

**\*RELATÓRIO TÉCNICO\***

**DEFININDO E UTILIZANDO  
BANCO DE DADOS COM O MODELO  
ENTIDADE-RELACIONAMENTO**

**Pedro Manoel da Silveira**

**NCE-25/90  
Outubro/90**

**Universidade Federal do Rio de Janeiro  
Núcleo de Computação Eletrônica  
Caixa Postal 2324  
20001 - Rio de Janeiro - RJ  
BRASIL**

**Este artigo foi publicado originalmente nos Anais do XXIII  
Congresso Nacional de Informática da SUCEsu, Rio de Janeiro, 1990.**



**UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA**

# Definindo e Utilizando Bancos de Dados com o Modelo Entidade-Relacionamento

## RESUMO

O presente artigo apresenta as bases de uma metodologia de definição e utilização de bancos de dados baseada no Modelo Entidade-Relacionamento, trazendo vantagens como simplicidade, clareza e uniformização do conhecimento sobre a base de dados para usuários, analistas e programadores. Suportando a proposta, nós apresentamos ferramentas e componentes de *software* que a viabilizam num ambiente de produção.

# Designing and Using Databases With the Entity-Relationship Model

## ABSTRACT

We present the practical basis of a methodology for defining and using databases with the Entity-Relationship Model. That brings facilities such as clarity, simplicity and a uniform level of information for technicians and end users. Along the proposal, we present the necessary tools to make it work on a production environment.

# 1 INTRODUÇÃO

Um dos aspectos mais importantes na utilização de uma base de dados é a sua descrição lógica. É fundamental que ela seja bem entendida por todos os usuários, desde o analista até o usuário casual. Os conceitos devem estar claros e ao alcance de todos. A maior parte dos erros ocorridos durante a utilização de uma base de dados advém de um pobre entendimento de suas construções e interpretação. Conhecer-la bem significa minimizar esses erros.

Mesmo assim, muitos sistemas de gerenciamento não dão a devida importância para o problema e as interfaces entre usuário e base de dados não destacam devidamente os aspectos necessários ao seu bom entendimento. A descrição do banco de dados fica relegada à obscuras construções, muitas vezes embutidas nos programas, cujo principal objetivo é descrever o sistema para o próprio computador, relegando a necessidade de ser clara para seus usuários a um segundo plano.

Este artigo apresenta as bases de uma metodologia de projeto e utilização de bancos de dados que tem por objetivo atacar essa questão, proporcionando aos seus usuários uma forma clara de comunicação e entendimento das estruturas da base de dados. O Modelo Entidade-Relacionamento foi tomado como alicerce desse trabalho, devido a sua simplicidade e aceitação pela comunidade de processamento de dados. Normalmente, o Modelo E-R suporta a fase de modelagem conceitual das bases de dados, enquanto que na implementação são empregados sistemas de gerenciamento de bancos de dados (SGBD) baseados em modelos mais simples ou orientados para aspectos da arquitetura do sistema hospedeiro. Nossa sugestão é que também a fase de implementação continue calcada no Modelo E-R. Essa estratégia traz vantagens como simplicidade, clareza e uniformização do conhecimento sobre a base de dados para usuários, analistas e programadores.

Ao longo do processo de utilização das bases de dados, as atividades são todas suportadas por interfaces gráficas que mantêm, sempre que possível, a estrutura da base de dados em evidência. Isso permite ao usuário ater-se aos problemas específicos de sua tarefa, deixando que o *software* direcione os procedimentos através de menus e mantenha as especificações dentro dos padrões existentes. As tarefas podem ser a confecção de relatórios simples, consultas *ad hoc*, integração a sistemas especialistas e programas de aplicação COBOL. Planeja-se também a criação de uma metodologia de desenvolvimento de novas aplicações integrada ao ambiente aqui apresentado.

O artigo está dividido em cinco seções, além desta introdução e das referências bibliográficas. A Seção 2 considera alguns aspectos de modelagem de dados e a Seção 3 apresenta o Modelo E-R utilizado e suas extensões. Na Seção 4 nós explicamos como seria a utilização através das interfaces aqui descritas e a Seção 5 descreve as ferramentas de *software* que suportam ao sistema. A Seção 6 conclui o trabalho.

## Sistemas isolados e necessidade de integração

A presente proposta surgiu no âmbito da Universidade Federal do Rio de Janeiro, onde buscava-se uma melhor integração dos vários sistemas de informação lá implantados. O modelo inicial de desenvolvimento havia sido o de aplicações isoladas, com escopo restrito e compartimentalizado, que rapidamente mostrou suas deficiências. Era necessário um todo maior do que a simples soma das partes.

Esse modelo não foi escolhido sem motivos: de um lado, a maneira de se trabalhar em processamento de dados induzia tal paradigma e, de outro lado, os diferentes estágios de organização e desenvolvimento dos órgãos internos da Universidade não propiciavam, e por vezes até impediam, uma abordagem homogênea às tarefas apoiadas por computação. Na vida real, entretanto, a unidade administrativa da organização não deixou de existir. Apesar das diferenças no grau de automação de cada órgão administrativo, a estrutura funcional supõe uma constante e eficiente troca de serviços e informações entre seus componentes. A segmentação existe apenas em sua modelagem computacional.

Essa situação fica evidente quando é necessário controlar as atividades da corporação num nível mais elevado do que o simplesmente operacional: como cruzar informações, comparar grandezas, avaliar tendências, obter graus de eficiência, se coisas que são de interesse global tem interpretações específicas e não seguem padronizações em tarefas distintas? Fica difícil para a Administração e mesmo para o pessoal técnico compatibilizar as informações, suprir as lacunas que eventualmente ocorrem entre os diversos sistemas e trabalhar produtivamente na direção de aproveitar a massa de informações disponível.

## Projeto UniversiData: dados como recurso central

Da idéia de analisar esses problemas e trabalhar na direção de melhores soluções, surgiu o Projeto UniversiData. O objetivo deste projeto é fornecer subsídios, tanto no que tange à organização dos sistemas como na parte

referente aos modelos lógicos e físicos dos dados, para que sejam obtidas soluções que minimizem os índices de duplicação, interpretações errôneas, lacunas de modelagem e heterogeneidade nos sistemas automatizados que operam na Universidade.

A idéia básica é que a linha do projeto se situe em torno do conceito de administração de dados. Deve-se buscar a homogeneização dos sistemas, juntamente com sua completude e simplicidade, através da manutenção de um controle rigoroso e disciplinado dos bancos de dados mantidos pelos processos automatizados. Note-se que, de imediato, faz-se uma opção por uma visão primariamente calcada nas informações (dados), em detrimento dos aspectos puramente funcionais daqueles processos. As idéias aqui expostas devem ser consideradas nesse contexto e as implicações daí decorrentes devem ser cuidadosamente avaliadas.

Uma delas é que estabelecer a priori as bases de dados que suportarão as atuais implementações e, possivelmente, algumas futuras, rompe, de certa forma, a natural correlação *dado-função*. É preciso que se tenha uma base de dados suficientemente ampla e genérica de modo a suportar novas implementações, sem mudanças radicais de interpretação ou estrutura. É preciso, ainda, adaptar de certa forma a metodologia de análise para que, no processo de concepção e desenvolvimento de novas aplicações, a base de dados seja aproveitada e utilizada de forma racional e correta, não havendo introdução de distorções no modelo global.

## Uniformidade nos vários níveis de utilização

Usualmente, os modelos de dados são utilizados para a definição lógico-conceitual dos sistemas, passando-se então à implementação em sistemas convencionais de gerenciamento de bancos de dados. Muita vezes a descrição do sistema fica empobrecida nessa passagem, e esta é a descrição que será utilizada como referência para programação e manutenção do sistema. No âmbito do Projeto UniversiData, a idéia é proporcionar um ambiente homogêneo para definição e manipulação dos bancos de dados. Desse modo, tanto usuários como analistas e programadores “falariam” uma linguagem comum, evitando-se grande parte dos erros e desencontros que são comuns em processamento de dados devido à diversidade das formas de expressão e documentação empregadas.

## Premissas e condições preliminares

O Projeto UniversiData integra-se a um ambiente de processamento de dados já desenvolvido e em pleno funcionamento. Esse fato implica numa série de restrições e condições preliminares que devem ser observadas, dentre as quais destacamos:

- existência de sistemas implantados em máquinas de grande porte. Esta característica restringe sensivelmente a capacidade de dispormos de interfaces gráficas, a não ser através de ligações em rede. Também é necessário que se conte com recursos apropriados para sistemas gerenciadores de dados para *mainframes*, tais como multi-processamento, controle de concorrência, recuperação de falhas e outros;
- “cultura” de programação COBOL, fazendo com que seja necessária a introdução de meios de acesso à interface através de programas escritos naquela linguagem;
- manutenção dos investimentos já realizados em *software*, introduzindo um mínimo de alterações;
- desejo de introduzir facilidades para a execução de tarefas simples de acesso ao banco de dados, tais como a produção de relatórios e a execução de consultas *ad hoc*;
- desejo de introduzir facilidades para utilização de técnicas de inteligência artificial para manipulação de bases de dados;
- manter o sistema o mais possível independente do *hardware* utilizado, permitindo utilização de bases de dados em redes, com uma interface completamente uniforme;
- domínio da tecnologia empregada, de modo a possibilitar mudanças e melhorias do *software* ao longo do tempo.

## 2. MODELAGEM DE DADOS

A modelagem do esquema conceitual de um banco de dados é de vital importância e deve refletir o mundo real das aplicações de forma simples, completa e correta. Não é trivial a obtenção de tais propriedades quando se deseja que, adicionalmente, o modelo possa ser implementado eficientemente. Deve-se escolher uma disciplina de modelagem que abrigue tais preocupações e que seja de utilidade comprovada. Assim, não se pode expandir demasiadamente

a liberdade de expressão do modelo, criando-se facilidades elegantes e visíveis, mas de difícil implementação, nem muito menos torná-lo tão simplório ao ponto da inutilidade.

## Modelos Semânticos

Muito se tem discutido sobre que modelos de dados seriam apropriados para aplicação em processamento de dados. Os chamados Modelos Semânticos,<sup>5</sup> surgidos ao longo da década de 70, fazem parte de uma tentativa dos pesquisadores para que a compreensão do que é modelado seja mais fácil e completa. Várias disciplinas de modelagem foram introduzidas, embora a maior parte delas tenha ficado apenas em protótipos experimentais e não tenham alcançado aplicação prática mais extensiva.

Dentre esses modelos de dados, destaca-se claramente o Modelo Entidade-Relacionamento, introduzido por Chen.<sup>1</sup> O Modelo E-R é baseado em conceitos extremamente simples mas que são satisfatórios para grande parte das aplicações convencionais de computadores. Há ainda várias extensões propostas,<sup>3,6</sup> melhorando o E-R e tornando-o apto para modelagem de casos mais específicos. Assim, optou-se pelo Modelo E-R como base para todo o processo de definição e manipulação do bancos de dados.

## 3. O MODELO E-R E ALGUMAS EXTENSÕES

nesta seção nós apresentamos os conceitos básicos do Modelo E-R, juntamente com as extensões que foram adotadas no Projeto UniversiData. Esta apresentação é bastante informal; o leitor interessado poderá encontrar detalhes e definições formais em dois artigos precedentes.<sup>8,9</sup>

### Elementos de esquemas E-R

Os elementos que podem ser usados para a construção de esquemas são os seguintes.

- a) *Constantes*, que expressam valores tais como números, strings e outros objetos cujas interpretações não variam.
- b) *Conjuntos de entidades e Relacionamentos*, aqui identificados por  $E_1, E_2, \dots$  e  $R_1, R_2, \dots$  respectivamente. A cada relacionamento  $R$  está associado um número inteiro  $n$ . Este corresponde ao número de entidades que participam do relacionamento.  $E_1, \dots, E_n$  é a lista de

entidades que participam de  $R$ , sendo que  $n$  deve ser igual ao grau de  $R$ .

- c) Para cada relacionamento  $R$  há ainda um conjunto de papéis associados, da forma  $\{p_1, \dots, p_n\}$ . Cada um desses papéis é de fato uma função da forma

$$p_i : R \rightarrow E_i$$

Ou seja,  $p_i$  é uma função que, aplicada a um elemento de um relacionamento, retorna um elemento da  $i$ -ésima entidade a ele associada. As funções *papel* são utilizadas para identificar quais os elementos das entidades participantes que estão associados nos relacionamentos.

- d) *Domínios*, aqui identificados como  $V_1, V_2, \dots$  são conjuntos de constantes e são utilizados para especificar os domínios dos atributos. Exemplos são conjuntos de strings, inteiros, reais.
- e) *Atributos* recebem a notação  $A_1, A_2, \dots$ . Cada atributo  $A$  é uma função da forma  $A : X, I \rightarrow V$ , onde  $X$  pode ser um conjunto de entidades ou um relacionamento,  $I$  é um inteiro e  $V$  um domínio. Atributos podem ser multi-valorados ou simples, dependendo do número máximo de ocorrências associado a eles. Para atributos simples, este valor é 1 e  $> 1$  para multi-valorados. O número inteiro  $I$  indica qual a ocorrência que a função relativa ao atributo deve retornar.
- f) *Restrições de Integridade* são fórmulas bem-formadas e sua aplicação será comentada com mais detalhes logo adiante.

## Propriedades de um esquema E-R

Algumas construções do Modelo E-R por nós utilizadas caracterizam propriedades do esquema e, por consequência, são consideradas como restrições de integridade. Estas são: *chaves*, *cardinalidades dos relacionamentos* e *entidades fracas*.

Chaves determinam combinações de atributos que unicamente identificam elementos de conjuntos de entidades e/ou relacionamentos. Para garantir que as chaves sejam corretamente implementadas é necessário respeitar a restrição de unicidade de seus valores.

As funções  $Cmax$  e  $Cmin$ ,<sup>4</sup> identificam para cada relacionamento e suas entidades associadas as combinações permitidas no que refere ao número de elementos associados. Assim, podemos ter combinações  $1 : n$ ,  $n : m$ ,



etc. Novamente, a manutenção das cardinalidades recai no caso geral de manutenção de restrições de integridade.

Entidades fracas são um caso especial de conjunto de entidades, tal que a identificação e existência de seus elementos dependem das instâncias de outras entidades. Um exemplo clássico que aparece na Figura 2, é o da entidade fraca DEPENDENTE, relacionada com a entidade PROFESSOR, de modo que a existência e a própria identificação de um dependente dá-se através do relacionamento que o une ao seu titular. Como nos casos acima, isso pode ser considerado como uma restrição de integridade.

## Generalização e especialização

Generalização é o termo usado quando duas ou mais entidades podem ter algumas de suas características abstraídas e, num nível superior, representadas como uma entidade comum, a qual possui atributos que se aplicam aos diferentes tipos reunidos. O processo inverso se denomina especialização, onde se pode ir do geral para o específico. Hierarquia de tipos é simplesmente a suposição de que um certo subtipo é sempre parte de um tipo superior, embora nem todos os elementos do supertipo devem estar presentes em algum dos subtipos correspondentes. Em todos os casos de especialização, a entidade que é subtipo herda os atributos de seu supertipo. Na Figura 2, a entidade PESSOA é uma generalização das entidades PROFESSOR e ESTUDANTE.

No nosso caso, uma dada entidade pode ter apenas uma outra entidade como seu supertipo. Também não é permitido haver recursão: se a rede de generalizações fosse representada como um grafo, não haveria ciclos.

## Ordens

Acesso ordenado não é suportado pelo Modelo E-R em sua definição original. Isto é devido à função inicial do modelo de servir como uma representação conceitual de dados, não havendo suporte para implementações físicas. Essa, entretanto, é uma importante característica dos bancos de dados e é extensivamente utilizada durante a programação. A obtenção de acesso ordenado a bancos de dados geralmente envolve:

- a) um domínio, que é um conjunto de elementos de entidades/relacionamentos que devem ser ordenados;
- b) uma expressão de seleção, que designa que elementos do domínio devem aparecer na ordem final;

- c) os determinantes da ordenação, que são atributos ou outros termos do domínio da ordenação que devem determinar o critério de ordenação a ser seguido.

#### 4. UTILIZAÇÃO ATRAVÉS DO MODELO E-R

A utilização do banco de dados através do Modelo E-R dá-se através da interação de cada um dos componentes de *software* com um servidor E-R, conforme ilustrado na Figura 1. Pode-se observar que componentes como a Interface de Consultas, o Gerador de Relatórios e a Interface para Sistemas Especialistas e os próprios programas de aplicação (em COBOL) comunicam-se com o servidor para acesso ao banco de dados objeto. Essa comunicação está representada por uma espécie de barra de comunicações, à qual também estão ligados o diagrama E-R e o dicionário de dados. Todo acesso ao banco de dados deve ser realizado à luz do diagrama E-R e do dicionário, uma vez que assim é possível verificar e interpretar as operações.

Esses dois componentes são gerados por ocasião da definição do esquema do banco de dados a partir do diagramador E-R. Nessa fase, são definidos elementos do Modelo E-R tais como entidades, relacionamentos, atributos, além dos elementos adicionais como generalizações, chaves, ordens de acesso, cardinalidade dos relacionamentos e eventuais restrições de integridade. Tudo isso é coletado e organizado no dicionário de dados que retrata o diagrama. Além dessas informações, o dicionário deve armazenar informações que sejam necessárias a outros componentes do projeto global, tais como relatórios, consultas e suporte às aplicações.

A ligação das interfaces com o servidor pode existir diretamente ou através de uma ligação remota. Isso viabiliza a existência de uma estação E-R, onde se tem acesso ao banco de dados objeto por meio de uma ligação de padrão definido, atendida pelos servidores remotos, localmente e no *mainframe*. Note-se que essa característica abre a possibilidade de que se aproveitem as facilidades gráficas de computadores de pequeno porte aliadas à capacidade de processamento do sistema de grande porte.

Na fase de projeto do banco de dados, é possível gerar a descrição do banco de dados objeto a partir do diagrama E-R. Esta conversão é feita observando-se a estrutura do diagrama original e suas propriedades. O dicionário de dados é também alimentado pelo Gerador de Bancos de Dados com informações que relacionam as primitivas do Modelo E-R com as primitivas da implementação do banco de dados objeto.

A função do servidor de banco de dados é traduzir as operações sobre o Modelo E-R para operações sobre o banco objeto, coletar as informações provenientes de sua execução e enviá-las aos módulos que as solicitaram. Essas operações tanto podem ser de consulta como de atualização. Todos os controles operacionais são aproveitados da implementação, de modo a manter o servidor simples e razoavelmente independente da máquina correntemente em uso.

## Operações sobre o Modelo E-R

As operações sobre o Modelo E-R que são suportadas pelo servidor estão descritas em detalhes em dois outros trabalhos.<sup>8,9</sup> Há quatro classes de operações:

- a) avaliação de expressões lógicas;
- b) avaliação de termos e funções;
- c) operações de busca;
- d) operações de atualização da base de dados.

No primeiro caso, aparecem as operações que correspondem à avaliação de expressões lógicas, especialmente o caso  $t \in X$ , onde é testada a condição de pertinência do termo  $t$  em relação ao conjunto  $X$ . O resultado de um teste dessa natureza é um valor *booleano*, verdadeiro ou falso.

A avaliação de termos corresponde à busca do valor de expressões formadas por atributos ou funções papel. Por exemplo, se  $x$  é conhecido, a expressão

$$nome(x)$$

pode ser avaliada se *nome* for um atributo do conjunto de entidades que é domínio para a variável  $x$ .

Operações de busca podem ser de dois tipos básicos: por chave e por ordem. No primeiro caso, dado um conjunto de valores que caracterizam uma chave de uma entidade ou relacionamento, é possível localizar o elemento em questão. No segundo caso, é possível seguir certas ordens no acesso a algum elemento da base de dados. Ordens foram melhor explicadas na Seção 3 e as operações correspondem a encontrar o próximo ou o anterior em relação a um elemento de referência. Por exemplo, é possível acessar as entidades em ordem alfabética se esta for uma das ordens mantidas pelo banco de dados. As ordens são definidas sobre o diagrama E-R e, quando da geração do banco

de dados objeto, são criadas as estruturas necessárias para sua manutenção e acesso.

Operações de atualização correspondem basicamente à inclusão e exclusão de elementos em entidades e relacionamentos e à alteração de atributos e funções.

Esse conjunto de operações é comum a todos os componentes de *software* ilustrados na Figura 1. Isto é, programas COBOL, interfaces de consulta e sistemas especialistas, gerador de relatórios devem traduzir seus procedimentos para uma linguagem comum de acesso à base de dados. Esta característica dá independência aos módulos componentes em relação à implementação objeto, pois a tradução das operações fica totalmente restrita ao servidor.

### Utilização por programas de aplicação

A utilização a partir de programas de aplicação merece especial destaque e, por isso, esta seção apresenta uma idéia de como se constróem programas para acesso ao banco de dados via Modelo E-R.

Suponha, em relação ao diagrama da Figura 2, que a operação seja acessar os atributos *nome* e *endereço* de um elemento de PESSOA cujo *cpf* seja igual a um determinado valor, e movê-los para variáveis auxiliares.

Sem pré-compilador:

```
...  
CALL ERFIND (pessoa, "CPF=xxxxx").  
CALL ERGET  (pessoa, nome, nomeaux\0.  
CALL ERGET  (pessoa, endereco, enderecoaux)
```

Com pré-compilador:

```
77  pessoa USAGE IS ENTITY-POINTER  
  
pessoa = THE pessoa SUCH THAT CPF="XXXXXX".  
MOVE nome(pessoa) TO nomeaux.  
MOVE endereco(pessoa) TO enderecoaux.
```

No primeiro caso, usa-se COBOL padrão com chamadas através de subrotinas. Note que, naquele caso, supõe-se que exista uma área de referência, *pessoa*,

em relação à qual são efetuados os acessos, enquanto que a rotina *erget* efetivamente acessa os atributos. Em algumas implementações COBOL seria necessário deferenciar a rotina *get* de acordo com o tipo do dado acessado, seja *string*, *integer*, *real*, etc.

No segundo caso, nós supomos a existência de um pré-compilador que transforma certas construções em chamadas como no primeiro caso. Agora, tem-se ponteiros de entidades que são variáveis como *pessoa* que referenciam elementos. Desse modo, é possível exprimir termos como *nome(pessoa)* com notável clareza e facilidade. Outro exemplo seria

c) Com pré-compilador:

```
...  
77 inscricao USAGE IS ENTITY-POINTER  
  
CREATE inscricao.  
MOVE pessoa TO estudante(inscricao)  
MOVE curso TO curso(inscricao).
```

O terceiro exemplo mostra melhor a facilidade de expressão com o Modelo E-R: a criação de um elemento na instância do relacionamento INSCRIÇÃO, a partir de duas entidades referenciadas pelas variáveis *pessoa* e *curso*. Note que estas deverão ter sido inicializadas por algum comando de busca como mostrado acima. No exemplo, *estudante(inscrição)* e *curso(inscrição)* são as funções papel das entidades ESTUDANTE e CURSO respectivamente, em relação ao relacionamento INSCRIÇÃO. Dessa forma, podemos tratá-las como termos e os comandos *move* constróem a instância do relacionamento. O comando *create* inicializa a variável *inscrição* com uma referência a um novo (e vazio) elemento de INSCRIÇÃO.

## 5. FERRAMENTAS DE SUPORTE

A idéia da utilização do Modelo E-R para definição e utilização de bases de dados tem sua implementação calcada sobre um conjunto de ferramentas de software. Montados sobre interfaces similares e amigáveis, caracterizando um ambiente integrado e homogêneo, esses módulos do sistema propiciam que o usuário desfrute plenamente das vantagens da presente metodologia.

*Dicionário de Dados:* Este componente é um dicionário de dados orientado para o Modelo E-R, e tem como característica principal sua plena capacidade de representar elementos do Modelo E-R. Essa é uma vantagem sobre

outros dicionários de caráter geral. Por exemplo, poucos são os dicionários que tratam fielmente da questão de atributos herdados, nos casos de generalização.

*Diagramador E-R:* O diagramador E-R é uma ferramenta gráfica para especificação de diagramas E-R. Trabalhando com o módulo de dicionário de dados, o diagramador E-R oferece facilidades extremamente amigáveis para a definição de esquemas para bases de dados.<sup>7</sup> O sistema trabalha com menus gráficos e direciona o usuário na sua tarefa. É aqui que se configura a “face” gráfica de um sistema. A partir desse diagrama, as demais ferramentas tomam o diagrama aqui definido como base, estabelecendo a unidade visual que o sistema oferece.

*Gerador de bancos de dados:* A partir do diagrama E-R, expresso no dicionário de dados, são geradas descrições da base de dados para implementação nas instalações correntemente em uso. Da análise das construções definidas pelo usuário no diagrama E-R, este módulo gera a descrição mais adequada para implementação no SGBD objeto. No protótipo construído para esta etapa, utiliza-se o software DMSII da Unisys como referência, mas outros poderão ser considerados.

*Servidor de Bancos de Dados:* Adicionalmente, existe um servidor de banco de dados que atende às solicitações do usuário. As operações, originalmente especificadas sobre uma visão E-R do banco de dados, são traduzidas para operações da implementação corrente e atendidas por este servidor. Os programas de aplicação tem acesso ao banco de dados pela chamada a bibliotecas de funções que operam sobre o banco de dados através do servidor.

Como as estações que deverão suportar as interfaces serão de pequeno porte, de modo a se poder aproveitar o baixo custo das facilidades gráficas, o acesso às informações na implementação objeto dar-se-á através de ligações micro-mainframe.

*Módulo de Consultas:* Especificar consultas *ad hoc* tem sido um recurso pouco explorado em bancos de dados de grande porte. Embora muitas das linguagens utilizadas tenham sido orientadas para a facilidade de uso, ainda não tornou-se prática usual o uso das mesmas por usuários não técnicos. No Projeto UniversiData pretende-se explorar o ambiente da interface gráfica para que se ofereça ao usuário casual a possibilidade de especificar consultas de modo simples e intuitivo.<sup>12</sup>

A idéia é construir a expressão da consulta, isto é, sua especificação,

num processo estreitamente ligado ao diagrama E-R, e dirigido pelo sistema. Além de possibilitar facilidades para consulta, essa implementação permite que se tenha uma interface gráfica para sistemas de grande porte, como o DMS II da Unisys, por exemplo. O sistema, além da interface em si, faz uso da facilidade de acesso ao servidor de banco de dados ligado através de uma linha de comunicação ao micro, possibilitando que, da interface gráfica, se tenha pleno acesso ao banco de dados

*Gerador de Relatórios:* Constituído por um conjunto de programas total ou semi-automáticos para geração de relatórios de média e baixa complexidade. Tal estratégia resulta numa menor dependência dos programas em relação às bases de dados implementadas, aumentando o grau de liberdade do administrador da base de dados no que se refere aos aspectos de modelagem e estrutura do banco. Ou seja, pela utilização de geradores de relatórios e outras facilidades do gênero, diminui-se o impacto que as mudanças na definição do banco de dados possam causar nos programas de aplicação.

*Geradores de Aplicações:* Esta é uma parte que foi incorporada recentemente ao projeto e visa estabelecer uma conexão entre a metodologia aqui adotada e técnicas de construção de sistemas num ambiente de administração de dados. A idéia é aproveitar o ambiente de descrição das bases de dados, expressão de consultas e geração de relatórios e integrá-lo na fase de definição da aplicação em si. Essa integração faria com que a especificação do sistema se beneficiasse diretamente das funções embutidas no ambiente de apoio.

*Integração com Sistemas Especialistas:* A construção de Sistemas Especialistas<sup>2</sup> tomou impulso significativo nos anos 80 e abriu uma nova e promissora linha de implementação para uma significativa classe de aplicações. Uma das fragilidades dessas implementações, entretanto, tem sido a dificuldade que existe na maior parte das *shells* atualmente em uso de se trabalhar com estruturas de dados mais complexas. Assim, a interface de Sistemas Especialistas e bancos de dados é interessante uma vez que seria vantajoso dispor da união das duas tecnologias.

Uma das linhas do Projeto UniversiData é a formalização de uma interface padrão que contemple a integração de *shells* trabalhando com o paradigma *Objeto-atributo-valor* com bancos de dados baseados no Modelo E-R.<sup>2,10,11</sup>

## 6. CONCLUSÕES

Este artigo apresentou as bases de uma metodologia para utilização de um ambiente integrado para desenvolvimento de bancos de dados baseados no Modelo E-R. O sistema é composto de uma série de módulos que, integradamente, estabelecem e direcionam a metodologia.

A grande vantagem de se usar tal processo é o grau de uniformização e independência que se pode obter. De um lado, obtem-se uma forma de expressão comum para usuários, analistas e programadores. Com isso pretende-se diminuir os problemas de entendimento e comunicação que frequentemente ocorrem entre esses grupos. De outro lado, busca-se um certo grau de independência no que se refere ao software objeto utilizado. Note que todos os programas da aplicação assim como os procedimentos da interface estão baseados na visão E-R do banco de dados. A única fase onde as particularidades da implementação são explicitamente referenciadas fica confinada ao servidor de bancos de dados. Mudar de instalação, nesse caso, significa apenas mudar de servidor.

É claro que esta mesma vantagem pode ser vista como desvantagem de uma ótica diferente. A performance pode ficar sensivelmente degradada, uma vez que se está introduzindo mais uma camada de controle no processo. Essa perda, entretanto, deve ser satisfatoriamente compensada pela melhor produtividade de um sistema baseado em conceitos simples, claros e intuitivos como é o ambiente E-R.

### Estágio atual de desenvolvimento

O software em questão encontra-se em adiantado estado de desenvolvimento. Por ocasião do Congresso da SUCESU, em São Paulo, em 1989, parte do mesmo foi demonstrado no estande das Universidades. Para 1990, pretende-se mostrar outros componentes já desenvolvidos como a ligação *micro-mainframe*, o módulo de consultas e a extração de relatórios. Paralelamente, tem-se trabalhado no desenvolvimento de uma ferramenta de apoio à análise de sistemas e geração de programas, que se integre no ambiente aqui descrito.

Baseadas em idéias no escopo deste projeto, estão em desenvolvimento três teses de mestrado, além de vários projetos de fim de curso de Informática, orientados pelo autor. Na Seção 7 deste artigo, há uma lista de referências que contém detalhes sobre os diversos aspectos aqui apresentados.



## Extensões futuras

São virtualmente ilimitadas as possibilidades de expansão de um *software* dessa natureza. De um lado, estão as extensões de refinamento da estrutura básica aqui delineada. Novas funções, menus mais ergonômicos, construções mais poderosas. De outro lado, exploração de características como distribuição e heterogeneidade de equipamentos em bancos de dados distribuídos.

Ainda existe a possibilidade de expansão relacionada com aspectos de desenvolvimento de sistemas, suporte à análise e geração de programas.

## 7. REFERÊNCIAS

1. CHEN, P., The Entity-relationship Model - Toward a Unified View of Data, *ACM Transactions on Database Systems* 1, 1(1976).
2. HARMON, P. and KING, D., *Expert Systems*, John Wiley & Sons, Inc., USA, 1985.
3. JUNET, M., FALQUET, G. and LEONARD, M., ECRINS/86: An Extended Entity-Relationship Data Base Management System and its Semantic Query Language, *Proceedings of the Twelfth International Conference on Very Large Databases*, Kyoto, Japan, 1986.
4. LENZERINI, M. and SANTUCCI, G., Cardinality Constraints in the Entity-Relationship Model, *Proceedings of the Third International Conference on Entity-Relationship Approach*, California, USA, 1983.
5. PECKHAM, M. and MARYANSKI, F., Semantic Data Models, *ACM Computing Surveys* 20, 3(September 1988).
6. SETZER, V. W., *Projeto Lógico e Projeto Físico de Bancos de Dados*, V Escola de Computação, Belo Horizonte, 1986.
7. SILVEIRA, P. M. Projeto de Bancos de Dados Auxiliado por Computador, in *Anais do XXI Congresso Nacional de Informática SUCESU*, Rio de Janeiro, 1988.
8. SILVEIRA, P. M. A Formalization of the E-R Model, in *Anais da IX Conferencia Internacional de la Sociedad Chilena de Ciencia de la Computacion*, Santiago, 1989.
9. SILVEIRA, P. M. Procedural Data Manipulation Operations for the E-

R Model, submetido ao V Simpósio Brasileiro de Bancos de Dados, Rio de Janeiro, 1990.

10. SILVEIRA, P. M. Interfacing the E-R Model with O-A-V Expert Systems Shells, a ser publicado como Relatório Técnico NCE/UFRJ, 1990.
11. SILVEIRA, P. M. Weighted Clauses and Neural Networks, a ser publicado como Relatório Técnico NCE/UFRJ, 1990.
12. TRINKENREICH, H. e SILVEIRA, P. M. Aspectos da Implementação de Interfaces Gráficas para Bancos de Dados, a ser publicado como Relatório Técnico NCE/UFRJ. 1990.

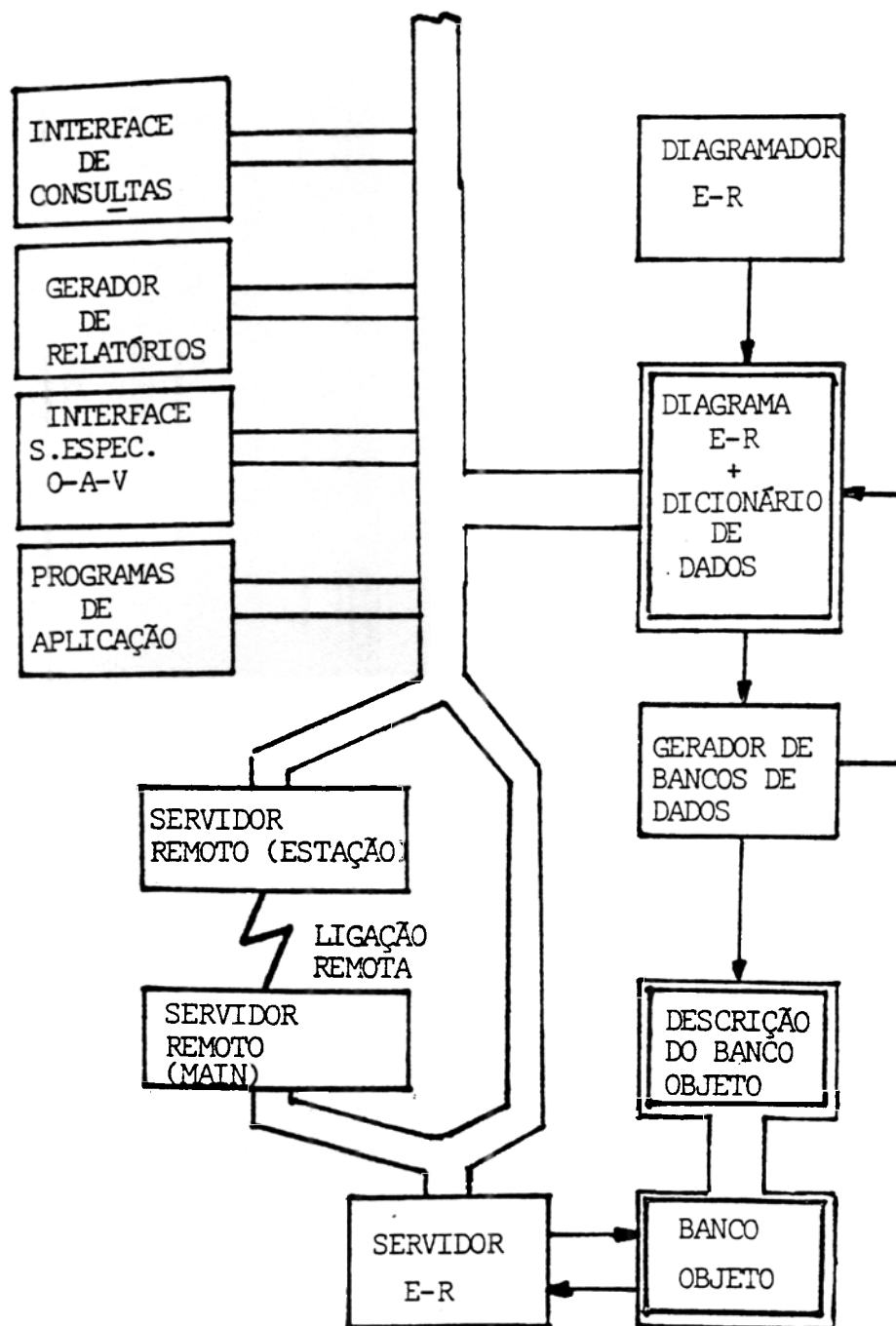


FIGURA 1. Utilização através do Modelo E-R

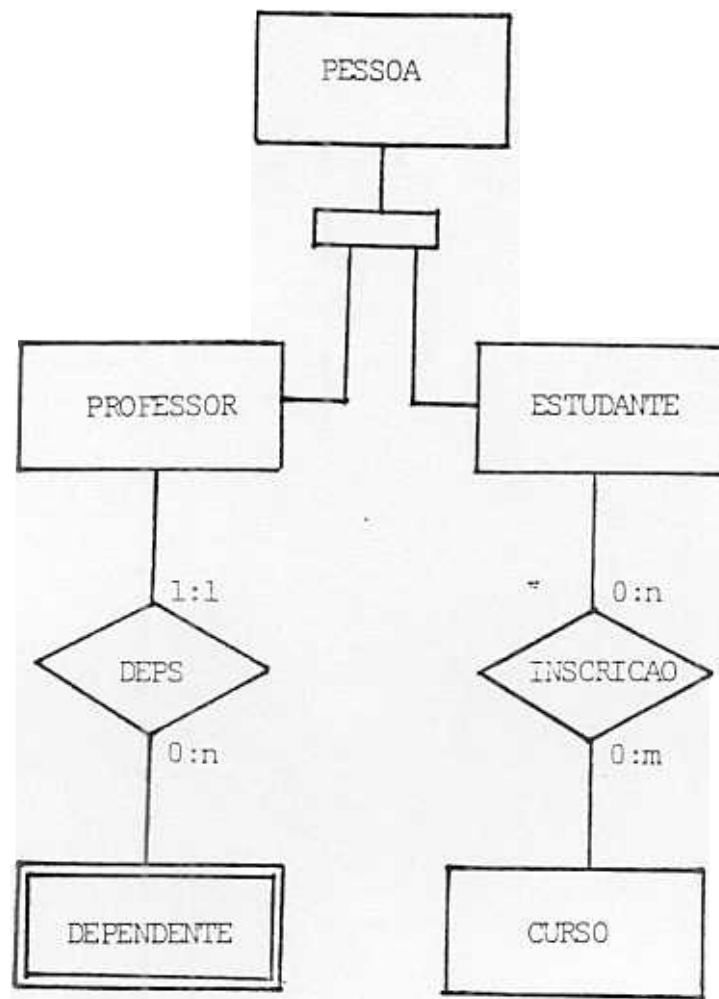


FIGURA 2. Exemplo de um esquema E-R