

MODELAGEM CONCEITUAL

1. Introdução ao Modelo Entidade-Relacionamento (MER)

Conforme comentado no capítulo anterior, o sistema de banco de dados deve prover uma visão abstrata de dados aos usuários, isolando-os de certos detalhes do BD. A arquitetura desta abstração se dá em três níveis. O mais externo, o nível de visões do usuário, descreve partes do banco que serão visualizadas pelos usuários. No nível intermediário, tem-se o nível conceitual (ou lógico), que descreve quais os dados estão armazenados e seus relacionamentos. Finalmente, no nível mais baixo, está o nível físico, descrevendo a forma como os dados estão realmente armazenados.

O foco deste capítulo é um aprofundamento das tarefas do nível conceitual, através da modelagem conceitual dos dados referentes ao negócio. O termo *negócio* refere-se ao problema em questão que se deseja realizar uma modelagem. A intenção de armazenar informações de alunos numa escola, por exemplo, sugere que o negócio seja "Registro acadêmico". Num outro exemplo, o negócio "Instituição financeira" poderá ser modelado tendo dados de correntistas e saldos.

Razões para a criação do modelo conceitual:

- Descreve exatamente as informações necessárias ao negócio. Para a modelagem, todas as regras do negócio deverão ser conhecidas e, cabe ao projetista, traduzi-las em informações relevantes ao banco;
- Facilita a discussão, seja entre o projetista e o usuário ou entre o projetista e sua equipe de trabalho;
- Ajuda a prevenir erros do futuro sistema;
- Uma forma de documentar o sistema *ideal*. Um sistema é *ideal* quando todas suas informações estão modeladas de acordo com certas condições;
- É a base para o projeto físico do banco de dados.

A abordagem utilizada aqui será a representação de dados no modelo relacional, utilizando-se, para tal, o **Modelo de Entidade-Relacionamento (MER)**. O MER é um modelo baseado na percepção do mundo real, que consiste em um conjunto de objetos básicos chamados de entidades e nos relacionamentos entre esses objetos. Foi proposto por Peter Chen, em 1976, como uma ferramenta de projeto de banco de dados. O MER apresenta como contribuições um maior grau de independência de dados que os modelos convencionais (de redes e hierárquico) e uma unificação de representação destes modelos, através do formalismo gráfico do Diagrama de Entidade-Relacionamento (DER).

São características do MER:

- Modela regras de negócio e não a implementação. A modelagem é dos dados requeridos para o negócio, baseado nas funcionalidades do sistema atual ou a ser desenvolvido. Para modelar um negócio, é necessário conhecer em detalhes sobre do que se trata.
- Possui uma sintaxe robusta, bem definida;
- Técnica amplamente difundida e utilizada. Atualmente, a maioria dos bancos de dados disponíveis no mercado utiliza a abordagem relacional como modelo de dados;
- Diagramas fáceis de entender e alterar.

Os objetivos de uma modelagem entidade-relacionamento são:

- Obter todas as informações requeridas sobre o negócio antes de sua implementação, tornando claras suas dependências;
- Dentro do possível, uma informação aparecer apenas uma vez no banco de dados. Uma modelagem que prevê o armazenamento de uma mesma informação em dois locais diferentes, deixa o sistema vulnerável quanto a possibilidade destas informações não serem as mesmas. No caso de uma inconsistência dos dados, qual delas deverá ser descartada?
- Facilitar o projeto do banco de dados, possibilitando a especificação de sua estrutura lógica.

2. Entidade e Instância

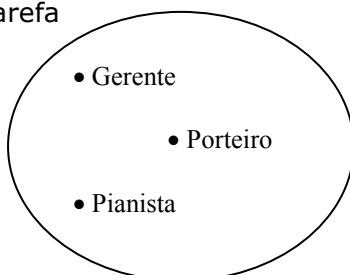
Num MER, uma **entidade** é um objeto, real ou abstrato, de relevância para o negócio. É uma categoria de idéias que são importantes ao negócio, as quais devem ser traduzidas em informação. Dois importantes aspectos de uma entidade é que possui **instâncias** e estas instâncias também são de interesse ao negócio. Pode-se considerar que uma instância identifica individualmente uma entidade. O quadro abaixo mostra alguns exemplos de entidades e instâncias:

Entidade	Instância
Pessoa	João, José, Antônio
Produto	Prego 12x12, File de Peixe Merluza
Tipo de produto	Plástico, papel, madeira
Tarefa	Professor, pianista, gerente
Versão do documento	1.2, 10.5

QUADRO 2.1 – EXEMPLOS DE ENTIDADES E INSTÂNCIAS

Observa-se que uma entidade possui várias instâncias e que cada instância está relacionada a uma entidade. Uma entidade representa um conjunto de instâncias que interessam ao negócio. Segue o exemplo abaixo:

Tarefa



Em termos físicos, uma entidade será uma *tabela* do BD e cada instância será uma *linha* (ou *registro* ou *tupla*). Exemplo:

Tabela Tarefa		Entidade
Código	Descrição	Instâncias
405	Gerente	
564	Porteiro	
321	Pianista	

3. Atributo e Domínio

Um **atributo** também representa algo significativo ao negócio. Um atributo é uma propriedade de uma entidade. É uma porção de informação que descreve, quantifica, qualifica, classifica e especifica uma entidade. Normalmente, uma entidade possui vários atributos. Interessa, em termos de modelagem conceitual, que estes atributos representem informações relevantes ao negócio. Atributos possuem valores (um número, um caracter, uma data, uma imagem, um som, etc), chamados de tipos de dados ou formato. Para um atributo particular, todas suas instâncias possuem os mesmos formatos. O quadro abaixo apresenta exemplos de entidades, instâncias e atributos.

Entidade	Instância	Atributo
Empregado	João, Antônio	Nome, idade, tamanho pé, dependentes, cidade
Carro	Escort, Gol	Cor, preço, modelo
Tarefa	Gerente	Código, Depto, Valor/hora, descrição

QUADRO 2.2 – EXEMPLOS DE ENTIDADES, INSTÂNCIAS E ATRIBUTOS

Algumas questões:

- O atributo Idade, da entidade Empregado, não parece ser uma boa escolha. O ideal seria Data Nascimento, ficando o cálculo da idade quando necessário. O armazenamento da informação idade é de difícil, senão impossível, atualização;
- O atributo Tamanho Pé, de Empregado, dependerá das regras de negócio. Imaginando que a finalidade de entidade Empregado seja a de armazenar dados sobre funcionários numa empresa que forneça uniforme de trabalho, o atributo é coerente. Se a empresa não possui esta política, ele é desnecessário;
- Uma importante decisão precisa ser tomada em relação a armazenar uma informação como um atributo ou uma entidade. O atributo Cidade, de Empregado, terá a cidade na qual o empregado reside. Se Cidade fosse uma entidade, alguns possíveis atributos seriam População, Área e Data Fundação. Aqui, novamente a escolha passa pelas regras de negócio. Normalmente, uma informação será um atributo se for de natureza atômica e será uma entidade quando possuir informações que possam (ou necessitem) ser relacionadas a outras entidades.

Em termos físicos, os atributos serão as *colunas* de uma tabela do BD. Exemplo:

Tabela Empregado

Nome	Idade
João	32
Antônio	25

Colunas

Atributo monovalorado ou atômico: assume um único valor, num certo instante de tempo, para cada instância. Exemplo: Nome, de Empregado

Atributo composto: formado por um ou mais sub-atributos. Exemplo: Cidade, de Empregado. Cidade pode ser composto pelo nome da cidade e o estado.

Atributo multivalorado: assume diversos valores. Seu nome, normalmente, é no plural. Exemplo: Dependentes, de Empregado.

Atributo determinante: identifica cada entidade como única. Exemplo: Código, em Tarefa.

Domínio de um atributo: conjunto de valores possíveis para um atributo. Exemplo: Idade deve estar entre 18 e 60, Estado deve ser "RS", "RJ", "SP", etc.

4. Relacionamentos

É uma estrutura que indica uma associação entre duas ou mais entidades. Alguns exemplos:

Entidade	Relacionamento	Entidade
Empregado Tarefa	Exerce É exercida por	Tarefa Empregado
Produto Tipo Produto	É classificado por um É uma classificação de	Tipo Produto Produto
Pessoa Reserva	Faz É feita por	Reserva Pessoa

QUADRO 2.3 – EXEMPLOS DE RELACIONAMENTOS

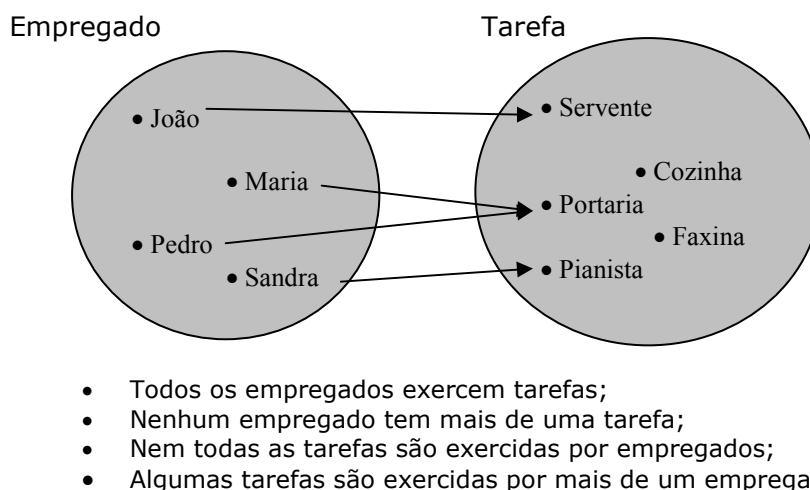


FIGURA 2.1 – EXEMPLO DE RELACIONAMENTO ENTRE EMPREGADO-TAREFA


O exemplo anterior serviu para que sejam apresentadas algumas questões referentes aos relacionamentos entre duas entidades. São elas:

1. Todo empregado **DEVE** (ou **PODE**) ter uma tarefa?
2. Toda tarefa **DEVE** (ou **PODE**) ser exercida por um empregado?
3. Um empregado pode exercer **UMA** ou **MAIS** tarefas? Ou então: Uma tarefa pode ser exercida por **UM** ou **MAIS** empregados?


Para as duas primeiras, a análise do relacionamento é de existência (obrigatório → DEVE ou opcional → PODE), enquanto que a terceira trata da cardinalidade do relacionamento (um-para-um, um-para-vários, vários-para-um ou vários-para-vários).

Um relacionamento entre duas entidades E1 e E2 deve ser *lido* da seguinte forma:

Cada E1 {deve / pode} nome_do_relacionamento {um / vários} E2



Existência



Cardinalidade

Existência (Obrigatório x Opcional)

Um relacionamento pode ser obrigatório ou não. Se ele existe, diz-se que é obrigatório. Se não existe, é opcional. A existência ou não de um relacionamento é identificada pelas palavras DEVE e PODE, por exemplo. Sem levar em conta a cardinalidade, as possíveis combinações da relação de existência entre as entidades Empregado e Tarefa são (observar que a coluna "Pode/Deve" é determinante da existência do relacionamento):

E1	Pode Deve	Nome do relacionamento	E2	Obrigatório Opcional
Cada Empregado	Deve	Exercer	Uma/várias Tarefas	Obrigatório
Cada Tarefa	Deve	Ser exercida por	Um/vários Empregados	Obrigatório
Cada Empregado	Pod	Exercer	Uma/várias Tarefas	Opcional
Cada Tarefa	Pod	Ser exercida por	Um/vários Empregados	Opcional

QUADRO 2.4 – EXISTÊNCIA DE RELACIONAMENTOS

Cardinalidade

É o número de entidades que podem estar associadas. Os relacionamentos binários podem ser: um-para-um (1:1), um-para-vários (1:N) ou vários-para-vários (N:N). Sem levar em conta a existência, as possíveis combinações para a cardinalidade do relacionamento entre as entidades Empregado e Tarefa são (observar que a coluna de E2 é determinante para a cardinalidade):

E1	Pode Deve	Nome do relacionamento	E2	Cardinalidade
Cada Empregado	Deve/pode	Exercer	Uma Tarefa	Um-para-um 1:1
Cada Tarefa	Deve/pode	Ser exercida por	Um Empregado	
Cada Empregado	Deve/pode	Exercer	Uma Tarefa	Um-para-vários 1:N
Cada Tarefa	Deve/pode	Ser exercida por	Vários Empregados	
Cada Empregado	Deve/pode	Exercer	Várias Tarefas	Vários-para-vários N:N
Cada Tarefa	Deve/pode	Ser exercida por	Vários Empregados	

QUADRO 2.5 – CARDINALIDADE DE RELACIONAMENTOS

A cardinalidade **um-para-um (1:1)** ocorre quando uma instância de E1 está associada no máximo a uma instância de E2 e uma instância de E2 está associada no máximo a uma instância de E1.

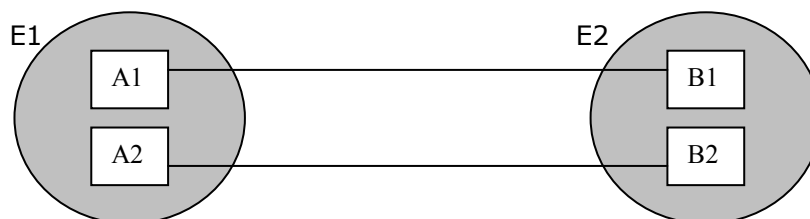


FIGURA 2.2 –CARDINALIDADE UM-PARA-UM

A cardinalidade **um-para-vários (1:N)** ocorre quando uma instância de E1 está associada a qualquer número de instâncias de E2, enquanto que uma instância de E2 está associada no máximo a uma instância de E1.

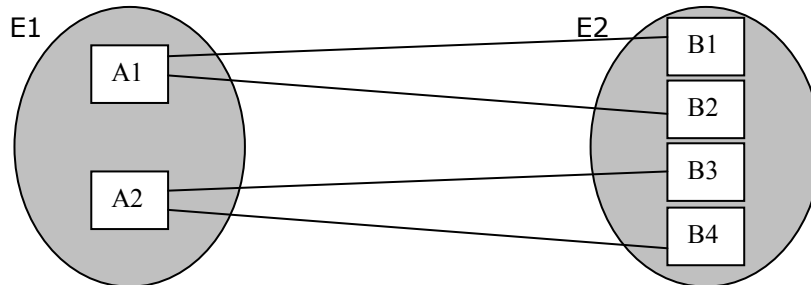


FIGURA 2.3 –CARDINALIDADE UM-PARA-VÁRIOS

A cardinalidade **vários-para-vários (N:N)** ocorre quando uma instância de E1 está associada a qualquer número de instâncias de E2 e uma instância de E2 está associada a qualquer número de instâncias de E1.

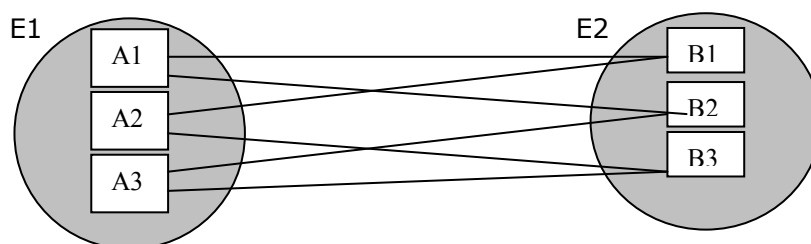


FIGURA 2.4 –CARDINALIDADE VÁRIOS-PARA-VÁRIOS

5. O Diagrama Entidade-Relacionamento (DER)

A estrutura lógica geral de um banco de dados pode ser expressa graficamente por um Diagrama de Entidade-Relacionamento. É uma representação gráfica do modelo ou parte do modelo. De acordo com o grau de complexidade do negócio ou o nível de detalhamento impresso pelo projetista do banco, um modelo pode ser representado por vários diagramas.

Um DER utiliza-se de um número de elementos gráficos. Infelizmente, não existe uma padronização na representação do diagrama. Serão apresentados dois padrões: um deles segundo Peter Chen e o outro uma pequena variação de um desenvolvido pela Oracle.

Os componentes do Diagrama de Entidade-Relacionamento, de acordo com Peter Chen são:

- **Retângulos:** representam as entidades
- **Elipses:** representam os atributos
- **Losangos:** representam os relacionamentos
- **Linhas:** ligam atributos a entidades e entidades a relacionamentos.

Exemplo de um DER envolvendo Empregados e Tarefas

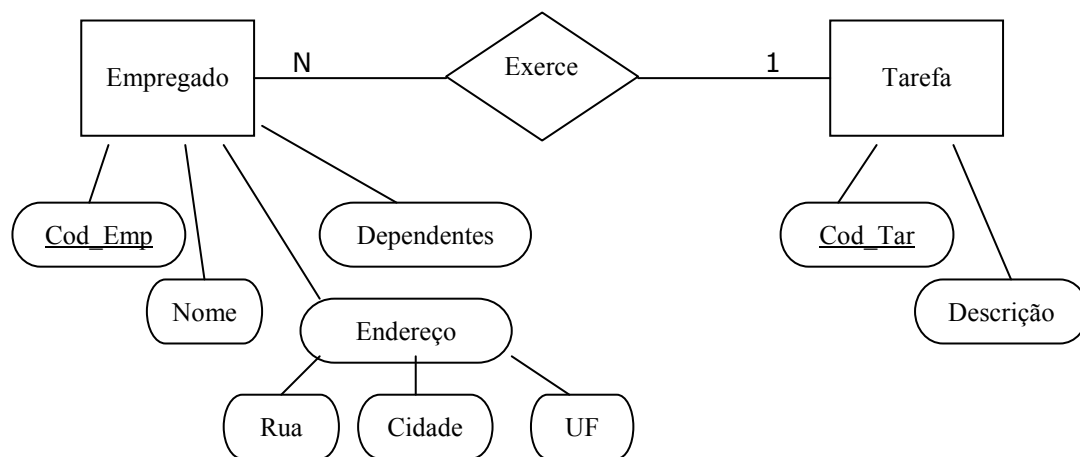


FIGURA 2.5 – EXEMPLO DE DER SEGUNDO PETER CHEN

Observa-se o seguinte:

- São duas as entidades: Empregado e Tarefa
- Atributos da entidade Empregado:
 - Cod_Emp (determinante - está sublinhado)
 - Nome (monovalorado)
 - Dependentes (multivalorado)
 - Endereço (composto)
 - Rua (monovalorado)
 - Cidade (monovalorado)
 - UF (monovalorado)
- Atributos da entidade Tarefa:
 - Cod_Tar (determinante - está sublinhado)
 - Descrição (monovalorado)
- O relacionamento entre Empregado e Tarefa possui cardinalidade 1:n

Uma variação do diagrama anterior é apresentada a seguir. Nota-se que os atributos determinantes não mais são sublinhados, pois possuem símbolos diferentes dos demais.

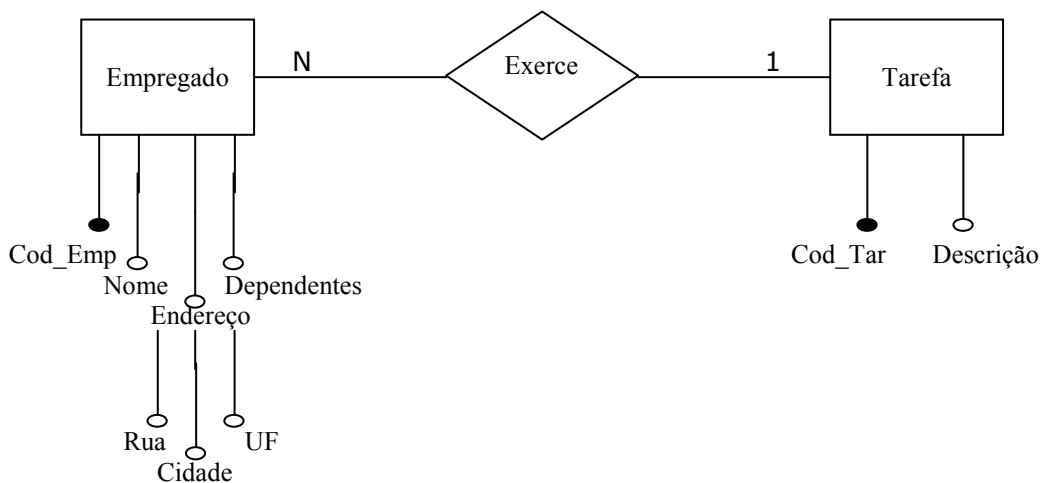


FIGURA 2.6 – UMA VARIAÇÃO DE DER

A representação adotada neste texto, adaptada de uma desenvolvida pela Oracle, será apresentada por intermédio de um exemplo.

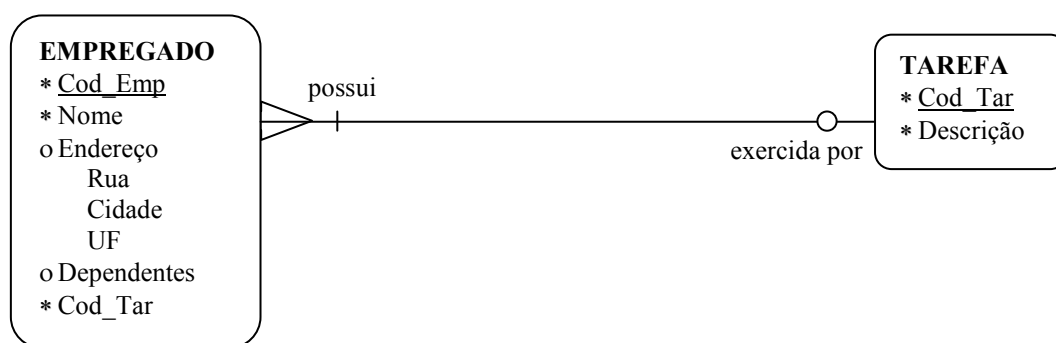


FIGURA 2.7 – DER ADAPTADO DO PADRÃO ORACLE

Sobre as entidades

São duas, Empregado e Tarefa, identificadas por letras maiúsculas. Abrigam os atributos correspondentes.

Sobre os atributos

Aqueles determinantes (atributos chave) são sublinhados. Possuem a idéia de *obrigatórios* ou *opcionais*, identificados, respectivamente, pelos símbolos “*” (asterisco) e “o” (círculo). Um atributo é obrigatório quando todas suas instâncias de sua entidade possuem a informação conhecida e disponível, enquanto que é opcional quando pode ser desconhecida ou não disponível ou, ainda, quando é conhecida, mas não muito importante para o negócio.

Sobre os relacionamentos

Não possui um símbolo específico para sua representação. A própria linha de ligação entre as entidades é o relacionamento.

O nome do relacionamento é colocado próximo ao ponto de partida da entidade.

Sentido	Lê-se
Empregado → Tarefa	Empregado <i>possui</i> Tarefa
Tarefa → Empregado	Tarefa <i>é exercida por</i> Empregado

QUADRO 2.6 – NOMES DE RELACIONAMENTOS

A cardinalidade do relacionamento é identificada pelas extremidades da linha. Do lado do Empregado, o símbolo semelhante a um *pé de galinha* na ponta representa VÁRIOS, e, do lado da Tarefa, uma ponta simples representa UM. Do DER apresentado, o relacionamento é de 1:n.

Cada Empregado pode (ou deve) possuir **UMA** Tarefa
Cada Tarefa pode (ou deve) ser exercida por **VÁRIOS** Empregados

Em relação à existência do relacionamento, ele pode ser obrigatório ou opcional. Os símbolos utilizados para esta identificação são, respectivamente, um traço cortando

a linha do relacionamento (lado do Empregado) e um círculo sobre a linha (lado do Depto).

Cada Empregado **DEVE** possuir uma Tarefa

Cada Tarefa **PODE** ser exercida por vários Empregados

Os quadros seguintes mostram as possíveis combinações para relacionamentos envolvendo a questão da existência, com seus respectivos símbolos.

Relacionamento 1:n	Lê-se
	Cada E1 deve existir em vários E2 Cada E2 deve existir em um E1
	Cada E1 pode existir em vários E2 Cada E2 deve existir em um E1
	Cada E1 deve existir em vários E2 Cada E2 pode existir em um E1
	Cada E1 pode existir em vários E2 Cada E2 pode existir em um E1

Relacionamento n:n	Lê-se
	Cada E1 deve existir em vários E2 Cada E2 deve existir em vários E1
	Cada E1 deve existir em vários E2 Cada E2 pode existir em vários E1
	Cada E1 pode existir em vários E2 Cada E2 deve existir em vários E1

Relacionamento 1:1	Lê-se
	Cada E1 deve existir em um E2 Cada E2 deve existir em um E1
	Cada E1 deve existir em um E2 Cada E2 pode existir em um E1
	Cada E1 pode existir em um E2 Cada E2 deve existir em um E1

QUADRO 2.7 – POSSÍVEIS RELACIONAMENTOS

6. Projeto de chaves e Regras de Integridade

A questão fundamental do projeto de chaves numa modelagem é eliminar ao máximo os efeitos da redundância. Uma chave é um conjunto de um ou mais atributos que, tomados coletivamente, permite identificar unicamente uma entidade.

Numa entidade que contenha mais de uma chave, cada uma é chamada de "chave candidata". Outros conceitos de chaves são relevantes neste contexto: chave primária (*primary key - pk*), chave alternativa (*alternate key - ak*) e chave estrangeira (*foreign key - fk*).

Chave primária: é o conjunto de atributos que identificam unicamente uma entidade. É o mesmo conceito de atributo determinante, já explicado. Exemplo: a figura 2.7 apresenta os atributos Cod_Emp e Cod_Tar como chaves primárias de suas entidades. Pode-se imaginar, como outro exemplo, uma entidade ALUNO contendo um atributo Nro de Matrícula. Este atributo é uma chave primária (*primary key - pk*), pelo fato de não existir dois alunos dentre os cadastrados com o mesmo Nro de Matrícula. Num BD, é permitida apenas uma chave primária a cada entidade.

Chave alternativa: também conhecida como chave secundária, é aquela chave candidata que não é primária. De acordo com as regras do negócio, às vezes pode ser conveniente identificar entidades por atributos que não são únicos dentre todas as instâncias. No exemplo da entidade ALUNO, um atributo Nome é sério candidato a ser uma chave alternativa (*alternate key - ak*). Desta forma, apesar de Nome não garantir que todas as instâncias sejam únicas por este atributo (uma mesma pessoa pode estar matriculada em dois cursos. Assim, a chave primária Nro de Matrícula é única dentre todos os alunos, mas Nome terá, pelo menos, em duas instâncias), as instâncias de ALUNO podem ser recuperadas pela chave alternativa. Pode-se ter tantas chaves alternativas quantas forem necessárias, numa mesma entidade.

Chave estrangeira: são atributos de uma entidade cujos valores aparecem como chave primária em outra entidade. A presença de uma chave estrangeira (*foreign key - fk*) numa entidade ocorre por força das regras de integridade referencial.

As regras de integridade num BD relacional são:

1ª regra - Integridade de entidade: nenhum valor de uma chave primária pode ser nulo. Em termos de DER, o atributo *pk* deve ser sempre obrigatório, nunca opcional.

2ª regra - Integridade referencial: numa entidade que possui uma chave estrangeira, cada valor desta chave só pode ser nulo ou igual a algum valor da *pk* correspondente no relacionamento. As alterações dos valores constituintes da *pk* ou a remoção de uma instância que contenha uma *pk* com uma *fk* associada em outra entidade pode causar problemas de integridade referencial.

A figura 2.8 abaixo apresenta um DER e algumas instâncias das entidades envolvidas. Este exemplo servirá para maiores explicações sobre o projeto de chaves e as regras de integridade da modelagem.

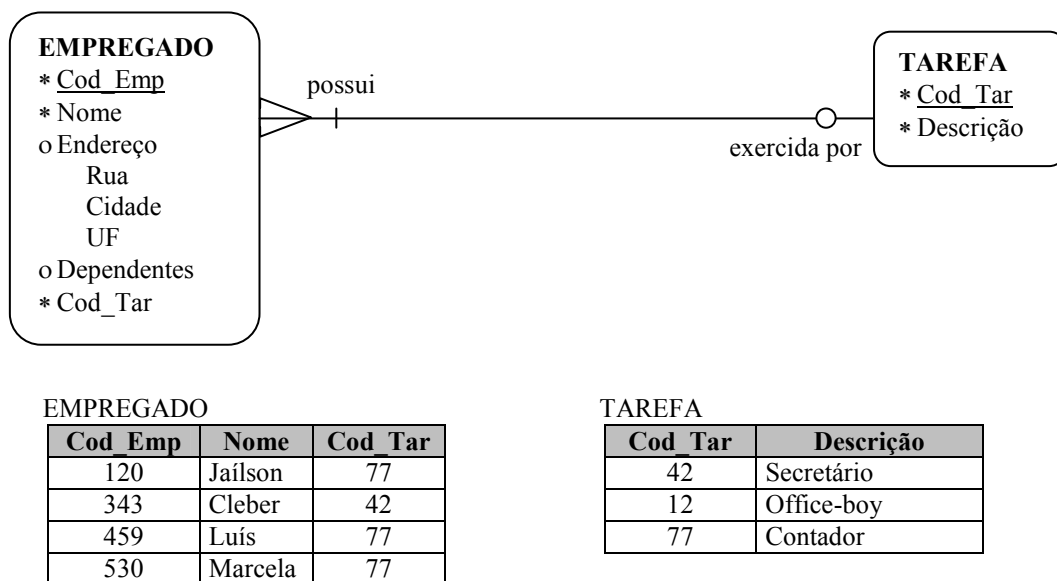


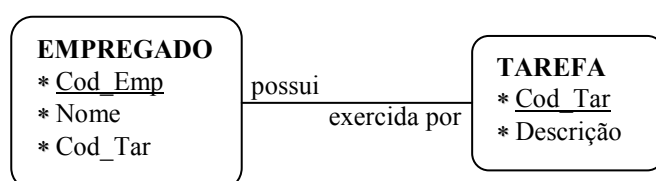
FIGURA 2.8 – DER ENTRE EMPREGADO-TAREFA

Questões sobre o exemplo:

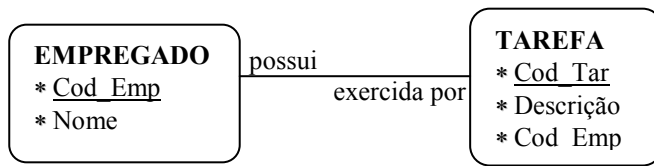
- Cod_Emp, de EMPREGADO e Cod_Tar, de TAREFA, são chaves primárias (*pk*)
- Cod_Tar de EMPREGADO é uma chave estrangeira (*fk*)
- As *pks* são atributos obrigatórios. Pela primeira regra de integridade, portanto, todas as instâncias de EMPREGADO e de TAREFA devem estar com esta coluna preenchida.
- Neste caso, a *fk* é um atributo obrigatório porque a regra de existência do relacionamento diz que "Cada empregado **deve** possuir uma tarefa". Poderia ser opcional, caso o relacionamento fosse "Cada empregado **pode** possuir uma tarefa".
- A remoção da instância "Secretário", de TAREFA, ou a troca do conteúdo de Cod_Tar, de TAREFA, de 42 para 39, por exemplo, causa problemas de integridade referencial (segunda regra).
- A remoção da instância "Office-Boy", de TAREFA, não fere a nenhuma regra de integridade, porque não existe nenhuma instância em EMPREGADO com *fk*=12.
- Em relação a existência do relacionamento do lado de TAREFA, o fato de ser opcional ("Cada Tarefa PODE ser exercida por vários Empregados"), faz que se tenha instâncias ("Office-Boy") sem necessariamente estar associado a algum empregado.

Uma análise entre as chaves de entidades e cardinalidade dos relacionamentos (1:1, 1:n ou n:n), leva-nos às seguintes observações:

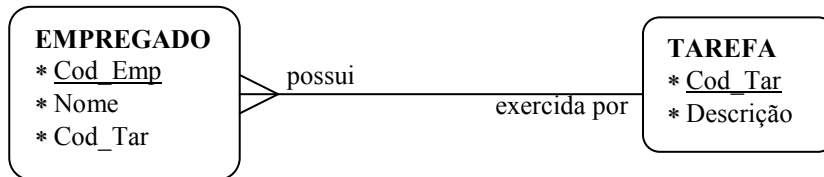
- Num relacionamento 1:1, a *fk* deve estar em umas das entidades relacionadas. Exemplo:



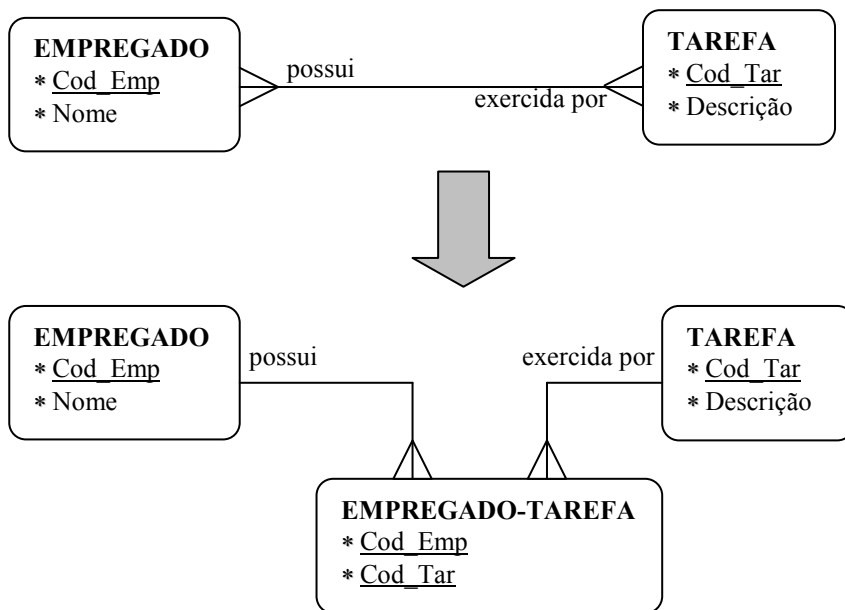
ou



- Num relacionamento 1:n, a *fk* deve estar na entidade da direção VÁRIOS. Exemplo:



- Num relacionamento n:n, deve ser criada uma nova entidade contendo como *pk* as *pks* das entidades relacionadas. Exemplo:



7. Normalização e Regras para a confecção de DER

Conforme já comentado, o esquema de um banco de dados é a estrutura geral do BD e, este esquema, pode ser representado de uma forma textual ou gráfica. Normalmente, utiliza-se o DER para a representação gráfica, ficando a textual como uma opção mais simples e rápida no processo de normalização dos dados.

Para uma modelagem conceitual de dados, é necessário um completo domínio das regras de negócio, de modo que o modelo resultante reflita numa estrutura que melhor represente os objetos envolvidos.

Têm-se duas alternativas para a criação de modelos conceituais: uma delas é através da normalização de dados e, a outra, por umas regras básicas de identificação dos elementos do DER na idéia geral do negócio.

Normalização de Dados

Consiste em definir o formato lógico adequado às entidades identificadas no negócio, com o objetivo de minimizar o espaço utilizado pelos dados e garantir as regras de integridade e a confiabilidade das informações.

A normalização é feita através da análise dos dados que compõem as entidades, utilizando um conceito chamado **Formas Normais (FN)**. As FN são conjuntos de restrições nos quais os dados devem satisfazê-las. Pode-se dizer que a estrutura está na primeira forma normal (1FN), se os dados que a compõem satisfizerem as restrições definidas para esta etapa.

A normalização completa dos dados é feita seguindo as restrições das três formas normais existentes, sendo que a passagem de uma FN para outra é feita tendo como base o resultado obtido na etapa anterior, ou seja, na FN anterior.

Para realizar a normalização dos dados, são imprescindíveis as definições dos atributos chave a cada entidade e será adotada a representação textual.

Como exemplo, criar um modelo conceitual para armazenar os dados de um boletim escolar, conforme a figura 2.9 abaixo:

ESCOLA ESTADUAL BARÃO DE MACAÚBAS											www.eebm.com.br				
Aluno: FRANCISCO JOSÉ GUSMÕES LIMA						Matrícula: 6969					Ano letivo: 2004				
Endereço: RUA DAS ACÁCIAS, 465, CARREIROS, SÃO FIDÉLIS - RJ															
Filiação: GERALDO TENÓRIO LIMA						Curso: 45 – INFORMÁTICA									
FELICIA MARIA GUSMÕES LIMA						Turno: DIURNO									
Disciplina	Turma	NOTAS					FALTAS				MF	TA	TF	Cond.	
		I	II	III	IV	Ex	I	II	III	IV					
13 - LÓGICA	A	7,0	8,0	6,0	7,0		0	4	6	0	7,0	23	10	Aprov	
08 - BDADOS	C	5,0	4,0	9,0	8,0	8,0	3	3	0	0	6,9	35	9	Aprov	

As notas I, II, III e IV referem-se aos quatro bimestres. A nota Ex é a nota do exame

As faltas I, II, III e IV referem-se aos quatro bimestres.

MF é a Média Final, TA é o Total de Aulas dadas, TF é o Total Faltas e Cond é condição de aprovação

FIGURA 2.9 – EXEMPLO DE UM BOLETIM ESCOLAR EMPREGADO-TAREFA

Primeira Forma Normal (1FN)

Consiste em retirar da entidade os elementos repetitivos, ou seja, aqueles dados que podem compor uma estrutura de vetor. Pode-se afirmar que uma entidade está normalizada na 1FN, se não possuir atributos repetitivos.

Do exemplo, pode-se iniciar pensando numa única entidade chamada BOLETIM, com o seguinte esquema:

BOLETIM (Nome Escola, Http Escola, Nome Aluno, Nro Matrícula, Ano Letivo, Endereço, Pai, Mãe, Cod Curso, Nome Curso, Turno, Relação de Disciplinas (Cod Disc, Nome Disc, Turma, Relação de Notas (I, II, III, IV, Ex), Relação de Faltas (I, II, III, IV), MF, TA, TF, Cond)).

É necessária a definição de uma *pk* para BOLETIM. Como o assunto trata de boletins escolares e cada aluno possui um, o atributo Nro Matrícula apresenta-se como um forte candidato a chave primária.

Da entidade, observa-se que existem várias disciplinas para cada aluno, sendo, portanto, elementos repetitivos que deverão ser retirados. Então, a primeira forma normal será:

BOLETIM (Nome Escola, Http Escola, Nome Aluno, Nro Matrícula, Ano Letivo, Endereço, Pai, Mãe, Cod Curso, Nome Curso, Turno)

MATRICULA (Nro Matrícula, Cod Disc, Nome Disc, Turma, Relação de Notas (I, II, III, IV, Ex), Relação de Faltas (I, II, III, IV), MF, TA, TF, Cond))

Como resultado desta etapa, ocorre um desdobramento dos atributos da entidade, a saber:

- Em BOLETIM, foram extraídos os atributos repetidos;
- Em MATRICULA, os atributos foram os extraídos de BOLETIM, tendo como chave primária Nro Matrícula e Cod Disc. Esta *pk*, em MATRICULA, é única.

Em MATRICULA, nota-se ainda uma repetição de notas e de faltas, que poderão dar origem a novas entidades. Por uma decisão de projeto, as entidades ficam:

BOLETIM (Nome Escola, Http Escola, Nome Aluno, Nro Matrícula, Ano Letivo, Endereço, Pai, Mãe, Cod Curso, Nome Curso, Turno)

MATRICULA (Nro Matrícula, Cod Disc, Nome Disc, Turma, MF, TA, TF, Cond)

NOTA e FALTA (Nro Matrícula, Cod Disc, Id Bimestre, Nota, Falta)

Aqui, mais uma vez, a nova entidade NOTA e FALTA tem sua *pk* composta da *pk* de MATRICULA acrescentada de Id Bimestre, uma identificação do bimestre.

Segunda Forma Normal (2FN)

Consiste em retirar, das entidades que possuem chaves compostas (atributo chave formado por mais de um atributo), os atributos que são funcionalmente dependentes de parte da chave. Pode-se afirmar que, uma entidade está na 2FN, se estiver na 1FN e não possuir atributos funcionalmente dependentes de parte da chave.

(1FN)	BOLETIM (<u>Nome Escola</u> , <u>Http Escola</u> , <u>Nome Aluno</u> , <u>Nro Matrícula</u> , <u>Ano Letivo</u> , <u>Endereço</u> , <u>Pai</u> , <u>Mãe</u> , <u>Cod Curso</u> , <u>Nome Curso</u> , <u>Turno</u>)
	MATRICULA (<u>Nro Matrícula</u> , <u>Cod Disc</u> , <u>Nome Disc</u> , <u>Turma</u> , <u>MF</u> , <u>TA</u> , <u>TF</u> , <u>Cond</u>)
	NOTA e FALTA (<u>Nro Matrícula</u> , <u>Cod Disc</u> , <u>Id Bimestre</u> , <u>Nota</u> , <u>Falta</u>)

A última configuração das entidades as apresenta na 1FN. Pode-se notar MATRÍCULA como a única entidade que possui uma *pk* composta com atributos dependentes (Nome Disc depende de Cod Disc). A segunda forma normal, então, define:

BOLETIM (<u>Nome Escola</u> , <u>Http Escola</u> , <u>Nome Aluno</u> , <u>Nro Matrícula</u> , <u>Ano Letivo</u> , <u>Endereço</u> , <u>Pai</u> , <u>Mãe</u> , <u>Cod Curso</u> , <u>Nome Curso</u> , <u>Turno</u>)
MATRICULA (<u>Nro Matrícula</u> , <u>Cod Disc</u> , <u>Turma</u> , <u>MF</u> , <u>TA</u> , <u>TF</u> , <u>Cond</u>)
DISCIPLINA (<u>Cod Disc</u> , <u>Nome Disc</u>)
NOTA e FALTA (<u>Nro Matrícula</u> , <u>Cod Disc</u> , <u>Id Bimestre</u> , <u>Nota</u> , <u>Falta</u>)

Da ação da 2FN, tem-se:

- Surgiu a nova entidade DISCIPLINA, tendo como *pk* parte da *pk* do atributo originário constante na dependência.
- O atributo Nome Disc foi excluído de MATRICULA por ser dependente de Cod Disc.

Terceira Forma Normal (3FN)

Consiste em retirar os atributos que são funcionalmente dependentes de outros atributos que não são chaves. Pode-se afirmar que, uma entidade está na 3FN, se estiver na 2FN e não possuir atributos dependentes de outros atributos não chaves.

(2FN)

BOLETIM (Nome Escola, Http Escola, Nome Aluno, Nro Matrícula, Ano Letivo, Endereço, Pai, Mãe, Cod Curso, Nome Curso, Turno)

MATRICULA (Nro Matrícula, Cod Disc, Turma, MF, TA, TF, Cond)

DISCIPLINA (Cod Disc, Nome Disc)

NOTA e FALTA (Nro Matrícula, Cod Disc, Id Bimestre, Nota, Falta)

(3FN)

BOLETIM (Nro Matrícula, Nome Aluno, Endereço, Pai, Mãe, Cod Curso, Turno)

CURSO (Cod Curso, Nome Curso)

MATRICULA (Nro Matrícula, Cod Disc, Turma, MF, TA, TF, Cond)

DISCIPLINA (Cod Disc, Nome Disc)

NOTA e FALTA (Nro Matrícula, Cod Disc, Id Bimestre, Nota, Falta)

A ação da 3FN provocou poucas alterações; surgiu uma nova entidade CURSO e o atributo Nome Curso, de BOLETIM, foi eliminado. Os demais ajustes foram em decorrência de uma melhoria da modelagem. São eles:

- Atributo Ano Letivo foi eliminado do modelo por não se relacionar com nenhuma entidade existente. Em termos práticos, pode-se imaginar que todos os dados referentes ao ano letivo fiquem armazenados em BD diferentes, mas com a mesma estrutura (um BD para o ano 2001, outro para 2002, ...).
- Os dados da escola não são relevantes nesta modelagem porque são constantes, ou seja, são iguais a todos os boletins.