualquer número de níveis.

Entidades e relacionamentos podem ter zero, um ou mais conjuntos de atributos que servem como identificadores.

Um atributo identificador indica que o valor deste atributo é distinto para cada instância desta entidade, sendo que o mesmo é utilizado como identificação da instância.

Algumas vezes é necessário formar-se um identificador composto, o que quer dizer que o identificador é formado por mais de um atributo, sendo que a combinação destes atributos deve ser única na entidade. Se nenhum identificador é conhecido, a população respectiva pode incluir duplicidades (ocorrências diferentes com o mesmo valor).

Duas generalizações são suportadas no modelo ERC+, "*is-a*" e "*may-be-a*".

A generalização "*is-a*" corresponde ao conceito de generalização (SMITH e SMITH, 1977) onde se identifica uma entidade genérica que possui as características principais de entidades específicas.

Por exemplo, podemos definir uma entidade genérica Estudante que indica uma generalização das entidades específicas Estudante_graduação e Estudante_pósgraduação.

A generalização "*may-be-a*" tem semântica semelhante, mas não requer uma dependência de inclusão entre o subtipo e o tipo. Este conceito de generalização "*may-be-a*" indica que a população do tipo entidade genérico não inclui o conjunto da

população do tipo entidade específico, ou seja, uma consulta realizada no tipo entidade específico não abrange os dados do tipo entidade genérico.

Quando existe a ocorrência de uma generalização, os atributos da entidade pai são comuns às entidades filhas, sendo que os atributos específicos de cada entidade são representados nas entidades filhas.

Através da figura 1, abaixo, mostramos um esquema ERC+ e a representação utilizada.

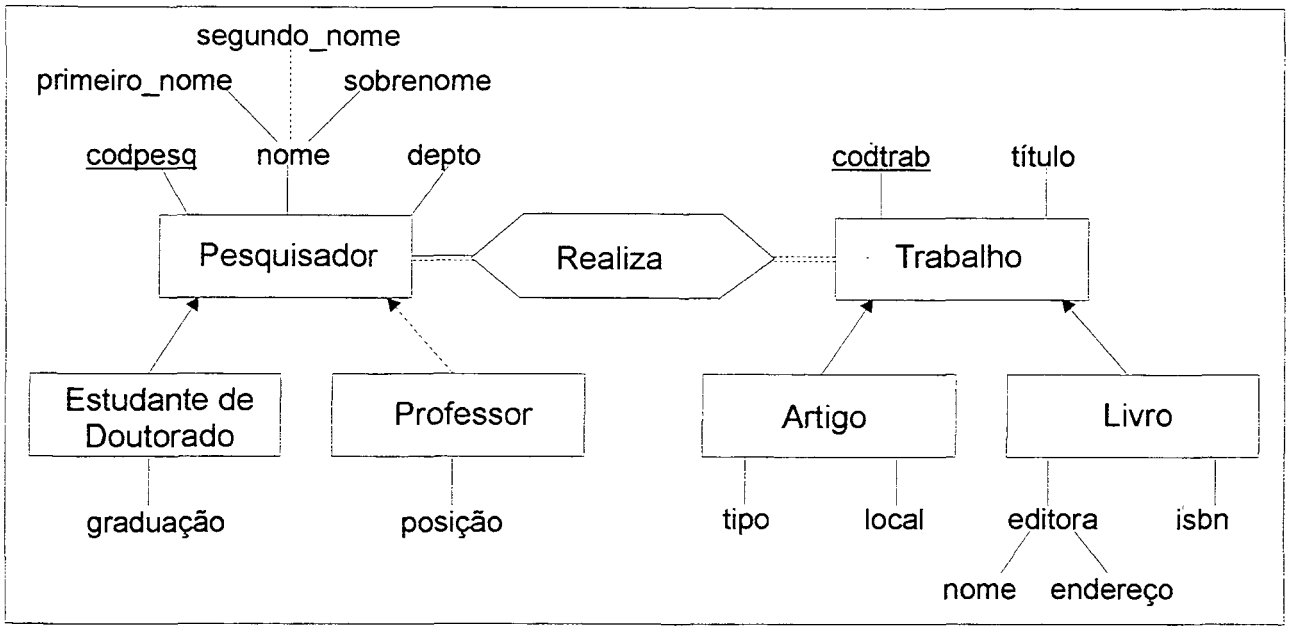


FIGURA 1: Objetos representados de acordo com o modelo ERC+.

Uma entidade é representada por um retângulo, sendo que o nome da entidade é inserido dentro do retângulo, com a inicial do nome em letra maiúscula.

Um atributo é representado em letras minúsculas com uma linha simples contínua unindo a sua entidade pai, se o atributo for obrigatório e uma linhas simples tracejada, se o atributo for opcional. Os atributos sublinhados indicam atributos identificadores.

Um relacionamento é representado por um losango com o nome do relacionamento em seu interior, com a inicial em letra maiúscula. Uma linha simples contínua identifica um relacionamento monovalorado e mandatório, 1:1; uma linha simples tracejada indica um relacionamento opcional monovalorado, 0:1; uma linha dupla tracejada representa um relacionamento multivalorado opcional, 0:n; e um relacionamento representado por uma linha tracejada e uma linha contínua indica uma representação mandatória multivalorada, 1:n.

Uma generalização "is-a" é indicada através de uma linha contínua com uma flecha apontando para a entidade genérica. Sendo que uma linha tracejada apontando para a entidade genérica indica uma generalização "*may-be-a*".

Como podemos visualizar na figura 1, a entidade Pesquisador está ligada à entidade Trabalho pelo relacionamento Realiza, que possui a seguinte semântica: "um pesquisador pode realizar zero ou mais trabalhos", e, " um trabalho pode ser realizado por um ou mais pesquisadores".

A entidade Pesquisador é uma generalização das entidades Estudante de Doutorado e Professor, onde a flecha formada por uma linha contínua indica uma generalização "*is-a*", que dita que todos os estudantes de doutorado são pesquisadores. A generalização indicada por uma flecha formada por uma linha tracejada indica uma generalização "*may-be-a*" que diz que os professores podem ou não ser pesquisadores. Por exemplo, no caso de um professor adjunto, onde o mesmo pode não exercer uma atividade de pesquisa.

Ainda na entidade Pesquisador, podemos verificar a existência de um atributo complexo, nome, que possui atributos obrigatórios, primeiro_nome e sobrenome, e um atributo opcional segundo_nome.

A entidade Trabalho é uma generalização das entidades Artigo e Livro, onde todos os artigos e livros são considerados trabalhos (generalizações "*is-a*").

Os atributos codpesq e codtrab são mostrados sublinhados por serem atributos chaves das entidades Pesquisador e Trabalho, respectivamente.

O modelo ERC+ é complementado com definições formais através de uma linguagem de manipulação, com uma álgebra associada (PARENT et al., 1989) para que seja possível manipular consultas em um banco de dados ERC+.

Podemos encontrar uma representação formal mais completa em PARENT e SPACCAPIETRA (1985) e PARENT e SPACCAPIETRA (1987); para uma comparação entre o modelo ERC+ e o modelo orientado a objeto devemos pesquisar em PARENT e SPACCAPIETRA (1989).

# 3. OS BANCOS DE DADOS DISTRIBUÍDOS

Em um sistema de banco de dados centralizado, todos os componentes do sistema residem em um único computador ou sítio. Estes componentes incluem os dados, o *software* SGBD (Sistema Gerenciador de Banco de Dados) e os demais componentes de *hardware* necessários para um sistema completo. Um banco de dados centralizado pode ser acessado remotamente via terminais conectados ao sítio.

A popularização das redes de computadores tornou possível o acesso a várias bases de dados permitindo a troca de informação. Este fato gerou a distribuição dos sistemas em múltiplos sítios que estão conectados via rede. Iremos discorrer, neste capítulo, sobre o desenvolvimento dos sistemas de bancos de dados distribuídos (DDBS – *Distributed Database Systems*) e sua classificação.

Podemos definir um banco de dados distribuído como sendo um conjunto de sistemas de bancos de dados que possuem facilidade para trocar dados e serviços entre si, porém estão espalhados em vários sítios e unidos através de uma rede.

Muitos fatores influenciaram o desenvolvimento dos bancos de dados distribuídos. Iremos citar, abaixo, algumas das potenciais vantagens destes sistemas:

- Distribuição natural de alguns sistemas de bancos de dados: muitos sistemas de bancos de dados são naturalmente distribuídos em diferentes sítios. Por exemplo, uma universidade possui vários campi situados em locais diferentes. É comum, neste caso, que os diferentes sistemas estejam localizados em sítios específicos. Muitos usuários locais acessam somente os dados de seu sítio específico, porém, os usuários gerais – tais como os administradores dos diversos campi – necessitam, muitas vezes, acessar as informações dos diversos sítios distribuídos. Podemos identificar, através deste exemplo, que os sítios locais descrevem uma situação minimizada do banco de dados global. A origem dos dados e a maioria dos usuários e aplicações estão localizadas em um sítio local.
- Aumento de segurança e disponibilidade dos dados: estas são as vantagens principais dos bancos de dados distribuídos. Existe uma maior segurança em um banco de dados distribuído, pois a probabilidade de um

sítio estar ativo, em um determinado momento, é muito maior do que se estivermos considerando um sistema centralizado. A disponibilidade dos dados recai sobre o fato de que estes dados podem estar ativos durante um maior período de tempo. Quando os dados estão distribuídos entre vários sítios, um sítio pode estar indisponível enquanto os outros estão ativos; assim, somente os dados que estão no sítio que está inativo ficarão inacessíveis. Este fato aumenta tanto a segurança quanto a disponibilidade dos dados. Para um aprimoramento, ainda maior, de um sistema de bancos de dados distribuídos, existe a possibilidade de replicação dos dados para mais de um sítio.

- Possibilidade de distribuição de alguns dados, enquanto outros continuam sendo controlados localmente: em alguns tipos de sistemas de bancos de dados distribuídos, conforme iremos verificar no decorrer deste capítulo, existe a possibilidade de controle dos dados e do *software* no sítio local. Porém alguns dados podem ser acessados por usuários de outros sítios através de acesso remoto. Isto permite o controle da distribuição dos dados através do sistema de bancos de dados distribuído.
- Aumento da *performance*: quando um banco de dados de grande porte é distribuído em múltiplos sítios, o mesmo é transformado em pequenos bancos de dados distribuídos. Como resultado, consultas e transações locais, acessando os dados de um único sítio, possuem uma *performance* melhor, devido ao tamanho do banco de dados. No caso das transações que precisam ser realizadas em mais de um sítio, as mesmas podem ser realizadas em paralelo, reduzindo o tempo de resposta.

A distribuição dos dados conduz ao aumento da complexidade da modelagem e da implementação do sistema. Para que um sistema de bancos de dados distribuídos possua todas as potencialidades listadas acima, o mesmo deve ter as seguintes funcionalidades:

- Habilidade de acesso aos sítios remotos, assim como a transmissão das consultas e dados através dos vários sítios via comunicação de rede.
- Aptidão de manter o rastreamento da distribuição e replicação dos dados no catálogo do sistema de bancos de dados distribuídos.

- Capacidade de planejar execuções estratégicas das consultas e das transações que acessam os dados de mais do que um sítio.
- Inteligência de decidir qual dado replicado deve ser acessado.
- Habilidade de manter a consistência das cópias dos dados replicados.
- Competência de recuperação das falhas dos sítios individuais assim como dos demais problemas de comunicação que possam ocorrer.

No caso de *hardware*, os seguintes fatores distinguem um banco de dados distribuído de um banco de dados centralizado:

- Existência de muitos computadores, chamados sítios ou nodos.
- Os sítios devem estar conectados por uma rede estruturada, para que possam transmitir os dados entre si.

Estes sítios podem estar localizados muito perto fisicamente, ou seja, no mesmo prédio ou sala, porém os mesmos devem estar conectados via uma rede local. Os sítios podem também estar localizados em regiões geográficas distantes e conectados por uma rede de longa distância.

O principal objetivo de um banco de dados distribuído é que os usuários possam acessar os sítios distribuídos como se os mesmos fossem um único banco de dados.

## 3.1. CLASSIFICAÇÃO DOS SISTEMAS DE BANCOS DE DADOS DISTRIBUÍDOS

O termo banco de dados distribuído pode descrever vários sistemas que possuem diferenças básicas entre si. Iremos descrever, nesta seção, os tipos de bancos de dados distribuídos e os critérios e fatores que os diferenciam.

O primeiro fator que é considerado é o grau de homogeneidade dos bancos de dados distribuídos (DDBS - *Distributed Database Systems*). Se todos os sítios utilizarem os mesmos *softwares*, o DDBS é chamado homogêneo, caso contrário, chama-se heterogêneo.

Outro fator, relacionado ao grau de homogeneidade, é o grau de autonomia local. Se todos os acessos ao DDBS tiverem que ser realizados via um cliente, então o sistema não possui autonomia local. Por outro lado, se existe a possibilidade de um acesso direto ao servidor, através de transações locais, o sistema possui um certo grau de autonomia.

Um terceiro aspecto que pode ser analisado, para categorizar os sistemas de bancos de dados distribuídos, é o grau de distribuição ou transparência, ou, alternativamente, o grau de integração dos esquemas. Este aspecto indica como as consultas são realizadas no banco de dados distribuído.

De acordo com os fatores citados acima, e conforme a figura 2, os sistemas de informações distribuídas são classificados em:

- Sistemas de banco de dados distribuído homogêneo.
- Sistemas multi banco de dados.
- Sistemas de banco de dados federado.

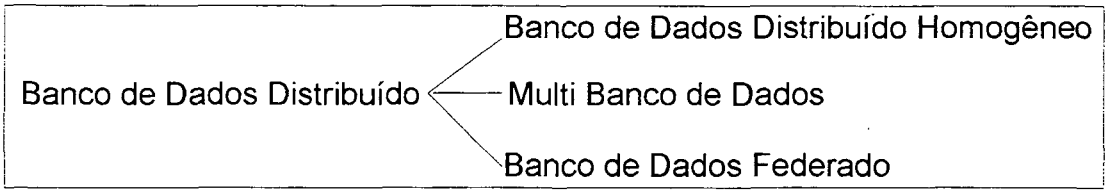


FIGURA 2: Classificação dos bancos de dados distribuídos.

### 3.1.1. Sistemas de bancos de dados distribuídos homogêneos

Um banco de dados distribuído é homogêneo quando o modelo de dados for o mesmo em todos os sítios (exemplo: relacional), os componentes de *software* (que executam as transações) forem os mesmos e compatíveis entre eles.

Os sistemas de bancos de dados distribuídos homogêneos são caracterizados pela utilização do mesmo método de acesso, estratégias de otimização, concorrência e modelos de dados. Podemos visualizar a arquitetura de um banco de dados distribuído homogêneo, através da figura 3, composta por *software* e *hardware* IBM.

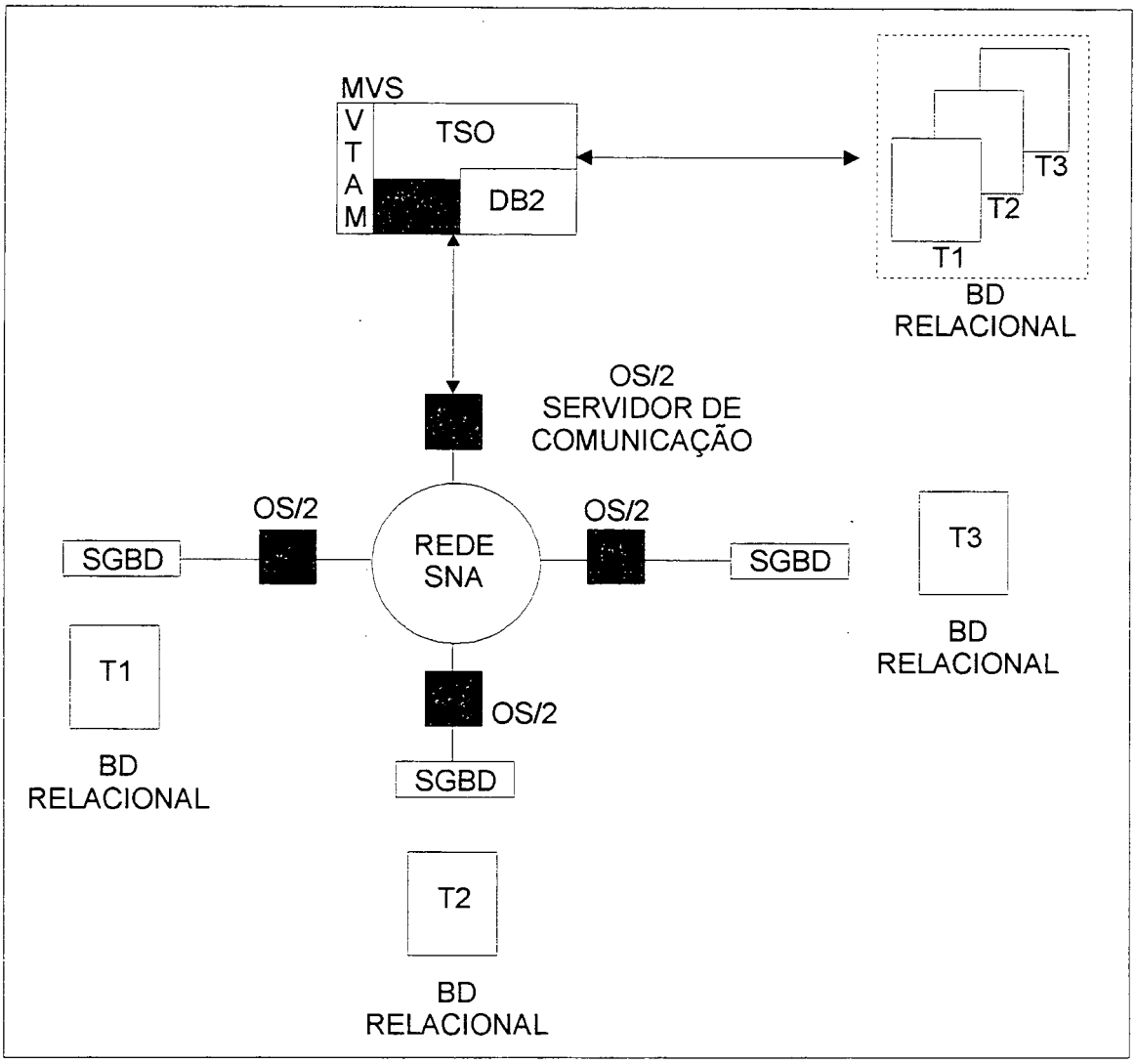


FIGURA 3: Arquitetura de um banco de dados distribuído homogêneo.

Esta arquitetura de sistema de banco de dados distribuído é composta por um sistema principal, sendo controlado pelo sistema operacional MVS, um sistema gerenciador de banco de dados relacional DB2 da IBM, *software* de comunicação VTAM e interface de *software* TSO.

O banco de dados relacional gerenciado pelo DB2 possui três tabelas: T1, T2 e T3.

A área tracejada representa a camada superior do sistema gerenciador de banco de dados distribuído.

Neste caso particular, o banco de dados distribuído será implementado através da leitura, no servidor de cada sítio, de uma das tabelas do banco de dados original. A tabela T1 apontará para o sítio 1, a tabela T2 para o sítio 2 e a tabela 3 para o sítio 3.

Quando o usuário solicitar uma consulta, a mesma será submetida à camada superior do *software* e distribuída para cada um dos sítios envolvidos na consulta.

Cada sítio executa sua parte e retorna o resultado ao servidor central, onde a junção dos dados é realizada. O resultado final é mostrado ao usuário que requisitou a consulta.

### 3.1.2. Sistemas multi banco de dados

Os sistemas multi banco de dados são caracterizados por modelos de dados, estratégias de concorrência, otimização e métodos de acesso não equivalentes.

Diferenciam-se dos bancos de dados distribuídos homogêneos pelo fato de que os modelos de dados que compõem o banco de dados global podem ser baseados em modelos relacionais, hierárquicos, de rede ou algum outro tipo de modelo de dados.

Pelos motivos descritos acima os sistemas multi banco de dados são chamados heterogêneos.

A figura 4, mostra um exemplo de uma arquitetura de multi banco de dados que é composta por uma camada central e dois sítios distribuídos. A camada central controla o acesso ao dicionário de dados ou banco de dados global. Cada sítio local possui um camada que acessa o SGBD local e o banco de dados. As camadas de acesso locais, juntamente com a camada de acesso central, compõem o multi banco de dados.

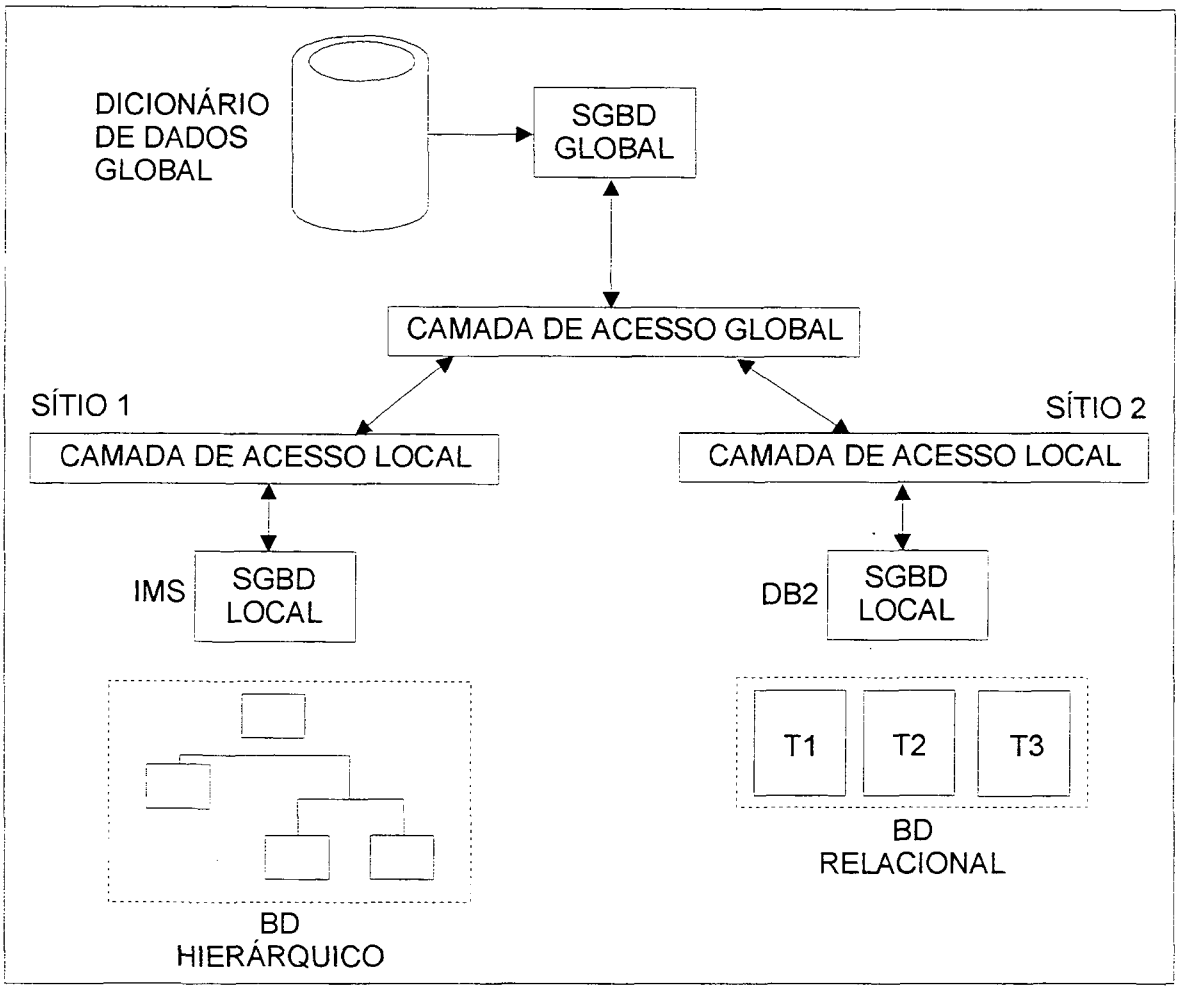


FIGURA 4: Arquitetura de um multi banco de dados.

Podemos verificar, através da figura 4, que este é um sistema heterogêneo pois os componentes locais estão baseados em dois diferentes SGBD's, que utilizam dois modelos de dados diferentes.

O dicionário de dados global contém informações que fazem com que estes dois bancos de dados sejam visualizados pelo usuário como um único.

Quando o usuário requer uma consulta distribuída, na camada global, a consulta é decomposta e transformada, para que possa ser executada nos bancos de dados locais. O usuário não percebe esta decomposição, pois quem gerencia os dados é a camada global.

### 3.1.3. Sistemas de bancos de dados federados

Os sistemas de bancos de dados federados são um caso especial de um sistema multi banco de dados. São completamente autônomos, não se baseiam no

dicionário global de dados para processar as consultas distribuídas e cada sítio pode associar-se, ou sair do sistema multi banco de dados, sem afetar os outros membros.

Como mencionamos em parágrafo anterior, os sistemas de bancos de dados federados não usam um dicionário de dados ou esquema global para processar a consulta distribuída. Cada nodo da federação possui um esquema de exportação e um de importação.

O esquema de *exportação* é utilizado para identificar os objetos de dados que o nodo irá compartilhar com os outros nodos da federação. O esquema de importação contém informações sobre a descrição dos objetos de dados que os outros nodos da federação irão compartilhar com este nodo. Deste modo, a consulta distribuída gerada, em cada nodo, é definida de acordo com a informação representada no esquema local de importação.

Os membros da federação concordam na utilização de protocolos de comunicação e métodos comuns para rotear as consultas e os dados distribuídos.

Através da figura 5 podemos visualizar um exemplo simples de uma arquitetura de um banco de dados federado composto por dois nodos. Temos um sistema com SGBD IMS (modelo hierárquico) no sítio 1 e, no sítio 2, um SGBD ORACLE (modelo relacional).

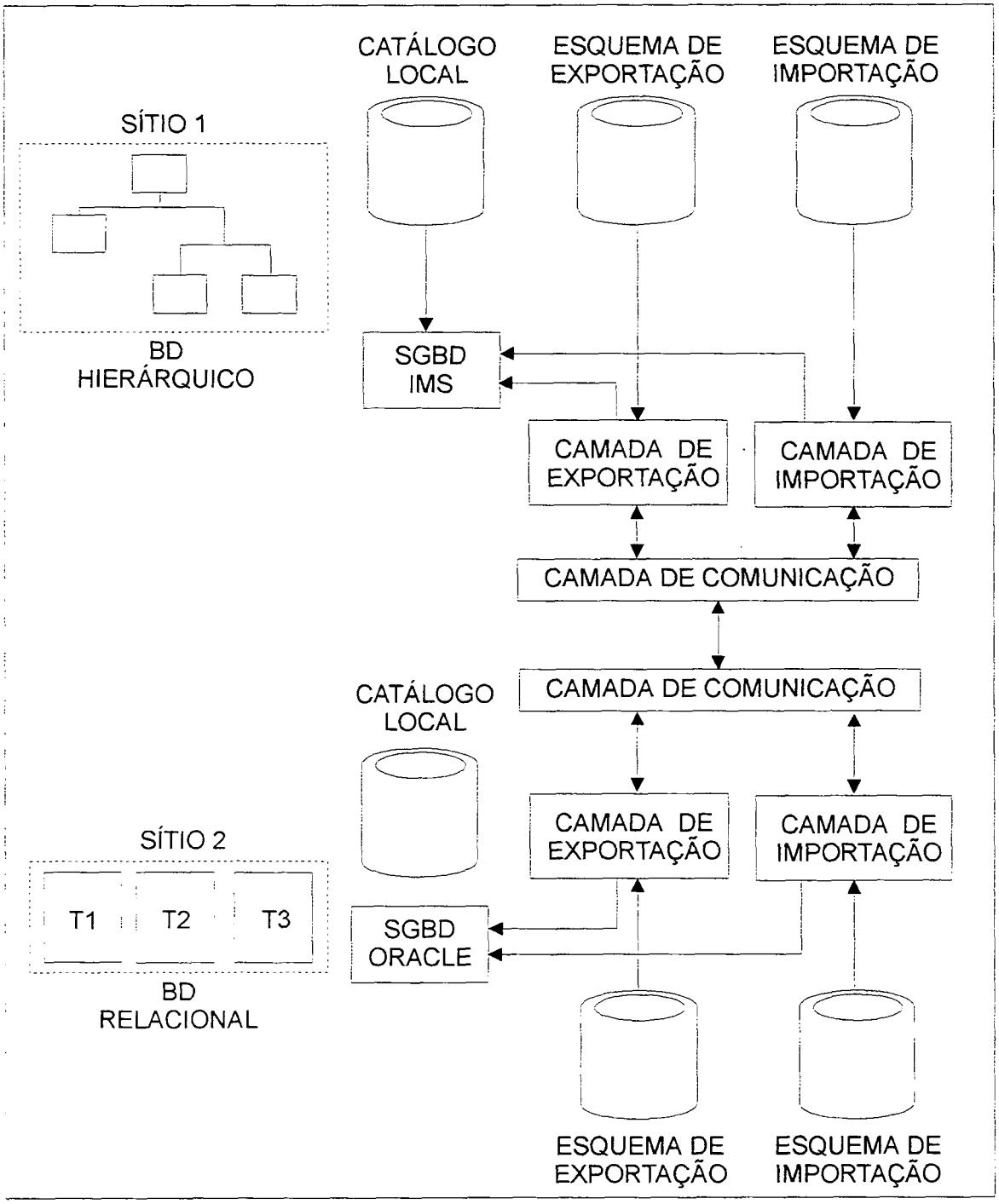


FIGURA 5: Arquitetura de um banco de dados federado.

## 4. A INTEGRAÇÃO DE BANCOS DE DADOS

As pesquisas atuais em integração de bancos de dados heterogêneos indicam duas áreas de pesquisa muito populares neste tema: a integração de esquemas globais e a integração de esquemas federados (SHETH e LARSON, 1990; BRIGHT; HURSON e PAKZAD, 1992).

Na integração de esquemas globais, esquemas correspondentes a cada banco de dados local são combinados em um único esquema integrado ou esquema global.

Por outro lado, na integração de bancos de dados federados, cada banco de dados local possui seu esquema de exportação, que corresponde à parte do esquema global que irá compartilhar os dados com os outros esquemas a serem integrados.

A partir destes esquemas, os administradores de bancos de dados locais podem usá-los para definir um esquema de importação – um esquema global parcial – representando as informações dos bancos de dados remotos que podem ser acessadas localmente.

As metodologias que utilizam os dois métodos citados acima, para realizar a integração de bancos de dados, são as principais, pois garantem uma maior interoperabilidade das bases heterogêneas.

A integração de esquemas é o processo de geração de um ou mais esquemas integrados a partir de esquemas iniciais. Estes esquemas iniciais representam a semântica dos bancos de dados a serem integrados e são usados como entrada no processo de integração. A saída do processo é um ou mais esquemas integrados, representando a semântica dos bancos de dados iniciais.

Os esquemas resultantes são representados usando um modelo de dados comum, sendo que escondem qualquer discrepância existente entre a semântica dos dados dos esquemas iniciais, ou mesmo qualquer desigualdade decorrente dos modelos nos quais os bancos de dados estão baseados. Estes esquemas devem possibilitar consultas aos múltiplos bancos de dados integrados.

Através da existência do esquema integrado, ou esquema global, os usuários não precisam saber da existência dos vários bancos de dados ou da localização dos

dados, pois a integração de esquemas fornece transparência na localização, distribuição e replicação dos dados, assim como na distinção dos modelos.

O termo integração de esquemas tem sido muito utilizado na literatura referindo-se a metodologias que facilitam a integração de esquemas, como definimos acima, da mesma forma que na integração de visões. Este fato decorre de que muitas das técnicas aplicadas no contexto de integração de esquemas são também utilizadas na integração de visões e vice-versa.

Contudo, os dois processos possuem diferenças importantes (SHETH e LARSON, 1990; SPACCAPIETRA; PARENT e DUPONT, 1992).

As metodologias de integração de visões trabalham com situações onde as visões:

- São homogêneas, ou seja, baseadas no mesmo modelo de dados.
- Não refletem o armazenamento real dos dados do banco de dados. Uma visão é um conjunto de tabelas que são derivadas de outras tabelas. Estas outras tabelas podem ser tabelas do banco de dados ou outras visões. Uma visão pode ser considerada uma tabela virtual, pois os dados que a compõem podem não estar armazenados, fisicamente, no banco de dados, ao contrário de tabelas, onde os dados estão armazenados, fisicamente, no banco de dados.
- Qualquer alteração nas tabelas do banco de dados base, que afetem uma visão, podem torná-la inválida.

Ao contrário, nas metodologias de integração de esquemas, os esquemas iniciais:

- Podem estar baseados em diferentes modelos de dados.
- Descrevem, exatamente, os dados que estão armazenados no banco de dados.
- São implementados em um SGBD.

O fato da complexidade da integração de esquemas, conforme já comentamos na introdução deste trabalho, nos leva a verificar que o processo de integração de esquemas não pode ser totalmente automatizado (SHETH e GALA, 1989; RAM e BARKMEYER, 1991). Apesar disto, algumas ferramentas foram desenvolvidas para minimizar o trabalho humano, conforme iremos comentar mais tarde, em seção específica deste trabalho.

Devido a toda problemática que envolve o processo de integração de esquemas, devemos nos deter ao fato de que o mesmo não é realizado em uma só etapa.

Após a construção do esquema integrado, o qual representa os bancos de dados iniciais, mudanças são necessárias se:

- Ocorrerem mudanças nas estruturas dos bancos de dados iniciais.
- As regras de restrições dos bancos de dados iniciais forem alteradas.

Assim, podemos verificar que um esquema integrado eficiente deve ser dinâmico ao ponto de poder sustentar tais mudanças dos bancos de dados iniciais.

Iremos utilizar, nesta pesquisa, metodologias que trabalham na integração de esquemas. Para isso iremos detalhar as fases de desenvolvimento de uma metodologia *típica* de integração de esquemas, que pode ser dividida em três fases (RAMESH e RAM, 1995), conforme a figura 6, abaixo:

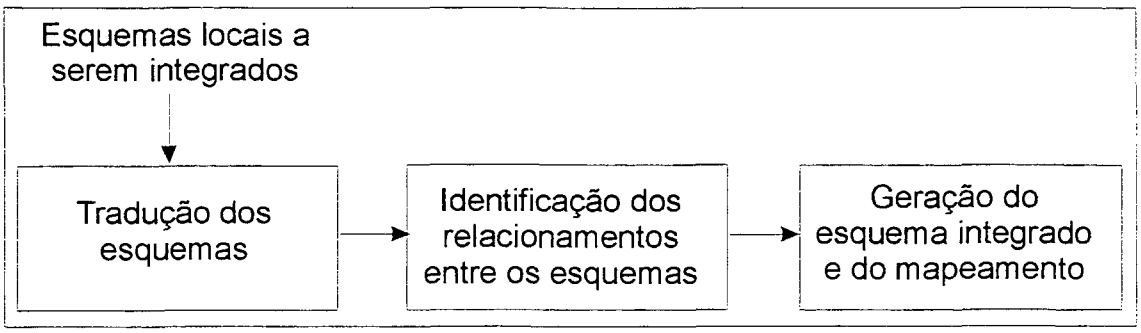


FIGURA 6: Fases de uma metodologia de integração de esquemas.

Na primeira fase da integração, que é a tradução dos esquemas, os esquemas que correspondem aos bancos de dados iniciais são modelados utilizando o mesmo modelo de dados. Tradicionalmente, utiliza-se um modelo de dados semântico, como o entidade relacionamento (CHEN, 1976).

Qualquer técnica de tradução de esquemas deve seguir as seguintes características:

- O esquema gerado através do modelo de dados comum deve representar fielmente a semântica dos bancos de dados iniciais.
- Deve ser possível realizar, no esquema traduzido (esquemas dos bancos de dados iniciais modelados com o mesmo modelo de dados), os mesmos comandos que são realizados no esquema inicial.

Iremos utilizar os esquemas S1 e S2 para exemplificar as três fases principais do processo de integração de bancos de dados heterogêneos.

Objetos do esquema S1:

Usuário (matrícula, login, senha, nome, endereço)

O objeto Usuário armazena dados de funcionários e professores.

Departamento (departamento, nome_dept)

Este objeto armazena dados dos departamentos.

Objetos do esquema S2:

Usuário (matrícula, nome, endereço)

O objeto usuário de S2 mantém as informações dos alunos, funcionários e professores que utilizam uma biblioteca.

Biblioteca (código, nome, endereço)

Este objeto armazena os dados das bibliotecas.

A figura 7 mostra os diagramas dos dois esquemas, S1 e S2, modelados através do modelo ERC+, que seria o primeiro produto de trabalho do processo de integração.

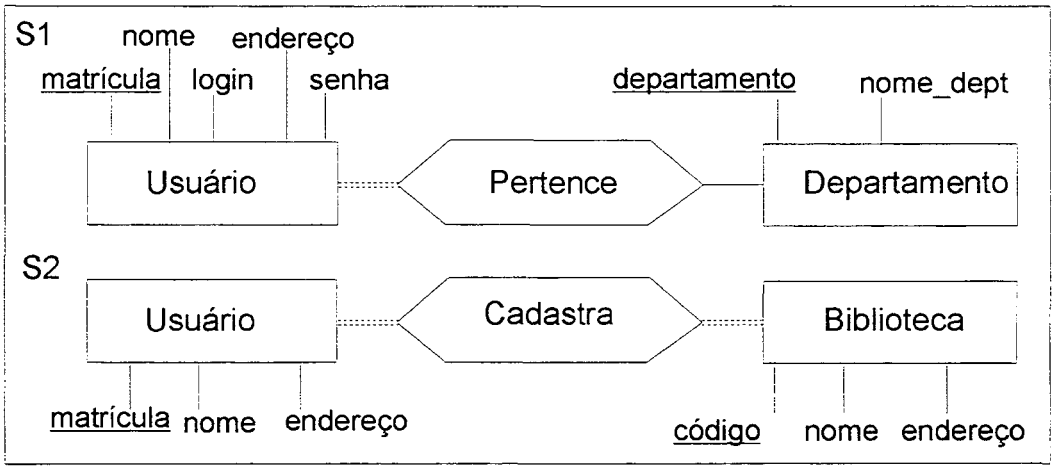


FIGURA 7: Diagramas ERC+ dos esquemas S1 e S2.

A segunda fase da integração, que se refere à geração de um inter-esquema de relacionamentos, tem o objetivo de identificar os objetos que podem estar relacionados nos esquemas traduzidos. Como, por exemplo, entidades, atributos e relacionamentos, e categorizar os relacionamentos existentes entre eles.

Esta tarefa é realizada através da análise da semântica dos objetos diferentes de cada banco de dados, verificando as propriedades das entidades, dos atributos e dos relacionamentos.

Nesta fase é importante a interação com os desenvolvedores, para que seja explorado o total entendimento dos sistemas, investigando as regras de integridade, as cardinalidades dos relacionamentos e os domínios dos atributos.

A última atividade desta fase é a geração de um conjunto de informações relevantes sobre os objetos dos bancos de dados. É muito importante que estas informações sejam acuradas porque elas serão utilizadas como entrada para a fase de geração do esquema integrado.

Por exemplo, no caso dos esquemas S1 e S2, podemos identificar as entidades Usuário em S1 e Usuário em S2 como relacionadas entre si, pois o conjunto de usuários do esquema S1 é um sub conjunto dos usuários de S2. Isto ocorre porque o domínio da entidade Usuário do esquema S2 abrange alunos, professores e funcionários e o domínio da entidade Usuário no esquema S1 refere-se a funcionários e professores, sendo assim, esta entidade pode ser generalizada.

A terceira fase da integração, que é a geração do esquema integrado, abrange resolver vários problemas de conflitos de heterogeneidade que podem existir entre os objetos dos bancos de dados.

O processo de geração do esquema integrado resolve estas diferentes categorias de problemas de heterogeneidade e gera um esquema integrado que esconde estas diversidades do usuário final.

De acordo com a figura 8, abaixo, podemos verificar que o esquema integrado que corresponde a figura 7 da página 27, generaliza a entidade Usuário e integra os atributos em comum na super classe. Temos que mudar o nome da entidade Usuário de S2 porque uma entidade específica não pode possuir o mesmo nome de uma entidade genérica.

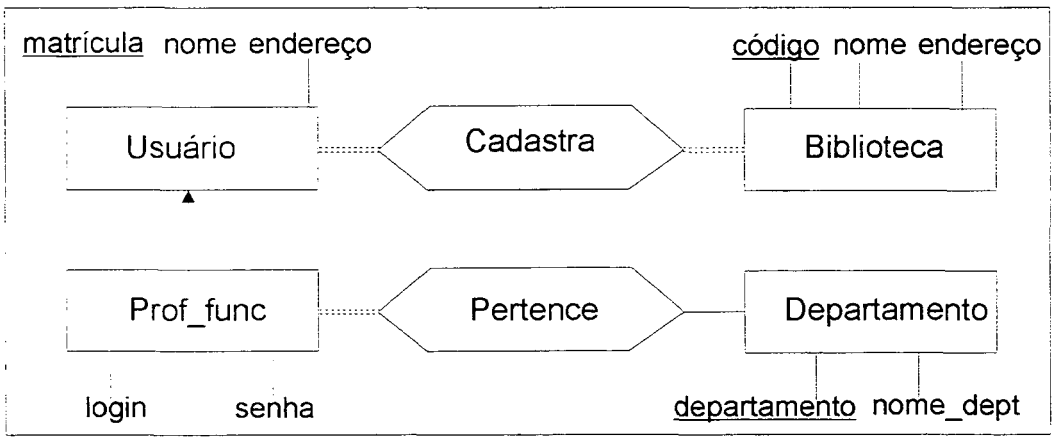


FIGURA 8: Esquema integrado de S1 e S2.

Uma outra atividade desta última fase do processo de integração refere-se à geração do mapeamento dos esquemas. Esta atividade envolve o armazenamento de informações sobre o mapeamento dos objetos do esquema integrado para os objetos dos bancos de dados locais. Este mapeamento é importante para que as consultas aos bancos de dados sejam realizadas com êxito.

É importante salientarmos que estas fases devem ser realizadas interativamente, para resolver os conflitos de heterogeneidade, e obter, como resultado, um esquema integrado coerente com os bancos de dados iniciais.

Podemos ilustrar outro exemplo da aplicação dos passos da integração, através das figuras 9, 10 e 11. Na figura 9, mostramos os diagramas de duas visões do banco de dados de uma biblioteca.

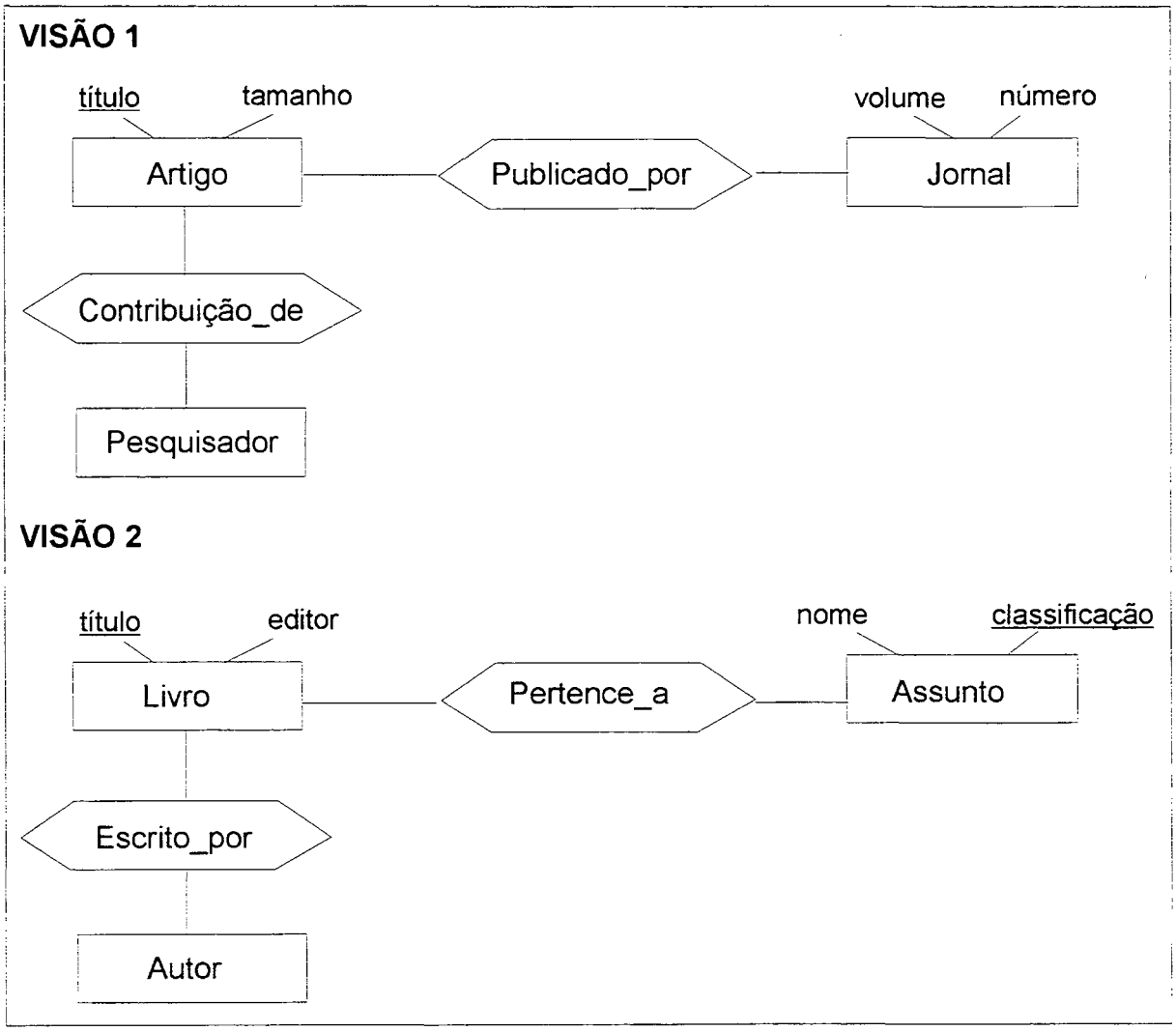


FIGURA 9: Diagrama de duas visões do banco de dados de uma biblioteca.

Durante a identificação da correspondência entre as duas visões, descobrimos que as entidades Pesquisador e Autor são sinônimos, assim como os relacionamentos Contribuição_de e Escrito_por.

De acordo com estes fatos, decidimos por modificar a Visão 1, mudando o nome da entidade Pesquisador para Autor e o nome do relacionamento Contribuição_de para Escrito_por, como mostramos na figura 10.

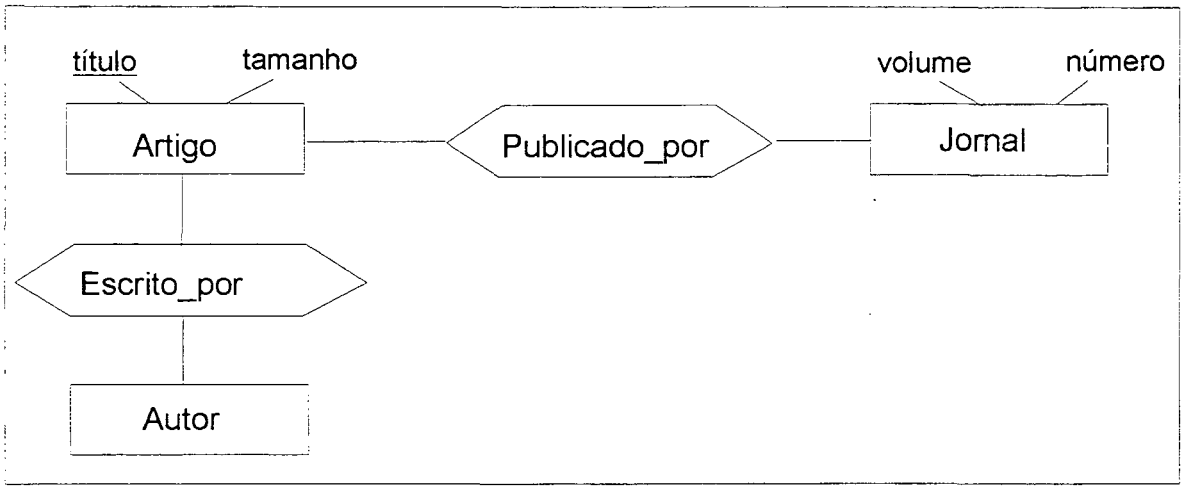


FIGURA 10: Modificação da Visão 1.

No esquema integrado, conforme podemos visualizar na figura 11, na próxima página, os relacionamentos Contribuição_de e Escrito_por são unidos no relacionamento Escrito_por. Assim como as entidades Autor e Pesquisador são unidas na entidade Autor.

Nós também generalizamos as entidades Artigo e Livro na entidade Publicação, com seu atributo comum título. O atributo editor e o relacionamento Pertence_a aplicam-se somente a entidade Livro, assim como o atributo tamanho e o relacionamento Publicado_por aplicam-se somente a entidade Artigo.

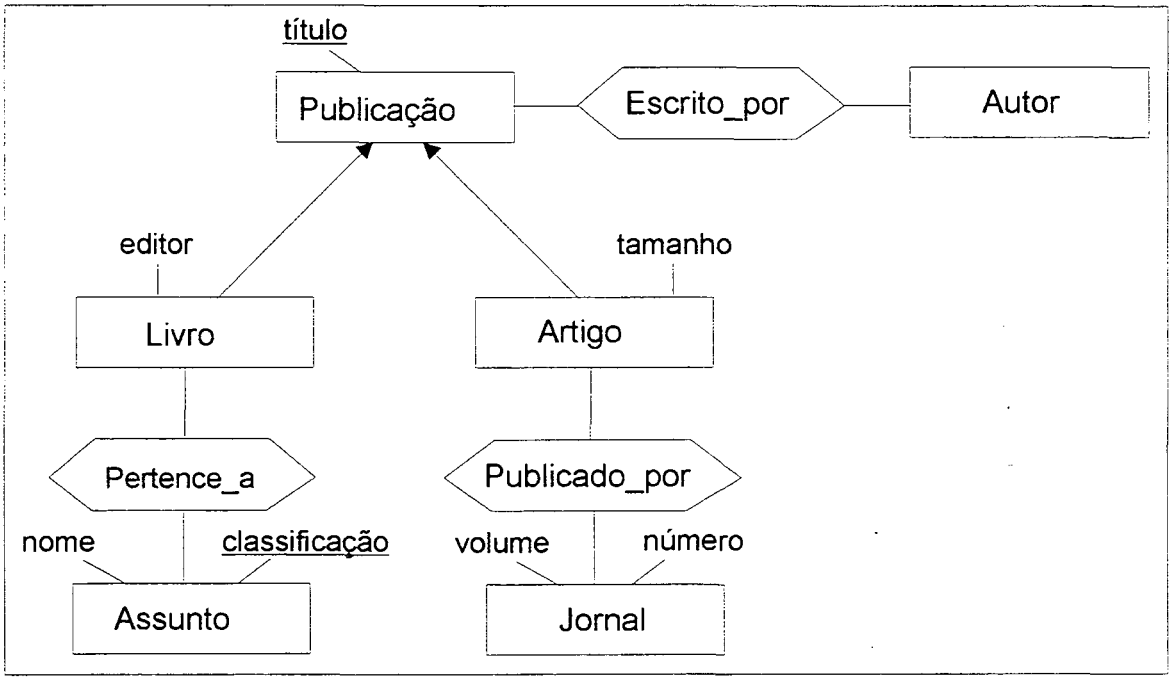


FIGURA 11: Esquema integrado após a união das visões 1 e 2.

## 4.1. OS CASOS DE CONFLITO

A representação dos conflitos é um dos principais desafios das metodologias de integração.

Dois desenvolvedores modelando um mesmo universo do discurso provavelmente irão descrever o mesmo objeto do mundo real de maneiras diferentes. Isto acontece porque os desenvolvedores podem ter diferentes percepções do mundo real, diferentes visões das informações ou devido ao uso de diferentes técnicas e ferramentas de modelagem.

SHETH e KASHYAP (1993) dividem os conflitos estruturais em duas classes: incompatibilidade de definições de domínio e incompatibilidades de definições de entidades.

Os conflitos de definições de domínio incluem problemas de nomes (homônimos e sinônimos), tipo, unidade, precisão, valor padrão e regras de integridade dos dados.

Os conflitos de definição de entidades abrangem equivalência das chaves, compatibilidade de uniões, isomorfismo dos esquemas e problemas de falta de itens de dados.

Fora desta classificação estão os conflitos de incompatibilidades referentes a abstrações e discrepância dos esquemas.

Iremos descrever, brevemente, os conflitos citados em SHETH e KASHYAP (1993).

Conflitos **sinônimos** ocorrem quando um conjunto de objetos equivalentes são designados com nomes diferentes; por exemplo, uma entidade Estudante em um esquema pode descrever o mesmo conceito de uma entidade Aluno em outro esquema.

Conflitos **homônimos**, por outro lado, ocorrem quando objetos que têm significados diferentes possuem o mesmo nome; por exemplo, uma entidade Colaborador pode representar professores de uma universidade em um esquema e funcionários de um departamento em outro esquema.

Conflitos de **tipos de dados** ocorrem quando objetos equivalentes têm diferentes tipos de dados. Na modelagem dos dados, podemos citar como exemplo o conceito de Departamento que pode ser uma entidade em um esquema e um atributo em outro.

O uso de diferentes unidades de medidas para descrever objetos iguais, como o preço expresso em diferentes moedas, resulta em conflitos de **unidade**.

Similarmente, os conflitos de **precisão** referem-se ao uso de diferentes granularidades para entidades equivalentes em diferentes esquemas.

Estes dois últimos casos podem ser resumidos como uma ocorrência de **conflito de domínio**, onde um atributo pode ter diferentes domínios em dois esquemas. Por exemplo, ISBN pode ser declarado como um inteiro em um esquema e como caracter em outro. Um conflito de unidade de medida pode ocorrer em um esquema que representa peso em gramas e outro em quilogramas.

Os problemas de **equivalência de chaves** resultam de duas ou mais relações modelando a mesma entidade através de chaves semanticamente diferentes.

Quando uma chave comum não é detectada, este conflito pode acarretar problemas na recuperação e gravação dos dados de diferentes entidades através de uma consulta simples ou múltipla.

Outro exemplo envolve restrições de diferenças estruturais em relacionamentos como Lecionar; onde um esquema pode representá-lo como 1:N

(um curso pode ser ministrado por um professor), enquanto que outro esquema pode representá-lo como M:N (um curso pode ser ministrado por mais do que um professor).

Uma **incompatibilidade de união** *entre* duas relações é gerada quando o número ou o domínio dos atributos não são equivalentes, ou, quando um mapeamento um para um entre os respectivos conjuntos de atributos não existe. Em RUSINKIEWICZ e CZEJDO (1987) o operador de união externa foi definido para tratar deste problema.

O conflito de **isomorfismo dos esquemas** refere-se ao uso de um número diferente de atributos para descrever objetos semanticamente similares. Um exemplo típico deste caso é a representação do atributo "nome" como um único atributo em um caso, e como "primeiro_nome" e "último_nome" em outro caso.

O problema de **falta de itens de dados** acontece quando objetos são definidos como um conjunto de atributos em um esquema e por um sub conjunto destes atributos em outro esquema. Algumas vezes, esta falta de atributos pode ser deduzida através de um mecanismo de inferência ou por valores padrões. Por exemplo, o valor "grad_est" de um atributo de uma entidade "estudante" pode ser deduzido como sendo "estudante graduado" e assim combinado com o valor explícito "estudante graduado" de um outro atributo.

As **incompatibilidades no nível de abstração** referem-se a conflitos de generalizações e agregações. Como um exemplo de agregação, consideramos uma entidade Publicação, que é representada em dois esquemas diferentes pela entidade Publ(num_publ, autor, título) em um esquema e pelas entidades Livro(isbn, autor, título) e Artigo(issn, autor, título) em outro esquema. Como podemos verificar, o primeiro esquema define a mesma entidade em um nível de abstração mais geral.

O conflito de **discrepância de esquemas** surge quando um dado em um esquema corresponde a um meta-dado em outro. A resolução deste caso normalmente requer a utilização de uma linguagem de programação, como a descrita em KRISHNAMURTHY; LITWIN e KENT (1991) que permite que as referências dos dados e dos meta-dados possam ser mescladas em uma única especificação.

Outra classificação que utilizamos para detectar e qualificar os casos de conflitos é a empregada por SPACCAPIETRA; PARENT e DUPONT (1992), que

enfatizam quatro razões que levam os desenvolvedores a representar um mesmo objeto do mundo real de maneiras diferentes:

- Os desenvolvedores não têm percepções iguais sobre o mesmo conjunto de objetos do mundo real, mas sim uma sobreposição dos conjuntos (inclusão ou interseção). Este tipo de conflito é chamado de **conflito semântico**. O conceito da generalização é utilizado para resolver este conflito. Como exemplo podemos citar o caso das entidades Usuário dos esquemas S1 e S2 (figura 7, página 27), pois o conjunto de usuários do esquema S1 é um sub conjunto dos usuários do esquema S2, visto que o domínio da entidade Usuário de S2 abrange alunos, professores e funcionários e do esquema S1 abrange somente os funcionários e professores de pós-graduação.
- Quando estão descrevendo conjuntos de objetos do mundo real onde existe uma relação entre eles, os desenvolvedores podem perceber propriedades diferentes em cada um. Este conflito é chamado **conflito descritivo** e inclui colisões de nome, como homônimos e sinônimos, domínio de atributos, escala, restrições e operações. Citamos aqui o caso do Sistema Administrativo Universitário que denomina Estudante para a entidade que armazena os dados de todos os alunos da UFPR, sendo que no Sistema de Controle de Pesquisa e Pós-Graduação a entidade que contém os dados dos alunos de pós graduação é chamada Pesquisador.
- Os desenvolvedores utilizam diferentes modelos de dados, por exemplo relacional e orientado a objeto; este caso é chamado **conflito heterogêneo**.
- Como último caso é citado o **conflito estrutural**, onde, utilizando o mesmo modelo de dados, desenvolvedores descrevem um objeto de maneiras diferentes em dois esquemas. Por exemplo, no modelo orientado a objeto um desenvolvedor pode escolher entre definir um objeto O como um componente de um objeto, como um novo objeto ou como um atributo do objeto O.

  A possibilidade de existência de conflitos estruturais dependerá do poder da semântica relativista (multiplicidade de possíveis representações de

um determinado objeto no mundo real), do modelo de dados utilizado e de sua habilidade para representar a mesma realidade de maneiras diferentes. Modelos semânticos e orientados a objetos possuem uma semântica relativista maior do que o modelo relacional.

Estes conflitos podem ser acumulados, pois discrepâncias entre os esquemas usualmente recaem sobre uma variedade de conflitos. Diferentes modelos de dados ou diferentes conjuntos de objetos do mundo real geram diferentes estruturas. Como exemplo iremos citar os seguintes esquemas, conforme a figura 12, abaixo:

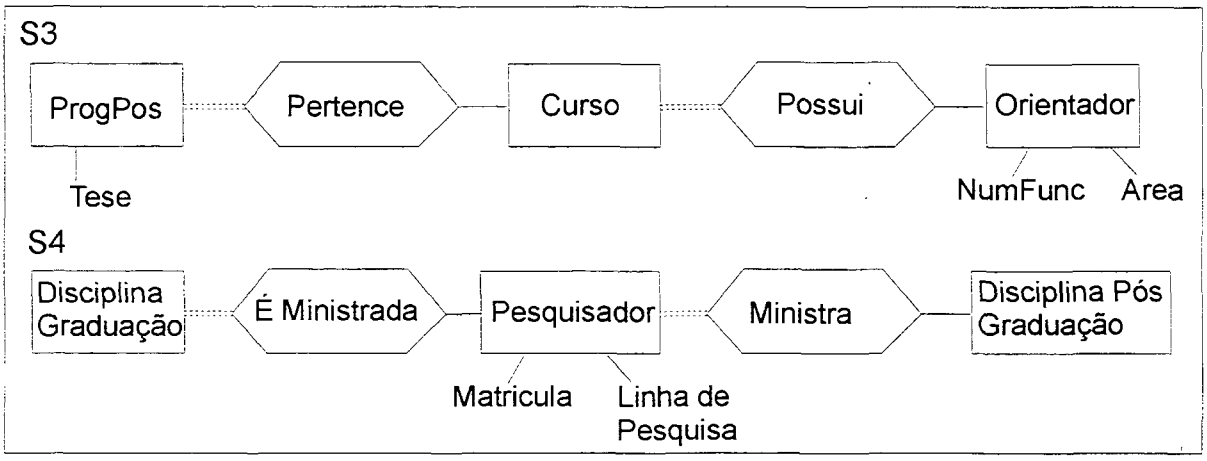


FIGURA 12: Diagrama dos esquemas S3 e S4.

Ambos os esquemas, S3 e S4, representam informações de professores de pós graduação. Iremos detectar, a partir destes esquemas, mostrados na figura 12, acima, alguns exemplos de conflitos.

O esquema S4 trata as disciplinas de graduação e de pós graduação, através das entidades Disciplina Graduação e Disciplina Pós Graduação, enquanto que o esquema S3, trata só as disciplinas de pós graduação, através da entidade ProgPos, temos, então, um conflito semântico.

Entre as entidades Orientador (S3) e Pesquisador (S4) temos dois tipos de conflitos descritivos. O primeiro é muito claro de identificar, existe um conflito de nomes sinônimos entre estas entidades, pois as duas representam o mesmo conjunto de objetos do mundo real e possuem nomes sinônimos. O segundo conflito descritivo, refere-se aos atributos das entidades Orientador(numfunc, area), em S3, e Pesquisador(matricula, linha de pesquisa), em S4, que são diferentes.

Uma informação sobre uma disciplina de pós graduação é obtida, em S4, através da entidade Disciplina de Pós Graduação, enquanto que em S3 é necessário

unir as informações das entidades ProgPos e Curso, este é um caso de conflito descritivo de operações.

Os casos de conflitos devem ser analisados com muito cuidado durante o processo de integração para que o esquema integrado represente os dados dos esquemas iniciais de uma maneira correta.

## 4.2. A INTEGRAÇÃO AUTOMATIZADA

Como pudemos verificar, pelas explanações citadas neste capítulo, o processo de integração de banco de dados, seja ele de esquemas ou de visões, é realizado de uma maneira exaustiva e complexa, que demanda grande interação entre os desenvolvedores dos sistemas e os usuários e consome bastante tempo dos mesmos.

Sendo assim, a automatização deste processo seria altamente desejável. Porém, esta automação apresenta um grande número de desafios para ser realizada com sucesso.

Estudos como os de SHETH e GALA (1989) puderam comprovar que o processo de integração de esquemas não pode ser totalmente transformado em um sistema automático devido à necessidade de interação com os desenvolvedores dos bancos de dados.

Isto acontece, principalmente, porque o processo de integração necessita do entendimento da semântica dos bancos de dados iniciais, usando representações do conhecimento que não podem ser totalmente repassadas para um *software*.

Outro motivo que dificulta a automação do processo de integração é que dois esquemas podem ser integrados de maneiras diferentes baseados na intenção de uso que têm (SHETH, 1991).

A automação que pode ser realizada está concentrada nas tarefas básicas, que não necessitam da interação dos desenvolvedores ou dos usuários.

Um dos primeiros esforços neste sentido foi realizado por DeSOUZA (1986), que realizou um trabalho focado na identificação dos relacionamentos entre os esquemas. O autor apresenta um sistema especialista que realiza a integração de esquemas conceituais definidos pelo Esquema Conceitual Abstrato – ACS (*Abstract Conceptual Scheme*) (STOCKER e CANTIE, 1983).

Um conjunto de funções, chamadas funções de semelhança (*resemblance functions*) são definidas para comparar os objetos dos esquemas. Estas funções utilizam os nomes e as estruturas dos objetos para identificar as semelhanças existentes entre eles.

Cada função de semelhança possui um peso que indica a importância que o usuário dá a ela. Por exemplo, se a similaridade entre os atributos é o critério mais importante, o peso associado a esta função é mais alto que os demais.

As principais contribuições de DeSOUZA (1986) foram:

- O uso de propriedades múltiplas dos objetos dos bancos de dados na análise dos objetos similares em um esquema.
- A associação de um peso a cada uma dessas propriedades.

A desvantagem que podemos demonstrar é que esta metodologia é específica para esquemas ACS e que este estudo não trata do passo de geração do esquema integrado.

A pesquisa de SHETH et al. (1988) apresenta uma ferramenta que conduz os usuários e os desenvolvedores através de um processo de integração de esquemas em cinco passos: Conjunto de Informação dos Esquemas; Equivalência de Classes de Criação e de Deleção (entidades e categorias); Equivalência de Classes de Criação e de Deleção (relacionamentos); Declarações dos Usuários (entidades e categorias); Declarações dos Usuários (relacionamentos).

Esta ferramenta requer uma grande interação com os usuários e desenvolvedores, e os mesmos podem especificar somente declarações de equivalência entre atributos, limitando o montante de informações semânticas que podem ser capturadas.

Esta deficiência foi endereçada na ferramenta BERDI (SHETH e MARCUS, 1992) que permite que os usuários definam relacionamentos entre os objetos que pertencem a um conjunto potencial de entidades relacionadas, chamadas grupos de entidades (*entity clusters*).

Existem várias outras ferramentas que automatizam, em algum passo, o processo de integração. Podemos visualizá-las através do quadro 1, que mostra, de forma resumida, em que fase as ferramentas automatizam o processo, o grau de automatização desta fase, o modelo de dados utilizado e as principais características destas ferramentas.

QUADRO 1: Resumo de ferramentas automatizadas de integração de esquemas.

| **Metodologia** | **Nível de Abstração** | **Modelo de Dados / Nome** | **Identificação dos Relacionamentos entre os esquemas** | **Geração do Esquema Integrado** |
|---|---|---|---|---|
| SIS, (DeSOUZA, 1986) | Conceitual | Semântico / Esquema Conceitual Abstrato | Automatizada | Não aplicável |
| SHETH et al. (1988) | Conceitual | Semântico / Modelo Entidade-Categoria - Relacionamento | Assistência do sistema para entradas manuais | Automatizada |
| MUVIS (HAYNE e RAM, 1990) | Visão | Semântico / Modelo de Dados Semântico | Automatizada | Automatizada / baseada em NAVATHE; EL-MASRI e LARSON (1986) |
| KAUL; DROSTEN e NUEHOLD (1990) | Conceitual | Orientado a Objeto / VODAK | Manual | A integração é realizada através da definição de visões usando construtores de classes |
| AHMED et al. (1991) | Conceitual | Orientado a Objeto / HOSQL | Manual | A integração é realizada através da definição de visões em HOSQL |
| SHOVAL e ZOHN (1991) | Visão | Semântico / Modelo Relacionamento Binário | Parcialmente automatizado através de entrada manual | Automatizada / focada na resolução de conflitos nos esquemas |
| BERDI (SHETH e MARCUS, 1992) | Conceitual | Semântico / Modelo Entidade-Categoria - Relacionamento | Parcialmente automatizado através de entrada manual | Automatizada |
| RAMESH e RAM (1995) | Conceitual | Semântico / Modelo Semântico Unificado | Automatizada através de entradas do usuário | Automatizada / arquivo facilitada para interação humana |

# 5. AS METODOLOGIAS DE INTEGRAÇÃO

Existem duas propriedades primárias que distinguem as metodologias de integração: (1) o nível de abstração ao qual a integração se detém, o qual dita os tipos de heterogeneidade que a metodologia deve considerar, e (2) a semântica transportada pelos esquemas de entrada.

A riqueza semântica dos bancos de dados iniciais depende, parcialmente, do modelo de banco de dados em uso. Assim, as metodologias de integração de bancos de dados são classificadas, dividindo-se aquelas que são baseadas no nível de abstração e outras que operam a partir do modelo de dados usado para representar os esquemas iniciais.

Uma terceira classificação está baseada no nível em que a metodologia pode trabalhar com as mudanças ocorridas nos sistemas bases da integração. Porém, poderemos ver de acordo com as colocações deste capítulo, que este tipo de divisão é paralela a classificação baseada no nível de abstração.

## 5.1. CLASSIFICAÇÃO BASEADA NOS NÍVEIS DE ABSTRAÇÃO

As metodologias de integração baseadas nos níveis de abstração podem ser divididas como operando em três níveis: (1) visão do usuário, (2) esquema conceitual, ou (3) nível dos dados.

Iremos começar nossa explanação com o nível mais comum – visão do usuário, onde a maioria das metodologias de integração podem ser classificadas.

O objetivo das metodologias desta categoria é integrar vários esquemas de usuários (representando visões dos usuários de um banco de dados) em um único esquema (SHETH e LARSON, 1990).

A maioria das visões de usuários é representada utilizando um modelo de dados comum. Para chegar a este resultado, estas metodologias necessitam de um passo de tradução de esquemas. Porém, como as visões não representam bancos de dados físicos, a maior parte do conhecimento semântico é fornecido pelo esquema do banco de dados.

Podemos considerar um outro caso de integração de visões, onde as mesmas não são visões estáticas, pois, no caso de multi banco de dados, o esquema integrado pode ser utilizado como ponto inicial para a criação de um novo banco de dados.

As metodologias reportadas em BATINI; LENZERINI e NAVATHE (1986), NAVATHE e GADGIL (1982), BATINI e LENZERINI (1984), BISKUP e CONVENT (1986) são exemplos de metodologias de integração de visão.

As metodologias que operam no nível dos esquemas conceituais geram um ou mais esquemas integrados a partir dos esquemas locais dos bancos de dados a serem integrados.

Para que as metodologias classificadas nesta categoria consigam cumprir seu objetivo, é necessário que elas possam interagir tanto com os problemas estruturais quanto com os de heterogeneidade semântica, e podem ser classificadas em:

- Metodologias de reestruturação de esquemas: são aquelas que geram um esquema integrado, aplicando operadores de reestruturação de esquemas, nos bancos de dados locais.
- Metodologias de geração de visões: são aquelas que geram uma representação integrada, desenvolvendo visões ou definindo consultas que são importantes no processo de integração, nos bancos de dados locais.

Exemplos de metodologias que seguem a linha de reestruturação de esquemas podem ser encontrados em EL-MASRI; LARSON e NAVATHE (1986), LARSON; NAVATHE e EL-MASRI (1989), e SPACCAPIETRA; PARENT e DUPONT (1992).

A principal diferença entre as metodologias que utilizam a estratégia de reestruturação de esquemas e as que usam integração de visões é o fato de que na reestruturação de esquemas, os esquemas a serem integrados são derivados de modelos de bancos de dados heterogêneos que representam um ou mais sistemas.

Exemplos de pesquisas que utilizaram metodologias de integração de visões podem ser encontrados em KAUL; DROSTEN e NUEHOLD (1990), AHMED et al. (1991), BERTINO (1991), e em KIM e SEO (1991).

Existem, também, metodologias um pouco mais antigas como as de MOTRO e BUNEMAM (1981), CASANOVA e VIDAL (1983), MANNINO e EFFELSBERG (1984), e TEMPLETON et al. (1987) que, apesar de adotarem o paradigma da geração de visões, utilizam um processo de geração de visão integrada muito parecido com o paradigma da reestruturação de esquemas.

A principal diferença entre a reestruturação de esquemas e as pesquisas que envolvem a geração de visões recai sobre a natureza estática da reestruturação de esquemas e a natureza dinâmica da geração de visões.

Uma geração de esquema integrado que utiliza a reestruturação de esquemas é uma representação que reflete a definição dos esquemas no momento em que o processo de integração é realizado. Qualquer mudança nos bancos de dados que originaram a integração, que afete o esquema integrado, demandará a reelaboração do processo de integração.

No método de integração de visões, a representação da integração é gerada através de uma definição de visões a partir de esquemas locais. Como resultado, se o esquema muda, novas visões devem ser definidas e, este caso só ocorre quando a mudança afeta as visões existentes no banco de dados integrado.

A terceira categoria das metodologias de integração trabalha com os dados dos bancos de dados. Estas metodologias baseiam-se nos valores atuais dos dados para realizar a integração.

A maioria dos trabalhos desta categoria têm seu foco na integração de banco de dados relacional. Estratégias de integração de instâncias dos dados são apresentadas em DeMICHIEL (1989), CHATTERJEE e SEGEV (1991) e PRABHAKAR et al. (1993).

As metodologias fundamentadas nos dados dos bancos de dados possuem dois problemas principais:

- Identificação das entidades: como uma entidade que representa o mesmo objeto no mundo real é representada nos diferentes bancos de dados.
- Conflitos de valores de atributos: como serão consideradas as diferenças entre valores de atributos que representam o mesmo objeto no mundo real.

Estas diferenças podem aparecer tanto em divergências de domínio de atributos quanto na diversidade de valores armazenados nos campos dos bancos de dados.

A integração de instâncias tem como meta resolver incompatibilidades dos dados de tuplas que não possuem chaves, o que dificulta a geração de união das tuplas. Podemos evidenciar, com isto, que mudanças nos valores dos dados podem anular qualquer integração realizada. Temos como exemplo a metodologia SIM (*Schema Integration Methodology*) de FANKAUSER; MOTZ e HUCK (1995).

Outra técnica que pode ser citada na categoria de integração de dados opera com as regras de integridade semântica e seu uso na integração de esquemas. RAMESH e RAM (1995) apresentam uma metodologia que descreve como as regras de integridade de múltiplos bancos de dados podem ser combinadas, para desenvolver regras no esquema global ou federado, e o uso destas regras de integridade, no processamento de consultas semânticas.

## 5.2. CLASSIFICAÇÃO BASEADA NO MODELO DE DADOS DOS ESQUEMAS INICIAIS

As estratégias de integração de esquemas são altamente dependentes da semântica fornecida pelos bancos de dados locais. Como este fato está intimamente ligado ao tipo de modelo utilizado, podemos classificar as metodologias de acordo com os modelos de dados usados para representar os modelos locais. Quatro modelos de dados têm sido empregados: modelo relacional, modelo semântico, modelo orientado a objeto e modelo baseado na lógica.

As primeiras metodologias de integração de esquemas que usaram o modelo relacional para representar seus esquemas locais foram as de AL-FEDAGHI e SCHEURMANN (1981) e CASANOVA e VIDAL (1983).

O problema da utilização do modelo relacional é o limitado poder de expressividade do modelo, o que resulta em capturas inadequadas da semântica dos esquemas. Porém, a existência de grande quantidade de bancos de dados relacionais, a simplicidade do modelo, assim como a existência de uma linguagem de consulta de alto poder, fazem com que as pesquisas nesta área sejam de valor expressivo.

Por conseqüência, o modelo relacional é escolhido por pesquisadores que pretendem desenvolver protótipos de sistemas de bancos de dados heterogêneos. Por exemplo, o protótipo e a metodologia descritos em DEEN; AMIM e TAYLOR, (1987), TEMPLETON et al. (1987), CHUNG (1990), KIM e SEO (1991) utilizam o modelo relacional.

O modelo e o banco de dados relacional são também a escolha de pesquisadores interessados em resolver problemas de integração baseados nos dados dos bancos de dados (DeMICHIEL, 1989; CHATTERJEE e SEGEV, 1991; PRABHAKAR ET AL., 1993).

As pesquisas baseadas no modelo semântico utilizam variantes do modelo entidade relacionamento para representar os esquemas locais, assim como para representar os esquemas integrados. BATINI e LENZERINI (1984), LARSON; NAVATHE e EL-MASRI (1989), SHOVAL e ZOHN (1991), SPACCAPIETRA; PARENT e DUPONT (1992) e SHETH; GALA e NAVATHE (1993) são exemplos de metodologias que pertencem a esta categoria.

A principal razão para usar o modelo de dados semântico é que ele pode expressar uma riqueza semântica maior que o modelo relacional, a qual pode ser bastante explorada durante a integração dos esquemas.

Como o modelo semântico é o mais usado para representar visões e esquemas conceituais, a maioria das metodologias deste grupo se encaixa na categoria de integração de esquemas conceituais ou visões, da classificação que citamos na seção anterior.

As pesquisas baseadas no modelo orientado a objeto são categorizadas à parte porque algumas metodologias desta categoria integram métodos quando estão integrando os esquemas.

Estas metodologias também tratam da integração de atributos complexos e hierarquia de objetos.

A maioria das metodologias baseadas no modelo orientado a objeto podem ser identificadas na categoria de integração de visões ou de esquemas conceituais da classificação que citamos anteriormente. Exemplos de pesquisas que fazem parte desta categoria são as de KAUL; DROSTEN e NUEHOLD (1990), BERTINO (1991), CZEJDO e TAYLOR (1991), GELLER et al. (1992) e, THIEME e SIEBES (1993).

As pesquisas que utilizam modelos baseados em lógica pertencem à última categoria e começaram a aparecer há pouco tempo na literatura.

Estas pesquisas representam um passo natural no desenvolvimento de metodologias de integração porque uma ordem lógica deve ser mostrada para que seja possível a representação da semântica dos bancos de dados relacionais de uma maneira formal.

Usando metodologias baseadas em lógica também é possível capturar mais semântica do que quando utilizamos o modelo semântico. Por exemplo, os modelos baseados em lógica nos permitem expressar as regras de integridade semântica, que são regras de integridade definidas pelo usuário e são consideradas importantes no desenvolvimento das consultas do esquema integrado (SHEKHAR et al., 1993).

RAMESH e RAM (1997) descrevem como as regras de integridade semântica podem ser utilizadas para facilitar a integração de esquemas.

WHANG; NAVATHE e CHAKRAVARTHY (1991) demonstraram que é mais fácil traduzir esquemas relacionais para esquemas baseados na lógica do que para modelos semânticos.

Podemos visualizar, através do quadro 2, algumas das principais metodologias de integração.

Cada linha do quadro 2 identifica uma metodologia, a coluna final indica o tipo do modelo utilizado para representar o processo de saída da integração e se essa saída é uma integração de esquemas ou de visões.

QUADRO 2: Principais metodologias de integração de bancos de dados. (continua)

| **Metodologia** | **Nível de Abstração** | **Modelo de Dados** | **Modelo de Dados de Saída / Tipo de Representação** |
|---|---|---|---|
| BATINI e LENZERINI (1984) | Conceitual | Semântico | Semântico / Esquema |
| TEMPLETON et al. (1987) | Conceitual | Relacional | Relacional / Visão |
| LARSON; NAVATHE e EL-MASRI (1989) | Conceitual | Semântico | Semântico / Esquema |
| DeMICHIEL (1989) | Dados | Relacional | ------ |

QUADRO 2: (continuação)

| Metodologia | Nível de Abstração | Modelo de Dados | Modelo de Dados de Saída / Tipo de Representação |
|---|---|---|---|
| KAUL; DROSTEN e NUEHOLD (1990) | Conceitual | Orientado a Objeto | Orientado a Objeto / Visão |
| BERTINO (1991) | Conceitual | Orientado a Objeto | Orientado a Objeto / Visão |
| KIM e SEO (1991) | Conceitual | Orientado a Objeto / Relacional | Orientado a Objeto / Visão |
| WHANG; NAVATHE e CHAKRAVARTHY (1991) | Conceitual | Lógico | Lógico / Visão |
| SHOVAL e ZOHN (1991) | Visão | Semântico | Semântico / Visão |
| AHMED et al.(1991) | Conceitual | Orientado a Objeto | Orientado a Objeto / Visão |
| CHATTERJEE e SEGEV (1991) | Dados | Relacional | ------ |
| SPACCAPIETRA; PARENT e DUPONT (1992) | Conceitual | Semântico | Semântico / Esquema |
| GOTTHARD; LOCKEMANN e NEUFELD (1992) | Visão | Orientado a Objeto | Orientado a Objeto / Visão |
| PRABHAKAR et al. (1993) | Dados | Relacional | ------ |
| JOHANNESSON (1993) | Conceitual | Lógico | Lógico / Esquema |
| RAMESH e RAM (1995); RAMESH e RAM (1997) | Conceitual / Dados | Semântico / Lógico | Semântico / Esquema |

As atuais metodologias de integração não trabalham com a heterogeneidade dos dados na fase de integração, visto que elas assumem que antes do processo de integração ser aplicado, os esquemas são traduzidos em esquemas equivalentes baseados em um único modelo de dados.

## 6. AS METODOLOGIAS UTILIZADAS

Utilizamos, neste trabalho, como metodologia principal, o conjunto de técnicas e processos descritos em SPACCAPIETRA; PARENT e DUPONT (1992). Como nem todos os casos de nossa aplicação puderem ser atendidos por esta metodologia utilizamos também, como apoio, a metodologia descrita em BATINI e LENZERINI (1984).

### 6.1. METODOLOGIA DESCRITA EM SPACCAPIETRA; PARENT e DUPONT (1992)

Iremos descrever, primeiramente, a metodologia de SPACCAPIETRA; PARENT e DUPONT (1992) que define o processo de integração de banco de dados como sendo um mecanismo que deriva um novo esquema de banco de dados distribuído a partir de especificações existentes.

Esta metodologia é implementada através de um processo bifásico conforme a figura 13 abaixo:

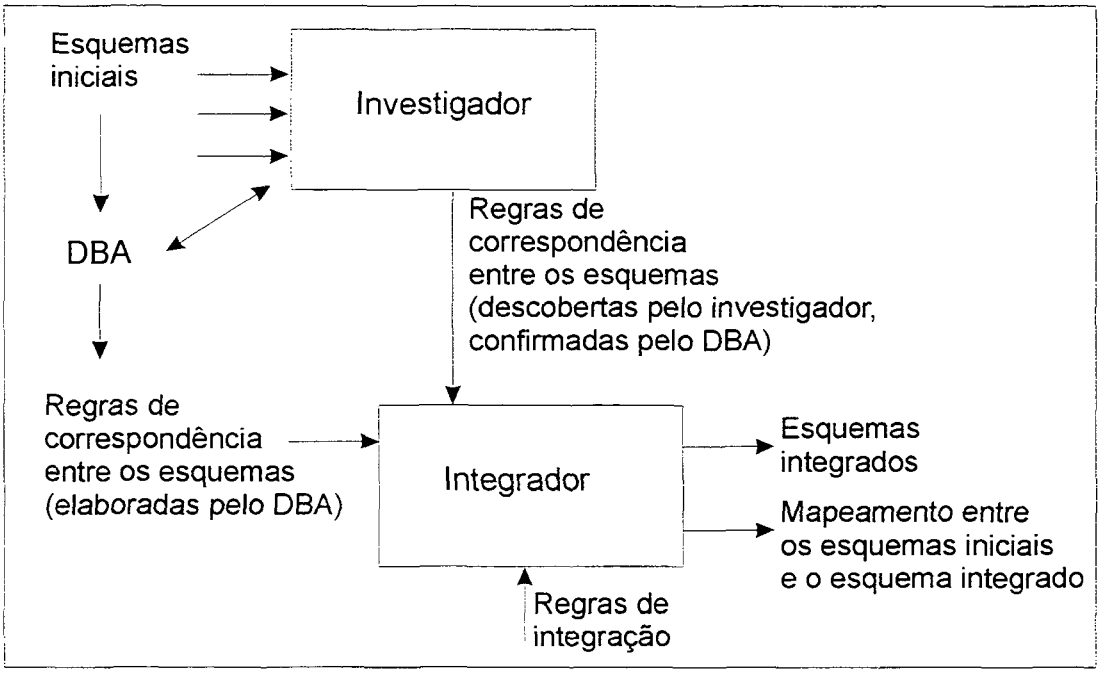


FIGURA 13: Processo bifásico de integração de esquemas de SPACCAPIETRA; PARENT e DUPONT (1992).

Primeiramente, devem ser identificados os pontos em comum e os pontos discrepantes de cada esquema. Podemos chamar esta fase de investigação. Essa

fase normalmente é manual, onde o DBA analisa os esquemas iniciais e define o conjunto das correspondências entre os esquemas.

Depois desta fase, a integração é realizada. O esquema integrado é construído de uma maneira semi automática, de acordo com as correspondências entres os esquemas e das regras de integração.

Interações entre o integrador e o DBA são importantes para resolver conflitos nos esquemas de entrada, cada vez que o integrador não possuir o conhecimento para realizar tal tarefa.

Os conflitos ocorrem quando os conceitos são modelados com representações diferentes. Quanto mais conflitos forem resolvidos pelo integrador maior a sua potência.

Esta metodologia assume, como ponto inicial, que as correspondências estão definidas. As principais características desta metodologia são:

- A metodologia estende o escopo da integração automática através dos seguintes passos: (1) resolvendo novos casos de conflito, como a integração de tipos de objeto e de atributos; (2) integrando não somente elementos (tipos de objetos e atributos) assim como as ligações existentes entre estes elementos. Isto é realizado através de regras apropriadas.
- Realiza a integração sem que os esquemas iniciais sejam modificados. Isto também se aplica no mapeamento das funcionalidades.
- Suporta a heterogeneidade de modelos de dados.
- Pode ser aplicada tanto na integração de banco de dados como na integração de visões.

### 6.1.1. Uma descrição genérica das correspondências entre os esquemas

Basicamente, esta metodologia de integração de esquemas está concentrada na idéia de repasse do conhecimento externo ao integrador, para o seu interior.

Os integradores existentes não sabem como mapear os esquemas, se existirem conflitos estruturais. Sendo assim, recorrem ao DBA para que este deixe os esquemas iniciais em conformidade para este item.

Assumindo que o conhecimento sobre a transformação dos esquemas será inserido no integrador, faremos com que ele possa deixar os esquemas em conformidade, resolvendo os problemas estruturais.

Similarmente, os integradores existentes não possuem conhecimento para diferenciar os modelos de dados. Assim, torna-se necessário transformar todos os esquemas iniciais no modelo de dados utilizado pelo integrador.

Aplicando o conhecimento sobre os modelos de dados, no integrador, permitimos que ele gerencie as correspondências entre as construções dos diferentes modelos de dados.

Nesta metodologia, a descrição das características semelhantes entre os esquemas, e as regras de integração, estão definidas em termos de pequenos conceitos genéricos, abstraindo-se de qualquer modelo de dados. Além de abranger esta estrutura, o integrador possui o conhecimento para solucionar conflitos específicos de determinados modelos de dados.

Assim como o integrador possui o conhecimento para manejar aspectos comuns de esquemas, a partir de diferentes modelos de dados, ele também é capaz de mostrar o esquema integrado resultante, nos diferentes modelos que suporta.

Sendo assim, as regras de integração foram definidas para manejar as diferenças existentes entre os diversos conceitos e comportamentos estruturais dos modelos de dados, o que suporta a resolução dos conflitos estruturais.

A estrutura específica desta metodologia está ilustrada na figura 14 abaixo:

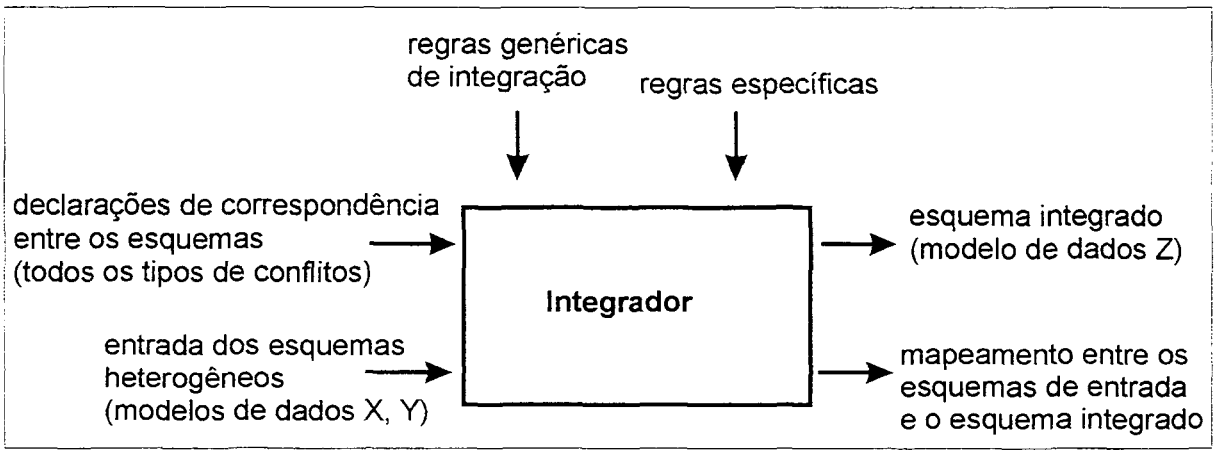


FIGURA 14: Estrutura proposta pela metodologia de SPACCAPIETRA; PARENT e DUPONT (1992).

Esta metodologia considera, como premissa, que as partes comuns dos esquemas foram identificadas pelo DBA e pelos usuários dos sistemas. Estas identificações das partes comuns são definidas utilizando as declarações de correspondência entre os esquemas.

Uma declaração de correspondência entre os esquemas é uma afirmação de que um objeto, em um esquema, está relacionado a outro objeto em outro esquema. As declarações identificadas são empregadas e, no caso de existirem conflitos semânticos, descritivos ou estruturais as suas devidas correspondências são também aplicadas.

O integrador recebe como entrada dois (ou mais) esquemas e as declarações existentes entre eles. O conjunto de declarações é lido e ordenado para o processamento. Cada declaração é considerada e, então, a regra de integração apropriada é aplicada, considerando-se o modelo de dados dos esquemas iniciais.

As regras de integração definem o que deve ser realizado no esquema integrado e como deve ser o mapeamento entre os esquemas.

### 6.1.2. O modelo genérico de dados

Esta metodologia utiliza uma ferramenta para definir as declarações genéricas e as regras de integração. O modelo genérico de dados - GDM (*Generic Data Model*) é um conjunto de conceitos de modelagem que permite identificar os conflitos entre os esquemas.

Com a utilização do GDM, pode-se modelar objetos com uma estrutura complexa de dados, possibilitando a inclusão de outros objetos como seus componentes. Isto simplifica a aplicação das regras de integração, assim como a resolução dos conflitos e a identificação de componentes, em um esquema, que podem ser considerados como objetos em outro esquema.

O GDM possui três conceitos de modelagem: objetos, valores dos atributos e atributos referência.

Um objeto GDM é a identificação de um objeto complementada com a estrutura dos dados existentes em uma tupla de atributos. Para cada atributo existe uma cardinalidade mínima e máxima que define o número de vezes que este atributo pode aparecer: zero, uma ou mais vezes.

Um atributo pode ser atômico ou complexo. Um atributo complexo é uma tupla de atributos. Atributos atômicos são valores de atributos (existe um domínio associado a ele, como inteiro, caracter ou data) ou atributos referência (quando o seu domínio é um tipo de objeto ou quando faz referência a um tipo de objeto).

Um exemplo de atributo referência pode ser visualizado na figura 15, abaixo, onde o tipo atributo depto é um atributo referência do objeto Depto.

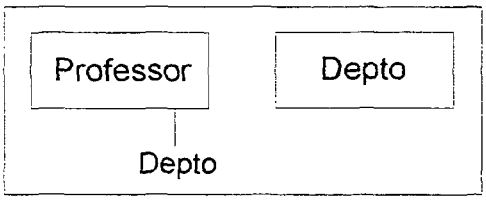


FIGURA 15: Exemplo de um atributo referência

Os atributos referência são considerados como bidirecionais. Como conseqüência, é utilizado o termo elemento para referir-se tanto a objetos como a atributos.

Existe também um conceito adicional no GDM: o *link*. O termo *link* denota qualquer direção de conexão existente entre dois elementos: *link* de um elemento de um atributo e um de seus valores ou atributos complexos, *link* de uma referência de um elemento e a referência de um atributo.

Os dois tipos de *links* que são considerados nesta metodologia estão descritos abaixo.

**Definição de *Link* (D1)**

Consideram-se X e Y como sendo tipos de objetos, valores ou atributos complexos, sendo que X — Y (ou Y — X) é um *link* se:

- Y for um valor ou um atributo complexo de X; assim pode-se definir X — Y como um ***link* de atributo**,
- X possuir um atributo referência, chamado r, apontando para o tipo de objeto Y; assim podemos definir $X \xrightarrow{r} Y$ como um ***link* de referência**. Se não houver ambigüidade (quando houver somente uma referência entre X e Y ) $X \xrightarrow{r} Y$ pode ser notado simplesmente como X — Y.

As cardinalidades dos *links* são utilizadas no processo de integração. As cardinalidades mínimas e máximas de X no *link* X — Y são os números mínimos e máximos de $y \in Y$ que podem ser denominados a $x \in X$ através do *link* X — Y.

O inverso ocorre para cardinalidades de Y no *link* X — Y. As cardinalidades X — Y são denotadas como: min(X):max(X), min(Y):max(Y), como é demonstrado a seguir:

min(X):max(X) =

cardinalidade mínima: máxima do atributo Y, se X — Y

*do atributo r, se* $X \overset{r}{\text{—}} Y$

min(Y):max(Y) =

1:n se X — Y (onde n=1 se Y for um identificador de X)

0:n se $X \overset{r}{\text{—}} Y$

Nos parágrafos seguintes podemos verificar alguns exemplos, assumindo que todos os atributos são obrigatórios e "não nulos".

Temos os seguintes esquemas relacionais, de acordo com a figura 7, da página 27:

S1: Usuário $\overset{\text{departamento}}{\text{—}}$ Departamento é um *link* de referência, com cardinalidades 1:1, 0:n. Indica que um usuário pertence a somente um departamento e que um departamento possui zero ou muitos usuários.

S2: Usuário – matrícula é um *link* de atributo.

Dois elementos em um esquema podem ser limitados diretamente por um *link*, ou, indiretamente, por uma composição de *links*, chamados caminhos. Por exemplo, ProgPos e Orientador em S3 (figura 12 da página 35) são limitados através do caminho ProgPos – Curso – Orientador.

**Definição de Caminho (D2)**

Consideram-se $X_1$, $X_2$, ..., $X_n$ como sendo elementos (tipos de objetos ou valores de atributos) em um esquema tal que $\forall\, i \in \{1, 2, ..., n-1\}$, $X_1$ está unido a $X_{i+1}$ por um *link* de atributo ou por um *link* de referência, onde $X_1$ — $X_2$ — ... — $X_n$ é um caminho.

A cardinalidade do caminho $X_1$ — $X_2$ — ... — $X_n$ é igual ao produto (**X**) das cardinalidades correspondentes nos componentes dos *links*:

Cardinalidade mínima de $X_1$ = $\mathbf{X}_{\,i \in [1:\, n-1]}$ cardinalidade mínima de $X_i$ em $X_i$ — $X_{i+1}$

Cardinalidade máxima de $X_1$ = $\mathbf{X}_{\,i \in [1:\, n-1]}$ cardinalidade máxima de $X_i$ em $X_i$ — $X_{i+1}$

Cardinalidade mínima de $X_n$ = $\mathbf{X}_{\,i \in [2:n]}$ cardinalidade mínima de $X_i$ em $X_{i-1}$ — $X_i$

Cardinalidade máxima de $X_n$ = $\mathbf{X}_{\,i \in [2:n]}$ cardinalidade máxima de $X_i$ em $X_{i-1}$ — $X_i$

Exemplo para S3 (figura 12 da página 35):

tese — ProgPos — Curso — Orientador — area é um caminho que associa a tese de um programa de pós de um curso a uma área de aplicação que um orientador possui. As cardinalidades deste caminho são 1:n, 0:n, porque uma tese pode estar relacionada a uma ou muitas áreas de pesquisa e uma área de pesquisa pode possuir zero ou muitas teses.

### 6.1.3. Estado real

Como já foi comentado anteriormente, a semântica das regras de correspondência é definida através de uma referência do elemento com o seu significado no mundo real.

Em LARSON; NAVATHE e EL-MASRI (1989) foi introduzido o conceito de "estado real no mundo" (*real world state*) de um objeto O, RWS(O), como sendo o conjunto das instâncias do objeto O em um dado momento.

Nesta metodologia este conceito foi estendido para os atributos, *links* e caminhos para que fosse possível trabalhar com cada conceito, indiferentemente do modelo de dados utilizado.

O conceito de RWS irá permitir a definição do significado das regras de correspondência relacionadas aos elementos de tipos diferentes (tipos de objetos e atributos), aos caminhos e aos *links*.

O RWS do valor de um atributo A, atômico ou complexo, pode ser definido da mesma forma que o RWS de um tipo de objeto, como sendo o conjunto dos objetos no mundo real que os valores de A representam.

No caso de atributos multi valorados, o conjunto de seus elementos é formado por valores simples, não por um conjunto de valores. Por exemplo, se "area" for um atributo multivalorado da entidade Orientador a qual contém dois orientadores, um com duas áreas de pesquisa: "engenharia de *software*" e "banco de dados" e outro com uma área de pesquisa: "inteligência artificial". O RWS será: RWS(area) = {engenharia de *software*, banco de dados, inteligência artificial}.

**Real estado de um elemento (tipo de objeto, atributo simples ou complexo) (definição D3)**

O RWS de um tipo de objeto O (ou de um atributo simples ou complexo A) é o conjunto de objetos no mundo real que o conjunto de ocorrências de O representa (assim como os valores de A).

Existe um mapeamento, um para um, entre o RWS de um conjunto de ocorrências ou valores dos tipos de objetos ou atributos. Um atributo pode ter o mesmo valor em diferentes objetos do banco de dados. Porém, quando se verifica o conjunto dos valores de um atributo, abstraindo-se dos valores duplicados, cada valor descreve somente um objeto no mundo real.

Nas definições anteriores de RWS não são tratados os atributos referência. Os atributos referência não suportam valores, assim como não podem ser visualizados através de correspondências com os objetos, mas sim com os *links* ou caminhos. Por esta razão, esta metodologia não se interessa por seu RWS como elemento, mas sim pelo RWS do *link* que expressam.

Um *link* X — Y, ou um caminho X — ... — Y, é uma conexão entre dois tipos de objetos X e Y. Seu RWS é composto por pares de objetos reais, um descrito por X e outro por Y, sendo que estes dois objetos no mundo real estão limitados por uma associação com o *link* ou com o caminho que representam.

**Real estado de um caminho (definição D4)**

O real estado no mundo de um caminho $X_1$- $X_2$-...-$X_n$, RWS($X_1$- $X_2$-...-$X_n$), é o conjunto de pares de objetos <$o_1$, $o_n$>, tal que $o_1 \in RWS(X_1)$ e $o_n \in RWS(X_n)$ e existem os objetos $o_2$, $o_3$,...,$o_{n-1}$ tal que $\forall$ i $\in$ {1,2,...,n-1}, $o_1 \in RWS(X_1)$, com $o_1$ e $o_{1+1}$ unidos pela real associação representada pelo *link* $X_i$ — $X_{i+1}$.

Exemplo (referindo-se a S3, na página 35):

RWS(tese — ProgPos — Curso — Orientador — area) é o conjunto de pares do tipo <tese, area>, associando-se, para cada Programa de Pós no RWS(ProgPos), cada tese à área de seu Orientador.

### 6.1.4. Regras de correspondências

Existem dois tipos de regras de correspondências: aquelas relativas a dois elementos e aquelas relativas a dois caminhos ou *links*.

As regras de correspondências entre os elementos identificam os conflitos semânticos, descritivos e estruturais existentes entre dois elementos:

- Se um dos elementos é um tipo objeto e o outro é um atributo, este caso representa um conflito estrutural.
- Dependendo do conjunto de relacionamentos que relacionam os RWS dos elementos, não existe um conflito semântico entre eles (≡), ou existe ($\supseteq$, $\cap$, $\neq$).
- Uma cláusula adicional, nesta regra, especifica se, e como, os atributos de dois elementos estão relacionados entre si. Um conflito descritivo acontece se existir pelo menos um atributo em um elemento que não possua correspondência de atributo com o outro elemento, ou se existir pelo menos um par de atributos relacionados porém não iguais.

#### 6.1.4.1. Regras de correspondências entre elementos

Primeiramente, iremos ilustrar quatro possíveis conjuntos de relacionamentos entre os RWS de elementos correspondentes. A definição formal será explanada de acordo com os exemplos.

Vamos considerar uma universidade que possui muitos departamentos, com bancos de dados diferentes, os quais devem ser integrados. Alguns dos bancos de dados locais armazenam o catálogo dos cursos oferecidos, descrevendo todos os cursos oferecidos pela universidade. O catálogo possui o mesmo formato em todos os departamentos: uma regra de equivalência irá relacionar todos os cursos em um mesmo conjunto.

Vamos supor que cada banco de dados local mantém os funcionários de seu departamento, utilizando o mesmo formato, e que cada funcionário trabalha, somente, em um departamento. Os tipos objeto Funcionário serão declarados como correspondentes porém desconexos (*disjoint*).

Suporemos, agora, que cada departamento mantém os registros de seus alunos. Diferentes departamentos podem possuir os mesmos alunos: os tipos objeto Aluno estarão relacionados por uma interseção de correspondências (*intersection*).

Finalmente, supomos que cada departamento tem seus fornecedores, porém, os mesmos devem ser escolhidos de acordo com um banco de dados geral mantido

pelo departamento central. Os tipos objeto Fornecedor Local se inter-relacionarão, mas serão declarados como uma correspondência de inclusão visto que somente o objeto Fornecedor do banco de dados do departamento central fará a inclusão dos dados.

**Regras de correspondência entre elementos (definição D5)**

Consideram-se $X_1$, $X_2$, como sendo dois elementos (tipos de objetos, tipos de atributos *simples ou complexos*), $X_1$ do esquema $S_1$ e $X_2$ do esquema $S_2$. Uma correspondência entre $X_1$ e $X_2$ pode ser declarada das seguintes maneiras:

- $X_1$ e $X_2$ são equivalentes, representados como: $X_1 \equiv X_2$
  onde seus estados, em qualquer momento, são $RWS(X_1) = RWS(X_2)$;
- $X_1$ contém $X_2$, representados como: $X_1 \supseteq X_2$
  onde seus estados, em qualquer momento, são $RWS(X_1) \supseteq RWS(X_2)$;
- Existe uma interseção entre $X_1$ e $X_2$, representados como: $X_1 \cap X_2$
  onde seus estados, em qualquer momento, são $RWS(X_1) \cap RWS(X_2) \neq \varnothing$;
- $X_1$ e $X_2$ são disjuntos, representados como: $X_1 \neq X_2$
  onde seus estados, em qualquer momento, são $RWS(X_1) \cap RWS(X_2) \neq \varnothing$;

Esta última regra diz que, mesmo que os elementos sejam disjuntos, as suas semânticas estão relacionadas e o DBA precisa uni-los, em um elemento mais genérico, no esquema integrado.

### 6.1.4.2. Regras de correspondência entre atributos

Sempre que dois elementos são declarados como correspondentes, regras complementares sobre a correspondência entre os atributos são necessárias para direcionar o integrador a produzir uma integração de estruturas, ou seja, a validar quais atributos são correspondentes.

No modelo desta metodologia, essas regras de correspondências de atributos de dois elementos correlatos, X e Y, são determinadas através dos atributos referência como declarações de caminho.

Os valores de atributos atômicos e complexos são declarados como parte de uma regra de correspondência entre X e Y, usando a cláusula "com correspondência de atributos". Esta cláusula define o conflito de descrição.

Similarmente à correspondência entre elementos, o conjunto de relacionamentos entre o conjunto de valores de dois atributos equivale a uma das *seguintes regras*:

- = os atributos possuem o mesmo valor.
- $\supseteq$ o(s) valor(es) de um atributo incluem o(s) valor(es) de outro atributo. Se ambos os atributos são mono-valorados, então os dois possuem o mesmo valor, em outro caso o atributo a ser incluído possui um valor nulo.
- $\cap$ os dois atributos são multi-valorados e o conjunto de seus valores se interseccionam.
- $\neq$ os valores dos atributos são sempre diferentes, mas eles estão relacionados. O DBA deve mesclar os dois em um atributo mais extenso, unindo os dois atributos.

A cláusula "com correspondência de atributos" define, para cada correspondência de atributo, qual é o conjunto de relacionamentos, e, caso exista, a função de mapeamento de domínios. Diferentes tipos de funções podem estar envolvidas:

- O mapeamento 1:1 define uma tradução dos domínios. Uma tabela de cruzamento de resultados deve ser utilizada neste caso.
  Por exemplo, valores em Dólar devem ser convertidos para valores em Real.
- Uma função de agregação, definindo o valor de um atributo mono-valorado, como sendo o resultado da agregação de um conjunto de valores de um atributo multivalorado.
  Por exemplo, um atributo número de filhos é igual ao somatório do atributo filhos de um outro banco de dados.
- Uma função de tupla definindo o valor de um atributo como sendo o resultado de um produto cartesiano de muitos atributos.
  Por exemplo, um atributo endereço é igual ao produto cartesiano dos atributos número, rua e cidade de um outro banco de dados.

Existem muitas outras considerações que podem ser exploradas, conforme podemos verificar em LARSON; NAVATHE e EL-MASRI (1989), porém não iremos citá-las aqui pois as mesmas não alteram a natureza do problema.

Como o objetivo desta metodologia não é analisar as muitas facetas de problemas existentes na integração de banco de dados, a definição D6, a seguir, trata somente do conjunto de relacionamentos existentes entre o conjunto de valores de dois atributos.

Cada $A_{1i}$ na definição D6 pode ser substituído pela função $f(A_{1i})$ que é uma tradução dos domínios ou das funções de agregação, ou por $f(A_{1i1}, A_{1i2}, ..., A_{1ip})$, que é uma função de tupla, ou por qualquer composição destas funções ou similaridades com $A_{2i}$.

**Regras de correspondência de valores entre atributos (definição D6)**

Considera-se $X_1$ <cor> $X_2$ como sendo uma declaração de correspondência entre elementos, onde <cor> ::= $= | \supseteq | \cap | \neq$.

Consideram-se $A_{11}, A_{12}, ..., A_{1n}$ como sendo os valores dos atributos de $X_1$, e $A_{21}, A_{22}, ..., A_{2n}$ como sendo os valores dos atributos de $X_2$. Se $X_1$ ou $X_2$ forem atributos atômicos, está implícito que eles possuirão somente um componente.

É chamado de "o" qualquer elemento comum ao estado real de $X_1$ ou $X_2$:

$o \in RWS(X_1) \cap RWS(X_2)$; e $e_1$, $e_2$ as ocorrências que representam "o" no banco de dados descrito por $S_1$, $S_2$.

Se $X_1$ ou $X_2$ são disjuntos, considera-se "o" como sendo um elemento hipotético contradizendo a disjunção, quer dizer, se existir este "o", teremos: $o \in RWS(X_1) \cap RWS(X_2)$

Então: $X_1$ <cor> $X_2$ com os atributos correspondentes:

$attcor_1(A_{11}, A_{21}), attcor_2(A_{12}, A_{22}), ..., attcor_i(A_{1n}, A_{2n})$,

é uma regra de correspondência que declara:

$X_1$ <cor> $X_2$ é verdadeiro, para cada $attcor_i(A_{1i}, A_{2i})$:

- Se $attcor_i(A_{1i}, A_{2i})$ corresponder a $A_{1i} = A_{2i}$
  então, para qualquer $o \in RWS(X_1) \cap RWS(X_2)$: $e_1.A_{1i} = e_2.A_{2i}$.
- Se $attcor_i(A_{1i}, A_{2i})$ corresponder a $A_{1i} \supseteq A_{2i}$
  então, para qualquer $o \in RWS(X_1) \cap RWS(X_2)$: $e_1.A_{1i} \supseteq e_2.A_{2i}$.
- Se $attcor_i(A_{1i}, A_{2i})$ corresponder a $A_{1i} \cap A_{2i}$
  então,
  é possível que para algum $o \in RWS(X_1) \cap RWS(X_2)$: $e_1.A_{1i} \cap e_2.A_{2i} \neq \varnothing$.
- Se $attcor_i(A_{1i}, A_{2i})$ corresponder a $A_{1i} \neq A_{2i}$

então, para qualquer $o \in RWS(X_1) \cap RWS(X_2)$: $e_1.A_{1i} \cap e_2.A_{2i} \neq \varnothing$, neste caso os dois atributos são semanticamente relacionados e o DBA precisa mesclá-los em um atributo mais extenso, unindo os dois.

As correspondências entre atributos não devem contradizer as regras de correspondências de seus elementos pais.

Cada regra de correspondência envolvendo um mapeamento de ocorrências de diferentes bancos de dados, ou seja, regras de equivalência entre elementos, inclusão e interseção, deve conter uma regra de igualdade de atributos 1 para 1, relacionando os identificadores: $A_{1i} = A_{2i}$ ou $A_{2i}$ = função bijuntiva ($A_{1i}$).

### 6.1.4.3. Regras de correspondências entre caminhos

Na análise dos relacionamentos entre os esquemas faz-se necessária a identificação das correspondências entre os caminhos.

Referindo-se aos esquemas S5 e S6, demonstrados na figura 16:

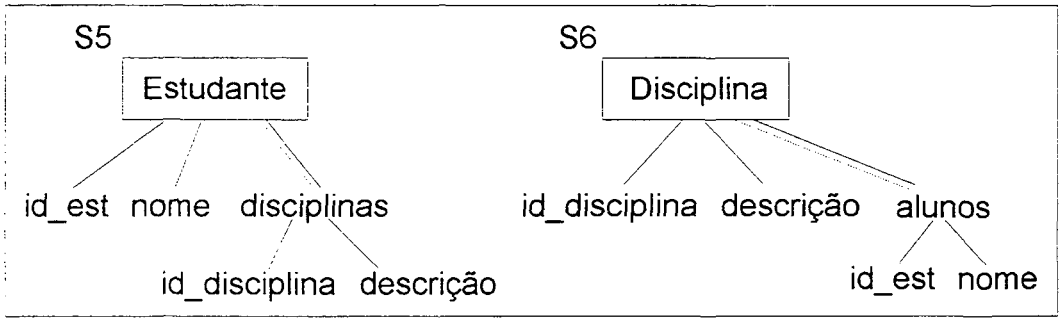


FIGURA 16: Diagrama dos esquemas S5 e S6.

Se supomos que os dois esquemas vêem exatamente os mesmos objetos (estudantes e disciplinas), as regras de correspondências entre os elementos de S5 e S6 são:

| | |
|---|---|
| Disciplina ≡ disciplinas | com a seguinte correspondência de atributos: id_disciplina = id_disciplina, descrição = descrição |
| alunos ≡ Estudante | com a seguinte correspondência de atributos: id_est = id_est, nome= nome |

Estas duas declarações acima irão gerar dois tipos de entidades, Disciplina e Estudante, no esquema integrado. Podemos reparar que o real estado das associações entre disciplinas e estudantes, descritas por S5 e S6, podem ser resumidas em um único objeto estuda. Consequentemente, o integrador irá gerar, no

esquema integrado, dois tipos de relacionamentos entre Disciplina e Estudante, um para expressar o *link* Estudante – disciplinas de S5 e outro para expressar o *link* Disciplina – alunos de S6, como podemos visualizar através da figura 17:

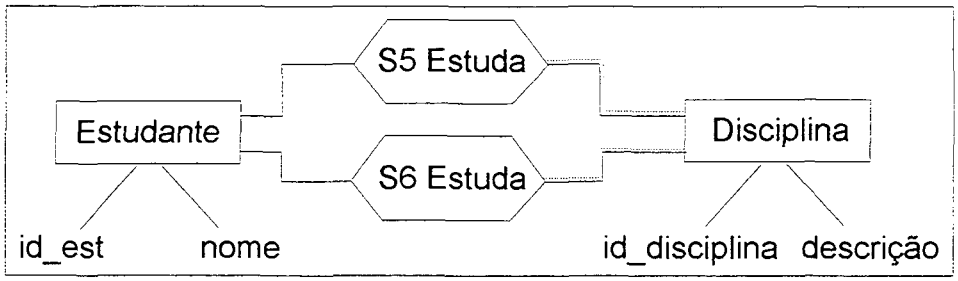


FIGURA 17: Esquema integrado de S5 e S6.

Para que o integrador integre estes dois *links* em um único relacionamento, produzindo o esquema integrado S7, que pode ser visualizado na figura 18, abaixo, o DBA deve especificar que os dois *links* possuem a mesma semântica.

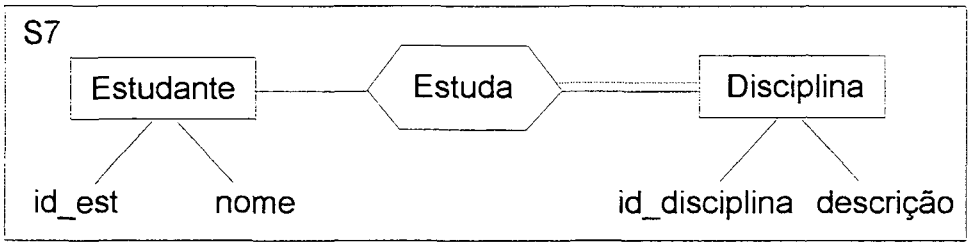


FIGURA 18: Esquema integrado S7.

Nesta metodologia o DBA iria definir a seguinte regra de correspondência entre caminhos (a qual será explicada no próximo parágrafo):

Estudante – disciplinas ≡ alunos – Disciplina

Dois caminhos, ou *links*, podem ser declarados como correspondentes somente se eles relacionarem elementos correspondentes. Aí está a razão da definição das regras de caminhos referirem-se não ao RWS de todos os elementos correspondentes, mas ao sub conjunto dos RWS os quais envolvem somente os objetos que têm um objeto correspondente no outro banco de dados. Na definição D7 discorre-se sobre este sub conjunto dos RWS.

**Regras de equivalência entre caminhos (definição D7)**

Considera-se $X_1 — X_2 — ... — X_n$ como sendo um caminho do esquema S, e $Y_1 — Y_2 — ... — Y_p$ o caminho do esquema S', tal que exista um regra de correspondência relacionando $X_1$ a $Y_1$ e $X_n$ a $Y_p$.

Considera-se $RWS'(X_1)$ como sendo o sub conjunto de $RWS(X_1)$, definido pelas restrições do objeto de $X_1$, os quais estão declarados nas regras de correspondências, com os objetos de $Y_1$. Tem-se $RWS'(Y_1)$, $RWS'(X_n)$ e $RWS'(Y_p)$ como sendo restrições similares dos respectivos RWS.

Temos $RWS'(X_1 — X_2 — ... — X_n)$ como sendo o sub conjunto de $RWS(X_1 — X_2 — ... — X_n)$ definido por suas restrições às partes de objetos em $RWS'(X_1)$ x $RWS'(X_n)$ e de maneira similar para $RWS'(Y_1 — Y_2 — ... — Y_p)$.

A declaração que mostra que os dois caminhos são equivalentes é a seguinte:

$$X_1 — X_2 — ... — X_n \equiv Y_1 — Y_2 — ... — Y_p$$

a qual expressa que:

$$RWS'(X_1 — X_2 — ... — X_n) \equiv RWS'(Y_1 — Y_2 — ... — Y_p).$$

De acordo com os esquemas S8 e S9, os quais são mostrados na figura 19, podemos citar alguns exemplos.

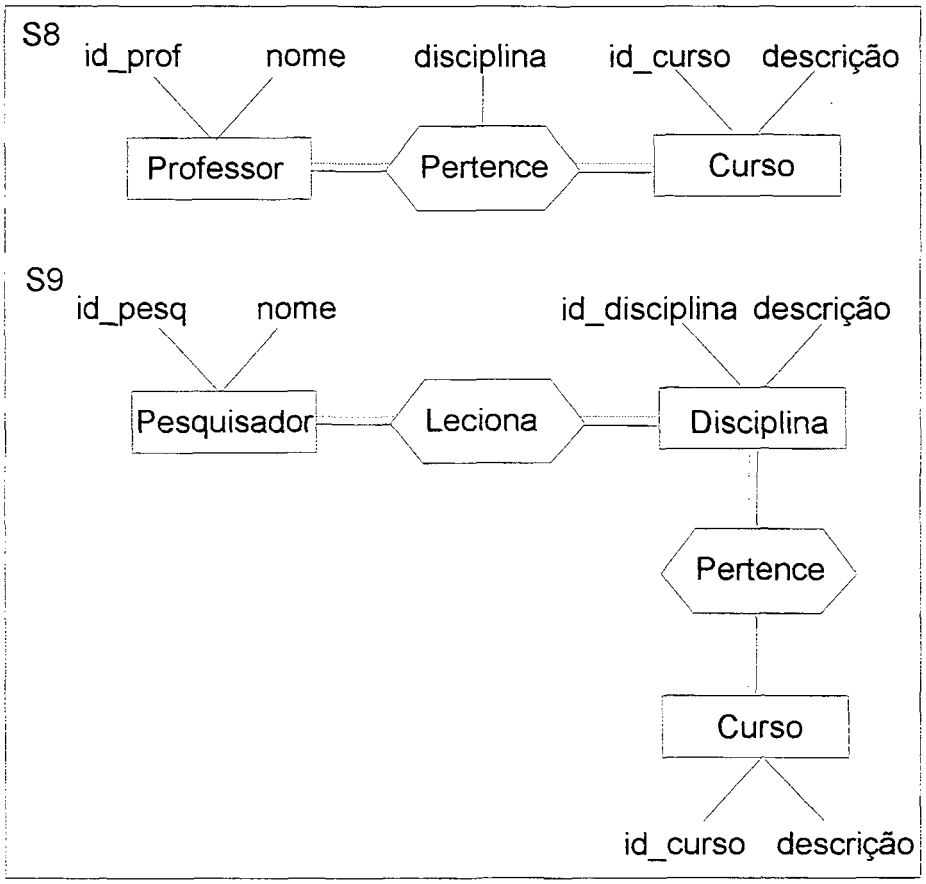


FIGURA 19: Diagrama dos esquemas S8 e S9.

Pesquisador $\subseteq$ Professor com os atributos correspondentes:

id_pesq $\subseteq$ id_prof

nome = nome

Curso = Curso com os atributos correspondentes:

id_curso = id_curso

descrição = descrição

Professor – Pertence – Curso ≡ Professor – Leciona – Disciplina – Pertence – Curso

### 6.1.5. A integração de esquemas

Iremos discorrer, nesta seção, sobre as regras de integração que geram o esquema integrado.

Esta metodologia abrange somente as regras de correspondência (regras de inclusão, interseção e exclusão não serão abrangidas). A mesma restringe-se, também, aos atributos atômicos e aos atributos complexos com componentes de atributos atômicos. Estas regras são geradas assumindo-se uma estratégia de integração binária, onde os esquemas são integrados dois a dois.

Cada regra de integração é gerada de acordo com o modelo genérico GDM. Customizações são demonstradas para os modelos relacional, entidade relacionamento e orientado a objeto.

Quando se aplica uma regra genérica a um modelo particular, as restrições que são específicas a um modelo devem ser levadas em consideração, tais como:

- Existência de dependências.
- Para a maioria dos modelos orientados a objetos, o fato de que os atributos referência estão relacionados diz que um *link* A $\rightarrow$ B e um *link* A $\leftarrow$ B não geram a mesma resposta aos usuários.
- Para os modelos ER, o fato de que alguns atributos são mandatórios e mono-valorados implica em que estes atributos não podem possuir um valor nulo e devem sempre apontar para um único objeto.

Para cada elemento e *link*, nos esquemas iniciais, o processo de integração de bancos de dados deve:

- Definir quais elementos devem ser inseridos no esquema resultante.

- Definir a distribuição da informação anexada a estes elementos, mostrando em qual banco de dados local está o sub conjunto das informações a serem visualizadas.
- Definir o mapeamento entre os esquemas iniciais e o esquema integrado. Quando n esquemas iniciais são integrados usando a estratégia binária, os esquemas integrados intermediários são utilizados como entrada para o próximo passo da integração. Neste caso é preferível gerar uma regra de correspondência, ao invés de mapeamentos, para encontrar condições iniciais para o próximo passo, ou seja, para a integração dos esquemas seguintes.

Estes mapeamentos suportam a tradução de consultas globais, no esquema integrado, para consultas locais, nos esquemas locais. Estes mapeamentos estão baseados na álgebra ERC+ (PARENT e SPACCAPIETRA, 1987). Esta álgebra estende a álgebra relacional para tratar os tipos de entidades, tipos de relacionamentos e estruturas dos atributos complexos.

Nenhum algoritmo de integração é sugerido nesta metodologia devido ao fato de que a escolha de um algoritmo depende da estratégia de integração utilizada e do grau de interação que o DBA irá realizar.

#### 6.1.5.1. Princípios da integração

As definições das regras de integração seguem dois princípios básicos, os quais são independentes do modelo:

- O escopo das regras de integração deve incluir a integração de elementos e de *links*.
- Sempre que existir um conflito estrutural entre dois esquemas, o esquema integrado irá utilizar a estrutura mais abrangente, ou seja, aquela que possuir o menor número de restrições.

A demonstração do primeiro princípio pode ser observada na seção 6.1.4.3. Regras de correspondências entre caminhos.

A comprovação do segundo princípio está apoiada no fato de que o esquema integrado deve suportar consultas e atualizações em todos os bancos de dados que o formam. Se diferentes restrições são utilizadas, em diferentes bancos de dados, a

restrição do esquema integrado deve ser a *mais fraca*, para que nenhum acesso seja rejeitado no esquema integrado.

Por exemplo, se o limite de idade de pessoas cadastradas em um banco de dados BD1 for entre 20 e 50 anos e de um BD2 for entre 20 e 65 anos, no esquema integrado o limite de idade deve estar entre 20 e 65 anos. A restrição do BD1 será garantida através de mapeamentos entre o esquema integrado e o BD1.

Este último princípio deve ser utilizado na estrutura dos dados. A identificação da estrutura com menor número de restrições depende do modelo de dados utilizado, conforme será descrito abaixo.

**O modelo GDM e o modelo orientado a objeto:**

Os tipos de elementos GDM são objetos e atributos. Eles diferem em dois aspectos: tipos de objetos possuem identificação, enquanto que atributos não.

Tipos de objetos *não* possuem dependência: eles podem estar ligados a outros tipos de objetos e a atributos, porém esta não é uma regra mandatória. Por outro lado, atributos têm que estar ligados a um e somente um outro elemento, que será seu elemento pai (tipo de objeto ou de atributo complexo). Esta dependência pode ser sumariada de acordo com o quadro 3.

QUADRO 3: Dependências existentes no modelo GDM e no modelo orientado a objeto.

| | Tipo de Objeto | Atributo |
|---|---|---|
| Tipo de Objeto | 0:n | 0:n |
| Atributo | 1:1 | 1:1 |

A *estrutura* menos restritiva, neste caso, é o tipo de objeto. Quando um tipo de objeto O1 de um esquema S1 corresponder a um valor ou a um atributo complexo A2 de um esquema S2, um tipo de objeto O deverá ser inserido no esquema integrado e um *link* deve ser adicionado para ligar O ao elemento pai de A2, como é mostrado na figura 20.

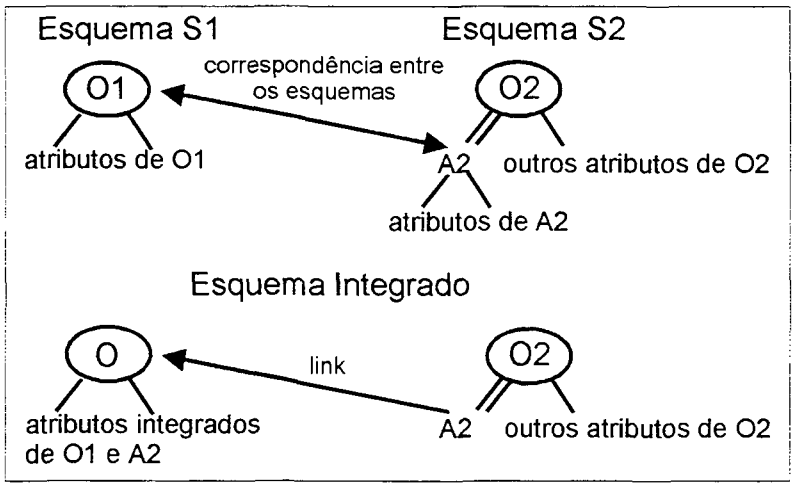


FIGURA 20: Correspondência de objetos.

**O modelo GDM e o modelo relacional:**

Neste caso, como podemos verificar no quadro 4, atributos e relações não possuem identidade. A estrutura menos restritiva é a relação.

QUADRO 4: Dependências existentes no modelo GDM e no modelo relacional.

| | Relação | Atributo |
|---|---|---|
| Relação | 0:n | 1:n |
| Atributo | 1:1 | 0:0 |

**O modelo GDM e o modelo entidade relacionamento:**

Neste caso são referenciados os modelos ER, como o ERC+, onde tipos de entidades e de relacionamentos possuem uma identidade, enquanto atributos não possuem.

Quando um tipo de entidade de um esquema S1 corresponder a um atributo ou a um tipo de relacionamento de um esquema S2, a estrutura menos restritiva é o tipo de entidade. No esquema integrado, um tipo de entidade será gerado com um tipo de relacionamento que expressará o *link* do atributo com todos os *links* de S2.

Quando um tipo de relacionamento e um atributo são correspondentes e suas dependências não são compatíveis, um tipo de relacionamento deve ligar, no mínimo a dois tipos de entidades, sendo que um atributo, deve ligar, no mínimo, a um tipo de entidade. Eles serão integrados em um tipo de entidade mais um tipo de relacionamento expressando o atributo e todos os *links*.

Pode-se visualizar um resumo dos parágrafos anteriores através do quadro 5.

QUADRO 5: Dependências existentes no modelo GDM e no modelo ER.

| | Tipo de Entidade | Tipo de Relacionamento | Atributo |
|---|---|---|---|
| Tipo de Entidade | 0:0 | 0:n | 0:n |
| Tipo de Relacionamento | 2:n | 0:0 | 0:n |
| Atributo | 1:1 | 1:1 | 1:1 |

### 6.1.5.2. Definições básicas para unir estruturas de atributos

Primeiramente é definido como dois atributos atômicos são unidos para produzir um atributo integrado.

**Integração de dois valores de atributos correspondentes de dois elementos equivalentes (definição D8)**

Consideram-se E1, E2 como sendo dois elementos correspondentes (tipos de objetos ou atributos complexos) de dois esquemas S1 e S2. Tem-se A1 e A2 significando valores de atributos atômicos de E1 e E2, respectivamente.

Se é declarado que A1 e A2 são correspondentes, então:

E1 ≡ E2, com seus atributos correspondentes: attcor(A1,A2).

E, neste caso, a integração de A1 e A2 é definida como um atributo simples de A, tal que:

- Seu nome é A1, exceto se o DBA escolher outro nome, o qual deve ser documentado.
- Seu domínio é definido como:

  se attcor(A1,A2) é A1 = A2 ou A1 $\supseteq$ A2 então, domínio(A1)

  se attcor(A1,A2) é A1 $\cap$ A2 ou A1 $\neq$ A2 então, domínio(A1) $\cup$ domínio(A2).
- Suas cardinalidades são definidas como:

  se attcor(A1,A2) é A1 = A2 ou A1 $\supseteq$ A2 ou A2=f(A1)

  então, cardmin(A) = cardmin(A1), cardmax(A)=cardmax(A1)

se attcor(A1,A2) é A1 $\cap$ A2

então, cardmin(A) = Max(cardmin(A1), cardmin(A2))

cardmax(A) = cardmax(A1) + cardmax(A2)

se attcor(A1,A2) é A1 $\neq$ A2

então, cardmin(A) = cardmin(A1) + cardmin(A2)

cardmax(A) = cardmax(A1) + cardmax(A2).

A partir das definições acima é determinado o operador de junção de integração - *integratejoin*, unindo dois elementos compostos, tipos de objetos ou atributos complexos.

**Junção de integração (definição D9)**

Tem-se E1, com os valores dos atributos ($A_{11}$, ... , $A_{1j}$, $B_1$, ..., $B_k$), e E2, com os valores dos atributos ($A_{21}$, ... , $A_{2j}$, $C_1$, ..., $C_h$), como sendo dois elementos (tipos de objetos ou atributos complexos) dos bancos de dados S1 e S2, conhecidos como equivalentes:

E1 $\equiv$ E2 com seus atributos correspondentes:

$attcor_1(A_{11}, A_{21})$, $attcor_2(A_{12}, A_{22})$, ..., $attcor_j(A_{1j}, A_{2j})$,

sendo que $attcor_1(A_{11}, A_{21})$ é a declaração que especifica o mapeamento 1:1 entre os identificadores de E1 e E2.

A operação:

E:= junção de integração (E1, E2, $attcor_1(A_{11}, A_{21})$, $attcor_2(A_{12}, A_{22})$, ..., $attcor_j(A_{1j}, A_{2j})$)

que cria um novo tipo de objeto E definido pelos seguintes itens:

- Sua estrutura consiste da união entre os atributos de E1 e E2, definidos como:
  - Um atributo $B'_i$ para cada atributo $B'_i$ de E1 o qual não possui contrapartida em E2; seu domínio e cardinalidades são iguais a aquelas relativas a $B_i$.
  - Um atributo $C'_i$ para cada atributo $C_i$ de E2 o qual não possui contrapartida em E1; seu domínio e cardinalidades são iguais a aquelas relativas a $C_i$.
  - Um atributo $A_i$ para cada $attcor_i(A_{1i}, A_{2i})$; $A_i$ é a integração de $A_{1i}$ e $A_{2i}$.
- O conjunto de valores deste objeto contém uma ocorrência "e" para cada objeto no mundo real dos RWS de E1 e E2. O valor de "e" é definido

como uma união entre os valores de E1 e as ocorrências de E2 que descrevem o objeto no mundo real, as quais estão unidas por um mapeamento 1:1 $attcor_1(A_{11}, A_{21})$:

- Para cada atributo $B'_1$: $e.B'_1 = e1.B_1$.
- Para cada atributo $C'_1$: $e.C'_1 = e2.C_1$.
- Para cada atributo $A_1$

  se $attcor_i(A_{1i}, A_{2i})$ indica $A_{1i} = A_{2i}$ ou $A_{1i} \supseteq A_{2i}$

  então $e.A_1 = e1.A_{1i}$

  se $attcor_i(A_{1i}, A_{2i})$ indica $A_{1i} \cap A_{2i}$ ou $A_{1i} \neq A_{2i}$

  então $e.A_1 = e1.A_{1i} \cup e2.A_{2i}$.

### 6.1.5.3. Integração de elementos locais e *links*

Regras de integração de modelos independentes aplicam-se somente a elementos e *links* que aparecem em apenas um dos esquemas a serem integrados.

**1ª Regra de Integração: integração de elementos locais e *links***

Cada elemento X1, de um esquema S1, o qual não possui contrapartida em outro esquema, é adicionado, como elemento, ao esquema integrado. O tipo de X é o mesmo de X1.

Regra de correspondência: $X \equiv X1$.

Mapeamento: $X = X1$.

Distribuição: X é X1 no banco de dados S1.

Cada *link*, X1 — Y1, do esquema S1, o qual não possui contrapartida em outro esquema, é adicionado, como um *link* X — Y, ao esquema integrado, onde X e Y são os elementos integrados correspondentes a X1 e Y1. O tipo de X — Y depende dos tipos de X e Y, como veremos na seção 6.1.5.5. Integração de dois *links*.

Regra de correspondência: X — Y ≡ X1 — Y1.

Mapeamento: X — Y = *rename* [X1 — Y1].

Distribuição: X — Y no banco de dados S1.

### 6.1.5.4. Integração de dois tipos de objetos

Esta seção considera somente os valores de atributos com os tipos de objetos e atributos referências que participam do *link* ou das regras de correspondências de caminhos. A integração destes elementos será discutida nas próximas seções.

**2ª Regra de Integração: integração de dois tipos de objetos e os valores de seus atributos**

Considera-se $X_1$, tendo atributos com valores ($A_{11}$, ..., $A_{1j}$, $B_1$, ..., $B_k$), e $X_2$, tendo atributos com valores ($A_{21}$, ..., $A_{2j}$, $C_1$, ..., $C_h$) como sendo dois tipos de objetos em dois esquemas, $X_1 \in S_1$, $X_2 \in S_2$, tal que:

$X_1 \equiv X_2$ com seus atributos correspondentes:

$attcor_1(A_{11}, A_{21})$, $attcor_2(A_{12}, A_{22})$, ..., $attcor_j(A_{1j}, A_{2j})$

os elementos, no esquema integrado, resultante da integração de $X_1$ e $X_2$ são tipos de objetos X, tal que:

- Seus nomes são um dos nomes de $X_1$, exceto se o DBA escolher outro nome.
- Suas estruturas consistem da união dos atributos de $X_1$ e $X_2$, como definido pela junção de integração de $X_1$ e $X_2$.

Regras de correspondências relacionando X a $X_1$ e $X_2$ são as seguintes:

$X \equiv X_1$ com seus atributos correspondentes:

$attcor_1(A_1, A_{11})$, $attcor_2(A_2, A_{12})$, ..., $attcor_j(A_j, A_{1j})$

$attcor_1(B'_1, B_1)$, $attcor_2(B'_2, B_2)$, ..., $attcor_k(B'_k, B_k)$

$X \equiv X_2$ com seus atributos correspondentes:

$attcor_1(A_1, A_{21})$, $attcor_2(A_2, A_{22})$, ..., $attcor_j(A_j, A_{2j})$

$attcor_1(C'_1, C_1)$, $attcor_2(C'_2, C_2)$, ..., $attcor_h(C'_h, C_h)$

Mapeamentos entre X, $X_1$ e $X_2$ podem ser definidos como:

X := junção de integração ($X_1$, $X_2$, $attcor_1(A_{11}, A_{21})$, $attcor_2(A_{12}, A_{22})$, ..., $attcor_j(A_{1j}, A_{2j})$)

X1 := projeção[$A_1$, ..., $A_j$, $B_1$, ..., $B_k$] X

X2 := *rename*[$A_1$:$A_{21}$ ..., $A_j$ : $A_{2j}$] projeção[$A_1$, ..., $A_j$, $C_1$, ..., $C_h$] X.

A descrição da atual distribuição dos registros que X armazena nos dois bancos de dados, S1 e S2, pode ser determinada, mais precisamente, pela divisão de X em fragmentos (particionamento vertical):

a projeção[$B_1$, $B_k$] X está no banco de dados $S_1$

a projeção[$C_1$, $C_h$] X está no banco de dados $S_2$.

Para cada $A_j$, se $attcor_j(A_{1j}, A_{2j})$ é:

- $A_{1j} = A_{2j}$ então $A_j$ é duplicado nos dois bancos de dados.
- $A_{1j} \supseteq A_{2j}$ então $A_j$ está no sítio $S_1$, e alguns valores são duplicados no banco de dados $S_2$.
- $A_{1j} \cap A_{2j}$ então $A_j$ é parcialmente duplicado.
- $A_{1j} \neq A_{2j}$ então $A_j$ é distribuído entre os bancos de dados $S_1$ e $S_2$.

Iremos exemplificar esta regra através dos seguintes esquemas:

S10: Disciplina_pos(id_disciplina, descrição)

S11: Disciplina (sigla_discip, nome, depto)

Vamos supor que os dois bancos de dados descrevem o mesmo tipo de disciplinas. A regra de correspondência seria:

Disciplina_pos ≡ disciplina com os atributos correspondentes:

id_disciplina = sigla_discip, descrição = nome

O esquema integrado seria:

Disciplina_pos(id_disciplina, descrição, depto)

A 2ª regra de integração pode ser aplicada diretamente nos seguintes casos:

- Nos modelos relacionais, na integração de relações sem qualquer chave externa.
- Nos modelos ER, na integração de tipos de entidades.
- Nos modelos orientados a objetos, na integração de tipos de objetos sem qualquer atributos referência.

### 6.1.5.5. Integração de dois *links*

A 3ª regra de integração trata dos *links* elementares (*links* de referência e de atributos) e permite a integração de dois *links* equivalentes, os quais unem elementos equivalentes. A 4ª regra de integração trata dos caminhos que são compostos por vários *links*.

**3ª Regra de Integração: regra de integração de *links***

Consideram-se $A_1$ e $B_1$ como sendo dois *links* de elementos (tipos de objetos, atributos atômicos ou complexos) do esquema $S_1$. $A_2$ e $B_2$ são dois elementos unidos (tipos de objetos, atributos atômicos ou complexos) do esquema $S_2$, com as seguintes regras de integração:

$A_1 \equiv A_2$

$B_1 \equiv B_2$

$A_1 — B_1 \equiv A_2 — B_2$

Considera-se A como sendo o elemento integrado, no esquema integrado, correspondente a $A_1$ e $A_2$.

Considera-se B como sendo o elemento integrado, no esquema integrado, correspondente a $B_1$ e $B_2$.

A integração dos *links* $A_1 — B_1$ e $A_2 — B_2$ é o *link* A — B. O tipo do *link* depende de A e B:

- Se A ou B são valores de atributos então A — B é um *link* de atributo.
- Se A e B são tipos de objetos então A — B é um *link* de referência: um atributo referência, nomeado como B, é adicionado a A e vice-versa.

Como as três regras de correspondência são equivalentes, as cardinalidade dos dois *links*, $A_1 — B_1$ e $A_2 — B_2$, são, necessariamente, as mesmas e as cardinalidades do *link* integrado A — B também serão as mesmas.

As regras de correspondências são:

$A — B \equiv A_1 — B_1$

$A — B \equiv A_2 — B_2$

E os mapeamentos:

$A — B \equiv [rename]A_1 — B_1$

$A — B \equiv [rename]A_2 — B_2$

Distribuição: o *link* A — B é duplicado, sendo armazenado nos dois bancos de dados, $S_1$ e $S_2$.

Discutiremos, agora, como a 3ª regra de integração é aplicada nos diferentes modelos.

Modelo Relacional

Como o modelo relacional não possui nenhuma restrição na referência de seus atributos (uma relação pode ter zero, uma ou muitas chaves externas), a 3ª regra de integração é aplicada sem nenhuma modificação. Conforme os exemplos:

S12:

Pesquisador (id_pesq, nome, endereço, linha_pesq)

Linha_Pesquisa (id_linhap, descrição)

Linha_Pesquisa•id_linhap $\supseteq$ Pesquisador•linha_pesq

S13:

Pesquisador (id_pesq, nome, endereço)

Linha_Pesquisa (id_linhap, descrição, pesquisador)

Pesquisador•id_pesq $\supseteq$ Linha_Pesquisa• pesquisador

Regras de correspondência entre S12 e S13:

Pesquisador $\equiv$ Pesquisador com os atributos correspondentes:

id_pesq = id_pesq, nome = nome, endereço = endereço

Linha_Pesquisa $\equiv$ Linha_Pesquisa com os atributos correspondentes:

id_linhap = id_linhap, descrição = descrição

Pesquisador – Linha_Pesquisa = Linha_Pesquisa - Pesquisador

Modelos Orientados a Objetos

A 3ª regra de integração deve ser modificada para tratar os *links* diretos da maioria dos modelos orientados a objetos.

Supõe-se um modelo orientado a objeto onde a referência dos atributos, como os valores dos atributos, são diretas. O significado de "direta" diz que a referência de um atributo permite acesso direto somente através da referência do elemento pai deste objeto. A 3ª regra é ajustada como segue:

3ª Regra para modelos orientados a objetos:

A integração de dois *links* equivalentes diretos:

$A_1 \rightarrow B_1 \equiv A_2 \rightarrow B_2$

gera um *link* direto A $\rightarrow$ B. Se B é um valor de um atributo, A $\rightarrow$ B é um *link* de atributo. Se B é um tipo de objeto, A $\rightarrow$ B é um *link* de referência: A carrega uma referência de atributo apontando para B.

*Links* diretos opostos, como $A_1 \rightarrow B_1$ versus $B_2 \rightarrow A_2$, não podem ser declarados como correspondentes. Neste caso, a 3ª regra não é aplicada e os dois

*links* são integrados através da 1ª regra. Como os elementos locais, ambos os *links* serão adicionados no esquema integrado.

Modelo Entidade Relacionamento

A 3ª regra de integração pode ser diretamente aplicada nos modelos ER quando da integração de *links* existentes entre tipos de entidades e tipos de relacionamentos.

Através da 1ª e da 3ª regra de integração pode-se definir regras dedutivas para integrar tipos de objetos com suas referências de atributos, quer dizer, seus relacionamentos.

*Links* de referência equivalentes são integrados,. considerando-se suas cardinalidades, adicionando-se *links* de referência locais.

Aplicando estas duas regras aos modelos ER e agindo conforme os princípios básicos de escolha das estruturas menos restritivas (tipos de entidades), no caso de conflitos estruturais, produz-se as seguintes regras:

Regra ER 1+3:

A integração de um tipo de entidade E1 e seu equivalente tipo de relacionamento n-ário R2 resulta em um tipo de entidade E e um tipo de relacionamento binário unindo E e os tipos de entidades que R2 unia.

A integração de dois tipos de relacionamento equivalentes, R1 e R2, resulta em um tipo de relacionamento o qual une todas as entidades que R1 ou R2 unia.

Podemos exemplificar este caso através dos esquemas S12 e S13 visualizados na figura 21.

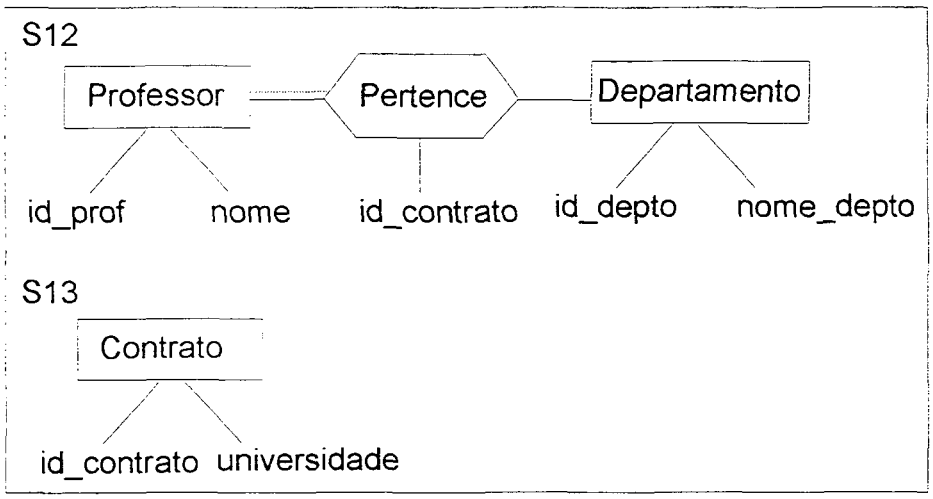


FIGURA 21: Diagrama dos esquemas S12 e S13.

Regras de correspondência entre S12 e S13:

Pertence ≡ Contrato com os atributos correspondentes:

id_contrato ≡ id_ contrato

O esquema integrado referente os esquemas S12 e S13 pode ser visualizado na figura 22.

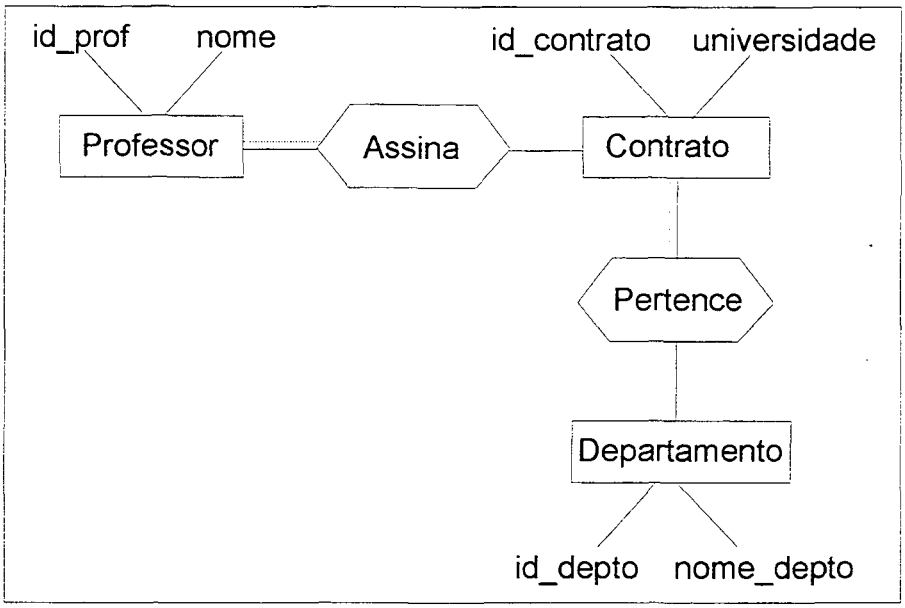


FIGURA 22: Esquema integrado de S12 e S13.

Este esquema integrado pode ser simplificado pela união de Assina+Contrato+Pertence em um único relacionamento se nenhum outro relacionamento estiver ligado a Contrato.

### 6.1.5.6. Integração de *links* e caminhos

O processo de integração não deve gerar informação redundante no esquema integrado. Quando se integra *links* e caminhos, é importante verificar se cada um sustenta informações independentes, ou se um pode ser deduzido a partir do outro.

Dois casos podem acontecer:

- Um *link* $A_1$ — $B_1$ é equivalente a um caminho $A_2$ — ... — $B_2$. Neste caso, mantém-se, no esquema integrado, somente o caminho $A_2$ — ... — $B_2$. O *link* direto será deduzido através de uma composição dos *links* de $A_2$ — ... — $B_2$.

- Dois caminhos $A_1$ — ... — $B_1$ e $A_2$ — ... — $B_2$ são equivalentes. Neste caso, os dois caminhos devem ser mantidos no esquema integrado. Apagando-se um caminho ocasionaria o desaparecimento de todos os *links* deste caminho, os quais não são equivalentes a nenhum outro *link* ou caminho. Uma regra de restrição é adicionada ao esquema integrado ditando que estes dois caminhos são equivalentes.

**4ª Regra de Integração: regras de integração de *links* e caminhos**

Consideram-se $A_1$, $B_1$, ..., $D_1$ como sendo elementos do esquema $S_1$, e $A_2$, $B_2$, ..., $D_2$ como sendo elementos do esquema $S_2$, com as seguintes regras de correspondências:

$A_1 \equiv A_2$, $D_1 \equiv D_2$.

Considera-se A (assim como D) como sendo os elementos de integração, no esquema integrado, correspondendo a $A_1$ e $A_2$ (assim como $D_1$ e $D_2$), então:

- A regra de correspondência entre um *link* e um caminho:

  $A_1$ — $D_1 \equiv A_2$ — $B_2$ —... — $D_2$

  gera, no esquema integrado, o caminho A — $B_2$' —... — D

  onde $B_2$' é o elemento integrado correspondente a $B_2$.

- A regra de correspondência entre dois caminhos:

  $A_1$ — $B_1$ ... — $D_1 \equiv A_2$ — $B_2$ —... — $D_2$

  gera, no esquema integrado, dois caminhos:

  $A_1$ — $B_1$'... — $D_1$ e $A_2$ — $B_2$'—... — $D_2$

  onde $B_1$' e $B_2$' são os elementos integrados correspondente a $B_1$ e $B_2$, e uma regra de integridade dita que dois caminhos unem as mesmas ocorrências.

Em ambos os casos, os caminhos gerados são criados de acordo com os conceitos de modelagem dos elementos unidos, conforme a 3ª regra de integração.

A 4ª regra de integração inclui a 3ª regra.

Um esquema será integrado à medida que o DBA descrever as regras de correspondências. Por exemplo, temos E1 e E2 como sendo dois esquemas com as seguintes regras de correspondências:

A1 ≡ A2, C1 ≡ C2, F1 ≡ F2

A1 — ... — C1 ≡ A2 — ... — C2

C1 — ... — F1 ≡ C2 — ... — F2

Se, ao invés de estabelecer as duas regras acima, o DBA declarar:

A1 — ... — C1 — ... — F1 ≡ A2 — ... — C2 — F2

Neste caso, menos conhecimento é passado ao integrador e a integração será mais lenta.

Modelo Relacional

A 4ª regra de integração, assim como a 3ª, é aplicada sem modificações.

Exemplo:

S14: Pesquisador (id_pesq, id_linhap , nome, cargo, ramal)

Linha_pesquisa (id_linhap, descrição)

Dependência inclusa:

Linha_pesquisa.id_linhap ⊇ Pesquisador.id_linhap

S15: Pesquisador (id_pesq, nome, cargo, ramal)

Linha_pesquisador (id_pesq, id_linhap)

Linha_pesquisa (id_linhap, descrição)

Dependência inclusa:

Linha_pesquisa.id_linhap ⊇ Linha_pesquisador.id_linhap

Pesquisador.id_pesq ⊇ Linha_pesquisador.id_pesq

Existem as seguintes regras de correspondências entre os esquemas S14 e S15:

Pesquisador ≡ Pesquisador com os atributos correspondentes:

id_pesq = id_pesq, nome = nome, cargo = cargo, ramal = ramal

Linha_pesquisa ≡ Linha_pesquisa com os atributos correspondentes:

id_linhap = id_linhap , descrição = descrição

Pesquisador – Linha_pesquisa ≡

Pesquisador – Linha_pesquisador – Linha_pesquisa

O esquema integrado resultante é equivalente a S15.

Modelo Orientado a Objeto

A 4ª regra de integração, assim como a 3ª, deve ser ajustada para que os *links* e os caminhos orientados na mesma direção possam ser integrados.

Modelo Entidade Relacionamento

A 4ª regra de integração, assim como a 3ª, é aplicada, sem modificações.

### 6.1.5.7. Integração de um tipo de objeto e um atributo

Um dos princípios básicos da integração, utilizados por esta metodologia, diz que, sempre que existirem conflitos de descrição nos esquemas locais, o esquema integrado transportará a representação menos restritiva para que possam ser realizados os devidos mapeamentos.

A integração de um tipo de objeto O e o valor de um atributo complexo A produz um tipo de objeto que tem a estrutura resultante da união das estruturas de O e A, conforme dita a 2ª regra de integração.

A distribuição e o mapeamento também são similares aos da 2ª regra. A principal diferença é que o objeto integrado, é unido, através de uma referência, de um atributo ao elemento pai de A.

**5ª Regra de Integração: integração de um tipo de objeto e um valor de um atributo complexo**

Considera-se $X_1$, com os seguintes valores dos atributos ($A_{11}$, ..., $A_{1j}$, $B_1$, ..., $B_k$), como sendo um tipo de objeto do esquema $S_1$, e $X_2$ como sendo um atributo complexo do elemento $E_2$ do esquema $S_2$ com os seguintes valores componentes do atributo ($A_{21}$, ..., $A_{2j}$, $C_1$, ..., $C_h$), ou um valor de um atributo atômico; neste caso, considera-se que $X_2$ possui, ele mesmo, um atributo componente.

A regra de correspondência é:

$X_1 \equiv X_2$ com seus atributos correspondentes:

$attcor_1(A_{11}, A_{21})$, $attcor_2(A_{12}, A_{22})$, ... , $attcor_j(A_{1j}, A_{2j})$,

considera-se E como sendo o elemento correspondente a $E_2$ no esquema integrado, os elementos, no esquema integrado, resultantes da integração de $X_1$ e $X_2$ são objetos do tipo X, e existe um *link* referência entre E e X, tal que:

- O atributo $X_2$ de $E_2$ é transformado em um atributo referência $X'_2$ referenciando X; as cardinalidades de $X'_2$ são iguais aquelas de $X_2$.
- O nome de X é o mesmo que o nome de $X_1$, exceto se o DBA escolher outro nome.
- A estrutura de X consiste da união dos atributos de $X_1$ e $X_2$, como definido na junção de integração de $X_1$ e $X_2$.

As regras de correspondência relacionando X a $X_1$ e $X_2$ são as seguintes:

$X \equiv X_1$ com seus atributos correspondentes:

$attcor_1(A_1, A_{11})$, $attcor_2(A_2, A_{22})$, ... , $attcor_j(A_j, A_{1j})$

$attcor_1(B'_1, B_{11})$, $attcor_2(B'_2, B_{22})$, ... , $attcor_j(B'_k, B_k)$

$X \equiv X_2$ com seus atributos correspondentes:

$attcor_1(A_1, A_{21})$, $attcor_2(A_2, A_{22})$, ... , $attcor_j(A_j, A_{2j})$

$attcor_1(C'_1, C_{11})$, $attcor_2(C'_2, C_{22})$, ... , $attcor_h(C'_h, C_h)$

$E — X \equiv E_2 — X_2$

O mapeamento entre o esquema integrado e os esquemas $S_1$ e $S_2$ pode ser definido como:

X = junção de integração($X_1$, $X_2$, $attcor_1(A_{11}, A_{21})$, $attcor_2(A_{12}, A_{22})$, ... , $attcor_j(A_{1j}, A_{2j})$)

$E — X =$ *rename*$[E_2 — X_2]$

X1 = projeção$[A_1, ..., A_j, B_1, ..., B_k]$ X

X2 = *rename*$[A_1{:}A_2, ..., A_j{:}A_{2j}]$ projeção$[A_1, ..., A_j, C_1, ..., C_h]$ X

Distribuição

- X é armazenado nos dois bancos de dados, $S_1$ e $S_2$, os quais são divididos em fragmentos conforme a 2ª regra de integração.
- O *link* E — X estará somente no banco de dados $S_2$.
- Se $X_1$ e/ou $X_2$ tiverem atributos referência, a 1ª e a 3ª regra são ativadas para adicionar ou integrar os *links*.

Modelo Relacional

A 5ª regra é aplicada conforme os parágrafos abaixo.

A integração de uma relação R1 de um esquema S1 e o valor de um atributo A2 de uma relação R2 de um esquema S2, gera, no esquema integrado, uma relação R com os atributos de R1 e a regra de referência de integridade unindo a relação R2', que é a relação integrada correspondente a R2, em R.

Exemplo:

S16: Pesquisador (id_pesq, id_linhap , nome, cargo, ramal)

S17: Linha_pesquisa (id_linhap, descrição)

Regras de correspondência entre S16 e S17:

Pesquisador.id_linhap ≡ Linha_pesquisa com os atributos correspondentes:

id_linhap = id_linhap

Seguindo estas colocações o esquema integrado é:

Pesquisador (id_pesq, id_linhap , nome, cargo, ramal)

Linha_pesquisa (id_linhap, descrição)

Linha_pesquisa.id_linhap ⊇ Pesquisador.id_linhap

A regra 5 transforma o valor do atributo id_linhap de S16 em um atributo referência; o que quer dizer que id_linhap, no esquema integrado, é uma chave estrangeira que referencia Linha_pesquisa.

Modelo Entidade Relacionamento

A 5ª regra, assim como a 3ª, precisa ser ajustada para o modelo ER, como será demonstrado a seguir.

A integração de um tipo de entidade X1 de um banco de dados S1 e um tipo de atributo X2 (que possui como elemento pai o tipo de entidade E2) de um banco de dados S2, gera um tipo de objeto X e um *link* E — X, onde E é o tipo de entidade correspondente a E2 no esquema integrado.

Como o tipo de entidade X1 pode estar ligado a um relacionamento em S1, X deve ser um tipo de entidade, e o *link* E — X deve ser um relacionamento binário unindo os tipos de entidades E e X.

As regras 5 e 3 (regra de integração de *links*) permitem a integração dos esquemas S5 e S6, conforme figura 17, da página 59.

Se dois bancos de dados são equivalentes e se a equivalência entre dois *links*, Estudante – disciplinas e alunos – Disciplina, é declarada então o esquema integrado corresponde a figura 18 da página 59.

Modelo Orientado a Objeto

A 5ª regra é aplicada sem nenhuma modificação.

## 6.2. METODOLOGIA DESCRITA EM BATINI e LENZERINI (1984)

Esta metodologia é dividida em três fases que realizam as seguintes atividades:

- Os diferentes tipos de conflitos existentes entre os diferentes esquemas dos usuários são identificados e resolvidos.
- Os esquemas são mesclados em um esquema integrado inicial.
- O esquema integrado é enriquecido e reestruturado conforme as especificações do sistema.

Muitas tarefas complexas devem ser gerenciadas durante a integração: encontrar as partes comuns entre os diferentes esquemas, encontrar as diferentes

representações escolhidas pelos analistas, em alguns casos descobrindo escolhas inapropriadas, e, finalmente, descobrir as propriedades de interseção dos esquemas.

Esta metodologia utiliza a definição de Universo do Discurso (UoD) que significa uma porção arbitrária do mundo real que é representada no esquema conceitual. A definição de objeto (do Universo do Discurso) significa qualquer objeto concreto ou abstrato no Universo do Discurso.

A metodologia de BATINI e LENZERINI (1984) tem, como objetivo, gerenciar várias tarefas complexas devido a alguns casos como:

- Muitas representações equivalentes existem no modelo para o mesmo universo do discurso (falta de ortogonalidade no modelo). Pode-se dizer que um esquema S1 é menos informativo que um segundo esquema S2 se, para toda instância i1 de S1, existe uma instância i2 de S2 que responda a um mesmo conjunto de consultas. Se S1 < S2 e S2 < S1, é dito que os bancos de dados são equivalentes.
- No processo de modelagem de banco de dados, diferentes percepções e requisitos do sistema podem ser modelados de maneiras diferentes, pelos analistas ou *designers* dos bancos de dados. Este caso é chamado de pluralismo de percepções.
- Escolhas errôneas podem ter sido realizadas no esquema para nomes ou propriedades dos conceitos, sendo que o modelo conceitual aplicado para o UoD1 e para o UoD2 pode não produzir, como resultado, o esquema correto de S1 e S2 mas dois esquemas S1' e S2' não equivalentes entre si (falta de confiabilidade na modelagem).

A integração, na metodologia de BATINI e LENZERINI (1984), é realizada seguindo a estratégia binária, onde os esquemas são integrados dois a dois, conforme a figura 23:

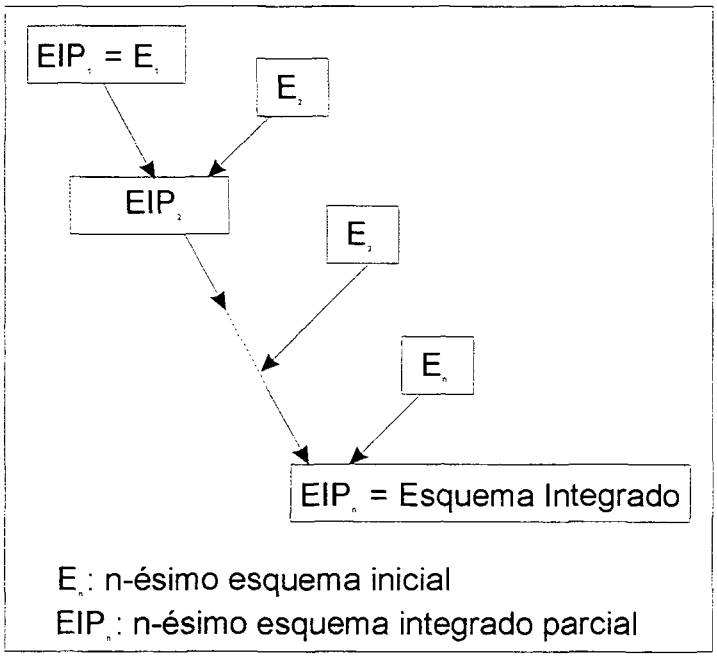


FIGURA 23: O processo de integração da metodologia de BATINI e LENZERINI (1984).

Nesta metodologia, podemos distinguir três etapas de trabalho que nos levam ao resultado final da integração: análise dos conflitos, mesclagem dos esquemas e reestruturação final.

### 6.2.1. Análise dos conflitos

Na primeira etapa desta metodologia é realizada a análise dos conflitos de nome e a compatibilidade entre os modelos.

Na análise dos conflitos de nome, verifica-se a ocorrência de homônimos e sinônimos.

Para evitarmos a explosão combinatória das comparações, uma heurística pode ser utilizada para checar os subconjuntos dos valores similares e suas restrições.

Para conceitos com o mesmo tipo, existem conjuntos de valores para os atributos; entidades e relacionamentos para as entidades; e atributos e entidades para os relacionamentos.

Para conceitos com tipos diferentes existem, para o caso de entidades e atributos, um conjunto de valores dos atributos e um conjunto de valores dos identificadores, e, no caso das entidades e relacionamentos, os atributos e um conjunto de valores dos identificadores.

O segundo item a ser verificado na etapa de análise dos conflitos é a compatibilidade dos modelos. Nesta etapa é realizada a análise dos esquemas, com o objetivo de verificar se um mesmo objeto, o qual possui o mesmo significado no mundo real, está modelado como o mesmo tipo nos dois esquemas.

O item de compatibilidade dos modelos pode ser dividido nos seguintes passos: passo de reestruturação sintática e checagem de compatibilidade.

O passo de reestruturação sintática foca o trabalho na análise dos conceitos que possuem o mesmo nome nos dois esquemas, porém, estão modelados com tipos diferentes.

São citados alguns tipos de incompatibilidade:

- Inconsistência de tipo entre um atributo de uma entidade e uma entidade: neste caso existe as seguintes possibilidades de transformações:

  **Transformação T1**: transformar um atributo em uma entidade.

  É chamada de E a entidade à qual o atributo pertence. A nova entidade é conectada a E como um novo relacionamento, no qual as cardinalidades de E possuem o mesmo valor que as cardinalidades do atributo original e a cardinalidade mínima da nova entidade é 1.

  A máxima cardinalidade da nova entidade é 1 se e, somente se, o atributo original for um identificador de E, senão a máxima cardinalidade será n.

  A nova entidade será um identificador parcial de E se, e somente se, o atributo original for um identificador desta entidade.

  Um atributo identificador pode ser introduzido para a nova entidade, com o mesmo valor do atributo original, conforme a figura 24a, 24b e 24c.

  Se mais atributos precisam ser transformados na mesma entidade, a mínima cardinalidade da nova entidade, em cada relacionamento é fixada em 0, conforme a figura 24d.

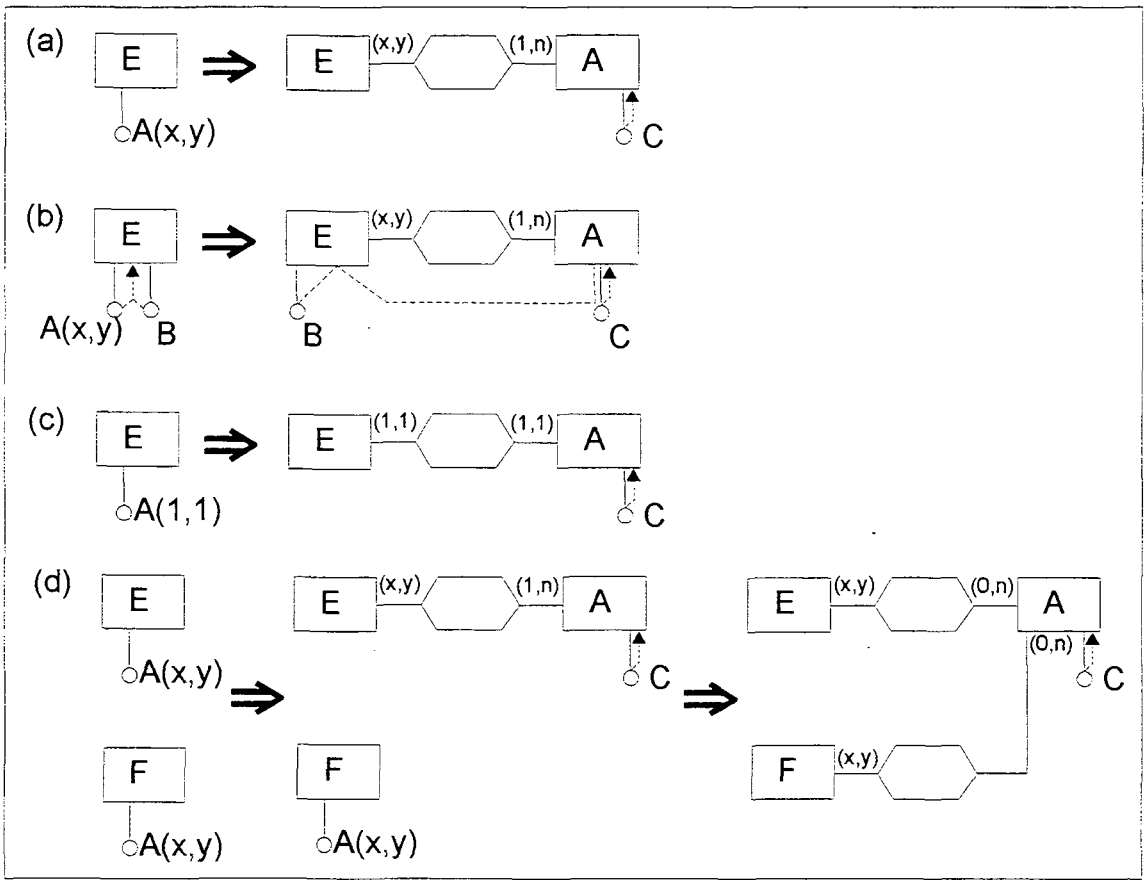


FIGURA 24: Exemplos da transformação T1.

**Transformação T2**: transformar uma entidade em um atributo.

Uma entidade pode ser transformada em um atributo de uma outra entidade E se, e somente se, a entidade estiver relacionada com E por intermédio de um relacionamento R, sendo que este relacionamento não possui atributos e a entidade possui apenas um atributo, o identificador.

A cardinalidade do atributo é uma herança do relacionamento R e o novo atributo é um identificador (ou identificador parcial) se, e somente se, a entidade original for identificadora (conforme figura 25a e 25b).

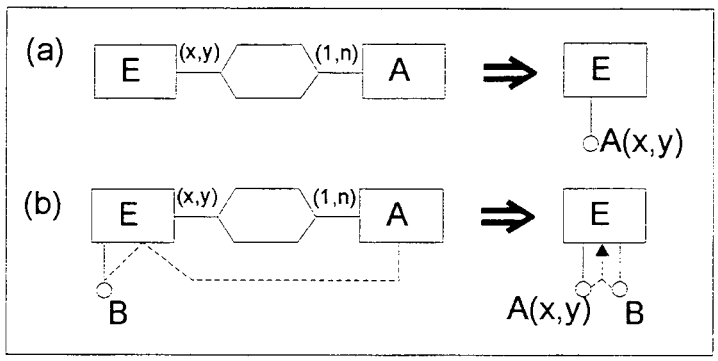


FIGURA 25: Exemplos da transformação T2.

No passo de checagem de compatibilidade, os conceitos dos dois esquemas que representam a mesma classe de objetos devem possuir o mesmo nome e o mesmo tipo.

As diferentes representações do mesmo objeto são agora analisadas em detalhes, verificando suas compatibilidades e escolhendo um modelo comum de representação.

Os conceitos podem ser considerados idênticos quando têm exatamente as mesmas características de modelagem, compatíveis quando as restrições de integridade não são contraditórias e incompatíveis caso não sigam nenhum dos casos citados.

A figura 26 mostra exemplos de conceitos compatíveis e na figura 27 podemos visualizar exemplos de conceitos incompatíveis.

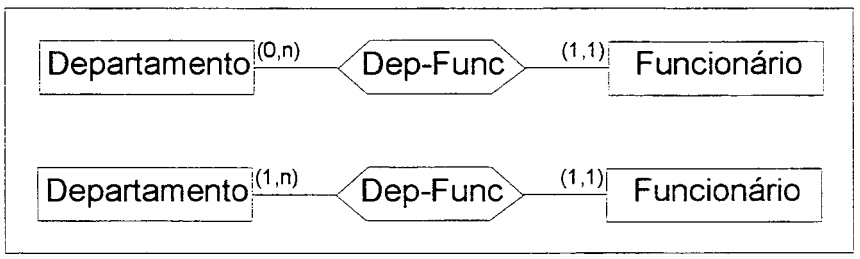


FIGURA 26: Exemplos de conceitos compatíveis.

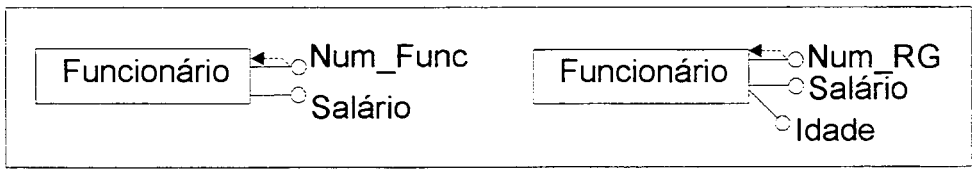


FIGURA 27: Exemplos de conceitos incompatíveis.

Uma definição formal de compatibilidade ou incompatibilidade depende do modelo conceitual utilizado para modelar o banco de dados. Alguns exemplos de incompatibilidade são:

- Diferentes cardinalidades para o mesmo atributo ou entidade (inter-relacionamento).
- Um identificador, em um esquema, não é considerado um identificador no outro esquema.
- Uma entidade é um subconjunto de uma outra entidade (transitividade) em um esquema, e o contrário acontece no outro esquema.
- Diferentes dependências funcionais são definidas para o mesmo atributo de uma entidade, em diferentes esquemas.

As soluções que podem resolver problemas de incompatibilidade são as seguintes, as quais são escolhidas de acordo com uma análise detalhada dos dois modelos.

- Uma das representações é escolhida.
- Uma representação comum é construída para que todas as restrições dos dois esquemas sejam suportadas no esquema integrado.

### 6.2.2. Mesclagem dos esquemas

Nesta fase, a parte comum do universo do discurso dos dois esquemas é representada com o mesmo nome, tipo e restrição.

Como conseqüência, os dois esquemas estruturados podem ser integrados, em um esquema integrado, que pode ser visualizado através de uma nomenclatura composta de três categorias, ou seja, um esquema desenvolvido em três cores que representam:

- Conceitos que pertencem somente ao esquema estruturado P1S1.
- Conceitos que pertencem somente ao esquema estruturado Si+1.
- Conceitos comuns.

### 6.2.3. Reestruturação final

O esquema integrado resultante da fase 6.2.2 (Mesclagem dos esquemas), é analisado para que seja obtida uma descrição mais confiável e clara do universo do discurso global da integração.

Nesta fase podem ser distinguidas mais três tarefas distintas: análise das propriedades dos inter-esquemas, análise dos ciclos redundantes e reestruturação do esquema.

As propriedades dos inter-esquemas dizem respeito às novidades existentes no modelo decorrente da integração, as quais estavam ocultas quando da elaboração de um esquema individual de um banco de dados mas, no momento da integração, tornam-se importantes devido aos diferentes conceitos dos esquemas.

Novas propriedades do esquema integrado podem surgir nesta fase, decorrentes de uma análise mais profunda deste esquema.

A análise dos ciclos redundantes é realizada no momento da integração de dois esquemas, onde podem ser originados ciclos que mais tarde se tornarão potenciais relacionamentos redundantes.

Deve-se atentar para este ponto solucionando-se este problema neste passo. Com o objetivo de conferir as redundâncias, os caminhos devem ser percorridos aos pares, para reconstruir os relacionamentos entre conceitos terminais dos caminhos.

Podem ser distinguidos três tipos de ciclos:

- Nenhum relacionamento pode ser eliminado sem perda de informação. Neste caso, uma reestruturação sem perda de informação pode recair no caso seguinte.
- Somente um relacionamento pode ser eliminado.
- Mais de um relacionamento pode ser eliminado. Este caso recairá em uma tarefa a ser executada no modelo físico que escolherá o caminho mais conveniente a ser seguido, do ponto de vista da *performance* do sistema.

A última tarefa a ser cumprida na fase de reestruturação final é a reestruturação do esquema. Esta tarefa está relacionada com aumentar a clareza e a expressividade do esquema, além de expressar, tanto quanto seja possível, pelo

modelo do esquema integrado, todas as restrições de integridade representadas nos modelos originais.

Como esta é a fase final do processo de integração, a mesma deve ser realizada com extrema qualidade, visto que somente nesta fase consegue-se obter uma visão global do universo do discurso da aplicação.

Uma característica comum às fases de análise dos conflitos e de reestruturação é que ambas requerem uma análise complexa dos esquemas para que sejam detectados grupos de conceitos a serem modificados e sejam escolhidas as devidas correspondências nas representações.

Esta metodologia sugere algumas indicações a serem denotadas:

- Anomalias de múltiplos nomes: situações em que muitos nomes/sinônimos correspondem a um mesmo conceito em um esquema e a diferentes conceitos em outro esquema.
- Inconsistência de tipo.
- Conceitos comuns: ocorre quando conceitos distintos têm muitas propriedades e restrições em comum em dois esquemas.
- Disparidade de conceitos: ocorre quando conceitos com o mesmo nome têm diferentes propriedades e restrições em dois esquemas.
- Ocorrência de novos ciclos (no esquema integrado), o que corresponde a geração de novos ciclos depois do passo de mesclagem, dos dois esquemas, no esquema integrado.

Essas indicações devem ser analisadas em conjunto com os analistas e *designers* do banco de dados, pois podem levar à detecção de conflitos nos esquemas, ou nas propriedades dos inter-esquemas, esquemas intermediários que levarão ao esquema integrado final.

# 7. O PROTÓTIPO

A UFPR, a exemplo de vários outros órgãos, possui bases de dados distribuídas em várias pró-reitorias e departamentos distintos. Estas bases estão estruturadas em plataformas e gerenciadores de bancos de dados não compatíveis e isolados, o que inviabiliza qualquer tratamento que envolva o seu compartilhamento.

As soluções adotadas são também associadas a tratamentos e acessos locais ou particulares a cada departamento e/ou pró-reitoria, os quais iremos chamar de áreas.

Os bancos de dados são distribuídos e heterogêneos, visto que estas áreas utilizam diferentes sistemas de bancos de dados, os quais são mantidos e operados diferindo em muitos aspectos que vão desde a forma de armazenamento dos dados até sua estrutura e semântica.

Este trabalho está contribuindo no processo de integração dos seguintes sistemas:

- Sistema de Automação Universitária, que se fundamenta no modelo hierárquico e utiliza, como SGBD, o DMS-II. Este sistema possui três módulos: Administração e Pessoal (SAU-02), Controle Acadêmico (SAU-05) e Controle de Protocolo (SAU-07). Como estes módulos possuem caraterísticas específicas, os mesmos podem ser tratados como sistemas individuais.
- Sistema de Gerenciamento de Usuários, Sistema de Controle de Pesquisa e Pós-Graduação e Sistema de Bibliotecas, que são baseados no modelo relacional e utilizam os seguintes componentes de *software*, MICROSOFT SQL SERVER, ORACLE e ACCESS, respectivamente.
- Repositório de Eventos Clínicos do Hospital de Clínicas, vinculado a UFPR, que se baseia no modelo orientado a objeto e utiliza o SGBD ORACLE versão 8i.

Este processo de integração gerará um esquema integrado que irá contribuir para o desenvolvimento de um *data warehouse*, ou seja, uma base de dados destinada a fornecer informações sobre os sistemas de aplicação envolvidos neste trabalho.

Como já comentamos em capítulos anteriores, o processo de integração demanda tempo de realização e análise dos sistemas existentes, sendo necessário e imprescindível a participação dos analistas responsáveis por cada sistema para que possamos compreender os objetos existentes em cada banco de dados, assim como a semântica dos dados.

O desenvolvimento do protótipo está sendo realizado por dois programadores e um integrador; sendo assim, devido ao volume de dados existente, o protótipo, em sua primeira versão, abrangeu somente os sub esquemas dos seguintes sistemas:

- Sistema de Automação Universitária - Administração e Pessoal (SAU-02).
- Sistema de Automação Universitária - Controle Acadêmico (SAU-05).
- Sistema de Bibliotecas (SIBI).
- Sistema de Controle de Pesquisa e Pós-Graduação (PRPPG).

O fato de trabalharmos no protótipo, em sua primeira versão, com apenas quatro sistemas e com os sub esquemas destes sistemas não anula o efeito de prova deste trabalho, visto que nosso objetivo é mostrar que as metodologias citadas podem ser aplicadas em casos reais.

Para a UFPR este fato não gera nenhuma alteração no produto final, que seria a integração de todos os sistemas citados, pois, desta maneira, este produto está sendo implementado em etapas.

Com o desenvolvimento do protótipo, em sua primeira versão, demostramos a aplicação das metodologias e completamos o ciclo do processo de integração de bancos de dados que abrange as fases de pré integração, identificação das correspondências, verificação da conformidade dos esquemas e integração.

Iremos descrever, nas seções 7.1, 7.2, 7.3 e 7.4 as funcionalidades destes sistemas, assim como os principais objetos utilizados no desenvolvimento do protótipo. Na seção 7.5 descreveremos o processo de integração.

## 7.1. SISTEMA DE AUTOMAÇÃO UNIVERSITÁRIA - ADMINISTRAÇÃO E PESSOAL (SAU-02)

O Sistema de Automação Universitária - Administração e Pessoal (SAU-02) foi desenvolvido para suportar as atividades de gerenciamento de recursos humanos

da UFPR, contendo todas as transações inerentes à legislação para servidores CLT, estatutários e regime jurídico único.

O SAU-02 permite o controle de toda a área de recursos humanos da UFPR, com informações em tempo real e à disposição dos próprios servidores.

Operacionalmente, comporta-se como um sistema de controle, implementando todos os atos administrativos, tais como: provimento, ascensão, alterações contratuais, histórico funcional, averbações, até a aposentadoria dos funcionários e professores da UFPR.

O SAU-02 contém as transações de pesquisa de dados referentes à situação funcional dos funcionários e professores da UFPR, tais como: cargo, função, anuênios, insalubridade, datas de admissão e demissão, faltas, licenças, afastamentos e contagem de tempo de serviço.

Os secretários são responsáveis pelo cadastramento de freqüência mensal, dados pessoais e férias, acessando os dados referentes aos servidores lotados na sua unidade.

Este sistema utiliza o SGBD DMS-II, que é fundamentado no modelo hierárquico.

As dificuldades encontradas durante as atividades do processo de integração que envolveram o SAU-02 recaem sobre o fato de que este sistema foi desenvolvido há 20 anos e, por esta razão, as pessoas responsáveis, atualmente, pela manutenção do mesmo, não possuem um conhecimento total do sistema.

Outros fatores que acarretaram uma maior dificuldade na compreensão do esquema do SAU-02 foram a falta de documentação relativa aos objetos do banco de dados e a dependência da equipe de manutenção deste sistema, com relação a compreensão dos dados.

O esquema inicial do SAU-02 pode ser visualizado através da figura 28.

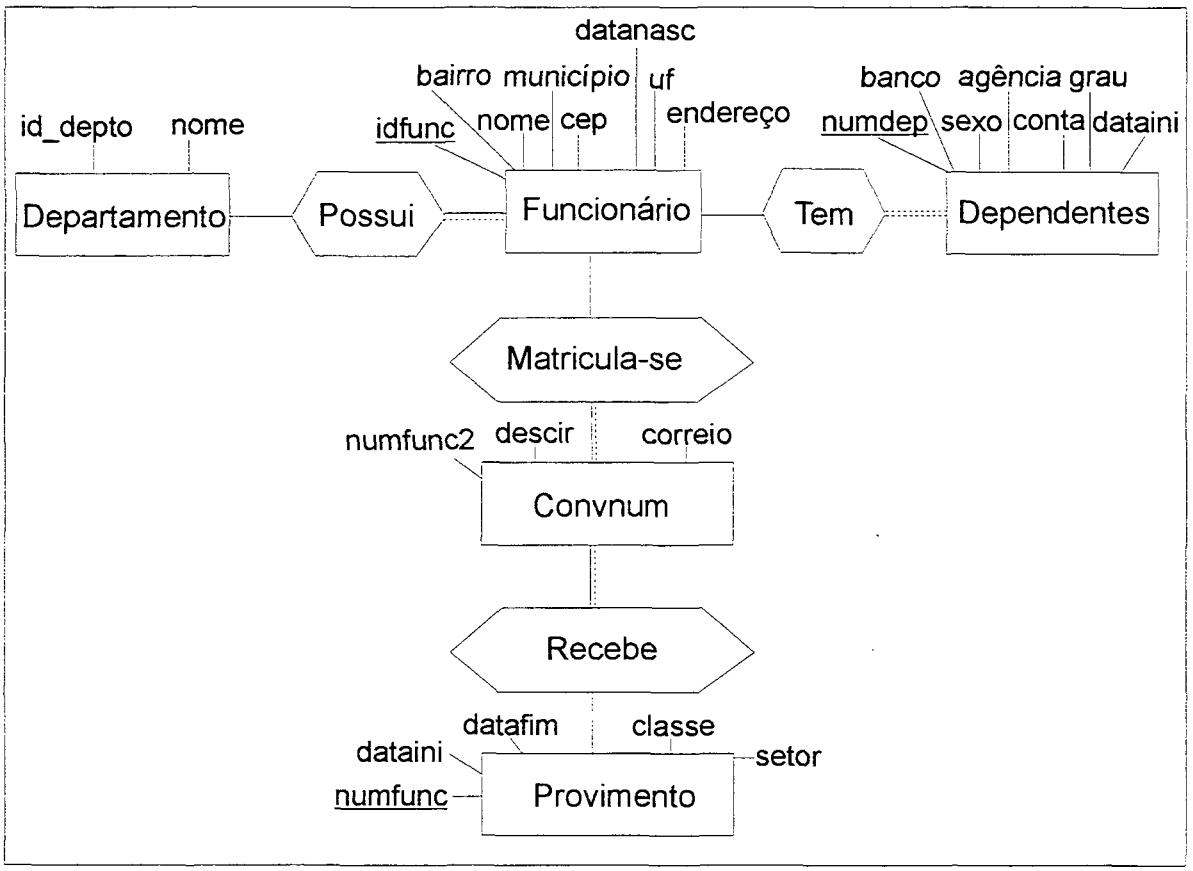


FIGURA 28: Diagrama dos principais objetos do SAU-02.

Iremos descrever, a seguir, as principais tabelas deste sistema.

**Tabela DPFUNC**

A tabela DPFUNC contém os atributos dos funcionários como nome, matrícula, fone, data nascimento e endereço. Esta tabela armazena características tanto de funcionários como de professores.

No banco de dados não há distinção entre funcionários e professores. Não há tabelas específicas para estas definições de objetos que possuem características específicas no mundo real.

No sistema os funcionários são chamados de PTA ou pessoal técnico administrativo. A distinção entre funcionários e professores é feita através da vantagem que cada um possui. Estes dados estão armazenados na tabela VANTAGEM.

Chave Primária: matrícula principal do funcionário (numfunc_dp).

**Tabela CONVNUM**

A tabela CONVNUM possui dados referentes ao cadastro de matrículas dos funcionários. Nesta tabela podem existir duas matrículas diferentes, relativas ao

atributos numfunc1 e numfunc2. A matrícula numfunc1 é chamada número de matrícula principal e a matrícula numfunc2 refere-se a um número alternativo.

A existência destes dois números de matrícula decorreu do fato de que uma pessoa pode exercer dois ou mais cargos ao mesmo tempo. Por exemplo, existem pessoas que, além de desempenharem as funções de professor, são, também, médicos no hospital de clínicas, sendo assim as mesmas recebem salário pelas duas funções que exercitam.

Nesse caso a pessoa terá dois registros na tabela CONVNUM. Um registro armazenará a matrícula de professor e o outro a de médico. O atributo numfunc1 é utilizado como chave de acesso aos dados pessoais do funcionário (referente as tabelas DPFUNC e DEPENDENTES), sendo que o atributo numfunc2 é utilizado para acessar os demais dados do funcionário (tabelas PROVIMENTO e VANTAGEM).

De acordo com o exemplo citado acima, este funcionário possuirá os seguintes dados gravados na tabela CONVNUM:

1° registro: numfunc1-CONVNM = 123456 , numfunc2-CONVNUM = 123456

2° registro: numfunc1-CONVNM = 123456 , numfunc2-CONVNUM = 667788

Como podemos visualizar pelo exemplo acima, o atributo numfunc1 possuirá sempre o mesmo valor, sendo que o atributo numfunc2 irá armazenar um valor diferente para cada função que a pessoa exerça.

Isto ocorre porque uma pessoa deve possuir somente um registro na tabela DPFUNC, que se refere ao atributo numfunc1 da tabela CONVNUM. O mesmo não acontece para o atributo numfunc2, pois este se relaciona com as tabelas PROVIMENTO e VANTAGEM, que possuem valores distintos de acordo com o cargo que a pessoa exerça.

Chave Primária: matrícula alternativa do funcionário (numfunc2_convnum).

Chave Estrangeira: matrícula principal do funcionário (numfunc1_convnum).

**Tabela DEPENDENTES**

Esta tabela armazena informações dos dependentes dos funcionários, que podem ser seus filhos, marido ou esposa e pessoas que se enquadrem nesta categoria, de acordo com o regimento da UFPR. Essas informações referem-se à data de nascimento, sexo e nome dos dependentes.

Chaves Primárias: nome e data-início.

Chave Estrangeira: matrícula principal do funcionário (numfunc_depend).

**Tabela PROVIMENTO**

Esta tabela armazena informações sobre o cargo ou a função que um funcionário exerce na UFPR.

Como provimento refere-se ao preenchimento de um cargo ou ofício público, por nomeação, promoção, transferência, reintegração, readmissão, aproveitamento ou reversão. Quando um funcionário se aposenta, cria-se um novo provimento para esta pessoa, assim como, cada vez que um funcionário muda de cargo (referências na tabela CLASSE) um provimento é fechado e um novo provimento é aberto para este funcionário.

Chaves Primárias: data-início e numreg.

Chave Estrangeira: matrícula alternativa do funcionário (numfunc_prov).

**Tabela CLASSE**

Contém a descrição de todos os cargos que um funcionário pode exercer na UFPR, como, por exemplo: Analista de Sistemas, DOC FNS SUP ADJ (docente), entre outros. Os principais atributos desta tabela são nome da classe, grupo e tipo.

Chaves Primárias: número da classe (código), data-início e data-fim.

**Tabela QUADRO**

Esta tabela contém a estrutura de cargos da UFPR, armazenando informações referentes ao número máximo de funcionários de uma classe, que podem trabalhar em um determinado setor, e quantos funcionários trabalham, atualmente, em cada setor.

Por exemplo: quando um analista de sistemas deseja saber se existe uma vaga referente ao seu cargo em outro setor, o sistema busca esta informação na tabela QUADRO e retorna o número de vagas, relativas ao cargo analista de sistemas, que existem em cada setor e o número de vagas preenchidas. Assim, subtraindo-se o número de vagas existentes, do número de vagas preenchidas, o analista de sistemas consegue saber se existe, ou não, uma vaga livre em outro setor.

Chave Primária: data-início.

Chaves Estrangeiras: setor, classe e numreg.

**Tabela SETOR**

A tabela SETOR diz respeito aos dados relativos aos departamentos. Contém a descrição de todos os setores existentes na UFPR, como, por exemplo, CCE – Centro de Computação Eletrônica, Reitoria, entre outros.

Chave Primária: sigla.

**Tabela CONTAGEMANT**

Esta tabela armazena informações referentes ao tempo que um funcionário exerce em cada cargo que possui na UFPR.

**Tabela VANTAGEM**

Esta tabela mantém os dados referentes a todas as vantagens que um funcionário possui.

A tabela VANTAGEM trata de maneira diferente os funcionários que exercem o cargo de professor. Por exemplo, sabemos que um funcionário é docente através do atributo que se refere ao tipo da vantagem, tipo-vant, que é populado com o valor "REG TRAB", e através do atributo cond-vant.

O atributo cond-vant, que possui um formato caracter com sete posições, diferencia um professor de um funcionário técnico administrativo, através das três primeiras posições deste atributo, que possuem o valor "DOC", para docentes, ou "ADM" para PTA (pessoal técnico administrativo).

As três posições finais do atributo cond-vant demonstram se o funcionário está ativo, inativo (aposentado) ou falecido (pensionista). No caso de ativo, o atributo cond-vant demonstra, ainda, se o funcionário tem contrato de trabalho RJU (regime jurídico único) ou CLT (consolidação das leis trabalhistas).

Por exemplo: Um professor ativo possui, no atributo tipo-vant, o valor "REG TRAB", e, no atributo cond-vant, os valores "DOC RJU", se possui contrato de trabalho RJU ou "DOC CLT", se o mesmo for um professor CLT. Um PTA aposentado tem o atributo cond-vant populado com o valor "ADM INA".

A tabela VANTAGEM mantém um histórico de vantagens dos funcionários, desta maneira, o registro mais recente é o que possui o atributo datainí-vant com o valor mais próximo da data em que está sendo realizada a consulta ao banco de dados.

Chave Primária: código da vantagem.

Chave Estrangeira: matrícula alternativa do funcionário (numfunc_prov).

## 7.2. SISTEMA DE AUTOMAÇÃO UNIVERSITÁRIA - CONTROLE ACADÊMICO (SAU-05)

O Sistema de Automação Universitária - Controle Acadêmico (SAU-05) foi desenvolvido para gerenciar as atividades de ensino da UFPR.

Trata-se de um sistema multiusuário, no qual as atividades administrativas referentes ao ensino (controle acadêmico) são efetuadas através de terminais de uma rede de teleprocessamento.

O SAU-05 contém o cadastro de dados pessoais dos alunos e mantém um histórico dos mesmos, incluindo o desempenho no vestibular, cursos em que está ou esteve matriculado, as notas e os créditos obtidos.

O SAU-05 controla os cursos oferecidos, efetua matrícula *on line*, controla a grade curricular e pré-requisitos, entre outras funções. As coordenações dos cursos são responsáveis, principalmente, pelos dados pessoais dos alunos de seu curso, matrícula e lançamento de créditos. Os departamentos didáticos são responsáveis pela abertura e consolidação de turmas, assim como pelo controle das notas das disciplinas do seu departamento.

Este sistema, assim como o SAU-02, utiliza o SGBD DMS-II, que é fundamentado no modelo hierárquico.

As dificuldades encontradas durante as atividades do processo de integração foram as mesmas que nos deparamos com o SAU-02: carência de conhecimento dos dados por parte da equipe responsável pelo sistema, falta de documentação dos objetos do banco de dados e dependência da equipe de manutenção deste sistema, com relação à compreensão dos dados.

Os principais objetos do esquema do SAU-05 podem ser visualizados através da figura 29.

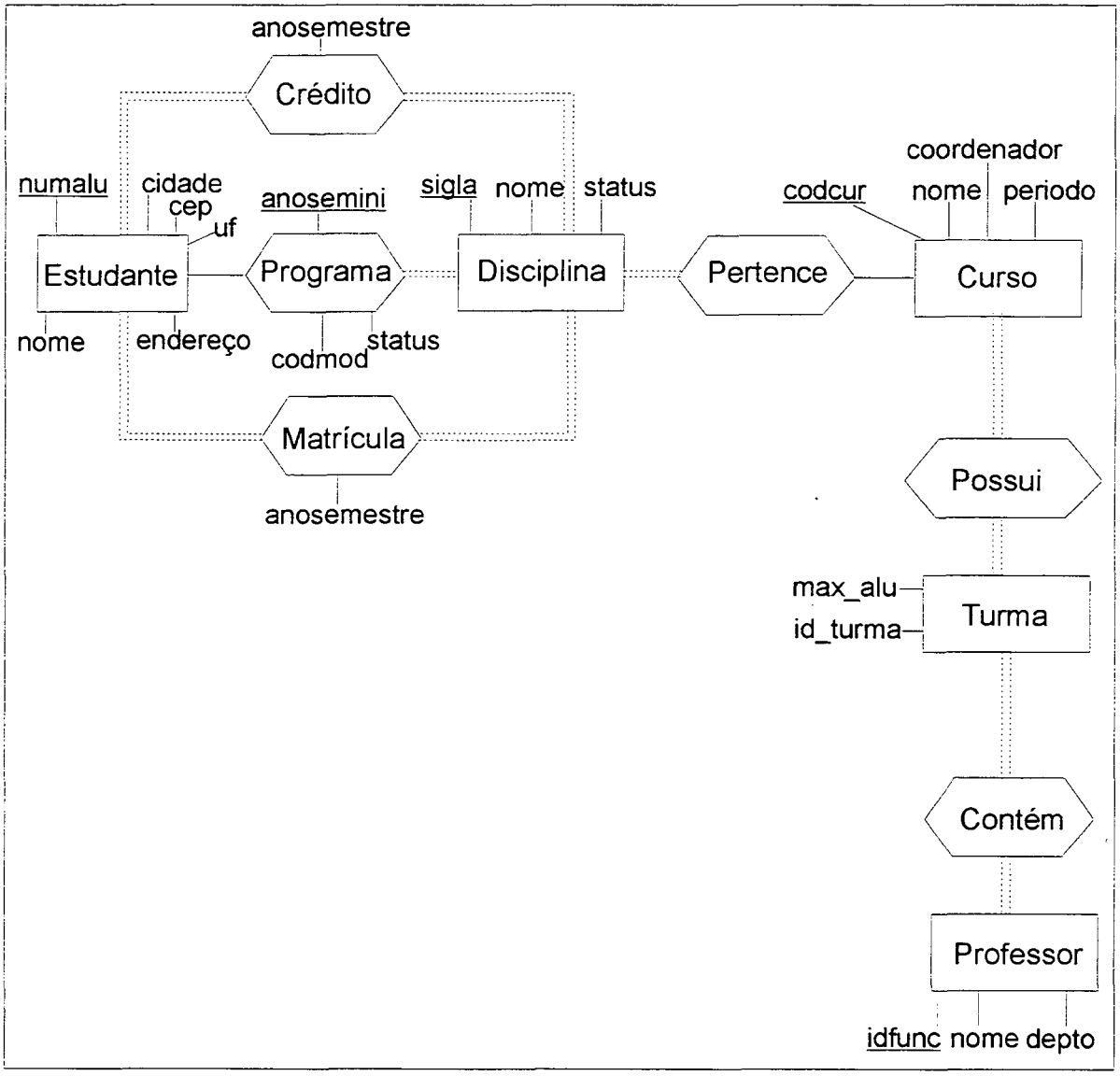


FIGURA 29: Diagrama dos principais objetos do SAU-05.

Iremos descrever, a seguir, as principais tabelas deste sistema.

**Tabela ALUNO**

Esta tabela contém os dados relativos tanto a alunos de graduação como os de pós graduação. Os atributos principais são: número de matrícula, nome, endereço, cidade, uf e cep.

Chave Primária: matrícula do aluno (numalu-alu)

**Tabela PROG**

É a tabela principal deste sistema, todo o aluno que ingressa na UFPR faz parte de um programa. Um programa estabelece relações entre um aluno e um curso.

Esta tabela possui atributos como status do programa e escore no vestibular. O status no programa pode ser ativo, cancelado ou concluído. Não existe status "trancado". Quando um aluno "tranca um curso", o status do programa continua ativo porém suas matrículas são canceladas

Chave Primária: ano semestre de início (anosemini-prog).

Chaves Estrangeiras: matrícula do aluno (numalu-prog), código do curso (codcur-prog).

**Tabela CREDITO**

Possui informações referentes a todas as disciplinas já cursadas por um aluno, nas quais ele tenha sido aprovado. As disciplinas em que o aluno reprovou ou as quais tenham sido trancadas não fazem parte desta tabela.

Quando um aluno recebe um crédito por equivalência de disciplina, ou seja, quando o mesmo recebe um crédito em uma disciplina sem cursá-la, por já ter realizado tal disciplina em outro curso ou instituição, a informação é armazenada na tabela CRÉDITO. Por este motivo, a tabela CRÉDITO não equivale a um subconjunto da tabela MATRÍCULA.

Chaves Primárias: ano semestre (anosem-cred).

Chaves Estrangeiras: matrícula do aluno (numalu-cred), código do curso (codcur-cred), ano semestre início (anosemini-cred), disciplina do crédito (discip-cred).

**Tabela CURSO**

Esta tabela possui a descrição dos cursos, como período do curso, coordenador e departamento.

Chaves Primárias: código do curso (codcur-curso).

**Tabela MODALIDADE**

Esta tabela é uma especialização da tabela CURSO, onde são armazenadas as informações referentes aos programas de especialização dos cursos.

Por exemplo: o curso de Engenharia Elétrica possui, como modalidades, Telecomunicações, Elétrica e Eletricidade; o curso de Informática possui, como modalidade, Informática, pois só possui esta especialização.

Chave Primária: código da modalidade (codmod-mod).

Chave Estrangeira: código do curso (codcur-mod).

**Tabela CURRICULO**

Esta tabela armazena informações referentes à lei que rege uma modalidade de determinado ano e semestre.

Por exemplo, a modalidade Informática do ano de 1996 é regida por um currículo x, e a modalidade Informática de 1999 é regida por um currículo y.

O currículo indica o número de créditos, a quantidade de horas de laboratório, o número de horas de estágio e o prazo máximo que o aluno possui para concluir tal modalidade.

A tabela CURRÍCULO armazena todas as diretrizes que ditam o funcionamento de uma modalidade.

Chave Primária: currículo (curr-curr).

Chaves Estrangeiras: código do curso (codcur-curr), código da modalidade (codmod-curr).

**Tabela MATRICULA**

Esta tabela armazena todas as matrículas efetuadas por um aluno em uma disciplina.

Estas matrículas podem possuir o status de aprovado, trancado, reprovado por nota, reprovado por frequência.

Esta tabela está ligada ao programa que o aluno cumpre na UFPR.

Chave Primária: ano semestre (anosem-matric).

Chaves Estrangeiras: matrícula do aluno (numalu-matric) , código do curso (codcur-matric) , ano semestre início (anosemini-matric), disciplina (discip-matric).

**Tabela GRADE**

Esta tabela contém informações sobre o conjunto de disciplinas de um determinado currículo. Cada grade está relacionada a um currículo.

Chaves Primárias: sem grade (sem-grade).

Chaves Estrangeiras: código do curso (codcur-grade), código da modalidade (codmod-grade), currículo (curr-grade), disciplina (discip-grade).

**Tabela TURMA**

Esta tabela contém a descrição de uma disciplina e um ano semestre. Seus atributos principais são total de alunos, máximo de alunos e status da turma.

Chaves Primárias: turma (turma-turma), ano semestre (anosem-turma)

Chave Estrangeira: sigla da disciplina (discip-turma).

**Tabela DISCIPLINA**

Nesta tabela são armazenadas todas as características de uma determinada disciplina, como, por exemplo, nome da disciplina, departamento a que ela pertence, duração, nota mínima para aprovação , freqüência mínima e quantidade de créditos.

Chave Primária: sigla da disciplina (sigla-discip).

**Tabela EMENTA**

Esta tabela contém a ementa de uma determinada disciplina, ou seja, contém o resumo do conteúdo programático de uma disiciplina.

Chave Primária: código da ementa (pag-ementa).

Chave Estrangeira: sigla da disciplina (sigla-ementa). .

**Tabela VESTIB**

É uma tabela que contém todas as provas e todas as notas que um determinado programa (aluno) realizou e alcançou no vestibular.

Chave Primária: prova (prova-vestib).

Chave Estrangeira: matrícula do aluno (numalu-vestib), código do curso (codcur-vestib), ano do primeiro semestre cursado (anosemini-vestib).

**Tabela TRANCPROG**

Esta tabela indica o modo de cancelamento de um programa.

Chave Primária: ano do semestre (anosem-tp).

Chaves Estrangeiras: matrícula do aluno (numalu-tp), código do curso (codcurs-tp), ano do primeiro semestre cursado (anosemini-tp).

**Tabela RESERVA DE TURMA**

Esta tabela é populada com dados que demonstram quantas vagas de um determinado curso foram reservadas para uma determinada turma.

Chave Primária: código da reserva (codres).

Chaves Estrangeiras: disciplina (discip-turma), ano do semestre (anosem-rt), turma (turma-rt), código do curso (codcurs-rt).

## 7.3. SISTEMA DE BIBLIOTECAS (SIBI)

O Sistema de Bibliotecas (SIBI) da UFPR é constituído pela Biblioteca Central (Sede Administrativa ) e as seguintes sub unidades, distribuídas geograficamente pelos campi da universidade, de acordo com os setores a que servem. São elas:

- Biblioteca Central.
- Biblioteca de Educação Física.
- Biblioteca de Ciências Humanas e Educação.
- Biblioteca de Ciências da Saúde / Sede Botânico.
- Biblioteca de Ciências Sociais Aplicadas.
- Biblioteca do Centro de Estudos do Mar.
- Biblioteca do Campi de Palotina.
- Biblioteca de Ciências Agrárias.
- Biblioteca de Ciências Biológicas.
- Biblioteca de Ciência e Tecnologia.
- Biblioteca de Ciências Jurídicas.
- Biblioteca de Ciências da Saúde.
- Biblioteca da Escola Técnica.
- Biblioteca do MAEP.

O SIBI controla os usuários que utilizam o serviço das bibliotecas citadas acima e possui um número aproximado de 23.000 (vinte e três mil) usuários cadastrados.

Este sistema é baseado no modelo relacional e utiliza o *software* MICROSOFT ACCESS.

Por ser um sistema que possui somente uma tabela e sendo o MS ACCESS um *software* que possui a capacidade de gerar uma boa documentação, não tivemos dificuldades no entendimento do mesmo. O esquema do SIBI pode ser visualizado através da figura 30.

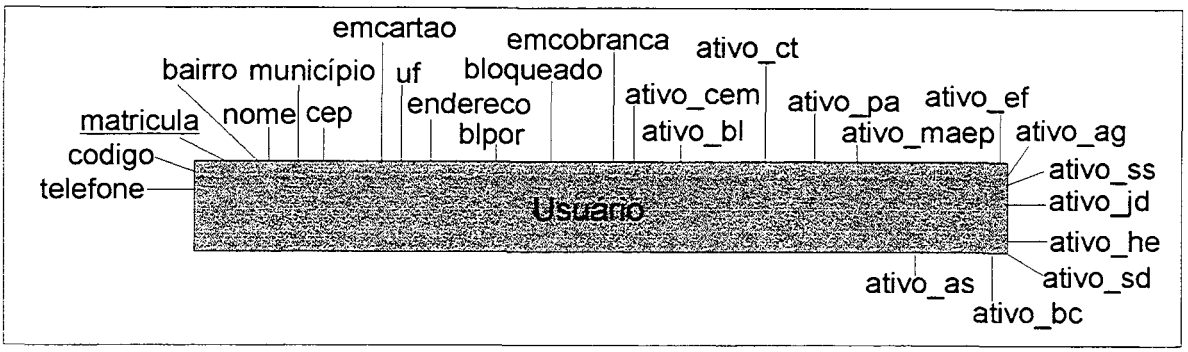


FIGURA 30: Diagrama dos principais objetos do SIBI.

Como podemos verificar através da figura 30, o único objeto que forma este esquema é a entidade Usuário, que descreve as características de todos os usuários assim como o seu *status* perante a biblioteca.

Um usuário possui um vínculo com qualquer 14 bibliotecas da UFPR. O usuário cadastrado no sistema tem sua condição atualizada automaticamente para não ativo a partir do início de cada exercício, ou seja, de cada ano letivo. Desde a implantação do SIBI, em 1998, nenhum registro foi eliminado.

Atributos da tabela Usuário: codigo, matricula, nome, endereco, bairro, município, uf, cep, telefone, emcartao, emcobranca, ativo_ag, ativo_bc, ativo_bl, ativo_cem, ativo_ct, ativo_ef, ativo_et, ativo_he, ativo_jd, ativo_pa, ativo_maep, ativo_as, ativo_sd, ativo_ss, bloqueado, blpor.

Chave Primária: número da matrícula do estudante na UFPR (estudante) ou número de matrícula SIAPE (funcionário).

Como os responsáveis por este sistema nos forneceram uma especificação de cada atributo desta tabela, iremos relatar, nos próximos parágrafos uma breve descrição do significado dos valores que cada atributo armazena assim como seus domínios, ou seja, o conjunto de valores possíveis para cada atributo.

**Atributo codigo**

Indica o número de identificação do usuário no sistema de bibliotecas.

Domínio: numérico.

**Atributo matrícula**

Armazena o número de matrícula do usuário na UFPR: no caso de estudante, este número é o seu número de matrícula do SAL-05 e no caso de professor ou funcionário, o seu número de matrícula SIAPE.

Domínio: caracter.

**Atributo nome**

Grava o nome do usuário.

Domínio: caracter.

**Atributo endereco**

Mantém o endereço do usuário.

Domínio: caracter.

**Atributo bairro**

Armazena o complemento do endereço do usuário.

Domínio: caracter.

**Atributo município**

Descreve a cidade onde o usuário reside.

Domínio: caracter.

**Atributo uf**

Indica o estado onde o usuário reside.

Domínio: caracter.

**Atributo cep**

Guarda o código de endereçamento postal do endereço do usuário.

Domínio: caracter.

**Atributo telefone**

Contém o número do telefone do usuário.

Domínio: caracter.

**Atributo categoria**

Este atributo corresponde a um indicador da categoria de usuário de biblioteca, à qual pertence o usuário. É destinado a balizar o limite e o prazo de empréstimo de livros observados por cada biblioteca. As categorias podem ser:

1. GRADUAÇÃO
2. DOUTORADO
3. MESTRADO
4. ESPECIALIZAÇÃO
5. PROFESSOR
6. FUNCIONÁRIO
7. OUTROS
8. ALUNO ESCOLA TÉCNICA

Domínio: numérico.

**Atributo emcartao**

É um atributo indicador utilizado para emissão de etiqueta de identificação para a carteira do usuário.

Domínio: verdadeiro ou falso.

**Atributo emcobranca**

É, também, um atributo indicador utilizado para emissão de carta cobrança padronizada e personalizada para o usuário.

Domínio: verdadeiro ou falso.

**Atributos ativo_ag, ativo_bc, ativo_bl, ativo_cem, ativo_ct, ativo_ef, ativo_et, ativo_he, ativo_jd, ativo_pa, ativo_maep, ativo_as, ativo_sd, ativo_ss**

Estes atributos indicam se o usuário é ativo em uma determinada biblioteca, de acordo com as siglas a seguir:

- ag - agrárias.
- bc - central.
- bl - biologia.
- cem - centro de estudos do mar.
- ct - ciência e tecnologia.
- ef - farmácia.
- et - escola técnica.
- he - humanas e educação.
- jd - jurídica.
- pa - Palotina.
- maep - museu Paranaguá.
- as - sociais aplicadas.
- sd - saúde.
- ss – sub sede ciência da saúde.

Domínio: verdadeiro ou falso.

**Atributo bloqueado**

Este também é uma atributo indicador utilizado para emissão de relatório de usuários bloqueados no sistema de bibliotecas.

Domínio: verdadeiro ou falso.

**Atributo blpor**

Armazena o código do funcionário responsável pelo bloqueio do usuário.

Domínio: caracter.

## 7.4. SISTEMA DE CONTROLE DE PESQUISA E PÓS-GRADUAÇÃO (PRPPG)

O Sistema de Controle de Pesquisa e Pós-Graduação (PRPPG) é uma base de dados que contém informações relevantes sobre as pesquisas desenvolvidas na UFPR ou que tenham sido realizadas com a sua participação oficial. Para ser incluída no sistema, a pesquisa deverá ser previamente aprovada pelo departamento a ela relacionado.

As informações relativas às pesquisas são as seguintes: número identificador da pesquisa - gerado pelo próprio programa, título, início e término previsto, fase em que se encontra, área do conhecimento relacionada à pesquisa e tipo da pesquisa, as quais podem ser iniciação científica, tese de mestrado ou doutorado ou pesquisa pura.

Este sistema também armazena dados do pesquisador, o qual pode ser um docente ou técnico da UFPR, ou um professor visitante.

O sistema da PRPPG armazena também dados da equipe da pesquisa, ou seja, os colaboradores, os quais podem ser alunos de graduação ou funcionários da UFPR.

O sistema da PRPPG emite relatórios por departamento, a fim de que o responsável pelo departamento tenha controle e conhecimento das pesquisas realizadas.

Este sistema utiliza o SGBD ORACLE, que é baseado no modelo relacional.

As dificuldades encontradas durante as atividades do processo de integração do sistema da PRPPG foram menores do que os obstáculos que tivemos que transpor com relação aos sistemas SAU-02 e SAU-05, citados nas seções anteriores.

Os diferenciais que nos fizeram ter uma maior compreensão dos dados do sistema da PRPPG foram os seguintes:

- Este sistema está em fase de desenvolvimento, portanto a equipe de trabalho possui um amplo conhecimento dos dados.
- Por ser um sistema novo, utiliza ferramentas de desenvolvimento mais atuais, que possibilitam uma maior documentação dos dados. Este sistema faz uso da ferramenta CASE, DESIGNER 200, da ORACLE, que gera uma documentação bastante precisa do banco de dados, permitindo, assim, uma análise mais confiável do esquema deste sistema.
- A equipe de desenvolvimento teve uma maior participação no processo, facilitando o entendimento da semântica dos dados.

O esquema dos principais objetos do Sistema de Controle de Pesquisa e Pós-graduação (PRPPG) pode ser visualizado através da figura 31.

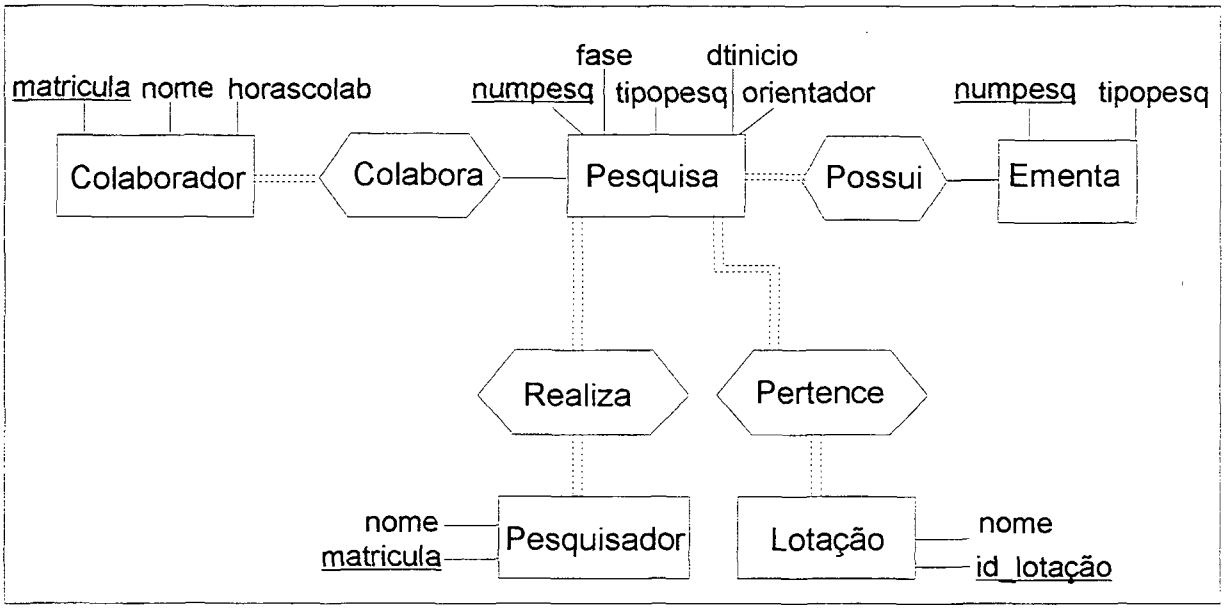


FIGURA 31: Diagrama dos principais objetos do Sistema da PRPPG.

Iremos descrever, a seguir, as principais tabelas deste sistema.

**Tabela COLABORADOR**

Nesta tabela são armazenadas informações sobre os colaboradores que participam em uma pesquisa, que podem ser alunos da graduação ou funcionários da UFPR.

Os atributos desta tabela são tipo da colaboração, horas de colaboração, data da colaboração, nome do colaborador e matrícula do colaborador.

Chave Primária: matrícula do colaborador.

**Tabela PESQUISA**

Esta tabela mantém os dados relativos a uma pesquisa, tais como, data de início e fim da pesquisa, horas de pesquisa, fase da pesquisa, tipo da pesquisa e data de aprovação da pesquisa.

Chave Primária: número da pesquisa.

Chaves Estrangeiras: matrícula do professor, lotação e área.

**Tabela PESQUISADOR**

Esta tabela armazena as informações de um pesquisador, tais como, matrícula, vínculo, nome e departamento. Um pesquisador pode realizar várias pesquisas diferentes. Um pesquisador pode ser um professor ou um aluno de pós graduação.

Chave Primária: matrícula do professor ou do aluno de pós graduação.

**Tabela LOTAÇÃO**

Esta tabela é populada com os dados dos departamentos da UFPR, tais como, nome do departamento e sua sigla.

Chave Primária: sigla da lotação.

**Tabela EMENTA**

Esta tabela armazena informações sobre o tema da pesquisa. Uma pesquisa pode ter várias ementas. Os atributos desta tabela são número da pesquisa, nome da ementa e tipo da pesquisa.

Chaves Primárias: número da pesquisa.

Chaves Estrangeiras: ementa.

**Tabela FONTES**

Esta tabela mantém informações sobre as fontes de financiamento da pesquisa, como os órgãos de financiamento.

Chave Primária: fonte.

**Tabela RELATÓRIO**

Esta tabela armazena informações sobre uma pesquisa. Possui atributos como fase, custo, período e chefe do departamento da pesquisa. Uma pesquisa pode possuir vários relatórios.

Chave Primária: número da pesquisa.

**Tabela FONTEREL**

Esta tabela associa uma pesquisa a suas fontes de financiamento. Uma pesquisa pode possuir várias fontes de financiamento e uma fonte de financiamento pode estar em várias pesquisas. Possui atributos como valor, data de referência, fonte e identificador da produção.

Chave Estrangeira: sigla da fonte.

**Tabela AREA**

É uma tabela que armazena as diversas áreas de pesquisa conforme classificação do CNPQ. Possui somente dois atributos, código da área e nome da área.

Chaves Primárias: código da área

## 7.5. SOLUÇÃO ADOTADA

Nosso protótipo envolveu o trabalho de onze analistas durante dois meses, na fase de preparo dos sistemas para a integração, além de um integrador durante seis meses e mais dois programadores com a função de implementar o acesso as quatro bases integradas.

Podemos considerar como sendo a equipe de desenvolvimento do protótipo o integrador e os programadores pois os analistas participaram somente da fase de investigação conforme iremos relatar nesta seção.

Inicialmente efetuamos uma avaliação geral dos sistemas citados nas seções anteriores e que estão envolvidos no desenvolvimento da primeira versão do protótipo, comparando os diferentes aspectos empregados, tais como, modelos de dados, gerenciadores de banco de dados, linguagens de acesso e funcionalidades dos sistemas.

Em seguida, no passo de investigação ou pré integração, modelamos todos os bancos de dados envolvidos, usando um modelo de dados comum, o ERC+. Esta fase foi inteiramente manual e crucial para a realização da integração, visto que é o alicerce principal no qual o esquema integrado irá buscar os dados.

Este passo da integração, ficou, a princípio, sob responsabilidade das equipes de desenvolvimento de cada sistema, visto que era necessário um conhecimento dos dados que cada sistema armazena e gerencia.

Todos os analistas da UFPR, responsáveis pelos sistemas envolvidos neste trabalho, foram treinados para executar as tarefas que compreendem este passo da integração da mesma maneira.

O treinamento, foi realizado durante uma semana e envolveu desde noções básicas de modelagem de dados a exercícios práticos.

Após o treinamento os analistas receberam um roteiro contendo as instruções, passo a passo, de como deveriam cumprir esta atividade, que compreendeu as seguintes tarefas:

- Modelagem do banco de dados conforme o modelo ERC+.
- Definição das instâncias dos bancos de dados.
- Descrição dos objetos dos bancos de dados, indicando os domínios, relacionamentos, chaves e limites de integridade dos dados.
- Descrição do mapeamento físico ou da linguagem de consulta dos bancos de dados.

As equipes de desenvolvimento, apesar de terem sido treinadas e sensibilizadas a desenvolver esta atividade, não conseguiram realizar todas as tarefas devido a problemas específicos de cada área, como, por exemplo, volume de trabalho cotidiano, falta de pessoal e, no caso dos sistemas mais antigos, como o SAU-02 e o SAU-05, falta de documentação e de entendimento do esquema geral dos dados.

Sendo assim, modelamos os sistemas envolvidos através de um estudo dos esquemas dos bancos de dados e consideramos, como um dos resultados finais desta atividade, a modelagem dos bancos de dados através do modelo ERC+, como mostrado nas figuras 28, 29, 30 e 31 das páginas 90, 95, 100 e 104, respectivamente.

Outro resultado deste passo da integração foi a descrição das funcionalidades das principais tabelas que compõem os sistemas que participam da primeira versão do protótipo, que estão expostas nas seções 7.1, 7.2, 7.3 e 7.4.

O fato de não termos obtido o resultado esperado neste passo da integração, por parte da equipe de desenvolvimento de cada sistema, aumentou o custo do

trabalho de integração, pois tivemos que realizar um estudo mais criterioso de cada sistema, despendendo mais tempo do que o esperado neste passo.

De posse dos modelos de dados e das descrições das tabelas dos sistemas, realizamos reuniões com os envolvidos no projeto, no caso o integrador e os programadores, com o objetivo de comparar os modelos e identificar os objetos em comum.

Iniciamos, desta maneira, um novo passo da integração, a identificação das correspondências e verificação da conformidade dos esquemas. Tratamos dos conflitos semânticos, descritivos e estruturais, além da heterogeneidade dos bancos de dados.

Nossa maior dificuldade, nesta fase, foi o fato de não existirem DBAs responsáveis por cada banco de dados, dificultando, assim, a solução dos casos de conflito, passo que prepara os esquemas para a integração.

A partir desta fase foi possível definir:

- Os objetos a serem inseridos no esquema integrado.
- O mapeamento entre os objetos a serem inseridos no esquema integrado e seus correspondentes nas bases originais.

Explanaremos, nos próximos parágrafos, estes resultados.

Identificamos, no sistema SAU-02, os seguintes objetos a serem integrados:

- Entidade Funcionário.
- Entidade Departamento.

Podemos visualizar através da figura 32, estes elementos, que estão identificados pela cor cinza.

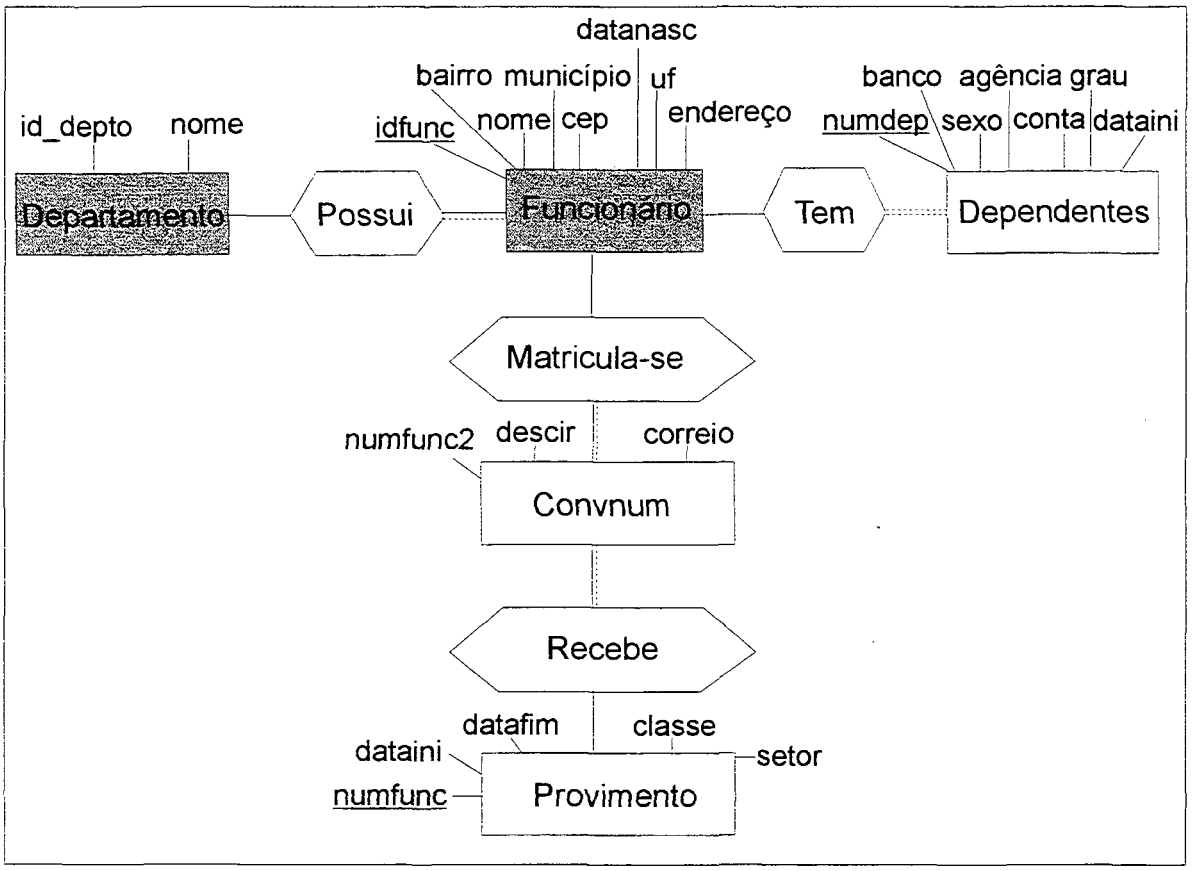


FIGURA 32: Diagrama dos principais objetos do SAU-02 realçando os objetos a serem integrados.

No sistema SAU-05 pudemos determinar os seguintes objetos a serem integrados:

- Entidade Professor.
- Atributo depto da entidade Professor.
- Entidade Estudante.

Os objetos acima podem ser identificados na figura 33, diagramados na cor cinza.

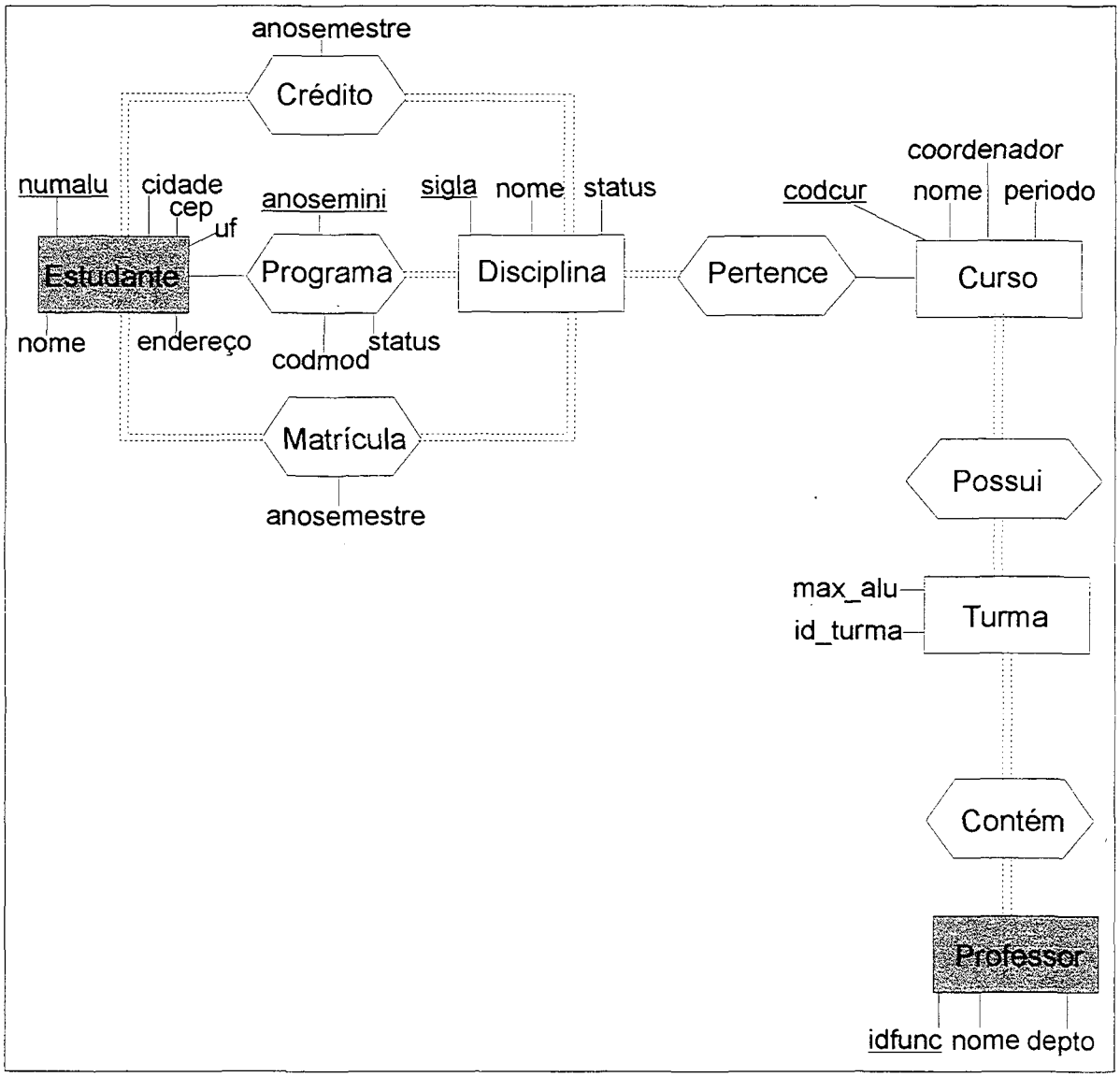


FIGURA 33: Diagrama dos principais objetos do SAU-05 realçando os objetos a serem integrados.

Como o Sistema de Biblioteca possui somente um objeto, a entidade Usuário, conforme a figura 30, da página 100, a mesma fará parte do sistema integrado.

O sistema da PRPPG originou os seguintes objetos a serem integrados:

- Entidade Colaborador.
- Entidade Pesquisador.
- Entidade Lotação.

Os objetos acima podem ser identificados na figura 34, na próxima página, identificados pela cor cinza.

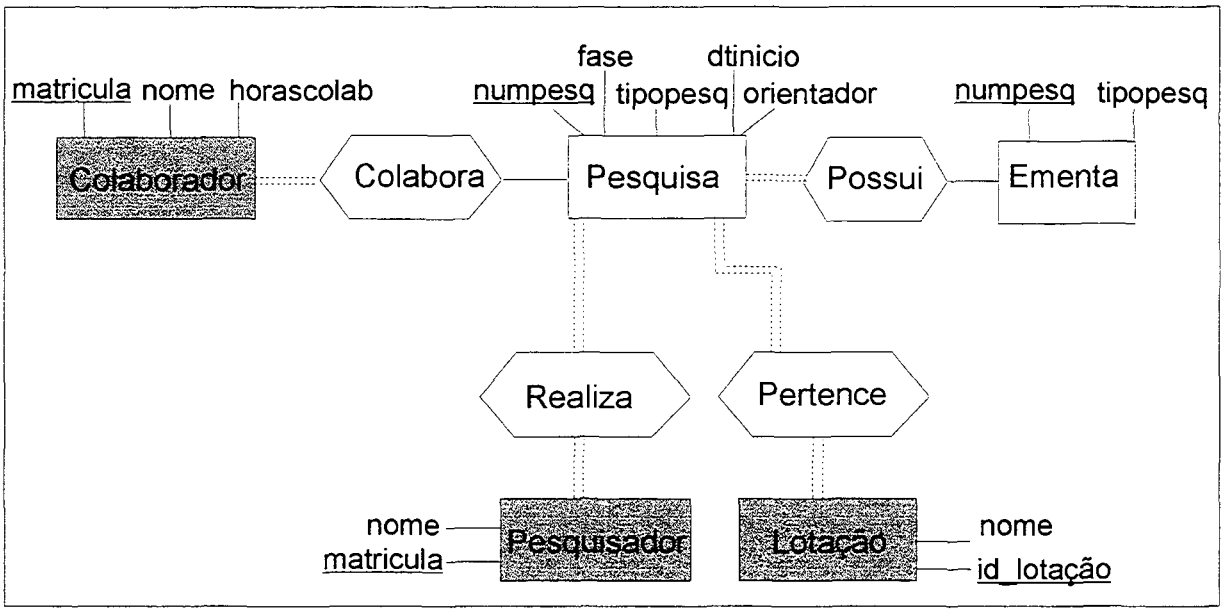


FIGURA 34: Diagrama dos principais objetos do sistema PRPPG realçando os objetos a serem integrados.

Na definição do mapeamento entre os esquemas utilizamos as regras de integração descritas no capítulo 6 deste trabalho.

Os itens abaixo identificam os casos de conflito e citam as regras empregadas para resolvê-los:

- Correspondência 1: entidade Funcionário (SAU-02) e entidade Professor (SAU-05).

  Através da definição D5 (página 55), temos:

  Funcionário $\supseteq$ Professor com os atributos correspondentes:

  idfunc = idfunc

  nome = nome

  Temos que criar, no esquema integrado, a entidade Funcionário como entidade genérica "*is-a*" de Professor.

- Correspondência 2: entidade Professor (SAU-05) e entidade Pesquisador (PRPPG). Através da definição D5 (página 55), temos:

  Professor $\cap$ Pesquisador com os atributos correspondentes:

  idfunc = matricula

  nome = nome

  Temos que criar, no esquema integrado, a entidade Pesquisador como um entidade genérica "*may-be-a*" de Professor.

- Correspondência 3: entidade Estudante (SAU-05) e entidade Pesquisador (PRPPG).

  Analisando os esquemas e a semântica dos banco de dados verificamos que a entidade Estudante abrange estudantes de pós graduação e de graduação, como somente os estudantes de pós graduação podem ser considerados pesquisadores, teremos que generalizar a entidade Estudante, criando as especializações Graduação e PósGraduação, assim esta entidade estará apta a ser integrada.

  Através da definição D5 (página 55), temos:

  Pesquisador $\cap$ PósGraduação com os atributos correspondentes:
  matricula = numalu
  nome = nome.

  Temos que criar, no esquema integrado, a entidade Pesquisador como um entidade genérica "*may-be-a*" de PósGraduação.

- Correspondência 4: entidade Departamento (SAU-02), atributo depto da entidade Professor (SAU-05) e entidade Lotação (PRPPG).

  Através da 2ª regra de integração (página 68).

  Departamento $\equiv$ Lotação *com seus atributos correspondentes:*
  id_depto = id_lotação
  nome = nome

  Através da transformação T1 (página 81) convertemos o atributo depto da entidade Professor (SAU-05) na entidade Departamento.

- Correspondência 5: entidade Estudante (SAU-05) e entidade Colaborador (PRPPG).

  Como já comentamos na correspondência 3, a entidade Estudante foi generalizada. Desta maneira, esta entidade já está apta para ser integrada com a entidade Colaborador pois somente os estudantes de graduação podem ser considerados colaboradores de uma pesquisa. Sendo assim, através da definição D5 (página 55), temos:

  Colaborador $\supseteq$ Graduação com os atributos correspondentes:
  matricula = numalu
  nome = nome

Temos que criar, no esquema integrado, a entidade Colaborador como um entidade genérica "*may-be-a*" de Graduação.

- Correspondência 6: entidade Funcionário (SAU-02) e entidade Colaborador (PRPPG).

  A definição D5 (página 55) implica em:

  Colaborador $\supseteq$ Funcionário com os atributos correspondentes:

  *matricula* = *idfunc*

  nome = nome

  Temos que criar, no esquema integrado, a entidade Colaborador como um entidade genérica "*may-be-a*" de Funcionário. .

- Correspondência 7: entidade Funcionário (SAU-02), entidade Estudante (SAU-05) e entidade Usuário (SIBI).

  Através da definição D5 (página 55), temos:

  Usuário $\cap$ Funcionário com os atributos correspondentes:

  matricula = idfunc

  nome = nome

  endereco = endereço

  *bairro* = *bairro*

  município = município

  uf = uf

  cep = cep

  Ainda pela definição D5 (página 55), temos:

  Usuário $\cap$ Estudante com os atributos correspondentes:

  matricula = numalu

  nome = nome

  endereco = endereço

  bairro = bairro

  município = cidade

  uf = uf

  cep = cep

  Como podemos verificar, através desta regra de correspondência, devemos criar, no esquema integrado a entidade Usuário como uma entidade genérica de Funcionário e de Estudante. Esta generalização é

do tipo "*may-be-a*" pois tanto os estudantes quanto os funcionários podem ser usuários da biblioteca.

A fase seguinte envolve a integração dos esquemas. Utilizamos a estratégia binária de integração, conforme ditam as metodologias que nos deram embasamento neste estudo, onde os esquemas intermediários são usados como entrada para a integração com o próximo esquema.

Como nesta primeira versão do protótipo quatro esquemas são integrados, realizamos a integração em três etapas.

A figura 35, da próxima página, ilustra a primeira etapa da integração, que envolve os sistemas SAU-02 e SAU-05.

Esta etapa gera o esquema intermediário 1, onde as correspondências 1 e 4, descritas nos parágrafos anteriores, são resolvidas.

Iremos utilizar o esquema intermediário 1, mostrado na figura 35, da próxima página, e o esquema do sistema da PRPPG (figura 31, página 104), como entrada para a segunda etapa da integração, gerando o esquema intermediário 2 que pode ser visualizado na figura 36, da página 116.

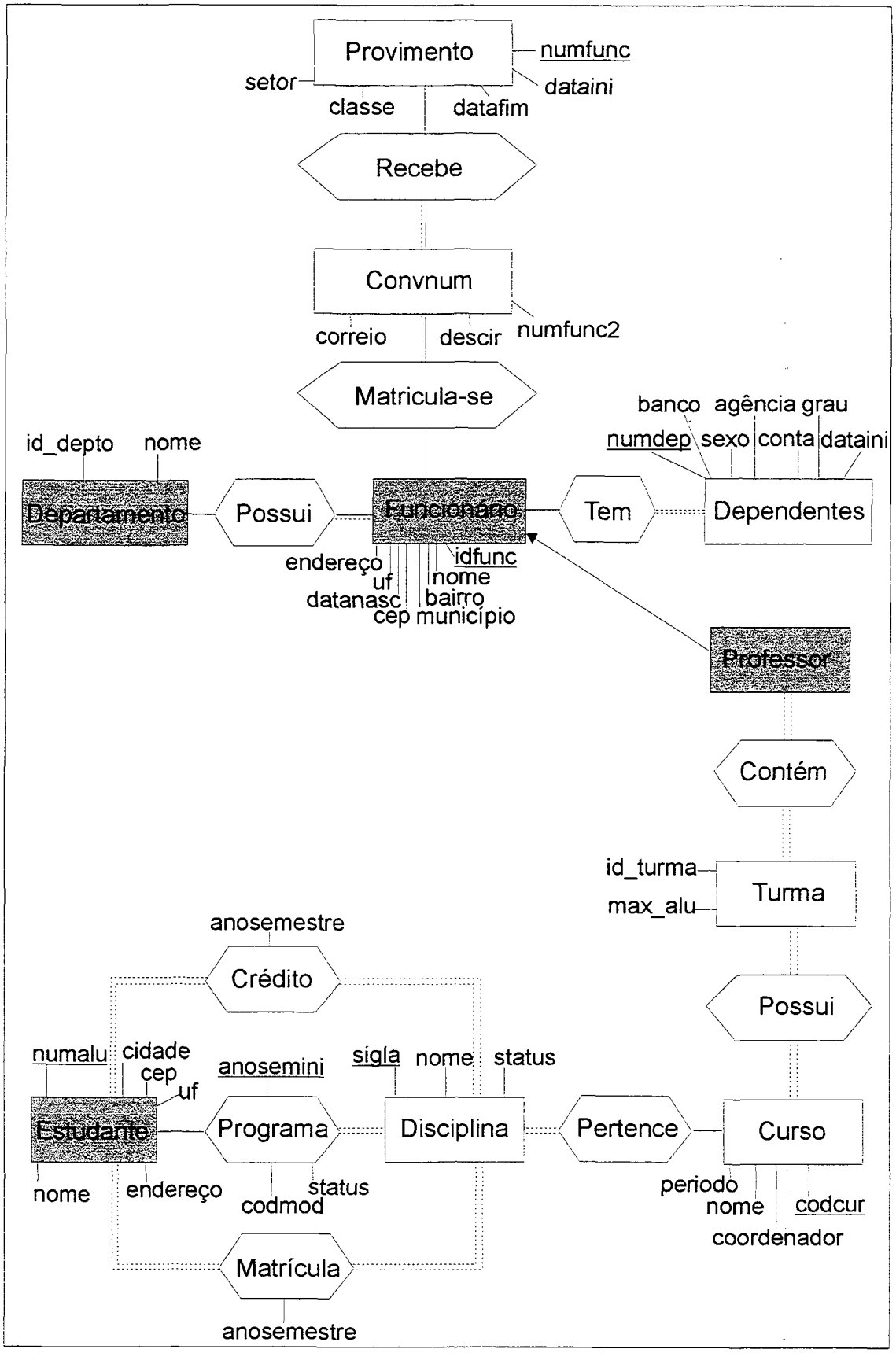


FIGURA 35: Primeira etapa da integração – esquema intermediário 1.

Conforme podemos verificar no esquema intermediário 2, através da figura 36, abaixo, neste passo resolvemos as correspondências 2, 3, 4, 5 e 6.

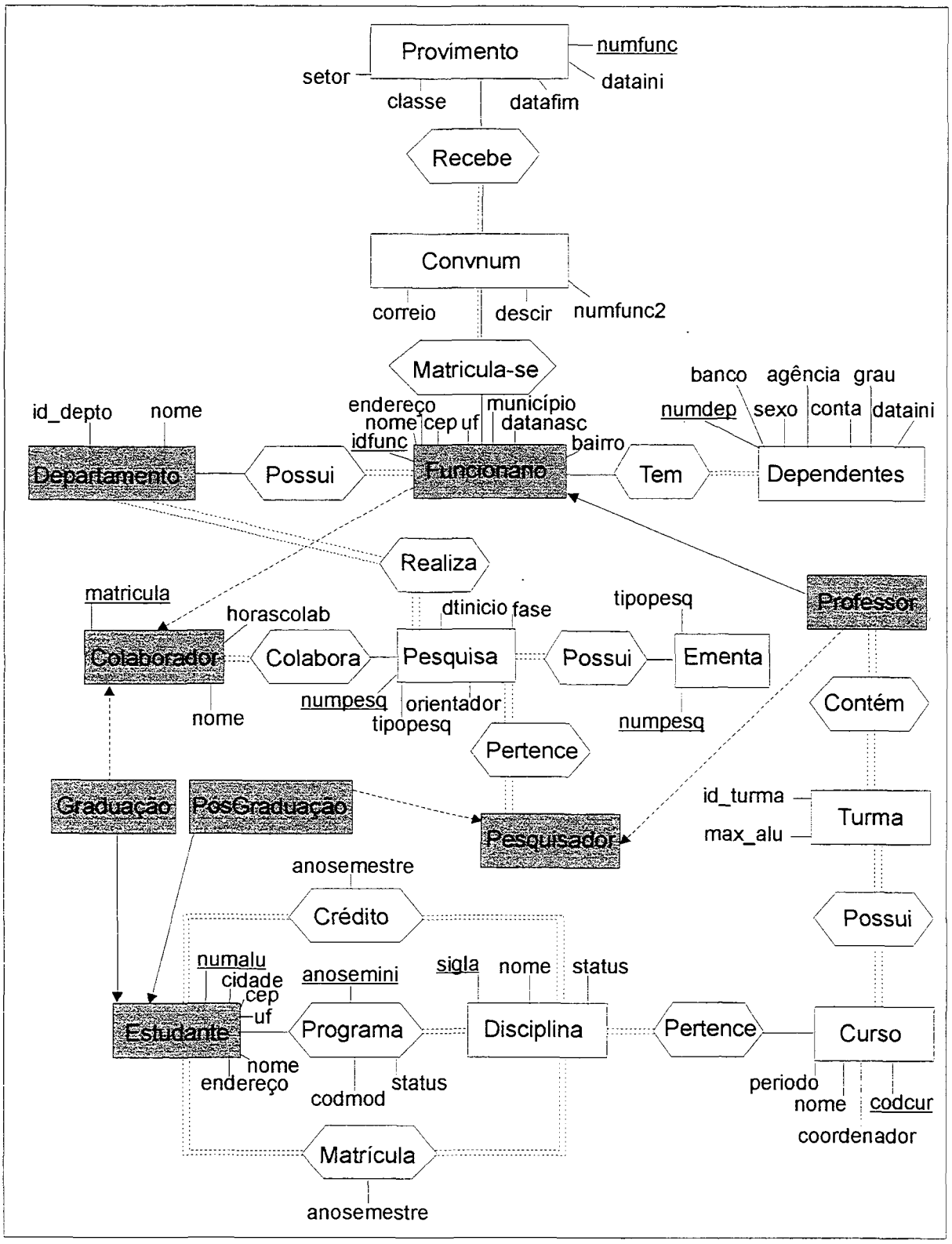


FIGURA 36: Segunda etapa da integração – esquema intermediário 2.

A terceira e última etapa do passo de integração pode ser visualizada através da figura 37, abaixo.

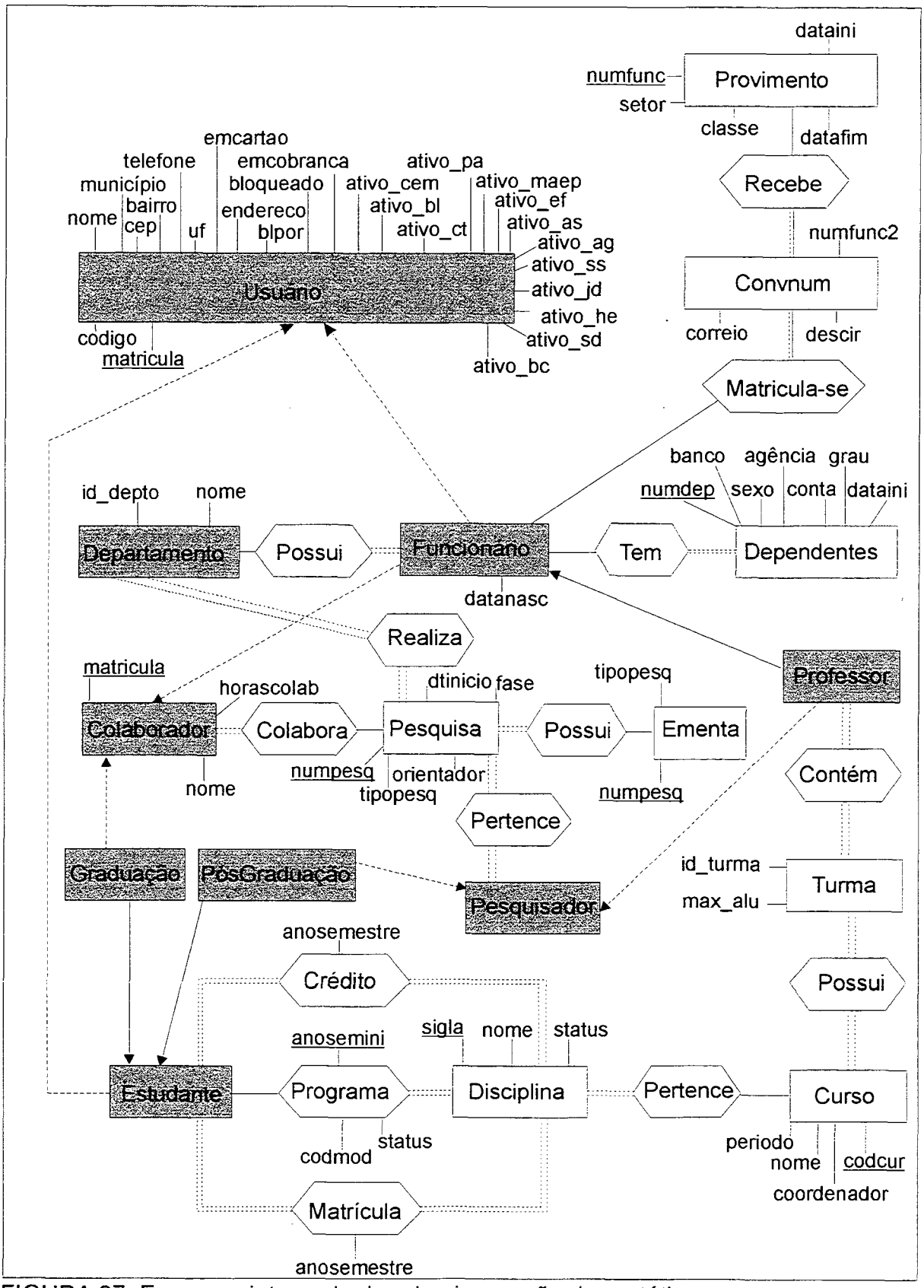


FIGURA 37: Esquema integrado da primeira versão do protótipo.

Como podemos verificar, através da figura 37 da página anterior, esta última etapa do passo de integração da primeira versão do protótipo, utiliza, como entrada, o esquema intermediário 2 e o esquema do SIBI, resolvendo a correspondência número 7.

A implementação de uma primeira versão do protótipo que permite a manipulação do esquema da figura 37, da página anterior, nos permite validar as regras de integração utilizadas pelas metodologias citadas no capítulo 6, as quais nos deram embasamento ao processo de integração. Verificamos a completeza destas metodologias, aplicando-as em um projeto prático.

A arquitetura geral de implementação do protótipo, envolvendo todos os sistemas a serem integrados, pode ser visualizada através da figura 38.

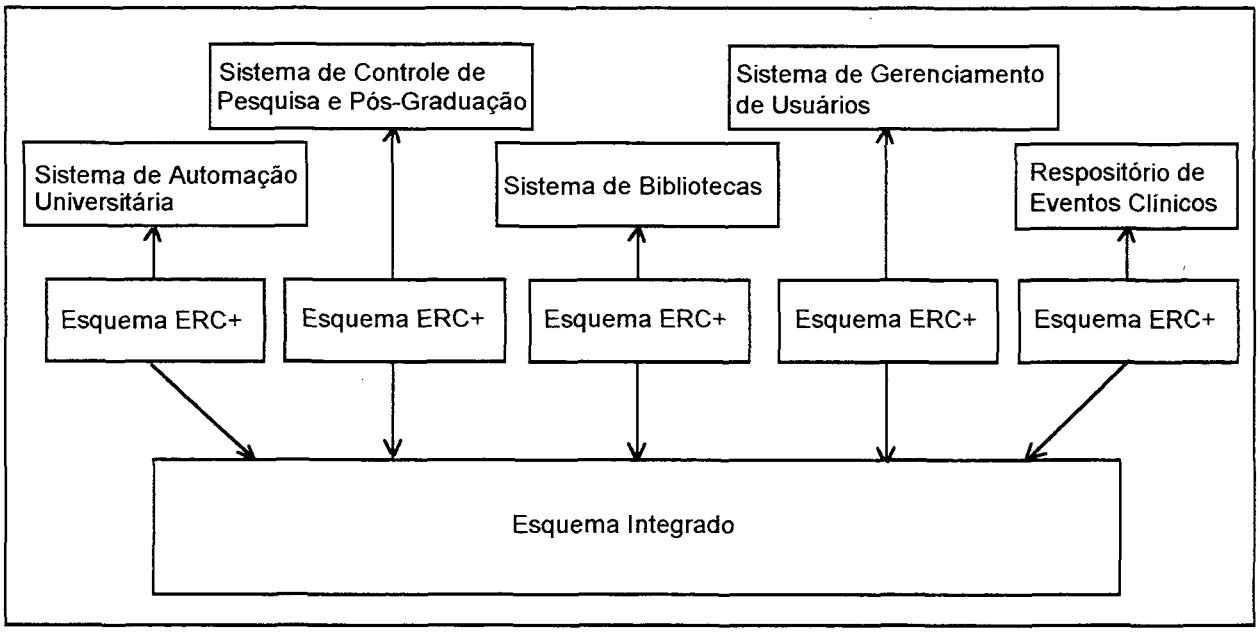


FIGURA 38: Arquitetura do esquema integrado.

A solução de mapeamento que utilizamos, visto que uma primeira versão do protótipo foi implementada, foi o mapeamento através de meta tabelas, as quais direcionam as declarações SQL para cada tabela correspondente.

Na figura 39, da próxima página, podemos visualizar este mapeamento, o qual está demonstrando não somente os sistemas que compõem a primeira versão do protótipo, mas sim todos os que irão participar do processo de integração.

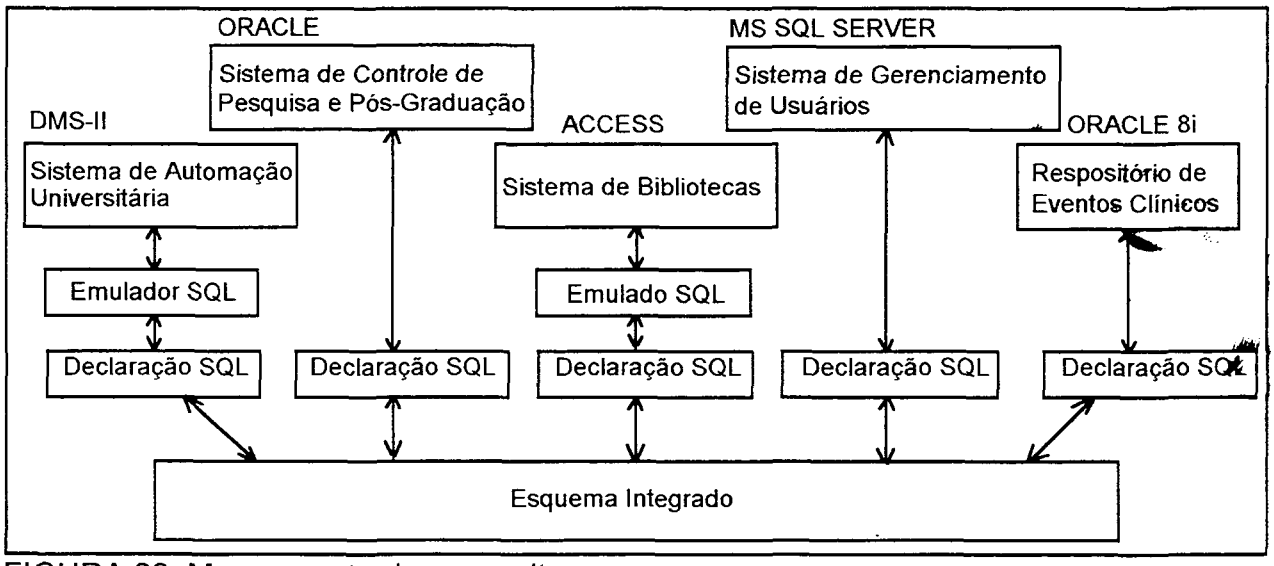


FIGURA 39: Mapeamento das consultas.

Esta primeira versão do protótipo valida nosso estudo de caso e prepara os programadores e a equipe de desenvolvimento das próximas versões para realizarem a integração de todos os sistemas.

# 8. CONCLUSÃO

Enquanto dados dentro de uma organização foram dispersando-se, a necessidade de acesso a informação através de uma visão consolidada tornou-se muito importante.

Por conseguinte, a pesquisa e o desenvolvimento de sistemas integrados tornaram-se um grande foco de pesquisa e desenvolvimento na área de informática.

Tais esforços voltaram-se para a criação de *data warehouses* que permitem que as organizações possam consolidar dados fisicamente, ou para sistemas de mediação e distribui-los, virtualmente, mantendo os vínculos com os sistemas existentes.

Na geração de produtos que se enquadrem em qualquer dos casos acima, os administradores de banco de dados (DBA's), das várias fontes de dados subjacentes, têm que entender e que relacionar os dados dos vários sistemas.

Quando nos deparamos com este fato podemos verificar que a criação de um esquema integrado, que faça uma analogia entre objetos de vários bancos de dados, é uma atividade essencial no desenvolvimento de sistemas que utilizem tecnologias relacionadas a *data warehouses* e a sistemas de mediação.

Este trabalho apresenta um estudo de caso da aplicação de duas metodologias de integração de bancos de dados, através da utilização de suas regras e conceitos, em uma aplicação prática.

Através desta aplicação, onde nos deparamos com problemas existentes no mundo real, desenvolvemos um protótipo, o qual pode ser utilizado para fundamentar a viabilidade do emprego destas metodologias na geração de esquemas integrados e em projetos práticos, visto que as mesmas foram elaboradas considerando exemplos hipotéticos e casos de uso teóricos.

Utilizamos as metodologias de SPACCAPIETRA; PARENT e DUPONT (1992) e de BATINI e LENZERINI (1984) pois, após uma análise de várias outras metodologias, as quais são citadas no capítulo 5, consideramos estas duas as mais completas e com um conjunto de regras mais factíveis.

Podemos verificar que a metodologia de SPACCAPIETRA; PARENT e DUPONT (1992) é mais completa do que a de BATINI e LENZERINI (1984), porém fizemos questão de citá-la pois, em alguns casos, a mesma pode ser utilizada de

uma maneira que vem simplificar o trabalho do integrador.

Um dos casos pode ser verificado neste trabalho, na correspondência 4, da página 112, onde utilizamos a transformação T1 (página 81), da metodologia de BATINI e LENZERINI (1984), de uma maneira mais simples, transformando um atributo em uma entidade, do que se tivéssemos aplicado a 2ª regra de integração de SPACCAPIETRA; PARENT e DUPONT (1992), citada na página 68.

Outro fato que justifica a utilização da metodologia de BATINI e LENZERINI (1984) é a nomenclatura de cores por ela utilizada para identificar os objetos de cada sistema no esquema integrado.

Por intermédio do desenvolvimento desta primeira versão do protótipo também pudemos comprovar que o desenvolvimento de um esquema integrado depende do entendimento dos objetos que compõem os sistemas que o formam.

A compreensão destes objetos é realizada através da análise dos esquemas de cada sistema. Para isso é necessário e desejável que todos os sistemas estejam modelados em um mesmo modelo de dados, facilitando o desenvolvimento de um esquema integrado que represente as relações, entidades, objetos, regras empresariais e integridade dos vários sistemas.

Por isso utilizamos, na modelagem dos esquemas iniciais do nosso estudo de caso, o modelo ERC+, que apresenta bons resultados na administração de problemas de sintaxe e suporta o relativismo semântico.

Conforme podemos verificar na seção 6.1, fizemos referência a outro modelo, o GDM. Este modelo foi utilizado porque a metodologia de SPACCAPIETRA; PARENT e DUPONT (1992) o emprega no processo de integração.

O GDM baseia-se no modelo ERC+ e o utilizamos na diagramação dos esquemas, pois possui conceitos novos, como *links* e caminhos, que são importantes no processo de integração, na detecção das correspondências entre os objetos.

Este trabalho demonstrou, também, que a participação das equipes de desenvolvimento, ou DBAs, na identificação das relações entre os objetos dos sistemas a serem integrados, é uma exigência consistente das metodologias de integração.

Porém, em grandes organizações, como é o caso da UFPR, esta tarefa é complicada, principalmente porque pode ser executada somente por alguns

indivíduos.

No nosso trabalho, como as equipes de desenvolvimento não puderam participar com toda a veemência na execução desta tarefa, o processo de integração tornou-se mais árduo e moroso.

A integração de banco de dados é um processo extremamente custoso e no estado atual muito difícil de quantificar. Podemos ter uma noção básica de horas/homem despendidos nesta primeira versão protótipo, de acordo com a tabela 1, que identifica a atividade, as pessoas que colaboraram e o tempo que a mesma foi realizada.

TABELA 1: Resumo das atividades do protótipo.

| **Atividade** | **Colaboradores** | **Tempo de Realização** |
|---|---|---|
| Treinamento das equipes responsáveis por cada sistema que o protótipo abrange. | - Orientador responsável por este trabalho. | 40 horas. |
| | - Integrador. | 40 horas. |
| | - 11 analistas. | 40 horas. |
| Passo de investigação ou pré integração. | - Integrador. | 80 horas. |
| | - 2 programadores. | 80 horas. |
| Identificação das correspondências e verificação da conformidade dos esquemas. | - Integrador. | 24 horas. |
| Integração. | - Integrador. | 16 horas. |
| Implementação. | - 2 programadores. | 160 horas. |

Através dos dados mostrados na tabela 1, podemos realizar uma estimativa percentual, em termos das atividades a serem desempenhadas em um processo de integração de bancos de dados.

Na realização deste cálculo percentual, consideramos o tempo total gasto no projeto, pelo integrador e pelos dois programadores, em relação a cada atividade. Não iremos considerar o trabalho do orientador e dos analistas que colaboraram

neste trabalho pois este pode ser considerado um caso específico. O resultado pode ser visualizado na tabela 2.

TABELA 2: Resumo das atividades do protótipo em termos percentuais.

| **Atividade** | **Colaboradores** | **Percentual** |
|---|---|---|
| Treinamento das equipes responsáveis por cada sistema que o protótipo abrange. | - Integrador. | 25 % |
| Passo de investigação ou pré integração. | - Integrador. | 50 % |
| | - 2 programadores. | 33,4 % |
| Identificação das correspondências e verificação da conformidade dos esquemas. | - Integrador. | 15 % |
| Integração. | - Integrador. | 10 % |
| Implementação. | - 2 programadores. | 66,6 % |

Estes dados, expostos em termos percentuais, nos possibilitam estimar o tempo de desenvolvimento, de qualquer processo de integração que realize as atividades conforme apresentamos neste trabalho e que tenha como colaboradores, um integrador e dois programadores. Os valores estão dispostos de acordo com o percentual que os colaboradores despenderam em cada atividade.

Estes valores também podem ser considerados em termos proporcionais, no caso de projetos que trabalhem com um número diferente de colaboradores.

Não podemos deixar de levar em consideração que o custo monetário destas atividades depende dos valores vigentes em cada organização, sendo que o tempo de desenvolvimento também pode variar de acordo com a habilidade dos colaboradores que participarão do processo, além do tamanho e complexidade dos sistemas a serem integrados.

Firmamos então, a partir de todas as exposições que fizemos, que os desenvolvedores de sistemas e os DBAs, de cada área organizacional, necessitam de metodologias eficientes que os ajudem a trabalhar no processo de integração de banco de dados, onde possam compartilhar o domínio do conhecimento e tomar decisões complexas, tais como detectar "o que deve ser integrado" e "como deve ser a integração".

Esta pesquisa teve como foco, através de uma aplicação prática, discutir a

natureza subjetiva da integração de banco de dados, aplicar metodologias de integração de sistemas de bancos de dados distribuídos heterogêneos e demonstrar os problemas enfrentados, com suas devidas soluções.

Finalmente, podemos concluir que, através do desenvolvimento deste trabalho, detectamos duas metodologias de integração, as quais compreenderam os principais casos de conflitos que podem ser encontrados em um processo de integração de banco de dados, aplicamos as regras ditadas pelas mesmas e desenvolvemos um protótipo, fechando o ciclo do processo.

## 8.1. CONTRIBUIÇÕES

Este trabalho deixa contribuições pela própria característica da primeira versão do protótipo, que é uma aplicação prática de metodologias de integração através da utilização de sistemas de um tamanho razoável e com uma problemática de heterogeneidade não trivial.

Através do desenvolvimento desta primeira versão do protótipo, fornecemos subsídios para o processo de integração dos sistemas que esta versão não englobou.

A implementação do esquema integrado, com a participação de todos os sistemas citados no capítulo 7, finalizará o trabalho de integração e permitirá que a UFPR inicie o desenvolvimento de um *data warehouse* tendo como base de consulta o esquema integrado.

Outra colaboração deste trabalho é que este estudo pode ser considerado como um roteiro a ser utilizado em um processo de integração de sistemas de bancos de dados heterogêneos.

Salientamos que o mesmo já está sendo utilizado pela equipe de desenvolvimento do projeto SAGU - Sistema de Apoio ao Gerenciamento Universitário, da UFPR.

Este projeto tem por objetivo integrar informações diversas, originárias de diferentes setores da universidade, possibilitando uma análise efetiva dos dados.

Destacam-se, ainda, mais cinco contribuições principais deste trabalho. A primeira implica na formação da equipe responsável por cada sistema em uma

metodologia comum de concepção de esquemas conceituais, o que permite prever uma maior facilidade na construção de novas bases.

Uma segunda contribuição que podemos citar é que a integração dos bancos de dados permitiu que as informações das bases operacionais, fossem vistas em um outro contexto, através de uma visão diferenciada.

Por exemplo, nesta primeira versão do protótipo, conseguimos fazer uma distinção entre os estudantes de graduação e os de pós graduação, que, em um contexto do sistema SAU-05 poderia ter uma importância quase insignificante e, quando trazida para o contexto do sistema da PRPPG, que controla as pesquisas da UFPR, torna-se imprescindível.

Essa mudança de contexto permite, muitas vezes, uma reinterpretação do significado e da importância dos dados. Este, então, é mais um benefício da integração pois criou-se uma maior consistência no controle dos dados, servindo como mecanismo de controle das bases de dados existentes, diminuindo a ocorrência de incoerências e duplicidade dos dados.

Uma quarta contribuição que podemos mencionar é que quanto mais abrangente for o processo de integração, maior o benefício gerado para a organização. Uma estratégia que leve em conta a integração de visões, além da integração dos esquemas das bases existentes, facilitará a construção de novas bases integrando-as progressivamente no esquema integrado.

Finalmente, este trabalho contribuiu na verificação de que a construção de sistemas de apoio à decisão e *data warehouses* a partir do esquema integrado torna-se mais fácil e que as atuais tecnologias de sistemas de mediação facilitam a integração de esquemas, pois podemos mapear declarações SQL para bancos de dados hierárquicos e orientados a objetos.

## 8.2. TRABALHOS FUTUROS

Seguindo uma seqüência natural e intuitiva, o próximo trabalho a ser realizado seria a implementação do mapeamento das consultas, a partir do esquema integrado, para os bancos de dados que o formam, conforme sugerimos na seção 7.4 e mostramos na figura 39 da página 119.

Outra tarefa a ser realizada seria a extensão das regras das metodologias de integração, citadas no capítulo 6, aos esquemas completos, pois em nosso protótipo utilizamos apenas sub-esquemas. A inserção dos demais sistemas citados no capítulo 7 também faria parte desta tarefa.

Seguindo as atuais linhas de pesquisa de integração de bancos de dados heterogêneos, este trabalho poderia ser utilizado como ponto de início da elaboração de uma ferramenta, ou seja, de um *software*, que automatize o processo de integração e auxilie o integrador, tornando o processo de integração mais ágil e dinâmico.

Esta ferramenta deveria realizar a modelagem dos esquemas voltada ao processo de integração, onde, após a modelagem dos esquemas iniciais, de uma maneira detalhada, seria embutida a inteligência dos dados.

Este *software* deveria ser capaz de armazenar, além da modelagem dos esquemas, informações sobre os dados, tais como, domínios dos atributos, glossário de termos dos sistemas iniciais, regras de integridade e características dos objetos que compõem os esquemas.

Assim, de uma maneira inteligente, a ferramenta poderia identificar os casos de conflito e as semelhanças existentes entre os esquemas, facilitando o processo de integração, pois teria a finalidade de edição e fusão dos esquemas.

Como já comentamos na seção 4.2, mesmo com a utilização de uma ferramenta automatizada a participação do integrador, ou DBA's responsáveis pelos sistemas que compõem o processo de integração é muito importante.

Com relação às metodologias, este trabalho poderia ser empregado na criação de uma nova metodologia de integração, que utilizasse as duas metodologias que mencionamos, criando uma terceira, que abrangeria os problemas práticos que tivemos no desenvolvimento deste trabalho, e poderia propor um conjunto de regras que seriam originadas a partir da fusão das regras das duas metodologias utilizadas neste trabalho, reformulando-as, voltando-se a aplicações práticas.

Finalmente, outro trabalho futuro que podemos mencionar seria a análise e mapeamento de atributos multivalorados no processo de integração.

## REFERÊNCIAS BIBLIOGRÁFICAS


ABITEBOUL, S.; HULL, R.; VIANU, V. Foundations of Databases. Addison-Wesley Publishing Company, 1995.

AHMED, R.; SMEDT P.; DU W.; KENT W.; KETABCHI M.; LITWIN A.; RAFII A.; SHAN M. The Pegasus heterogeneous multidatabase system. IEEE Computer, v. 24, n. 12, p. 19-27, Dec. 1991.

AL-FEDAGHI; SCHEURMANN, P. Mapping considerations in the design of schemas for the relational model. IEEE Transactions on Software Engineering, v. 7, n. 1, p. 99-111, Sep. 1981.

ATZENI, P.; TORLONE, R. A metamodel approach for the management of multiple models and the translation of schemes. Information Systems, v. 18, n. 6, Jun. 1993.

BATINI, C.; LENZERINI, M. A methodology for data schema integration in the entity relationship model. IEEE Transactions on Software Engineering, v. 10, n. 6, p. 650-664, Sep. 1984.

BATINI, C.; LENZERINI, M.; NAVATHE, S. B. A comparative analysis of methodologies for database schema integration. ACM Computing Surveys, v. 18, n. 4, p. 323-364, Dec. 1986.

BERTINO, E. Integration of heterogeneous data repositories by using object-oriented views. Proceedings of IMS'91 – The First International Workshop on Interoperability in Multidatabase Systems, 1991, p. 22-39.

BISKUP, J.; CONVENT, C. A formal view integration method. Proceedings of the ACM SIGMOD, 1986, p. 398-407.

BRIGHT, M. W.; HURSON, A. R.; PAKZAD, S. H. A taxonomy and current issues in multidatabase systems. IEEE Computer, v. 25, n. 3, p. 50-60, Mar. 1992.

BRODIE, L.; STONEBRAKER, M. Migrating Legacy Systems: Gateways, Interfaces, And The Incremental Approach. San Francisco: Morgan Kaufmann, 1995.

CASANOVA, M. A.; VIDAL, M. V. P. Towards a sound view integration methodology. Proceedings of the ACM SIGACT/SIGMOD, New York: ACM, 1983, p. 36-47.

CHATTERJEE, A.; SEGEV, A. A probabilistic approach to information retrieval in heterogeneous databases. Proceedings of the First Workshop on Information Technology Systems, 1991, p. 107-124.

CHEN, P. The Entity-Relationship Model – toward a unified view of data. ACM: Transactions on Database Systems, v. 1, n. 1, p. 9-36,1976.

CHUNG, C. W. Dataplex: An access to heterogeneous distributed databases. Communications of the ACM, v. 33, n. 1, p. 70-80, 1990.

CZEJDO, B.; TAYLOR, M. Integration of database systems using an object-oriented approach. Proceedings of IMS'91 – The First International Workshop on Interoperability in Multidatabase Systems, 1991, p. 30-37.

DEEN, S. M.; AMIM, R. R.; TAYLOR, M. C. Implementation of a prototype for Preci. Computer Journal, v. 30, n. 2, p. 157-162, 1987.

DeMICHIEL, L. Resolving database incompatibility: an approach to performing relational operations over mismatched domains. IEEE Transactions on Knowledge and Data Engineering, v. 1, n. 4, p. 484-493, 1989.

DeSOUZA, M. SIS – a schema integration system. Proceedings of the Fifth British National Conference on Databases, 1986, p. 167-185.

EL-MASRI, R.; LARSON, J.; NAVATHE, S. B. Schema integration algorithms for federated database and logical database design. Technical Report, Honeywell Systems Development Division, 1986.

ELMASRI, R.; NAVATHE S. B. Fundamentals of Database Systems. Menlo Park (CA): Library of Congress Cataloging-in-Publication Data, 1994.

FANKAUSER, P.; MOTZ, R.; HUCK, G. Schema integration methodology. Deliverable D4-4/1, IRO-DB, p. 8629, 1995.

GELLER, J.; MEHTA, A.; PERL, Y.; NEUHOLD, E.; SHETH, A. P. Algorithms for structural schema integration. Proceedings of the Second International Conference on Systems Integration, 1992, p. 604-614.

GOTTHARD, W.; LOCKEMANN, P. C.; NEUFELD, A. System-guided view integration for object-oriented databases. IEEE Transactions on Knowledge and Data Engineering, v. 4, n. 1, p. 1-22, 1992.

HAYNE, S.; RAM, S. Multi-user view integration system (MUVIS): an expert system for view integration. Proceedings of the Sixth International Conference on Data Engineering, Los Alamitos (CA): IEEE Computer Society Press, Feb. 1990.

HEIMBIGNER, D.; McLEOD, D. A federated architecture for information systems. ACM Transactions on Office Information Systems, v. 3, n. 3, p. 253-278, Jul. 1985.

HURSON, A. R.; BRIGHT, M. W.; PAKZAD, H. Multidatabase systems: an advanced solution for global information sharing. IEEE Computer Society Press, Los Alamitos (CA), 1994.

JOHANNESSON P. Schema transformation as an aid in view integration. Proceedings of the Fifth International Symposium on Advanced Systems Engiuneering, 1993, p. 71-92.

KAUL, M.; DROSTEN, K.; NUEHOLD, E. J. Viewsystem: integrating heterogeneous information bases by object-oriented views. Proceedings of the Sixth International Conference on Data Engineering, 1990, p. 2-10.

KIM, W.; SEO, J. Classifying schematic and data heterogeneity in multidatabase systems. IEEE Computer, v. 24, n. 12, p. 12-18, Dec. 1991.

KRISHNAMURTHY, R.; LITWIN, W.; KENT, W. Language features for interoperability of databases with schematic discrepancies. Proceedings of the ACM SIGMOD, New York: ACM, May 1991, p. 40-49.

LARSON, J.; NAVATHE, S. B.; EL-MASRI, R. A theory of attribute equivalence and its applications to schema integration. IEEE Transactions on Software Engineering, v. 15, n. 4, p. 449-463, Apr. 1989.

MANNINO, M. V.; EFFELSBERG, W. Matching techniques in global schema design. Proceedings of the First International Conference on Data Engineering, 1984, p. 418-425.

MOTRO, A.; BUNEMAM P. Constructing superviews. ACM SIGMOD Record, 1981, p. 56-64.

NAVATHE, S. B.; GADGIL, S. G. A methodology for view integration in logical database design. Proceedings of the Eighth VLDB Conference, 1982, p. 142-162.

NAVATHE, S.; EL-MASRI, E.; LARSON, J. Integrating user views in database design. IEEE Computer, v. 19, n. 1, p. 50-62, 1986.

PARENT, C.; ROLIN, H.; YÉTONGNON, K.; SPACCAPIETRA, S. An ER calculus for the entity-relationship complex model. 8th International Conference on Entity-Relationship Approach, Toronto, Oct. 18-20, 1989, p. 75-98.

PARENT, C.; SPACCAPIETRA, S. A model and an algebra for entity-relation type databases. Technology and Science of Informatics, Special Issue: Databases, v.6, n. 8, p. 623-642, Nov. 1987.

PARENT, C.; SPACCAPIETRA, S. About entities, complex objects and object-oriented data models. Information System Concepts: An In-depth Analysis, E.D. Falkenberg and P. Lindgreen Eds., North-Holland, 1989, p. 193-223.

PARENT, C.; SPACCAPIETRA, S. An algebra for a general entity-relationship model. IEEE Transactions On Software Engineering, v. 11, n. 7, p. 634-643, Jul. 1985.

PRABHAKAR, S.; RICHARDSON, J.; SRIVASTAVA, J.; LIM, E. P. Instance-level integration in federated autonomous databases. Proceedings of the 26th Annual Hawaii International Conference on System Sciences, v. 3, p. 62-69, 1993.

RAM, S.; BARKMEYER, E. The unifying semantic model for accessing multiple heterogeneous databases in a manufacturing environment. Proceedings of IMS'91 – The First International Workshop on Interoperability in Multidatabase Systems, 1991, p. 212-216.

RAMESH, V.; RAM, S. A methodology for interschema relationship identification in heterogeneous databases. Proceedings of the Hawaii International Conference on Systems and Sciences, 1995, p. 263-272.

RAMESH, V.; RAM, S. Integrity constraint integration in heterogeneous databases: an enhanced methodology for schema integration. Information Systems, v. 22, n. 8, p. 423-446, 1997.

RUSINKIEWICZ, M.; CZEJDO, B. An approach to query processing in federated database systems. In: Proceedings of the 20th Hawaii International Conference on System Sciences, 1987.

SHEKHAR, S.; HAMIDZADEH, B.; KOHLI, A.; COYLE, M. Learning transformation rules for semantic query optimization: A data driven approach. IEEE Transactions on Knowledge and Data Engineering, v. 5, n. 6, p. 950-964, 1993.

SHETH, A. P. Issues in schema integration: perspective of an industrial researcher. ARO Workshop on Heterogeneous Databases, 1991.

SHETH, A. P.; MARCUS, H. Schema analysis and integration: methodology, techniques and prototype toollkit. Technical Report TM-STS-019981/1, Bellcore, 1992.

SHETH, A.; GALA, S. Attribute relationships: an impediment in automating schema integration. In: Proceedings of the NSF Workshop on Heterogeneous Databases, Dec. 1989.

SHETH, A.; GALA, S.; NAVATHE, S. On automatic reasoning for schema integration. International Journal on Intelligent and Cooperative Information Systems, v. 2, n. 1, Mar. 1993.

SHETH, A.; KASHYAP, V. So far (schematically), yet so near (semantically). Proceedings of the IFIP TC2/WG2.6 Conference on Semantics of Interoperable Database Systems, DS-5. Amsterdam: North-Holland, Nov. 1993.

SHETH, A.; LARSON, J. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys, v. 22, n. 3, p. 183-236, Sep. 1990.

SHETH, A.; LARSON, J.; CORNELIO, A.; NAVATHE, S. B. A tool for integrating conceptual schemata and user views. Proceedings of the Fourth International Conference on Data Engineering, Los Alamitos (CA):IEEE Computer Society Press, Feb. 1998, p. 176-183.

SHOVAL, P.; ZOHN, S. Binary-relationship integration methodology. Data and Knowledge Engineering, v. 6, p. 225-250, 1991.

SMITH, J.; SMITH, D. Database abstractions: aggregation and generalization. TODS, v. 2, n. 2, Jun. 1977.

SPACCAPIETRA, S.; DEMO, B.; PARENT, C. SCOOP: a system for integrating existing heterogeneous distributed data bases and application programs. Proceedings IEEE INFOCOM Conference, San Diego, Apr. 1983, p. 18-21.

SPACCAPIETRA, S.; PARENT, C. ERC+: an object based entity relationship approach. In: Conceptual Modelling, Database and CASE: An Integrated View of Information Systems Development. John Wiley, 1992.

SPACCAPIETRA, S.; PARENT, C. View integration: a step forward in solving structural conflicts. IEEE Transactions on Knowledge on Data Engineering, v. 6, n. 2, Apr. 1994.

SPACCAPIETRA, S.; PARENT, C.; DUPONT, Y. Independent assertions for integration of heterogeneous schemas. Very Large Database Journal, v. 1, n. 1, 1992.

STOCKER, P. M.; CANTIE, R. A target logical scheme: the acs. Proceedings of the Ninth VLDB Conference, 1983.

TEMPLETON, M.; BRILL, D.; DAO, S. K.; LUND, E.; WARD, P.; CHEN, A. L. P.; MacGREGOR, R. Mermaid: a front-end to distributed heterogeneous databases. Proceedings of the IEEE, v. 75, n. 5, p. 695-708, May 1987.

THIEME, C.; SIEBES, A. Schema integration in object-oriented databases. Proceedings of the Fifth International Symposium on Advanced Information Systems Engineering, CaiSE'93, 1993, p. 54-70.

WHANG, W. K.; NAVATHE, S. B.; CHAKRAVARTHY, S. Logic-based approach for realizing a federated information system. Proceedings of IMS'91 – The First International Workshop on Interoperability in Multidatabase Systems, 1991, p. 92-100.