Glossário

Capítulo 3

**Aplicação de um banco de dados:** refere-se a um banco de dados em particular e aos programas a ele associados, que implementam consultas e atualizações.

**Programas de aplicação:** oferecem interfaces gráficas, que utilizam formulários e menus para interagir com os usuários finais das aplicações. Tradicionalmente, o projeto e teste dos programas de aplicação são tratados no domínio da Engenharia de Software.

**Modelo entidade-relacionamento:** é um modelo de dados conceitual de alto nível muito popular, empregado no projeto conceitual de um banco de dados. Muitas ferramentas de projeto aplicam seus conceitos. Descreve os dados como *entidades, relacionamentos* e *atributos.*

**Diagrama ER:** é a notação diagramática associada ao modelo ER.

**UML (Linguagem de Modelagem Universal):** metodologia de modelagem de objetos popular no projeto e engenharia de software. Especifica o projeto detalhado dos módulos de software e suas interações, utilizando vários tipos de diagrama.

**Diagramas de classe:** parte das metodologias UML, similar, em muitos aspectos, aos diagramas ER. Os diagramas de classe especificam a estrutura do esquema do banco de dados e definem as *operações* em objetos. As operações podem ser utilizadas para especificar os *requisitos funcionais* durante o projeto do banco de dados.

**Levantamento e análise de requisitos:** etapa em que o projetista entrevista o possível usuário do banco de dados para entender e documentar seus *requisitos de dados.*

**Requisitos de dados:** é o registro conciso dos requisitos do usuário, especificados em formulário de forma detalhada e completa.

**Requisitos funcionais:** definidos em paralelo à especificação dos requisitos de dados, consistem em *operações* que serão empregadas no banco de dados, incluindo recuperações e atualizações. [Em projeto de software, é comum o uso de diagramas de fluxo de dados, diagramas de sequencia, cenários e outras técnicas para a especificação de requisitos funcionais.]

**Projeto conceitual:** fase em que se cria o *esquema conceitual.*

**Esquema conceitual:** descrição concisa dos requisitos de dados dos usuários, incluindo descrições detalhadas de tipos entidade, relacionamento e restrições (expressos usando conceitos fornecidos pelo modelo de dados de alto nível).

**Projeto lógico:** também conhecida como *mapeamento do modelo de dados,* consiste em transformar um modelo de dados de alto nível em um modelo de dados de implementação. Cada SGBD utiliza um modelo de dados de implementação, como o relacional.

**Entidade:** objeto básico que o modelo ER representa, ‘algo’ do mundo real com existência independente. Pode ser um objeto de existência física (casa) ou conceitual (empresa). Cada entidade possui *atributos.*

**Atributo:** propriedade particular que descreve uma entidade. Possui um valor para uma dada entidade. Há vários tipos de atributos: simples versus composto, univalorado versus multivalorado, e armazenado versus derivado.

**Atributos simples (atômicos):** não são divisíveis.

**Atributos compostos:** podem ser divididos em partes menores e podem formar uma hierarquia. Seu valor é a concatenação dos valores componentes de seus atributos simples.

**Atributos monovalorados:** possuem um único valor para uma dada entidade.

**Atributos multivalorados:** *números de valores* diferentes, como por exemplo Cor em um carro de duas cores. Devem ter um limite inferior e superior.

**Atributos armazenados:** guardados.

**Atributos derivados:** podem ser inferidos através de dois ou mais atributos armazenados, como por exemplo, Idade, que pode ser determinado pela data de nascimento e a data atual. Representados no diagrama EP por elipses pontilhadas.

**Valores null:** pode significar valor não aplicável para um dado atributo ou valor ‘desconhecido’. O valor desconhecido pode ser classificado em dois casos: existe, mas está faltando ou não sabe se existe.

**Atributos complexos:** atributos compostos e multivalorados aninhados de maneira arbitrária, separando componentes por vírgulas e mostrando os atributos entre chaves.

**Tipo entidade:** define uma coleção (ou conjunto) de entidades que possuem os mesmos atributos. Cada tipo entidade no banco de dados é descrito por seu nome e atributos.

**Conjunto de entidades:** coleção de todas as entidades de um tipo entidade particular. É representado no diagrama ER como uma caixa retangular, incluindo seu nome.

**Chave (ou restrição de unicidade):** atributo de um tipo entidade cujos valores são distintos para cada uma das entidades do conjunto. É chamado *atributo-chave.*

**Tipo entidade fraca:** um tipo entidade que não possui chave.

**Domínio:** determina o conjunto de valores válidos para os atributos de cada entidade. Não são mostrados no diagrama ER. São determinados utilizando-se *tipos de dados* presentes em linguagens de programação.

**Relacionamentos implícitos:** quando um atributo de uma entidade refere-se à outra entidade. No projeto inicial dos tipos entidade, os relacionamentos são capturados na forma de atributos.

**Tipo relacionamento:** define um conjunto de associações entre n tipos entidade. Matematicamente, o conjunto de relacionamentos é um conjunto de *instâncias de relacionamento*. Nos diagramas ER, tipos relacionamento são mostrados como caixas em forma de losango.

**Grau de relacionamento:** número de entidades que participam desse relacionamento. Grau dois é chamado *binário*, grau três *ternário*.

**Papel:** o que uma entidade participante executa em cada instância de relacionamento, ajuda a explicar o significado do relacionamento.

**Relacionamentos recursivos:** o mesmo tipo entidade participa mais de uma vez em um tipo relacionamento em papéis diferentes. *Nomes dos papéis* não são tecnicamente necessários, entretanto, nesses casos, tornam-se essenciais para definir o sentido de cada participação (considere o relacionamento Supervisão).

**Restrições em tipos relacionamento:** limitam a possibilidade de combinações de entidades que podem participar do conjunto de relacionamentos correspondente. São determinadas pela situação (e regras) do minimundo que as representa.

**Razão de cardinalidade:** para relacionamentos binários, especifica o número máximo de instâncias de relacionamento em que uma entidade pode participar. As razões de cardinalidade possíveis são 1:1, 1:N, N:1 e M:N. São representadas nos diagramas ER pela exibição de 1, M ou N nos losangos.

**Restrição de participação:** também chamada *restrição de cardinalidade mínima,* determina se a existência de uma entidade depende de sua existência relacionada à outra entidade, pelo tipo relacionamento. Há dois tipos: *total* e *parcial*.

**Dependência de existência:** outra forma de nos referirmos à participação total. Representada com linha dupla.

**Restrições estruturais:** referenciam à restrição de participação e à razão de cardinalidade, juntas.

**Atributos de tipos relacionamento:** similares àqueles dos tipos entidade. Atributos de tipos relacionamento 1:1 ou 1:N podem ser migrados para um dos tipos entidade participantes, sendo que no em 1:N um atributo pode ser migrado apenas para o tipo entidade do lado N. Cabe ao projetista decidir onde os atributos devem ser colocados.

**Tipo entidade fraca:** tipo entidade que não possui seu próprio atributo-chave, em contraste ao *tipo entidade regular*. Entidades fracas são identificadas por estarem relacionadas, por meio da combinação com valores de seus atributos, a um *tipo identidade identificador* ou *tipo entidade proprietária*. Tipo entidade fraca sempre possui uma *restrição de participação total.* Normalmente possuem uma *chave parcial*. São identificados em diagramas ER pela borda dupla no retângulo e no losango.

**Chave parcial:** conjunto de atributos que identifica, de modo exclusivo, as entidades fracas que estão relacionadas a uma mesma entidade proprietária. No pior caso, será composta por todos os atributos da entidade fraca.

Capítulo 4

**Modelo entidade-relacionamento estendido:** um modelo ER expandido que inclui conceitos de *modelagem semântica de dados,* permitindo atender a requisitos mais complexos.

**Subclasse:** (de um tipo entidade) subgrupo significativo de um tipo entidade, que necessita ser representado explicitamente. Os membros da subclasse representam a mesma entidade do mundo real que qualquer membro da superclasse correspondente, mas em um *papel específico.*

**Superclasse:** um tipo entidade que agrupa subclasses.

**Relacionamento classe/subclasse:** o relacionamento entre uma superclasse e qualquer de suas subclasses.

-🡪 parei às 04:27 do dia 11/12/2009 na página 62, item 4.2