

# DEDUÇÃO NO PROCESSAMENTO DE CONSULTAS DOS SISTEMAS GERENCIADORES DE BANCO DE DADOS

Judith Pavón Mendoza e Edit Grassiani Lino de Campos

Departamento de Engenharia de Computação e Sistemas Digitais

Escola Politécnica da Universidade de São Paulo

Av. Professor Luciano Gualberto - Travessa 3 - 158

Cidade Universitária - Butantã

São Paulo - SP CEP 005508 - 900 - Brasil

## Palavras Chaves

Bancos de Dados, Dedução, Regras, Processador de Consultas, Modelo Entidade-Relacionamento.

## Resumo

Este trabalho tem como objetivo principal propor um mecanismo de dedução no processamento de consultas de um Sistema Gerenciador de Banco de Dados convencional. Propõe-se a utilização de uma estrutura adicional no meta-banco de dados e um algoritmo que interprete esta estrutura, de maneira a implementar a navegação através de relacionamentos entre classes em uma consulta. O mecanismo apresentado influencia a formulação de consultas em SQL, podendo simplificar a sintaxe da mesma.

## 1 INTRODUÇÃO

Nos sistemas de Inteligência Artificial (IA), especificamente nos Sistemas Especialistas (SE's), o conhecimento tem sido implementado com uma linguagem de programação lógica, como LISP ou PROLOG. O formalismo lógico é uma maneira eficaz de derivar novos conhecimentos a partir de outros, a exemplo da dedução matemática ou inferência lógica (Mylopoulos et. al., 1984 apud Luger, 1993).

As regras de produção (Hayes-Roth, 1985) são as estruturas de conhecimento mais utilizadas em implementações de SE's. Isto pode ser atribuído à excelente habilidade das regras representarem heurísticas. Os Sistemas Baseados em Regras (SBR's) utilizam a regra de inferência "modus ponens" para manipular o conhecimento. Portanto, o processo de solução de problemas, neste tipo de sistemas, consiste em realizar uma série de inferências para criar um caminho entre a definição do problema e a solução (González, 1993).

Em um Sistema Gerenciador de Banco de Dados (SGBD) convencional também é possível realizar dedução, com a utilização de linguagens de consulta, tais como Structured Query Language (SQL). Um Banco de Dados (BD) armazena informações de dois tipos: intensional e extensional (Motro, 1986; 1994). A informação intensional de um BD refere-se ao conjunto de definições de estruturas de dados de um BD. No modelo relacional, a informação intensional refere-se às definições de tabelas ou relações base, definições de visões, e as restrições de integridade; nos BD's dedutivos a informação intensional refere-se às definições de predicados, regras de inferência e restrições de integridade; e nos BD's Orientados a Objetos (BDOO's) refere-se à definição de várias classes e suas hierarquias associadas. A informação extensional de um BD refere-se ao conjunto de instâncias do BD, isto é, os próprios dados.

Neste trabalho pretende-se realizar dedução sobre o BD intensional, utilizando as informações do meta-banco de dados ou dicionário de dados (DD) para auxiliar na interpretação de consultas. O meta-banco de dados convencional contém uma série de arquivos com informações referentes às especificações de tabelas, atributos, chaves, etc., faltando, porém, informações referentes aos relacionamentos. Propõe-se uma estrutura adicional ao DD que possua informações sobre os relacionamentos existentes entre as tabelas e um algoritmo que interprete esta estrutura, de forma a implementar dedução no processamento de consultas de um SGBD convencional.

Este trabalho está organizado da seguinte forma: no item 2 são descritos alguns dos trabalhos realizados nos últimos anos referentes ao assunto tratado neste artigo, no item 3 faz-se uma análise da estrutura de um DD convencional quanto à sua adequação à inclusão de mecanismos de dedução no processamento de consultas, propõe-se uma nova estrutura adicional para tal finalidade, assim como um algoritmo que interprete a mesma. No item 4 são especificadas algumas observações em relação a proposta realizada. Finalmente, no item 5, são apresentadas as conclusões do trabalho.

## 2 PESQUISAS RECENTES

Nos últimos anos, a integração de Sistemas de Banco de Dados (SBD's) e sistemas de regras de produção tem sido abordada mais freqüentemente (Delcambre, 1988; Dayal, 1988; Pirashed, 1992; Stonebraker, 1992; Anwar, 1993; Parsaye, 1989; Von Halle, 1995; Widom,

1996; Hanson, 1996), uma vez que tal integração proveria um mecanismo uniforme para o tratamento de vários aspectos, tais como, restrições de integridade, manutenção de dados derivados<sup>1</sup>, “triggers”, “alatters”, segurança, controle de versões, entre outros.

As regras podem ser classificadas em duas categorias básicas (Van Assche, 1990): regras estáticas e regras dinâmicas. As regras estáticas são aquelas que não modificam o estado do BD; simplesmente deduzem uma nova informação através da inferência realizada sobre fatos ou dados conhecidos. As regras dinâmicas são aquelas que modificam o estado do BD toda vez que sejam satisfeitas as condições especificadas. As regras estáticas podem ter uma representação extensional, intensional ou procedimental, e as regras dinâmicas geralmente têm somente uma representação procedimental (Pavón, 1996). Frequentemente, os Sistemas Especialistas (SE's) utilizam mais as regras estáticas do que as dinâmicas, pois o objetivo principal nestes sistemas é a realização de dedução no sentido de chegar às conclusões (Schutzer, 1987). Dado este aspecto, pretende-se dar maior ênfase as regras estáticas neste trabalho.

A categoria extensional das regras estáticas refere-se às regras que são representadas através de informações extensionais do BD, correspondendo a tabelas no modelo relacional. Monarchi (Monarchi, 1992) apresenta uma proposta de representação de regras no modelo E-R de forma genérica, e a implementação das mesmas em forma de tabelas. Utiliza como exemplo o SE Mycin (Shortliffe, 1976 apud Schildt, 1989), o primeiro SE de sucesso no mercado, desenvolvido para o diagnóstico e tratamento de doenças infecciosas, onde o conhecimento médico é representado por um conjunto de regras. A dedução é realizada sobre o BD extensional, com o auxílio das tabelas construídas a partir do conhecimento do problema.

A categoria intensional das regras estáticas refere-se as regras representadas através de informações intensionais do BD. A representação dos componentes de um diagrama E-R no DD é interpretado como um conjunto de regras que pertence a esta categoria, tendo em vista que ela restringe a navegação pelo diagrama para os relacionamentos definidos no sistema, da mesma forma em que as regras em lógica de predicados permitem a inferência. Sciore (Sciore, 1994) apresenta uma forma de implementar inferência no modelo E-R utilizando o conceito de relação universal, com a finalidade de prover o processador de consultas com a capacidade de inferir os relacionamentos existentes entre as classes referidas por consultas.

Sendo as regras elementos muito importantes para o processo de inferência na área de IA, este trabalho utiliza as regras estáticas para a incorporação de mecanismos de dedução no processador de consultas de um SGBD, com o do Dicionário de Dados.

### 3 MECANISMOS DE DEDUÇÃO NO PROCESSAMENTO DE CONSULTAS

Atualmente, nos SGBD's convencionais, quando o usuário, ao formular uma consulta em SQL, indicar o que seria equivalente ao caminho a ser percorrido no diagrama E-R para obter a resposta desejada (processo de navegação). Considere-se o seguinte diagrama E-R da Figura 1.

---

<sup>1</sup> São dados temporários resultantes de uma consulta sobre um esquema conceitual ou sobre uma visão.

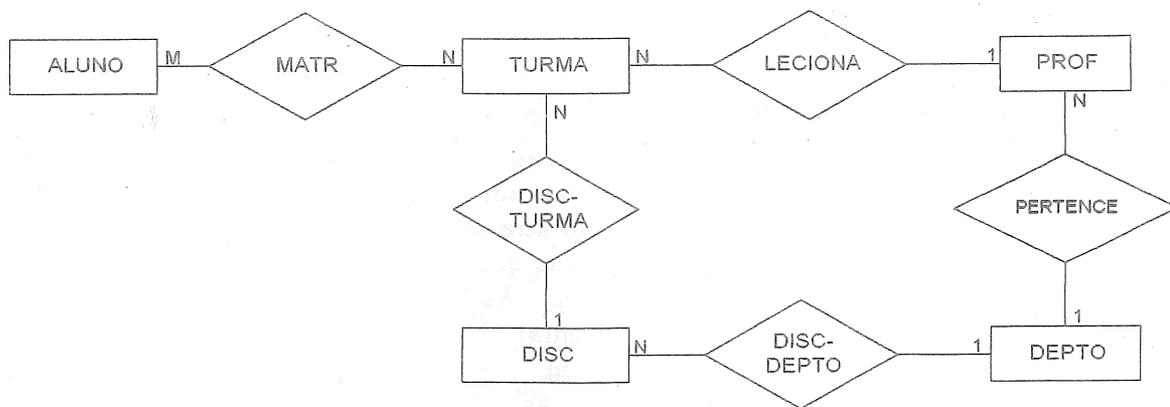


Figura 1 Exemplo de um diagrama E-R

O esquema lógico para o diagrama E-R da Figura 1 é:

ALUNO (codaluno, nome, data-nasc, datavest, endereço, cidade)

MATR (codaluno, codturma, data-matr)

TURMA(codturma, coddisc, codprof, nrosala, dia, horaini)

PROF (codprof, coddepto, nome-prof, data-adm)

DEPTO (coddepto, nome-depto)

DISC (coddisc, nome-disc, coddepto)

Considere-se a seguinte consulta, onde o usuário deseja obter os nomes de todos os alunos do professor chamado "João". Convencionalmente a formulação desta consulta em linguagem SQL seria a seguinte:

```
select ALUNO.nome
from ALUNO, MATR, TURMA, PROF
where ALUNO.codaluno = MATR.codaluno AND
      MATR.codturma = TURMA.codturma AND
      TURMA.codprof = PROF.codprof AND
      PROF.nome = "João"
```

Na consulta acima o usuário indica na cláusula "from" todas as tabelas presentes no caminho que conduz à resposta desejada. É importante salientar que este caminho é justamente o caminho a ser percorrido no diagrama E-R e que a navegação, através do diagrama, é representada pelo uso implícito de uma operação "join" entre estas tabelas.

O processamento da consulta é realizada com o auxílio das informações intensionais armazenadas no DD. No entanto, faltam informações explícitas sobre os relacionamentos existentes entre as tabelas no DD. Portanto, nos SGBD's convencionais, é necessário que o usuário conheça a estrutura do BD para formular uma consulta corretamente, de forma a indicar o caminho que conduz à resposta desejada. Seria muito útil se os SGBD's deduzissem o caminho a percorrer no diagrama E-R, a partir de um mínimo de informações fornecidas pelo usuário na consulta, isto é, fornecendo somente a(s) tabela(s) ou atributo(s) origem (ex. ALUNO.nome) e a(s) tabela(s) ou atributo(s) destino (ex. PROF.nome = "João").

Foram analisados os arquivos do DD de alguns SGBD's relacionais e identificou-se um arquivo de interesse para a implementação de dedução no processamento de consultas, o

arquivo que armazena informações sobre as chaves, tanto primárias como estrangeiras e através das chaves estrangeiras poderia-se detetar os relacionamentos entre as tabelas, mas esta operação seria muito custosa, em termos de tempo de processamento, principalmente se o diagrama E-R for complexo.

É importante ressaltar que para realizar dedução somente é necessário identificar se existe ou não um relacionamento entre as tabelas, de forma a indicar os possíveis caminhos de navegação. A seguir, é proposta uma estrutura como uma extensão do DD com a finalidade de representar o conhecimento referente aos relacionamentos entre as tabelas de uma maneira simples e direta, de forma a facilitar a realização de dedução por parte do processador de consultas.

### 3.1 Uma Estrutura de Apoio para Dedução

A extensão consiste basicamente na representação dos relacionamentos entre as tabelas num arquivo adicional ao DD convencional (arquivo de relacionamentos), aproveitando as informações armazenadas nos arquivos já existentes no DD. Na primeira linha e primeira coluna estão representados os identificadores (id) de todas as tabelas do BD. Os relacionamentos entre estas tabelas são representados pelo conteúdo de cada célula, de uma forma simples, como descrito a seguir:

- valor 1: significa que existe um relacionamento entre as tabelas identificadas na linha e na coluna correspondente.
- valor 0: significa que não existe um relacionamento entre as tabelas identificadas na linha e na coluna correspondente.
- valor -1: significa que existe um autorelacionamento.

O arquivo de relacionamentos para o exemplo da Figura 1 é o seguinte:

Tabela 1 Conteúdo do arquivo de relacionamentos para o exemplo da Figura 1.

id-tabela	201	202	203	204	205	206
201	0	1	0	0	0	0
202	1	0	1	0	0	0
203	0	1	0	1	0	1
204	0	0	1	0	1	0
205	0	0	0	1	0	1
206	0	0	1	0	1	0

Os identificadores apresentados acima correspondem às seguintes tabelas base:

- 201 - tabela ALUNO
- 202 - tabela MATR
- 203 - tabela TURMA
- 204 - tabela PROF
- 205 - tabela DEPTO
- 206 - tabela DISC

No exemplo dado, o relacionamento existente entre TURMA e PROF, representado no diagrama E-R como o relacionamento LECIONA, é implementado com uso de chave estrangeira, portanto, o arquivo de relacionamentos não inclui LECIONA explicitamente, existindo, no entanto, o valor 1 na célula correspondente à interseção da linha e coluna correspondentes às tabelas TURMA e PROF.

### 3.2 Algoritmo para Dedução

Um algoritmo interpreta o arquivo de relacionamentos de forma a conduzir o processo de navegação através dos relacionamentos. Este algoritmo faz parte do processador de consultas, que inicialmente executa o pré-processamento da consulta, fornecendo ao algoritmo, como entrada, a identificação das tabelas envolvidas.

Foi escolhida a estrutura de árvores para representar a navegação através do diagrama E-R. Em termos gerais, o algoritmo baseia-se no percurso de uma árvore de possíveis caminhos, onde cada nó representa uma tabela do BD. Esta árvore é construída à medida que são encontrados os relacionamentos existentes entre as tabelas de origem e destino final da consulta, com a ajuda do arquivo de relacionamentos, finalizando-se o processo quando se encontra o caminho que corresponde à resposta da consulta. O algoritmo se processa no sentido de encontrar o caminho mais curto.

O algoritmo consiste basicamente dos seguintes passos:

1. Inicia-se a construção da árvore de possíveis caminhos com a primeira tabela que aparece na sentença "SELECT" da consulta formulada, expressa em SQL.

2. A partir deste nó inicial, criam-se os ramos da árvore, levando em consideração os relacionamentos existentes entre as tabelas (arquivo de relacionamentos), com o intuito de encontrar um caminho até a tabela destino.

3. A medida que são criados os nós, verifica-se se o nó criado já não consta num mesmo ramo da árvore, o que significa, em termos de navegação, que está-se retornando a um caminho já percorrido. O autorelacionamento neste contexto é uma exceção, isto é, o algoritmo permite retornar uma vez a um nó já criado se existir um autorelacionamento.

Uma vez encontradas todas as tabelas de interesse num mesmo ramo ou espalhadas em vários ramos, o processo de busca continua até encontrar todos os caminhos possíveis à resposta. Se existir mais de um caminho que conduz à resposta à consulta, se procede à estratégia de escolha de um caminho, procedimento que será descrito no item 4, mas, se não for definida nenhuma estratégia, o algoritmo assume o caminho mais curto. No final, o algoritmo fornece como saída os identificadores das tabelas que compõem o caminho escolhido, deixando a cargo do processador de consultas realizar as operações de "join" entre tais tabelas para obter os dados desejados.

### 3.3 Exemplos

Tomando o diagrama E-R Figura 1 como referência, a respectiva árvore de possíveis caminhos da consulta formulada no item 3 é apresentada na Figura 2.

A representação da árvore de possíveis caminhos da consulta formulada no item 3 é apresentada na Figura 2.

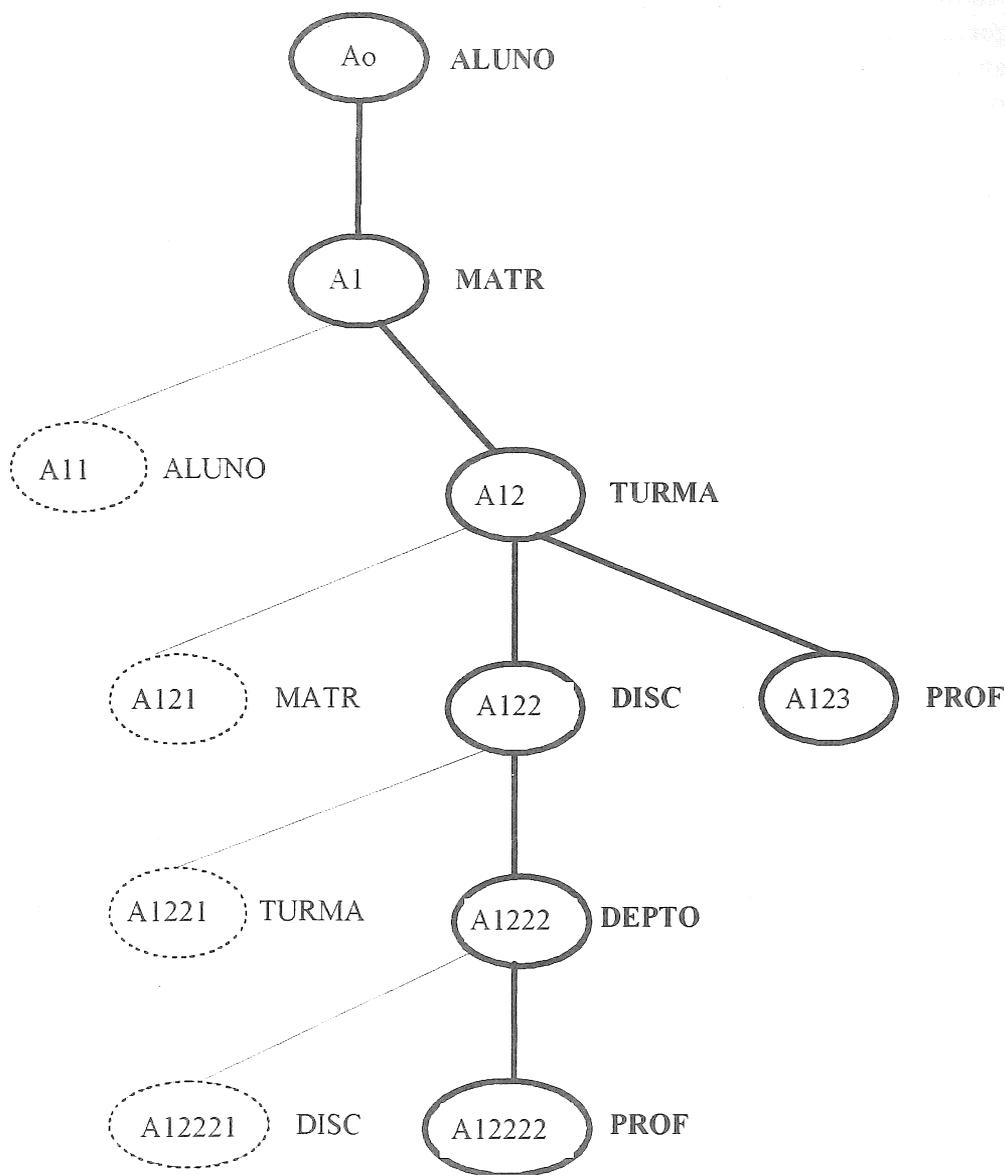


Figura 2 Representação da navegação numa árvore de possíveis caminhos.

As tabelas de interesse neste caso são ALUNO e PROFESSOR. O algoritmo inicia a árvore, pela criação do nó raiz (Ao), onde armazena o código da primeira tabela de interesse, que corresponderia à tabela ALUNO. A seguir, procura no arquivo de relacionamentos todas as ligações existentes entre esta tabela e as demais, de maneira a traçar os caminhos possíveis; neste caso, foi encontrada somente uma tabela (MATR), representada por um nó. Em seguida, parte-se de novo à busca dos relacionamentos de MATR com outras tabelas, e assim sucessivamente, até encontrar todas as tabelas de interesse. Cada vez que é criado um nó, verifica-se se não está se

retornando a um caminho já percorrido; ao mesmo tempo, também é verificado se este nó corresponde a alguma tabela de interesse. A Figura 2 mostra a árvore completa, onde os ramos que representam os caminhos possíveis aparecem em negrito. Os nós apresentados com círculos pontilhados são os retornos por um caminho já percorrido; portanto, eles são identificados de forma a que o algoritmo não os referencie novamente no processo de busca. Observa-se, portanto, que a mesma consulta pode ser respondida de várias formas, dependendo de quantos caminhos alternativos são encontrados.

#### 4 ESTRATEGIA DE ESCOLHA DE UM CAMINHO

Existem casos em que podem ocorrer consultas ambíguas, isto é, consultas em cujas árvores existe mais de um caminho que leva à resposta desejada. Maier (Sciore, 1994) define que, na linguagem PIQUE, proposta para um modelo de Instância Universal (Association-Object Data Model), uma consulta ambígua é interpretada como a união de todos os caminhos possíveis, isto é, a união de todas as classes presentes nestes caminhos. Sciore (Sciore, 1994) propõe uma técnica que rejeita consultas ambíguas e apresenta duas soluções alternativas: que o usuário indique algumas classes num determinado caminho (“name drop”) de maneira a sugerir ao processador de consultas um caminho possível e a outra alternativa é definir caminhos prioritários, sendo que, o caminho de prioridade mais alta é o mais curto. Observe-se que esta colocação nem sempre é correta, existindo situações nas quais o caminho mais curto não conduz à resposta correta para a consulta formulada.

O algoritmo apresentado neste trabalho assume o caminho mais curto se não definir nenhuma estratégia, mas, também existe a possibilidade de permitir a intervenção do usuário, para fornecer dados adicionais que permitam indicar um caminho que conduza à resposta desejada entre os vários encontrados

É importante salientar, que muitas vezes o caminho mais curto não é o de menor custo, considerando o processo de otimização de consultas realizado normalmente por um processador de consultas, sendo que, as tabelas envolvidas na junção do caminho mais curto podem ser de um tamanho maior do que as envolvidas em um outro caminho mais cumprido. Os dados para definir a melhor estratégia para escolher o caminho de menor custo, encontram-se no DD convencional o metabanco de dados, tais como, o número de tuplas de uma determinada relação, tamanho de uma tupla de uma determinada relação e outras informações de interesse para a estimação de custo de processamento de consultas. Atualmente, o algoritmo está sendo adaptado para achar o caminho de menor custo utilizando estas informações do metabanco de dados.

#### 5 IMPACTO SOBRE CONSULTAS SQL

A inclusão da capacidade de dedução no processamento de consultas, tem, como consequência, um aspecto interessante, que é uma possível simplificação das consultas formuladas em SQL. Uma vez que o processador de consultas, desde que possua a capacidade de dedução, encontra os caminhos que levam à resposta, o usuário, em princípio, não precisaria especificar a cláusula “from”, onde normalmente seriam definidas as tabelas envolvidas no “join”.

Assim, a consulta: "Listar os nomes de todos os alunos do professor chamado João' ", poderia ser formulada da seguinte forma:

```
select ALUNO.nome  
where PROF.nome = "João"
```

É claro que, neste caso, seria necessário indicar apenas as tabelas e atributos de interesse (tabelas e atributos de origem e destino), reduzindo desta forma a quantidade de sentenças na especificação de uma consulta. Esta maneira de formular a consulta não implica o conhecimento prévio da estrutura do esquema do BD ou do conteúdo do DD.

## 6. CONCLUSÕES

Neste trabalho foi analisado o conteúdo do meta-banco de dados ou DD de um BD convencional em relação à sua adequação na inclusão de mecanismos de dedução no processamento de consultas; a partir desta análise propõe-se uma estrutura adicional para esse fim e um algoritmo para interpretá-la. Desta forma, apresenta-se uma alternativa para realizar dedução sobre BD's intensionais, utilizando especificamente as regras estáticas, descritas no meta-banco de dados referentes aos relacionamentos existentes entre as tabelas. Esta proposta simplifica a formulação de consultas, no sentido que o usuário somente precisa especificar, na consulta, as tabelas e atributos de interesse.

Atualmente o algoritmo finaliza o processo de busca somente ao encontrar todos os caminhos possíveis à resposta, assumindo o caminho mais curto como resposta à consulta, mas está sendo adaptado para permitir a intervenção do usuário através de uma interface gráfica, de maneira a indicar o caminho mais apropriado ("name drop"), e também está sendo adaptado para que o próprio algoritmo possa escolher a melhor estratégia para encontrar o caminho de menor custo.

## REFERÊNCIAS BIBLIOGRAFICAS

- ANWAR, E; MAUGIS, L.; CHAKRAVARTHY, S. A new perspective on rule support for object-oriented databases. In: INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA. SIGMOD'93. Washington, 1993. p.99-108.
- DAYAL, U.; et. al. The HIPAC project: combining active databases and timing constraints. *Sigmod Record*, v.17, n.1, p.51-70, Mar. 1988.
- DELCAMBRE, L.M.L.; ETHEREDGE, J.N. A self-controlling interpreter for the relational production language. In: INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA. SIGMOD'88. Chicago, 1988, p.396-403.
- FREUNDLICH, Y. Knowledge bases and databases. *IEEE Computer*, p.51-7, Nov. 1990.
- GONZALEZ, A. J.; DANKEL, D. D. *The engineering of knowledge-based systems: theory and practice*. Englewood cerffs, Prentice-Hall, 1993.
- HANSON, E. N. The design and implementation of the Ariel active database rule system. *IEEE Transactions on Knowledge and Data Engineering*, v.8, n.1, p.157-72, Feb. 1996.

- HAYES-ROTH, F. Rule-based systems. *Communications of the ACM*, v.28, n.9, p.921-32, Sept. 1985.
- LUGER, G. F.; STUBBLEFIELD, W. A. *Artificial intelligence*. Benjamin/Cummings, 1993.
- MONARCHI, D. E.; SMITH, J. R. The representation of rules in the ER model. *Data & Knowledge Engineering*, v.9, n.1, p.45-61, Oct. 1992.
- MOTRO, A. SEAVE: A mechanism for verifying user presuppositions in query systems. *ACM Transactions on Office Information Systems*, v.4, n.4, p. 312-330, Oct. 1986.
- MOTRO, A. Intensional answers to database queries. *IEEE Transactions on Knowledge and Data Engineering*, v.6, n.3, p.444-54, June. 1994.
- PARSAYE, K.; CHIGNELL, M.; KHOSHAFIAN, S.; WONG, H. *Intelligent databases*. New York, John Wiley, 1989.
- PAVON, J. *Representação de Estruturas de Conhecimento em Sistemas de Banco de Dados*. São Paulo, 1996. 119 p. Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo.
- PIRASHED, H.; HELLERSTEIN, J. M.; HASAN, W. Extensible/rule based query rewrite optimization in Starburst. In: *INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA. SIGMOD'92*. Berkeley, p.39-48, May 1992.
- SCHILDT, H. *Inteligência Artificial Utilizando Linguagem C*. Trad. de Cláudio Gaiger Silveira e Mônica Soares Rufino. São Paulo, McGraw-Hill, 1989.
- SCHUTZER, D. *Artificial intelligence: an applications-oriented approach*. New York, Van Nostrand Reinhold, 1987.
- SCIORE, E. Query abbreviation in the entity-relationship data model. *Information Systems*. v.19, n.6, p.491-511, July 1994
- STONEBRAKER, M. The integration of rule systems and database systems. *IEEE Transactions on Knowledge and Data Engineering*, v.4, n.5, p.415-23, Oct. 1992.
- VAN ASSCHE, F.; LOUCOPOULOS, P.; SPELTINX, G.; VENKEN, R. A declarative approach to the dynamics of information systems. In: *ARTIFICIAL INTELLIGENCE IN DATABASES AND INFORMATION SYSTEMS (DS-3)*. *IFIP'90*. Amsterdam, 1990. p. 567-74.
- VON HALLE, B. Digging for business rules. *Database programming & design*, p.11-3, Nov. 1995.
- WIDOM, J.; FINKELSTEIN S. A syntax and semantics for set-oriented production rules in relational database systems. *IEEE Transactions on Knowledge and Data Engineering*, v.8, n.4, p.583-95, Aug. 1996.