

FeST - Uma Ferramenta para Projeto Conceitual de Banco de Dados

Marco A. Casanova¹, Alexandre P. de Carvalho¹, Lorenzo F.G.G.M. Ridolfi^{1,2}

¹Centro Científico Rio
IBM Brasil
Caixa Postal 4624
20.001, Rio de Janeiro, RJ - Brasil

²Departamento de Informática
Pontifícia Universidade Católica do RJ
R. Marquês de S. Vicente, 225
22.453, Rio de Janeiro, RJ - Brasil

1. CARACTERÍSTICAS DA FERRAMENTA

Este trabalho apresenta uma ferramenta para auxiliar o projeto de banco de dados relacionais chamada FeST ("Ferramenta para Síntese de Tabelas Relacionais) com as seguintes características:

- especificação de esquemas conceituais através de duas linguagens alternativas, denominadas FeST-ERE e FeST-SDM, baseadas em variantes do modelo entidade-relacionamento [10];
- mapeamento automático do esquema conceitual em um esquema lógico relacional, descrito em SQL, incluindo a geração de todas as chaves primárias e estrangeiras necessárias ao tratamento de integridade referencial, conforme implementada no sistema de gerência de bancos de dados DB2 [5];
- otimização do esquema lógico através da minimização do número de tabelas e do número de chaves estrangeiras;
- controles adicionais sobre o processo de otimização através de comandos específicos;
- geração automática de visões para oferecer uma representação relacional bem próxima ao esquema conceitual original, e para capturar a herança de atributos entre conjuntos de entidades em hierarquias de especialização.

A ferramenta FeST compõe-se de quatro módulos, a saber: os analisadores sintáticos para as linguagens FeST-ERE e FeST-SDM, o otimizador e o gerador de código. Os analisadores sintáticos encarregam-se da validação sintática do esquema conceitual definido pelo projetista, de acordo com a linguagem escolhida. O otimizador, por sua vez, tenta reduzir o número de chaves estrangeiras e tabelas, combinando em uma única tabela a representação de um esquema de entidade com a representação de outros esquemas de entidade ou de relacionamento. Finalmente, o gerador de código produz os comandos para definição das tabelas, índices e visões.

A linguagem FeST-ERE é um subconjunto da linguagem para modelagem conceitual definida em [3], o processo de otimização do esquema conceitual baseia-se nos algoritmos publicados em [4,6,11] e a geração de código segue em linhas gerais as idéias contidas em [2]. Por fim, a ferramenta FeST encontra-se descrita em mais detalhe em [1,7,8].

```

define schema ERE ACADEMICO
  define domain TIPO_NOME      char(30) not null
  define entity ALUNO
    attributes NUM_ALUNO      integer not null,
               NOME           TIPO_NOME
    key        NUM_ALUNO
  define entity PROFESSOR
    attributes NUM_PROF       integer not null,
               NOME           TIPO_NOME
    key        NUM_PROF
  define relationship ORIENTA
    over        PROFESSOR     cascade,
               ALUNO         cascade
    identifier ALUNO
endschema
define internal schema
  drop
endschema

```

Exemplo de um esquema conceitual na linguagem FeST-ERE

TIPO_NOME a ser usado por todos os atributos referentes a nomes de pessoas. Os comandos seguintes definem os conjuntos de entidades ALUNO e PROFESSOR e um conjunto de relacionamentos entre ambos, ORIENTA. Note que o quarto comando indica que ALUNO é identificador de ORIENTA, o que significa que cada aluno possui um único orientador. Este comando define ainda, através da palavra-chave cascade, que a remoção de um professor (ou de um aluno) deve se propagar para a remoção de todos os relacionamentos que possam envolver o professor (ou do relacionamento que possa envolver o aluno).

O único comando do esquema interno especifica que o gerador de código deve incluir um comando DROP TABLE T antes de criar uma tabela T (e, de forma semelhante, comandos DROP antes da criação de índices e visões).

Para o esquema conceitual da Figura 1, a ferramenta FeST gera o esquema relacional em SQL mostrado no Apêndice. O esquema relacional contém apenas duas tabelas, uma representando o esquema de entidade PROFESSOR e outra representando o esquema de entidade ALUNO juntamente com o esquema de relacionamento ORIENTA. Esta segunda tabela possui uma chave estrangeira referenciando a primeira para capturar corretamente a semântica de ORIENTA. O esquema relacional também contém duas visões, correspondendo a ALUNO e ORIENTA, para que o usuário possa ter acesso a estes dados em separado. A ferramenta produz ainda um relatório explicativo da tradução, omitido aqui por razões de espaço.

```

define schema SDM ACADEMICO

  define domain TIPO_NOME      char(30) not null

  define entity PROFESSOR
    attributes NUM_PROF        integer not null,
               NOME            TIPO_NOME
    key        NUM_PROF

  define entity ALUNO
    depends    PROFESSOR as ORIENTADOR,
    attributes NUM_ALUNO       integer not null,
               NOME            TIPO_NOME
    key        NUM_ALUNO

endschema

```

Figura 2: Exemplo de um esquema conceitual na linguagem FeST-SDM

3. A LINGUAGEM FEST-SDM

A principal característica da linguagem FeST-SDM consiste em combinar os conceitos de entidade e de relacionamento em um único conceito. Mais precisamente, um conjunto de entidades E na linguagem FeST-SDM possui um nome, um conjunto de atributos com seus respectivos domínios e uma lista de chaves. O conjunto E pode ainda ser definido como uma especialização de outros, dos quais herda todos os atributos e chaves, como na linguagem FeST-ERE. Porém, além destas características, E pode estar associado a uma lista de conjuntos de entidades, chamados de *pais* de E . Neste caso, E define também, implicitamente, relacionamentos, no sentido tradicional.

O esquema conceitual da Figura 2 ilustra o uso da linguagem FeST-SDM. O primeiro comando define o domínio TIPO_NOME a ser usado por todos os atributos referentes a nomes de pessoas. Os dois comandos seguintes definem os conjuntos de entidades PROFESSOR e ALUNO. Note que ALUNO depende de PROFESSOR no papel de ORIENTADOR e, assim, implicitamente define um conjunto de relacionamentos entre PROFESSOR e ALUNO (intuitivamente interpretado neste exemplo como o relacionamento de *ser orientado de*).

4. CONCLUSÕES

Em resumo, a ferramenta FeST oferece como vantagens: (1) legibilidade e concisão na especificação do esquema conceitual; (2) eliminação da tradução manual para a linguagem SQL, incluindo geração automática dos comandos para integridade referencial; e (3) otimização de código, incorporando algumas heurísticas de mapeamento comumente empregadas.

A ferramenta encontra-se disponível em duas versões, uma para VM/CMS e outra para o PC-DOS, inteiramente compatíveis entre si. Na versão VM/CMS, o usuário pode ainda optar pela diagramação do esquema conceitual através do editor gráfico

EDIGRAF [9], que se encarrega de gerar automaticamente a especificação do esquema na linguagem FeST-ERE ou FeST-SDM.

A ferramenta FeST, como se apresenta hoje, passou por três versões preliminares, sendo o resultado de um projeto iniciado no Centro Científico Rio em 1988, em sucessão ao projeto CHRIS [11]. Além dos autores deste trabalho, participaram em épocas diferentes do projeto as seguintes pessoas: Luiz Tucherman, Anelise Pacheco, Maurício Roma Cavalcanti, Alvan Figueiredo, Pedro Mendes Gualandi e José Carlos M. Magno.

REFERÊNCIAS

- [1] Carvalho, A.P., Casanova, M.A., Ridolfi, L.F.G.G.M., "Especificação de uma Ferramenta para Projeto Conceitual de Bancos de Dados", Relatório Técnico CCR132, Centro Científico Rio, IBM Brasil (Novembro 1991).
- [2] Casanova, M.A., Carvalho, A.P., Ridolfi, L.F.G.G.M., Laender, A.H.F., "An analysis of table constraints in SQL2 based on the entity-relationship model", Proc. 10th Int'l. Conf. on the Entity-Relationship Approach, San Mateo (Out. 1991).
- [3] Casanova, M.A., Tucherman, L., Gualandi, P.M., Pacheco, A. and Cavalcanti, M.R., "A data definition language for an extended entity-relationship model", Relatório Técnico CCR072, Centro Científico Rio, IBM Brasil (Jul. 1989).
- [4] Casanova, M.A., Tucherman, L. and Laender, A.H.F. "Algorithms for designing and maintaining optimized relational representations for entity-relationship schemas", Proc. 9th Int'l. Conf. on Entity-Relationship Approach, Switzerland (Out. 1990).
- [5] IBM, "IBM DATABASE 2 Version 2 - SQL Reference Release 2".
- [6] Pacheco, A., "Tradução correta de esquemas entidade-relacionamento em esquemas relacionais", Anais do 9º Congresso da Sociedade Brasileira de Computação, Uberlândia (Jul. 1989).
- [7] Ridolfi, L.F.G.G.M., Carvalho, A. P., Casanova, M. A., "Uma Ferramenta para Projeto Conceitual de Bancos de Dados baseada no Modelo Entidade-Relacionamento", Relatório Técnico CCR130, Centro Científico Rio, IBM Brasil (Novembro 1991).
- [8] Ridolfi, L.F.G.G.M., Cavalcanti, M.R., Carvalho, A.P., "Implementação de uma Ferramenta para Projeto Conceitual de Bancos de Dados", Relatório Técnico CCR133, Centro Científico Rio, IBM Brasil (Novembro 1991).
- [9] Teixeira, R. R., "Editor Gráfico Semântico no apoio a ambientes de desenvolvimento", Relatório Técnico CCR127, Centro Científico Rio, IBM Brasil (Jul. 1991).
- [10] Teorey, T.J., Yang, D. and Fry, J.P., "A logical design methodology for relational databases using the extended entity-relationship model", ACM Computing Survey 18:2 (Jun. 1986).
- [11] Tucherman, L., Casanova, M.A., Furtado, A.L., "The CHRIS consultant - A tool for database design and rapid prototyping", Information Systems Vol. 15., No. 2 (May 1990).

Apêndice - Tradução para SQL do Esquema Conceitual da Figura 1

DROP TABLE T_ALUNO

```
CREATE TABLE T_ALUNO (
  NUM_ALUNO      INTEGER          NOT NULL,
  NOME           CHAR(30)         NOT NULL,
  NUM_PROF       INTEGER          )
```

DROP INDEX I1_ALUNO

```
CREATE UNIQUE INDEX I1_ALUNO ON T_ALUNO
  ( NUM_ALUNO )
```

DROP TABLE PROFESSOR

```
CREATE TABLE PROFESSOR (
  PRIMARY KEY ( NUM_PROF ),
  NUM_PROF     INTEGER          NOT NULL,
  NOME         CHAR(30)         NOT NULL )
```

DROP INDEX I1_PROFESSOR

```
CREATE UNIQUE INDEX I1_PROFESSOR ON PROFESSOR
  ( NUM_PROF )
```

```
ALTER TABLE T_ALUNO
  FOREIGN KEY ( NUM_PROF )
    REFERENCES PROFESSOR          ON DELETE SET NULL
```

DROP VIEW ALUNO

```
CREATE VIEW ALUNO (
  NUM_ALUNO,
  NOME
  AS SELECT  T_ALUNO.NUM_ALUNO,
            T_ALUNO.NOME
  FROM      T_ALUNO )
```

DROP VIEW ORIENTA

```
CREATE VIEW ORIENTA (
  NUM_ALUNO,
  NUM_PROF
  AS SELECT  T_ALUNO.NUM_ALUNO,
            T_ALUNO.NUM_PROF
  FROM      T_ALUNO
  WHERE     T_ALUNO.NUM_PROF          IS NOT NULL )
```
