



Estruturação e Modelagem de Bancos de Dados

Jugurta Lisboa Filho

Universidade Federal de Viçosa
Departamento de Informática
36570-000 Viçosa/MG – Brasil
E-mail: jugurta@dpi.ufv.br

Curitiba-PR, maio de 2001

SUMÁRIO

1	CONCEITOS EM SISTEMAS DE BANCO DE DADOS	1
1.1	Banco de Dados (BD)	1
1.2	Sistema de Gerenciamento de Banco de Dados (SGBD)	1
1.3	Programa de Aplicação	2
1.4	Sistema de Banco de Dados (SBD)	2
1.5	Fases do Projeto de Banco de Dados	4
1.6	Modelos de Dados	5
2	NOÇÕES BÁSICAS DO MODELO RELACIONAL	6
2.1	Modelo Relacional	6
2.2	Álgebra Relacional	7
2.3	SQL – Linguagem de Consulta Estruturada	10
3	PROJETO DE BANCO DE DADOS	13
3.1	Processo de Modelagem Conceitual	13
3.2	Modelo Entidade-Relacionamento (E-R)	14
3.3	Projeto Lógico de Banco de Dados	16
3.4	Ferramentas CASE	20
4	PROJETO DE BANCOS DE DADOS GEOGRÁFICOS	22
4.1	Dado, Informação e Fenômeno Geográfico	22
4.2	Requisitos de Modelagem para Aplicações de SIG	24
4.3	Modelando Banco de Dados Geográficos com UML-GeoFrame	27
4.4	Projeto Lógico de Banco de Dados Geográficos	33
4.5	Estudo de Caso	36
5	BIBLIOGRAFIA	39

1 CONCEITOS EM SISTEMAS DE BANCO DE DADOS

O componente de armazenamento de um Sistema de Informação Geográfica (SIG), denominado sistema de banco de dados geográficos, estrutura e armazena os dados de forma a possibilitar a realização das operações de análise e consulta. Devido à complexidade das aplicações que são desenvolvidas a partir de um SIG, projetar o banco de dados geográficos tem sido um dos grandes desafios para as organizações usuárias desses sistemas.

O desenvolvimento de aplicações de SIG tem sido feito, normalmente, de forma não metodológica, tendo como resultado diversos problemas decorrentes de abordagens evolutivas desordenadas.

O projeto de um banco de dados deve ser realizado com o apoio de um modelo de dados de alto nível, também conhecido como modelo conceitual. Durante vários anos, as pesquisas no campo dos modelos de dados para SIG centraram-se na busca por estruturas de dados para o armazenamento de dados georreferenciados, o que ficou conhecido como “debate *raster-vector*” [COU 92]. Atualmente, sabe-se que para o projeto de banco de dados geográficos são necessários novos modelos conceituais. Essa necessidade foi identificada já no final da década de 80. Desde então, diversos modelos específicos para modelagem de dados em aplicações de SIG têm sido propostos na literatura.

O objetivo deste curso é apresentar uma metodologia atual para estruturação e modelagem conceitual de dados geográficos, utilizando técnicas bem consolidadas de projeto de banco de dados (ex.: Modelo E-R) e, também, técnicas mais modernas, como a linguagem UML. A fim de fornecer uma base conceitual sobre Sistemas de Bancos de Dados, neste capítulo são apresentados diversos conceitos relacionados a Sistema de Gerenciamento de Banco de Dados (SGBD) relacional.

Esta apostila é uma nova edição da apostila utilizada nos cursos “Banco de Dados para GIS”, apresentado no GISBrasil’99 e “Estruturação e Modelagem de Banco de Dados”, apresentado no GIS Brasil’2000.

1.1 Banco de Dados (BD)

Por banco de dados entende-se a coleção dos dados propriamente dita. Algumas definições de banco de dados encontrada na literatura incluem: “Um banco de dados é uma coleção de dados relacionados” [ELM 00]; “Um banco de dados é uma coleção de dados operacionais armazenados, sendo usados pelos sistemas de aplicação de uma determinada organização” [DAT 81].

Um banco de dados pode ser mantido manualmente ou por computador. Por exemplo, imagine um banco de dados composto de fichas de publicações em uma biblioteca. Um banco de dados sempre é povoado com dados para um propósito específico, ou seja, ele contém informações que circulam pela aplicação e elementos da própria aplicação. Numa aplicação de distribuição de energia elétrica, por exemplo, o banco de dados armazena a descrição de cada consumidor mas também contém dados sobre o consumo mensal de cada consumidor.

Os dados armazenados em um banco de dados devem respeitar as regras da aplicação, definidas pelas restrições de integridade. Por exemplo, se existe uma restrição do tipo “o consumo de energia elétrica não pode ser negativo”, não pode haver nenhum registro de consumo de energia cujo valor seja negativo.

1.2 Sistema de Gerenciamento de Banco de Dados (SGBD)

Um Sistema de Gerenciamento de Banco de Dados constitui-se de uma coleção de programas que permite criar e manter o BD. Assim, um SGBD é um software de propósito geral que facilita o processo de definição, construção e manipulação do BD. Por software de propósito geral entende-se que o SGBD é usado não como uma ferramenta final, mas como um software através do qual os programas de aplicação (descritos a seguir) são desenvolvidos.

As principais características de um SGBD incluem o suporte a:

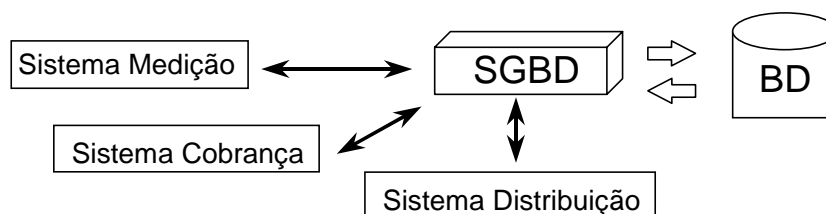
- ❑ Definição do BD - especificação e descrição detalhada dos tipos, estruturas e restrições dos dados a serem armazenados no BD;
- ❑ Construção do BD - processo de carga inicial dos dados em um meio de armazenamento controlado pelo SGBD;
- ❑ Manipulação do BD - abrange as alterações realizadas nos dados para refletir mudanças ocorridas no ambiente. Isto inclui as operações de inclusão e exclusão de dados;
- ❑ Consulta aos dados - tipo de operação mais comum realizada por usuários que necessitam extrair informações armazenadas no BD.

1.3 Programa de Aplicação

Os programas de aplicação são sistemas desenvolvidos para um propósito específico (ex.: sistema de estoque, sistema de vendas, folha de pagamento). Todo o acesso e manipulação de dados realizado pelo programa de aplicação é feito por intermédio do SGBD. Assim, o programa de aplicação realiza as funções da aplicação, enquanto o SGBD serve de base para o gerenciamento dos dados manipulados pelo programa de aplicação.

Um exemplo de programa de aplicação é um sistema de controle de consumo de energia elétrica. Uma de suas funções seria executar o cálculo do consumo mensal de energia elétrica, a partir dos dados de leitura dos medidores. O algoritmo que descreve a política de cobrança de acordo com a quantidade consumida e o tipo de consumidor é implementado pelo programa de aplicação, enquanto o acesso aos dados referentes ao consumo de eletricidade e aos consumidores é feito através do SGBD. O programa de aplicação garante restrições de integridade que não podem ser controladas pelo SGBD.

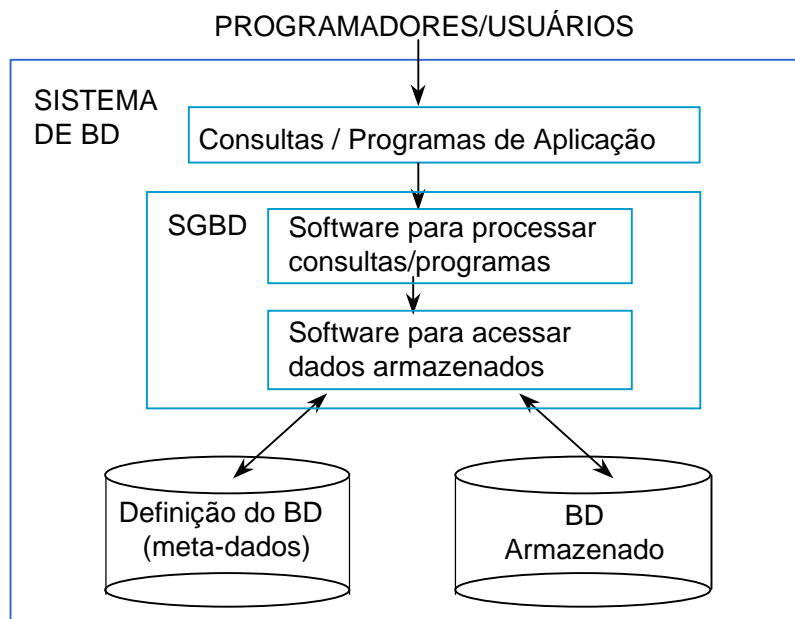
O programa de aplicação implementa interface e relatórios específicos como, por exemplo, emissão de extrato de consumo, enquanto o SGBD pode ser usado para possibilitar ao usuário elaborar consultas eventuais (*ad hoc*), não previstas / fornecidas pelo programa de aplicação. A figura abaixo ilustra que o acesso ao BD, pelos programas de aplicação, é feito através do SGBD.



1.4 Sistema de Banco de Dados (SBD)

A existência de bancos de dados nas organizações é uma coisa muito comum hoje em dia, ou seja, as pessoas já estão habituadas a usarem diversos termos relacionados a banco de dados de forma bastante informal. É comum se ouvir no noticiário que um determinado órgão está elaborando a construção de um novo banco de dados. No entanto, para os profissionais que estão envolvidos com o desenvolvimento de sistemas computacionais é importante ter a compreensão dos diferentes termos utilizados a este respeito.

Assim, Sistema de Banco de Dados (figura a seguir) é um sistema composto pelos programas de aplicação, pelo SGBD e pelo BD, para um determinado conjunto de aplicações.



Muitos programas de aplicação são construídos sem o uso de SGBD para gerenciar os dados. Neste caso, o próprio programa de aplicação é responsável por implementar as rotinas que realizam o armazenamento e recuperação dos seus dados.

A seguir estão listadas uma série de vantagens e a desvantagem do uso de SGBD no desenvolvimento de programas de aplicação:

Vantagens

- ❑ Diminui a redundância dos dados - Em uma organização que possui diversos programas de aplicação é comum o compartilhamento de dados entre os diversos sistemas. Quando o número de sistemas aumenta, ou também, quando cresce a complexidade destes sistemas, é comum que alguns dados sejam repetidos a fim de simplificação da implementação e da manutenção dos sistemas. Como consequência, surgem as redundâncias, o que são sérias fontes de dados inconsistentes nas organizações.
- ❑ Padroniza a definição e o acesso aos dados - O SGBD fornece linguagens de definição e acesso aos dados que devem ser utilizadas por todos os programas de aplicação, criando-se assim, uma interface única de acesso aos dados armazenados no BD.
- ❑ Assegura independência física e lógica dos dados - Uma das principais vantagens do uso de SGBD é exatamente a liberdade que o administrador do BD tem para promover alterações nas estruturas dos dados sem que estas mudanças provoquem uma série de manutenções em cascata. Independência física significa que alterações podem ser feitas no nível físico do BD sem provocar mudanças no nível lógico. Por exemplo, pode-se mudar o local onde está armazenado o BD (troca de servidor). Independência lógica significa que alterações podem ser feitas no nível lógico do BD sem provocar alterações nos programas de aplicação que utilizam as estruturas do BD modificadas. Por exemplo, pode-se incluir um novo campo em uma tabela sem ter de recompilar todos os programas que acessam aquela tabela.
- ❑ Consistência dos dados em caso de falhas - O SGBD deve dispor de dispositivos que tragam o BD para um estado consistente mesmo após a ocorrência de falhas como, por exemplo, falta de luz. Sistemas de back-ups devem ser providos para que o SBD fique imune também a falhas de hardware.
- ❑ Manutenção de restrições de integridade da aplicação - Muitas verificações que são realizadas durante a obtenção de dados pelos programas de aplicação podem ser transferidas para o SGBD. Por exemplo, a verificação de valores fornecidos em uma data são validados com

relação a se ter um mês com valores entre 1 e 12, o mesmo ocorrendo com o valor do dia (que depende do mês).

- ❑ Segurança dos dados em ambiente multi-usuário - Controle de acesso e de transações no caso de multi acesso a um mesmo elemento de dado pode ser feito de forma confiável pelo SGBD.

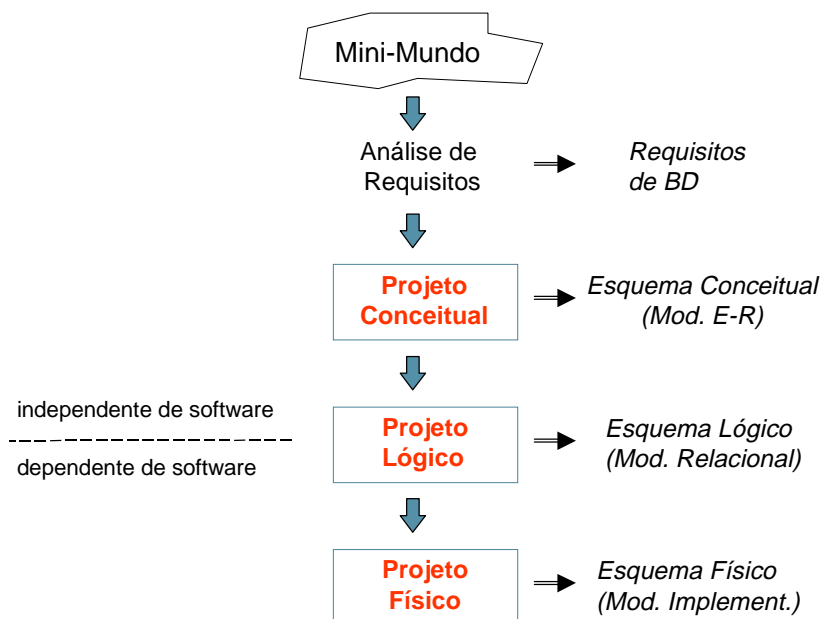
Desvantagem

A grande desvantagem do uso de um SGBD está exatamente na existência de um overhead para prover segurança, controle de concorrência, recuperação e funções de integridade.

1.5 Fases do Projeto de Banco de Dados

O processo de desenvolvimento de um banco de dados está intimamente relacionado com o ciclo de vida do desenvolvimento de software onde, a cada etapa, novas informações e detalhes são acrescentados ao projeto do software [PRE 87]. No projeto de banco de dados as informações que comporão o banco de dados são especificadas utilizando-se modelos de dados em diferentes níveis de abstração, iniciando por modelos de alto nível de abstração e refinando-se o modelo até que sejam incorporados detalhes específicos, relacionados ao armazenamento dos dados.

Segundo Elmasri [ELM 00], o projeto de banco de dados deve ser dividido em três etapas, nas quais são empregados diferentes tipos de modelos. A figura a seguir ilustra as três fases de projeto de banco de dados (*conceitual, lógico e físico*) e seus respectivos produtos (*esquema conceitual, esquema lógico e esquema físico*).



- ❑ Projeto Conceitual – Nesta fase é elaborado o esquema conceitual do banco de dados, onde são utilizados modelos semânticos que empregam construtores de abstração de alto nível para descrever os requisitos de dados das aplicações (ex.: modelo E-R [CHE 76]). Normalmente, são utilizadas linguagens (modelos) bastante simples, que facilitam a comunicação e o entendimento entre usuários e projetistas. São identificadas e definidas quais as entidades que serão representadas no banco de dados, suas estruturas (atributos) e os relacionamentos existentes entre elas. No projeto conceitual não são considerados aspectos sobre o sistema de computação (software/hardware) que será utilizado.
- ❑ Projeto Lógico – Nesta fase é elaborado o esquema lógico do banco de dados com base no tipo de modelo de SGBD que será utilizado. O esquema lógico independe do software a ser usado, mas é dependente de um modelo de dados. São gerados a partir da aplicação de

regras de transformação (mapeamento) dos construtores de abstração utilizados no esquema conceitual em elementos de representação de dados de um dos modelos de banco de dados (ex.: relacional, hierárquico, orientado a objetos, objeto-relacional).

- **Projeto Físico** – Define os aspectos de implementação física do banco de dados como, por exemplo, estruturas de armazenamento, caminhos de acesso, particionamento e agrupamento. Estão diretamente relacionados a um SGBD específico e permitem, ao projetista, planejar aspectos ligados à eficiência do sistema de banco de dados.

1.6 Modelos de Dados

Para Brodie [BRO 84], um modelo de dados é uma coleção de conceitos bem definidos matematicamente, que nos auxilia a pensar e expressar as propriedades estáticas e dinâmicas das aplicações. Numa visão mais específica de banco de dados, um modelo de dados consiste de três componentes: uma coleção de tipos de objetos, uma coleção de operadores e uma coleção de regras de integridade [COD 79].

Como exemplo dos principais modelos de banco de dados pode-se citar os modelos clássicos: Rede, Hierárquico e Relacional. Este último é o modelo em uso na maioria dos SGBD existentes, estando instalados em 90% das organizações atualmente [DBM 94].

Com a evolução do hardware e do software, novos modelos foram desenvolvidos para atender, principalmente, a uma série de aplicações ditas não convencionais (ex.: CAD, GIS, Automação de Escritório). Estes novos modelos são classificados como Modelos Orientados a Objetos ou Modelos Objeto-Relacional.

Modelos Semânticos

Os modelos semânticos fornecem um conjunto de conceitos (construtores), que nos auxiliam a pensar e expressar as propriedades estáticas e dinâmicas das aplicações. A característica básica de um modelo, como o próprio termo explicita, é que ele é uma abstração da realidade [PEU 84]. A distância existente entre a maneira na qual as entidades existem na realidade e a maneira como estas entidades são representadas internamente nos computadores levaram ao surgimento de modelos de dados em diferentes níveis de abstração. No nível lógico estão os modelos citados anteriormente (ex.: relacional, objeto-relacional), os quais são implementados por SGBD, incluindo suas linguagens para definição e manipulação dos dados. No nível conceitual, são empregados os modelos semânticos, os quais são voltados para o entendimento por parte das pessoas (usuários e projetistas), embora possam ser processados por ferramentas CASE.

Ullman [ULL 82] define um modelo de dados (semântico) como uma descrição geral de um conjunto específico de entidades e os relacionamentos existentes entre estes conjuntos de entidades. Uma entidade pode ser qualquer coisa que seja distinguível (ex.: um conceito, um evento, um objeto). Um conjunto de entidades agrupa entidades que possuem características em comum (ex.: Rios, Cidades e Pessoas). Características de entidades são descritas por meio de atributos que possuem valores específicos de um determinado domínio (ex.: nome e população de cidades). O exemplo mais conhecido de modelo semântico é o modelo Entidade-Relacionamento (E-R), descrito em detalhes no capítulo 3.

Um modelo de dados semântico fornece uma base formal (notacional e semântica) para ferramentas e técnicas usadas para suportar a modelagem semântica do BD. Modelagem semântica é o processo de abstração no qual, somente os elementos essenciais da realidade observada são enfatizados, descartando-se os elementos não essenciais

2 NOÇÕES BÁSICAS DO MODELO RELACIONAL

Neste capítulo é feita uma descrição resumida do modelo relacional, que é o modelo utilizado na grande maioria dos SGBD que são integrados com Sistemas de Informação Geográfica (SIG), constituindo os chamados modelos de arquitetura dual. Exemplos de SIG que implementam a arquitetura dual incluem o ARC/INFO, MGE, Idrisi e Spring. Nestes sistemas, normalmente o SGBD relacional é usado para armazenar os dados descritivos, enquanto os dados geométricos são armazenados por meio de estruturas de dados mantidas em arquivos. Algumas estruturas de dados geométricos (ex.: topologia) também são armazenadas no SGBD relacional.

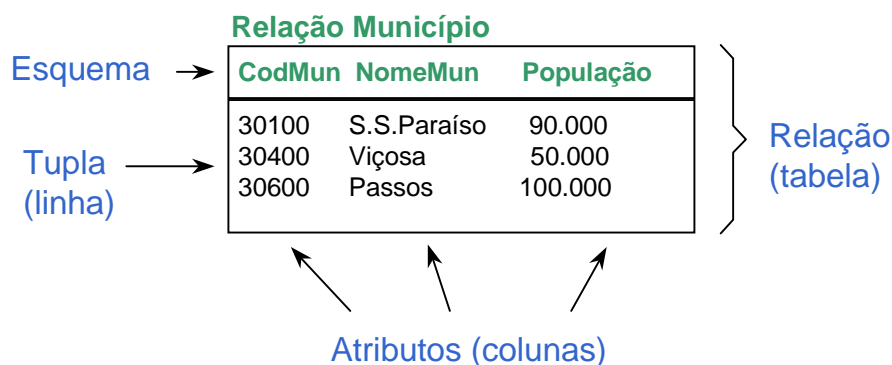
Existem SIG que são construídos com base em outras arquiteturas diferentes da arquitetura dual. Os sistemas SIRO-DBMS e SYSTEM 9, por exemplo, são baseados em sistemas com suporte a campos longos, enquanto os sistemas SmallWorld e GEO₂ são construídos com base em modelos orientados a objetos.

2.1 Modelo Relacional

O modelo relacional foi proposto por Ted Codd em 1970. Fundamentado em uma forte base matemática, possibilitou a definição precisa de operadores de consulta aos dados.

No modelo relacional, o BD é composto por uma coleção de tabelas. Um único tipo construtor, que é a tabela (ou relação), é usado para estruturar todos os dados do BD. O modelo fornece um conjunto de linguagens para definição (DDL – *Data Definition Language*), para manipulação (DML – *Data Manipulation Language*), e para consulta ao BD (QL – *Query Language*). A linguagem de consulta SQL - *Sequential Query Language*, a qual engloba as funções das demais linguagens é a linguagem de consulta padrão para SGBD relacionais.

A figura abaixo ilustra os principais elementos do modelo relacional. Uma relação compreende toda a tabela incluindo sua identificação (ex.: Município), a definição das colunas (ou atributos da relação) e os dados (tuplas ou linhas da relação).



Outros conceitos empregados no modelo relacional incluem:

- ❑ Esquema do BD - Descrição textual (ou gráfica) dos dados e suas estruturas. Compreende a definição dos esquemas de todas as tabelas do BD. Não altera com frequência.
- ❑ Instâncias do BD - Conjunto de dados armazenados no BD em um determinado instante de tempo, que respeita a definição de um esquema de BD. Pode alterar com frequência.
- ❑ Estado do BD - Conjunto de dados armazenados em um BD em um determinado instante. Uma transação no BD deve levar o BD de um estado consistente a outro estado consistente. Caso contrário esta transação deve ser desfeita.

Em uma relação, cada coluna só pode conter valores de um determinado tipo. A definição do tipo de valor que pode ser armazenado em uma coluna é feita através da especificação de um domínio. Assim, tem-se:

- Domínio de atributo - Conjunto de valores possíveis de um atributo. Exemplos de domínios incluem os valores inteiros, reais, cadeias de caracteres, datas e valores lógicos ou *booleanos* (verdadeiro/falso).

Um atributo pode ser usado para identificar uma determinada tupla (linha) em uma relação. Os conceitos abaixo descrevem os diversos tipos de atributos identificadores de uma relação:

- Atributo chave - Atributo que possibilita a identificação de uma tupla na relação.
- Chave primária - Atributo chave que é o identificador principal de uma relação. Pode ser composta de mais de um atributo. Isto ocorre quando um único atributo não é suficiente para identificar uma tupla na relação.
- Chave estrangeira - Atributo em uma relação cujo valor aparece como chave primária em uma outra relação.

É importante observar que o acesso a uma relação não necessita ser feito através de chaves, ou seja, as tuplas de uma relação podem ser acessadas, por exemplo, de forma seqüencial. Atributos chave servem para situações em que o usuário deseja acessar uma tupla específica no conjunto total das tuplas. Um exemplo é o acesso ao cadastro de contribuintes da Receita Federal a partir do CPF do contribuinte. Neste caso, CPF deve ser um atributo chave. A chave é uma propriedade do esquema da relação.

- Restrições de integridade - São imposições que devem ser satisfeitas pelo SGBD para uma determinada relação. Restrições são especificadas no esquema e devem ser mantidas em todas as instâncias deste esquema.

Existem diversos tipos de restrições, os principais são:

- Restrições de domínio - O valor de cada atributo deve ser atômico e pertencer ao domínio especificado.
- Restrições de chave - Todas as tuplas de uma relação devem ser distintas.
- Integridade referencial - O valor de uma chave estrangeira só pode ser nulo ou igual a algum valor da chave primária na relação correspondente.

Além dos conceitos referentes ao esquema do BD, um segundo aspecto importante do modelo relacional é o tipo de linguagem fornecida. Conforme descrito anteriormente, o modelo relacional provê linguagens para definição, manipulação e consulta ao BD. A possibilidade do usuário realizar consultas ao BD que não estavam previstas inicialmente no programa de aplicação, deu ao usuário um grande poder de uso, uma vez que ele agora independe do pessoal técnico para implementar um novo relatório. A partir de um conhecimento básico de uma linguagem de consulta, o próprio usuário pode explorar diferentes visões dos dados armazenados.

A linguagem de consulta no modelo relacional é a Linguagem SQL (*Structured Query Language*), a qual é baseada em um padrão definido internacionalmente pela ANSI. As seções seguintes apresentam uma breve introdução às linguagens Álgebra Relacional e SQL.

2.2 Álgebra Relacional

A álgebra é uma linguagem de consulta procedural, consistindo de um conjunto de operações que permitem a extração de dados do BD. As operações podem ser aplicadas a uma única relação (unária) ou a duas relações (binária), tendo como resultado sempre uma nova relação.

As operações na álgebra relacional podem ser classificadas em: operações específicas para SGBD relacionais (seleção, junção, projeção, divisão, renomear e atribuição); e operações da teoria de conjuntos (união, interseção, diferença e produto cartesiano). O objetivo da apresentação de operações da álgebra neste curso é apenas o de facilitar o entendimento das operações da linguagem SQL. Assim, apenas três operações serão descritas (seleção, projeção e produto cartesiano). Leitores interessados no assunto podem recorrer a vasta bibliografia de SGBD relacionais como, por exemplo, [ELM 00].

O seguinte exemplo de BD será usado para exemplificar o uso de operações da álgebra relacional.

PROPRIETÁRIO

idProp	nomeProp	ender	telProp
10	José Carioca	Av. N.S. Copa,123	99-884466
20	Francisco Bento	Rua Curva, 47	99-125544
30	Mônica Aparecida	Av. da Paz, 51	98-337768

FAZENDA

idFaz	nomeFaz	áreaFaz	dono
7663	Sta Rosa	300	10
4355	Rancho Fundo	350	10
8998	Cavalo Gordo	253	10
7644	Cavalo Magro	500	20
3388	Dolar Furado	640	20
7676	Vila Cristina	150	30
8444	Alba Navarra	321	30

Seleção (σ)

Seleciona um conjunto de tuplas (linhas) que satisfaçam uma dada condição. A relação resultante terá os mesmos atributos da relação original, ou seja, o mesmo esquema.

A sintaxe da operação de seleção é a seguinte:

$$\sigma_{\text{condição}}(\text{RELAÇÃO})$$

Onde a cláusula de condição pode conter os seguintes operadores: <, >, ≤, ≥, =, ≠, ∨ (ou), ∧ (e), (−) não. A consulta "Selecione fazendas com mais de 300 ha" deve ser expressa da seguinte forma:

$$\sigma_{\text{área} > 300}(\text{FAZENDA})$$

O resultado está apresentado na seguinte tabela:

TABELA RESULTANTE

idFaz	nomeFaz	áreaFaz	dono
4355	Rancho Fundo	350	10
7644	Cavalo Magro	500	20
3388	Dolar Furado	640	20
8444	Alba Navarra	321	30

Projeção (Π)

A operação de projeção permite ao usuário selecionar um conjunto de atributos (colunas) em uma relação. A relação resultante tem apenas os atributos projetados, sendo que eventuais tuplas duplicadas são eliminadas.

A sintaxe da operação de projeção é a seguinte:

$$\Pi_{\text{lista de atributos}} (\text{RELAÇÃO})$$

Onde a lista de atributos indica as colunas a serem projetadas. A consulta "Crie uma tabela com o nome e o telefone dos proprietários" deve ser expressa como:

$$\Pi_{\text{nomeProp, telProp}} (\text{PROPRIETÁRIO})$$

Gerando a seguinte tabela resultante:

TABELA RESULTANTE

nomeProp	telProp
José Carioca	99-884466
Francisco Bento	99-125544
Mônica Aparecida	98-337768

Produto cartesiano (\times)

As duas operações anteriores são operações unárias, ou seja, só podem operar sobre uma única tabela por vez. Algumas consultas só podem ser resolvidas a partir do cruzamento de duas ou mais tabelas. Por exemplo, imagine a consulta "Encontre o nome dos proprietários de fazendas com mais de 400 ha". Como o nome dos proprietários é um atributo da tabela PROPRIETÁRIO e a área da fazenda é um atributo da tabela FAZENDA, é necessário consultar as duas tabelas.

Para resolver este problema, na álgebra relacional é necessário "criar" uma nova tabela cujo esquema seja uma concatenação das duas tabelas desejadas. A operação produto cartesiano possibilita esta concatenação. Sua sintaxe é a seguinte:

$$\text{RELAÇÃO X RELAÇÃO}$$

O esquema da relação $A \times B$ = esquema de A concatenado ao esquema de B. Assim, para resolver a consulta "Encontre o nome dos proprietários de fazendas com mais de 400 ha" pode-se expressar da seguinte forma:

$$\Pi_{\text{nomeProp}} (\sigma_{\text{áreaFaz} > 400 \wedge \text{idProp} = \text{dono}} (\text{PROPRIETÁRIO X FAZENDA}))$$

O esquema da tabela PROPRIETÁRIO X FAZENDA será

idProp, nomeProp, ender, telProp, idFaz, nomeFaz, áreaFaz, dono

A tabela resultante é mostrada abaixo:

TABELA RESULTANTE

nomeProp
José Carioca
Francisco Bento

Uma forma alternativa de resolver esta consulta é usando a operação de atribuição para criar uma tabela temporária.

Temp <- Proprietário X Fazenda

$\Pi_{\text{nomeProp}} (\sigma_{\text{áreaFaz} > 400 \wedge \text{idProp} = \text{dono}} (\text{temp}))$

A tabela abaixo exemplifica algumas tuplas da tabela TEMP:

TEMP

idProp	nomeProp	ender	telProp	idFaz	nomeFaz	áreaFaz	dono
10	José Carioca	Av. N.S.Copa,123	99-884466	7663	Sta Rosa	300	10
10	José Carioca	Av. N.S.Copa,123	99-884466	4355	Rancho Fundo	350	10
10	José Carioca	Av. N.S.Copa,123	99-884466	8998	Cavalo Gordo	253	10
10	José Carioca	Av. N.S.Copa,123	99-884466	7644	Cavalo Magro	500	20
.....					
20	Francisco Bento	Rua Curva, 47	99-125544	7663	Sta Rosa	300	10
20	Francisco Bento	Rua Curva, 47	99-125544	4355	Rancho Fundo	350	10
20	Francisco Bento	Rua Curva, 47	99-125544	8998	Cavalo Gordo	253	10
.....					
30	Mônica Aparecida	Av. da Paz, 51	98-337768	7663	Sta Rosa	300	10
.....					

Importante observar que na tabela TEMP ocorrem tuplas cujos dados não são totalmente válidos. Por exemplo, a quarta tupla tem como identificador do proprietário (idProp) o valor 10 e como dono o valor 20. Este é um exemplo de tupla que deve ser desconsiderada na consulta, o que justifica a inclusão da condição que idProp deva ser igual a dono. Caso contrário, muitas tuplas falsas seriam incluídas na relação resultante. Observe que todo o segundo bloco de tuplas (idProp = 20) tem como dono o valor 10, assim todas essas tuplas são também falsas.

2.3 SQL – Linguagem de Consulta Estruturada

A linguagem de consulta SQL (*Structured Query Language*) foi desenvolvida originalmente como a linguagem de consulta do SGBD System R, da IBM. Após se tornar um padrão *de fato*, foi adotada internacionalmente como a linguagem de consulta para SGBD relacionais.

A primeira versão do padrão foi estabelecida pela ANSI em 1986, sendo batizada de SQL1. Em 1992 o padrão foi revisado, dando origem ao atual padrão, também conhecido por SQL2. Uma terceira versão (SQL3) encontra-se em desenvolvimento, devendo ser oficializada nos próximos anos. Entre as novidades do novo padrão SQL3 está a inclusão de conceitos de orientação a objetos e o suporte a objetos espaciais.

Na realidade, SQL é muito mais do que uma linguagem de consulta, ela é uma linguagem para ser usada como a linguagem do banco de dados, uma vez que ela inclui:

- Linguagem de consulta (QL)
- Linguagem de definição de dados (DDL)
- Linguagem de manipulação de dados (DML)
- Interface para linguagens de manipulação de dados embutida
- Autorização de acesso
- Restrições de integridade
- Controle de transações

A seguir, uma visão simplificada da linguagem SQL é apresentada, cujo objetivo é apenas o de permitir que o leitor possa ter uma idéia de como é possível extrair dados de um BD através da linguagem de consulta de um SGBD relacional.

Definição do esquema (tabelas)

Um esquema de BD é composto não só de tabelas, mas também por definições de visões, restrições de integridade e autorização de acesso. Todas as informações sobre o esquema de um BD são armazenadas no catálogo do BD, o qual é gerenciado pelo SGBD.

A definição de uma tabela é feita através do comando CREATE TABLE, exemplificada na declaração abaixo:

```
CREATE TABLE FAZENDA (
    idFaz      INTEGER      NOT NULL,
    nomeFaz    CHAR (30),
    áreaFaz    INTEGER,
    dono       CHAR (20),
    PRIMARY KEY (idFaz),
    FOREIGN KEY (dono) REFERENCES PROPRIETÁRIO );
```

A cláusula NOT NULL especifica que o atributo não pode conter valores nulos. A cláusula PRIMARY KEY indica qual (ou quais) atributo será a chave primária da tabela. A especificação da cláusula FOREIGN KEY (chave estrangeira) permite ao SGBD garantir a restrição de integridade referencial. Assim, o campo DONO na tabela FAZENDA ou terá um valor nulo, ou o valor deverá ser um valor existente na tabela PROPRIETÁRIO.

Consultas em SQL

Por ser uma linguagem declarativa o usuário não necessita especificar o algoritmo a ser seguido pelo programa para a execução da consulta, como ocorre nas linguagens de programação procedurais (ex.: Pascal, C, Cobol). O usuário necessita apenas especificar "o que" ele quer e o SGBD deve resolver a consulta. Como consequência, SQL não é uma linguagem computacionalmente completa, ou seja, nem tudo que é possível realizar com uma linguagem procedural é possível realizar com SQL. Por outro lado, o banco de dados pode ser consultado de forma exploratória pelo usuário, sem que este tenha que possuir grandes habilidades em programação.

Em SQL, as consultas são realizadas através do comando SELECT-FROM-WHERE. A estrutura básica do comando de consulta é a seguinte:

```
SELECT    <lista de atributos>
FROM      <lista de tabelas>
WHERE     <condição>
```

Uma consulta típica:

```
SELECT    A1, A2, A3, ..., An
FROM      R1, R2, R3, ..., Rn
WHERE     C
```

é equivalente, em álgebra relacional, a:

$$\prod_{A_1, A_2, A_3, \dots, A_n} (\sigma_C (R_1 \times R_2 \times R_3 \times \dots \times R_n))$$

A seguir são apresentados alguns exemplos de consulta em SQL.

- ❑ “Obter fazendas com mais de 400 ha”
SELECT *
FROM FAZENDA
WHERE áreaFaz > 400

- ❑ “Obter nome dos proprietários de fazenda com menos de 200 ha”
SELECT nomeProp
FROM PROPRIETÁRIO, FAZENDA
WHERE áreaFaz < 200 and idProp = dono

O conjunto completo de cláusulas do comando SELECT é o seguinte:

```
SELECT <lista de colunas>  
FROM <lista de tabelas>  
[WHERE <condição>]  
[GROUP BY <colunas para agrupamento>]  
[HAVING <condição de agrupamento>]  
[ORDER BY <lista de atributos>]
```

Apenas as cláusulas SELECT e FROM são obrigatórias. Quando existentes, as demais cláusulas devem aparecer na ordem especificada acima.

Funções Agregadas

Funções de agregação permitem computar valores sobre grupos de linhas de uma tabela. As funções mais utilizadas estão listadas abaixo:

- ❑ COUNT : contagem de linhas ou valores
- ❑ SUM : soma
- ❑ MAX : máximo
- ❑ MIN : mínimo
- ❑ AVG : média

As funções agregadas podem ser aplicadas sobre toda a tabela ou sobre um grupo de linhas. Permitem calcular valores sobre grupos de linhas, definidas pela cláusula GROUP BY. Vejamos alguns exemplos:

- ❑ “Obter área total das fazendas de cada proprietário”
SELECT dono, sum(áreaFaz)
FROM FAZENDA
GROUP BY dono

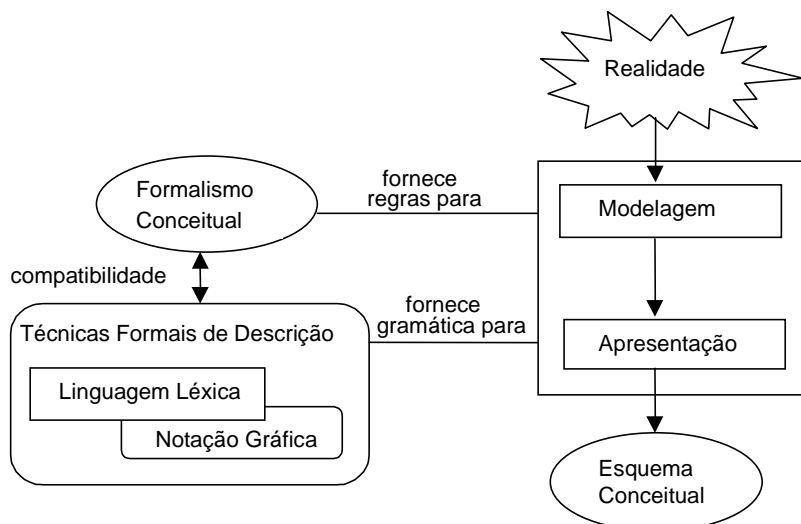
- ❑ “Listar fazendas em ordem decrescente de área”
SELECT nomeFaz, áreaFaz
FROM FAZENDA
ORDER BY áreaFaz DESC

3 PROJETO DE BANCO DE DADOS

3.1 Processo de Modelagem Conceitual

Um modelo de dados fornece uma base formal (notacional e semântica) para ferramentas e técnicas usadas para suportar a modelagem de dados. Modelagem de dados é o processo de abstração onde somente os elementos essenciais da realidade observada são enfatizados, descartando-se os elementos não essenciais.

O processo de modelagem conceitual compreende a descrição e definição dos conteúdos dos dados, além de suas estruturas e de regras aplicáveis aos dados [EUR 96]. A modelagem conceitual é sempre feita com base em algum formalismo conceitual (ex.: Entidade-Relacionamento, Orientação a Objetos), independente do nível de abstração empregado. O resultado da modelagem, denominado esquema conceitual, é apresentado através de uma linguagem formal de descrição, que possui uma sintaxe e/ou uma notação gráfica. A figura abaixo ilustra o processo de modelagem conceitual de dados.

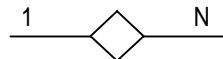


Para cada formalismo conceitual existe um conjunto de linguagens de descrição de esquema que são compatíveis com o formalismo. O formalismo fornece um conjunto de regras que são usadas no processo de modelagem da realidade, enquanto que a linguagem de descrição fornece uma gramática para a apresentação do esquema conceitual resultante da modelagem. A linguagem léxica possibilita o processamento computacional do esquema, enquanto a notação gráfica é mais adequada para facilitar o entendimento e a comunicação entre humanos (ex.: usuários e projetistas).

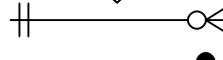
A seguir, são listadas as principais linguagens de modelagem disponíveis, classificadas segundo o formalismo empregado. A figura ao lado ilustra a notação gráfica correspondente a um relacionamento 1:N.

Formalismo E-R:

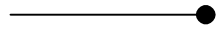
Peter Chen



James Martin (eng. informações)

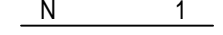


IDEF1X

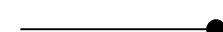


Formalismo da orientação a objetos:

OOA – Análise orientada a objetos



OMT – Técnica de modelagem de objetos



UML – Linguagem de modelagem unificada



3.2 Modelo Entidade-Relacionamento (E-R)

O modelo Entidade-Relacionamento (E-R), proposto por Peter Chen em 1976, baseia-se na percepção de um universo constituído por um grupo básico de objetos chamados entidades e por relacionamentos entre estes objetos [CHE 76].

O modelo E-R foi desenvolvido para facilitar o projetista na tarefa de projetar bancos de dados relacionais. É um modelo semântico (ou conceitual) de alto nível, ou seja, usa uma linguagem que está mais próxima do usuário do que da máquina (computador). Um esquema de dados elaborado com o modelo E-R especifica “quais” os dados que serão representados e não “como” os dados serão armazenados. Assim, é um modelo que é independente do SGBD a ser utilizado na implementação do BD.

Por possuir uma notação gráfica bastante simples, é de fácil aprendizado. Além de facilitar o entendimento da estrutura do BD por parte dos usuários leigos, melhora a comunicação entre os projetistas e os usuários.

A seguir são apresentados os elementos construtores do modelo E-R.

Entidade

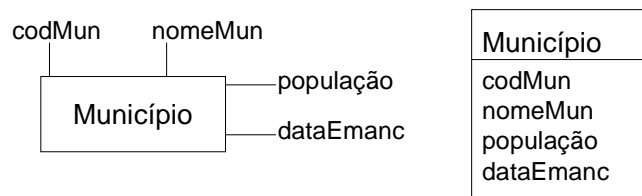
Todo fenômeno de interesse da aplicação é representado, ou modelado, como uma entidade. Uma entidade pode ter existência física (ex.: rio, estrada, igreja) ou existência conceitual (ex.: empresa, cargo, município).

Entidades são modeladas através do mecanismo de abstração conhecido como classificação, no qual as entidades que são similares são descritas uma única vez, formando um conjunto entidade. As características de uma entidade são modeladas através de atributos da entidade (ex.: nome e população de um município).

Ao contrário do modelo relacional, no qual todo atributo deve ser atômico, no modelo E-R diversos tipos de atributos podem ser especificados. São eles:

- ❑ Atômico (ex.: nome, população)
- ❑ Composto (ex.: endereço = [rua+núm+bairro+cep])
- ❑ Derivado (ex.: idade = dataHoje - dataFundação)
- ❑ Multivalorado (ex.: {telefones para contato})

A notação gráfica para representar um conjunto entidade é um retângulo, podendo os atributos estarem dentro ou fora dos limites do retângulo, de acordo com a linguagem gráfica usada. As figuras abaixo ilustram duas alternativas de modelagem do conjunto entidade Município e seus atributos.



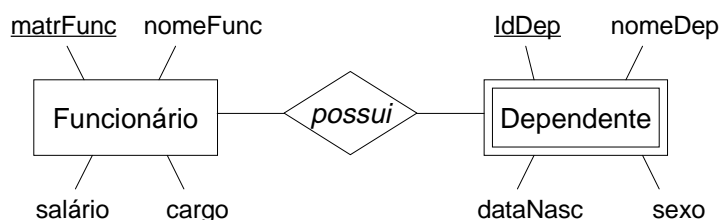
Chave Primária

Como na implementação de um banco de dados relacional, o conceito de chave primária é de fundamental importância, mesmo sendo um conceito mais próximo da implementação, é importante que sejam identificadas os atributos que comporão a chave primária de um conjunto entidade.

Chave primária é o atributo (ou conjunto de atributos) que identifica(m) univocamente cada entidade em um conjunto entidade. No exemplo acima, o atributo codMun é um candidato a ser escolhido como chave primária para o conjunto entidade Município.

Entidade Fraca

Alguns conjuntos entidade não possuem um conjunto de atributos capaz de identificar univocamente uma determinada entidade. Neste caso, sua existência depende da existência de outra entidade. Um exemplo clássico de entidade fraca ocorre quando um sistema de pessoal possui dados sobre os dependentes de seus funcionários. É comum que os funcionários tenham um número de matrícula na empresa que é usado como atributo chave da tabela de funcionários. Por outro lado, normalmente os dependentes dos funcionários não possuem uma matrícula própria, bem como não possuem um conjunto mínimo de atributos que possam constituir a chave primária da tabela de dependentes. Neste caso, o conjunto entidade Dependente deve ser modelado como uma entidade fraca. A figura abaixo ilustra a notação gráfica usada para representar conjuntos de entidades fracas.



Relacionamento

O segundo construtor do modelo E-R é o construtor para representação de relacionamentos. Um relacionamento descreve uma associação entre duas ou mais entidades. Por exemplo, as expressões abaixo sinalizam relacionamentos importantes que deverão ser representados no esquema do BD.

Salvador é capital da Bahia.

Curitiba é capital do Paraná.

Belo Horizonte localiza-se em MG.

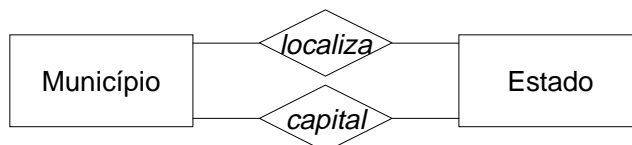
Viçosa localiza-se em MG.

Da mesma forma que entidades semelhantes são agrupadas em conjuntos entidade, relacionamentos semelhantes são agrupados em conjuntos relacionamentos e modelados uma única vez. Assim, os relacionamentos acima poderiam ser classificados da seguinte forma:

Município é capital de Estado.

Município localiza-se em Estado.

Uma notação gráfica para representação desses relacionamentos é exemplificada na figura abaixo.



A modelagem dos relacionamentos é uma das tarefas mais importantes no projeto conceitual. É a partir dos relacionamentos que o banco de dados poderá ser explorado através de consultas. O projetista de uma aplicação deve ter a habilidade de escolher, dentre uma infinidade de possíveis relacionamentos, aqueles que são importantes de acordo com os objetivos do sistema.

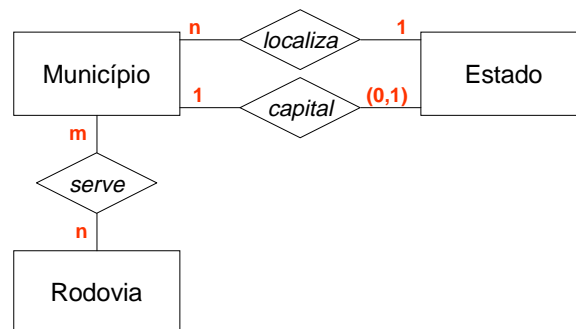
Cardinalidades

No modelo E-R, os relacionamentos devem especificar um importante tipo de restrição de integridade que é a sua cardinalidade. A cardinalidade de um relacionamento indica o número de entidades às quais uma outra entidade pode estar associada via um relacionamento.

Em um relacionamento entre dois conjuntos entidade A e B, existem três tipos básicos de cardinalidades:

- ❑ 1:1 (um para um) – cada entidade no conjunto entidade A pode estar associada a no máximo uma entidade no conjunto entidade B e vice-versa;
- ❑ 1:N (um para muitos) – cada entidade no conjunto entidade A pode estar associada a nenhuma ou muitas entidades no conjunto entidade B e cada entidade no conjunto entidade B só pode estar associada a uma única entidade no conjunto entidade A;
- ❑ M:N (muitos para muitos) – cada entidade no conjunto entidade A pode estar associada a nenhum ou muitas entidades no conjunto entidade B, e vice-versa.

A figura abaixo ilustra a representação de cardinalidades nos relacionamentos:

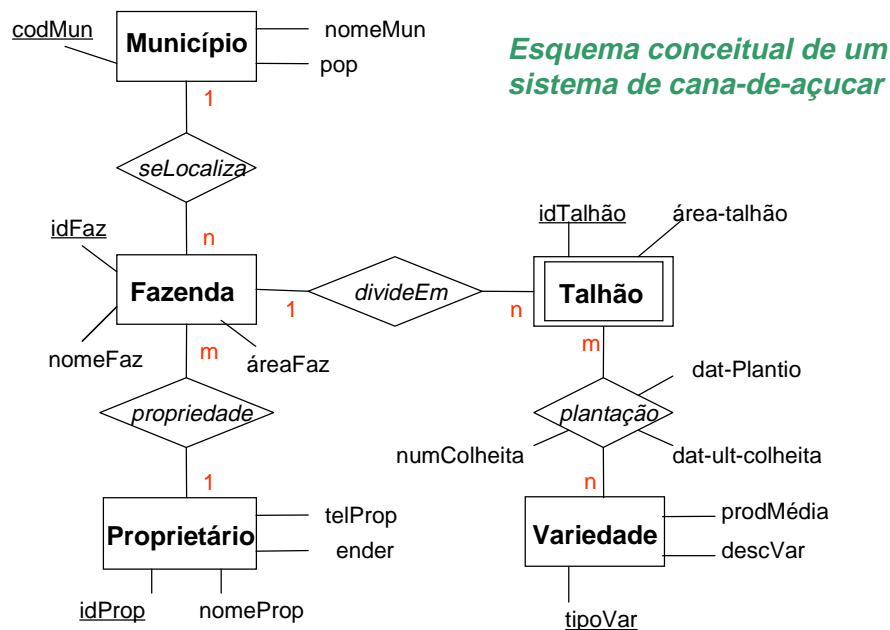


É possível representar cardinalidades mínimas e máximas de um relacionamento, como mostrado na figura acima. Neste exemplo, a cardinalidade 1:(0,1) significa que cada estado, no conjunto entidade Estado, possui exatamente um município que é sua capital e que cada município, no conjunto entidade Município, pode não ser capital de estado, ou ser capital de no máximo um estado.

3.3 Projeto Lógico de Banco de Dados

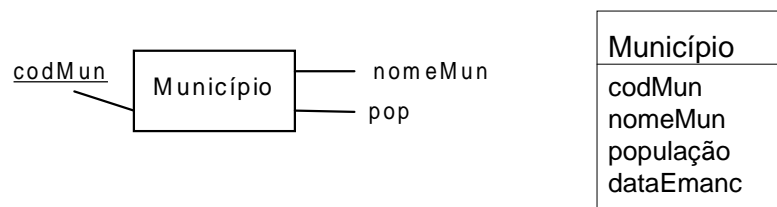
A fase de projeto lógico consiste em transformar, ou “mapear”, um esquema conceitual de dados em um esquema lógico. Nesta fase, a primeira decisão a ser tomada é a escolha do SGBD no qual o BD será implementado. Esta escolha determina o modelo de dados a ser utilizado (ex.: modelo relacional, orientado a objetos, objeto-relacional). Nesta seção é apresentado um conjunto de regras a serem aplicadas para transformação de esquemas E-R em esquemas relacionais.

A seguinte figura, a qual ilustra um esquema E-R de um sistema de cana-de-açúcar, é usada para exemplificar as regras de transformação.



Transformando Entidades

Normalmente, cada entidade dá origem a uma tabela, cujas colunas correspondem aos atributos da entidade. Duas notações para o modelo relacional, uma textual e outra gráfica, são apresentadas abaixo. A notação gráfica é a notação do modelo IDEF1X, utilizada na ferramenta CASE ERWin.



A notação textual é uma versão simplificada, na qual os domínios dos atributos são suprimidos para efeitos didáticos. Os atributos que fazem parte da chave são sublinhados. Inicialmente, no esquema E-R do sistema de cana-de-açúcar, as seguintes tabelas poderiam ser deduzidas a partir desta regra:

- Município (cod-mun, nomeMun, pop)
- Proprietário (idProp, nomeProp, ender, telProp)
- Fazenda (idFaz, nomeFaz, áreaFaz)
- Talhão (idFaz, idTalhão, áreaTalhão)
- Variedade (tipoVar, descrVar, prodMédia)

Observe que a tabela Talhão possui uma chave dupla devido a entidade ser fraca. A aplicação das demais regras provocarão mudanças em algumas destas tabelas, como será visto a seguir.

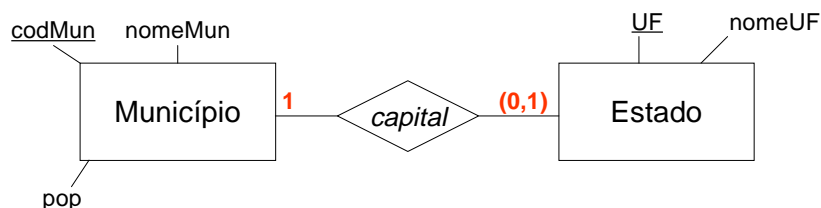
Transformando Relacionamentos

Relacionamentos podem ser transformados de duas formas distintas: (1) através da inclusão do atributo chave de uma das entidades, na tabela resultante da transformação da outra entidade envolvida no relacionamento; (2) definição de uma tabela específica para armazenar os relacionamentos, a qual tem como chave primária, as chaves primárias das duas tabelas envolvidas.

Portanto, a transformação de relacionamentos depende de vários aspectos do problema (sistema) em questão. Exemplos que influenciam na decisão de qual a melhor forma de transformar os relacionamentos incluem, por exemplo, cardinalidade opcional, volume de entidades envolvidas em um determinado tipo de relacionamento, volume de uso deste relacionamento em consultas futuras, etc. Assim, embora existam regras que orientem este tipo de transformação, é importante o projetista ter em mente que aspectos da aplicação devem ser considerados para uma escolha mais adequada em cada caso. As regras apresentadas a seguir são um resumo dos casos mais comuns de transformação. Apenas alguns casos especiais são comentados.

Relacionamentos 1:1

Relacionamentos 1:1 normalmente são transformados através da inclusão de um atributo extra em umas das tabelas envolvidas. A escolha de qual entidade vai receber o atributo chave da outra entidade merece uma análise especial do projetista. A figura abaixo ilustra um esquema E-R e um possível esquema relacional equivalente.



Município (cod-mun, nomeMun, pop, UF*)

*UF referencia Estado

Estado (UF, nomeUF)

A inclusão de um asterisco (“*”) no campo UF significa que UF é uma chave estrangeira para outra tabela, no caso a tabela Estado. Uma outra alternativa para a transformação deste esquema E-R está mostrado abaixo.

Município (cod-mun, nomeMun, pop)

Estado (UF, nomeUF, cod-mun-capital*)

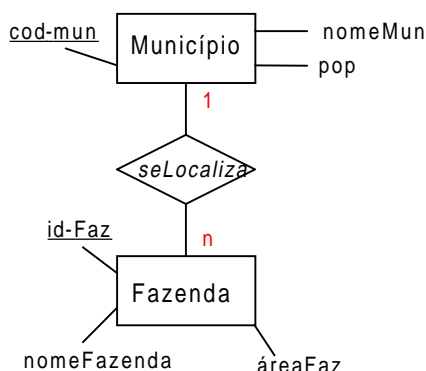
*cod-mun-capital referencia Município

Qual das alternativas é a melhor? Analise o que ocorre com as colunas UF, da tabela Município (primeira alternativa) e cod-mun-capital, da tabela Estado (segunda alternativa).

Relacionamentos 1:N

Ao contrário dos relacionamentos 1:1, em que há a opção de escolha, em um relacionamento 1:N a entidade do lado “N” sempre vai receber o(s) atributo(s) chave da entidade do lado “1”, a menos que o projetista opte por transformar o relacionamento em uma tabela própria (caso excepcional).

A figura abaixo ilustra um esquema E-R e o esquema relacional equivalente.



Município (cod-mun, nomeMun, pop)

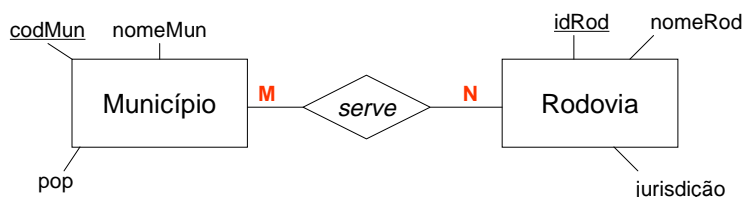
Fazenda (idFaz, nomeFaz, áreaFaz, codMun*)

*codMun referencia Município

Qual o problema de se incluir o campo IdFaz na tabela Município?

Relacionamentos M:N

A única forma possível de transformação de relacionamentos M:N é através da definição de uma tabela adicional, cuja chave é composta pelas chaves primárias das duas entidades envolvidas. Isto ocorre porque no modelo relacional há a restrição de domínio atômico. Dessa forma, não é possível incluir um conjunto de rodovias na tabela Município e vice-versa. A figura abaixo ilustra um esquema E-R e o esquema relacional equivalente.



Município (cod-mun, nomeMun, pop)

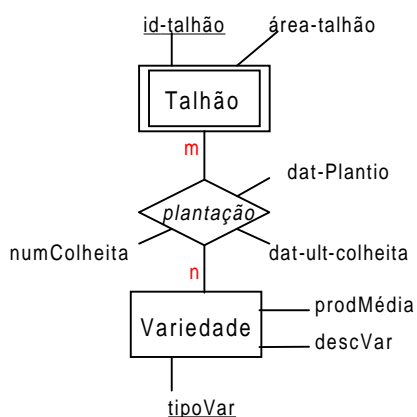
Rodovia (idRod, nomeRod, jurisdição)

MunRod (cod-mun*, idRod*)

* cod-mun referencia Município

* idRod referencia Rodovia

Uma situação que ocorre com frequência é um relacionamento ter atributos. A figura abaixo ilustra parte do diagrama E-R do sistema de cana-de-açúcar, em que o relacionamento *plantação* possui atributos próprios. O exemplo também ilustra a transformação de relacionamentos envolvendo uma entidade fraca.



Talhão (id-talhão, área-talhão)

Variedade (tipoVar, descVar, prodMédia)

Plantação (idFaz*, id-talhão*, tipoVar*, numColheita, dat-Plantio, dat-ult-colheita)

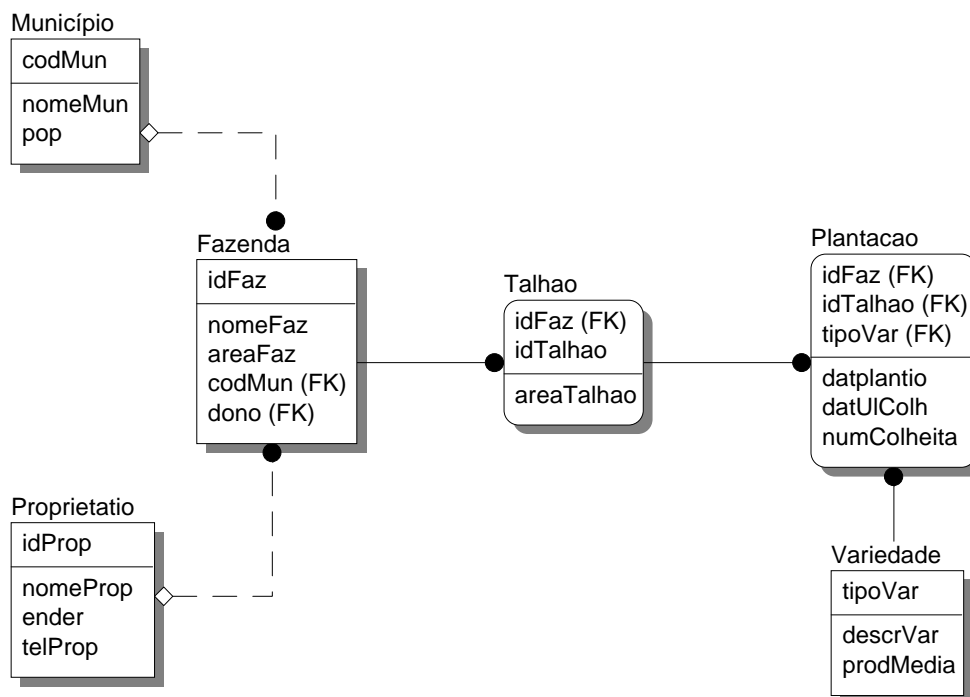
*idFaz, id-talhão referenciam Talhão

*tipoVar referencia Variedade

3.4 Ferramentas CASE

As regras de transformação apresentadas anteriormente resolvem grande parte dos problemas de transformação de esquemas conceituais em esquemas relacionais. Como se pode notar, a base para escolha da regra a ser empregada é o tipo de cardinalidade do relacionamento. Portanto, o projetista deve reservar especial atenção à modelagem dos relacionamentos e suas cardinalidades.

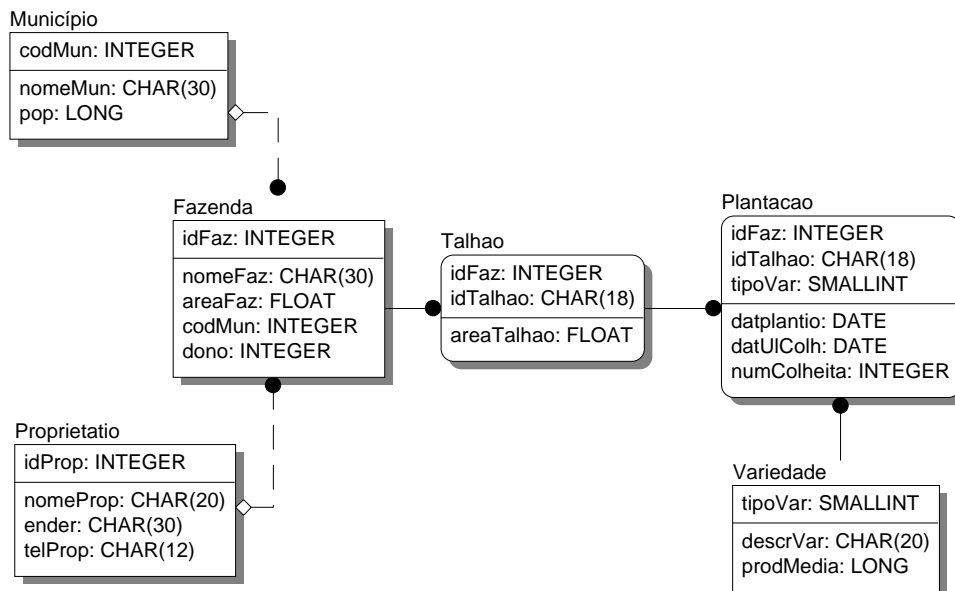
Estas regras também são empregadas pelos programas de apoio a projeto de BD, conhecidos por ferramentas CASE. A figura abaixo ilustra o esquema completo do sistema de cana-de-açúcar elaborado através da ferramenta CASE ERWin.



A figura ilustra o que na ferramenta ERWin é chamado de Nível Lógico, uma vez que o nível de detalhe apresentado está mais próximo do modelo relacional do que do modelo conceitual, mesmo embora a figura se assemelhe bastante com um esquema E-R. Alguns detalhes podem ser observados na figura acima. São eles:

- ❑ As figuras representadas no formato retangular representam tabelas do modelo relacional e não entidades do modelo E-R. Observe a existência da tabela Plantação, que no modelo E-R corresponde a um relacionamento;
- ❑ Existem dois tipos de representação gráfica para tabelas, retângulos com bordas retas e abaloadas. Retângulos com bordas abaloadas sinalizam tabelas resultantes da transformação de entidades fracas (ex.: Talhão) ou tabelas resultantes da transformação de relacionamentos (Plantação). Observe a presença de atributos que são chaves estrangeiras (FK – *foreign key*) fazendo parte da chave primária destas tabelas (ex.: idFaz (FK) em Talhão).
- ❑ A linha tracejada representa um relacionamento 1:N entre duas entidades fortes, que é transformado pela inclusão de uma chave estrangeira na tabela do lado N (que não participa da chave primária). Por exemplo, o relacionamento Fazenda – Proprietário é representado por uma linha tracejada (com um diamante na ponta que não significa nada). Neste caso, a ferramenta inclui automaticamente na tabela Fazenda, o campo idProp, o qual foi renomeado para “dono”, assinalado como (FK).
- ❑ A linha contínua indica a transformação de um relacionamento 1:N, envolvendo uma de entidade fraca, ou de um relacionamento M:N. Neste caso o sistema inclui automaticamente a chave estrangeira como chave primária da tabela resultante.

A figura abaixo ilustra o mesmo esquema relacional do sistema de cana-de-açúcar, na visão chamada Modelo Lógico Detalhado, na qual cada atributo está associado a um domínio específico.



Estas informações são suficientes para a geração automática do esquema do BD em um SGBD. A ferramenta CASE ERWin, por exemplo, gera código automático para diversos SGBD (ex.: Oracle, DB2, SQL Server).

A seguir o mesmo esquema é apresentado na notação textual simplificada:

Município (cod-mun, nomeMun, pop)

Proprietário (idProp, nomeProp, ender, telProp)

Fazenda (idFaz, nomeFaz, áreaFaz, codMun*, dono*)

*codMun referencia Município

*dono referencia Proprietário

Talhão (idFaz*, idTalhão, áreaTalhão)

*idFaz referencia Fazenda

Plantação (idFaz*, id-talhão*, tipoVar*, numColheita, dat-Plantio, dat-ult-colheita)

*idFaz, id-talhão referenciam Talhão

*tipoVar referencia Variedade

Variedade (tipoVar, descrVar, prodMédia)

4 PROJETO DE BANCOS DE DADOS GEOGRÁFICOS

4.1 Dado, Informação e Fenômeno Geográfico

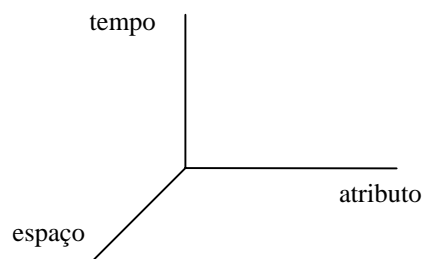
Uma informação é obtida a partir do processamento ou da contextualização de dados brutos. Um dado bruto corresponde a um valor para uma medida observada. Por exemplo, a data 7/9/1822 é um dado bruto. Quando este dado está associado a um contexto torna-se informação. Assim, na expressão “O Brasil tornou-se independente de Portugal em 7 de setembro de 1822”, o dado é transformado em informação.

De modo equivalente, informação geográfica é resultado do processamento de dados geográficos. Um dado geográfico refere-se a uma medida observada de um fenômeno que ocorre sobre/sob a superfície terrestre, onde a localização da observação é um componente fundamental do dado.

O termo fenômeno geográfico compreende, de forma abrangente, a qualquer ocorrência que pode ser: natural (ex.: um lago, a pressão atmosférica, uma formação geológica); antrópica (ex.: uma rodovia, um hospital, divisão territorial política); de fatos (ex.: uma epidemia, uma batalha); ou mesmo de objetos ainda inexistentes (ex.: o planejamento de um gasoduto, projeto de uma usina hidroelétrica).

Componentes da informação geográfica

Segundo Chrisman [CHR 97], a informação geográfica possui três componentes básicos: atributo, espaço e tempo, que possibilitam responder, respectivamente, a três perguntas: “o quê?”, “onde?” e “quando?”. Segundo Worboys [WOR 95], cada um desses componentes determina uma categoria de dimensão ao longo da qual os valores são medidos.



Categorias de dimensões da informação geográfica

Para que um dos três componentes possa ser medido deve-se fixar um segundo componente e usar o terceiro como controle (Sinton, 1978 Apud [CHR 97]). Por exemplo, para medir a altitude (atributo) de uma determinada região, fixa-se o componente tempo, faz-se variar a posição no espaço (controle) e toma-se o valor da altitude para os diversos pontos na região. Um segundo exemplo compreende a medição da temperatura em uma estação meteorológica. Neste caso, a localização da estação é fixa (espaço), o período de tempo da observação serve de controle e a temperatura (atributo) é medida para cada unidade de tempo estabelecida. A definição de qual dimensão pretende-se fixar, usar como controle ou medir, possibilita uma grande variedade de estruturas de medidas da informação geográfica.

Componente Espaço

O componente espacial descreve a localização geográfica e a forma geométrica do fenômeno descrito pela informação geográfica, além de relacionamentos com outros fenômenos geográficos. Como a principal função de um SIG é possibilitar a realização de operações de análise espacial, o componente espacial é o mais importante no contexto de SIG.

Um dado geográfico é, antes de mais nada, um dado espacial, isto é, descreve a forma geométrica (coordenadas numéricas) de um objeto no espaço. Por exemplo, a forma de um cubo no espaço \mathbb{R}^3 ou uma circunferência projetada no espaço \mathbb{R}^2 são dados espaciais. Assim, uma linha unindo dois pontos quaisquer em uma imagem de ecografia é um dado espacial, mas não um dado geográfico.

Quando o dado espacial está relacionado com sua localização sobre a superfície terrestre, ou seja, são utilizadas coordenadas geográficas (ex.: latitude/longitude), este dado, além de ser espacial, é um dado geográfico, também conhecido por dado geo-espacial ou dado georreferenciado.

Devido a uma herança cultural no desenvolvimento de SIG, decorrente do uso de mapas em papel, existe uma predominância quase total no processamento de dados bidimensionais, embora seja possível o processamento de dados espaciais em três dimensões.

Componente Atributo

Um fenômeno geográfico possui características qualitativas e quantitativas que são descritas de forma textual e/ou numérica. O componente atributo, também conhecido por atributo descritivo ou atributo não-espacial, descreve as características não espaciais de um fenômeno geográfico. Nome, população e orçamento anual de um município são exemplos de atributos descritivos.

Atributos descritivos são os mais freqüentes na maioria dos sistemas de informação em geral. A maioria dos SIG utilizam um SGBD para o gerenciamento dos atributos descritivos.

Segundo Aronoff [ARO 89], existe um certo grau de incerteza que é inerente ao componente atributo. Por exemplo, uma floresta de Eucaliptos pode não ser formada por árvores 100% do tipo Eucalipto ou um bairro residencial pode possuir algumas pequenas indústrias instaladas. Para alguns tipos de análises pode ser importante considerar a existência dessas incertezas, embora normalmente este tipo de problema seja desconsiderado pelos usuários de SIG.

Componente Tempo

Todo fenômeno geográfico é eminentemente temporal, ou seja, está associado a um instante ou intervalo de tempo em que este ocorre ou em que é observado [PEU 95]. O componente tempo pode ser crítico para a informação geográfica, dependendo do tipo de fenômeno e do tipo de aplicação em que este está sendo utilizado.

O componente tempo, embora muitas vezes esteja implicitamente associado aos demais componentes da informação geográfica, tem sido tratado como uma simples informação complementar. Por exemplo, todo mapa está associado ao período de tempo relativo à sua confecção e ao período de tempo relativo à validade da informação apresentada. Um novo mapa retratando as divisões provinciais do Brasil no século XVII ilustra esta situação.

A maioria dos sistemas são projetados para fornecer informação atual sobre os fenômenos geográficos. Por exemplo, um sistema de rede de telefonia deve fornecer aos seus usuários, a informação mais atualizada possível. Desta forma, a informação que é extraída do sistema é tida como verdadeira para o momento da consulta. É comum a existência de aplicações onde os aspectos temporais não são contemplados ou são tratados de maneira superficial. No entanto, existem aplicações onde o aspecto temporal é fundamental. Por exemplo, em um sistema de cadastro urbano, os limites dos lotes podem ser modificados (divididos ou unidos a outros lotes), proprietários podem ser trocados, edificações são construídas. O histórico dessas modificações no cadastro territorial é um fator importante neste tipo de aplicação.

4.2 Requisitos de Modelagem para Aplicações de SIG

Devido à necessidade de tratar as três dimensões da informação geográfica, as aplicações de SIG impõem alguns requisitos especiais de modelagem que devem ser suportados pelos modelos conceituais no projeto de banco de dados para estas aplicações. A seguir é descrito o conjunto de requisitos mínimos que um modelo conceitual de dados para aplicações de SIG deve suportar [LIS 99a].

Fenômeno Geográfico e Objeto Convencional

Em um banco de dados geográficos existem, além dos dados referentes aos fenômenos geográficos, outros objetos convencionais, presentes na maioria dos sistemas de informação. Por exemplo, uma fazenda é um fenômeno geográfico quando suas informações espaciais (ex.: os limites da fazenda) estão armazenadas no banco de dados. Neste mesmo banco de dados pode-se ter dados sobre os proprietários de fazendas, considerados objetos convencionais por não terem informações espaciais associadas. Em um esquema conceitual é importante que se possa diferenciar, facilmente, entre classes (ou entidades) descrevendo esses dois tipos de objetos.

Visões de Campo e de Objetos

Segundo Goodchild [GOO 92], a realidade geográfica pode ser observada segundo duas visões: de campo e de objetos.

Na visão de campo a realidade é modelada por variáveis que possuem uma distribuição contínua no espaço. Toda posição no espaço geográfico pode ser caracterizada através de um conjunto de atributos como temperatura, tipo de solo e relevo, medidos para um conjunto de coordenadas geográficas.

Na visão de objetos, a realidade consiste de fenômenos individuais, bem definidos e identificáveis. Cada fenômeno na visão de objetos tem suas propriedades individuais e ocupa um determinado lugar no espaço. A realidade é modelada como um grande espaço onde os fenômenos estão distribuídos sem que, necessariamente, todas as posições do espaço estejam ocupadas.

A modelagem de fenômenos geográficos na visão de objetos¹ é um processo natural e direto (ex.: rios são descritos pela classe ou entidade *Rio*). No entanto, um campo geográfico (ex.: *Altimetria*) não pode ser modelado diretamente como uma classe, pois *Altimetria* não é um objeto e, conseqüentemente, não pode ser descrito diretamente por uma classe.

Um modelo conceitual de dados deve prover construtores especiais para modelar tanto os campos quanto os objetos geográficos. A maioria dos modelos existentes não suporta a modelagem dos fenômenos geográficos que são percebidos na visão de campo.

Aspectos Temáticos

Os fenômenos geográficos possuem vários atributos, dentre os quais o que fornece sua localização geográfica (atributo espacial). A localização e a forma dos fenômenos geográficos são representadas através de objetos espaciais, associados a um sistema de coordenadas. Um objeto representando um fenômeno geográfico (ex.: um rio) deve estar associado a um ou mais objetos espaciais que representam sua localização e sua forma sobre a superfície terrestre. Em um SIG, esses objetos espaciais não são tratados isoladamente, mas sim, em grupos que representam fenômenos com características e relacionamentos em comum. Diferentes termos como *tema*, *camada* e *plano de informação* são encontrados na literatura, referindo-se a estes agrupamentos.

No nível interno de um SIG, a organização dos dados espaciais é feita em camadas físicas (ex.: conjunto de polígonos representando os limites dos lotes urbanos). No entanto, embora o projeto de camadas físicas seja um problema a ser tratado nas etapas de projeto lógico ou físico, diversos autores afirmam que é importante que camadas conceituais sejam definidas durante a fase de projeto conceitual [WOR 95, HAD 96]. Para Hadzilacos [HAD 96], camadas conceituais não necessitam ter

¹ É importante ressaltar a diferença entre os conceitos de *visão de objetos*, em contrapartida à visão de campos, e *modelos de objetos*, referente à orientação a objetos.

uma relação um-para-um com camadas físicas. Por exemplo, uma única camada conceitual, como *Hidrografia*, pode dar origem a diferentes camadas físicas, uma contendo somente os rios, outra contendo somente os lagos, etc.

Aspectos Espaciais

A localização geográfica e a forma espacial dos fenômenos geográficos podem ser implementadas, em um SIG, através de objetos espaciais. Todo objeto espacial possui uma geometria, que representa a forma espacial do fenômeno, sendo que suas coordenadas devem estar registradas com base em um determinado sistema de coordenadas (ex.: latitude e longitude) e um sistema de projeção (ex.: UTM).

Embora os aspectos espaciais dos fenômenos geográficos possam parecer detalhes que não deveriam ser tratados durante a modelagem conceitual, a sua inclusão no esquema conceitual tem sido fator fundamental na comunicação com o usuário [LIS 00].

Na visão de objetos, os fenômenos geográficos são representados por objetos espaciais do tipo ponto, linha, polígono ou combinações destes, enquanto que na visão de campo uma superfície contínua pode ser representada, por exemplo, através de modelos numéricos, conjuntos de isolinhas, polígonos adjacentes e grade de células.

Diferentes abordagens têm sido propostas para modelagem conceitual dos aspectos espaciais dos fenômenos geográficos. A abordagem mais comum é a que define uma associação entre a classe que descreve o fenômeno e a classe do tipo de objeto espacial correspondente a sua representação espacial. Alguns modelos, no entanto, utilizam pictogramas² para substituir esta associação.

Múltiplas Representações

Uma das características das aplicações geográficas é a possibilidade de existência de múltiplas representações para um mesmo fenômeno geográfico. Esta necessidade surge em resposta à complexidade da realidade a ser representada e às diferentes visões que os usuários têm de um mesmo fenômeno. Um fenômeno geográfico pode ser representado em diferentes escalas ou projeções, inclusive por diferentes objetos espaciais.

A existência de múltiplas representações pode ser modelada através da inclusão de várias associações entre o fenômeno geográfico e os tipos de objetos espaciais correspondentes.

Relacionamentos Espaciais

Uma das tarefas mais importantes quando se está modelando os dados de uma aplicação é a identificação de quais os relacionamentos que deverão ser mantidos no banco de dados, dentre os possíveis relacionamentos observáveis na realidade. No domínio das aplicações geográficas este problema é bem complexo, uma vez que o número de relacionamentos possíveis de serem mantidos é ainda maior, devido à existência dos relacionamentos espaciais entre os fenômenos geográficos.

A maioria dos SIG fornece estruturas especiais para o armazenamento explícito de alguns tipos de relacionamentos espaciais. Normalmente são mantidos somente os relacionamentos de adjacência ou de conectividade, deixando os demais tipos de relacionamentos espaciais (ex.: pertinência, cruzar) para serem calculados a partir das coordenadas espaciais dos objetos. Por outro lado, existem aplicações nas quais alguns relacionamentos espaciais possuem significado semântico relevante, tal como vizinhança e cruzamento de ruas. Nestes casos o projetista necessita especificar estas informações no modelo de dados.

As cardinalidades associadas aos relacionamentos formam um conjunto de restrições de integridade que devem ser mantidas entre as instâncias dos objetos no banco de dados. Para os relacionamentos espaciais, novos tipos de cardinalidade podem ser definidos, tais como, associação espacial, pertinência, etc. Um modelo conceitual de dados para SIG deve fornecer meios para que o projetista represente os relacionamentos a serem mantidos no banco de dados geográficos. Isto inclui tanto os relacionamentos convencionais como os relacionamentos espaciais.

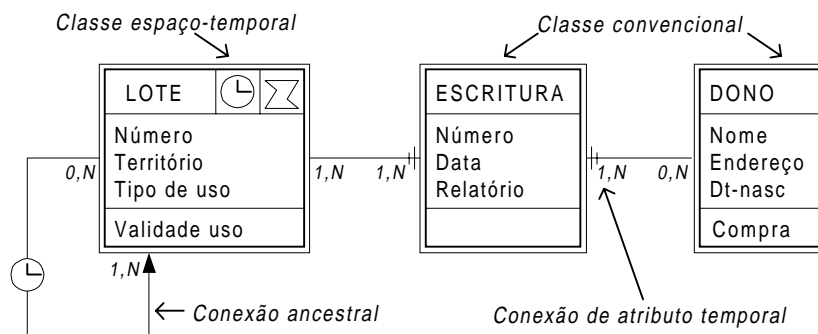
² Pequenos símbolos gráficos incluídos na definição das classes/entidades.

Aspectos Temporais

A maioria dos SIG disponíveis atualmente considera os fenômenos como se o mundo existisse somente no presente. Dados geográficos são alterados ao longo do tempo, mas o histórico dessas transformações não é mantido no banco de dados.

Para Hadzilacos [HAD 96], a necessidade de os dados geográficos estarem qualificados com base no tempo, se deve à necessidade de se registrar estados passados de forma a possibilitar o estudo da evolução dos fenômenos geográficos. Para possibilitar uma análise de dados com base na evolução dos fenômenos geográficos é necessário adicionar aos SIG as potencialidades dos sistemas de bancos de dados temporais. Isto implica na inclusão de informações temporais relacionadas aos fenômenos geográficos e da extensão das linguagens de consulta disponíveis para que estas suportem cláusulas de condição associadas a aspectos temporais [WOR 95].

Alguns modelos conceituais propostos na literatura estendem a notação gráfica para ressaltar a existência dos aspectos temporais em determinadas classes. O modelo GeoOOA [KOS 97], por exemplo, utiliza um pictograma especial (símbolo de um relógio), para diferenciar classes temporais. Além disso, a notação gráfica do modelo é estendida para representar dois tipos especiais de relacionamentos temporais: *conexão ancestral* e *conexão de atributo temporal*. A figura abaixo ilustra um trecho da modelagem de um sistema de cadastramento de lotes.



No modelo GeoOOA, um relacionamento de conexão ancestral permite a associação entre versões de um mesmo objeto. Por exemplo, um lote pode ser dividido, em um determinado momento, dando origem a dois novos lotes. A conexão de atributo temporal é usada para destacar uma associação decorrente da necessidade de modelar o aspecto temporal de um atributo. Na figura acima, a classe *Escritura* é resultado da modelagem dos dados históricos do lote. Na modelagem não temporal, o usual é que cada lote tenha uma única associação com o proprietário atual.

Modelos Conceituais de Dados para SIG

Os requisitos apresentados anteriormente fazem com que modelos conceituais convencionais, como o modelo E-R, não sejam adequados para modelagem de bancos de dados geográficos, ou seja, para modelagem de aplicações de SIG. Atualmente, diversas propostas de modelos conceituais de dados específicos para bancos de dados geográficos podem ser encontrados na literatura. Estes modelos são, na verdade, extensões dos modelos tradicionais. Abaixo são citados os principais modelos encontrados na literatura, classificados de acordo com o tipo de formalismo usado como base:

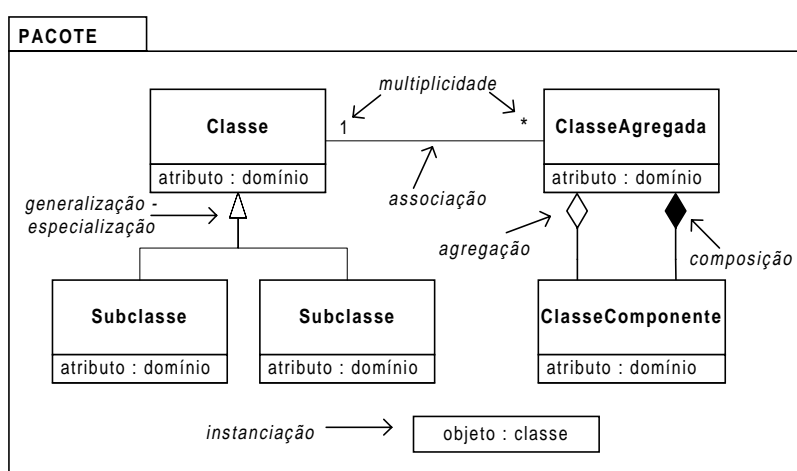
- Formalismo E-R: Modul-R [BED 96], GISER [SHE 97], Geo-ER [HAD 97];
- Formalismo OO: GMOD [PIR 97], Geo-OMT [BOR 97], GeoOOA [KOS 97], MADS [PAR 98], Perceptory [BED 99], UML-GeoFrame [LIS 99].

Uma análise crítica desses modelos, feita com base nos requisitos descritos nesta seção é apresentada em [LIS 99a].

4.3 Modelando Banco de Dados Geográficos com UML-GeoFrame

Nesta seção é apresentada a abordagem UML-GeoFrame, ou seja, o uso da linguagem UML (*Unified Modeling Language*) [BOO 98], com base no *framework* GeoFrame [LIS 99], para a modelagem conceitual de bancos de dados geográficos. Esta solução possibilita a obtenção de esquemas conceituais de dados numa linguagem bastante clara e, portanto, de fácil entendimento por parte dos usuários.

Não é objetivo desse material, a apresentação da linguagem UML. Para maiores detalhes o leitor pode consultar, por exemplo, [BOO 98]. Apenas uma pequena parte dos recursos da UML são utilizados, os quais fazem parte do modelo de classes, cujos principais construtores gráficos são mostrados na figura abaixo. Por questões, também de simplificação, a parte da modelagem dinâmica (métodos) não está sendo abordada.



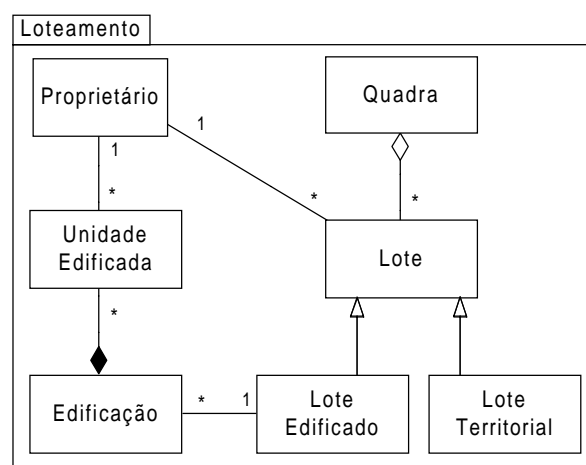
Os mecanismos de abstração e os respectivos elementos construtores da linguagem UML, que são utilizados na abordagem UML-GeoFrame, estão resumidos a seguir:

- ❑ **Classificação** – Nos modelos orientados a objetos, um fenômeno de interesse da aplicação é representado como um objeto, o qual possui uma estrutura, capaz de armazenar suas características (atributos) e um comportamento, descrito pelo conjunto de operações que podem ser realizadas com o objeto. Objetos semelhantes são modelados através da definição de uma classe, a qual especifica um conjunto de atributos que descrevem a estrutura e um conjunto de métodos (ou operações) que definem o comportamento dos objetos definidos pela classe.
- ❑ **Instanciação** – Um objeto pertencente a uma classe é dito ser uma instância desta classe.
- ❑ **Generalização** – Classes semelhantes podem ser agrupadas e descritas de forma generalizada. Neste caso, as propriedades da classe genérica (superclasse), ou seja, atributos, métodos e associações, são herdadas pelas classes que foram generalizadas (subclasses).
- ❑ **Especialização** – A especialização é o mecanismo inverso da generalização, no qual uma classe genérica (superclasse) pode ser especializada em uma ou mais classes específicas (subclasses), as quais herdam as propriedades da superclasse, além de novas propriedades poderem ser definidas.
- ❑ **Associação** – Representa os relacionamentos que podem haver entre objetos de diferentes classes. Multiplicidade é o nome dado à cardinalidade de uma associação.
- ❑ **Agregação** – Tipo especial de associação para representar relacionamentos entre objetos compostos e suas partes.
- ❑ **Composição** – Tipo especial de agregação na qual a existência do objeto composto (o todo) depende da existência dos objetos componentes (suas partes).

Além dos mecanismos de abstração relacionados anteriormente, a abordagem UML-GeoFrame utiliza dois elementos próprios da linguagem UML. São eles:

- ❑ Pacote – Usado para particionar um diagrama de classes através do agrupamento de elementos da linguagem UML.
- ❑ Estereótipo – Mecanismo de extensão dos construtores da linguagem UML. Permite que o projetista defina novos construtores e o utilize na elaboração de diagramas UML.

A figura abaixo ilustra um exemplo de diagrama de classe elaborado usando a UML. Algumas leituras que podem ser feitas neste diagrama são: uma Quadra é uma agregação de Lotes; Lote Edificado e Lote Territorial são especializações de Lote; uma Edificação é uma composição de Unidades Edificadas, as quais estão associadas a um proprietário cada; o diagrama de classes foi incluído em um pacote denominado Loteamento.

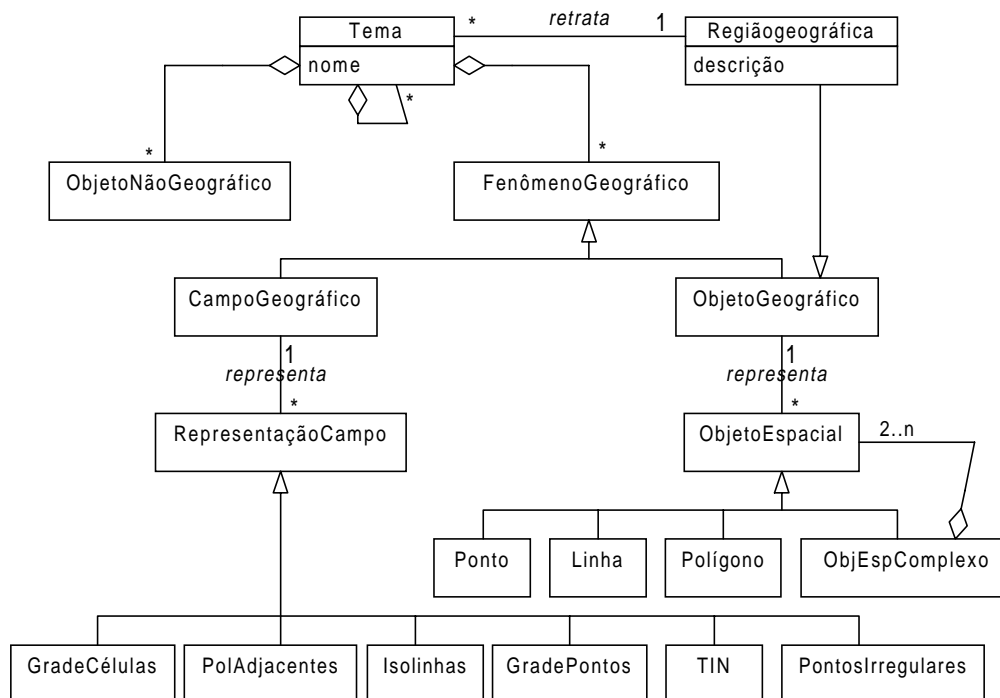


De forma semelhante ao modelo E-R, o diagrama de classes da linguagem UML também não é adequado à modelagem de banco de dados geográficos, por não atender a vários dos requisitos impostos pelas aplicações de SIG. A seguir é descrito o GeoFrame, o qual estende os construtores da linguagem UML através da definição de um conjunto simples de estereótipos para serem usados na modelagem de banco de dados geográficos.

GeoFrame

GeoFrame é um *framework* conceitual que fornece um diagrama de classes básicas para auxiliar o projetista nos primeiros passos da modelagem conceitual de dados de uma nova aplicação de SIG [LIS 99]. Um *framework* pode ser definido como um projeto genérico em um domínio que pode ser adaptado a aplicações específicas, servindo como um molde para a construção de aplicações. Esta definição fornece uma visão bem mais abrangente sobre a potencialidade de um *framework* do que as definições apresentadas por autores mais ligados à programação orientada a objetos. Por exemplo, Ralph Johnson define um *framework* como sendo “um projeto reutilizável de um programa, ou parte de um programa, expresso como um conjunto de classes” [JOH 92].

Como instrumento de reutilização, um *framework* não necessita estar implementado em uma linguagem de programação para fornecer a solução parcial a uma família de problemas. O *framework* GeoFrame (figura seguinte) foi elaborado sob esse enfoque mais genérico, onde o mesmo expressa a idéia de um projeto conceitual parcial para uma família de aplicações geográficas.



O GeoFrame foi definido de acordo com as regras do formalismo da orientação a objetos, utilizando a notação gráfica do diagrama de classes da linguagem UML. As subseções seguintes descrevem as classes do diagrama.

Tema e RegiãoGeográfica

As classes TEMA e REGIÃO GEOGRÁFICA formam a base de qualquer aplicação geográfica. Toda aplicação geográfica tem como objetivo o gerenciamento e a manipulação de um conjunto de dados para uma determinada região de interesse, constituindo um banco de dados geográfico. Assim, para cada região geográfica pode-se especificar uma coleção de temas.

Outra vantagem da utilização do conceito de temas, no esquema conceitual, é que ele funciona como um mecanismo para redução da complexidade em grandes esquemas. É comum existirem aplicações geográficas com centenas de entidades modeladas (ex.: SIGPROGB [LIS 97]). O uso de temas permite ao projetista dividir o esquema em subesquemas coesos, nos quais são agrupadas classes fortemente relacionadas entre si. Em grandes projetos, conjuntos de temas afins podem ser agrupados em um tema mais genérico, formando uma hierarquia de temas.

ObjetoNãoGeográfico e FenômenoGeográfico

Em um banco de dados geográficos existem, além dos dados referentes àqueles fenômenos georreferenciados, com ou sem representação geoespacial, objetos convencionais presentes em qualquer sistema de informação. Alguns objetos não possuem referência com relação a uma localização geográfica, sendo tratados, genericamente, como instâncias de subclasses da classe OBJETONÃO GEOGRÁFICO.

A classe abstrata FENÔMENO GEOGRÁFICO generaliza qualquer fenômeno cuja localização em relação à superfície terrestre seja considerada. Por exemplo, um lote de terra é uma instância de FENÔMENO GEOGRÁFICO caso seus atributos espaciais estejam representados no banco de dados.

Fenômenos geográficos e objetos não geográficos podem estar relacionados entre si (associação *relacionaCom*), como no caso em que "todo lote pertence a um proprietário". A modelagem desse tipo de relacionamento permite que os dados armazenados em um SIG estejam integrados com os demais sistemas de informação da organização.

CampoGeográfico e ObjetoGeográfico

Fenômenos geográficos são percebidos, na realidade, segundo as visões dicotômicas de campo e de objeto [GOO 92]. Essas duas visões acarretam diferentes maneiras de modelagem dos fenômenos geográficos. As classes CAMPOGEOGRÁFICO e OBJETOGEORÁFICO especializam a classe FENÔMENOGEOGRÁFICO, permitindo ao projetista especificar, de forma distinta porém integrada, os campos e os objetos geográficos, respectivamente.

A classe abstrata OBJETOGEORÁFICO é uma generalização de todas as classes do domínio que são percebidas na visão de objetos. Neste caso estão incluídas aquelas classes que representam fenômenos geográficos que podem ser individualizados, ou seja, que possuem identidade própria e suas características podem ser descritas através de atributos.

A classe abstrata CAMPOGEOGRÁFICO generaliza os fenômenos que se enquadram na visão de campo. Campos geográficos são modelados como funções sobre uma variável. Com base nesta definição, um campo geográfico pode ser modelado conceitualmente como subclasse de CAMPOGEOGRÁFICO. Alguns tipos de campos geográficos, os chamados categóricos, possuem uma associação especial com uma classe que representa a imagem da função de mapeamento do campo, uma subclasse de OBJETONÃOGEORÁFICO.

ObjetoEspacial

É uma classe abstrata cujas subclasses formam o conjunto mínimo de construtores para a especificação do componente espacial dos fenômenos geográficos na visão de objetos. Alguns fenômenos geográficos podem apresentar dimensão espacial complexa, ou seja, composta por outros objetos espaciais (ex.: um arquipélago). Assim, a classe OBJETOESPACIAL, possui as subclasses (PONTO, LINHA, POLÍGONO e OBJESPACIALCOMPLEXO), que constituem o conjunto mínimo de classes necessárias para a fase do projeto conceitual.

RepresentaçãoCampo

Os aspectos espaciais de um campo geográfico são abstraídos de forma diferente dos aspectos espaciais de um objeto geográfico. Para especificação do componente espacial de um campo geográfico são usadas as subclasses da classe REPRESENTAÇÃOCAMPO. São elas: GRADECÉLULAS, POLADJACENTES, ISOLINHAS, GRADEPONTOS, TIN e PONTOSIRREGULARES.

Um mesmo campo geográfico pode ter seu componente espacial abstraído de diferentes formas, ou seja, através de mais de um desses modelos. Por exemplo, o campo *Temperatura* pode ser abstraído por meio de um conjunto de pontos irregularmente distribuídos ou por meio de isolinhas. Problema semelhante ocorre com os objetos geográficos cujos componentes espaciais podem ser percebidos, ora por formas alternativas, ora por formas duplas, dependendo de aspectos como a escala com a qual se pretende capturar a forma espacial de cada fenômeno. A possibilidade de haver múltiplas representações é sinalizada no GeoFrame, através das associações (1:n), denominadas *representa*.

Processo de Modelagem UML-GeoFrame

A abordagem UML-GeoFrame permite a solução da maioria dos requisitos de modelagem apresentados na Seção 4.2. Um esquema conceitual de dados geográficos construído com base no GeoFrame inclui, por exemplo, a modelagem dos aspectos espaciais da informação geográfica e a diferenciação entre objetos convencionais e objetos/campos geográficos.

O processo de modelagem com base na abordagem UML-GeoFrame envolve três etapas:

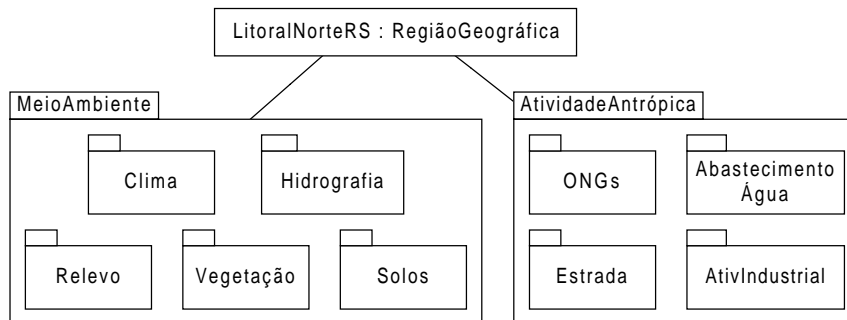
- ❑ Passo 1: identificar temas e sub-temas para cada área geográfica;
- ❑ Passo 2: para cada tema, elaborar o sub-diagrama de classes. Associar classes de diferentes temas;
- ❑ Passo 3: modelar o componente espacial para cada fenômeno geográfico identificado.

A seguir, cada passo é apresentado em maiores detalhes.

Passo 1 – Identificar temas por região geográfica

Uma das primeiras escolhas a ser realizada, no desenvolvimento de uma nova aplicação de SIG, é a escolha da área geográfica para a qual os dados serão coletados e analisados. A seguir, partindo-se dos objetivos iniciais da nova aplicação, deve ser feita a identificação dos principais temas para os quais os dados serão coletados e mantidos no banco de dados geográfico.

Na abordagem UML-GeoFrame, temas são representados através do construtor Pacote da linguagem UML. Os diversos temas identificados na aplicação são modelados através da elaboração de um diagrama hierárquico de temas para cada região geográfica da aplicação. A figura abaixo ilustra o diagrama de temas do sistema de gerenciamento costeiro da região norte do litoral do estado do Rio Grande do Sul [LIS 97].

**Passo 2: Construir diagramas de classes por tema**

A partir da análise dos requisitos da aplicação, um diagrama de classe deve ser elaborado para cada tema especificado. Com base no diagrama de classes do *framework* GeoFrame, as classes da aplicação devem ser modeladas como subclasses de uma das classes OBJETO GEOGRÁFICO, CAMPO GEOGRÁFICO ou OBJETO NÃO GEOGRÁFICO.

O problema que ocorre é o surgimento de um grande número de relacionamentos de especialização entre as diversas classes da aplicação e estas três classes do GeoFrame. A solução proposta pela abordagem UML-GeoFrame é a substituição destes relacionamentos de especialização por estereótipos. A figura abaixo ilustra o conjunto total de estereótipos utilizados na abordagem UML-GeoFrame.

<i>Fenômeno geográfico e Objeto convencional</i>	<i>Componente espacial de objetos geográficos</i>	<i>Componente espacial de campos geográficos</i>
△ Objeto geográfico ▴ Campo geográfico △ Objeto não geográfico	● Ponto — Linha □ Polígono * Obj. espacial complexo	●● Pontos irregulares ●●● Grade de pontos □ Polígonos adjacentes ⊖ Isolinhas □ Grade de células ⊗ TIN
<<função>> função categórica		

Estereótipos do *framework* GeoFrame

O primeiro conjunto de estereótipos (*Fenômeno geográfico e Objeto convencional*) é usado para diferenciar os dois principais tipos de objetos pertencentes a um banco de dados geográficos: (1) *Fenômeno geográfico*, que é especializado em *Objeto geográfico* (△) e *Campo geográfico* (▴), segundo as duas formas de percepção dos fenômenos geográficos; (2) *Objeto não geográfico*, ou seja, objetos sem representação espacial são modelados de forma tradicional e são identificados através do estereótipo (△).

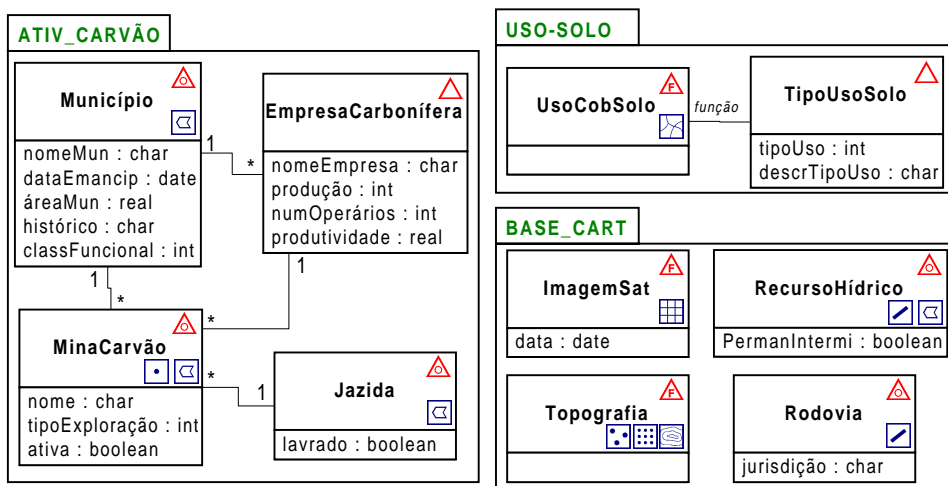
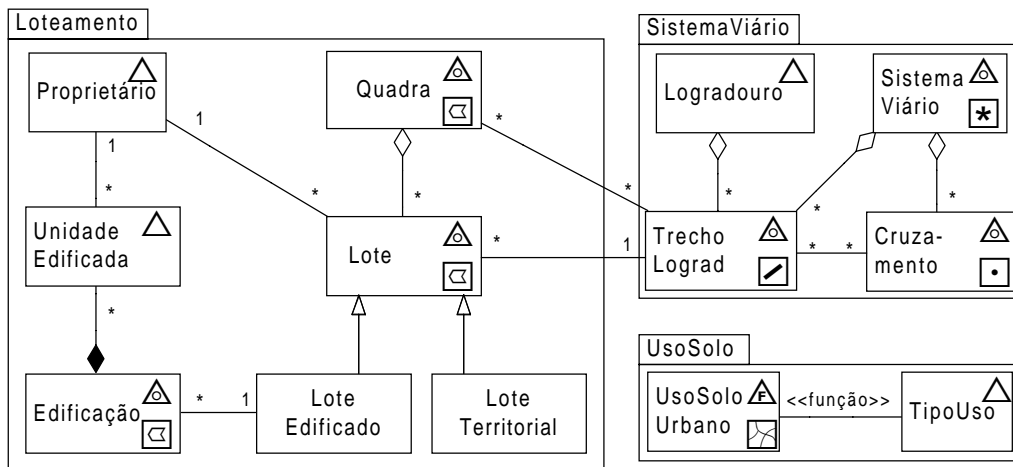
Passo 3: Modelar o componente espacial dos fenômenos geográficos

O terceiro passo, que pode ser executado simultaneamente ao segundo passo, implica na especificação dos possíveis tipos de representação espacial dos fenômenos geográficos. De acordo com o diagrama de classes do GeoFrame, o componente espacial dos campos e objetos geográficos é modelado através de associações (*Representa*) entre as classes da aplicação e as subclasses de REPRESENTAÇÃOCAMPO e OBJETOESPACIAL, respectivamente. Novamente, a fim de evitar a poluição visual do diagrama de classes, estas associações são substituídas por estereótipos.

O segundo e o terceiro conjuntos de estereótipos apresentados na figura anterior (*Componente espacial de objetos geográficos* e *Componente espacial de campos geográficos*) são usados para a modelagem do componente espacial de fenômenos segundo as visões de objeto e de campo, respectivamente. A existência de múltiplas representações é modelada através da combinação de dois ou mais estereótipos em uma mesma classe. Por exemplo, uma classe *Município* pode ter duas formas de abstração de seu componente espacial, pontual e/ou poligonal, o que é especificado pelo par de estereótipos (□□).

Por último, o estereótipo <<função>> é usado para caracterizar um tipo especial de associação que ocorre quando da modelagem de campos categóricos. Segundo Chrisman [CHR 97], numa estrutura de cobertura categórica o espaço é classificado em categorias mutuamente exclusivas, ou seja, uma variável possui um valor do tipo categoria em todos os pontos dentro de uma região (ex.: tipos de solos).

As figuras abaixo apresentam dois exemplos de diagrama de classes de aplicações de SIG nas áreas de aplicação de gestão urbana e monitoramento ambiental, respectivamente, as quais foram modeladas segundo a abordagem UML-GeoFrame.



4.4 Projeto Lógico de Banco de Dados Geográficos

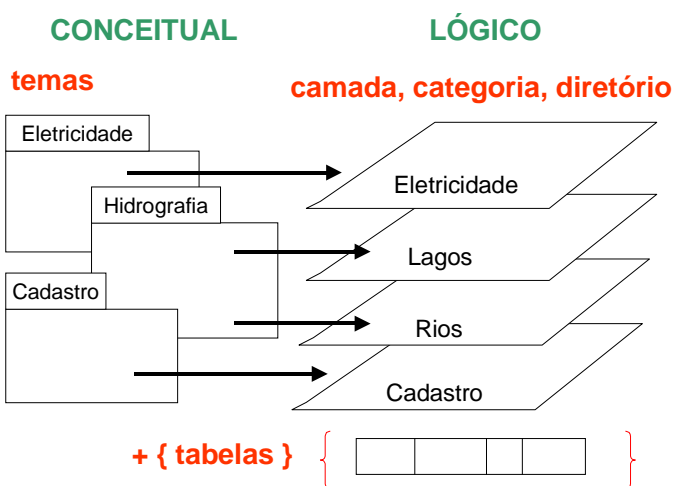
As mesmas regras de transformação de esquema conceitual em esquema lógico, apresentadas na Seção 3, podem ser aplicadas também em banco de dados geográficos. No entanto, as regras apresentadas anteriormente não são suficientes, devido à necessidade de transformação do componente espacial dos fenômenos geográficos.

Ao contrário dos SGBD relacionais, que têm como base um modelo bem definido, os softwares de SIG não possuem um modelo de dados único, ou seja, cada SIG tem seu próprio modelo lógico e suas próprias estruturas de armazenamento. Isto faz com que não exista um conjunto único de regras de transformação para bancos de dados geográficos.

O consórcio *Open GIS (Open GIS Consortium - OGC)*, fundado em 1994 por fornecedores de software, companhias de telecomunicações, universidades, órgãos governamentais, entre outros, é uma organização internacional que está criando novas padronizações técnicas e comerciais para garantir interoperabilidade em SIG [BUE 98]. O consórcio busca criar uma especificação de software e novas estratégias empresariais, a fim de tornar os sistemas de geoprocessamento abertos e integrar os dados geográficos e as operações necessárias para manipular esses dados. No momento em que os softwares de SIG estiverem integrados ao modelo de dados *Open GIS*, aí será possível a unificação das regras de transformação.

Nesta seção é apresentado um conjunto genérico de diretrizes para transformação de esquemas conceituais, elaborados segundo a abordagem UML-GeoFrame, para um esquema lógico espaço-relacional, encontrado na maioria dos SIG que são construídos segundo uma arquitetura dual.

Normalmente, numa visão simplificada, pode-se dizer que os SIG organizam os dados descritivos em SGBD relacionais e os dados espaciais através de um conjunto de camadas, também denominadas de *layer*, cobertura ou plano de informação, dependendo do software de SIG em questão. Assim, um esquema conceitual UML-GeoFrame deve ser transformado em um conjunto de tabelas e um conjunto de camadas com dados geo-espaciais. A figura seguinte ilustra esta idéia.

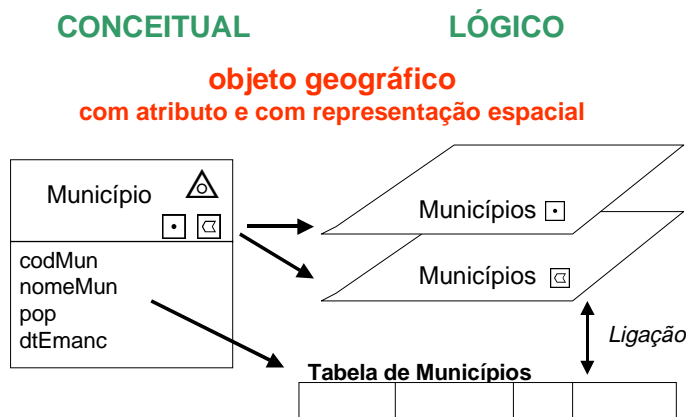


Inicialmente, as regras apresentadas para transformação de esquemas E-R em esquemas relacionais devem ser aplicadas para a transformação de classes e associações em tabelas. A seguir, são apresentadas as diretrizes para as diversas possibilidades de transformação do componente espacial dos fenômenos geográficos.

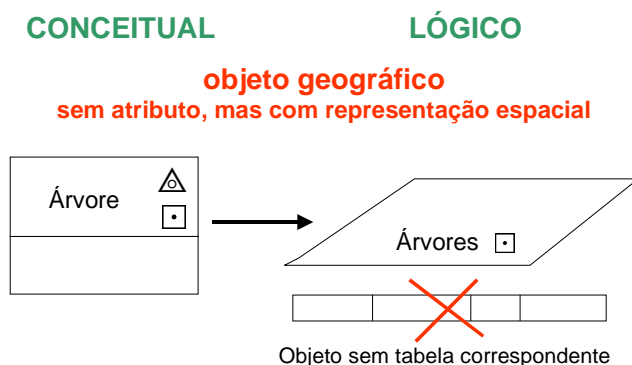
Transformando Objetos Geográficos (△)

Existem duas possibilidades para transformar o componente espacial das classes da aplicação modeladas como subclasses da classe OBJETO GEOGRÁFICO, do GeoFrame. A primeira é a transformação mais comum, na qual o objeto geográfico possui atributos descritivos e representação espacial. Neste caso, os atributos descritivos são transformados em colunas de uma tabela correspondente no SGBD, enquanto cada tipo de representação espacial dá origem a uma camada (ou *layer*) que contém a forma, a localização e, em alguns casos, os relacionamentos espaciais dos fenômenos geográficos representados pelas instâncias da classe sendo transformada.

É importante observar que uma camada no nível lógico pode vir a ser implementada através de múltiplas camadas físicas, dependendo do tipo de estrutura de dados utilizada no SIG em questão. Por exemplo, a figura abaixo ilustra a transformação da classe Município. Em um SIG como o MGE da Intergraph, além das camadas para armazenamento das formas poligonais e pontuais, seria adicionada uma camada específica para armazenar a localização da toponímia, ou seja, do texto formatado contendo o nome dos municípios.



A segunda possibilidade de transformação é quando o objeto geográfico não possui atributos descritivos, mas apenas representação espacial. Neste caso, não existem atributos descritivos para serem transformados. A transformação do componente espacial é idêntica ao caso anterior. A figura abaixo ilustra um exemplo típico.

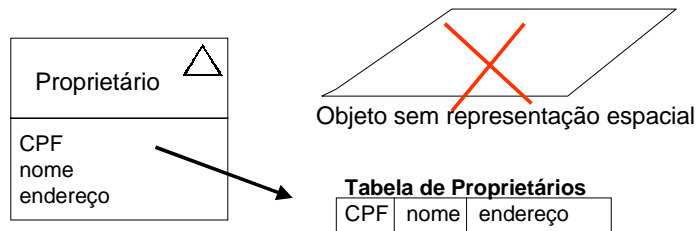


Transformando Objetos Não Geográficos (△)

A transformação de objetos não geográficos, ou seja, possui apenas atributos descritivos, é feita de forma idêntica à transformação de entidades apresentada na Seção 3.3. A figura seguinte ilustra a transformação da classe Proprietário em uma tabela do SGBD.

CONCEITUAL LÓGICO

objeto não geográfico

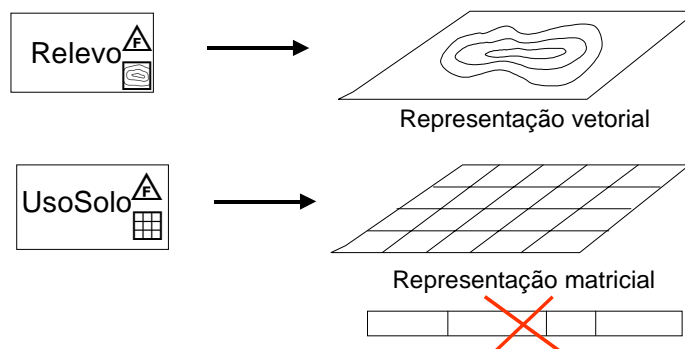


Transformando Campos Geográficos (A)

A transformação de campos geográficos, na maioria da vezes, é feita através da escolha de um tipo de representação espacial para o campo sendo transformado. A figura abaixo ilustra dois exemplos de campos geográficos que não geram informações no SGBD.

CONCEITUAL LÓGICO

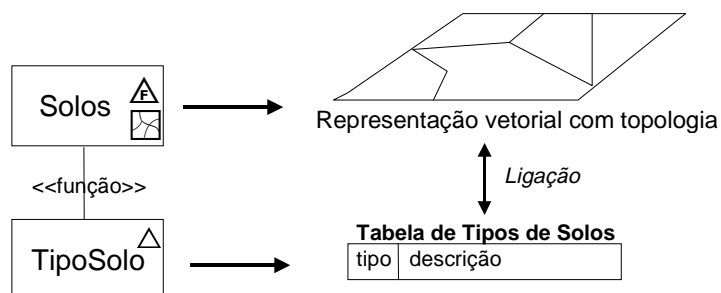
campo geográfico



Alguns campos geográficos, no entanto, podem estar diretamente relacionados a algum tipo de legenda. Na abordagem UML-GeoFrame este tipo de associação é feita através de um estereótipo especial identificado por <<função>>. A figura abaixo ilustra um exemplo típico de campo geográfico que possui uma ligação especial com uma tabela correspondente no SGBD.

CONCEITUAL LÓGICO

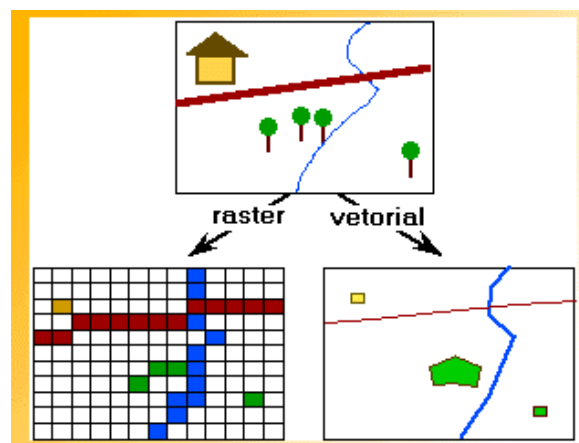
campo geográfico



Em alguns SIG a tabela associada a este tipo de campo geográfico pode não estar necessariamente armazenada no SGBD, mas sim, ser mantida junto com a estrutura dos dados espaciais. Exemplos de SIG que trabalham desta forma incluem o Idrisi e o Spring.

Um aspecto importante a ser observado é que estes exemplos servem para chamar a atenção do leitor sobre a diferença existente entre o nível de abstração em que os fenômenos geográficos são classificados nas visões de campo e de objetos, e o nível de abstração mais baixo, no qual as estruturas de dados são classificadas como sendo vetoriais ou matriciais (*raster*). Observe na figura anterior que o campo geográfico está sendo transformado em uma estrutura vetorial com topologia. Este mesmo campo geográfico poderia ter sido transformado para uma estrutura de dados matricial, onde cada célula estaria associado a um determinado tipo de solo.

Da mesma forma, o componente espacial de um fenômeno geográfico percebido na visão de objetos (ex.: estados brasileiros) pode ser transformado em uma camada física implementada tanto numa estrutura de dados matricial como numa estrutura de dados vetorial (com ou sem topologia). A figura abaixo ilustra fenômenos geográficos percebidos na visão de objetos (rio, estrada, etc) podendo ser implementados através dos dois tipos de representação espacial.



4.5 Estudo de Caso

O governo de um novo país africano, chamado Zimbabum, enviou seus técnicos ao Brasil para serem treinados em geoprocessamento. Uma de suas necessidades é o auxílio na estruturação do banco de dados geográficos. A seguinte descrição foi obtida a partir de uma conversa inicial.

O país tem como língua oficial o português, mas existem mais de 20 línguas faladas por diferentes tribos. Atualmente, todo o país está dividido em 14 grandes propriedades, que pertenciam aos antigos colonizadores. Todas as propriedades agora pertencem ao governo central, o qual deseja dividi-las, fornecendo um pedaço de terra a cada tribo, segundo os seguintes critérios:

- ❑ As poucas cidades existentes serão transformadas em pólos industriais e comerciais, devendo ter os seus limites estabelecidos num raio de 20km da mancha urbana atual. Esta área será denominada município. Cada município desenvolve diversos tipos de atividades. É importante manter o valor bruto da produção anual de cada município para cada tipo de atividade;
- ❑ Cada propriedade será dividida entre as tribos que ali se encontram, bem como para grupos de imigrantes com mais de 800 membros. Cada pedaço de terra, denominado fazenda, terá entre 20 e 30 Km². Tribos com mais de 1000 membros poderão receber mais de uma fazenda, que devem ser adjacentes;
- ❑ Cada fazenda deve ter: no mínimo 3 nascentes de rios ou 2 nascentes e um lago; 40% de mata nativa; 50% de área com menos de 30 graus de declividade; 50% de terras férteis, sem considerar áreas até 50m de leito de recursos hídricos para rios de categoria 1 e até 20m para rios de categoria 2 ou 3;

- As fazendas não distribuídas permanecerão sendo do governo para futuras partilhas ou incentivo à imigração. As atuais propriedades serão administradas por governos regionais, que receberão 50% das fazendas não distribuídas, além de administrar as cidades que se encontram em seus limites. Cada governo regional terá uma cidade que será sua capital;
- Será feito o levantamento das trilhas e caminhos existentes, com vistas a serem abertas novas estradas, bem como o levantamento dos principais rios que servirão como meio de transporte fluvial.

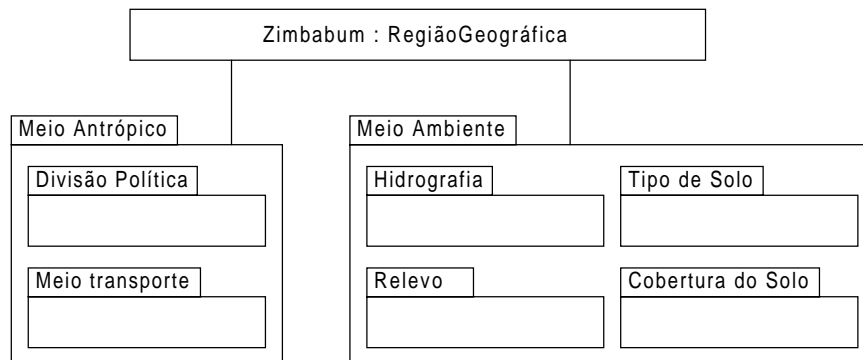
Problemas

Usando a notação UML-GeoFrame, elaborar:

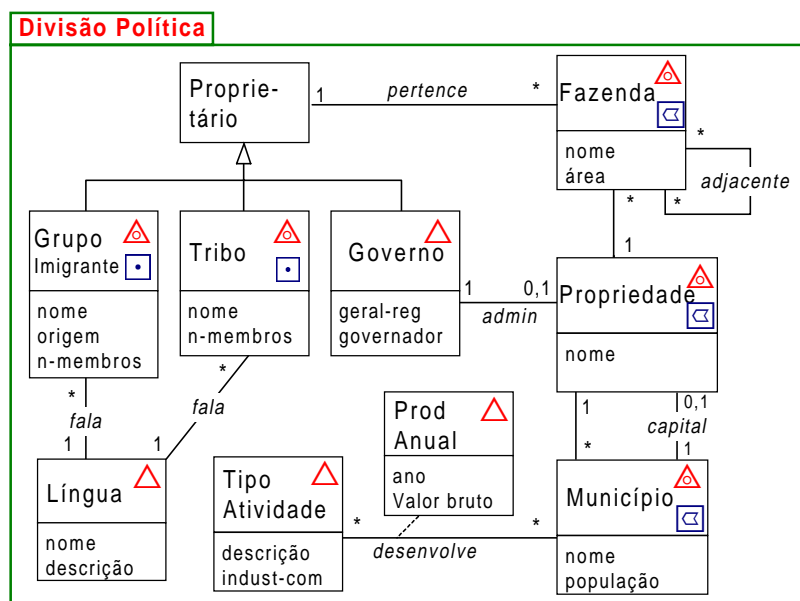
- 1) O diagrama de temas;
- 2) O esquema conceitual de dados;
- 3) O esquema lógico descritivo (esquema relacional) e o esquema lógico espacial (planos de informação e tipo de representação espacial).

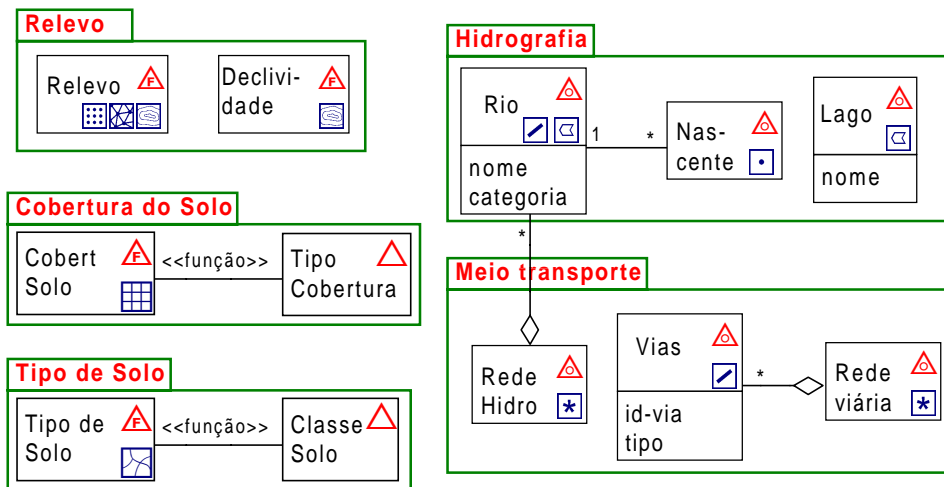
Solução

- 1) A figura abaixo apresenta o diagrama de temas elaborado.



- 2) As figuras seguintes apresentam os diagramas de classes para cada tema identificado.





3) O esquema lógico descritivo (esquema relacional) e o esquema lógico espacial (planos de informação e tipo de representação espacial) ficam como sugestão de exercício para o leitor. Utilize as regras de transformação apresentadas nas seções 3.3 e 4.4.

5 BIBLIOGRAFIA

- [ARO 89] ARONOF, S. **Geographic Information Systems: a management perspective**. Canada: WDL Publications, 1989.
- [BED 96] BÉDARD, Y. et al. Adapting data models for the design of spatio-temporal databases. **Computers, Environment, and Urban Systems**, Bristol-UK, v.20, n.1, 1996.
- [BED 99] BÉDARD, Y. Visual modeling of spatial databases: towards spatial PVL and UML. To appear in **Geomatica**, June, 1999.
- [BUE 98] BUEHLER, K.; MCKEE, L. **The OpenGIS guide**. Massachusetts, USA. Available at: <<http://www.OpenGIS.org/techno/guide.html>>. Jun, 1998.
- [BOO 98] BOOCH, G.; JACOBSON, I.; RUMBAUGH, J. **The Unified Modeling Language User Guide**. Addison-Wesley, 1998.
- [BOR 97] BORGES, K. A. V. **Modelagem de Dados Geográficos: uma extensão do modelo OMT para aplicações geográficas**. Belo Horizonte: Fundação João Pinheiro, 1997. Dissertação de Mestrado.
- [BRO 84] BRODIE, M. L. On the development of data models. In: BRODIE, M. L.; MYLOPOULOS, J.; SCHMIDT, J.W. (Eds.). **On Conceptual Modeling**. New York: Springer-Verlag, 1984. p.19-48.
- [CHE 76] CHEN, P.P.S. The entity-relationship model: Towards a unified view of data. **ACM Trans. Database System**, New York, n.1, 1976.
- [CHR 97] CHRISMAN, N. **Exploring Geographic Information Systems**. New York: John Wiley & Sons, 1997.
- [COD 79] CODD, E. F. Extending the database relational model to capture more meaning. **ACM Trans. Database Systems**, New York, v.4, n.4, p.397-434, 1979.
- [COU 92] COUCLELIS, H. People manipulate objects (but cultivate fields): beyond the raster-vector debate in GIS. In **Theories and methods of spatial-temporal reasoning in geographic space**. Berlin: Springer Verlag, 1992. p.65-77. (Lecture Notes in Computer Science 639).
- [DAT 81] DATE, C. J. **An Introduction to Database Systems**. 3. ed. Reading: Addison-Wesley, 1981.
- [DBM 94] DBMS On Line. **A new direction in DBMS: Montage Software's Dr. Michael R. Stonebraker takes the wraps off his new object-relational DBMS**. DBMS Interview, 1994.
- [ELM 94] ELMASRI, R.; NAVATHE, S. B. **Fundamentals of Database Systems**. 2.ed. Menlo Park, CA: Addison-Wesley, 1994.
- [EUR 96] EUROPEAN COMMITTEE FOR STANDARDIZATION. **Geographic information – data description – conceptual schema language**. Brussels: CEN, 1996. (Report CR 287005).
- [GOO 92] GOODCHILD, M. F., Geographical data modeling. **Computers & Geosciences**, London, v.18, n.4, p.401-408, 1992.
- [HAD 96] HADZILACOS, T.; TRYFONA, N. Logical data modelling for geographical applications. **International Journal of Geographical Information Systems**, London, v.10, n.2, p.179-203, 1996.
- [HAD 97] HADZILACOS, T.; TRYFONA, N. An extended entity-relationship model for geographic applications. **SIGMOD Record**, V.26, N.3, 1997.

- [JOH 92] JOHNSON, R. E. Documenting frameworks using patterns. In: OBJECT-ORIENTED PROGRAMMING SYSTEMS, LANGUAGES AND APPLICATIONS CONFERENCE - OOPSLA, 1992, Vancouver. **Proceedings...** Vancouver: ACM, 1992.
- [KOS 97] KÖSTERS, G. et al. GIS-Application Development with GeoOOA. **International Journal of Geographical Information Science**, London, v.11, n.4, 1997.
- [LIS 97] LISBOA FILHO, J.; IOCHPE, C.; GARAFFA, I. M. Modelos conceituais de dados para aplicações geográficas: uma experiência com um SIG interinstitucional. In: SIMPÓSIO BRASILEIRO DE GEOPROCESSAMENTO, 4., São Paulo, 1997. **Anais...** São Paulo: UPUSP, 1997.
- [LIS 99] LISBOA FILHO, J.; IOCHPE, C. Specifying analysis patterns for geographic databases on the basis of a conceptual framework. In: ACM SYMPOSIUM ON ADVANCES IN GEOGRAPHIC INFORMATION SYSTEMS, 7., 1999, Kansas City, USA. **Proceedings...** Kansas City: ACM Press, 1999.
- [LIS 99a] LISBOA FILHO, J.; IOCHPE, C. Um estudo sobre modelos conceituais de dados para projeto de bancos de dados geográficos. **Revista IP-Informática Pública**, Belo Horizonte, v.1, n.2, 1999.
- [LIS 00] LISBOA FILHO, J.; IOCHPE, C.; HASENACK, H.; WEBER, E. J. Modelagem conceitual de banco de dados geográficos: o estudo de caso do projeto PADCT/CIAMB. In: UFRGS/CENTRO DE ECOLOGIA. **Energia e Meio Ambiente**. Porto Alegre: Editora da Universidade, 2000.
- [PAR 98] PARENT, C. et al. Modeling spatial data in the MADS conceptual model. In: INTERNATIONAL SYMPOSIUM ON SPATIAL DATA HANDLING. 1998, Vancouver, Canada. **Proceedings...** Vancouver, 1998.
- [PEU 84] PEUQUET, D. J. A conceptual framework and comparison of spatial data models. **Cartographica**, [S.l.], n.21, p66-113, 1984.
- [PEU 95] PEUQUET, D. J. An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. **International Journal of Geographical Information Systems**, London, v.9, n.1, p.7-24, 1995.
- [PIR 97] PIRES, F. **Um ambiente computacional para modelagem de aplicações geográficas**. Campinas: Unicamp, Instituto de Computação, 1997. Tese de Doutorado.
- [PRE 87] PRESSMAN, R.S. **Software Engineering: a practitioner's approach**. 2.ed. New York: McGraw-Hill, 1988.
- [SHK 97] SHEKHAR, S. et al. Data models in geographic information systems. **Communications of the ACM**, New York, v.40, n.4, 1997.
- [ULL 82] ULLMAN, J. D. **Principles of Database Systems**. 2. ed. Rockville: Computer Science Press, 1982.
- [WOR 95] WORBOYS, M.F. **GIS: A Computing Perspective**. London: Taylor and Francis, 1995.