

# QUICK-DB: Um Ambiente para Projeto e Manipulação de Bases de Dados

PEDRO MANOEL SILVEIRA

*Núcleo de Computação Eletrônica — UFRJ  
Caixa Postal 2324, Cep 20001, Rio de Janeiro, RJ, Brasil  
E-mail: pedro@ufrj.bitnet*

## RESUMO

Este trabalho apresenta QUICK-DB, um ambiente para projeto, manutenção e manipulação de bancos de dados construídos sobre uma extensão do modelo Entidade/Relacionamento. QUICK-DB é composto por um conjunto de ferramentas de software que o tornam prático, simples e didático no desenvolvimento de aplicações de bancos de dados. Estas ferramentas incluem diagramador gráfico, dicionário de dados, folheador, interface para regras de produção, gerador de aplicações, servidor de bancos de dados e gerador de esquemas.

## ABSTRACT

This article presents QUICK-DB, a graphical environment for project, maintenance and manipulation of databases built over an extension of the Entity/Relationship Model. A set of software tools make QUICK-DB simple, practical and comprehensive for the development of database applications. These tools include the schema drawer, data dictionary, browser, expert systems shell, program generator, database server and database schema generator.

**1. INTRODUÇÃO.** Este trabalho apresenta os fundamentos de QUICK-DB, um ambiente completo para projeto, manutenção e manipulação de bancos de dados construídos sobre o Modelo Entidade/Relacionamento. QUICK-DB é composto por um conjunto de ferramentas de software que o tornam prático, simples e didático no desenvolvimento de aplicações de bancos de dados.

QUICK-DB está baseado na idéia de um ambiente uniforme e unificado, gráfico, onde todos os componentes referem-se a uma extensão do Modelo E-R. Este enfoque permite que seus usuários sigam uma linha de trabalho consistente, simplificando a metodologia e evitando a introdução de conceitos estranhos ao modelo básico.

**2. FACILIDADES.** Pela utilização de QUICK-DB é possível:

- Definir diagramas do tipo Entidade-Relacionamento completos, incluindo extensões como atributos referenciais, atributos virtuais, acesso ordenado, generalização;
- Verificar diagramas;
- Construir automaticamente um dicionário de dados durante a especificação do diagrama;
- Gerar descrições de bancos de dados equivalentes aos diagramas para implementação em sistemas de grande porte;
- Gerar bibliotecas de objetos e métodos para acesso aos bancos de dados descritos nos diagramas;

- Definir sistemas de regras de produção para a construção de sistemas especialistas, fortemente acopladas aos bancos de dados;
- Definir atributos virtuais através de sistemas de cláusulas simples;
- Folhear os bancos de dados por sobre uma interface gráfica;
- Gerar aplicações simples para manutenção dos bancos de dados.
- ; Ter acesso remoto a outras bases.

Além das funções listadas acima, já implementadas, QUICK-DB prevê também a possibilidade de:

- Gerar relatórios automaticamente a partir das bases.

**3. FUNCIONALIDADE.** A Figura 1 apresenta o diagrama funcional dos módulos de QUICK-DB. Todos os componentes interagem sobre interfaces similares, gráficas.

A utilização do Modelo E-R para definição e utilização de bases de dados está calcada sobre um conjunto de ferramentas de software, descritas resumidamente a seguir.

*Diagramador E-R:* O diagramador é uma ferramenta gráfica para especificação de diagramas E-R estendidos. O sistema trabalha com *mouse* e menus gráficos e direciona o usuário nas suas tarefas. É aqui que se configura a “face” gráfica de um sistema. As demais ferramentas tomam o diagrama aqui definido como base, estabelecendo a unidade visual que o sistema oferece.

*Dicionário de Dados:* O dicionário de dados é automaticamente construído por ocasião da especificação do diagrama do banco de dados. Orientado para o Modelo E-R, tem como característica principal sua plena capacidade de representar elementos do Modelo E-R e extensões, tais como as abstrações de generalização e agregação. O acesso ao dicionário de dados é feito através de um conjunto de métodos e atributos.

*Gerador de esquemas de bancos de dados:* A partir do diagrama E-R, expresso pelo dicionário de dados, são geradas descrições da base de dados para implementação nas instalações correntemente em uso.<sup>12</sup> Da análise das construções definidas pelo usuário no diagrama E-R, este módulo gera a descrição mais adequada para implementação no SGBD objeto.

*Servidor E-R:* Todo acesso à base de dados é feito através do Servidor E-R. Este é constituído por um conjunto de objetos e métodos cuja manipulação permite acesso e operações sobre a base. Os programas de aplicação têm acesso ao banco de dados pela chamada às bibliotecas de funções que operam sobre o banco de dados através do servidor.

*Banco de Dados Autônomo:* O sistema é capaz de atuar como banco de dados autônomo, na própria estação de trabalho. Para tanto, há uma versão do servidor de banco de dados que manipula um sistema de arquivos local.

Tanto para o servidor como para o banco autônomo, os objetos, atributos e métodos são gerados de tal forma que os usuários podem ter acesso ao banco e ao mesmo tempo dispor de seus ambientes de programação preferidos.

*Módulo de Consultas:* A idéia é construir a expressão da consulta num processo estreitamente ligado ao diagrama E-R, e dirigido pelo sistema.<sup>1</sup> Além de possibilitar facilidades para consulta, essa implementação permite que se tenha uma interface gráfica para sistemas de grande porte.

Esta facilidade permite a inspeção direta da base e navegação pelo esquema, sobre o diagrama, podendo ser útil para o melhor entendimento do significado das construções definidas pelo projetista. Numa fase posterior, esta interface será estendida para um sistema onde filtros possam ser definidos, de modo a automatizar o processo para certas

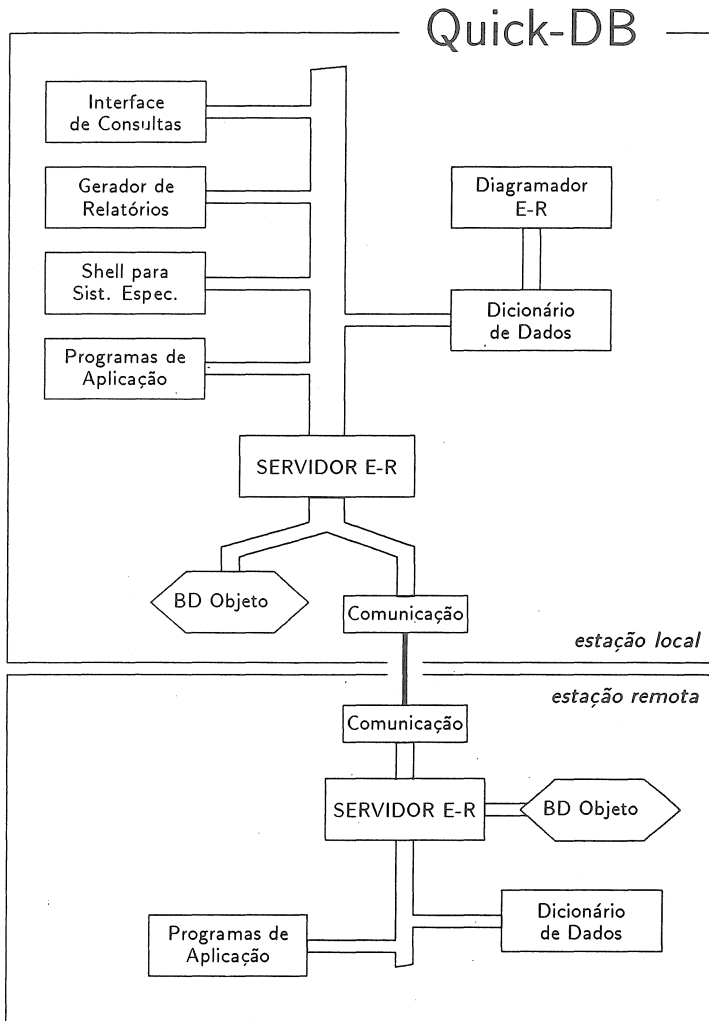


Figura 1. Diagrama funcional

consultas mais simples.

*Gerador de Relatórios:* Constituído por um conjunto de programas total ou semi-automáticos para geração de relatórios de média e baixa complexidade.

*Geradores de Aplicações:* O módulo de geração é capaz de produzir programas para os tipos de transação mais usuais como inclusão, exclusão e alteração de entidades e relacionamentos. Nestas operações, são observadas as propriedades do esquema tais como cardinalidades, dependências de inclusão e exclusão, chaves e atributos obrigatórios. Os programas gerados fazem uso da biblioteca de métodos do Turbo Pascal para a definição de interfaces, obtendo-se programas compactos, robustos e com interfaces bastante elaboradas.

*Shell para Sistemas Especialistas:* Quick-DB dispõe de uma *shell* para sistemas espe-

cialistas baseada no paradigma O-A-V (*object-attribute-value*) para representação do conhecimento.<sup>8</sup> A linguagem para construção de regras abriga elementos como termos e quantificadores, que são acoplados ao servidor de banco de dados de forma nativa. *Servidor Remoto*: Quick-DB dispõe de um servidor de arquivos que pode resolver referências a bases e arquivos armazenados em estações remotas. Os métodos de acesso podem decidir que determinado acesso deve ser encaminhado a outras máquinas, caso haja indicação a respeito no diretório distribuído. Isso é feito de forma totalmente transparente para os programas, e pode ser configurado independentemente do esquema.

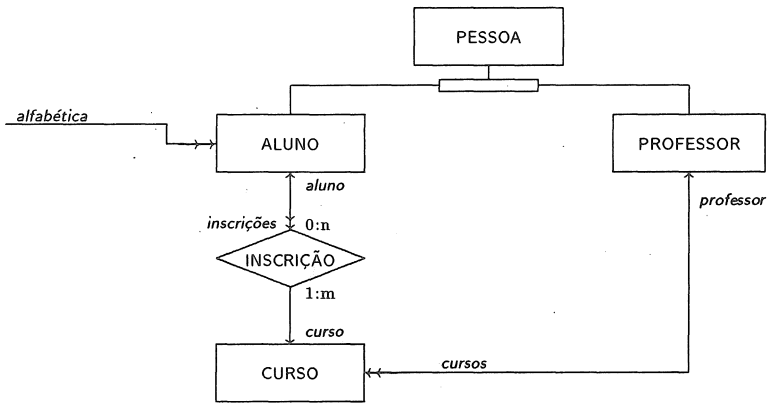


Figura 2. Exemplo simplificado de um diagrama E-R estendido

4. O MODELO ENTIDADE-RELACIONAMENTO E EXTENSÕES. QUICK-SHELL trabalha com bancos de dados que seguem a disciplina de modelagem do Modelo E-R básico estendido. Esta seção apresenta, de modo resumido, as características do modelo resultante. Maiores detalhes são apresentados em outros trabalhos relacionados a este projeto.<sup>11,12</sup>

**ELEMENTOS BÁSICOS.** Os elementos básicos são *Conjuntos de Entidades e Relacionamentos*. A cada relacionamento está associado um inteiro, dito ser o grau do relacionamento, e um conjunto de funções papel, que mapeiam cada instância do relacionamento em seus componentes. *Atributos* são funções que mapeiam entidades e relacionamentos em suas propriedades. Os atributos podem ser mono ou multi-valorados, quando retornam valores atômicos ou listas de valores, respectivamente. *Conjuntos de valores* são utilizados para definir os domínios dos atributos.

*Restrições de Integridade* são basicamente as propriedades do esquema, tais como chaves, entidades fracas, cardinalidades, domínios de atributos e outras expressões que eventualmente devam ser respeitadas quando da instanciação da base de dados.

Este conjunto de elementos para a construção de esquemas E-R é bastante conhecido e aparece fartamente na literatura.

**EXTENSÕES.** No âmbito do nosso projeto, apenas a estrutura básica do Modelo E-R foi utilizada, uma vez que foram introduzidas várias extensões no sentido de tornar o modelo mais rico do ponto de vista semântico e prático. O Modelo E-R utilizado no âmbito deste projeto foi estendido pelos conceitos descritos abaixo.

O conceito de *generalização*<sup>9</sup> foi implementado de forma simples, visando uma solução robusta e eficiente. Todas as implicações ocasionadas pelo conceito são suportadas, tais como herança de chaves e ordens (vide abaixo). Múltipla herança de atributos não é permitida, ou seja, cada subtipo pode estar ligado a apenas um supertipo. No exemplo da Figura 2, ALUNO e PROFESSOR são especializações da entidade PESSOA.

*Atributos referenciais* são atributos que retornam referências a outras entidades do esquema. Este conceito é importante e extremamente útil, uma vez que elimina, em muitos casos, a necessidade de introduzirmos relacionamentos 1:n fictícios, que são melhor expressos com o presente conceito. No exemplo da Figura 2, *professor* é um atributo referencial de CURSO e, quando aplicado a uma instância desta entidade, retorna uma instância de PROFESSOR.

*Ordens* permitem acesso a conjuntos de entidades ou relacionamentos de tal forma que a seqüência em que as instâncias são obtidas fique previamente estabelecida. Uma ordem pode ser parcial ou total. A ordem é total quando todos os elementos do conjunto de entidades, ou do relacionamento que é objeto da ordem podem ser alcançados através da ordem. Ordens podem ser definidas sobre conjuntos de entidades, relacionamentos ou atributos referenciais. Na Figura 2, a ordem *Alfabética*, definida sobre a entidade ALUNO, é uma ordem total. Todas as instâncias daquela entidade podem ser obtidas pela varredura da ordem. Na mesma figura, a ordem *inscrições* é parcial, porque acessa apenas as instâncias de INSCRIÇÃO associadas a uma determinada instância da entidade ALUNO. Da mesma forma, *courses* é parcial e definida sobre a relação inversa do atributo referencial *professor*, que tem PROFESSOR como domínio. Ou seja, para um dado professor *p* a ordem *courses(p)* retorna todo curso *c* tal que *professor(c)=p*.

*Entidades associativas* são relacionamentos que também podem assumir o papel de entidade participante em outros relacionamentos. Entidades associativas podem ter atributos, funções-papéis, ordens e cardinalidades.

*Domínios estruturados* podem ser definidos para os atributos de uma forma genérica. Assim, conjuntos de valores podem caracterizar domínios simples ou compostos. No segundo caso, o domínio é formado por conjuntos de tuplas, cujos componentes, por sua vez, podem ter domínios compostos ou não.

*Atributos virtuais*, que podem ser especificados através de regras de produção ou como expressões.

**5. ACESSO AO BANCO DE DADOS: OBJETOS E MÉTODOS.** O acesso à base de dados é feito através de um conjunto de métodos, aplicáveis às classes de objetos que formam a base.<sup>7</sup> O programa que comporta a definição de cada método é gerado automaticamente a partir do diagrama E-R definido para o banco de dados. Das definições de entidades, relacionamentos, ordens e outras características do esquema, o gerador monta todas as rotinas necessárias para acesso a base de dados. Com isso, todos os nomes presentes no esquema têm métodos e/ou atributos correspondentes no programa gerado, o que torna a programação fácil e confortável para quem acessa a base de dados.

Este conjunto de métodos é denominado *Servidor-ER*.

Há basicamente quatro classes de objetos: *banco de dados*, *conjunto*, *ordem* e *elemento*.

A classe *banco de dados* identifica a base de dados como um todo, e aceita os seguintes métodos básicos:

- ABRE preparação para acesso;
- FECHA encerramento de acesso;
- cria geração da base, com instâncias vazias. item DESTROI apaga uma base de dados.

Um objeto da classe *banco de dados* tem ainda um conjunto de atributos e métodos auxiliares que servem para identificação e controles operacionais.

A classe *conjunto* representa cada um dos conjuntos de entidades  $E$  e relacionamentos  $R$  existentes na base de dados. Os seguintes métodos se aplicam:

- POPULACAO número de elementos na instância do conjunto;
- ABRE preparação para acesso;
- FECHA encerramento de acesso;
- cria geração inicial;
- DESTROI apaga um conjunto da base;
- PONTEIRO obtenção de ponteiro para a instância de um elemento.

Há ainda um conjunto de métodos de acesso restrito que não são relevantes neste contexto. O método *ponteiro* é utilizado para acesso a uma instância sem que seja necessária a definição de uma variável, permitindo um forma de navegação implícita pela base de dados.

A classe *elemento* abriga cada um dos elementos de determinada instância de um conjunto de entidades  $E$  ou de um relacionamento  $R$ , através da própria classe e de suas subclasses. Os métodos do tipo básico são:

- REFERENCIA referência física a um elemento dentro de sua instância;
- SUBSTITUTO referência lógica (surrogate) que identifica unicamente um elemento dentro da base.

Para cada conjunto de entidades  $E$  e relacionamento  $R$ , há um subtipo de elemento especificamente definido, que agrega os atributos de  $E$  ou  $R$ . Um subtipo de elemento, em geral, suporta os seguintes métodos:

- ARMAZENA adição de elemento;
- ATUALIZA regravação de elemento;
- REMOVE remoção de elemento;
- INSTANCIA obtenção da instância de um elemento.

Há ainda alguns métodos que tratam especificamente de elementos sujeitos a generalização e especialização. Nestes casos, observa-se que alguns dos métodos acima podem ou não ser aplicados, dependendo das circunstâncias.

Por fim, a classe *ordem* trata do acesso à base de dados. Uma instância da classe *ordem* pode ser imaginada como uma lista de referências a elementos da base. Assim, pode-se ter várias ordens definidas sobre um conjunto de entidades ou relacionamento. Através de cada uma dessas ordens, pode-se acessar os elementos segundo critérios de ordenação que são definidos por ocasião da confecção do projeto lógico do banco de dados. A classe *ordem* admite os seguintes métodos:

- INICIA posicionamento no início da ordem;
- PROXIMO obtenção do próximo elemento;

- FIM teste de fim de lista;  
PROCURA busca de um elemento que contenha atributo igual ou posterior a determinada chave;  
ENCONTRA busca de um elemento que contenha exatamente determinada chave.

6. UTILIZAÇÃO. O paradigma de utilização de QUICK-DB passa, basicamente, por três fases: *a*) definição do esquema, *b*) geração dos módulos servidores e *c*) manipulação da base de dados, através da linguagem hospedeira e das ferramentas associadas.

A definição do esquema se dá graficamente e, na medida em que o usuário vai dispondo as entidades, relacionamentos e conexões, o sistema direciona a complementação das informações necessárias para que fiquem completas as definições. Todo o processo se dá com auxílio de *pop-up menus* e *mouse*.

As dimensões do espaço de desenho são virtualmente infinitas. A limitação desse espaço é um problema sério em muitos diagramadores, pois bases de dados de maior porte podem abrigar um número de elementos gráficos da ordem de dezenas, ou até mesmo centenas. O acesso a uma “lupa” de magnificação regulável permite que o usuário se oriente e se movimente no espaço de desenho com relativa facilidade.

O diagramador oferece, também, facilidades para documentação dos esquemas. Para impressão dos diagramas completos, QUICK-DB trabalha com um sistema de paginação que elimina as desvantagens das “colagens” de papéis. Também são disponíveis opções para obtenção de textos prontos para os formatadores  $\text{T}_{\text{E}}\text{X}$  e  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ .

Além dessas facilidades, o diagramador é capaz de verificar a correção e consistência dos diagramas.

A geração dos servidores é a fase onde QUICK-DB, a partir das informações do dicionário de dados, gera programas que atuarão como servidores das bases de dados.<sup>2</sup> O servidor E-R é composto pelas definições de uma coleção completa de objetos e métodos que permitem a utilização da base a partir de uma linguagem hospedeira. No momento, utilizamos o Turbo Pascal (a partir da versão 5.5) neste papel, mas a estrutura do sistema permite a migração para outra linguagem (como C++, por exemplo).

Quando o programador preparar programas de acesso à base, basta que invoque a biblioteca de definições e, de modo simples e estruturado, poderá construir os procedimentos necessários. Todos os componentes do diagrama estarão disponíveis, bem como um conjunto de definições auxiliares que facilitam a manipulação dos dados.

Outra forma de acesso se faz através do folheador da base, que permite a verificação das instâncias e a navegação pelas conexões do diagrama, respeitando ordens e relacionamentos.

Também têm acesso à base de dados, através do servidor, os programas gerados pelo gerador de aplicações e pela *shell* para sistemas especialistas. Estes últimos, entretanto, definem bases próprias (usualmente denominadas bases de conhecimento) que constituem uma classe de atributos virtuais a serem incorporados ao esquema.

7. CONCLUSÕES. O sistema em questão está implementado em Pascal, com aproximadamente 35.000 linhas de código. Uma das premissas que adotamos por ocasião da estruturação deste projeto foi a de que as soluções pudessem ser integralmente implementadas, com resultados práticos imediatos.

As técnicas de implementação adotadas permitem que o sistema seja facilmente transportado para outros ambientes de desenvolvimento, notadamente C++.

Extensões. Entre as extensões já estudadas e planejadas, encontram-se:

- novos tipos de dados manipulados sob o paradigma de orientação a objetos, principalmente tipo texto e figuras geométricas;
- acoplamento com servidores SQL.

Trabalhos Correlatos. A abrangência deste trabalho torna-o, de certa forma, não característico. Há na literatura menção de diagramadores E-R, interfaces com sistemas especialistas,<sup>4,5,6</sup> folheadores, geradores de aplicações, dicionários de dados, de forma geral. O mérito de QUICK-DB é reunir sob um mesmo paradigma de utilização todas essas facilidades, oferecendo aos seus usuários um ambiente completo, uniforme e consistente para o desenvolvimento de aplicações de bancos de dados.

## 8. REFERÊNCIAS.

1. CZEJDO, B., EIMASRI, R., et al.; *A Graphical Data Manipulation Language for an Extended Entity/Relationship Model*, IEEE Computer v.23, p.26-36, March 1990.
2. DELVAUX, M. C.: Geração de Rotinas Orientadas a Objetos para Manipulação de um Banco de Dados, Projeto Final de Curso, Instituto de Matemática, Rio de Janeiro, 1991.
3. GARDARIN G., VALDURIEZ P.; *Relational Databases and Knowledge Bases*, Addison-Wesley Publishing Company, 1989.
4. JARKE, MATTHIAS, VASSILIOU, YANNIS; Coupling Expert Systems with database Management Systems. In: Reitman, W. (ed). *Artificial Intelligence Applications for Business*, Norwood, NJ:Ablex, 1985, cap. 5 p. 65-85.
5. 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.
6. KERSCHBERG, L.; KORTX: An Expert Database System Shell for Knowledge Based Entity relationship model, *Int. Conf. on the Entity/Relationship Approach*, pp 174-187, 1989.
7. MEYER B., *Object-Oriented Software Construction*, New York, Prentice-Hall, 1988.
8. PACITTI, E.C. and SILVEIRA, P.; Um Sistema de regras de Produção Fortemente Acoplado a Bancos de Dados, Dissertação de Mestrado, in *Anais do VII Simpósio Brasileiro de Bancos de Dados*, Porto Alegre, 1992.
9. PECKHAM, M. and MARYANSKI, F.; Semantic Data Models, *ACM Computing Surveys* 20, 3(September 1988).
10. SILVEIRA, P. M.; A Formalization of The E-R Model, *Proceedings of the IX Conferencia Internacional de la Sociedad Chilena de la Ciencia de la Computation*, Santiago, Chile, 1989.
11. SILVEIRA, P. M.; Procedural Data Manipulation Operations for the E-R Model, *Proceedings of the XVI Conferência Latino Americana de Informática*, Assunção, Paraguay, 1990.
12. TOSO, V.E.F., SILVA, M.F.; Gerador de Banco de Dados do Quickdb. Projeto Final de Curso de Informática - IM/UFRJ Julho de 1990.