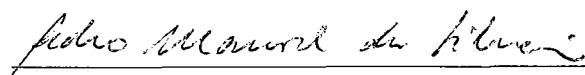


ASPECTOS DA IMPLEMENTAÇÃO DE INTERFACES GRÁFICAS PARA CONSULTAS A BANCOS DE DADOS

Helena Trinkenreich

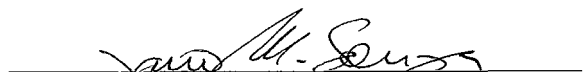
TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO EM ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:



Prof. Pedro Manoel da Silveira, Ph.D.

(Presidente)



Prof. Jano Moreira de Souza, Ph.D.



Prof. Fernando Silva Pereira Manso, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

MAIO DE 1991

TRINICENREICH, HELENA

Aspectos da Implementação de Interfaces Gráficas para Consultas a Bancos de Dados [Rio de Janeiro] 1991

viii, 122 p., 29,7cm. (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 1991)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Interfaces Gráficas 2. Modelo Entidade-Relacionamento 3. Bancos de Dados 4. Orientação a Objetos

I. COPPE/UFRJ II. Título (série).

A minha família e
ao Roberto

AGRADECIMENTOS

- Aos meus pais que, mesmo não estando hoje presentes, sempre contribuíram para o meu crescimento pessoal e profissional;
- Ao Pedro Manoel da Silveira por ter-me orientado neste trabalho;
- Ao Jano Moreira de Souza por sua compreensão, principalmente, nesta fase final do trabalho;
- Ao Fernando Silva Pereira Manso por ter participado em minha banca de tese;
- A Selma Regina Sá da Silva por ter-me auxiliado com as normas das Referências Bibliográficas; e
- Aos colegas do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro (NCE / UFRJ) pelo apoio e liberação da carga horária de trabalho durante o tempo de desenvolvimento desta tese.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para obtenção do grau de Mestre em Ciências (M.Sc.).

ASPECTOS DA IMPLEMENTAÇÃO DE INTERFACES GRÁFICAS PARA CONSULTAS A BANCOS DE DADOS

Helena Trinkenreich

Maio de 1991

Orientador : Pécro Manoel da Silveira

Co-Orientador : Jano Moreira de Souza

Programa : Engenharia de Sistemas e Computação

O objetivo deste trabalho é a implementação do Módulo de Consultas a bancos de dados que compõe a ferramenta de desenvolvimento de *software* chamada QUICK-DB. Esta ferramenta utiliza o Modelo Entidade-Relacionamento como base, permitindo, por meio de interface gráfica, a definição de diagramas E-R.

O Módulo de Consultas foi implementado a partir da interface gráfica desenvolvida no QUICK-DB. Através do esquema E-R de uma base de dados desenhado na tela, o usuário pode, com o auxílio do *mouse* e através de menus, efetuar consultas ao BD, tornando mais amigável, fácil e intuitiva sua manipulação pelo usuário.

Foram adotados dois conceitos no desenvolvimento deste módulo. O primeiro deles é o *folheamento* que permite ao usuário obter informações do BD a partir do instanciamento das estruturas do banco. Seus dados são mostrados na tela através de janelas. O segundo conceito é o de *navegação* que pode ser aplicado de duas formas. A primeira diz respeito ao acesso a uma dada entidade ou relacionamento. O usuário pode navegar por suas instâncias, através de uma ordem pré-definida, uma a uma. A outra forma de navegação é aquela que permite ao usuário estabelecer um caminho a ser percorrido no esquema E-R que faz com que ele possa consultar instâncias de entidades associadas a outras entidades por meio de algum relacionamento.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Master of Science (M.Sc.).

ASPECTS OF THE IMPLEMENTATION OF GRAPHICAL INTERFACES TO DATABASES QUERIES

Helena Trinkenreich

May, 1991

Thesis Supervisor : Pedro Manoel da Silveira

Co-Supervisor : Jano Moreira de Souza

Department : Systems Engineering and Computing

This work consists of the implementation of a database query system which is part of the software tool QUICK-DB. This tool allows for definition and manipulation of Entity-Relationship diagrams, through a graphical interface.

The software module for queries was implemented using the graphical interface of QUICK-DB. Starting from a diagram in the monitor screen, the user can use the mouse in a simple and intuitive manner.

Two main concepts are used in the software: browsing, that allows the instantiation of the boxes and their inspection, and navigation, where the user can navigate through the instances following the access orders defined for the database diagram. Another form of navigation allows the user to reach other instances following given relationships.

ÍNDICE

Capítulo I - Introdução	1
Capítulo II - Revisão Bibliográfica	3
11.1. Introdução	3
11.2. O que existe nesta área ?	3
11.3. Análise das Aplicações	11
11.4. Conclusão	15
Capítulo III - Linguagens de Consulta	16
111.1. Introdução	16
III.2. Avaliação de Linguagens e Modelos de Consultas	16
111.3. Descrição de Consultas	22
111.4. Conclusão	28
Capítulo IV - Modelos de dados	31
IV.1. Introdução	31
IV.2. Considerações sobre Modelos	31
IV.3. O Modelo Entidade-Relacionamento	33
IV.4. Extensões ao Modelo Entidade-Relacionamento	36
IV.5. Especificação Formal de uma Linguagem	38
IV.6. Conclusão	43
Capítulo V - O Projeto UniversiData	44
V.1. Introdução	44
V.2. O Projeto e seus Objetivos	44
V.3. Os Componentes do QUICK-DB	46
V.4. Conclusão	50
Capítulo VI - O Servidor de Banco de Dados	51
VI.1. Introdução	51
VI.2. O que é Servidor ?	53

VI.3. Como funciona o Servidor ?	54
VI.4. Os Métodos	57
VI.5. Conclusão	69
Capítulo VII - Interface Gráfica de Consultas	71
VII.1. Introdução	71
VII.2. Objetivos	71
VII.3. Abrangência	72
VII.4. Definição das Características	73
VII.5. O Folheamento	76
VII.6. Conclusão	111
Capítulo VIII - Conclusões	114
Referências Bibliográficas	118

CAPÍTULO I

INTRODUÇÃO

Atualmente, as formas de consultas ad hoc a bancos de dados não são inteiramente satisfatórias. O uso de linguagens de consulta de formulação genérica é pouco difundido, o que resulta na pequena exploração das facilidades de consultas aos bancos de dados. Até mesmo aqueles usuários que possuem experiência com computadores apresentam dificuldades em utilizar linguagens ditas de alto nível. Os usuários pouco experientes, em geral, não costumam ter disposição, paciência ou habilidade para aprender e usar estas linguagens corretamente. Este problema aumenta quando o volume de informações é muito diversificado levando a esquemas de bancos de dados complexos.

Na realidade, é possível identificar algumas razões para que isso esteja ocorrendo. Existem fatores que contribuem para a dificuldade na utilização e compreensão das linguagens de consulta. Alguns desses fatores são:

- Linguagens existentes como QUEL, SQL, QBE e outras não são tão naturais;
- Há dificuldade do usuário em conhecer, com maiores detalhes, a organização do banco de dados quando se constrói expressões de consulta que envolvam recuperação de informações através de um processo de seleção mais complexo;
- O usuário precisa lembrar-se de coisas como nomes, formatos e unidades dos atributos. Este problema torna-se maior quando o banco de dados possui um esquema complexo com muitos tipos de registros e atributos; e
- Os modelos de dados são semanticamente pobres e as linguagens de consulta de alto nível são baseadas em conceitos matemáticos com os quais os usuários não estão, necessariamente, familiarizados. A solução para isto estaria no uso de um modelo que torne mais clara a interface do usuário com os dados complexos.

Por esta razão desenvolvemos estudos sobre a aplicação de interfaces gráficas às consultas de caráter genérico a bancos de dados de grande porte, por entendermos tratar-se de uma alternativa promissora no que se refere àquelas consultas. Utilizando uma interface gráfica, o usuário não se preocuparia em guardar tantas informações, teria acesso mais simplificado aos dados, poderia construir sua consulta gradativamente.

Devido à sua característica semântica e gráfica, o modelo de dados no qual está baseado este trabalho, é o Modelo Entidade-Relacionamento que é usualmente utilizado na fase de definição conceitual de uma base de dados. Nosso objetivo é ele sirva, também, como uma janela para enxergarmos a representação interna da base de dados. É apresentada uma proposta de como efetuar consultas a um banco de dados utilizando um diagrama E-R como suporte para a interface gráfica e tendo como base uma linguagem formal definida sobre este diagrama.

A fim de esclarecermos melhor os pontos considerados no trabalho, apresentaremos, inicialmente, no capítulo II, alguns trabalhos anteriormente desenvolvidos que utilizam interfaces gráficas tendo o Modelo E-R como base na formulação de consultas pelo usuário. No capítulo III, serão tecidas considerações de ordem geral sobre linguagens de consulta para o usuário casual de banco de dados, destacando e analisando alguns aspectos que as tornam pouco amigáveis. Em seguida, no capítulo IV serão abordadas características específicas do Modelo Entidade-Relacionamento, com propostas de extensão para o mesmo, visto que o sistema no qual está inserido este trabalho é inteiramente baseado naquele modelo. Depois dos esclarecimentos preliminares, no capítulo V descreveremos o ambiente no qual encontra-se inserido o presente trabalho, isto é, o Projeto UniversiData. O capítulo VI abordará com maiores detalhes o Servidor de Bancos de Dados que contém bibliotecas de funções com métodos associados aos objetos do BD seguindo a orientação a objetos do Turbo Pascal 5.5. O capítulo VII tratará do módulo de consultas desenvolvido neste trabalho que é o nosso assunto propriamente dito. Finalmente, no capítulo VIII teremos a conclusão deste trabalho.

CAPÍTULO II

REVISÃO BIBLIOGRÁFICA

II.1 - Introdução

Vários trabalhos têm sido desenvolvidos na área de Interfaces Gráficas para a manipulação de bases de dados. Muitos deles se apoiam no Modelo Entidade-Relacionamento [2] para definir a interface com o usuário. Vamos, então, descrever, neste capítulo, alguns desses trabalhos, tecendo algumas observações a seu respeito.

11.2 - O que existe nesta área ?

Dentre os artigos a que tivemos acesso, consideramos alguns mais interessantes que outros no que diz respeito a essa área de aplicação. Estes artigos foram selecionados para serem descritos devido ao enfoque objetivo que apresentam no que tange ao desenvolvimento de *softwares*. Estes trabalhos, de um modo geral, consistem de *softwares* aplicados à manipulação de bancos de dados, alguns mais completos que outros. Neles são implementadas interfaces gráficas para facilitar sua utilização por parte dos usuários não experientes e, de alguma forma, baseiam-se no Modelo E-R.

Artigo 1:

Zhang e Mendelzon [36] desenvolveram o *gql/ER*, do que trata o artigo *A Graphical Query Language for Entity-Relationship Databases*. Segundo eles a busca de uma interface amigável ao usuário é de grande importância. Consideraram que os aspectos seguintes são essenciais para que se tenha êxito:

- . auxílio à navegação;
- . gráfico;

- . folheamento;
- . ser conciso; e
- . inodelo de dados.

Seu funcionamento se dá a partir de um esquema E-R de uma base de dados. Ao apontarmos o *mouse* para *um* nó (entidades e relacionamentos), estaremos selecionando elementos para fazerem parte da consulta. As condições para a consulta deverão ser fornecidas através de um tipo de menu contendo os atributos daquele nó, onde deveremos selecionar o atributo desejado e preencher o valor a que ele deverá atender. Isto poderá ser feito para cada nó. O resultado da consulta poderá ser visto a todo instante caso o usuário assim o deseje. A interpretação da consulta poderá ser feita de duas formas. A primeira é automática, ou seja, baseada em um algoritmo proposto pelos autores, o próprio sistema estabelece o caminho que une os nós selecionados. O resultado, portanto, poderá não ser o desejado. Desta forma, o usuário deverá mudar sua opção e escolher a segunda forma de interpretação: a manual. Nesta outra forma será o próprio usuário quem deverá especificar as conexões entre os nós.

O modelo de banco de dados utilizado para este *software* é o Relacional [29][1]. O Modelo E-R é usado, somente, para servir de interface gráfica permitindo que o usuário formule suas consultas. Ao serem especificadas, as consultas E-R são mapeadas em consultas relacionais. No menu que é mostrado na tela, onde o usuário poderá escolher alguma opção, são apresentadas operações algébricas como união, interseção e diferença. O protótipo desenvolvido a partir deste trabalho foi implementado usando somente a forma manual de interpretação.

Artigo 2:

Wong e Kuo [35] desenvolveram o *software* GUIDE, descrito no artigo GUIDE:

Graphical User Interface for Database Exploration. Segundo o artigo, GUIDE é um sistema que usa dispositivos gráficos como ferramentas para servirem de interface a bancos de dados complexos. São analisadas as dificuldades no uso de linguagens de consultas para bancos de dados grandes e complexos. O esquema do banco é mostrado como uma rede de tipos de entidade e relacionamento. As consultas podem ser expressas como caminhos nessa rede. Utilizam uma versão do modelo E-R para representar explicitamente os relacionamentos entre as entidades. O esquema do modelo E-R pode ser mostrado como uma rede de objetos, cada um representando uma entidade ou relacionamento. Consultas podem ser expressas como travessias ao longo da rede de entidades, podendo-se usar cores para indicar seus caminhos. Partes do esquema podem ser feitas visíveis ou invisíveis, conforme escolha do usuário. As consultas poderão ser construídas gradativamente pelo usuário, obtendo resultados intermediários.

Os autores dividem o sistema em quatro estágios de formulação de consultas:

- a) Estágio de Definição de Dados: caberá ao DBA prover informações sobre o esquema que dizem respeito às entidades, relacionamentos e seus atributos. O layout gráfico do esquema é alimentado no sistema. Ao DBA deverão ser dadas algumas facilidades para especificar um layout gráfico do esquema, construir um diretório hierárquico de assuntos para os objetos do esquema e um diretório hierárquico de atributos para cada entidade e relacionamento e, finalmente, especificar os níveis de importância das entidades e relacionamentos no esquema;
- b) Estágio de Exploração do Esquema: neste estágio o usuário poderá usar o diretório de assuntos para determinar a parte mais importante do esquema para que, a partir daí, o esquema possa ser analisado em vários níveis de abstração, onde somente os objetos acima de um dado nível tornam-se visíveis na tela. O usuário poderá, também, escolher objetos irrelevantes que deverão ser removidos da tela, ocultando, graficamente, o esquema. Com a parte relevante selecionada e mostrada no nível desejado de detalhes, o usuário estará pronto para formular

suas consultas;

- c) Estágio de Formulação de Consultas: o usuário poderá construir a consulta gradativamente. Poderão ser mostrados resultados de consultas parciais. As consultas deverão ser definidas pelo usuário atravessando a rede através de um caminho que é diferenciado por cores. Pode incluir e remover condições e formular várias consultas locais em diversas partes do esquema. Se desejar, poderá unir essas consultas locais e obter um resultado único;
- d) Estágio de Mostrar Resultados: o resultado pode ser mostrado de diversas formas, entre elas grafos e diagramas de barras.

A interação com GUIDE é feita através da seleção de comandos de um menu localizado à direita da tela, usando joystick.

Artigo 3:

Rogers e Cattell [18] relataram em seu artigo *Entity-Relationship Database User Interfaces* uma experiência com interfaces de usuários para bancos de dados orientados ao modelo E-R. Nele são descritas duas ferramentas. A primeira chama-se *SchemaDesign* e a segunda, *DataBrowse*. *SchemaDesign* é definido como sendo uma ferramenta baseada em janelas com a qual os usuários poderão criar e mostrar, graficamente, o esquema da base de dados usando um diagrama E-R. É mais comumente usada por DBA's para definirem bancos de dados. No entanto, é também útil para vermos o esquema de um BD já existente. Contém uma janela de mensagens para mensagens e uma subjanela de comandos para nela digitarmos informações ou clicarmos comandos com o uso de *mouse*. A subjanela de edição é usada para *mouse* e operações de entrada por teclado. O diagrama E-R inicialmente mostrado na tela possui entidades e relacionamentos. O relacionamento pode ser transformado em entidade ao definirmos uma chave para ele. Há três passos principais para o uso do *schemadesign*:

- . Identificar 'entidades', onde o usuário deverá identificar objetos do mundo real que ele queira que sejam modelados no BD;
- . Criar tabelas de entidades para eles, onde uma tabela é criada para cada objeto através de menus e *mouse* contendo nomes dos campos e seus respectivos tipos; e
- . Identificar Características, onde o usuário deverá analisar as características dos campos das entidades, determinando se eles deverão ser considerados atributos ou formar relacionamentos associando aquelas entidades a novas tabelas geradas a partir desses campos.

Uma entidade não é considerada completamente definida até que uma chave seja definida para ela.

Esta ferramenta permite ainda que o usuário faça modificações no esquema da tela, o que provocará a reconfiguração automática do banco de dados existente conforme necessário para adaptar-se ao novo diagrama.

Por sua vez, *DataBrowse* é definido como sendo um programa baseado em janelas que permite a visualização e edição de entidades lógicas ao invés de simples registros encontrados em bancos de dados relacionais convencionais. Pode ser usado para folhear e editar informações no esquema, sem mudá-lo. Possui um número de subjanelas que, de cima para baixo na tela, são:

- . subjanela para mensagens;
- . subjanela de comandos para selecionar entidades e relacionamentos, para mudar de um BD para outro, e outras operações;
- . subjanela de edição e folheamento para ver e editar dados;
- . subjanela de textos para mostrar e editar campos que sejam textos, onde esta

- subjanela só será criada se o BD possuir este tipo de campo; e
- . subjanela para mostrar desenho com uma subjanela de edição de desenho que, da mesma forma, só será criada se o BD possuir campos que sejam desenhos.

O *DataBrowse* possui ferramentas de edição de textos, também via *scanners*. Possui desenhos que podem ser armazenados e mostrados. Suas características fundamentais são orientação central a entidade e capacidade de folheamento para andar entre as entidades.

Artigo 4:

Schneider e Trepiecl [19] são responsáveis pelo artigo intitulado *A Graphical Query Language Based on an Extended E-R Model*. Este artigo apresenta uma adaptação da linguagem gráfica de consultas CANDID para um modelo ER estendido com relacionamentos ISA. Aquela linguagem é ativada através de uma interface que também permite definição de esquema e atualização de dados. CANDID baseia-se nos seguintes princípios:

- . seleção de elementos sobre um grafo de requisitos;
- . efeito gráfico imediato de qualquer comando;
- . possibilidade de uma composição de requisitos incremental; e
- . possibilidade de incorporar o resultado em um esquema externo.

O *software* divide-se em janelas de esquema, de passos, de recapitulação, de resultados e de comandos.

Artigo 5:

Czejcło, Elmasri e Rusinkiewicz [5] desenvolveram uma ferramenta de interface gráfica que utiliza o Modelo E-R como base para manipulação da base de dados. O artigo que contém descrição desta ferramenta é *A Graphical Data Manipulation Language for an Extended Entity-Relationship Model*. Para tanto criaram uma extensão ao modelo definido por Chen que chamaram ECER (*Extended Conceptual Entity-Relationship Model*). Os operadores de manipulação do esquema correspondem a operadores algébricos formais definidos para este modelo que, no entanto, não são visíveis ao usuário final. Para este usuário, sua única visão será aquela dada pelo diagrama E-R do esquema. Os operadores gráficos especificados pelo usuário são transformados em sequências de operadores relacionais para processamento. A sequência de operadores relacionais só é realmente executada quando a consulta estiver completamente especificada. Isto permite que uma otimização seja realizada antes que a consulta seja executada.. Possui módulos de recuperação de dados a partir do BD e, também, de atualização desses dados. O que foi feito neste trabalho foi, previamente, propor uma linguagem gráfica de consultas para o modelo E-R de Chen. A partir dele, extensões para incluir cálculos foram desenvolvidas.

Uma entidade individual é identificada por um valor chave de *surrogate* único que serve para distingui-la de outras entidades no BD. Uma instância de um relacionamento deve ser unicamente identificada pelas instâncias das entidades participantes.

A ferramenta não foi completamente implementada. Um possível layout da tela para a interface gráfica é uma lista de operadores no alto da tela, uma área reservada na parte de baixo para mensagens entre o sistema e o usuário, e o restante para mostrar o diagrama atual. Diagramas grandes poderão ser mostrados usando janelas ou zoom. O diagrama atual mostrado na tela corresponde a uma visão dos dados e pode ser interpretado como uma consulta..

As operações gráficas que podem ser usadas durante o processo de formulação de consultas são, entre outras:

. *Restringir* : esta operação impõe uma condição booleana sobre os atributos de

entidades e relacionamentos a fim de restringir as instâncias nesses conjuntos aquelas que satisfazem à condição. Seleciona-se o operador e, depois, o atributo, digitando a condição;

- . *Remover* : é usada para remover entidades, relacionamentos e atributos do diagrama. Tem efeito de propagação removendo relacionamentos associados a entidades removidas e subclasses de generalização/especialização removidas;
- . *Mostrar* : este operador traduz o diagrama final em uma sequência de operações que podem ser executadas sobre o BD. Essa execução vai nos gerar o resultado da consulta que poderá ser apresentado ao usuário. Faz com que junções sejam executadas entre as relações restantes no diagrama e posterior projeção dos atributos remanescentes. Isto abrangerá somente as entidades restantes após a operação de restrição. O resultado é mostrado sob a forma de uma tabela;

Manter : é o inverso de *Remover*, pois inarca aqueles que não desejamos remover;

- . *Duplicar* : necessário às consultas que envolvem referências múltiplas e distintas à mesma entidade.

Há ainda operadores que efetuam cálculos sem, no entanto, atualizar o BD.

Artigo 6:

Fogg [10], no artigo *Lessons from a "Living In a Database" Graphical Query Interface*, descreve o *LID* ("Living In a Database"). Nesta proposta, o banco de dados é visto sob a perspectiva de uma tupla. Quando quisermos fazer uma consulta, deveremos selecionar a entidade desejada. Nesse momento, uma tupla, a tupla atual daquela entidade, é mostrada com os nomes de seus atributos e respectivos valores.

O *software* utiliza-se do modelo E-R para representar na tela os caminhos que deverão ser seguidos para obtenção do resultado da consulta. Não trata-se de uma

modelagem E-R propriamente dita. Os losangos e retângulos usados que assemelham-se à representação E-R, aqui servem somente para indicar que a tupla atual de uma entidade está conectada a pelo menos uma tupla de outra entidade, através de determinado relacionamento.

Ao ser mostrada a tupla atual da entidade escolhida, automaticamente aparecem na tela as entidades e relacionamentos que possuem, pelo menos, uma tupla associada àquela primeira tupla. Isto significa que entidades e relacionamentos serão mostrados na tela se e somente se estiverem relacionados diretamente com a entidade cuja tupla estiver sendo mostrada e se houver, pelo menos, uma tupla nestas entidades que esteja relacionada àquela.

Ao ser escolhida uma entidade, uma lista de tuplas é colocada na lista atual de tuplas, onde a primeira tupla da lista é aquela que será mostrada. Se a tupla que desejarmos não for essa, deveremos solicitar que uma busca seja feita ao longo desta lista tendo como chave para a busca uma cadeia de caracteres.

O folheamento é permitido, fazendo com que os elementos mostrados na tela mudem.

11.3 - Análise das Aplicações

Com relação aos trabalhos descritos, podemos fazer aqui tecer algumas observações a seu respeito.

Artigo 1:

A partir do trabalho de Zhang e Mendelzon [36] intitulado *A Graphical Query Language for Entity-Relationship Databases*, podemos relacionar os seguintes pontos:

- O *software* não representa graficamente o auto-relacionamento. Ele pode existir

embora de forma não perceptível. Isto pode tornar complicado ao usuário não experiente a visualização da representação do esquema da base de dados;

- O modelo de dados utilizado a nível conceitual é o Modelo Entidade-Relacionamento, enquanto que a base de dados foi implementada através do Modelo Relacional [29][1];
- O usuário deverá optar por operações algébricas que deverão ser aplicadas às condições. Entendemos que isto deveria ser transparente ao usuário; e
- Embora eles afirmem considerar importante a existência do *folheamento* no *software*, não foi devidamente definido neste artigo. Preocuparam-se mais com a navegação pelo esquema., não mencionando o folheamento dentro de uma mesma entidade, por exemplo.

Artigo 2:

Do trabalho de nome *GUIDE: Graphical User Interface for Database Exploration* [35], desenvolvido por Wong e Kuo, podemos tecer as seguintes considerações:

- A explicação sobre as operações gráficas não é clara;
- Os menus e a manipulação gráfica são bastante complicados, sendo pouco amigáveis a um usuário não experiente;
- Não parece prever generalização ou especialização;
- Não permite o *folheamento*, e sim, respostas em forma de lista, não estando muito clara a forma com que isto é feito;
- Não ficou claro como o usuário tem conhecimento dos atributos das entidades no momento da formulação das condições. Obrigar o usuário a mantê-los em sua memória seria bastante ruim; e

- Não diz como é feita a alimentação do esquema no sistema. Não parece ser através de um diagrama desenhado na tela.

Artigo 3:

Por sua vez, o artigo denominado *Entity-Relationship Database User Interfaces* [18] escrito por Rogers e Cattell tem como características:

- *O software* baseia-se no modelo E-R mas utiliza-se de sistema não puramente relacional no projeto do banco de dados;
- A interface com o usuário, embora interessante, apresenta poucas características gráficas. O usuário nem sempre parte de um diagrama E-R para fazer suas consultas;
- Não é permitida ou facilitada a navegação ao longo das instâncias das entidades ou dos caminhos. A navegação se dá através dos campos (referências); e
- Não é assim tão amigável ao usuário casual. Hoje, as características gráficas são mais levadas em consideração.

Artigo 4:

Fazendo uma rápida análise sobre o trabalho de Schneider e Trepied [19], de nome *A Graphical Query Language Based on an Extended E-R Model*, verificamos que:

- Este sistema propõe uma interface complexa ao entendimento do usuário casual. Os passos a serem seguidos para a formulação das consultas são bem diferentes da forma com que raciocinamos normalmente, o que dificulta sua manipulação por parte do usuário;

- A divisão em janelas apresentada não é muito natural à compreensão do usuário, principalmente o não experiente; e
- As consultas são estáticas, não permitindo navegação pelo banco de dados.

Artigo 5:

A partir da ferramenta desenvolvida por Czejdo, Elmasri e Rusinskiewicz [5] descrita no artigo intitulado *A Graphical Data Manipulation Language for an Extended Entity-Relationship Model*, observamos que:

- Se o esquema do banco de dados for muito grande e complexo, será inviável manter os atributos das entidades e relacionamentos mostrados na tela, na interface gráfica, além de ser bastante trabalhoso remover ou manter uma grande quantidade de entidades, relacionamentos e atributos para formular uma consulta. Se o zoom reduzir muito as proporções do diagrama, dificilmente continuaremos enxergando os atributos;
- Permite consulta estática, e não dinâmica, ou seja, não promove o uso de *fobheamento*, olhando várias instâncias do mesmo elemento. Ao invés disso, fornece o resultado em forma de tabela; e
- Não oferece resultados parciais das consultas.

Artigo 6:

Finalmente, quanto ao artigo escrito por Fogg [10] de nome *Lessons from a "Living In a Database" Graphical Query Interface*, podemos notar que:

- A busca de uma tupla não é feita através de chaves ou ordens, ou, pelo menos, o autor não menciona tal fato;

- O modo de efetuar consulta é bastante interessante, embora o usuário não tenha contato com o esquema como um todo, o que prejudica um pouco a navegação;
- Pelo que diz o autor, parece-nos que o esquema do Modelo E-R não é usado na criação do banco de dados. Este esquema é usado, somente, na fase de consultas;
- Não estabelece condições de seleção das tuplas; e
- Uma consulta que envolva várias entidades não permitirá a visualização do resultado como um todo, já que somente os atributos de uma entidade são vistos de cada vez.

11.4 - Conclusão

Como pudemos observar, há *softwares* bastante interessantes na área de interfaces gráficas, utilizando o conceito ou o próprio esquema da modelagem E-R. No entanto, não é possível identificarmos uma tendência predominante nestes trabalhos. De um modo geral, seus objetivos são semelhantes, visando fornecer interfaces amigáveis aos usuários e facilidade de formulação de consultas com *navegação e folheamento*. O que podemos verificar, entretanto, é que esses dois conceitos são, muitas vezes, tratados de forma pouco padronizada.

CAPÍTULO 111

LINGUAGENS DE CONSULTA

III.1 - Introdução

O objetivo deste capítulo é descrever e avaliar algumas das linguagens e modelos tradicionais de consultas visando destacar as atuais necessidades dos usuários em geral e, particularmente, daqueles casuais ou não-experientes. O intuito deste trabalho é buscar formas mais amigáveis de consulta a bancos de dados.

111.2 - Avaliação de Linguagens e Modelos de Consultas

Codd [4] estabeleceu as bases de duas linguagens para exprimir-se as operações do Modelo Relacional [1][29] usando técnicas matemáticas para definição ou manipulação de conjuntos [4]. São elas:

- *Linguagens Algébricas* (Álgebra Relacional)
- *Linguagens de Cálculo de Predicados* (Cálculo Relacional)

A Álgebra Relacional se utiliza de um conjunto de operações de alto nível, onde as consultas são expressas através da aplicação dessas operações a relações. Há, basicamente, oito operações utilizadas na Algelsra. No entanto, apenas cinco delas são consideradas primitivas. São elas Seleção, Projeção, União, Produto *Cartesiano* e Diferença. As outras três operações podem ser obtidas a partir da combinação dessas, sendo elas *Junção*, *Interseção* e *Divisão*.

Por sua vez, o Cálculo Relacional, que se utiliza de predicados de lógica e não daquelas operações, representa uma alternativa à Algelsra Relacional na formulação de consultas, onde elas descrevem um conjunto de tuplas através da especificação do predicado a que as tuplas devem satisfazer. Esta categoria subclivicle-se, ainda, em

outras duas que são *Cálculo de Predicados Orientado a Tuplas* e *Cálculo de Predicados Orientado a Domínio*, dependendo dos objetos tratados serem tuplas ou elementos do domínio de algum atributo da relação.

Comparativamente, enquanto a Álgebra Relacional é procedimental, ou seja, estabelece um procedimento de "como" resolver um problema recuperando os dados e exigindo um certo conhecimento de programação, o Cálculo Relacional está mais próximo de uma linguagem semelhante à forma, como pensamos, sendo não procedimental, descritivo e que apenas estabelece "qual" é o problema, ou seja, qual é o subconjunto de dados a recuperar [6].

Na realidade, Álgebra Relacional e Cálculo Relacional são, fundamentalmente, equivalentes. Segundo Codd [4], Álgebra é tão poderosa quanto Cálculo e vice-versa. Toda consulta representada através de uma delas pode ser igualmente representada através da outra [11].

Contudo, não iremos nos ater a detalhes destas linguagens tradicionais de consultas, o que fugiria ao escopo deste trabalho. Através da apresentação de exemplos de consultas a um banco de dados fictício, discutiremos características de algumas das linguagens desenvolvidas baseadas nestas duas categorias aqui descritas.

A fim de que possamos analisar algumas destas linguagens, utilizaremos, como exemplo, o modelo que está descrito na Figura 1.

Este diagrama mostrado na Figura 1 foi baseado no Modelo Entidade-Relacionamento [2], tendo sido utilizada uma versão simplificada. A partir do diagrama, descrevemos as estruturas relacionais a ele correspondentes. ALUNO, DISCIPLINA e PROFESSOR são mostrados em forma de relações, bem como CURSA, LECIONA e PRÉ-REQUISITO. Veremos o Nucleo E-R mais detalhadamente no capítulo seguinte.

As relações contidas no esquema da Figura 1 são descritas como segue:

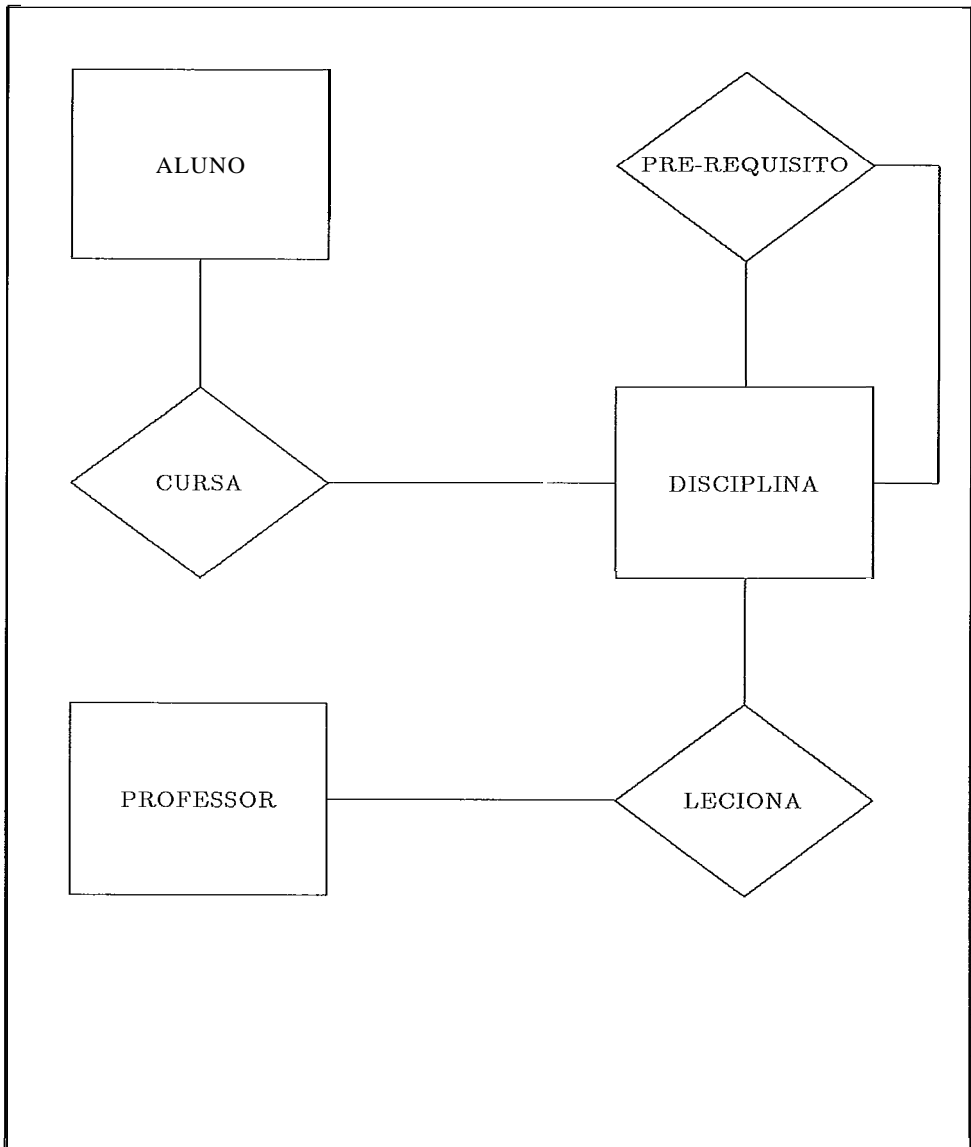


Figura 1. Esquema exemplo

PROFESSOR (PNOOME)

DISCIPLINA (DNOOME, CRÉDITOS)

ALUNO (ANOOME)

CURSA (ANOOME, DNOOME, NOTA)

LECIONA (PNOOME, DNOOME, PERÍODO)

PRÉ-REQUISITO (DNOOME, PREREQ)

Obs:

- A relação PROFESSOR representa os professores da base de dados e, nela, o atributo PNOOME, representando o nome do professor é também chave da relação;
- A relação DISCIPLINA representa as disciplinas da base de dados e, nela, os atributos que correspondem ao nome e número de créditos da disciplina são, respectivamente, DNOOME e CRÉDITOS. Somente o primeiro representa a chave da relação;
- A relação ALUNO representa os alunos da base de dados e, nela, o atributo relativo ao nome do aluno, ANOME, é também sua chave;
- A relação CURSA representa a associação entre as relações ALUNO e DISCIPLINA. Seus atributos que correspondem ao nome do aluno, ao nome da disciplina e à nota do aluno naquela disciplina são ANOME, DNOOME e NOTA. Tanto ANOME quanto DNOOME são chaves desta relação;
- A relação LECIONA, por sua vez, representa a associação entre as relações PROFESSOR e DISCIPLINA. Seus atributos que correspondem ao nome do professor, ao nome da disciplina e ao período em que o professor lecionou naquela disciplina são PNOOME, DNOOME e PERÍODO. Tanto PNOOME quanto DNOOME são chaves desta relação;
- Finalmente, a relação PRÉ-REQUISITO representa a associação dupla da relação DISCIPLINA com ela mesma. Possui dois atributos que correspondem a nomes de disciplinas. Receberam nomes diferentes com a finalidade de facilitar sua manipulação. São eles DNOOME e PREREQ, onde tanto um quanto o outro são chaves da relação.

Como já mencionamos, linguagens foram e têm sido desenvolvidas com base na Álgebra e no Cálculo Relacional, visando acesso a bancos de dados para recuperação

ou modificação das informações. Entre estas linguagens podemos citar QUEL, SQL, QBE, entre outras. Para efeito de discussão, no entanto, iremos expressar aqui, em nosso exemplo, consultas utilizando três formas de manipulação de dados, quais sejam:

- *SQL*
- *QBE*
- *Codasyl*

A linguagem SQL (*Structured Query Language*) [7] utiliza um conjunto de palavras-chaves que formam blocos estruturados e que torna possível definir, manipular e controlar dados em um banco de dados relacional. Trata-se de uma linguagem mais facilmente assimilada pelo usuário, embora apresente dificuldades em se expressar consultas em que sejam necessárias as operações de *União* e *Diferença*. Esta situação será vista mais adiante.

Por sua vez, QBE (*Query By Example*) [37] tem como característica principal a formulação de consultas a uma base de dados através do preenchimento de tabelas ao invés da formulação escrita de comandos. A resposta à consulta também se dá via tabelas, o que lhe dá um aspecto mais amigável de manipulação de dados do que o SQL [6].

Com relação ao *Codasyl* (*Conference on Data Systems Language*) [3][29] não se trata propriamente de uma linguagem. Para esclarecermos sua presença como forma de manipulação de dados, baseamo-nos nas referências [6], [29], [34], [15] e [27].

Codasyl é uma Conferência (*Conference on Data Systems Languages*) com o objetivo de definir algumas normas e padrões no que se refere a linguagens, tanto de programação quanto de manipulação de bases de dados. O IDMS (*Integrated Data Management System*) é um produto da Cullinet Software Inc que roda no IBM com sistemas operacionais como DOS, MVS, etc. O IDMS é, provavelmente, o exemplo melhor conhecido do que se refere ser um *Sistema Codasyl* (ou, às vezes, *Sistema*

DBTG), ou seja, um sistema baseado nas propostas do DBTG (*Data Base Task Group*) do Comitê de Linguagem de Programação (posteriormente renomeado para Comitê COBOL) do *Codasyl*, a organização responsável pela definição do COBOL. O relatório final do DBTG foi produzido em 1971 e vários sistemas nele baseados foram construídos durante a década de 70, entre eles o IDMS com base no Modelo de Redes [29][27].

O relatório DBTG continha propostas para três linguagens de Bancos de Dados distintas: uma linguagem de descrição de dados esquema (LDD esquema), uma linguagem de descrição de dados subesquema (LDD subesquema) e uma linguagem de manipulação de dados (LMD). O propósito das três linguagens era como segue:

- A LDD esquema era uma linguagem para descrever um Banco de Dados estruturado em redes. O termo “esquema” do DBTG corresponde muito aproximadamente ao termo “esquema conceitual” da ANSI/SPARC [9] mas a LDD esquema do DBTG é realmente muito mais interno que conceitual (veremos no Capítulo III);
- A LDD subesquema era uma linguagem para definir uma visão externa do Banco de dados;
- A LMD consistia de um conjunto de operadores para manipulação de um Banco de Dados em redes definido através das duas linguagens de descrição dos dados. Assume-se que o usuário seja um programador de aplicações. A LMD, então, terá uma sintaxe compatível com aquelas da linguagem de programação sendo utilizada.

Ainda com base no Date [6], utilizaremos aqui, para exemplificar nossas consultas, o IDMS que é um Sistema de Gerenciamento de Bancos de Dados (SGBD). Visando facilitar a compreensão do que encontra-se aqui exposto, utilizaremos o nome do modelo *Codasyl*, advindo do modelo de redes, para referenciarmo-nos à especificação da consulta que será mostrada.

Descrita com base no *Codasyl*, a consulta tem como característica ser procedimental, ou seja, uma consulta é formada por uma linguagem de programação acrescida de uma lista de comandos na LMD que compõem um procedimento ou programa que deve ser executado.

Baseado nestas características, podemos concluir que, devido ao seu aspecto gráfico, diferentemente das demais formas de consulta tradicionais, QBE, dentre elas, é a linguagem de mais fácil assimilação pelo usuário não-programador, isto é, aquela que menos exige conhecimento de programação por parte daqueles que irão utilizá-la.

111.3 - Descrição de Consultas

Com o objetivo de fazermos uma melhor avaliação comparativa destas linguagens relacionais e do *Codasyl* acompanhado de uma linguagem de programação, iremos, então, determinar quatro consultas distintas a serem aplicadas ao banco de dados descrito na Figura 1. Com base nos resultados teremos um melhor esclarecimento sobre a questão.

Apresentamos em seguida as consultas. São elas:

Consulta 1 : Obtenha os nomes dos alunos que cursam Cálculo

Consulta 2 : Obtenha os nomes de todos os professores que não lecionam *Física*

Consulta 3 : Obtenha as disciplinas lecionadas pelo professor João ou cursadas pelo aluno José

Consulta.4 : Obtenha as disciplinas que são pré-requisitos dos pré-requisitos de Cálculo *IV*

Aplicaremos a cada uma delas as linguagens SQL e QBE. Devido ao caráter mais complexo do modelo, apenas a consulta 1, por ser mais simples, será especificada utilizando-se como base o modelo *Codasyl*, pois os procedimentos expressos através dele são muito extensos e não traria um maior significado a este trabalho.

Em SQL as consultas são expressas da seguinte forma [6][7]:

Consulta 1 :

```
SELECT ANOME
FROM CURSA
WHERE DNOME = Cálculo
```

Consulta 2

```
SELECT DNOWIE
FROM PROFESSOR
WHERE NOT EXISTS
    ( SELECT *
      FROM LECIONA
      WHERE PNOME = PROFESSOR.PNOME
        AND DNOME = Física )
```

Consulta 3 :

```
SELECT DNOME
FROM LECIONA
WHERE PNOME = João
UNION
SELECT DNOME
FROM CURSA
WHERE ANOME = José
```

Consulta 4 :

```
SELECT PREREQUISITO1.PREREQ
FROM PREREQUISITO1 , PREREQUISITO2
WHERE PREREQUISITO2.PREREQ = PREREQUISITO1.DNOME
AND PREREQUISITO2.DNOME = Cálculo IV
```

Utilizando-se QBE na formulação das consultas, teremos [6][13][37]:

Consulta 1 :

CURSA	ANOME	DNOME
	P..ANOMEX	Cálculo

Consulta 2

PROFESSOR	PNOME	LECIONA	PNOME	DNOME
	P..PNOMEX	¬	SNOMEX	Física

Consulta. 3 :

LECIONA	DNOME	PNOME	CURSA	DNOME	ANOME
	_DNOMEX	João		_DNOMEY	José
	P.			_DNOMEX	
	P.			_DNOMEY	

Consulta. 4 :

PRE-REQUISITO	DNOME	PREREQ	
	_DNOMEX Cálculo IV	_PREREQY _PREREQX	P._PREREQY

Quanto ao modelo *Codasyl*, acrescido da linguagem de programação COBOL, a consulta 1 poderia ser expressa da seguinte forma [29][6][34][27]:

```
Move CALCULO to DNOME in DISCIPLINA
Find Calc DISCIPLINA
Find First CURSA Within DISCIP-CURSA
While CURSA Found
Perform
  Find Owner Within ALUNO-CURSA
  Get ALUNO
  (incluir ANOME In ALUNO à lista de resultados)
  Find Next CURSA Within DISCIP-CURSA
End-Perform
```

Observações:

1. ALUNO-CURSA e DISCIP-CURSA são formas de acesso à relação CURSA. A primeira está embutida em ALUNO e a segunda, em DISCIPLINA;
2. Mostraremos, a seguir, a definição das estruturas e dos acessos, segundo a LDD para COBOL do Codasyl:

Record Name is DISCIPLINA.

Location Mode is Calc Using DNAME Duplicates Not Allowed.

02 DNAME PIC X(30).

Record Name is ALUNO.

Location Mode is Calc Using ANOME Duplicates Not Allowed.

02 ANOME PIC X(40).

Record Name is CURSA.

Location Mode is via DISCIP-CURSA Set.

02 PERÍODO PIC X(4).

02 NOTA PIC X(3).

Set Name is DISCIP-CURSA.

Order is Next.

Owner is DISCIPLINA.

Member is CURSA Optional Manual.

3. A consulta 1 aplicada ao modelo Codasyl tem suas estruturas físicas aqui traduzidas de uma forma livre. A definição encontra-se incompleta pois foi construída abrangendo apenas os elementos da Figura 1 que fazem parte desta consulta que são DISCIPLINA, ALUNO e CURSA.

Vamos, então, a partir da análise das características de cada uma das quatro consultas, somada à análise das características de cada uma das três formas de acesso descritas, proceder à avaliação dos aspectos relevantes de cada expressão de consulta no que diz respeito à dificuldade que elas apresentam em suas representações.

Quanto à consulta 1, ela é considerada simples por ser composta somente de operações básicas da Álgebra Relacional. Tais operações pertencem à família de consultas que usam *Seleção*, *Projeção* e *Junção*. Por esta razão ela não oferece dificuldades em ser expressa em nenhuma das formas.

Com respeito à consulta 2, no entanto, não podemos afirmar o mesmo. Devido à negação do fato de "Lecionar Física", surge um complicador na representação da consulta, devendo esta ser feita através da diferença entre conjuntos. Na Álgebra Relacional esta operação é chamada DIFERENÇA. Em QBE esta diferença é complicada de ser feita, podendo ser expressa através de inserção e deleção de registros de uma relação fictícia e por meio do operador de negação. Utilizando SQL a operação de DIFERENÇA torna necessário o aninhamento de consultas e o uso da negação NOT EXISTS. Todos estes artifícios tornaram difícil a visualização ou percepção do resultado.

Com relação à consulta 3, ela utiliza UNIÃO entre conjuntos. Sua representação em SQL é clara. Apesar disso, da mesma forma que na consulta 2, especificada em SQL, podemos perceber que a operação de UNIÃO simplesmente reúne duas consultas que, no caso, são simples como a consulta 1. UNIÃO e DIFERENÇA são operações que, em SQL, não conseguem ser expressas implicitamente como o são as operações *Seleção, Projeção, Junção*. Entretanto, em QBE não é assim tão claro que se está processando uma operação de UNIÃO. A representação não permite esta clareza.

A consulta 4 trata de um auto-relacionamento, ou seja, é um relacionamento entre duas entidades cujos atributos e respectivos domínios são iguais. No caso, a entidade é DISCIPLINA. Uma disciplina está relacionada a outra pelo fato da segunda ser pré-requisito da primeira.. Este tipo de consulta é complicado de ser representado e compreendido tanto em SQL quanto em QBE. Podemos verificar esta dificuldade quando, em SQL, precisamos qualificar as variáveis do relacionamento PRÉ-REQUISITO com seu nome seguido do sufixo 1 ou 2 (PRÉ-REQUISITO1.DNOME, PRÉ-REQUISITO2.PREREQ, etc). Em QBE o mesmo ocorreu, ou seja, as variáveis foram qualificadas com X e Y como sufixo (DNOMEX, PREREQY, etc). A visualização torna-se ruim.

Já a consulta utilizando como base o modelo Coclasyl, como pudemos observar no exemplo mostrado, possui descrição bastante longa. Aquela consulta 1 que, em

SQL, ocupou três linhas ou três comandos, com o Codasyl transformou-se em um pequeno programa repleto de especificações de estruturas.

Como podemos verificar, nenhuma das três formas de expressar consultas a um Banco de Dados é realmente simples. A consulta com base no modelo *Codasyl* e a linguagem SQL são procedimentais, sendo a primeira bem mais trabalhosa que a segunda, mas ambas exigem do usuário conhecimento prévio em programação. Por sua vez, a linguagem QBE, embora de aparência gráfica e não exigindo tais requisitos daqueles que dela se utilizam, não permite uma razoável percepção da consulta que está sendo expressa através dela como foi visto. Como já mencionamos, é visualmente pobre, o que torna não trivial a formulação de consultas por um usuário não experiente.

11.4 - Conclusão

Com base nas observações tecidas ao longo deste capítulo, podemos verificar que as formas tradicionais de consultas a bancos de dados não satisfazem, isoladamente, às características necessárias à formulação de consultas de modo amigável.

Vamos citar algumas das características encontradas nas consultas especificadas:

- a) Como pudemos observar, a consulta 1 foi expressa mais facilmente que as demais em todas as formas de elaboração de consultas. Isto deve-se ao fato desta consulta requerer, em correspondência com a Álgebra Relacional, somente as operações pertencentes à família de consultas que usam *Seleção*, *Projeção* e *Junção*. Consultas expressas a partir dessas três operações são consideradas relativamente fáceis de serem formuladas através de quaisquer modelos;
- b) As consultas 2 e 3, principalmente se observadas através de sua expressão em SQL, mostram que, ao serem formuladas, deram origem a consultas simples expressas nos moldes mostrados no item a, ligadas uma à outra através de um operador

explicitamente identificado. Esta situação retrata a consulta que necessita das operações algébricas chamadas DIFERENÇA e UNIÃO. A primeira é representada pelo NOT EXISTS e a segunda, pelo UNION. Todas as consultas que precisam ser expressas utilizando estas operações são consideradas mais complicadas, mesmo quando a formulação das consultas ocorrer através de interfaces gráficas;

- c) A consulta 4 apresenta, também, um certo grau de dificuldade ao ser expressa pelo fato de ser recursiva. O fato de, através dessas formas tradicionais de consultas, o usuário precisar dar nomes às estruturas do BD e a seus atributos, faz com que a expressão se torne complicada, de difícil compreensão e formulação.

O usuário não experiente deveria ter acesso a uma forma simplificada de expressar consultas ao BD. Para isso, seria interessante que pudéssemos descartar as desvantagens de cada uma das formas apresentadas, atendo-nos aos seus pontos positivos. Deste modo, devemos aproveitar, de cada uma das formas de expressão de consultas, suas características que visam facilitar sua formulação:

- A linguagem QBE tem como característica importante, para facilitar a manipulação por parte do usuário, o seu aspecto gráfico. O usuário não precisará conhecer ou lembrar de detalhes dos atributos das estruturas do BD. Não há uma especificação formal da linguagem para que se elaborem as consultas, ou seja, o que é feito internamente para acessar o BD é transparente ao usuário. Caberá ao usuário especificar somente condições, atributos desejados, entre outros;
- A linguagem SQL, por sua vez, não possui caráter gráfico. No entanto, muitas consultas apresentam dificuldades ou até impossibilidades em serem expressas graficamente. A definição de condições de seleção é um dos casos mais complicados que podemos citar. Pensemos em uma consulta que envolva aquelas operações mencionadas DIFERENÇA e UNIÃO. Seria bastante complexo especificá-la graficamente. Neste caso, precisamos obter do SQL uma outra característica importante na formulação de consultas: especificação de consultas com o uso de

alguns comandos, ou seja, uma certa forinalização de linguagem;

- Também o aspecto procedimental advindo das expressões desenvolvidas a partir do modelo Coclasyl poderão ser interessantes na forinalização da linguagem.

CAPÍTULO IV

MODELOS DE DADOS

IV.1 - Introdução

O Modelo de dados adotado neste trabalho foi o Modelo Entidade-Relacionamento, segundo Chen [2], com algumas extensões. A vantagem principal na escolha deste modelo está na possibilidade de utilizá-lo no nível conceitual do projeto (trata-se de um modelo semântico) e, também, como representação do esquema do banco de dados gerado. Isto porque, 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), em geral com base na arquitetura do sistema hospedeiro. O que o projeto *UniversiData* [25], onde este trabalho está inserido, assume é que, também na fase de implementação, o projeto da base de dados deverá ser baseada no Modelo E-R. Um aspecto de grande importância nessa abordagem está na uniformização do conhecimento da base de dados para usuários, analistas e programadores.

Em seguida, veremos algumas das principais considerações que nos levaram a esta escolha.

IV.2 - Considerações sobre Modelos

Como se sabe, a arquitetura proposta pela ANSI/SPARC [9] para sistemas de Bancos de Dados é dividida em três níveis que são [29][6]:

- Externo : é aquele que descreve o mundo real segundo a visão de cada usuário individualmente;

- Conceitual : consiste na representação de todas as informações contidas no banco de dados, como no nível externo, embora se encontre com uma visão mais abstrata do que aquela a partir da qual os dados serão fisicamente armazenados; e
- Interno : este nível retrata a forma física final com que os dados serão armazenados em meios computacionais.

Tradicionalmente, sistemas que utilizam-se de Bancos de Dados têm sido desenvolvidos seguindo esta arquitetura. Pouca atenção tem sido dada à modelagem conceitual da base de dados. A preocupação maior tem ficado por conta da implementação. A Modelagem do Esquema Conceitual, entretanto, é de vital importância e deve refletir o mundo real das aplicações de forma simples, completa e correta.

Uma aplicação é bem sucedida se for feita uma modelagem de acordo com a realidade, com clareza e facilidade de manipulação a partir do problema inicial. É fundamental que os usuários envolvidos possuam completo entendimento da descrição lógica da base de dados. A não atenção a este aspecto conduz a muitos erros quando de sua implementação em sistemas convencionais de gerenciamento de Bancos de Dados. Esses erros são mantidos desde a fase de programação e por toda a fase de manutenção do sistema.

Ao longo do tempo várias propostas de modelos de dados têm sido feitas visando possibilitar captar mais informações e representá-las através dos modelos. Fazem parte destas discussões os Modelos Semânticos [16] surgidos ao longo da década de 70. Desejava-se representar as informações de acordo com o Modelo Conceitual com visão mais próxima do usuário. O conjunto de propriedades que se desejava que o modelo possuísse seria [13]:

- o modelo conceitual deve poder evoluir;
- deve poder representar uma visão estável do mundo real;

- deve suportar uma série de modelos externos; e
- deve ser fácil de implementar.

Surgiu, então, o Modelo Entidade-Relacionamento proposto por Chen [2] que, embora simples, atende a grande parte das aplicações convencionais de computadores. Com o passar do tempo, muitas propostas de extensão a este Modelo foram e têm sido feitas.

Vamos, então, descrever as características mais importantes do Modelo Entidade-Relacionamento.

IV.3 - O Modelo Entidade-Relacionamento

Introduzido por Chen, o Modelo Entidade-Relacionamento foi apresentado como sendo uina "base para unificação de diferentes visões dos clacos". Essas visões seriam aquelas relativas aos três principais modelos de clacos que são o Relacional, de Redes e Hierárquico [1][4][27][29][28][6][34]. Desde que foi introduzida em 1976, sua metoclogia tem sido considerada uma importante disciplina para análise e modelagem de dados. O modelo apresenta um alto grau de independência dos dados e incorpora algumas das importantes informações semânticas do mundo real, servindo como representação do Modelo Conceitual do Banco de Dados.

Os elementos básicos desta representação são entidades e relacionamentos. Vamos descrever os principais elementos do MER [20]:

- *Entidade* : é algo que pode ser distintamente identificado. Entidades são representações de "objetos" do mundo real e podem ser grupadas formando conjuntos de entidades (objetos são os seres, os fatos, as coisas e os organismos sociais). Assiin sendo, são entidades as representações abstratas de um professor, de uma clata, de uma disciplina, de um departamento, de um livro, etc. Um grupo de en-

tidades com características semelhantes formam um conjunto de entidades como, por exemplo, o conjunto de professores, o conjunto das datas, etc. Os conjuntos de entidades são graficamente representados no MER por um retângulo;

- *Atributos* : são informações que descrevem as entidades. Assim, segundo Chen, um atributo pode ser compreendido como uma função que liga um ponto de um conjunto de entidades a um ponto de um conjunto de valores. O domínio da função é um conjunto de entidades e o contra-domínio é um conjunto de valores. Como exemplos podemos citar: *nome, endereço, salário*, etc;
- *Relacionamentos* : são associações entre elementos dos conjuntos de entidades. Há informações sobre as entidades que se encontram embutidas nos relacionamentos como, por exemplo, o fato do PROFESSOR *Pedro* lecionar a DISCIPLINA *Computação I*. Como DISCIPLINA é uma entidade, em geral, não pode ser atributo de PROFESSOR. Portanto, representa-se um relacionamento entre PROFESSOR e DISCIPLINA. Também os relacionamentos podem possuir atributos como vimos no atributo *nota* de CURSA. A representação gráfica de um relacionamento se faz através de um losango;
- *Auto-Relacionamento* : é um tipo especial de relacionamento que é caracterizado por relacionar elementos de um conjunto de entidades com elementos desse mesmo conjunto de entidades, como é o caso de PRÉ-REQUISITO que relaciona disciplinas com disciplinas. Isto é, *Cálculo I* é pré-requisito de *Cálculo II*;
- *Papel ("Role")* : papel de uma entidade em um relacionamento é a função que ela desempenha nesse relacionamento. Há casos onde é muito importante se representar os papéis. Um desses casos é a dupla ligação entre a entidade DISCIPLINA e o relacionamento PRÉ-REQUISITO, onde um dos papéis seria "*disciplina*" e o outro seria "*pré-requisito*".

Além dos elementos básicos mostrados, o MER possui, também, outros elementos que são caracterizados como propriedades do esquema, sendo considerados, portanto,

restrições de integridade. São eles:

- **Entidade Fraca** : é uma entidade existencialmente dependente de uma outra entidade, via um relacionamento. Para exemplificarmos, consideremos a Figura 2. A entidade **DEPENDENTE** é fraca pois possui dependência existencial da entidade **EMPREGADO**, isto é, não haverá sentido na existência de dependentes que não possuam algum empregado associado a eles. A representação gráfica da entidade fraca faz-se através de dois retângulos, como podemos ver na Figura 2;
- **Chaves** : são conjuntos de um ou mais atributos que identificam exatamente uma instância de uma determinada entidade ou relacionamento;
- **Cardinalidade** : expressa o número máximo de instâncias de uma entidade que podem estar relacionadas a uma instância de outra entidade, via um relacionamento. Em um relacionamento binário a cardinalidade de mapeamento pode ser $1:1$, $1:n$, $n:1$ e $m:n$. No exemplo da Figura 2, podemos observar que um empregado pode ter n dependentes enquanto que um dependente pode estar associado a somente 1 empregado. Vendo de outra forma, temos que a uma instância de **EMPREGADO** podem corresponder até n instâncias de **DEPENDENTE**, enquanto que uma instância de **DEPENDENTE** pode estar combinada com apenas uma instância de **EMPREGADO**.

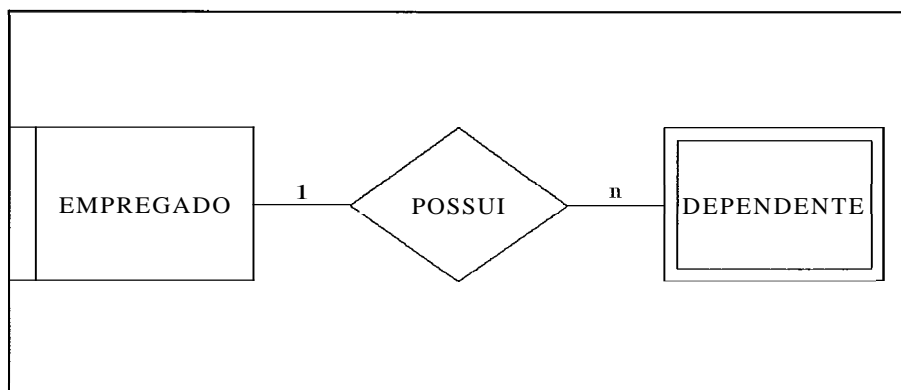


Figura 2. Entidade Fraca / Cardinalidade

IV.4 - Extensões ao Modelo Entidade-Relacionamento

Com o passar do tempo, muitos trabalhos têm sido realizados com base na proposta inicial de Chen [2] e em muitos deles foram propostas extensões ao Modelo Entidade-Relacionamento. Neste trabalho apresentaremos somente as extensões que utilizaremos [25].

A primeira delas é a adoção da representação de Hierarquia de Tipos através da Generalização e Especialização. A *Generalização* é uma abstração das partes comuns de duas ou mais entidades gerando uma entidade em nível superior, ou super-entidade, ou supertipo. Por sua vez, a *Especialização* é uma abstração das diferenças gerando entidades em nível inferior, ou sub-entidades, ou subtipos. A entidade que é subtipo herda os atributos de seu supertipo [16]. Consideraremos para efeito deste trabalho que uma entidade pode ter somente um supertipo. Como exemplo, poderíamos criar uma generalização de nome PESSOA que englobaria as características em comum entre as entidades PROFESSOR e ALUNO. Sua representação se faz através de um triângulo como se vê na Figura 3.

A segunda extensão adotada trata-se das ordens que definem um acesso, similar às chaves, não obrigatoriamente único, às entidades e relacionamentos. O acesso ordenado não é suportado pelo Modelo Entidade-Relacionamento em sua definição original devido a sua função inicial de servir à representação conceitual de dados, e não à implementação física. Acesso ordenado a bancos de dados é largamente utilizado durante a programação e pode ser obtido a partir de:

- um *domínio*, que consiste no conjunto de elementos de entidades ou relacionamentos que devem ser ordenados;
- uma expressão de seleção, que determina quais os elementos do domínio que deverão aparecer na ordem final; e

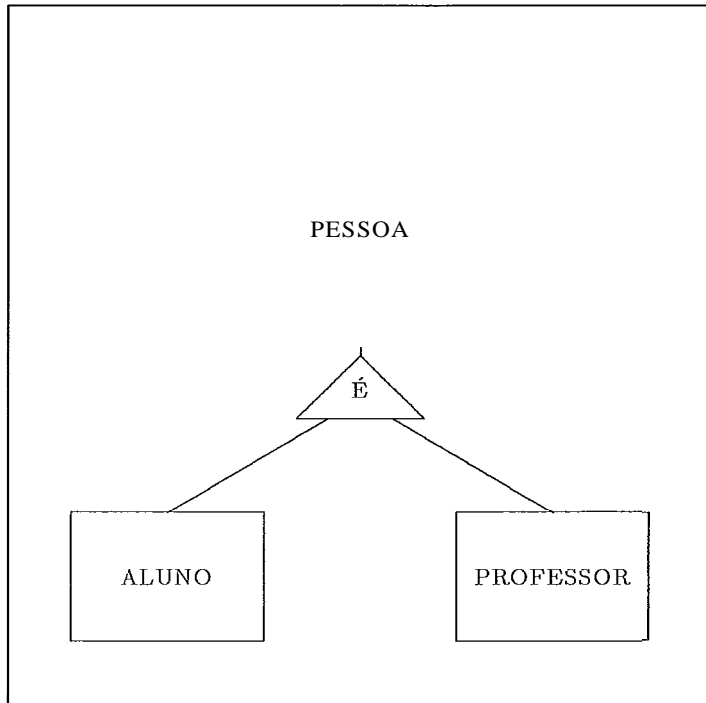


Figura 3. Generalização

- os *determinantes da ordenação*, que são atributos ou outros termos do domínio da ordenação que determinam o critério de ordenação a ser seguido.

Poderíamos ter em ALUNO (domínio) os elementos ordenados alfabeticamente pelo nome (determinante) como vemos na Figura 4.

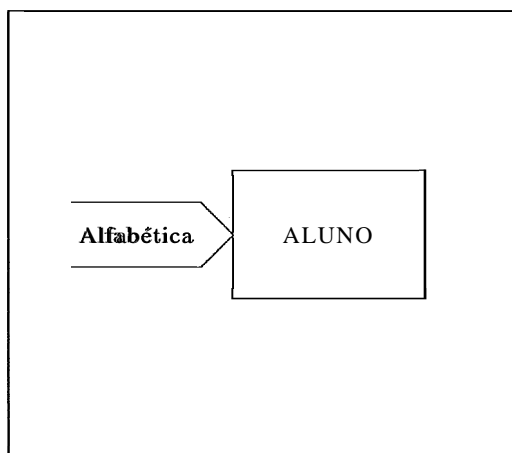


Figura 4. Ordens

Por sua vez, a terceira extensão adotada para o modelo E-R diz respeito à cardinalidade do relacionamento. Isto é, além do valor máximo já descrito, haverá, também, um valor mínimo de correspondência entre as instâncias das entidades envolvidas no relacionamento [12]. Sendo assim, temos a Figura 5 constituindo uma extensão à Figura 2, mostrando que um empregado pode ter no mínimo zero e no máximo n dependentes enquanto que um dependente deve ter no mínimo 1 e no máximo também 1 empregado relacionado a ele.

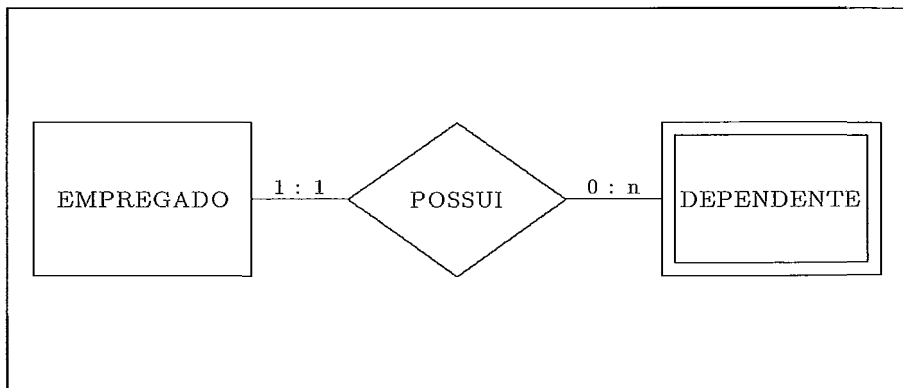


Figura 5. Cardinalidade

IV.5 - Especificação Formal de uma Linguagem

O Modelo E-R é utilizado para representação das estruturas do banco de dados. Não contém, no entanto, uma linguagem própria para representação de manipulações. Há inúmeras propostas nesse sentido. O que faremos aqui, entretanto, é definir uma expressão geral para ser usada na construção de um modelo teórico [23][24].

Para que se chegue à expressão geral é preciso que sejam definidos os elementos sintáticos como mostraremos em seguida. Veremos que há a facilidade de, juntando-se os elementos sintáticos, expressar melhor quaisquer tipos de consultas.

Os elementos sintáticos são:

- *Variáveis* : v , v_1 , v_2 , ...;
- *Constantes* : c , c_1 , c_2 , ...; (representadas diretamente pelos seus valores)
- *Símbolos de Atributos* : a , a_1 , a_2 , ...;
- *Símbolos de Funções Papel* : ρ , ρ_1 , ρ_2 , ...;
- *Símbolos de Funções* : f , f_1 , f_2 , ...; cada símbolo de função f tem um grau n associado ($n \geq 0$)
- *Símbolos de Agregações* : g , g_1 , g_2 , ..
- *Negação* : \neg ;
- *Conectores Lógicos AND, OR* : \wedge , \vee ;
- *Símbolos de Entidades* : E , E_1 , E_2 , ...;
- *Símbolos de Relacionamentos* : R , R_1 , R_2 , ...; cada símbolo de relacionamento R tem um grau n associado ($n \geq 2$)
- *Símbolos Relacionais* : “=” , “>” ; outros símbolos relacionais podem ser obtidos da combinação destes com o símbolo de negação
- *Termos* : Há dois tipos de termos: Termos-C e Termos-E:

São Termos-C; que denotam elementos no domínio das constantes:

- (i) uma variável v seguida de *um* ponto e um símbolo de atributo, como $v.a$, por exemplo;
- (ii) um símbolo de função de grau n , seguido de uma lista da forma (t_1 , \dots , t_n) onde t é um termo;

(iii) uma constante; e

(iv) uma expressão da forma $g.n(Q)$, onde n é um inteiro, Q é uma consulta.

São Termos-E; que denotam instâncias de entidades ou relacionamentos:

(i) uma expressão da forma $\rho(t)$, onde t é um Termo-E;

(ii) uma variável.

- *Átomos* : São átomos:

(i) uma expressão da forma $t \in X$, onde t é um Termo-E e X é um símbolo de entidade ou relacionamento;

(ii) dois Termos-C separados por um símbolo relacional; e

(iii) dois Termos-E separados pelo símbolo relacional “=”

- *Quantificadores* : se e é uma variável e X é um símbolo de entidades E ou um símbolo de relacionamentos R :

(i) $\forall v \in X$ é um quantificador universal (restrito);

(ii) $\exists v \in X$ é um quantificador existencial (restrito);

- *Fórmulas* : fórmulas bem formadas (*fbf*) são recursivamente definidas como:

(i) um átomo é uma *fbf*;

(ii) se ϕ é uma *fbf*, então $\neg\phi$, (ϕ) são *fbf*s;

(iii) se ϕ_1 e ϕ_2 são *fbf*s, então $\phi_1 \wedge \phi_2$, $\phi_1 \vee \phi_2$ são *fbf*s;

(iv) se ϕ é uma *fbf* e π é um quantificador, então $\pi\phi$ é uma *fÓf*.

- Consultas : uma consulta é uma expressão da forma.

$$t_1, \dots, t_n \mid v_1 \in X_1, \dots, v_k \in X_k \mid \phi(v_1, \dots, v_k)$$

onde t_i é um termo, X_i é um símbolo de entidades ou relacionamentos e ϕ é uma fórmula bem formada. As únicas variáveis livres ocorrendo em t_1, \dots, t_n e em ϕ são v_1, \dots, v_k , n é o número de componentes de Q ou o grau de Q.

Analisando os elementos sintáticos apresentados, faremos algumas observações como seguem.

Uma consulta admite apenas Termos-C como resultados. Isto exclui variáveis que são entidades, o que é razoável, pois não haveria sentido ter-se abstrações como resultado de uma consulta.

Os símbolos de funções referem-se a funções básicas como *seno*, *abs*, etc. Funções são aplicadas a Termos-C. Funções de agregação, por sua vez, aplicam-se a conjuntos (consultas) em determinada coluna, indicada pelo n .

Obs: Devemos lembrar que, embora tenhamos mencionado a existência de ordens como integrantes do Modelo Entidade-Relacionamento que iremos adotar, elas não estão aqui definidas. Isto porque o maior interesse desta seção consiste na formulação de uma expressão geral, não necessariamente completa, mas que permitisse que compreendêssemos melhor a especificação formal da linguagem.

Considere os seguintes exemplos de consultas:

Exemplo 1: Listar os nomes dos alunos.

al.ANOME | *al E ALUNO*

Exemplo 2: Listas os nomes dos alunos, disciplinas cursadas e notas.

$$al.ANOME, dis.DNOME, (al,dis).NOTA \mid al \in ALUNO, \\ clis \in DISCIPLINA \mid CURSA(al, clis)$$

al e dis são quantificadores sobre os conjuntos de entidades ALUNO e DISCIPLINA. Note que $(al,dis).NOTA$ denota o atributo de uma instância do relacionamento CURSA que, garantidamente, existe devido à fórmula CURSA (al,dis) , isto é, a condição de que a instância (al,dis) está no relacionamento CURSA é verdadeira.

Exemplo 3: Listar a média das notas de *Cálculo IV*

$$avg1\{ (al,dis).NOTA, al.ANOME \mid al \in ALUNO, \\ clis \in DISCIPLINA \mid dis.DNOME = "CalculoIV" \}$$

Neste exemplo a função *avg* foi aplicada à primeira coluna do resultado da consulta que lista notas e nomes dos alunos para disciplinas cujos nomes sejam Cálculo IV. Note que foi necessário acrescentar ANOME ao conjunto ao qual avg se aplica, de modo a evitar que duplicações desaparecessem do resultado, pois uma expressão denota sempre um conjunto de valores.

Exemplo 4: Listar os departamentos tal que todos os alunos estejam inscritos em alguma de suas disciplinas oferecidas.

$$d.NOME \mid d \in DEPARTAMENTO \mid \\ \forall al \in ALUNO \exists dis \in DISCIPLINA, \\ CURSA(al, clis) \wedge OFERECE(d, clis)$$

Neste exemplo, nós introduzimos quantificadores na fórmula da expressão geral. Observe que a única variável livre em ϕ é d , associada com DEPARTAMENTO fora da fórmula.

Uma linguagem é considerada completa se através dela for possível expressar qualquer tipo de consulta. Dessa forma, a partir da expressão geral de consultas a que chegamos seremos capazes de avaliar as classes de expressões que são cobertas pela linguagem de manipulação de dados adotada.

IV.6 - Conclusão

Neste capítulo, após uma breve introdução sobre modelos de dados, descrevemos aspectos do Modelo Entidade-Relacionamento adotado neste trabalho. O Modelo E-R é utilizado no nível conceitual de projeto de banco de dados e, também, como representação da implementação da base de dados. Através dele o usuário pode apresentar maior facilidade em usar a interface gráfica para ter acesso ao banco de dados, pois trata-se de um modelo bem mais amigável do que as formas de consultas mostradas no capítulo anterior. Com a finalidade de ampliarmos a utilização deste modelo, sugerimos, ainda, algumas extensões relativas a ordens, generalizações e cardinalidades dos relacionamentos. Foi também desenvolvida uma expressão geral de consultas com base no Modelo.

CAPÍTULO V

O PROJETO UNIVERSIDATA

V.1 - Introdução

Com base nos capítulos anteriores onde foram analisados linguagens de consultas a bancos de dados e modelos de dados com ênfase ao Modelo E-R, podemos, então, falar sobre o ambiente no qual se insere este trabalho. Esse ambiente é, na verdade, um projeto que está sendo desenvolvido no Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro denominado UniversiData [25] e cujos objetivos e características iremos descrever.

V.2 - O Projeto e seus Objetivos

Desenvolvido no Núcleo de Computação Eletrônica o Projeto UniversiData tem como objetivo integrar as informações ao gerar um banco de dados único dos diversos sistemas da UFRJ, "fornecendo 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 o número de duplicações, interpretações errôneas, lacunas de modelagem e heterogeneidade nos sistemas automatizados que operam na Universidade" [25].

A fim de promover a evolução deste Projeto e com a preocupação de mantê-lo organizado apesar de sua complexidade e mantível, surgiu, então, a idéia do desenvolvimento de uma ferramenta automatizada que permitisse construir esse banco de dados único com características favoráveis:

- a) ao bom entendimento do mesmo por parte do usuário;

- b) ao processo de manutenção dos dados;
- c) à facilidade de se executar tarefas simples de acesso ao banco de dados tais como produção de relatórios e execução de consultas *ad hoc*;
- d) ao desejo de se introduzir facilidades para utilização de técnicas de inteligência artificial para manipulação da base de dados; e
- e) à manutenção do sistema o mais possível independente do equipamento utilizado permitindo a utilização da base de dados em redes, com uma interface completamente uniforme.

Essa ferramenta trata-se do QUICK-DB, que não é uma linguagem, e sim um ambiente integrado de acesso a Bancos de Dados através da modelagem ER [22].

O QUICK-DB é um sistema de apoio à modelagem de dados e ao projeto de bancos de dados. Tem como características:

- a) a utilização de um modelo somente na fase Conceitual e como representação da base de dados gerada, no caso, o modelo adotado é o Entidade-Relacionamento com as extensões já descritas anteriormente; e
- b) a utilização de interfaces gráficas:
 - b.1) na construção de diagramas;
 - b.2) na manipulação de dados; e
 - b.3) na formulação de consultas.

V.3 - Os Componentes do QUICK-DB

O QUICK-DB é formado por vários componentes interligados como descrito na Figura 6.

Estes componentes representam módulos do sistema que formam um conjunto de ferramentas de *softwares* cujas interfaces são similares e amigáveis e que caracterizam um ambiente integrado e homogêneo. Os componentes em questão são descritos em seguida:

- *Diagramador E-R* : este componente consiste de uma ferramenta gráfica amigável a partir da qual o QUICK-DB permite que o usuário especifique os diagramas E-R, orientado por meio de menus. Durante essa fase o usuário deverá definir os elementos do Modelo E-R tais como entidades, relacionamentos, atributos, bem como os elementos adicionais como chaves, ordens de acesso, generalizações, cardinalidades dos relacionamentos e eventuais restrições de integridade. O diagramador foi desenvolvido utilizando a linguagem de programação Turbo Pascal 5.5 [32]. O diagrama gerado servirá como base para todos os demais componentes juntamente com o dicionário de dados;
- *Dicionário de Dados* : à medida que o diagrama estiver sendo definido, como mostramos no item anterior, o sistema gera o dicionário de dados correspondente. Dessa forma ele é totalmente orientado ao Modelo E-R, ou seja, é caracterizado por representar, a nível de dicionário de dados, todos os elementos desse modelo, como por exemplo a herança de atributos por parte de entidades que participam de uma *Generalização*. O dicionário deverá armazenar, também, informações necessárias a outros componentes do projeto como um todo, tais como relatórios, consultas e suporte às aplicações;

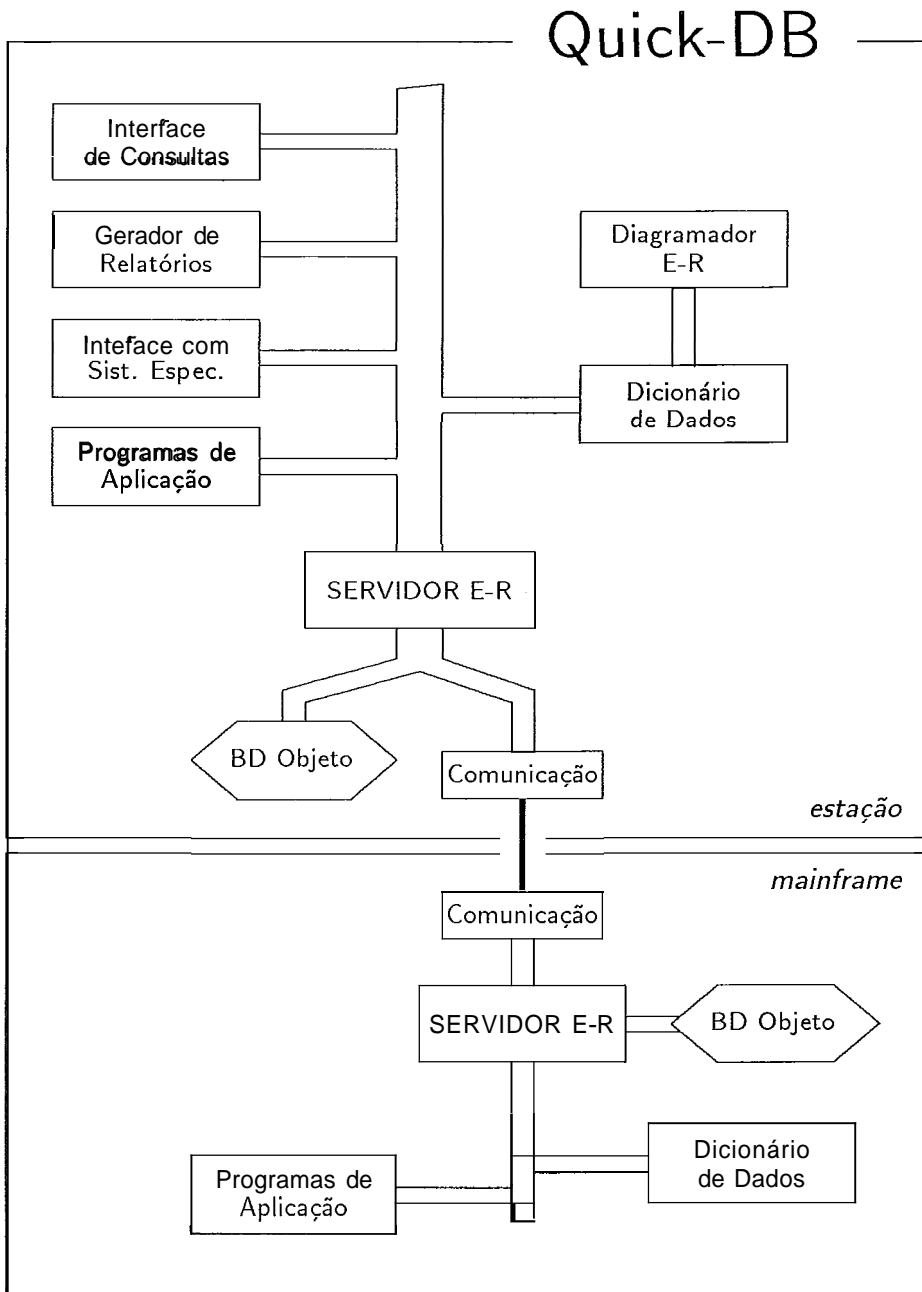


Figura 6. Componentes do QUICK-DB

- *Gerador de Bancos de Dados* : na fase de projeto da base de dados, a partir do diagrama E-R especificado e armazenado no dicionário de dados, a descrição do Banco de Dados Objeto é gerada na linguagem de definição de Bancos de Dados correspondente ao SGBD utilizado. Observam-se para esta conversão a estrutura do diagrama original e suas propriedades. Há um protótipo construído para esta etapa responsável por gerar a descrição de um BD para ser usado remotamente [21]. Utiliza-se para isto a LDD (Linguagem de Definição de Dados) desenvolvida pela Unisys chamada DASDL que faz parte do *software* DMSII. No entanto, outras linguagens poderão ser também consideradas. Além deste protótipo envolvido com base de dados remota, há um outro utilitário que é responsável por gerar a descrição de uma base de dados local juntamente com um conjunto de métodos capazes de permitir a manipulação dos dados desse BD autônomo criado na própria estação de trabalho [8]. Ambos os geradores, tanto aquele relacionado ao ambiente remoto quanto aquele responsável pelo BD local, necessitam da existência de um servidor que permita a ligação das interfaces com o BD de forma direta ou remota. Na ligação remota deverá haver uma comunicação entre o microcomputador e o equipamento de grande porte que permita ao servidor ter acesso ao BD Objeto. Por sua vez, na ligação direta, como o próprio nome o diz, o servidor terá acesso ao BD Objeto diretamente sem mais comunicações;
- *Servidor de Bancos de Dados* : o acesso ao banco de dados através do Modelo E-R é feito por cada um dos componentes de *software* com um servidor E-R. A comunicação entre os módulos de consulta ou atualização e o Banco de Dados Objeto é representada por uma espécie de barra de comunicações que liga também o diagrama E-R e o dicionário de dados. Como já vimos, a ligação das interfaces com o Servidor pode ser direta ou remota, o que torna viável a existência de uma estação E-R onde se tem acesso ao BD Objeto através de uma ligação de padrão definido, atendida pelos servidores remotos, localmente e no equipamento de grande porte. Isto torna possível o aproveitamento das facilidades gráficas de computadores de pequeno porte aliadas à capacidade de processamento do sistema de grande porte. É o servidor que traduz uma consulta

feita ao BD através da interface representada pelo diagrama E-R em operações sobre esse BD Objeto. Sua função é coletar as informações provenientes de sua execução e enviá-las aos módulos que a solicitaram. Os programas de aplicação têm acesso ao banco de dados pela chamada a bibliotecas de funções que operam sobre o BD através do Servidor. Vamos falar deste assunto com maiores detalhes mais adiante;

- *Módulo de Interfaces de Consultas* : este módulo é responsável pela execução de consultas *ad hoc* a bancos de dados utilizando-se de interface gráfica amigável, orientada por menus. Seu objetivo é permitir que usuários casuais, não técnicos, encontrem facilidade em sua utilização, formulando consultas, dirigidos pelo sistema, diretamente sobre um diagrama E-R. Através dele o usuário poderá consultar, por meio de interface gráfica, não só bancos de dados locais como também remotos, de grande porte, com o auxílio de servidores. Deve-se deixar claro, no entanto, que o usuário não terá conhecimento do ambiente que estará sendo utilizado. Este módulo é o assunto de que trata este trabalho;
- *Gerador de Relatórios* : este módulo consiste num grupo de programas total ou parcialmente automáticos que visa gerar relatórios de média ou baixa complexidade. Seu objetivo é tornar as aplicações cada vez mais independentes da forma com a qual a base de dados foi implementada, no que se refere aos aspectos de modelagem e estrutura do banco, de modo a diminuir o impacto das mudanças na definição do BD sobre os programas de aplicação;
- *Gerador de Aplicações* : este componente tem como objetivo estabelecer uma conexão entre a metodologia adotada pelo QUICK-DB e as metodologias de desenvolvimento de sistemas. Dessa forma, a construção de sistemas estará se beneficiando das funções já definidas no ambiente de apoio oferecido pelo QUICK-DB tais como descrição das bases de dados, expressão de consultas e geração de relatórios. O resultado desse gerador é a construção de programas de aplicação que utilizam o servidor em tempo de compilação [17];

- *Interface com Sistemas Especialistas* : baseada no paradigma *Objeto-Atributo-Valor* para representação do conhecimento foi definida neste módulo uma linguagem para construção de regras que abriga elementos como termos e quantificadores, que são acoplados ao servidor de bancos de dados de forma natural e não ambígua. Regras passam a ser uma alternativa adicional para consulta e manipulação do banco de dados [14].

V.4 - Conclusão

Descrevemos, neste capítulo, as características do Projeto *UniversiData* no qual está inserido este trabalho. A ferramenta *QUICK-DB*, desenvolvida no âmbito deste projeto, utiliza o Modelo *E-R* para que o usuário defina o esquema de banco de dados no nível conceitual. A partir deste momento o usuário poderá solicitar que seja gerada a descrição do *BD*, através da qual o *BD Objeto* é gerado. Várias aplicações poderão ser usadas para acessar o *BD*, entre elas, *Interfaces de Consultas*, *Gerador de Relatórios*, *Gerador de Aplicações* e *Interface com Sistemas Especialistas*. Para isso foi desenvolvido um módulo gerador de métodos para bancos de dados orientados a objetos.

CAPÍTULO VI

O SERVIDOR DE BANCOS DE DADOS

VI.1 - Introdução

O trabalho que apresentaremos consiste no Módulo de Consultas do QUICK-DB que será visto com maiores detalhes no capítulo seguinte. Podemos adiantar, no entanto, que, visamos delimitar o assunto a ser tratado neste trabalho, atemo-nos às consultas que acessam o Banco de Dados local anteriormente definido. Desse modo, iremos citar as características do ambiente que envolve nosso Módulo de Consultas:

- O Gerador de Bancos de Dados gera a descrição de um BD para ser usado em microcomputador através do Turbo Pascal 5.5;
- O BD gerado é local à estação de trabalho;
- O Gerador de BD gera, ainda, um conjunto de métodos associados aos objetos definidos no Modelo E-R que nos permite manipular os dados do BD Local diretamente;
- Ao conjunto de métodos para bancos de dados orientados a objetos [26] gerado denominamos Servidor de Banco de Dados, que serve de ligação entre o *software* aplicativo, no caso a Interface de Consultas, e o BD Objeto propriamente dito.

Vamos, então, considerar o esquema exemplo da Figura 7 que será usado como base neste capítulo e no seguinte para maiores esclarecimentos a respeito do assunto enfocado. As características do diagrama são descritas em seguida:

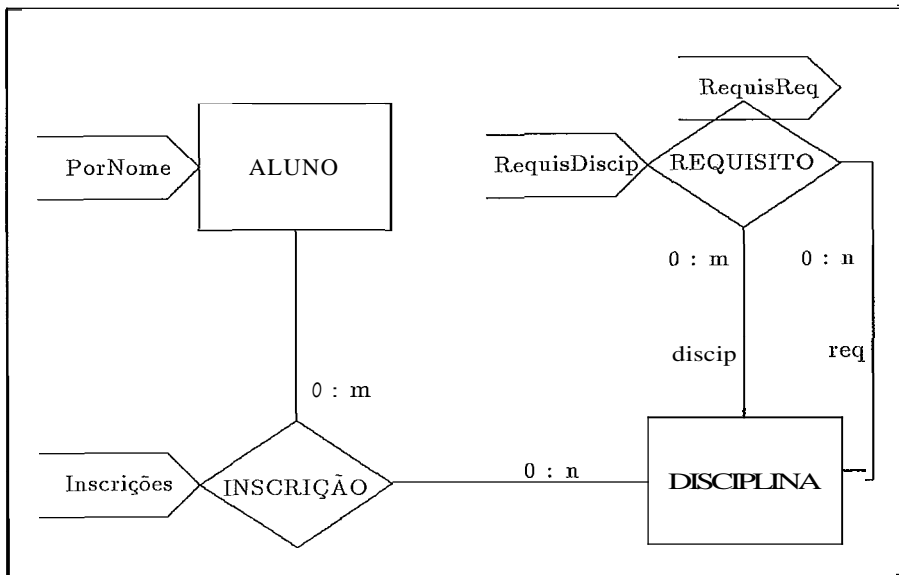


Figura 7. Esquema exemplo

Entidade : ALUNO

Atributos : REGISTRO , ANOME , CPF

Chaves : PORREGISTRO

: PORCPF

Ordens : PORNOME (onde o determinante da ordem é ANOME)

Entidade : DISCIPLINA

Atributos : CÓDIGO , DNOOME

Chaves : PORCÓDIGO

Ordens : nenhuma

Relacionamento : REQUISITO

Atributos : nenhum

Papéis : *discip* e *req*

Chaves : nenhuma

Ordens : REQUISDISCIP (onde o determinante da ordem é CÓDIGO de DISCIPLINA e a origem da ordem é o papel *discip*)

: REQUISREQ (onde o determinante da ordem é CÓDIGO de DISCIPLINA e a origem da ordem é o papel *req*)

Relacionamento : INSCRIÇÃO

Atributos : PERÍODO

Chaves : nenhuma

Ordens : INSCRIÇÕES (onde o determinante da ordem é CÓDIGO de DISCIPLINA e a origem da ordem é o papel que vem de ALUNO)

VI.2 - O que é o Servidor ?

Para que a aplicação possa comunicar-se com o Banco de dados Objeto, é preciso que haja uma linguagem que traduza as solicitações do usuário em acesso propriamente dito ao BD. Para isso seria interessante desenvolver uma linguagem simples que se utilizasse de bibliotecas de funções previamente definidas. Essas bibliotecas baseiam-se na descrição da base de dados conforme especificado através do Diagramador E-R e do Dicionário de Dados. Tanto o servidor para sistemas de grande porte que exige uma ligação entre este e o microcomputador quanto o servidor para sistemas de pequeno porte utilizam-se das bibliotecas de funções para poderem manipular o BD. No entanto, ao tratar-se de BD local, construiu-se um utilitário que, a partir do esquema E-R especificado, ao ser executado, gera um servidor de banco de dados contendo as referidas bibliotecas [8]. Esse servidor, desenvolvido para o Turbo Pascal 5.5 [31], possui interface orientada a objetos através dos métodos e manipula um banco de dados local autônomo. Os objetos, atributos e métodos são gerados de modo que os usuários possam ter acesso ao banco com facilidade e, ao mesmo tempo, contar com todas as facilidades de seus ambientes preferidos de programação. As bibliotecas nele geradas para a interface com o Turbo Pascal 5.5 permitem que o usuário continue usando seu ambiente de desenvolvimento na mesma linguagem. As aplicações geradas e também a interface com sistemas especialistas usam esta facilidade gerando programas Pascal que acessam o banco via métodos.

Para que possamos compreender melhor, vamos considerar a Figura 8. Nela temos que o módulo de *Programa Pascal* deverá utilizar-se do *Servidor para Bancos*

de *Dados Orientados a Objetos* [26]. Esse *Servidor* é o nome dado ao conjunto de métodos que, em tempo de compilação, servirá para acessar o BD Local ou remoto. Este último, podendo ser desenvolvido em DMSII da Unisys tornando necessário um protocolo de comunicação entre o microcomputador e o equipamento de grande porte. Já no primeiro caso, o acesso pode ser feito diretamente.

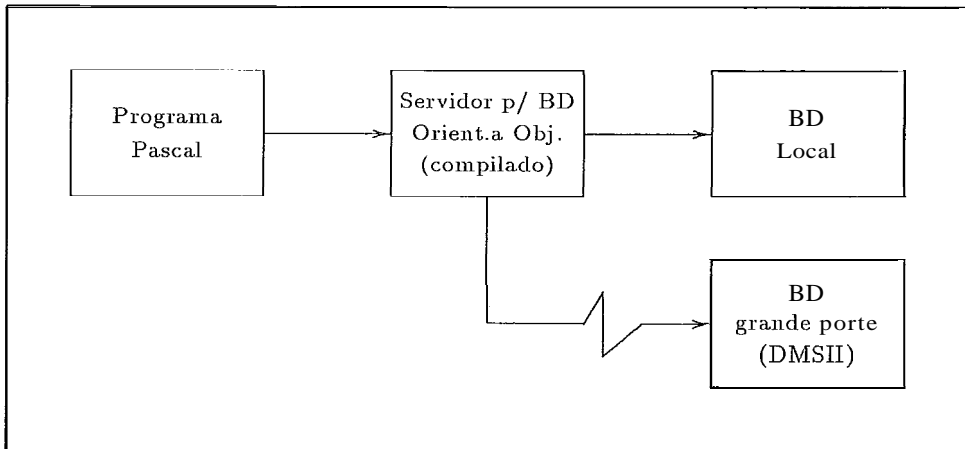


Figura 8. Esquema para Programas de Aplicação

Entretanto, como já mencionamos, o que será tratado neste trabalho é o módulo de Interface de Consultas que não consiste de uma aplicação gerada, mas sim de uma aplicação interativa. Nessa situação os métodos contidos no *Servidor para BD's Orientados a Objetos* serão interpretados como mostra a Figura 9.

VI.3 - Como funciona o Servidor ?

Como já vimos, o acesso à base de dados é realizado através de um conjunto de métodos que aplicam-se às classes de objetos que formam a base. Este conjunto é bastante simples, o que torna seu uso através da programação fácil e intuitivo.

Para cada uma das quatro classes de objetos existentes em um esquema E-R são gerados métodos a elas associados. As classes são:

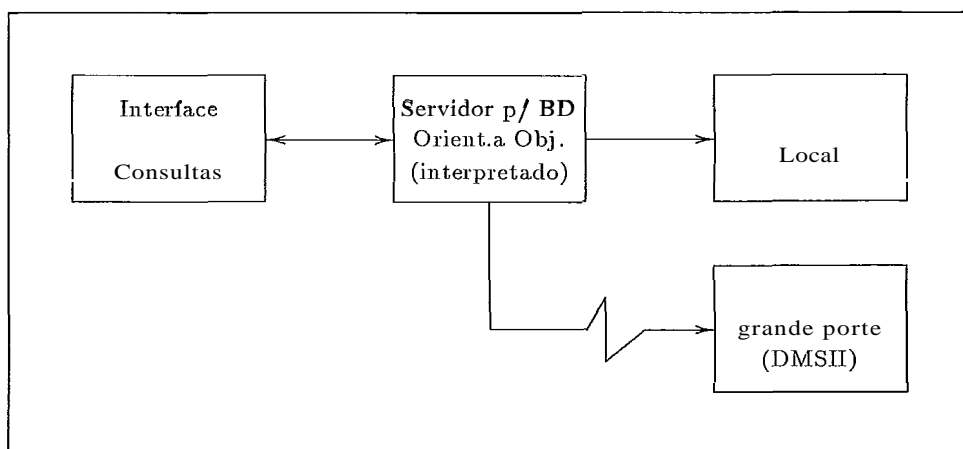


Figura 9. Esquema para Interfaces de Consultas

- banco de dados, chamado de BANCO pelo Servidor;
- conjunto, chamado de ENTIDADE pelo Servidor;
- elemento, chamado por este mesmo nome no Servidor; e
- ordem.

Antes de passarmos à descrição dos métodos associados a essas classes, precisamos fazer algumas observações sobre o conteúdo do Servidor a fim de melhor esclarecermos o assunto.

- a) Os métodos são gerados para essas classes de objetos, para cada um dos elementos especificados no esquema E-R. Envolvem entidades, relacionamentos e ordens;
- b) Para o Servidor, as entidades e relacionamentos são tratados igualmente, ou seja, são arquivos de dados que podem ter atributos e ordens associadas. Para as entidades e relacionamentos do esquema são gerados os mesmos métodos;
- c) As chaves são consideradas como sendo também ordens. A única diferença entre

ambas está no fato da primeira ser especificada como ordem que não contém duplicação de chaves;

- c) Todas as entidades e relacionamentos do esquema possuirão arquivos de dados do tipo *dat*. Aqueles que tiverem chaves e/ou ordens também definidas, terão um arquivo de índices do tipo *ind* para cada uma delas;
- e) Cada entidade ou relacionamento terá, além de suas ordens e chaves definidas, uma ordem automática através da qual poder-se-á ler seu arquivo de dados na ordem sequencial física dos registros. Esta ordem dará origem também a um arquivo do tipo *ind*;
- f) As classes básicas de objetos mostradas possuem tipos do Pascal cujos nomes, respectivamente, são T-BANCO, T-ENTIDADE, T-ELEMENTO e T-ORDEM, válidos para quaisquer esquemas. Estes tipos são de objetos, ou seja,

T_BANCO = object

T_ENTIDADE = object

T_ELEMENTO = object

T_ORDEM = object

- g) Cada uma dessas classes pode ter atributos, *funções* e *procedimentos* (estas duas últimas do Pascal) associados a elas. Estas informações serão vistas mais adiante;
- h) Cada uma dessas classes de objetos pode dar origem a subclasses de objetos, dependendo do esquema E-R especificado. Temos, por exemplo, que ALUNO é uma subclasse de ENTIDADE, ou seja,

T_ENTIDADE_ALUNO = object (T_ENTIDADE)

onde T_ENTIDADE_ALUNO herda todos os atributos, *funções* e *procedimentos* da classe-mãe (T_ENTIDADE), além das ordens, *funções* e *procedimentos* que ele mesmo possui;

- i) Muitos métodos são executados chamando, também, outros métodos para serem executados. Esses últimos podem, também, chamar outros métodos e assim por diante;
- j) O acesso ao banco de dados é feito através do Turbo Pascal Database Toolbox [33] cujos comandos utilizados são `Taccess.OpenIndex`, `MakeIndex`, `CloseIndex`, `GetRec`, `ClearKey`, `SearchKey`, `PrevKey`, `NextKey`, `FileLen`, `FindKey`, `AddKey`, `DeleteKey`, `OpenFile`, `CloseFile`, `MakeFile`, `PutRec`, `AddRec` e `DeleteRec`;
- k) Cada instância de um arquivo possui um *surrogate* único que serve de referência lógica. Nas instâncias dos arquivos gerados a partir de um relacionamento, além de seu *surrogate* único, há ainda o *surrogate* de cada uma das entidades envolvidas naquele relacionamento. Também as ordens têm suas instâncias ordenadas a partir dos *surrogates* das instâncias originais.

VI.4 - Os Métodos

Vamos, então, descrever, mais detalhadamente, as classes de objetos mostradas com os métodos a elas associadas.

A classe *banco* de dados, do tipo `TBANCO`, é responsável pela identificação da base de dados como um todo. Ela possui três métodos que referem-se a todas as estruturas físicas do BD, ou seja, entidades, relacionamentos e ordens. Suas descrições são as seguintes:

OPENALL

Este método abre acesso a todos os arquivos gerados no banco a partir das entidades, relacionamentos e ordens especificadas no esquema E-R. Sua execução se dá através da chamada de outro método `OPEN`, definido para entidades, rela-

cionamentos e ordens. Para o exemplo da Figura 7, teríamos:

ALUNO.open;

ALUNO.PORREGISTRO.open;

ALUNO.PORCPF.open;

ALUNO.ALUFISICA.open; (ordem fisica)

ALUNO.PORNOME.open;

INSCRICAO.open;

INSCRICAO.INSFISICA.open;

INSCRICAO.INSCRICOES.open;

DISCIPLINA.open;

DISCIPLINA.PORCODIGO.open;

DISCIPLINA.DISFISICA.open;

REQUISITO.open;

REQUISITO.REQUISDISCIP.open;

REQUISITO.REQUISREQ.open;

REQUISITO.REQFISICA.open;

CLOSEALL

Este método procede de modo inverso ao anterior, embora possua as mesmas características. Ele é encarregado de encerrar o acesso aos mesmos arquivos

abertos no método OPENALL, com exceção das ordens físicas:

ALUNO.close;

ALUNO.PORREGISTRO.close;

ALUNO.PORCPF.close;

ALUNO.PORNOME.close;

INSCRICAO.close;

INSCRICAO.INSCRICOES.close;

DISCIPLINA.close;

DISCIPLINA.PORCODIGO.close;

REQUISITO.close;

REQUISITO.REQUISDISCIP.close;

REQUISITO.REQUISREQ.close;

CREATEALL

Este método é responsável por criar toda a base de dados, ou seja, ele gera, através da chamada de outro CREATE, definido para entidades, relacionamentos

e ordens, as estruturas do Banco com instâncias vazias. Teríamos assim:

ALUNO.create;

ALUNO.PORREGISTRO.create;

ALUNO.PORCPF.create;

ALUNO.ALUFISICA.create;

ALUNO.PORNOME.create;

INSCRICAO.create;

INSCRICAO.INSFISICA.create;

INSCRICAO.INSCRICOES.create;

DISCIPLINA.create;

DISCIPLINA.PORCODIGO.create;

DISCIPLINA.DISFISICA.create;

REQUISITO.create;

REQUISITO.REQUISDISCIP.create;

REQUISITO.REQUISREQ.create;

REQUISITO.REQFISICA.create;

Por sua vez, a classe conjunto é tratada no servidor como ENTIDADE. Os métodos associados a essa classe são gerados para cada entidade e relacionamento definido no esquema E-R. Estes métodos têm como objetivo a manipulação interna dos arquivos de dados. São eles:

CARDINALIDADE

Este método devolve o número de instâncias contidas no arquivo para o qual o método foi invocado;

OPENDATA

Este método é responsável por abrir o acesso ao arquivo de dados, ou seja, tipo *dat*, para o qual o método foi invocado;

CREATEDATA

Ao ser chamado de uma entidade ou relacionamento, este método gera a estrutura do BD correspondente com instâncias vazias;

CLOSE

Este método encerra o acesso ao arquivo correspondente à entidade ou relacionamento que o chamou. Ex: Seja o método `T_ENTIDADE.CLOSE`. Para a entidade `ALUNO`, como vimos anteriormente, a chamada a esse método seria `X.CLOSE`, onde `X` é uma variável do tipo `T_ENTIDADE_ALUNO`;

GETREC, PUTREC, ADDREC e DELETEREC

Estes métodos são responsáveis, respectivamente, pela leitura, alteração, inclusão e remoção de instâncias a partir de um *surrogate*.

Obs:

- 1) Em geral, esses métodos não são chamados diretamente pelo programa de aplicação, mas sim, a partir de outros métodos;
- 2) Além desses métodos, a classe conjunto possui, também, atributos. Tanto os métodos quanto os atributos serão herdados por todas as subclasses da classe descrita por `T_ENTIDADE`.

Ainda sobre essa classe, como já mencionamos, ela possui subclasses relativas às entidades e relacionamentos definidos. As subclasses possuem métodos e, como

atributos, as ordens e chaves definidas no esquema E-R que são de um tipo que é subclasse do tipo `T_ORDEM`. Esta última subclasse será vista mais adiante quando definirmos a classe `ordem`. Por exemplo, temos:

```
T_ENTIDADE_ALUNO = object ( T_ENTIDADE )
    PORREGISTRO      : T_ORDEM_ALUNO_1;
    PORCPF           : T_ORDEM_ALUNO_2;
    ALUFISICA       : T_ORDEM_ALUNO_DEFAULT;
    PORNOME         : T_ORDEM_ALUNO_3;
```

```
procedure OPEN;
```

```
procedure CREATE;
```

```
function PONTEIRO ( surrogate : SurroType ) : PtrALUNOType;
```

onde os métodos `OPEN` e `CREATE` são chamados, respectivamente, como `X.OPEN` e `X.CREATE` onde `X` é uma variável do tipo `T_ENTIDADE_ALUNO` que, por sua vez, executam, respectivamente, os métodos `OPENDATA` e `CREATEDATA`. Já o método `PONTEIRO`, dado um *surrogate*, instancia a entidade `ALUNO` através da chamada a outro método e devolve o endereço desta instância.

Já a classe *elemento* refere-se às instâncias propriamente ditas das entidades e relacionamentos. Seu tipo é `T_ELEMENTO` e possui o atributo *surrogate* que, junto com o método que ela possui, é herdado por suas subclasses. O único método existente neste tipo é:

REFERÊNCIA

Este método é executado somente através da chamada de outros métodos, ou seja, é transparente ao programador. Dado um *surrogate*, ele devolve uma referência física da instância.

Esta classe possui, também, subclasses relativas às entidades e relacionamen-

tos definidos. As subclasses de ELEMENTO são do tipo T_ALUNO, T_INSCRICAO, T_DISCIPLINA e T_REQUISITO. Cada uma delas possui atributos e métodos associados às entidades e relacionamentos que as representam. Os atributos, na verdade, englobam mais do que somente atributos definidos no esquema como REGISTRO, ANOME e CPF para o tipo T_ALUNO. Dependendo da representação no diagrama E-R, outros tipos de atributos podem ocorrer. Vamos exemplificar com base na Figura 7:

- a) Em T_ALUNO ocorre o atributo `Count_ALUNO_INSCRIÇÃO` devido à existência da ordem INSCRIÇÕES em INSCRIÇÃO que tem origem em ALUNO. Este contador guarda o número de inscrições em disciplinas que estão associadas a cada instância de ALUNO;
- b) Em T_DISCIPLINA ocorrem três atributos: `Count_DISCIPLINA_INSCRIÇÃO`, `Count_DISCIP-REQUISITO` e `Count_REQ-REQUISITO`. O primeiro contador deve-se à existência da ordem INSCRIÇÕES em INSCRIÇÃO que tem como determinante o atributo CÓDIGO da entidade DISCIPLINA. Ele guarda o número de alunos inscritos em cada disciplina. Os outros dois contadores devem-se à existência, respectivamente, das ordens REQUISDISCIP e REQUISREQ. Cada um deles está associado a um papel definido no diagrama, respectivamente, *discip* e *req*. Sendo assim, o contador `Count_DISCIP-REQUISITO` guarda o número de requisitos que cada disciplina possui, enquanto que `Count_REQ-REQUISITO` guarda o número de disciplinas que têm como requisito a disciplina da instância atual;
- c) Os relacionamentos, no entanto, possuem, como atributos, os papéis que os unem a entidades, sendo do tipo *surrogate*. Em T_INSCRICAO são definidos os *surrogates* de nomes ALUNO e DISCIPLINA. Esses nomes correspondem às entidades porque seus respectivos papéis não foram denominados. Em T_REQUISITO são definidos DISCIP e REQ como *surrogates*. Como mencionamos anteriormente, eles contêm os *surrogates* das entidades ligadas aos relacionamentos. Os *surrogates* de ALUNO, DISCIPLINA, INSCRIÇÃO e REQUISITO estão contidos no atributo *Surrogate* que

eles herdam de T_ELEMENTO.

Alguns métodos são gerados, igualmente, para todas essas subclasses. São eles:

STORE, UPDATE e DELETE

São responsáveis, respectivamente, pela inclusão de instâncias no arquivo, atualização e remoção;

INSTANCIA

É responsável por manter alocada uma instância lida para que seja possível sua manipulação.

Estes *procedimentos* são chamadas por X.INSTANCIA, X.STORE, etc, sendo X uma variável do tipo T_ALUNO.

Há, também, métodos variados, dependendo da representação existente no diagrama E-R. São eles:

- a) Em T_ALUNO há uma *function* chamada INSCRIÇÕES, que é o nome da ordem de INSCRIÇÃO que tem ALUNO como origem. Esta orclein é chamada parcial pois somente acessa as instâncias de INSCRIÇÃO que estão associadas a uma determinada instância de ALUNO. Isto é, para cada aluno X, é como se *inscrições* (X) fosse um atributo que retorna uma lista de instâncias de INSCRIÇÃO associadas aquele aluno [14]. Ela devolve o endereço de um objeto do tipo orclein, no caso INSCRIÇÕES, para o qual a instância atual de ALUNO está apontando. É chamado por X.INSCRIÇÕES, onde X é uma variável do tipo T_ALUNO;
- b) Em T_DISCIPLINA há duas *funções* chamadas REQUISDISCIP e REQUISREQ que devem-se ao fato das ordens correspondentes terem origem nos papéis, respectivamente, *discip* e *req*. A primeira devolve o endereço da orclein REQUISDISCIP para a qual a instância atual de DISCIPLINA está apontando. A segunda faz o mesmo

para REQUISREQ.

Obs: Existe uma outra representação no E-R que levaria à geração de outro método que não consta neste exemplo. Lembramos, então, a Figura 3 contendo um exemplo de Generalização. Consideremos que um servidor tenha sido gerado para este caso. Haveria, então, os tipos T_PESSOA, T_ALUNO e T_PROFESSOR como subtipos de T_ELEMENTO. Vejamos os seguintes aspectos:

- . Os atributos de PESSOA são CPF e NOME;
- . Os atributos de ALUNO são REGISTRO e CURSO;
- . Os atributos de PROFESSOR são COD-DOCENTE e DEPARTAMENTO;
- . Os atributos de PESSOA são herdados por ALUNO e PROFESSOR;
- . Sob o tipo T_PESSOA não haverá os métodos STORE, UPDATE e DELETE. Eles serão aplicados somente nos subtipos;
- . Dois novos métodos passariam a existir sob T_PESSOA: IS_ALUNO e IS_PROFESSOR. Dada a instância atual de PESSOA, o primeiro método verifica se ela diz respeito a ALUNO e o segundo, a PROFESSOR.

Finalmente, a classe ordem, do tipo T_ORDEM, possui uma série de atributos e métodos herdados pelos seus subtipos. É a partir desta classe que o acesso aos arquivos é feito, para posterior instanciação e manipulação dos dados. Pode haver várias ordens definidas sobre as entidades e relacionamentos. Como já mencionamos, as ordens podem ser totais ou parciais. A ordem é total quando todos os elementos do conjunto de entidades ou do relacionamento que é objeto da ordem podem ser alcançados através dela. A ordem POR_NOME de ALUNO, por exemplo, abrange todas as ocorrências de ALUNO. Por sua vez, a ordem é parcial quando acessa somente uma lista de instâncias associadas a uma instância de origem. É o caso da ordem INSCRIÇÕES de ALUNO. Através dela, somente as instâncias de INSCRIÇÃO associadas

à instância de um dado ALUNO são alcançadas [14]. Alguns dos atributos que a ordein possui são os *surrogates* das instâncias anterior e posterior na ordein especificada e o sentido de navegação, ascendente ou descendente, que a busca está seguindo. Os métodos definidos para a classe ordem são:

CREATEIND

Este método assemelha-se ao anterior, sendo responsável por gerar estruturas de índices com instâncias vazias;

OPENIND

Este método não é chamado diretamente pela aplicação. Sua execução parte da chamada de outro método. O OPENIND é responsável por abrir arquivos de índices;

RESET

É responsável por posicionar o ponteiro no início da ordem, preparando-a para que seja lida;

EOO

Este método testa o fim da lista de instâncias se a navegação através da ordem for ascendente. Caso contrário, testa seu início;

NEXT

Este método obtém a próxima instância da ordem. É utilizado para navegação tanto no sentido ascendente quanto descendente;

SEARCH

Este método promove a busca de uma instância dentro da ordein, mediante uma

chave. A instância a ser apontada será aquela que atende exatamente à chave ou, caso esta não exista, a imediatamente posterior;

FIND

Executa a busca de uma instância que atenda exatamente à chave;

DELETEKEY

Este método é chamado por outros métodos e é responsável por remover a associação entre uma instância e uma determinada chave;

ADDKEY

Este método inclui a associação;

CHANGEKEY

Este método muda a associação;

CLOSE

Fecha os arquivos de índices para acesso.

Alguns subtipos são gerados na classe ordem. Estes subtipos dizem respeito as ordens definidas no diagrama E-R. Por exemplo, em ALUNO da Figura 7, há uma ordem e duas chaves especificadas que dão origem a três ordens no Servidor, sendo cada uma delas um subtipo distinto. Uma outra ordem, automática, é também gerada como outro subtipo que é a ordem física do arquivo de dados. Cada um desses subtipos, além de herdar as características de T-ORDEM, tem os métodos OPEN e CREATE definidos. O primeiro chama o método OPENIND indicado anteriormente e o segundo chama CREATEIND.

A partir do desenvolvimento da Interface de Consultas, surgiu a necessidade de ampliarmos os métodos já definidos. Dois deles encontram-se já implantados no Servidor através de seu Gerador. Ambos encontram-se especificados sob a classe ordem. São eles:

RESETBACK

É responsável por posicionar o ponteiro no fim da ordem;

INVERTESENTIDO

O sentido inicial da ordem é dado pela execução dos métodos RESET e RESETBACK. Caso desejemos inverter este sentido, dinamicamente, devemos chamar o método INVERTESENTIDO.

Os métodos anteriormente descritos foram desenvolvidos para que fossem compilados juntamente com suas aplicações. No entanto, visando implementar interfaces gráficas para consultas a bancos de dados utilizando os métodos, necessitamos de uma forma que tornasse os métodos genéricos o bastante para que não precisássemos mencionar, na aplicação, os nomes das estruturas particularizadas do BD como ALUNO, T_DISCIPLINA, ALUNO.PORNOME.open, etc. A partir dessa necessidade, passamos o servidor por um refinamento onde incluímos três novos métodos interpretáveis que são descritos da seguinte forma:

- a) Método ALOCA - não faz parte de nenhum objeto e seu objetivo é alocar uma área na memória para a estrutura da entidade ou relacionamento. Ao ser chamado, o método devolve o endereço desta área. Ele permite que o nome da estrutura a ser acessada não precise ser mencionada na aplicação, ou seja, *aluno*.INSTANCIA, onde *aluno* é uma variável do tipo T_ALUNO poderia ser substituído por *X*.INSTANCIA onde *X* é do tipo T_ELEMENTO;
- b) Método COPIA - este método é definido sob a classe ordem, ou seja, aplica-se a

todas as ordens definidas no Servidor. É responsável pela obtenção do nome da ordem que o chamou. O motivo de sua existência é, basicamente, o mesmo do anterior: evitar a necessidade de mencionar nomes na aplicação;

- c) Método ATRIBSTR, ATRIBREAL, ATRIBINT, ATRIBLINT, ATRIBORD - estes métodos são definidos sob a classe elemento e têm como objetivo devolver o conteúdo do atributo que lhe é fornecido como parâmetro. Eles são em número de cinco, pois os atributos que estão contidos em um subtipo de ELEMENTO (T_ALUNO, T_REQUISITO, etc) podem ser *cadeias de caracteres*, *reais*, *inteiros*, *longints* ou *ponteiros* para ordens (INSCRIÇÕES).

VI.5 - Conclusão

Neste capítulo, como introdução ao objeto deste trabalho, ou seja, a Interface de Consultas, tratamos o Servidor de Banco de Dados que é gerado a partir de um módulo que faz parte da ferramenta QUICK-DB: o Gerador do Servidor de Bancos de Dados. O Servidor contém um conjunto de métodos que está relacionado com o esquema de BD definido a partir de sua especificação sobre um diagrama E-R. O esquema é dividido em quatro classes que são banco de dados, conjunto, elemento e ordem. Cada uma dessas classes foi descrita incluindo seus métodos e suas subclasses. Estas herdam as características da classe-mãe. Os métodos foram, também, descritos. O papel dos métodos é possibilitar a comunicação entre as aplicações e o banco de dados, podendo este ser local ou remoto. No primeiro caso, o banco de dados encontra-se na própria estação de trabalho e é manipulado diretamente pelo usuário através de um programa de aplicações que pode ser gerado ou por meio de uma aplicação interativa. Já no segundo caso, o banco de dados encontra-se em um ambiente de grande porte. A aplicação poderá manipular os dados na base através de uma ligação entre o microcomputador e o de grande porte. Nesse caso o micro será usado como interface com o usuário, captando informações do equipamento de grande porte.

Através do uso dos métodos, cria-se uma maior facilidade na programação das aplicações. Essas aplicações têm os métodos nela embutidos e juntamente com eles serão compiladas. Alguns refinamentos a esses métodos foram desenvolvidos neste trabalho visando atender ao módulo de Interface Gráfica para Consultas. Outros métodos, entretanto, foram propostos aqui para que, interpretáveis, possam trazer caráter genérico às aplicações como é o caso da Interface mencionada. Tal generalidade evitará que nomes das estruturas, ordens, atributos sejam citados dentro das aplicações.

CAPÍTULO VII

INTERFACE GRÁFICA DE CONSULTAS

VII.1 - Introdução

Nos capítulos anteriores descrevemos os assuntos mais relevantes à evolução deste trabalho, partindo do que hoje existe em termos de formas de acesso a BD, a propostas de aperfeiçoamento através da adoção de modelos semânticos como o MER. Mostrou-se, ainda, uma descrição do ambiente no qual se insere este trabalho e os paradigmas nele adotados.

Agora, no entanto, iremos, de fato, definir o Módulo de Consultas do QUICK-DB, seus objetivos, filosofia e aplicação a fim de que possamos, posteriormente, descrever suas características de implementação. Para isso iremos considerar o esquema exemplo retratado na Figura 7.

VII.2 - Objetivos

Como já vimos anteriormente, há uma grande preocupação atualmente em se desenvolver sistemas mais facilmente utilizáveis pelos usuários de um modo geral. O uso do computador está muito difundido nos diversos setores da sociedade e aqueles que o operam não são, necessariamente, pessoas apropriadamente treinadas para tal. São usuários administrativos, pesquisadores das mais diversas áreas de conhecimento, professores, enfim, pessoas que desejam utilizar o computador não como fim, mas como meio. Nesse panorama, dentro da ferramenta QUICK-DB, surge o módulo direcionado a Interfaces de Consultas.

O objetivo deste módulo é permitir que qualquer usuário, mesmo aquele casual, possa efetuar consultas *ad hoc* a um banco de dados sem nenhum conhecimento de

linguagem de computação, da estrutura dessa base de dados e demais aspectos físicos. Para tanto, o usuário deverá ser conduzido à formulação das consultas.

As consultas deverão ser efetuadas sobre um esquema E-R que representa a base de dados. Isto implica em que o usuário terá como visão simplesmente o diagrama E-R e seus componentes. A programação da interface gráfica foi desenvolvida utilizando as ferramentas gráficas do Turbo Pascal 5.5 [30].

VII.3 - Abrangência

Para que o usuário possa não se preocupar com as características físicas de implementação da base de dados, o QUICK-DB foi desenvolvido visando possibilitar um acesso facilitado ao banco de dados com utilização de interface gráfica amigável.

São duas as formas de acesso ao BD definidas através da interface gráfica do módulo de consultas.

A primeira delas é aquela em que o usuário deve determinar a consulta desejada estabelecendo uma série de elementos necessários à sua formulação. Alguns desses elementos são as informações que deverão ser mostradas como resultado e as condições de seleção que deverão ser adotadas. Grande parte da especificação da consulta deverá ser feita com o auxílio do próprio esquema através do uso de menus, evitando assim que o usuário precise conhecer e escrever muitos detalhes. Desse modo, apenas aquilo que a ferramenta não puder fornecer ao usuário deverá ser especificado por ele como por exemplo algumas condições de seleção. Para isso, uma espécie de linguagem geral, simplificada, de consultas deverá ser definida. Os termos dessa linguagem serão convertidos em parâmetros e passados ao Servidor de Banco de Dados.

A segunda forma de acesso é definida através do chamado *folheamento*. Por meio do *folheamento* o usuário poderá efetuar consultas não tão complexas ao BD. Nele o usuário será orientado por meio de menus necessitando especificar somente

os valores a que as chaves deverão atender. As demais informações serão extraídas da definição do banco de dados representada no esquema E-R à medida em que o usuário selecionar seus elementos através da manipulação de *mouse*. Este tipo de consulta permite que o usuário tenha acesso às instâncias das estruturas definidas no esquema E-R, individualmente.

Com relação à primeira forma de acesso mencionada, consideramos que, devido ao seu caráter complexo, seria muito difícil e complicado chegarmos a uma forma geral de consultas. Isto porque o objetivo seria o de permitir ao usuário que este formule consultas ditas completas ao BD. Devido ao elevado grau de dificuldade dessa forma de acesso, optamos, neste trabalho, pela segunda forma, o *folheamento*.

As consultas completas serão avaliadas em uma outra fase do trabalho, sendo um objetivo futuro.

VII.4 - Definição das Características

Como mencionamos, o *folheamento* permitirá ao usuário ter acesso às informações contidas, individualmente, em cada instância das estruturas criadas a partir de um esquema E-R. O acesso a essas instâncias far-se-á através do uso das ordens ou chaves.

As ordens estabelecem uma forma ordenada de acesso às instâncias de uma entidade ou relacionamento onde há um conjunto de atributos que determinam esta ordenação. Na Figura 7, *NOME* de *ALUNO* é um atributo que determina a ordem *POR NOME* fazendo com que enxerguemos a entidade *ALUNO* ordenada alfabeticamente por aquele atributo.

Por sua vez, as chaves assemelham-se bastante às ordens. As instâncias serão vistas também ordenadas pelos atributos que determinam a ordenação orientada pela chave.

As diferenças fundamentais existentes entre essas formas de acesso são duas. A primeira deve-se ao fato de não poder haver duplicação de chave e a segunda, ao fato do acesso por chaves permitir que, através de uma condição inicial, seja localizada uma determinada instância. Isto implica em que a primeira instância acessada através da chave é aquela que atende à condição enquanto que na ordem, é a primeira ou a última instância da estrutura solicitada. É importante lembrar que ambas, tanto chave quanto ordem, esta última quando for total, estabelecem uma ordenação de todas as instâncias existentes em cada uma das entidades ou relacionamentos conforme tenham sido previamente definidas durante modelagem do esquema E-R.

Tendo considerado a ordenação das instâncias, descreveremos, então, as funções que serão oferecidas pelo módulo de consultas do QUICK-DB através do *folheamento*:

- *Consulta às características dos elementos do esquema*

Esta função tem como objetivo permitir que o usuário conheça mais detalhadamente os elementos que compõem o diagrama como as entidades e relacionamentos, mostrando seus atributos, ordens e chaves com suas respectivas especificações;

- *Consulta às instâncias de uma dada entidade ou relacionamento*

O objetivo desta função é permitir que o usuário tenha acesso ao conteúdo das instâncias das estruturas representadas pelas entidades e relacionamentos. A pesquisa às instâncias é feita através das formas de ordenação que foram especificadas no esquema E-R. Pode ocorrer tanto por meio de chaves quanto por meio de ordens. Quando o acesso for feito utilizando uma chave, o usuário deverá determinar seu valor para que a instância desejada seja localizada. No caso do acesso ser feito por meio de ordens, o usuário deverá somente informar se deseja a primeira ou a última instância daquela ordem. Esta função, especificamente, na forma com que está sendo definida, envolve somente uma entidade ou rela-

cionamento por vez. Um exemplo que pode ser citado, sempre sobre a Figura 7 é

Obter alunos em ordem de nome

que faria acesso à entidade ALUNO através da ordem definida PORNOME. Conforme foi definida a função, caso o usuário opte pela primeira instância da ordem PORNOME, o primeiro aluno será localizado e apontado, como por exemplo o aluno *Abel*. Podemos notar que o único elemento envolvido nesta consulta é a entidade ALUNO;

- *Navegação dentro de uma dada entidade ou relacionamento*

Através desta função o usuário poderá navegar pelas instâncias de uma determinada estrutura (entidade ou relacionamento), a partir da função anterior, seguindo aquela ordenação, isto é, via a ordem ou a chave que fôra especificada. Por exemplo, o usuário poderá navegar pelas instâncias de ALUNO em ordem de nome através de PORNOME ou em ordem de registro através da chave PORREGISTRO. Ainda considerando o exemplo anterior, o usuário poderá navegar para o próximo aluno desta ordem fazendo com que fosse localizado e apontado, por exemplo, o aluno *Antônio*. Veremos esta função com maiores detalhes mais adiante;

- *Navegação por entre os elementos do esquema*

Além de consultar isoladamente cada um dos elementos do esquema, o *folheamento* do módulo de consultas permite que o usuário navegue por entre as entidades e relacionamentos. Esta navegação é feita ao longo de um caminho que deve ser estabelecido por ele. Deste modo, a consulta poderá partir de uma entidade, passar por um relacionamento, dirigir-se a outra entidade e assim por diante. Vale lembrar que uma navegação entre elementos de um esquema E-R deve partir, obrigatoriamente, de uma entidade em direção a um relacionamento e, por sua vez, em direção a uma entidade, etc. A possibilidade ou não de se

processar a navegação depende da existência ou não de ordens que definam o caminho que deverá ser seguido e da relação existente entre os dois elementos que estão envolvidos no caminho. Quando a relação for $n : 1$ é sempre possível navegar. Se a relação for $1 : n$ é preciso que haja a ordem. Podemos exemplificar esta função com o auxílio da seguinte consulta:

*Obtei- as disciplinas nas quais o aluno
de registro 871718200 encontra-se inscrito*

Observe que seria necessário apontar para a entidade ALUNO através de sua chave PORREGISTRO, caminhar para o relacionamento INSCRIÇÃO via a ordem INSCRIÇÕES e, então, seguir até a entidade DISCIPLINA para obter as disciplinas. Em alguns casos não bastará indicar somente os elementos que serão percorridos, no caso ALUNO-INSCRIÇÃO-DISCIPLINA. A situação mais tradicional é o auto-relacionamento, ou seja, dada a consulta

*Obter as disciplinas que constituem
requisitos da disciplina MAB112*

não bastará traçar o caminho DISCIPLINA-REQUISITO-DISCIPLINA. Será necessário, também, estabelecer qual o caminho a ser seguido, isto é, por que “papel” passará. No caso, DISCIPLINA-discip-REQUISITO-req-DISCIPLINA.

Dadas as funções a que atende o módulo de consultas, passaremos, então, a descrever como comporta-se o *folheamento*.

VII.5 - O Folheamento

O QUICK-DB utiliza-se de *mouse* para operar sobre o esquema E-R, ou seja, sempre que for preciso escolher um elemento do diagrama ou função, o usuário deverá clicar sobre ele. O primeiro botão do *mouse* é utilizado para clicarmos aqueles elementos ou funções. O segundo botão funciona como < Escape > para desistir

dos menus que não possuam a opção RETORNO como veremos adiante. Por sua vez, o terceiro botão quando pressionado mostra a função de *Auxílio* do *software* que desenha o diagrama (Diagramador E-R).

Ao ser clicado um elemento do esquema E-R, ele é marcado na tela, diferenciando-o dos demais. Estará marcado sempre o último elemento ou caminho selecionado.

O que faremos aqui nesta parte do capítulo é, através de exemplos de consultas, mostrar as características e funcionamento do módulo gráfico de Interface de Consultas. As consultas dadas como exemplo começam da mais simples, incrementando seu grau de dificuldade, até as mais complexas, com o objetivo de fazer com que a filosofia desta ferramenta seja lentamente absorvida por todos.

De início, o diagrama E-R desejado deverá ser carregado para a tela. Seja este esquema como aquele descrito na Figura 7. Vamos, então, iniciar a descrição do módulo de consultas a partir de:

Obter alunos em ordem de nome

Esta consulta deverá mostrar os dados de cada aluno contido na entidade ALUNO. Suas instâncias deverão ser obtidas a partir de uma leitura sobre essa entidade segundo a ordem alfabética de nome de aluno, PORNOME.

Os passos a serem seguidos para obtenção do resultado são:

1) Clicar sobre a entidade ALUNO

Dado que esta entidade não foi ainda instanciada, o menu da Figura 10 será mostrado através de uma janela aberta na mesma tela.

1.a) A opção INSTANCIAR leva o usuário a especificar a forma com que este deseja chegar a uma instância do elemento clicado;

1.b) A opção ESQUEMA leva ao conhecimento do usuário os detalhes da especi-

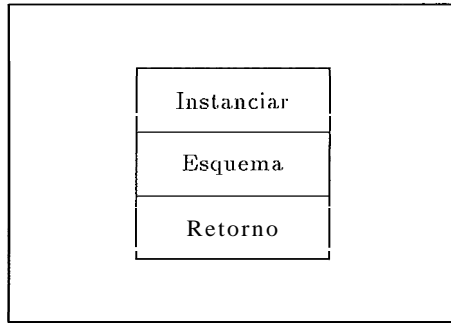


Figura 10. Menu para Instanciação

ficação da entidade ALUNO;

- 1.c) A opção RETORNO faz com que desapareça este menu voltando à situação anterior, ou seja, somente o diagrama E-R na tela ou passo 1.

Obs.: A opção RETORNO presente nos menus, quando escolhida, faz com que se retorne ao menu ou estado anterior da tela.

- 2) Clicar sobre a opção INSTANCIAR

Uma vez clicada esta opção, uma outra janela será aberta mostrando o próximo menu conforme a Figura 11.

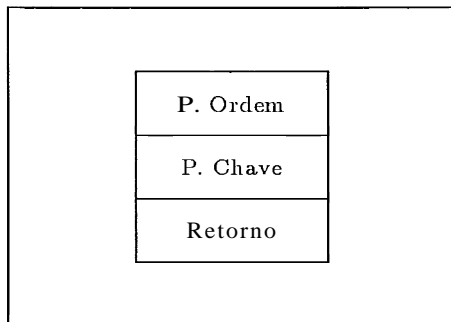


Figura 11. Menu para escolha da ordenação

- 2.a) A opção P.Ordem, se escolhida, determina que o usuário deseja ter acesso à entidade clicada ALUNO através de alguma das ordens que tenham sido especificadas para ela;

2.b) A opção P.CHAVE, se escolhida, determina que o usuário deseja ter acesso à entidade chamada ALUNO através de alguma das chaves que tenham sido especificadas para ela;

2.c) A opção RETORNO terá, nesse caso, o mesmo efeito daquele descrito no passo 1.c.

3) Clicar sobre a opção P.ORDEM

Esta opção deve ser escolhida já que a consulta pede que os alunos estejam em ordem de nome, ou seja, o acesso deverá ser feito pela ordem PORNOME. Nesse momento, as ordens especificadas para a entidade ALUNO deverão ser mostradas sempre através de janelas para que a próxima opção possa ser escolhida. Veja a Figura 12.

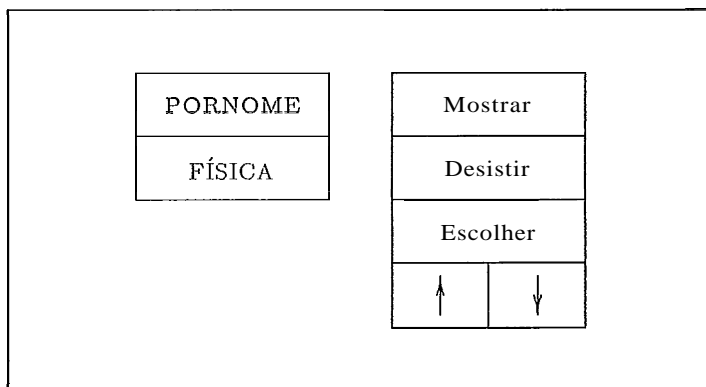


Figura 12. Menu para escolher Ordem de ALUNO

3.a) A opção MOSTRAR permite que o usuário veja as características da ordem que estiver marcada através do recurso de vídeo reverso. De início a primeira ordem estará sempre marcada. Esta opção é idêntica àquela existente na opção ESQUEMA como veremos mais adiante;

3.1) A opção DESISTIR. faz com que a Figura. 12 suma, voltando ao menu da Figura 11 e ao passo 3;

- 3.c) A opção ESCOLHER deve ser escolhida quando a ordem desejada estiver marcada em vídeo reverso;
- 3.d) As setas para cima e para baixo devem ser clicadas caso o usuário queira andar com a marca de vídeo reverso pelas ordens, respectivamente, para cima e para baixo.
- 4) Clicar sobre a opção ESCOLHER com a marca de vídeo reverso sobre a ordem PORNOME

Uma vez escolhida a ordem, resta-nos saber se a instância a que desejamos ter acesso neste momento é a primeira ou a última desta ordem. Assim, é mostrado o menu conforme a Figura 13. O usuário deverá optar por PRIMEIRO ou ÚLTIMO conforme o usuário queira o primeiro ou o último aluno daquela ordem, respectivamente. Caso queira desistir da ordem escolhida, o usuário poderá clicar o botão de *< Escape >* do *mouse* retornando ao menu da Figura 11 e ao passo 3. Caso não haja ordens definidas em ALUNO, uma mensagem de erro é mostrada.

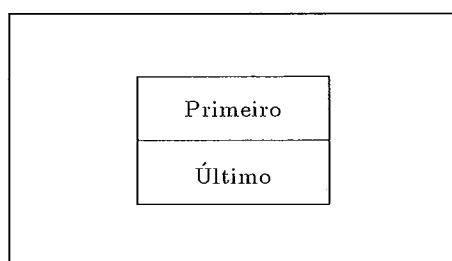


Figura 13. Menu para escolha de Primeiro/Último

- 5) Clicar sobre a opção PRIMEIRO

É nesse momento que a estrutura representada pela entidade ALUNO será instanciada e, no caso em questão, no primeiro aluno da ordem. O menu que é mostrado nesse instante é o da Figura 14, que diz respeito aos elementos instanciados.

- 5.a) A opção MOSTRAR faz com que sejam relacionados para o usuário os valores

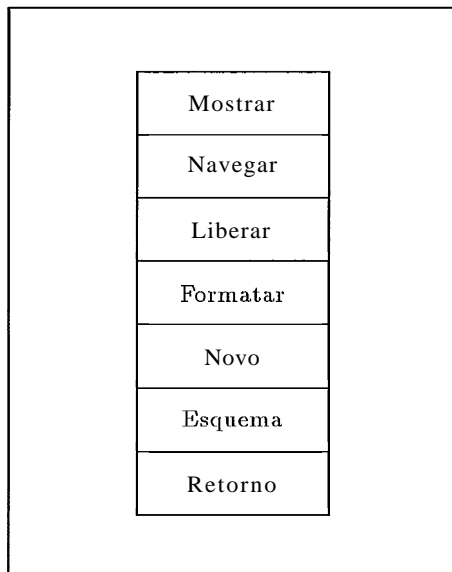


Figura 14. Menu para elementos instanciados

dos atributos da instância que estiver sendo apontada;

- 5.b) A opção **NAVEGAR** permite ao usuário, conforme já descrevemos anteriormente, navegar dentro do elemento clicado, no caso **ALUNO**, ou através de outros elementos do esquema **E-R** tendo como base um determinado caminho que deverá ser estabelecido;
- 5.c) A opção **LIBERAR** libera a instância que estiver sendo apontada naquele momento no elemento em questão, ou seja, **ALUNO**. Se o elemento fizer parte de um caminho que tenha sido traçado durante a navegação, todas as instâncias dos elementos que fazem parte deste caminho serão igualmente liberadas;
- 5.d) A opção **FORMATAR** permite que o usuário selecione os atributos que ele deseja que sejam mostrados através da opção **MOSTRAR**;
- 5.e) A opção **NOVO** faz com que o usuário possa determinar um novo caminho a seguir para a formulação de uma outra consulta, usando uma entidade já instanciada;
- 5.f) A opção **ESQUEMA** comporta-se como já descrito no passo *1.b*;

5.g) A opção **RETORNO** faz com que desapareçam os menus, voltando ao passo 1. Deve-se lembrar, no entanto, que o estado das instâncias é mantido, ou seja, se a entidade **ALUNO** encontrava-se instanciada no primeiro aluno, ao se clicar de novo esta entidade, o mesmo aluno estará sendo apontado.

6) Clicar sobre a opção **MOSTRAR**

Como já dissemos, são mostrados os valores dos atributos do aluno instanciado. Além da janela de valores, é mostrada em outra janela a palavra **RETORNO** que, ao ser clicada, faz desaparecer os valores mantendo somente o menu da Figura 14 junto com o esquema E-R.

7) Clicar sobre a opção **NAVEGAR** (ainda sobre o menu da Figura 14)

Aparecerá o menu da Figura 15. Após escolher a opção desejada, voltará ao passo 6 com a Figura 14.

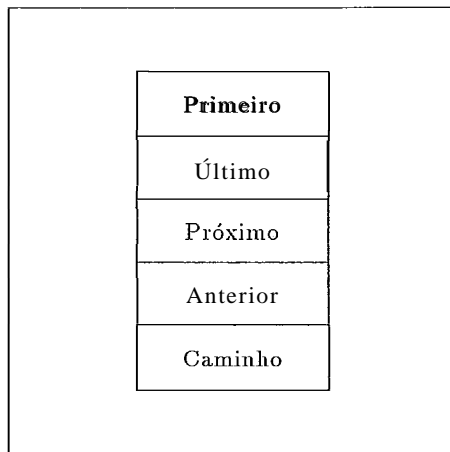


Figura 15. Menu para Navegação

7.a) A opção **PRIMEIRO** no caso da ordenação por ordem fará com que o primeiro aluno desta ordem seja instanciado;

7.b) A opção **ÚLTIMO** no caso da ordenação por chave fará com que o último aluno desta ordem seja instanciado;

Obs.: Quando a ordenação se dá por meio de chaves as opções PRIMEIRO e ÚLTIMO comportam-se de modo diferente. Ambas instanciam o mesmo aluno que é aquele cuja chave foi fornecida anteriormente.

- 7.c) A opção PRÓXIMO faz com que o próximo aluno daquela ordenação seja instanciado;
- 7.d) A opção ANTERIOR faz com que o aluno precedente ao atual, naquela ordenação, seja instanciado;
- 7.e) Esta opção CAMINHO é escolhida quando a navegação envolve outros elementos do esquema estabelecendo que rumo a consulta deverá seguir.

Nesse momento o usuário poderá navegar quanto desejar. Caso chegue ao início do arquivo ou ao seu final, aparecerá uma mensagem indicando *Fim de Arquivo*. Depois bastará retornar e o último aluno instanciado continuará assim até que se queira liberá-lo. Enquanto isto não ocorrer, sempre que ALUNO for clicado, aparecerá o menu da Figura 14.

Vamos, agora, acrescentar dois complicadores à consulta anterior. O primeiro deles é a mudança da ordenação. Agora, o aluno a ser mostrado deverá ser identificado por seu número de registro que constitui uma chave definida em ALUNO de nome PORREGISTRO. O segundo complicador é o resultado da consulta. Aqui desejamos obter de aluno somente seu nome.

A consulta que abrangerá tais complicadores é:

Obter nome do aluno de registro 871718200

Consideremos que tenhamos liberado a entidade ALUNO, instanciada na consulta anterior.

Para executá-la, repetem-se os passos 1 e 2. Os demais passos são os seguintes:

3) Clicar sobre a opção P.CHAVE

Esta opção deve ser escolhida já que a consulta pede que os alunos estejam ordenados pela chave PORREGISTRO. Nesse momento as chaves definidas para a entidade ALUNO deverão ser mostradas como na Figura 16. Os passos 3a, 3b, 3c e 3d são similares aos descritos com relação à consulta anterior. Caso não hajam chaves definidas para a entidade ALUNO, uma mensagem de erro será mostrada.

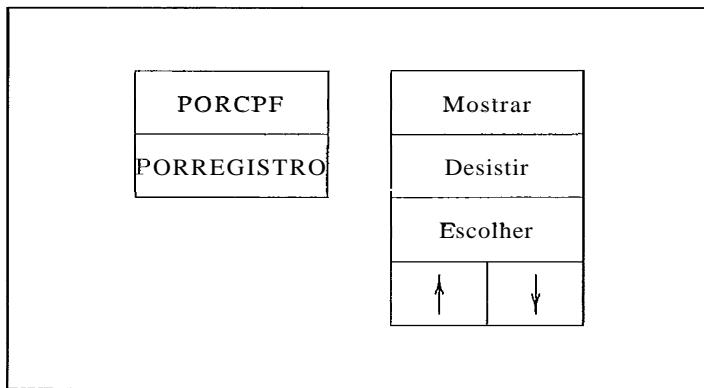


Figura 16. Menu para escolher Chave de ALUNO

4) Clicar sobre a opção ESCOLHER com a marca de vídeo reverso sobre a chave PORREGISTRO

Uma vez escolhida a chave precisamos ter o valor que será atribuído a REGISTRO. Para isso é aberta uma janela onde o usuário deverá digitar o valor 871718200. Uma vez digitado o valor, é mostrado o menu da Figura 14 caso tenha conseguido encontrar aquele aluno. Se, por outro lado, não tiver sido possível localizá-lo, uma mensagem de erro é mostrada, indicando que o aluno não foi encontrado.

5) Clicar sobre a opção FORMATAR

Como é pedido pela consulta, devemos ter somente o atributo ANOME como resultado. Nesse caso aparece um menu com os atributos como na Figura 17.

6) Clicar ESCOLHER sobre o atributo ANOME e retornar

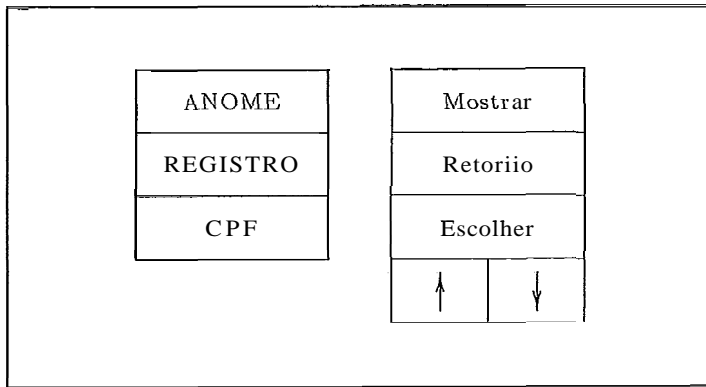


Figura 17. Menu para Formatação dos Atributos de ALUNO

Daí para frente os passos 7 e 8 são similares aos passos 6 e 7 da consulta anterior, respectivamente.

A opção **ESQUEMA** está presente em dois menus e possibilita ao usuário, a qualquer momento, tomar conhecimento das especificações do elemento clicado. Com essa finalidade a sequência de passos pode ser descrita como segue:

1) Clicar sobre a opção **ESQUEMA**

A partir daí é mostrado o menu da Figura 18 onde o usuário deverá escolher que especificação da entidade ou relacionamento deverá ser vista.

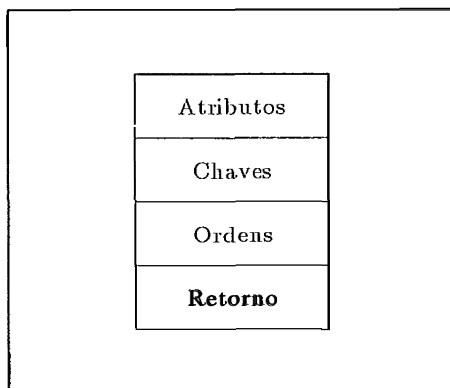


Figura 18. Menu para descrever o Esquema

1.a) A opção **ATRIBUTOS** possibilita ao usuário ver quais atributos pertencem

ao elemento clicaclo;

- 1.b) A opção CHAVES permite ao usuário conhecer as chaves definidas para o elemento clicaclo;
- 1.c) A opção ORDENS permite que o usuário tome conhecimento das ordens especificadas para o elemento clicaclo;
- 1.d) A opção RETORNO faz com que volte à tela em que mostra somente o esquema E-R.

Vamos assumir que desejamos ver as três opções, ou seja, atributos, chaves e ordens, uma de cada vez.

2) Clicar sobre a opção ATRIBUTOS

Nesse momento será mostrado o menu da Figura 19 onde o atributo a ser mostrado encontra-se marcado em vídeo reverso.

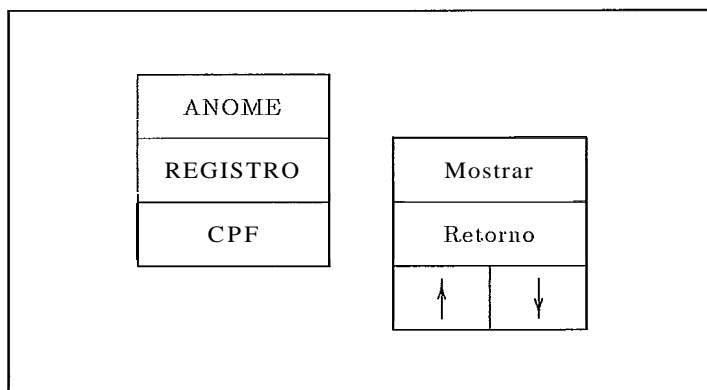


Figura 19. Menu para Mostrar Atributos de ALUNO

- 2.a) A opção MOSTRAR faz com que vejamos o tipo e a propriedade do atributo marcado;
- 2.b) A opção RETORNO nos leva de volta ao menu da Figura 18 ou ao passo 2;

2.c) As setas fazem com que mudemos a máscara de vídeo reverso de um atributo para outro.

3) Clicar a opção *MOSTRAR* sobre o atributo *ANOME*

O atributo *ANOME* foi definido como alfanumérico de tamanho 40 caracteres e não possui repetições. Sendo assim, aparecerá uma janela com o seguinte aspecto:

TIPO : ALPHA 40

PROPR : *SINGLE VALUED*

Uma outra janela com a palavra *RETORNO* sendo apontada pelo *mouse* é mostrada para que, ao ser clicada, desapareçam essas características do atributo.

4) Clicar sobre a janela *RETORNO*

Ao ser clicado sobre esta janela desaparecem as características do atributo e também essa janela, retornando ao passo 3 com o menu da Figura 19 para escolher de mais algum atributo, se desejar.

5) Clicar sobre a opção *RETORNO* do menu da Figura 19

Uma vez clicada esta opção a tela volta ao estado do passo 2 ou ao menu da Figura 18.

6) Clicar sobre a opção *CHAVES*

Nesse momento será mostrado o menu da Figura.20 onde a chave a ser mostrada encontra-se marcada em vídeo reverso.

6.a) A opção *MOSTRAR* faz com que vejamos os atributos que determinam a chave marcada e seus tipos;

6.b) A opção *RETORNO* nos leva de volta ao menu da Figura 18 ou ao passo 2;

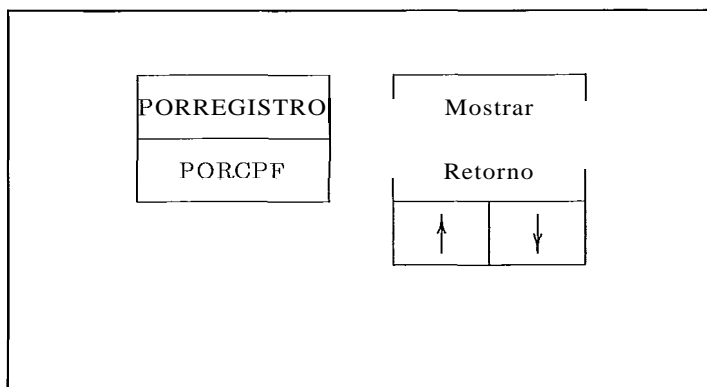


Figura 20. Menu para Mostrar Chaves de ALUNO

6.c) As setas funcionam da mesma forma com que foi descrito anteriormente.

7) Clicar MOSTRAR sobre a chave PORCPF

A chave PORCPF foi definida como sendo composta somente pelo atributo CPF que, por sua vez, foi especificado como alfanumérico de tamanho 11 caracteres. Também no caso de chaves aparecerão as duas janelas sendo a primeira da forma:

CPF A 11

A segunda janela é de RETORNO.

8) Clicar sobre a janela RETORNO

Ao clicar sobre esta janela volta-se ao passo 7 ou ao menu da Figura 20.

9) Clicar sobre a opção RETORNO do menu da Figura 20

A tela voltará ao estaclo do passo 2 ou do menu da Figura 18.

10) Clicar sobre a opção ORDENS

Nesse momento será mostrado o menu da Figura 21 onde a ordem a ser mostrada encontra-se marcada em vídeo reverso.

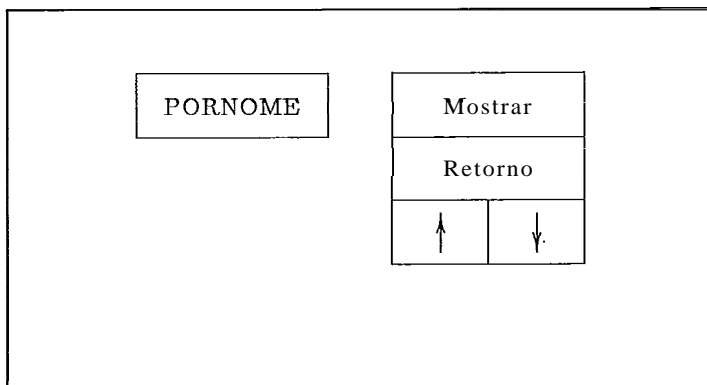


Figura 21. Menu para Mostrar Ordens de ALUNO

- 10.a) A opção MOSTRAR faz com que vejamos as características da ordem marcada;
- 10.b) A opção RETORNO nos leva de volta ao menu da Figura 18 ou ao passo 2;
- 10.c) As setas funcionam da mesma forma com que foi descrito anteriormente.

Obs: Observe que a ordem FÍSICA (registros na ordem em que estão fisicamente armazenados) já não aparece nesse menu. O motivo deve-se ao fato deste tipo de ordenação não possuir determinantes, origem e outros.

11) Clicar MOSTRAR sobre a ordem PORNOME

A ordem PORNOME foi definida sobre a entidade ALUNO tendo como atributo determinante ANOME. Também no caso das ordens aparecerão as duas janelas sendo a primeira da forma:

Nome : *PORNOME*
Dominio : *ALUNO*
Expressao : *Falso*
Determinante : *ANOME (A)*
Duplicacoes : *ultimo registro*
Origem : *Sistema*

Sendo que: *Domínio* consiste da entidade ou relacionamento onde foi definida a orclein em questão; *Expressão* indica uma seleção prévia de registros que poderão fazer parte da orclein; *Determinante* é o conjunto de atributos a partir do qual as tuplas encontram-se ordenadas (aquele A entre parênteses simboliza o sentido da ordenação significando, neste caso, que a orclein é ascendente); *Duplicações* indicam em uma consulta clue quando houver cluplicação de determinante o registro a ser lido será o primeiro ou o último registro daqueles com o mesmo determinante; *Origem da ordem* indica de que papel ela parte (no caso em questão, a ordem não parte de um papel. Ela somente ordena o próprio domínio).

Suponhamos agora que o elemento clicado inicialmente tenha sido o relacionamento INSCRIÇÃO. Ao ser clicada a opção MOSTRAR sobre a orclein INSCRIÇÕES apareceria.:

Nome : INSCRICOES
Domínio : INSCRICAO
Expressao : Falso
Determinante : CODIGO(A)
Duplicacoes : ultimo registro
Origem : ALUNO

Podemos verificar que a origem é o papel que relaciona ALUNO com INSCRIÇÃO. Como não há nome do papel definido no esquema, é assumido o nome da entidade envolvida, no caso, ALUNO. Os registros de INSCRIÇÃO encontram-se ordenados em grupos para cada aluno. Deste modo, para cada aluno há um grupo de disciplinas ordenadas por CÓDIGO nas quais o aluno está inscrito. A única forma de navegar por entre elementos se dá através da definição de ordens como no caso de INSCRIÇÕES, isto é, só será possível navegar de ALUNO para INSCRIÇÃO e por sua vez para DISCIPLINA através de uma orclein que indique esta direção como é o caso de INSCRIÇÕES.

A segunda janela é a de RETORNO.

12) Clicar sobre a janela RETORNO

Ao clicar sobre esta janela, volta-se ao passo 11 ou ao menu da Figura 21.

13) Clicar sobre a opção RETORNO do menu da Figura 21

A tela voltará ao estado do passo 2 ou do menu da Figura 18.

14) Clicar sobre a opção RETORNO do menu da Figura 18

A tela voltará ao seu estado no passo 1 ou ao menu que era então mostrado, podendo ser a Figura 10 ou a Figura 14. Dependerá de onde partiu a escolha da opção ESQUEMA.

Mesmo tendo apresentado alguns complicadores, a consulta anterior ao Banco de Dados é, ainda., considerada simples. Vamos, então, elevar um pouco mais o grau de dificuldade das consultas formuladas sobre o esquema E-R, incluindo a navegação por entre os elementos do esquema. Seja a seguinte consulta:

Obter nome das disciplinas nas quais o aluno de registro 871718200 encontra-se inscrito

Esta consulta visa obter os nomes das disciplinas que estejam associadas a um determinado aluno instanciado através de seu número de registro. Uma vez identificado o aluno por seu registro, uma ordem parcial, INSCRIÇÕES, faz com que o grupo de disciplinas nas quais este encontra-se inscrito seja instanciado, uma disciplina de cada vez.

Como podemos observar, esta consulta englobará as seguintes funções:

instanciação da entidade ALUNO através de sua chave PORREGISTRO;

navegação através de um caminho em direção ao relacionamento INSCRIÇÃO, via a ordem INSCRIÇÕES com origem em ALUNO;

- . através do determinante da ordem INSCRIÇÕES que é CÓDIGO de DISCIPLINA, representado por meio de surrogates, navegação ao longo de um caminho em direção à entidade DISCIPLINA, onde haverá o surrogate correspondente aquele citado;
- . navegação dentro do grupo de disciplinas correspondente aquele aluno.

Vejamos então como se comportaria o conjunto de passos adotados com a finalidade de efetuar esta consulta a partir de uma tela onde apenas o esquema E-R está sendo retratado e onde não há entidades instanciadas até o momento:

Obs: Não repetiremos aqui as explicações sobre as opções que já tenham sido dadas anteriormente.

1) Clicar sobre a entidade ALUNO

Aparecerá o menu da Figura 10.

2) Clicar sobre a opção INSTANCIAR

Será mostrado o menu da Figura 11.

3) Clicar sobre a opção P.CHAVE

A chave que se deseja é PORREGISTRO. Deve-se, então, posicionar a marca de vídeo reverso sobre ela ao aparecer o menu da Figura 16.

4) Clicar sobre a opção ESCOLHER

Uma janela será aberta com o objetivo de captar o valor que desejamos seja atribuído à chave, no caso, 8'71'715200. Uma vez digitado o valor, aparecerá o menu da Figura 14.

5) Clicar sobre a opção NAVEGAR

Ao ser clicada esta opção surgirá o menu da Figura 15. Como o que desejamos é caminhar por entre os elementos do esquema, a opção escolhida deverá ser a última.

6) Clicar sobre a opção CAMINHO

Uma vez escolhida esta opção sumirão todos os menus ficando na tela somente o esquema E-R. A partir desse ponto precisaremos indicar qual seria o próximo elemento a fazer parte do caminho a ser seguido.

7) Clicar sobre o relacionamento INSCRIÇÃO

A primeira coisa a ser feita será verificar se o relacionamento INSCRIÇÃO é ligado à entidade ALUNO através de algum papel. Se não o for, uma mensagem de erro será dada. Caso contrário, dá prosseguimento à consulta. Para que seja possível processar a navegação entre elementos, é necessário que se tenha definido alguma ordem que indique o caminho a ser seguido. No caso em questão essa ordem é INSCRIÇÕES que, como já dissemos anteriormente, tem sua origem em ALUNO. Esta definição torna viável a navegação por este caminho. E mostrado, então, o menu da Figura 10.

8) Clicar sobre a opção INSTANCIAR

Será mostrado o menu da Figura. 11.

9) Clicar sobre a opção P.ORDEM

Como já foi dito, deve haver, obrigatoriamente, uma ordem em INSCRIÇÃO que tenha sua origem em ALUNO de modo a permitir que se navegue no sentido ALUNO-INSCRIÇÃO. Esta ordem é INSCRIÇÕES. Ao ser clicada esta opção aparecerá o menu da Figura 22.

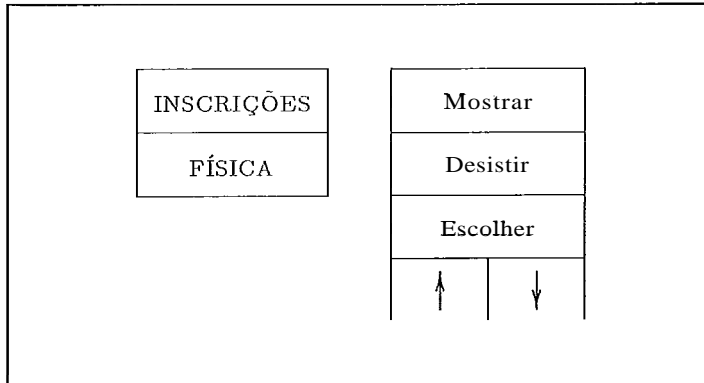


Figura 22. Menu para selecionar Ordem de INSCRIÇÃO

- 10) Clicar a opção ESCOLHER sobre a ordem INSCRIÇÕES

Será verificado se a ordem escolhida atende às características exigidas, ou seja, se ela tem como origem o papel que liga a entidade ALUNO ao relacionamento INSCRIÇÃO. Quando não é determinado o nome do papel, a origem escolhida é a entidade da qual partiu, isto é, ALUNO. Surgirá., então, o menu da Figura 13.

- 11) Clicar sobre a opção PRIMEIRO

Nesse instante o relacionamento INSCRIÇÃO será instanciado apontando para a primeira inscrição do aluno cujo registro é 871718200. Será mostrado em seguida o menu da Figura 14.

- 12) Clicar sobre a opção NAVEGAR

Voltará assim a aparecer o menu da Figura 15. O que desejamos é caminhar ainda mais, em direção à entidade DISCIPLINA. Portanto a opção a ser escolhida deverá ser CAMINHO.

- 13) Clicar sobre a opção CAMINHO

A partir desta escolha apenas o esquema E-R estará na tela. Deveremos indicar o próximo elemento que compõe a consulta.

14) Clicar sobre a entidade DISCIPLINA

Mais uma vez será feita uma verificação para constatar-se se há algum papel ligando INSCRIÇÃO a DISCIPLINA. Se o resultado for negativo uma mensagem de erro será mostrada. Se for afirmativo será dada continuidade à consulta aparecendo o menu da Figura 10.

Obs: Note, entretanto, que teremos aqui uma diferença importante com relação ao passo 7. Ocorre que, como vimos, para se caminhar de ALUNO para INSCRIÇÃO precisaríamos de uma ordem que indicasse este caminho. Isto deve-se ao fato de, para cada aluno, haver um número x ($x \geq 0$) de inscrições em disciplinas às quais só poderíamos ter acesso partindo de um determinado aluno, ou seja, a relação entre ALUNO e INSCRIÇÃO é de $1 : n$. O mesmo não ocorre quando caminhamos de INSCRIÇÃO para DISCIPLINA. Dado que para cada inscrição há uma disciplina correspondendo, representada através de seu surrogate, não precisamos de uma ordem para esta navegação pois a relação entre esses dois elementos é de $n : 1$ (essa navegação é sempre permitida). Observe que, se a consulta fosse da forma:

Obter nomes dos alunos que cursam a disciplina Física I

nós não poderíamos executá-la pois não há uma ordem definida em INSCRIÇÃO tendo como origem a entidade DISCIPLINA.

15) Clicar sobre a opção INSTANCIAR

Como já dissemos anteriormente, para cada aluno instanciado em ALUNO há um conjunto de inscrições de INSCRIÇÃO que, através da ordem INSCRIÇÕES, encontra-se ordenado com base em seu determinante CÓDIGO. Por essa razão não precisaremos estabelecer uma forma de acesso à DISCIPLINA. Deste modo, após escolhermos a opção INSTANCIAR, passaremos diretamente para o menu da Figura 14.

16) Clicar sobre a opção **FORMATAR**

Aparecerá o menu da Figura 23 onde a marca de víceo reverso deverá ser posicionada sobre o atributo que desejamos ver, ou seja, **DNOME**.

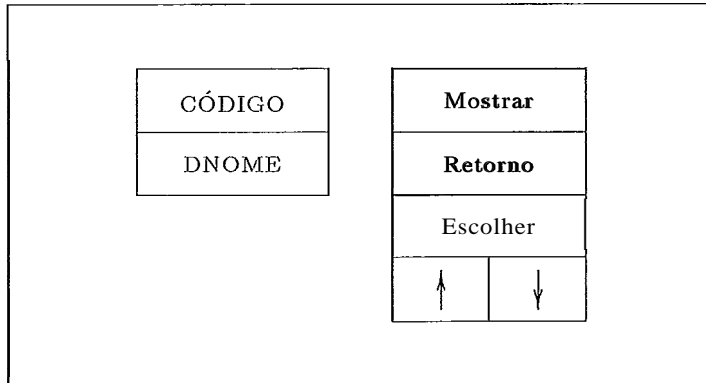


Figura 23. Menu para Formatação de Atributos de DISCIPLINA

17) Clicar a opção **ESCOLHER** sobre o atributo **DNOME**

Ao se escolher o atributo **DNOME** o menu da Figura 23 permanece na tela.

18) Clicar sobre a opção **RETORNO** da Figura 23

Esta opção faz com que retornemos ao menu da Figura 14 onde poderemos agora ver o resultado da consulta.

19) Clicar sobre a opção **MOSTRAR**

Agora veremos o atributo **DNOME** correspondente à disciplina da primeira inscrição em ordem de **CÓDIGO** cursada pelo aluno de registro 871718200 inostrada em uma janela. Uina outra janela com a palavra **RETORNO** também é inostrada e devemos clicá-la para retornarmos ao menu da Figura 14. Se desejarmos conhecer as disciplinas daquele aluno cleveremos navegar ao longo das instâncias de **INSCRIÇÃO**.

20) Clicar sobre a opção RETORNO da Figura 14

Voltará a aparecer na tela somente o esquema E-R.

21) Clicar sobre o relacionamento INSCRIÇÃO

Aparecerá o menu da Figura 14 já que este elemento encontra-se instanciado.

22) Clicar sobre a opção NAVEGAR

Nesse momento aparecerá o menu da Figura 15 que nos permitirá navegar pelas inscrições.

22.a) A opção PRIMEIRO deverá instanciar o relacionamento INSCRIÇÃO sobre a primeira inscrição em ordem de CÓDIGO de disciplina correspondente ao aluno de registro 871718200;

22.b) A opção ÚLTIMO deverá instanciar o relacionamento INSCRIÇÃO sobre a última inscrição em ordem de CÓDIGO de disciplina correspondente ao aluno de registro 871718200;

22.c) A opção PRÓXIMO fará com que a próxima inscrição daquele aluno, em ordem de CÓDIGO, seja instanciada;

22.d) A opção ANTERIOR fará com que a inscrição anterior daquele aluno, em ordem de CÓDIGO, seja instanciada;

22.e) A opção CAMINHO neste caso não se aplica pois a consulta não envolve outros elementos além dos já clicados.

23) Clicar sobre a opção PRÓXIMO

A próxima inscrição do aluno de registro 871718200 será instanciada e voltará ao menu da Figura 14.

24) Clicar sobre a opção **RETORNO** da Figura 14

Assim retornaremos ao diagrama E-R.

25) Clicar sobre a entidade **DISCIPLINA**

Aparecerá o menu da Figura. 14.

26) Clicar sobre a opção **MOSTRAR**

Este passo comportar-se-á como o passo 19, sendo que a disciplina aqui mostrada é aquela correspondente à próxima inscrição do aluno instanciado, seguindo a ordem **INSCRIÇÕES**. A janela contendo a palavra **RETORNO** deverá ser clicada retornando ao menu da Figura 14.

27) Clicar sobre a opção **RETORNO** da Figura 14

Retornaremos ao esquema E-R.

Lembramos que o usuário terá acesso nesse momento somente ao grupo de disciplinas para as quais há uma inscrição por parte do aluno de registro 871718200. Caso o usuário deseje informações a respeito do próximo aluno segundo a ordenação através da chave **PORREGISTRO**, ele deverá seguir os seguintes passos:

28) Clicar sobre a entidade **ALUNO**

Como essa entidade faz parte da consulta e encontra-se instanciada, é mostrado o menu da Figura 14.

29) Clicar sobre a opção **NAVEGAR**

Uma vez escolhida esta opção, surgirá o menu da Figura 15.

30) Clicar sobre a opção **PRÓXIMO**

O próximo aluno cujo registro for imediatamente superior a 871718200, já que os alunos encontram-se ordenados pela chave PORREGISTRO, será instanciado e, automaticamente, a ordem INSCRIÇÕES estará apontando para a primeira inscrição deste aluno. Também a entidade DISCIPLINA estará sendo apontada para a disciplina correspondente. Bastará, então, seguir os passos já vistos anteriormente em busca dos dados das várias instâncias de INSCRIÇÃO.

Vejamos agora uma situação ainda mais complexa que aquela descrita anteriormente. A consulta envolve o auto-relacionamento mostrado na Figura 7 que é composto da entidade DISCIPLINA e do relacionamento REQUISITO. Note que nesse caso específico nós determinamos os nomes dos papéis, já que haveria dois caminhos ligando aqueles dois elementos. Observe ainda que foi necessário definir duas ordens distintas REQUISDISCIP e REQUISREQ, cada uma delas tendo origem em um papel, respectivamente, *discip* e *req*. Caso queiramos utilizar a opção ESQUEMA para vermos as especificações dessas ordens, teremos:

Nome : *REQUISDISCIP*

Dominio : *REQUISITO*

Espressao : *Falso*

Determinante : *CODIGO(A)*

Dzippliccicoes : *ultimo registro*

Origem : *discip*

Nome : *REQUISREQ*

Dominio : *REQUISITO*

Expressao : *Falso*

Determinante : *CODIGO(A)*

Duplicacoes : *ultimo registro*

Origem : *req*

Em um auto-relacionamento uma entidade está relacionada a ela mesma através de um relacionamento. Vamos exemplificar esta situação para que fique mais claro. Seja a consulta:

*Obter nomes das disciplinas que sejam requisitos
de disciplinas na ordem física dos registros*

Esta consulta visa obter os nomes das disciplinas, na ordem física de suas instâncias na entidade DISCIPLINA, que sejam requisitos de quaisquer outras disciplinas.

Vamos então definir algumas das características dessa consulta:

- . Baseia-se em um auto-relacionamento que nos leva a passar por duas vezes pela entidade DISCIPLINA ao longo de nosso caminho de navegação;
- . A consulta deverá partir como sempre da entidade, no caso DISCIPLINA, em direção ao relacionamento REQUISITO passando pelo papel *discip*;
- . A entidade DISCIPLINA deverá ser instanciada por meio de sua ordem física em busca das disciplinas;

A ordem que possibilita esta primeira parte do caminho é REQUISREQ;

- . A segunda parte do caminho partirá do relacionamento REQUISITO em direção à entidade DISCIPLINA passando pelo papel req;
- . Os requisitos estarão organizados em ordem de CÓDIGO de DISCIPLINA, ou seja, para cada disciplina haverá um grupo de requisitos ordenados por seus códigos.

Uma vez dada uma visão global do roteiro da consulta, vejamos os passos mais detalhados que o usuário deverá seguir para obter o resultado esperado. Consideremos, agora, que esta consulta será feita sobre um esquema onde já há um caminho definido anteriormente, ou seja, o caminho ALUNO-INSCRIÇÃO-DISCIPLINA, conforme foi descrito. Tal situação introduz um outro complicador em nossa atual consulta.

Como veremos, em alguns momentos, precisaremos especificar a que caminho estaremos nos referindo:

1) Clicar sobre a entidade *DISCIPLINA*

Como a entidade *DISCIPLINA* encontra-se instanciada, embora esta instanciação deva-se à consulta anterior, o menu que aparecerá será o da Figura 14. O objetivo é saber se o usuário pretende dar sequência à consulta anterior ou estabelecer uma nova.

2) Clicar sobre a opção *NOVO*

Esta opção fará com que o caminho atual seja desmarcado e somente a entidade *DISCIPLINA* seja marcada na tela. Nesse momento, aparecerá o menu da Figura 10.

3) Clicar sobre a opção *INSTANCIAR*

Será mostrado o menu da Figura 11.

4) Clicar sobre a opção *P.ORDEM*

A ordem desejada é *FÍSICA*. Deve-se posicionar a marca de vídeo reverso sobre ela ao aparecer o menu da Figura 24.

5) Clicar a opção *ESCOLHER* sobre a ordem *FÍSICA*

Neste momento não será mostrado o menu da Figura 13, pois a ordem selecionada é a física e, portanto, não é permitido que ela aponte para o seu último registro, ou seja, a opção *ULTIMO* não é permitida.. A entidade *DISCIPLINA* será instanciada apontando para a primeira disciplina do arquivo físico. Será mostrado, em seguida, o menu da Figura 14.

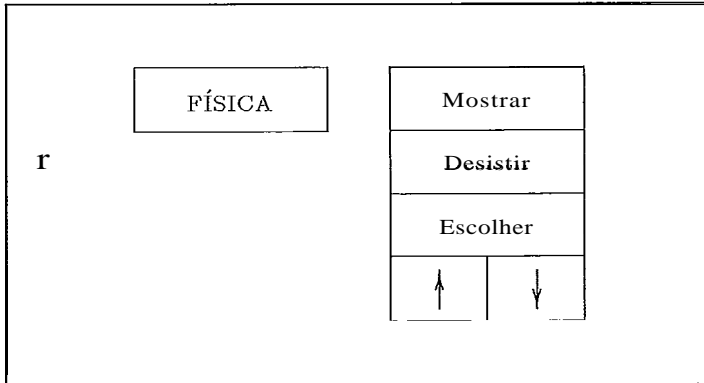


Figura 24. Menu para selecionar Ordem de DISCIPLINA

Obs: Da mesma forma que nas consultas anteriores, podemos solicitar que sejam mostrados os atributos da instância de DISCIPLINA neste momento. No entanto não iremos considerar aqui esta situação.

6) Clicar sobre a opção NAVEGAR

Ao ser clicada esta opção surgirá o menu da Figura 15.

7) Clicar sobre a opção CAMINHO

Uma vez que desejamos caminhar por entre os elementos do esquema, a opção a ser escolhida deverá ser CAMINHO. Nesse momento sumirão todos os menus permanecendo na tela somente o esquema E-R. Em seguida precisaremos indicar o próximo elemento a fazer parte do caminho.

8) Clicar sobre o relacionamento REQUISITO

O relacionamento REQUISITO é ligado à entidade DISCIPLINA através de um papel. Devido ao fato da resposta ser afirmativa podemos criar sequência à consulta estabelecendo a ordem que nos permitirá esta navegação, ou seja, uma que tenha origem no papel que desejamos seguir. Este papel é o *discip*. É mostrado, então, o menu da Figura 10.

9) Clicar sobre a opção INSTANCIAR

Será mostraclo o menu da Figura 11.

10) Clicar sobre a opção P.ORDEM

É necessário que haja uma ordem em REQUISITO que, tendo sua origem em *discip*, permita a navegação no sentido DISCIPLINA-REQUISITO. Esta ordem é REQUISREQ. Ao ser clicacla esta opção aparecerá o menu da Figura 25.

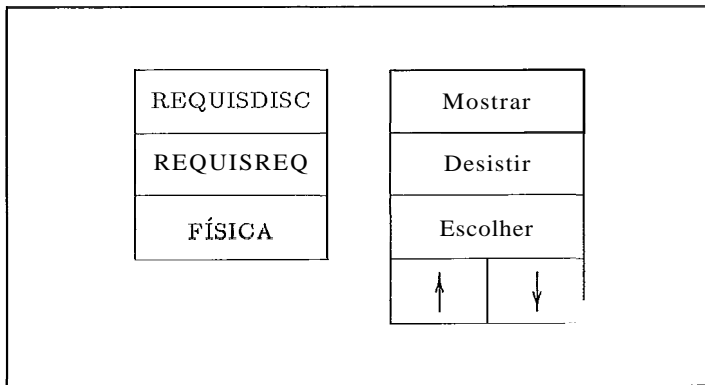


Figura 25. Menu para selecionar Ordem de REQUISITO

11) Clicar a opção ESCOLHER sobre a ordem REQUISREQ

Deverá ser verificado, inicialmente, se a ordem escolhida atende às características esperadas, ou seja, se ela tem como origem o papel *discip* que liga a entidade DISCIPLINA ao relacionamento REQUISITO. Surgirá, então, o menu da Figura 13.

12) Clicar sobre a opção PRIMEIRO

O relacionamento REQUISITO será instanciado apontando para o primeiro requisito da disciplina já instanciada. Será mostrado em seguida o menu da Figura 14.

13) Clicar sobre a opção NAVEGAR

Aparecerá o menu da Figura 15.

14) Clicar sobre a opção CAMINHO

Assim continuaremos a navegação pelo esquema indicando o próximo elemento.

15) Clicar sobre a entidade DISCIPLINA

Nessa segunda passagem por esta entidade deveremos definir que resultado desejamos obter. Inicialmente uma verificação deverá ser feita com o intuito de constatar se há um outro papel ligando REQUISITO a DISCIPLINA. Este papel é o req. Essa navegação é permitida também pela relação $1 : 1$ existente sobre o papel it req no sentido REQUISITO-DISCIPLINA, ou seja, dado um requisito, há apenas uma disciplina que corresponda ao seu surrogate. Aparecerá o menu da Figura 10.

16) Clicar sobre a opção INSTANCIAR

Para cada disciplina instanciada no passo 5, há um conjunto de requisitos em REQUISITO. Esses requisitos encontram-se ordenados segundo a ordem REQUIS-REQ, ou seja, através do seu determinante CÓDIGO. Por esse motivo não precisaremos definir uma forma de acesso para esta última entidade. Assim, após escolhida esta opção, passaremos diretamente para o menu da Figura 14.

17) Clicar sobre a opção FORMATAR

Aparecerá o menu da Figura 23 onde a marca de víbleo reverso deverá ser posicionada sobre o atributo DNOOME.

18) Clicar a opção ESCOLHER sobre o atributo DNOOME

O menu da Figura 23 permanecerá na tela.

19) Clicar a opção **RETORNO** da Figura 23

Retornar-se-á ao menu da Figura 14, onde poderemos obter o resultado.

20) Clicar sobre a opção **MOSTRAR**

O atributo **DNOME** que nos será mostrado é aquele pertencente à disciplina cujo código é o primeiro do conjunto de requisitos que atendem à ordem **REQUISREQ**, correspondente à disciplina instanciada no passo 5. A janela contendo a palavra **RETORNO** deverá ser clicada para que voltemos ao menu da Figura 14. Se desejarmos conhecer os requisitos da disciplina inicialmente instanciada poderemos navegar ao longo das instâncias de **REQUISITO**.

21) Clicar sobre a opção **RETORNO** da Figura 14.

Voltará a aparecer somente o esquema E-R na tela.

22) Clicar sobre o relacionamento **REQUISITO**

Aparecerá o menu da Figura 14 já que este relacionamento encontra-se instanciado.

23) Clicar sobre a opção **NAVEGAR**

Surgirá o menu da Figura 15 que nos permitirá navegar pelos requisitos.

23.a) A opção **PRIMEIRO** deverá instanciar o relacionamento **REQUISITO** sobre o primeiro requisito em ordem de **CÓDIGO** de disciplina correspondente à disciplina instanciada em primeiro lugar;

23.b) A opção **ÚLTIMO** deverá instanciar o relacionamento **REQUISITO** sobre o último requisito em ordem de **CÓDIGO** de disciplina correspondente à disciplina instanciada em primeiro lugar;

23.c) A opção PRÓXIMO fará com que o próximo requisito daquela disciplina, em ordem de CÓDIGO, seja instanciado;

23.d) A opção ANTERIOR fará com que o requisito anterior daquela disciplina, em ordem de CÓDIGO, seja instanciado;

23.e) A opção CAMINHO neste caso não se aplica pois a consulta não envolve outros elementos além dos já clicados.

24) Clicar sobre a opção PRÓXIMO

O próximo requisito da disciplina instanciada será apontado e voltará ao menu da Figura 14.

25) Clicar sobre a opção RETORNO da Figura 14

Retornaremos ao esquema E-R.

26) Clicar sobre a entidade DISCIPLINA

Como este passo parte de um clic sobre uma entidade do esquema e a entidade clicada DISCIPLINA aparece em dois caminhos distintos, sendo possível, a partir dela, estender o caminho de ambas as consultas, necessitamos saber de que caminho desejamos tratar agora. Aparecerá, então, o menu da Figura 14.

27) Clicar sobre a opção MOSTRAR

Ao ser clicada esta opção, para conhecermos o caminho a ser escolhido, será mostrado o menu da Figura 26, acompanhado da pergunta

Entidade Instanciada !

Deseja continuar neste caminho ?

27.a) A opção SIM fará com que a consulta permaneça no caminho atualmente marcado;

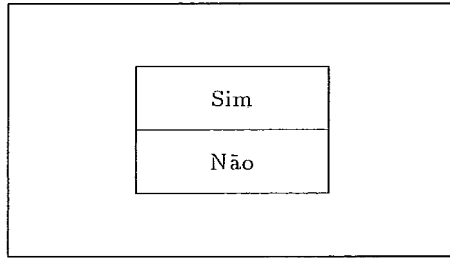


Figura 26. Menu Sim/Não

27.13) A opção NÃO fará com que a consulta tenha o seu caminho mudado, podendo o usuário especificar qual será ele.

28) Clicar sobre a opção SIM

Esta opção deve ser escolhida já que o caminho desejado é o atual. Nesse momento, aparecerá o menu da Figura 27, pois há duas entidades DISCIPLINA na consulta do caminho atual, cada uma com uma instância diferente:

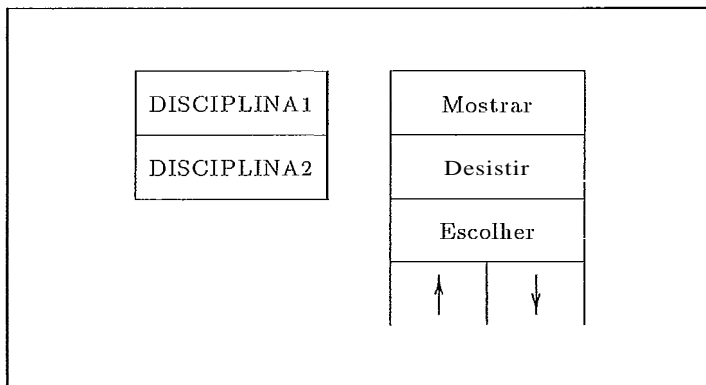


Figura 27. Menu para selecionar a Entidade DISCIPLINA

28.a) A opção MOSTRAR aqui terá efeito diferente das anteriores que encontravam-se em menus semelhantes. Ela deverá mostrar a instância atual da entidade escolhida;

28.b) A opção DESISTIR faz retornar ao esquema E-R desaparecendo o menu;

28.c) A opção ESCOLHER marca a entidade escolhida pelo usuário;

28.d) As setas permitem a movimentação entre as entidades do menu.

29) Clicar a opção ESCOLHER sobre a entidade DISCIPLINA2

Nesse momento, sumirá o menu da Figura 27 e será inostrado o atributo DNOOME correspondente ao próximo requisito da disciplina inicialmente instanciada. Ao ser clicacla a janela contendo a palavra RETORNO, voltaremos ao menu da Figura 14.

30) Clicar sobre a opção RETORNO da Figura 14

Retornaremos ao esquema E-R.

Semelhante à consulta anterior, o usuário poderá querer conhecer os requisitos da próxima disciplina na orclem física de DISCIPLINA. Os passos que ele deverá seguir são:

31) Clicar sobre a entidade DISCIPLINA

Pela mesma razão descrita no passo 26, voltará a aparecer o menu da Figura 27.

32) Clicar a opção ESCOLHER sobre a entidade DISCIPLINA1

Passaremos para o menu da Figura. 14.

33) Clicar sobre a opção NAVEGAR

Uma vez clicacla esta opção aparecerá o menu da Figura 15.

34) Clicar sobre a opção PRÓXIMO

A próxima disciplina da orclem física será instanciada e, automaticamente, a ordein REQUISREQ estará apontando para o primeiro requisito desta nova disciplina. Bastará, então, seguir os passos já vistos anteriormente em busca dos

dados das várias instâncias de REQUISITO.

Obs: Seguindo a ordem física não será possível navegar usando ANTERIOR.

Há uma situação que não foi, ainda, mencionada. Neste exato momento, nosso esquema da Figura 7 está na seguinte posição:

- Dois caminhos foram definidos e encontram-se instanciados: ALUNO-INSCRIÇÃO-DISCIPLINA e DISCIPLINA1-REQUISITO-DISCIPLINA2;
- O caminho atualmente marcado é DISCIPLINA1-REQUISITO-DISCIPLINA2.

Suponhamos, agora., que queiramos retornar ao primeiro caminho para obtermos a próxima disciplina em que o aluno instanciado está inscrito. Para isso, deveremos seguir os seguintes passos:

1) Clicar sobre a entidade DISCIPLINA

Como esta entidade já, encontra-se instanciada, o menu da Figura 14 será mostraclo.

2) Clicar sobre a opção NAVEGAR

Como esta é a Ultima entidade participante instanciada nos dois caminhos marcados no esquema, será.necessário que saibamos se o usuário deseja permanecer navegando através do caminho atual ou através de outro caminho, no caso o primeiro caminho. Por esta razão, o menu da Figura 26 deverá ser mostraclo, acompanhado daquela pergunta já mostrada.

3) Clicar sobre a opção NÃO

Como o caminho desejado não é o atual, a opção NÃO deverá ser escolhida, no que será mostrado o menu da Figura 28:

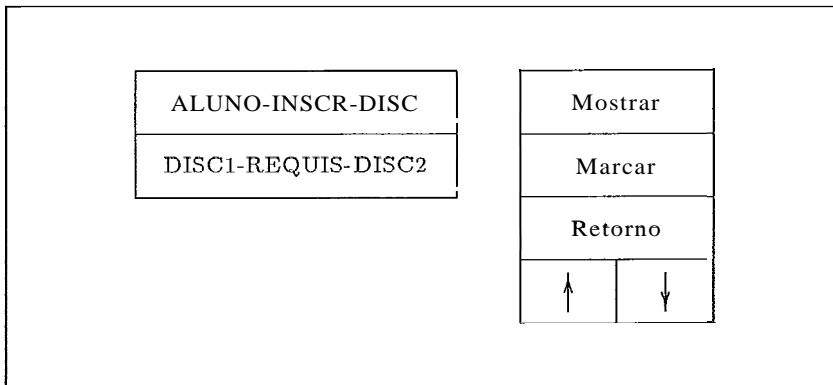


Figura 28. Menu para selecionar o Caminho

- 3.a) A opção **MOSTRAR** fará com que seja mostrada a atual instância da entidade clicada no caminho selecionado;
- 3.11) A opção **MARCAR** fará com que seja marcado o caminho escolhido, passando este a ser o caminho atual a ser navegado;
- 3.c) A opção **RETORNO** fará com que desapareça este menu, executando a navegação solicitada., mostrando o menu da **Figura 15**;
- 3.d) As setas permitem a movimentação entre os caminhos mostrados no menu.
- 4) Clicar a opção **MARCAR** sobre o caminho **ALUNO-INSCRIÇÃO-DISCIPLINA**

Permanecerá o menu da **Figura 28** com uma marca ao lado do caminho escolhido.
- 5) Clicar sobre a opção **RETORNO**

Este menu desaparecerá, voltando ao esquema com o menu da **Figura 15** na tela.
- 6) Clicar sobre a opção **PRÓXIMO**

Nesse momento, a próxima instância de **DISCIPLINA** relativa àquela entidade do caminho **ALUNO-INSCRIÇÃO-DISCIPLINA** será apontada, aparecendo o menu da **Figura 14**.

7) Clicar sobre a opção MOSTRAR

Será mostrada ao usuário a instância apontada com seus valores.

Queremos lembrar que os menus das Figuras 26 e 28 serão mostrados nessa mesma situação descrita, ou seja, no caso da existência de mais de um caminho instanciado, também para as opções escolhidas do menu da Figura 14 MOSTRAR, NAVEGAR, LIBERAR e FORMATAR.

VII.6 - Conclusão

Como pudemos observar neste capítulo, a Interface Gráfica de Consultas, associada ao Modelo Entidade-Relacionamento, é uma forma bem mais simples de fazer consultas *ad hoc* a um Banco de Dados do que aquelas tradicionais descritas nos capítulos iniciais. Sua interface com o usuário é bem mais amigável. A utilização do diagrama E-R, a partir do qual o BD foi definido e gerado, como base para a formulação de consultas, proporciona ao usuário, principalmente o não experiente, um alto grau de satisfação, já que sua interação com o *software* é bastante intuitiva. Além disso, ele mesmo poderá ter sido responsável pela elaboração do esquema, inicialmente.

Sobre o esquema E-R do banco de dados, a orientação do usuário através de menus, nos quais as opções deverão ser escolhidas por meio de *mouse*, proporciona, ainda, a ele, uma forma bem mais fácil e agradável de exercer seu trabalho.

O tipo de consulta que, ao usuário, será permitido elaborar, no entanto, está limitado aquele em que a interface gráfica poderá, sozinha, responder. O fato é que há duas formas de consultas ao BD, ambas utilizando-se dessas interfaces. Uma delas é aquela proposta e implementada neste trabalho, ou seja, utiliza-se somente de interfaces gráficas para efetuar consultas ao BD. As consultas puramente gráficas trazem dificuldade em se expressar acessos a uma base de dados quando as consultas forem

muito complexas, envolvendo condições de seleção ou manipulação de fragmentos de consultas, por exemplo. A segunda forma de acesso, por sua vez, tem definida, juntamente à interface gráfica, uma linguagem simplificada, elaborada a partir de uma forma geral de consultas. Assim, essa linguagem permitirá a formulação de consultas mais complexas como na implementação do uso de condições de seleção das instâncias das estruturas do BD. Essas condições, internamente, funcionarão como as operações algébricas mostradas anteriormente, UNIÃO e DIFERENÇA, por exemplo, ou condições de desigualdade, entre outras.

Devido ao elevado grau de complicação advindo da determinação de uma forma geral de consultas que se aplique às consultas mais completas, optamos por desenvolver, no âmbito deste trabalho, a primeira forma de acesso descrita. A outra forma, mais completa, será alvo de outra fase de desenvolvimento da Interface Gráfica de Consultas, a ser estudada futuramente.

Com relação à forma de acesso voltada somente à Interface Gráfica adotada neste trabalho, dois conceitos foram considerados: *folheamento* e navegação.

O *folheamento* permite que tenhamos acesso à base de dados através da manipulação do *mouse* sobre o esquema E-R e sobre os menus. As instâncias do BD serão mostradas através do acesso às estruturas por meio de chaves ou ordens. As ordens permitem um acesso ordenado às instâncias das entidades e relacionamentos segundo sua sequência física ou segundo um conjunto de atributos que forma uma chave, definido quando da especificação do esquema E-R. Esta chave não é, necessariamente, única. As ordens, quando parciais, permitem que uma lista de instâncias de um relacionamento, associadas a uma instância de uma entidade, seja acessada. As chaves, por sua vez, identificam unicamente uma instância. O acesso às estruturas através da chave é feito quando o usuário fornece, por meio do teclado, os valores dos atributos que a compõem.

A navegação é um conceito que visa permitir maior complexidade das consultas formuladas ao BD. Através dela, o usuário poderá, não apenas caminhar pelas

demais instâncias da estrutura instanciada, como também, especificar um caminho envolvendo mais relacionamentos e entidades do esquema. A navegação dentro de uma mesma estrutura permitirá que o usuário, segundo uma mesma ordenação, veja a próxima instância, a anterior, ou, nos casos de acesso por ordem, a primeira ou a última instâncias também. Já a navegação através de um caminho de entidades e relacionamentos permitirá ao usuário conhecer instâncias de uma entidade que estejam associadas a uma instância de uma outra entidade através de um relacionamento, obtida por meio de chave ou ordem.

Alguns dos conceitos tratados neste trabalho fazem parte, também, de outros trabalhos nesta área de aplicação. Alguns deles aproximam-se mais que outros da filosofia aqui adotada. Analisaremos suas características no capítulo seguinte.

O Módulo de Consultas com Interfaces Gráficas foi desenvolvido utilizando o Turbo Pascal [32]. Sua programação envolveu o módulo do Turbo Pascal 5.5 com orientação a objetos [31]. A interface gráfica que desenha o diagrama E-R, as janelas, os menus, foi desenvolvida através do *Turbo Graphics* [30] deste mesmo *software*. Também o uso e o acesso ao banco de dados foi implementado com a utilização de outra parte do pacote Turbo Pascal que é o *Database Toolbox* [33].

CAPÍTULO VIII

CONCLUSÕES

Os sistemas tradicionalmente desenvolvidos, que utilizavam-se de computadores, embora tivessem a preocupação em oferecer facilidades aos seus usuários, muitas vezes usavam mecanismos inadequados para tal. Dessa forma, quase que apenas os usuários inais experientes ou com base mais técnica eram capazes de manipulá-los. Muito investimento em treinamento de usuários era necessário. Um exemplo de dificuldades normalmente encontradas seria no acesso a Bancos de Dados.

Ao longo do tempo tem havido uma crescente necessidade de se desenvolver sistemas para todas as classes de usuários, ou seja, dos inais especialistas aos inais inexperientes. Objetivando uma maior proximidade entre os sistemas e esses usuários não-técnicos, passou-se a estudar, mais profundamente, formas que permitissem esse entrosamento. Surgiram, então, sistemas desenvolvidos utilizando interfaces gráficas que tornam o software inais amigável para o usuário. Essas interfaces são bastante importantes quando tratamos de acesso a bancos de dados.

Como pudemos observar neste trabalho, a preocupação aqui é exatamente nesta direção, ou seja, manipulação de dados em uma base. Através do *software* QUICK-DB desenvolvido no Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro, que é um ambiente integrado de acesso a bancos de dados através de um esquema na modelagem E-R, procuramos contribuir com mais essa alternativa para solucionar tais problemas. Nesse sentido, desenvolvemos, dentro daquele ambiente, um Módulo de Consultas que tem como função permitir que o usuário, através de interfaces gráficas voltadas para o Modelo E-R, tenha acesso aos dados do BD. Com o auxílio de menus e uso de mouse, a partir do diagrama E-R, através do qual o BD é gerado, podemos executar consultas *ad hoc* à base de dados. O Módulo de Consultas encontra-se implementado em microcomputador PC-compatível com programação Pascal. A base da implementação é o *software* QUICK-DB já funcionando, acrescido de novas *units* do Pascal e com poucas alterações sobre as que

já existiam. A implementação está praticamente completa, necessitando somente de alguns ajustes finais.

Com relação à abrangência deste trabalho, especificamente, o usuário, para formular suas consultas ao banco de dados, terá à sua disposição dois conceitos de consulta que são o *folheamento* e a *navegação*. O primeiro permite que ele obtenha informações do BD através do instanciamento de suas estruturas. O segundo, por sua vez, pode ser aplicado de duas formas. A primeira forma corresponde ao acesso a uma entidade ou relacionamento, onde o usuário pode navegar por suas instâncias, através de uma ordem previamente definida. A segunda forma de navegação permite que o usuário determine um caminho no E-R composto de entidades e relacionamentos onde suas instâncias podem também ser folheadas.

Para ter acesso às instâncias das estruturas, duas formas foram adotadas neste módulo: chaves e ordens. Chaves e ordens são formadas por um conjunto de um ou mais atributos (formam uma chave). As chaves determinam um acesso único a uma instância enquanto que nas ordens pode haver duplicação das chaves de acesso. As ordens podem ser totais ou parciais. As totais determinam uma ordenação de todas as instâncias existentes na entidade ou relacionamento onde elas se aplicam. Já a ordem parcial é aquela que faz com que cada grupo de instâncias de uma entidade, associadas a uma instância de outra entidade, esteja ordenado.

Orientação a Objetos foi, também, usada no desenvolvimento de métodos que compõem bibliotecas de funções. Refinamentos a esses métodos foram propostos. Alguns deles já se encontram implantados, fazendo parte do programa que gera um dos módulos do QUICK-DB, o Servidor de Bancos de Dados. Os demais foram implementados somente dentro do Módulo de Consultas.

O Módulo de Consultas está implementado em microcomputador PC-compatível utilizando a linguagem Turbo Pascal versão 5.5, onde há um módulo do *software* com orientação a objetos. A programação da interface gráfica foi feita utilizando o Turbo Graphix. O acesso ao BD, por sua vez, utiliza o Turbo Database Toolbox.

Algumas dificuldades foram encontradas quando de sua implementação. A principal está no fato do equipamento utilizado não possuir uma configuração muito adequada ao desenvolvimento e execução deste *software*. A interface gráfica não possui uma boa resolução já que a placa usada é a CGA. Foi necessário, ainda, o uso de *overlay* já que a capacidade de memória do micro utilizado, de 640 Kbytes, estava estourando. Um passo a ser dado brevemente diz respeito à conversão da versão atual do *software* para a versão 6.0 do Turbo Pascal. Isto, muito embora já tenha sido tentado, não foi ainda possível devido às mudanças sofridas pelo Turbo Pascal, exatamente no que diz respeito ao uso de *overlays*.

Revisão Bibliográfica

Com relação à revisão bibliográfica mostrada no capítulo II deste trabalho a respeito dos artigos desenvolvidos nesta área de aplicação, podemos concluir que:

- Um aspecto que consideramos muito importante no Módulo de Consultas do Quick-Db está no fato dele lidar com somente um modelo de dados, o MER, a nível conceitual e na implementação da base de dados;
- O que pudemos verificar é que *navegação* e *folheamento* são, muitas vezes, tratados de forma pouco padronizada nestes trabalhos;
- A interface de consultas do Quick-Db oferece ao usuário ambas as facilidades: *folheamento* e *navegação*.

O que pudemos notar nos artigos descritos é que os conceitos de *folheamento* e *navegação* se confundem. Entre eles não encontramos nenhum que preveja ambas as facilidades. O que os autores ainda procuram, de um modo geral, são formas que sejam as melhores. Assim, também o trabalho que apresentamos aqui é uma contribuição a essa busca.

Contribuição

A contribuição que este trabalho vem fazer em sua área de aplicação pode ser descrita da seguinte forma:

- Foi realizado um estudo sobre interfaces gráficas de consultas;
- É proposto um refinamento ao acesso a bancos de dados. Este acesso é realizado através da utilização de métodos desenvolvidos com Orientação a Objetos;
- Foi realizada a implementação de um módulo de consultas a bancos de dados com base no Modelo E-R utilizando interface gráfica; e
- Duas formas de acesso a bancos de dados através de interface gráfica E-R foram implementadas permitindo a navegação pelo esquema. São as chaves e as ordens.

Extensões

Algumas extensões a este trabalho são propostas:

- Refinamento dos métodos de instanciação, facilitando o acesso à base de dados;
- Formulação de expressões mais completas para consultas. Para que possamos implementar formas mais completas de consultas, uma linguagem simplificada deverá ser desenvolvida permitindo o uso de condições de seleção como desigualdades, entre outros complicadores;
- Introdução de manipulação de fragmentos de consultas, de modo a permitir que se faça diferenças de conjuntos e união como permitem as operações da Álgebra Relacional, respectivamente, DIFERENÇA e UNIÃO; e
- Especificação de visões da base de dados pelos métodos.

REFERÊNCIAS BIBLIOGRÁFICAS

1. CHAMBERLIN, D.D. Relational Data-Base Management Systems. *ACM Computing Surveys*. New York, v.8, n.1, p.43-66, March 1976.
2. CHEN, P. The Entity-relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems*. New York, v.1, n.1, p.9-36, March 1976.
3. CODASYL DATA BASE TASK GROUP REPORT. New York: ACM, 1971. IMPR. 1975. 296p.
4. CODD, E.F. *Relational Completeness of Data Base Sublanguages, in Database Systems*. Englewood Cliffs: Prentice-Hall, 1972. (Courant Computer Science Symposia Series, v.6)
5. CZEJDO, B. et al. A Graphical Data Manipulation Language for an Extended Entity-Relationship Model. *IEEE Computer*. New York, v.23, n.3, p.26-36, March 1990.
6. DATE, C.J. *An Introduction to Database Systems*. 4 ed. Reading: Addison-Wesley, 1986. (Systems Programming Series, v.1)
7. DATE, C.J. *Guide to SQL Standard*. Reading: Addison-Wesley, 1987. IMPR. 1988. 205p.
8. DELVAUX, M. *Geração de Rotinas Orientadas a Objetos para Manipulação de um Banco de Dados*. Instituto de Matemática: UFRJ, 1991. Projeto Final de Curso (em andamento)
9. FDT Bulletin of ACM - SigMod. ANSI/X3/SPARC - Study Group on Data Base Management Systems. New York, v.7, 11.2, 1975. 140p. Interim Report.

10. **FOGG, D.** Lessons from a "Living in a Database" Graphical Query Interface. *ACM SIGMod Record*. New York, v.14, 11.2, p.100-106, 1984.
11. **KLUG, A.** Equivalence of Relational Algebra and Relational Calculus Query Languages. *Journal of the ACM*. New York, v.29, n.3, p.699-717, July 1982.
12. **LENZERINI, M. ; SANTUCCI, G.** Cardinality Constraints in the Entity-Relationship Model. In: INTERNATIONAL CONFERENCE ON ENTITY-RELATIONSHIP APPROACH, 3. 1983. California. *Proceedings of the Third International Conference on Entity-Relationship Approach* s.n.t.
13. **LOBEL, L.** *Especificação e Implementação de uma Linguagem de Consultas baseada no Modelo de Entidades e Relacionamentos.* Departamento de Informática: PUC/RJ, Abril 1981. (Dissertação de Mestrado)
14. **PACITTI, E.C. ; SILVEIRA, P.M.** *QUICK-SHELL: Especificação e Implementação de uma Shell para Sistemas Especialistas que Acessa Bancos de Dados,* submetido ao VI Simpósio Brasileiro de Bancos de Dados, 1991, Manaus.
15. **PARIMALA, N.** et al. A Query Facility to a Network DBMS. *The Computer Journal*. New York: The British Computer Society. v.32, n.1, p.55-62, February 1989.
16. **PECKHAM, M. ; MARYANSKI, F.** Semantic Data Models. *ACM Computing Surveys*. New York, v.20, 11.3, p.153-189, September 1988.
17. **RIBEIRO, C.D.** *Uma Proposta de Implementação de um Gerador de Transações voltado para o Modelo E-R.* COPPE/Sistemas: UFRJ. (Tese em andamento)
18. **ROGERS, T.R. ; CATTELL, R.G.G.** Entity-Relationship Database User Interfaces. In: INTERNATIONAL CONFERENCE OF ENTITY-RELATION-

- SHIP APPROACH, 6. 1987. New York. *Proceedings of the 6th International Conference of Entity-Relationship Approach*. New York: s. ecl. 1987. p.322-335.
19. SCHNEIDER, M. ; TREPIED, C.A. *A Graphical Query Language Based on an Extended E-R Model*. Laboratoire Informatique: Université de Clermont-Ferrand II, France, s.d.
20. SETZER, V.W. *Projeto Lógico e Projeto Físico de Banco de Dados*. Belo Horizonte: V Escola de Computação, 1986.
21. SILVA, M.F. ; TOSO, V.E.F. *Gerador de Bancos de Dados do Quick-Db*. Instituto de Matemática: UFRJ, 1990.
22. SILVEIRA, P.M. *Projeto de Banco de Dados Auxiliado por Computador*. In: CONGRESSO NACIONAL DE INFORMÁTICA, 21. 1988. Rio de Janeiro. *Anais do XXI Congresso Nacional de Informática*. Rio de Janeiro: s.ed., s.d. p.195-198.
23. SILVEIRA, P.M. *A Formalization of the E-R Model*. In: CONFERÊNCIA INTERNACIONAL DE LA SOCIEDAD CHILENA DE CIENCIA DE LA COMPUTACION, 9. 1989. Santiago. *Anais da IX Conferência Internacional de la Sociedad Chilena de Ciencia de la Computacion*. s.n.t.
24. SILVEIRA, P.M. *Procedural Data Manipulation Operations for the E-R Model*. In: CONFERÊNCIA LATINO-AMERICANA DE INFORMÁTICA, 16. 1990. Assunção, Paraguai. *Anais da XVI Conferência Latino-Americana de Informática*. s.n.t.
25. SILVEIRA, P.M. *Definindo e Utilizando Bancos de Dados com o Modelo Entidade-Relacionamento*. In: CONGRESSO NACIONAL DE INFORMÁTICA, 23. 1990. Rio de Janeiro. *Anais do XXIII Congresso Nacional de Informática*. s.n.t.

26. SILVEIRA, P.M. *An Object-Oriented Shell for Entity-Relationship Databases*. (Será publicado como Relatório Técnico NCE/UFRJ).
27. TAYLOR, R.W. ; FRANK, R.L. *Coclasyl Data-Base Management Systems*. *ACM Computing Surveys*. New York, v.8, n.1, p.67-103, March 1976.
28. TSICHRITZIS, D.C.; LOCHOVSKY, F.H. *Hierarchical Data-Base Management: a survey*. *ACM Computing Surveys*. New York, v.8, n.1, p.105-123, March 1976.
29. TSICHRITZIS, D.C.; LOCHOVSKY, F.H. *Data Base Management Systems*. New York: Academic Press, 388p., 1977. (Computer Science and Applied Mathematics Series).
30. **TURBO PASCAL GRAPHIX TOOLBOX OWNER'S HANDBOOK**. Borland International Inc., July 1985.
31. **TURBO PASCAL OBJECT-ORIENTED PROGRAMMING GUIDE**. MICRO; Borland, 5.5 Scotts Valley, CA, Borland, 24/05/1990.
32. **TURBO PASCAL USER'S GUIDE**. MICRO; Borland 5.0, Scotts Valley, CA, Borland, 30/05/1990.
33. **TURBO PASCAL DATABASE TOOLBOX**. MICRO; Borland 4.0, Scotts Valley, CA, IBM, 15/06/1990.
34. ULLMAN, J.D. *Principles of Database Systems*, s.l.: Computer Science Press, 1980.
35. WONG, H.K.T. ; KUO,I. **GUIDE: Graphical User Interface for Database Exploration**. In: CONFERENCE ON VERY LARGE DATA BASES, 8. 1982. Mexico City. *Proceedings of the Eighth Conference on Very Large Data Bases*, s.n.t.

36. **ZHANG, Z.-Q. ; MENDELZON, A.O.** A Graphical Query Language for Entity-Relationship Databases. In: INTERNATIONAL CONFERENCE OF ENTITY-RELATIONSHIP APPROACH, 3. 1983. Anaheim, California. *Proceedings of the 3rd International Conference of Entity-Relationship Approach*, s.n.t. p.441-448.

37. **ZLOOF, M.M.** Query by Example. In: NATIONAL COMPUTER CONFERENCE. 1975. Anaheim, California. *AFIPS Conference Proceedings 1975, National Computer Conference*, s.l.: AFIPS, 1975. p.431-438. (AFIPS Conference Proceedings; v.44)