

UNA METODOLOGIA DE PROJETO DE BANCOS DE DADOS PARA UN SOFTWARE ESPECIFICO

V. W. Setzer

R. Lapyda

Instituto de Matemática e Estatística
Universidade de Sao Paulo, Brasil

1. INTRODUÇÃO

Um dos grandes problemas no projeto de bancos de dados usando sistemas comerciais, também chamados "pacotes", é o perigo de se limitar a visão da realidade, reduzindo-a a certas características dos modelos seguidos pelos sistemas em questão. Por exemplo, o uso de pacotes que empregam o modelo hierárquico, como o IMS /DAT 77/ e o SYS-2000 /MRI 72/, pode condicionar os usuários a encararem todas as estruturas de seu problema como hierárquicas. Ou, na melhor das hipóteses, o usuário tem consciência dessa limitação e deve introduzir no projeto de seu banco de dados uma série de artifícios, tais como arquivos auxiliares ou redundâncias, de maneira que o modelo final adapte-se às restrições ou à eficiência do pacote. O projetista pode tender a ficar muito preocupado com esses artifícios, criando um modelo que deixa de ter uma clara correspondência com a realidade. Uma consequência desse fato pode consistir na dificuldade de introduzir alterações no modelo.

Para contornar esses inconvenientes, consideramos interessante o uso de um *modelo conceitual* que não sofre restrições devidas a problemas de implementação. Ele deve ser uma abstração do mundo real em um nível puramente formal, porém com características adequadas a sistemas de informação. Nesse sentido, escolhemos o Modelo Entidade-Relacionamento, "Entity-Relationship Model" (MER), de Chen /CHE 76/ como modelo conceitual. Não entraremos em detalhes desse modelo, fazendo apenas uma breve introdução ao Diagrama Entidade-Relacionamento (DER), que constitui o cerne do MER. Serão abordadas apenas as principais características do DER. A seguir, fazemos breve introdução ao modelo de dados do sistema ADABAS /SOF 71/, inclusive com algumas considerações sobre as suas linguagens de consulta. Finalmente, apresentamos uma metodologia para conversão do DER para os arquivos ADABAS.

Na medida das possibilidades e do espaço, empregaremos uma descrição informal.

2. O MODELO ENTIDADE-RELACIONAMENTO

O modelo introduzido por Chen considera que o mundo real pode ser representado conceitualmente como uma coleção de *conjuntos de entidades (CE)* e de *conjuntos de relacionamentos (CR)* existentes entre entidades. Não se pode definir o que vem a ser uma entidade; a grosso modo, trata-se da abstração de qualquer objeto, ser, ou evento da realidade. Os relacionamentos são associações formais entre elementos de CEs. Por exemplo, podemos ter os CEs ALUNOS e DISCIPLINAS, e um CR HISTÓRICO ESCOLAR. Cada elemento deste associa um elemento do conjunto ALUNOS a um elemento do conjunto DISCIPLINAS. O DER correspondente a esses tres conjuntos seria o da fig.1.



fig.1

Nesse diagrama há uma notação adicional, que é lida $m:n$ ("m para n"), e que caracteriza o tipo do CR HISTÓRICO ESCOLAR. Nesse caso, fica explícito que um elemento de ALUNOS, isto é, um aluno, pode ser relacionado ou associado a mais do que um elemento de DISCIPLINAS e vice-versa. Isto é, interpretando o CR como representando a informação de quais alunos concluíram quais disciplinas, podemos deduzir do tipo $m:n$ que cada aluno pode ter concluído um número qualquer (não limitado) de disciplinas, e que cada disciplina pode ter sido concluída por um número qualquer de alunos.

Vamos estender a notação de Chen, introduzindo números naturais quaisquer em lugar de m e/ou n , e que pode melhor ser explicada por meio do exemplo da fig.2.



fig.2

Podemos associar ao DER da fig.2 a interpretação de que cada aluno pode estar matriculado em até 8 disciplinas, cada disciplina pode ter um número qualquer de alunos nela matriculado.

Em particular, pode-se ter CRs dos tipos $1:n$ e $1:1$. Note-se que os CRs MATRÍCULA e HISTÓRICO ESCOLAR podem coexistir, isto é, dois CEs podem ser relacionados por mais de um CR.

A cada CE ou CR pode-se associar funções (no sentido matemático) que levam cada elemento desses conjuntos a um só elemento de um conjunto de valores. Essas funções são chamadas de *atributos*; sua representação diagramática é dada na fig.3, conforme Chen /CHE 77/.

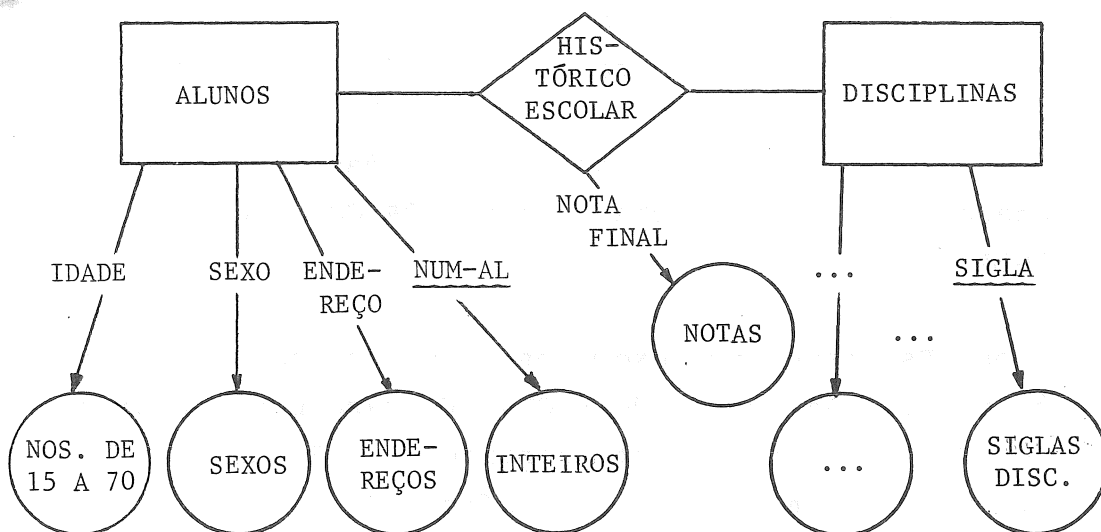


fig.3

Na fig.3, são atributos IDADE, SEXO, ENDEREÇO, NUM-AL, NOTA FINAL; são conjuntos de valores NOS.DE 15 A 70, SEXOS, ENDEREÇOS, INTEIROS, NOTAS, SIGLAS DISC. Sua interpretação pode ser entendida como representando propriedades de cada entidade ou relacionamento. Assim, IDADE é uma função que leva um elemento do conjunto de alunos a um único valor que é um número natural, atribuindo dessa maneira um determinado valor de idade para cada aluno. Da mesma forma, podemos entender NOTA FINAL como uma nota associada a cada par aluno-disciplina, representando com qual nota o aluno concluiu uma determinada disciplina. Note-se que esse atributo origina-se no CR, e não é característica (atributo) de elementos de ALUNOS isoladamente, e nem de elementos de DISCIPLINAS.

Na fig.3 os atributos NUM-AL e SIGLA são especiais, pois constituem-se em *chaves primárias*, ou simplesmente *chaves* dos CEs a que são aplicados. Uma chave é um atributo cuja função é uni-unívoca (isto é, 1 para 1). Assim, para cada elemento do CE a chave leva a um único elemento do conjunto de valores e a esse elemento só corresponde um elemento do CE segundo essa chave. Por exemplo, cada aluno tem apenas um número de aluno e vice-versa. Os valores das chaves servem, assim, para identificar os elementos do CE correspondente. Pode acontecer que uma

chave tenha que ser formada por mais do que um atributo. As chaves serão indicadas sublinhando-se os nomes dos atributos.

Os CRs também tem chaves; estas são constituídas das chaves dos CEs relacionados. Na fig.3, a chave de HISTÓRICO ESCOLAR é o par (NUM-AL, SIGLA).

Note-se que os CRs vistos nos exemplos acima são de classe *binária*, isto é, relacionam apenas dois CEs. Um caso particular de CR binário é o auto-relacionamento binário, um CR que relaciona um CE consigo mesmo, como o exemplo da fig.4.

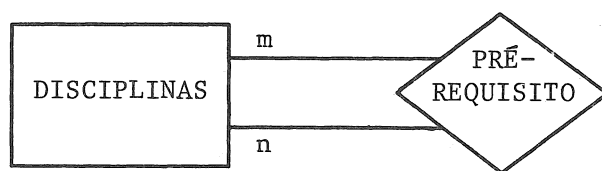


fig.4

Pode-se ter classes de CR relacionando mais do que dois CEs, como por exemplo a classe *ternária*.

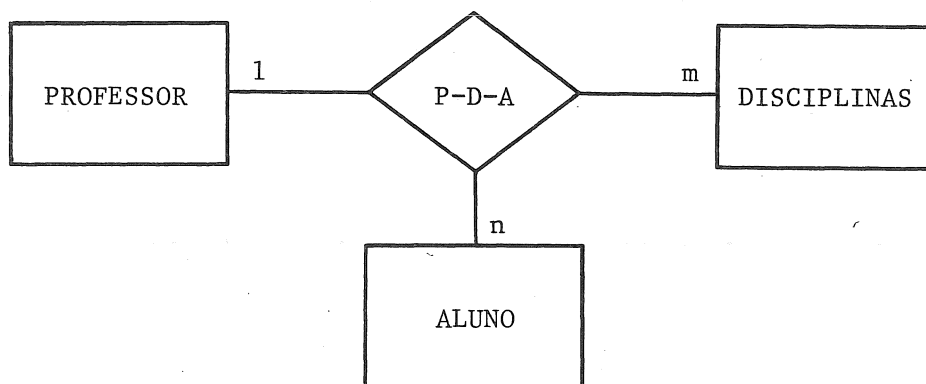


fig.5

Na fig.5 mostramos um CR ternário do tipo 1:m:n. Interpretaremos esse CR como sendo o fato de que a cada par aluno-disciplina cor responde no máximo um professor, isto é, cada aluno só pode ter um professor por disciplina. Por outro lado, um aluno pode estar cursando com um certo professor um número qualquer de disciplinas, e um professor pode ministrar uma certa disciplina a qualquer número de alunos.

Os CRs ternários podem ainda ser do tipo 1:1:n, m:n:p, etc.

Existem interessantes extensões do MER (/S-S 79/, /S-N 79/) que não serão utilizadas neste trabalho, pois é nossa intenção permanecer no mais simples nível, simplificando tanto a compreensão do método a ser exposto, quanto a sua execução em casos práticos. Também não abordaremos casos extremamente particulares, como relacionamentos do tipo exemplificado na fig.6.

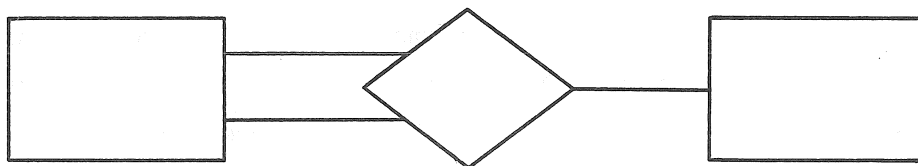


fig.6

3. O MODELO DE DADOS DO SISTEMA ADABAS

Neste item, faremos uma introdução ao sistema ADABAS, principalmente no que concerne ao seu modelo lógico de dados, isto é, como visto pelos usuários. Essa introdução será feita a partir dos conceitos vistos no MER, não se restringindo aos termos empregados normalmente em ADABAS.

3.1. E-ARQUIVOS

Aos CEs do MER faremos corresponder "files" ou arquivos, do ADABAS, e que denominaremos de "entity files" ou E-arquivos. Aos atributos dos CEs corresponderão "fields" ou E-atributos. Os conjuntos de valores do MER não podem ser totalmente especificados, sendo parcialmente definidos pelas características "standard length" e "standard format", e que denominaremos de *formato* do atributo. Por exemplo, as características do CE ALUNOS podem ser representadas em ADABAS por meio da declaração (alguns detalhes serão omitidos):

ALUNOS

01, NA, 6, F, DE	(número do aluno)
01, ID, 2, F	(idade)

01, EN, 25, A (endereço)

01, SX, 1, A (sexo)

Cada linha depois do nome ALUNOS do E-arquivo corresponde a um E-atributo. Nestes, temos especificados, pela ordem: um nível do atributo, como em COBOL ou PL/I; o identificador do atributo; o formato, composto das especificações de comprimento e tipo (F corresponde a inteiro, A a alfanumérico); o primeiro atributo contém uma especificação 'DE', que o caracteriza como um *descriptor*, passando com isso a ser um atributo que pode ser usado nas linguagens de consulta dentro dos *critérios de seleção* de registros.

Note-se que nem sempre há uma correspondência total entre os conjuntos de valores do MER e os formatos dos atributos do ADABAS. De fato, no caso do atributo ID, não é possível especificar os números naturais de 15 a 70 como constituindo o conjunto de valores desejado; da mesma maneira não é possível restringir o formato de SX somente aos valores 'M' e 'F'.

Em alguns CEs e em todos os CRs do MER, ocorrem chaves compostas de mais de um atributo. Existe uma correspondência direta destas chaves compostas em ADABAS: são os denominados "super-descriptors", que envolvem mais do que um atributo de um arquivo e funcionam, do ponto de vista dos critérios de seleção, como um único descriptor. No restante deste trabalho, consideraremos descritores em geral, sem especificar se são "super-descriptors" ou não.

3.2. CONEXÕES

Os CRs do MER serão representados por construções do ADABAS que denominaremos de *conexões*, que podem ser de duas classes: *implícitas* ou *explícitas*.

3.2.1. CONEXÕES IMPLÍCITAS

Essas conexões são implementações de relacionamentos do MER que usam apenas "couplings" ou *acoplamentos*, entre E-arquivos do ADABAS. Esses acoplamentos tem as seguintes características:

- a) Acoplamentos podem ser feitos entre dois e somente dois arquivos distintos, isto é, são sempre binários;
- b) Dois arquivos só podem ser acoplados por meio de um único acoplamento. Cada arquivo pode ser acoplado a no máximo 80 outros arquivos;
- c) Se os arquivos A_1 e A_2 estão acoplados, existe uma associação implícita entre registros de A_1 e registros de A_2 ;
- d) O acoplamento entre dois arquivos A_1 e A_2 é feito chamando-se dinamicamente uma rotina utilitária do sistema, à qual são fornecidos os identificadores de A_1 e A_2 , bem como os identificadores de um atributo de cada um, cuja declaração tenha a especificação 'DE';
- e) Se um acoplamento entre dois arquivos A_1 e A_2 é feito através dos atributos t_1 de A_1 e t_2 de A_2 , uma associação implícita é estabelecida entre cada registro de A_1 e os registros de A_2 para os quais t_1 e t_2 tem o mesmo valor, e vice-versa. A intersecção dos conjuntos de valores de t_1 e t_2 não deve, portanto, ser vazia;
- f) Em um arquivo A, um atributo t declarado simplesmente como 'DE' permite que cada registro de A tenha apenas um valor para t. Se, além de 'DE', t for declarado como 'MU' ("multiple"), cada registro de A poderá conter de zero a 191 valores diferentes de t. Esta característica do ADABAS será empregada na implementação simplificada de certos CRS do MER, como veremos no item 4.

Nas figs. 7 e 8 damos exemplos de conexões implícitas, sem atributos especificados como sendo do tipo 'MU' e, na fig.9, com essa especificação.

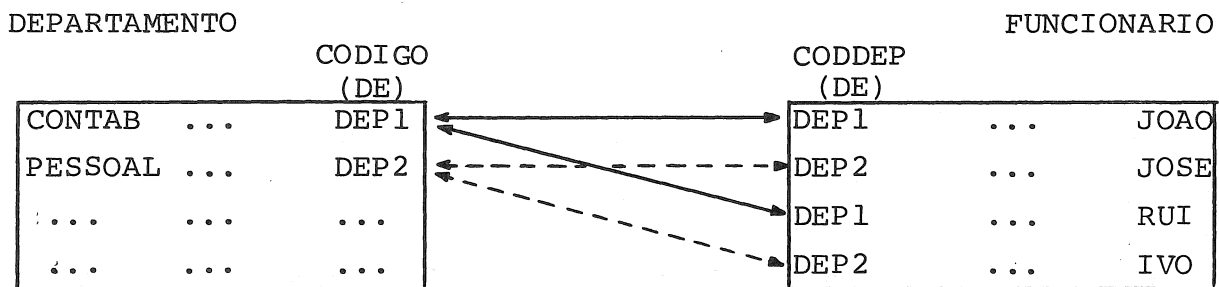


fig. 7

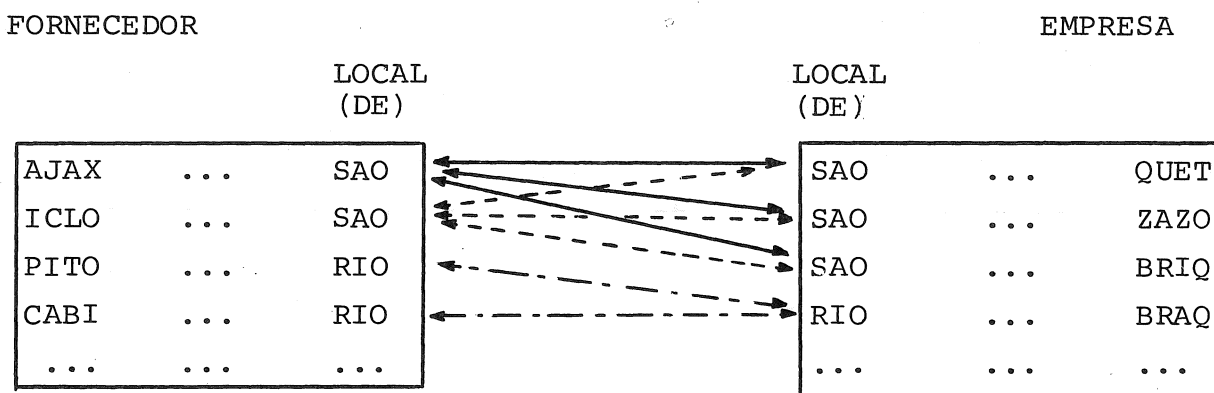


fig. 8

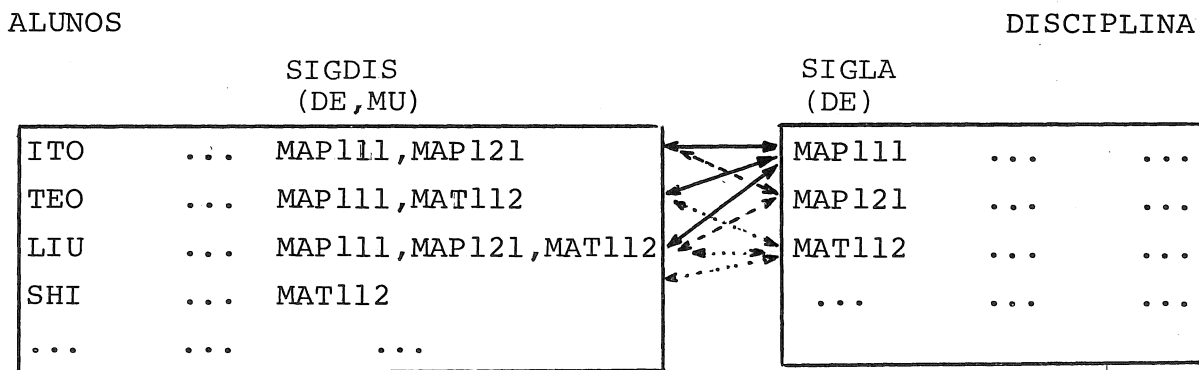


fig. 9

Note-se que na fig. 7 o atributo chave do arquivo DEPARTAMENTO foi duplicado em FUNCIONARIO para obter-se o acoplamento; o mesmo se deu na fig. 9 com o atributo SIGLA de DISCIPLINA que foi duplicado como o atributo 'MU' SIGDIS de ALUNOS. Já na fig. 8, os atributos de nome LOCAL são próprios dos arquivos FORNECEDOR e EMPRESA, não tendo havido necessidade de duplicação.

As conexões implícitas estão sujeitas às restrições dos acoplamentos, como por exemplo o fato de serem sempre binários.

3.2.2. CONEXÕES EXPLÍCITAS

Essas conexões são implementações de relacionamentos do MER pela introdução de arquivos auxiliares que denotaremos por *R-arquivos* cujos registros são formados por *R-atributos*. Para estabelecer-se uma conexão explícita entre dois arquivos E_1 e E_2 (que em geral serão E-arquivos, com uma exceção), é introduzido um R-arquivo R e dois acoplamentos, um entre R e E_1 e o outro entre R e E_2 . Para implementar esses acoplamentos, um atributo t_1 de E_1 e um atributo t_2 de E_2 são duplicados em R, constituindo os atributos r_1 e r_2 . Uma associação entre um registro de E_1 com um valor v_1 de t_1 e um registro de E_2 com um valor v_2 de t_2 é estabelecida pela introdução de um registro de R com valores de r_1 e r_2 iguais a v_1 e v_2 respectivamente. Na fig.10 damos um exemplo dessa conexão, onde E_1 corresponde a ALUNOS, E_2 a DISCIPLINAS, R a HISTÓRICO ESCOLAR, t_1 a NUM, r_1 a SIGLA, r_2 a SIGD.

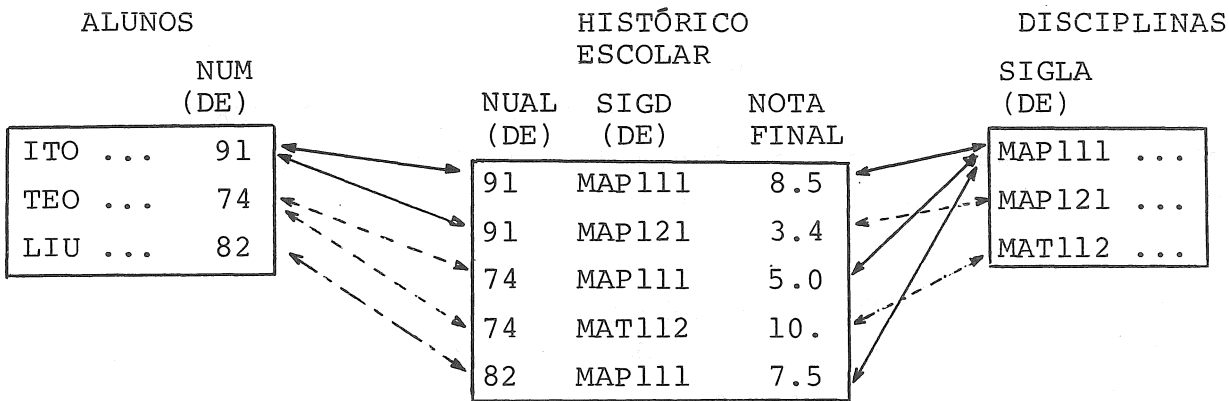


fig.10

Note-se que as conexões explícitas podem ser ternárias, ligando tres E-arquivos; deve-se duplicar um descritor de cada um destes no R-arquivo da conexão, e introduzir-se tres acoplamentos do R-arquivo para cada um dos E-arquivos. Em geral, podemos ter conexões explícitas n-árias.

3.3 LINGUAGENS DE CONSULTAS

Não entraremos aqui nos detalhes das várias linguagens de consulta do ADABAS. Importa-nos descrever uma característica fundamental da linguagem básica de consulta que usa linguagens hospedeiras ("host languages"), bem como da linguagem de consulta direta ("query language") ADASCRIP^T e da linguagem independente NATURAL devido ao impacto na simplicidade de formulação e na eficiência das consultas. Trata-se do fato de que as consultas que utilizam acoplamentos só permitirão a seleção de registros de um arquivo A, sujeitos a determinadas condições sobre atributos de A do tipo 'DE', e sujeitos a condições sobre atributos de arquivos A₁, A₂, ..., A_n acoplados cada um *diretamente* a A; essas condições devem formar uma expressão conjuntiva (isto é, do tipo "e") em relação aos vários arquivos. Exemplificando, usando ADASCRIP^T, podemos ter, para os arquivos da fig.10:

```
find all records in file HISTORICO-ESCOLAR  
      with NOTA-FINAL < 5.0 and  
      coupled to file ALUNOS with IDADE > 30  
                                and SEXO = 'M'  
      and coupled to file DISCIPLINAS  
                                with PERIODO = 'NOTURNO'
```

Note-se que supomos a existência dos atributos IDADE e SEXO em ALUNOS e PERIODO em DISCIPLINA.

Antes da palavra reservada coupled só pode ser usada a conjunção and. Assem, tanto não se pode usar outros conectivos lógicos entre acoplamentos, como não se pode *encadear* acoplamentos em uma só consulta, como por exemplo "ache registros de A acoplados a registros de B, que por sua vez estão acoplados a registros de C". Mais concretamente, seria possível formular, para os arquivos da fig.10, a consulta "obter os registros de alunos que tiveram nota final menor do que 5.0 em disciplinas do período noturno" (novamente consideramos que DISCIPLINAS contém um atributo PERÍODO).

4. CONVERSÃO DO MER PARA O ADABAS

Neste item, introduziremos um procedimento para conversão sistemática do DER para o ADABAS. Devemos, para isso, adotar um critério geral, que será o seguinte: os esquemas de conversão serão tais, que permitam buscas na linguagem básica (cujas restrições se refletem nas outras linguagens) usando, sempre que possível, somente acoplamentos, isto é, sem que seja necessário colecionar valores de algum atributo de um arquivo para posteriormente usá-los como valores de descritores na busca de registros do mesmo ou de outro arquivo. Em termos do ADABAS isto equivale a trabalhar, quando possível, apenas com as ISNs que são, essencialmente, endereços dos registros.

Serão construídos dois conjuntos: Q com arquivos e C com acoplamentos. O conjunto Q consistirá de E-, R- e X-arquivos; estes últimos serão definidos mais adiante. Suponhamos que o DER a ser convertido tenha os CEs T_i , $i=1,2,\dots,t$ e os CRs L_i , $i=1,2,\dots,l$. Os passos desse procedimento são os seguintes:

- a) *Inicialização*: faça $Q:=\Phi$ e $C:=\Phi$ onde Φ é o conjunto vazio.
- b) *Geração*: nesta fase são criados arquivos que implementam os CEs e os CRs do MER no ADABAS

- b₁) Para cada T_i do MER, $i=1,2,\dots,t$

- crie um E-arquivo E_i , com os E-atributos correspondentes aos atributos de T_i ; a chave de T_i deve ser especificada em E_i como descritor ('DE');

- faça $Q:=Q \cup \{E_i\}$, isto é, introduza E_i em Q.

- b₂) Para cada L_i do MER, $i=1, 2, \dots, l$

- b₂₁) se L_i não for um auto-relacionamento,

- crie um R-arquivo R_i , estabelecendo uma conexão explícita com os E-arquivos que representam os CEs relacionados por L_i , através dos acoplamentos $c_1^i, c_2^i, \dots, c_p^i$, supondo-se um CR p-ário. Se existirem atributos próprios de L_i , estes são introduzidos em R_i . Os atributos dos E-arquivos que são duplicados em R_i são aqueles que representam as chaves dos CEs. Portanto, se uma destas chaves envolver mais do que um atributo, é necessário de-

clará-la como "superdescriptor" (V. 3.1).

- faça $Q := QU\{R_i\}$, $C := CU\{c_1^i, c_2^i, \dots, c_p^i\}$

b_{22}) se L_i for um auto-relacionamento binário, auto-relacionando um CE T_j

- seja E_j o E-arquivo correspondente a T_j . Crie um arquivo auxiliar X_i , cujo tipo denominaremos de *X-arquivo*, tendo um só atributo x_i proveniente da duplicação do atributo e_j de E_j correspondente à chave de T_j . Os valores assumidos por x_i serão os valores de e_j . Crie um acoplamento c_o^i entre E_j e X_i ;

- crie um R-arquivo R_i estabelecendo uma conexão explícita entre E_j e X_i através dos acoplamentos c_1^i e c_2^i ; os atributos de L_i são introduzidos em R_i ;

- faça $Q := QU\{X_i, R_i\}$, $C := CU\{c_o^i, c_1^i, c_2^i\}$.

c) *Redução*: nesta fase os conjuntos Q e C são diminuídos, pela substituição de algumas conexões explícitas por conexões implícitas. Lembramos que a cada CR L_k corresponde um R-arquivo R_k , $k=1, 2, \dots, l$.

c_1) Para cada par de E-arquivos $E_i, E_j \in Q$, para os quais exista apenas uma conexão explícita binária através de R_k e dos acoplamentos c_1^k e c_2^k que usam os atributos t_i de E_i e t_j de E_j , respectivamente,

c_{11}) se L_k é do tipo 1:1

- escolha um dos E-arquivos, por exemplo E_i , segundo um ou mais dos critérios descritos abaixo, e duplique nele t_j de E_j , dando origem a t_i^j ; introduza em E_i os atributos de R_k provenientes dos atributos de L_k ;

- execute o procedimento P (descrito após c_{14}).

Critérios de escolha de E_i :

- *Critério de ausência no CR*: Seja n_p o número médio estimado de registros de E_p que não participam da conexão para $p=i, j$. Escolhe-se E_i se $n_i < n_j$.

Aplicando-se esse critério os novos atributos de E_i terão menos valores vazios do que se a duplicação fosse feita no outro sentido. Em particular, esse critério não é muito importante no ADABAS devido às possibilidades de compressão dos registros, que elimina destes os atributos com valores vazios.

- *Critério da frequência de consultas*: Para n consultas ao banco de dados, estima-se em média n_i (n_j) consultas que localizam registros de E_i (E_j) usando-se, entre outros atributos, condições sobre a chave de E_j (E_i) e/ou atributos do CR. Escolhe-se E_i se $n_i > n_j$.

Com esse critério maximizam-se as buscas que não usam acoplamento.

- *Critério de alteração da chave*: Para n consultas ao banco de dados estimam-se em média n_i (n_j) atualizações das chaves de E_i (E_j). Escolhe-se E_i se $n_i > n_j$.

Com esse critério, minimiza-se o número de consultas que alteram ambos os arquivos simultaneamente. Com isso, não só aumenta-se a eficiência, mas também evitam-se problemas de integridade.

- *Critério de estabilidade de presença no CR*: Para n consultas ao banco de dados, estimam-se em média n_i (n_j) saídas (cf. definição abaixo) de registros pertencentes a E_i (E_j) da conexão entre E_i e E_j . Escolhe-se E_i se $n_i < n_j$.

Dados dois arquivos A_1 e A_2 ligados por alguma conexão, dizemos que um registro de A_1 sai da conexão se ele, após uma operação de atualização, deixa de estar associado (cf. 3.2) a algum registro de A_2 .

Com esse critério, minimiza-se o número de registros que devem ser alterados pela movimentação de registros na conexão. Existe uma movimentação de um registro do arquivo A associado a b_1 de B segundo uma certa conexão quando ele passara associar-se a b_2 de B segundo essa conexão, deixando a associação com b_1 , $b_1 \neq b_2$.

c_{12}) se L_k é do tipo 1:n conforme a fig.11

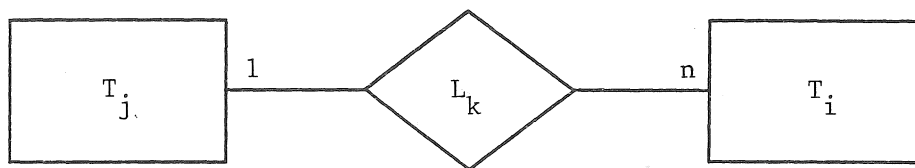


fig.11

- duplique no E-arquivo E_i o atributo t_j de E_j , dando origem a t_i^j ; introduza em E_i os atributos de R_k provenientes dos atributos de L_k ;

- execute o procedimento P.

Observe-se que há uma possibilidade particular de conversão da conexão explícita em implícita. Trata-se do caso em que L_k é do tipo $1:x$, em que $x \leq 191$ (no sentido do item 2), sem atributos. Nesse caso, poderemos optar por duplicar t_i em E_j , dando origem a t_j^i . Este último deve ser declarado com a especificação 'MU' (V. 3.2.1). Existe uma vantagem nessa organização: haverá uma diminuição do tempo de processamento de consultas que solicitam apenas os valores de t_i , isto é, a chave de T_i , usando como critério de seleção valores de atributos de E_j , pois nesse caso o acoplamento não é usado. Por outro lado, existe uma grande desvantagem nessa organização em relação à anterior, sem 'MU': nesta, ocorre um controle automático da integridade, no sentido da preservação do tipo $1:n$, enquanto que usando 'MU' essa preservação fica a cargo do usuário.

c_{13}) se L_k é do tipo $m:y$, em que $1 < y \leq 191$, sem atributos, conforme a fig.12

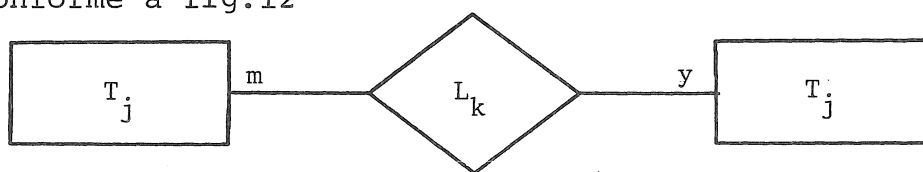


fig.12

- duplica no E-arquivo E_i o atributo t_j de E_j , dando origem a t_j^i com especificação 'MU';
- execute o procedimento P.

c_{14}) se L_k é do tipo $x:y$ em que $1 < x, y \leq 191$, sem atributos, conforme a fig.13

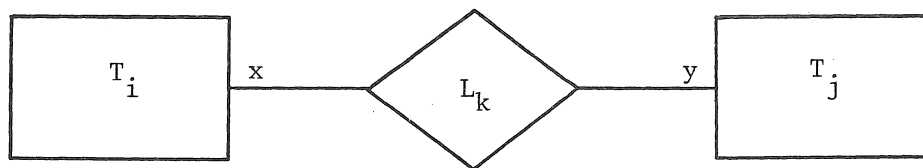


fig.13

- escolha um dos E-arquivos correspondentes a T_i e T_j , por exemplo, E_i , segundo um ou mais dos critérios descritos abaixo, e duplique nele t_j de E_j , dando origem a t_j^i com especificação 'MU'.

cação 'MU';

-execute o procedimento P.

Critérios de escolha de E_i :

-*Critério de ausência no CR*: Idem ao critério de mesmo nome apresentado em c_{11} .

-*Critério da frequência de consultas*: Idem a c_{11} , lembrando que neste caso não existem atributos do CR.

-*Critério de alteração da chave*: Para m consultas ao banco de dados, estimam-se em média m_i (m_j) atualizações das chaves de E_i (E_j); estima-se que cada registro de E_i (E_j) esteja associado em média a n_i (n_j) registros de E_j (E_i). Escolhe-se E_i se $m_i n_i > m_j n_j$.

-*Critério de movimentação no CR*: Para n consultas ao banco de dados, estimam-se em média n_i (n_j) movimentações (cf. a definição em c_{11}) de registros de E_i (E_j) na conexão correspondente ao CR L_k . Escolhe-se E_i se $n_i > n_j$.

Esse critério tem efeito análogo ao critério de estabilidade de presença de c_{11} .

-*Critério do comprimento dos registros*: Estima-se que cada registro de E_i (E_j) esteja associado em média a n_i (n_j) registros de E_j (E_i); seja s_i (s_j) o comprimento da chave t_i (t_j). Escolhe-se E_i se $n_i s_j < n_j s_i$.

Com isso, conseguem-se registros com menor tamanho, relativamente à parte variável devido ao grau de multiplicidade do atributo especificado com 'MU'.

Procedimento P:

-crie uma conexão implícita entre E_i e E_j , usando t_i^j e t_j , dando origem ao acoplamento c_0^k ;

-faça $Q := Q - \{R_k\}$, $C := C \cup \{c_0^k\} - \{c_1^k, c_2^k\}$.

c_2) Para cada par de E-arquivos $E_i, E_j \in Q$, para os quais existam duas ou mais conexões explícitas binárias $R_{k1}, R_{k2}, \dots, R_{kr}$, $r \geq 2$, correspondendo aos CRs $L_{k1}, L_{k2}, \dots, L_{kr}$ respectivamente,

c_{21}) se existir um e apenas um CR L_{kp} , $p=1, 2, \dots, r$ que satisfaz à restrição c_{1g} , $1 \leq g \leq 4$, execute com esse L_{kp} o procedimento descrito no item c_{1g} ;

Note-se que se L_{kp} tiver atributos, estes serão incorporados a um E-arquivo; esse fato pode causar problemas para outros CRs, já

que estes não terão nada a ver com os atributos de L_{kp} . Assim sendo, pode-se eventualmente decidir por ignorar o passo c_{21} em alguns desse casos.

c_{22}) se existirem CRS $L_{kp_1}, L_{kp_2}, \dots, L_{kp_q}$, $1 \leq p_h \leq r$, $q \geq 1$, $1 \leq h \leq q$, satisfazendo cada um a uma das restrições $c_{11}, \dots, c_{14}$, selecione um L_{kp_h} , segundo um ou mais dos critérios abaixo e execute o procedimento correspondente à restrição satisfeita por L_{kp_h} .

-*Critério de espaço*: Selecione o L_{kp_h} cujo R-arquivo seja estimado como o que ocupa o maior espaço de arquivamento.

-*Critério de eficiência*: Selecione o L_{kp_h} para o qual se estima que a transformação de sua conexão explícita em implícita produza o maior aumento de eficiência às consultas ao sistema.

Esse critério depende não só dos tipos de consulta, mas também da frequência relativa de cada tipo em relação aos outros tipos.

-*Critério da ausência de atributos*: Escolha um dos CRS que tenha o menor número de atributos.

Com esse critério evitam-se ou diminuem-se os efeitos indesejáveis comentados em c_{21} .

5. CONCLUSÕES

Cremos ter mostrado com este trabalho a viabilidade da utilização do MER no projeto de bancos de dados usando pacotes comerciais, tendo para isso empregado o sistema ADABAS com exemplo. É verdade que esse sistema, devido à sua simplicidade, presta-se bem a esquematizações e formalizações. Parece-nos, no entanto, que a mesma metodologia poderia ser desenvolvida para outros pacotes, eventualmente com maior dificuldade.

Por falta de espaço, deixamos de apresentar um diagrama esquemático para o ADABAS, mostrando os vários tipos de conexões entre arquivos, o que poderia ilustrar os passos dos procedimentos de conversão. Pelo mesmo motivo, deixamos de apresentar um exemplo de aplicação.

Um tópico a ser desenvolvido em detalhe seria o de mostrar que o procedimento descrito produz arquivos em ADABAS que, se vistos

como relações do Modelo Relacional /COD 70/, estão em geral em 3ª Forma Normal /COD 72/. Este não é o caso das conexões implícitas que empregam a especificação 'MU', pois aí temos relações que não estão nem em 1ª Forma Normal. Com isso, a metodologia aqui descrita corresponde, praticamente, a um projeto "Top-Down" de relações normalizadas implementadas em ADABAS. A correspondência do ADABAS com o Modelo Relacional foi abordada por /REN 76/ sem que, no entanto, fosse dada uma metodologia de estruturação do modelo.

REFERÊNCIAS

- /CHE 76/ Chen, P.P.-S. "The Entity-Relationship Model - Toward a Unified View of Data", *ACM Transactions on Data Base Systems* 1,1, Mar. 1976, pp. 9-36.
- /CHE 77/ Chen, P.P.-S. "The Entity-Relationship Model - a basis for the enterprise view of data", *Proc. AFIPS 1977 NCC*, Vol.46, AFIPS Press, Montvale, pp. 77-84.
- /COD 70/ Codd, E.F. "A Relational Model of Data for Large Shared Data Banks", *Comm. ACM* 13, June 1970, pp. 377-387.
- /COD 72/ Codd, E.F. "Further Normalization of the Data Base Relational Model", in Rustin, R.(ed.) *Data Base Systems*, Courant Computer Science Symposium 6, Prentice-Hall, Englewood Cliffs, 1972.
- /DAT 77/ Date, C.J. *An Introduction to Database Systems* (2nd ed.), Addison Wesley, Reading, 1977.
- /MRI 72/ SYS 2000 Publications: General Information Manual, Basic Reference Manual, MRI Systems Corp., Austin, 1972.
- /REN 76/ Renaud, D. "ADABAS and the Relational Model", 3rd International ADABAS User's Conference, San Francisco, May 76.
- /S-N 79/ Santos, C.S., Neuhold, E.J., Furtado, A.L. "A data type approach to the Entity-Relationship Model", PUC, Rio de Janeiro, 1979.
- /SOF 71/ ADABAS Publications: ADABAS Introduction, ADASCRIP User's Manual, ADABAS Reference Manual, NATURAL - User's Guide, Software AG, Darmstadt.
- /S-S 79/ Schiffner, G., Scheuermann, P. "Multiple views and abstractions with an Extended-Entity-Relationship Model" *Computer Languages* 4, 1979, pp. 139-154.