

Banco de Dados Relacional



Rio Grande do Norte

CONSELHO REGIONAL

Marcantoni Gadelha
Presidente

DIREÇÃO REGIONAL

Ronaldo Rezende
Diretor Regional

DIVISÃO DE FORMAÇÃO PROFISSIONAL

Zilca Maria de Macedo Pascoal
Diretora de Formação Profissional

Sumário

1. A IMPORTÂNCIA DO ESTUDO DOS DADOS NO DESENVOLVIMENTO DE UM SISTEMA	5
2. O QUE É UM BANCO DE DADOS	6
3. OBJETIVOS E REQUISITOS DE UM BANCO DE DADOS AUTOMATIZADO	7
3.1 EVITAR REDUNDÂNCIAS	7
3.2 FACILITAR ATUALIZAÇÕES	7
3.3 EVITAR INCONSISTÊNCIA DE DADOS	7
4. NÍVEIS DE ABSTRAÇÃO	8
4.1 MUNDO "REAL"	8
4.2 MODELO DESCRITIVO	8
4.3 MODELO CONCEITUAL	8
4.4 MODELO OPERACIONAL	9
4.5 MODELO INTERNO	9
5. PROJETO DESCENDENTE (TOP-DOWN)	10
6. DO MUNDO "REAL" AO MODELO DESCRITIVO	11
7. DO MODELO DESCRITIVO AO CONCEITUAL	12
7.1 DEFININDO OS CONJUNTOS DE INFORMAÇÃO	12
7.2 O MODELO DE ENTIDADE-RELACIONAMENTO (MER)	12
7.3 VALIDAÇÃO DO MER	30

1. A IMPORTÂNCIA DO ESTUDO DOS DADOS NO DESENVOLVIMENTO DE UM SISTEMA

Estudar e projetar Bancos de Dados (BD's) de um sistema é hoje tão importante, ou talvez mais, quanto estudar a funcionalidade dele. Para entendermos melhor esta afirmação, vejamos o seguinte exemplo:

Um banco descobre que seu programa de controle de aplicações não é bom. Por causa de rotinas mal escritas ele demora o dobro do tempo necessário para fazer a maioria de suas operações. A equipe de desenvolvimento do Banco resolve então construir um novo programa. Enquanto o novo programa não está pronto o velho vai sendo usado. Quando o novo programa ficar pronto e testado, ele é implantado e pronto.

Agora imagine que, no mesmo banco, descobriram que havia redundância de informação. Toda vez que se registrava uma aplicação de um cliente o nome dele era gravado junto com as informações da aplicação. Bem, o nome do cliente é um só. E, assim sendo, deve estar registrado em um só local, e não em vários. Para ter noção do problema imagine que descubram que o nome de um cliente está errado. Vamos ter que consertar em todos os locais nos quais ele aparece. Então a equipe de desenvolvimento do banco resolve remodelar o banco de dados. Até aí tudo bem. Mas como implantar este novo modelo se todas as informações estão armazenadas no modelo velho? E vale lembrar que, enquanto a equipe pensa o que fazer, o problema está aumentando. Pois novas informações não param de ser registradas. É resolvido então que o banco tem que parar de trabalhar por uns dias até que todas as informações antigas sejam convertidas para o novo formato. Vocês conseguem imaginar isto.

2. O QUE É UM BANCO DE DADOS

É um agrupamento de informações interligadas que podem conter diversos conjuntos de dados ou somente um. Por exemplo: sua agenda pessoal é um banco de dados. Nela você tem um grupo de informações interligadas: Nome, Endereço, Telefone e Data de Aniversário, que tem uma relação forte pois identificam uma pessoa conhecida. Nesse caso, o conjunto de dados é um só. Já no caso de uma ficha de empréstimos de livros de uma biblioteca, que também é um banco de dados, geralmente temos as seguintes informações: código, título, usuários que emprestaram, as datas de empréstimo e as de devolução. Abaixo um exemplo da ficha.

BIBLIOTECA DO CIN		
FICHA DE LIVRO		
Código: 67.986 CRÔNICA DE UMA MORTE ANUNCIADA		
<u>Usuário</u>	Empréstimo	Devolução
JULIANA	08/09/97	18/09/97
ROBSON	20/09/97	25/09/97
CLÁUDIO	30/09/97	

Percebam que código e título pertencem ao conjunto de informações de livros (estão fortemente ligadas) e usuários, datas de empréstimo e devolução os quais pertencem a outro conjunto: empréstimos ou movimentos de livros. Existe uma ligação entre elas, porém elas pertencem a conjuntos distintos.

Um dos problemas cruciais em modelagem de banco de dados é justamente identificar quais são os conjuntos de informação existentes em um sistema.

3. OBJETIVOS E REQUISITOS DE UM BANCO DE DADOS AUTOMATIZADO

3.1 EVITAR REDUNDÂNCIAS

Nos sistemas manuais a redundância de dados é normal e necessária. Em algumas bibliotecas, por exemplo, cada vez que você aluga um livro o título dele é registrado em sua ficha. Isto facilita uma posterior pesquisa. Em um sistema automatizado, isso seria um desastre, pois além de ocupar mais espaço em disco do que o necessário, uma posterior mudança no título do livro levaria o sistema a percorrer milhares de registros. Nos sistemas automatizados, nós sempre perseguiremos o mínimo de redundância possível.

3.2 FACILITAR ATUALIZAÇÕES

Como vocês virão no tópico anterior, um banco de dados mal projetado pode dificultar a atualização das informações. Uma das principais vantagens dos computadores é justamente permitir que se atualize (edite) dados com rapidez e eficiência. Se você cria um BD que dificulta a atualização, então este BD é um problema. Concorda!?

3.3 EVITAR INCONSISTÊNCIA DE DADOS

Imagine que você crie uma estrutura de BD que permita que seja alocado um instrutor para uma turma cuja disciplina não faz parte do elenco de disciplinas que o instrutor pode lecionar. Isso é uma inconsistência de dados. E você tem como evitar isto em seu projeto.

4. NÍVEIS DE ABSTRAÇÃO

No desenvolvimento de um sistema de banco de dados temos que levar em conta que o nosso trabalho vai basear-se em um modelo real. Ou seja, que existe no mundo real. E na maioria dos casos não está nem parcialmente automatizado. Então, temos que ter condições de visualizar, compreender e interpretar o modelo desde o real (o que existe) até o modelo operacional que será implementado.

4.1 MUNDO "REAL"

No nível do mundo "Real" nós não temos um modelo formal de informações. Elas estão dispostas sem limitações. O projetista de banco de dados tem que definir o que interessa do mundo real para o seu projeto. Os objetos do mundo real são os seres, os fatos, as coisas e os organismos sociais.

4.2 MODELO DESCRITIVO

Neste modelo você já filtrou o que interessa no mundo real e já estabelece alguns limites para a organização da informação utilizando linguagens não formais. Perceba que, neste nível, você começa a ter algo palpável, pois, neste modelo, já serão colocadas impressões a respeito de como os dados irão organizar-se. No nível de mundo real, você irá observar, entrevistar, pesquisar para poder ter subsídios para o modelo descritivo. Resumindo: o mundo real não é modelado, ele existe e pronto. Já o modelo descritivo é produto do seu trabalho. Por ser um nível onde não usamos ainda linguagens formais, a escolha da linguagem a ser utilizada é bem subjetiva. Alguns podem querer usar o próprio português, outros, podem querer usar alguma linguagem gráfica. Este é um nível de idéias e pensamentos de como as informações irão organizar-se.

4.3 MODELO CONCEITUAL

Nosso primeiro nível formal: Aqui definiremos estruturas de informação que servirão de base para o nosso modelo operacional. Nesse nível, são identificados os conjuntos de informação e as ligações existentes entre eles. Utilizaremos, nesta fase, o Modelo de Entidade e Relacionamento (MER) e sua linguagem gráfica (Diagrama de Entidade e Relacionamento).

4.4 MODELO OPERACIONAL

É o banco de dados automatizado através da utilização de um Sistema Gerenciador de Banco de Dados (SGBD), tais como: ACCESS, Dbase, Paradox, Oracle, SQL Server, Lotus Approach. Cada um destes SGBD's tem sua maneira de implementar o seu modelo operacional, porém com pequenas diferenças entre eles.

4.5 MODELO INTERNO

É a organização dos bits de um banco de dados em um disco, ou seja, um Sistema Operacional que organiza os dados em um disco. Esse nível, geralmente, não é objeto de preocupação do programador. É interessante que o programador saiba como este processo ocorre porém, manipular este modelo é trabalho para cientistas da computação.

5. PROJETO DESCENDENTE (TOP-DOWN)

No desenvolvimento do nosso projeto de banco de dados, utilizaremos uma abordagem descendente. Começaremos no mundo real e terminaremos no modelo operacional. No caso de projetos de banco de dados, não temos muito o que decidir a respeito de abordagens, pois é quase uma imposição o projeto descendente.

Na passagem do mundo real para o modelo descritivo e deste para o conceitual, não temos ferramentas que nos ajudem na transição. Porém, na transição do modelo conceitual para o operacional, utilizaremos um processo chamado decomposição. A inexistência de ferramentas de transição, nos primeiros níveis, é consequência da informalidade dos dois primeiros níveis. Sem modelos formais é difícil estabelecer formalidades de passagem.

6. DO MUNDO “REAL” AO MODELO DESCRITIVO

Como já comentamos, a construção do modelo descritivo a partir do mundo real dá-se através de uma filtragem. Por meio de observações, entrevistas e pesquisas o programador irá definir o que lhe interessa no mundo real. Para exemplificar, vamos ver um modelo descritivo do sistema da coordenação pedagógica do SENAC.

Através de entrevistas e pesquisas nos documentos e relatórios utilizados, você chegou as seguintes conclusões:

É registrado, para cada instrutor, um número de matrícula, o nome dele, sua data de admissão e as disciplinas que ele pode ministrar;

Cada disciplina pode ter vários instrutores habilitados a lecioná-la e é registrado para cada uma delas um código, o nome dela e sua carga horária;

As turmas tem um código e é registrada data de início e término, horário, instrutor, sala e disciplina da turma;

Cada turma tem um instrutor, porém este pode ser substituído por outro;

Cada turma está associada a uma sala, mas ela pode mudar de sala;

Cada turma tem uma só disciplina;

As salas podem ser teóricas ou práticas e cada uma tem uma identificação.

Estas afirmações são o seu modelo descritivo.

7. DO MODELO DESCRITIVO AO CONCEITUAL

7.1 DEFININDO OS CONJUNTOS DE INFORMAÇÃO

O primeiro passo na construção de um modelo conceitual é o reconhecimento dos conjuntos de informação. Uma boa dica para encontrar os conjuntos está no próprio modelo descritivo. No exemplo acima, perceba que estão descritas ligações entre seres, coisas, fatos e organismos sociais. Identificando estes elementos você estará isolando os conjuntos de informação que neste caso são: instrutores, disciplinas, turmas e salas.

Outra maneira de identificar os conjuntos é listar as informações do sistema e descobrir quem elas identificam.

Informação

Matrícula
Carga Horária
Teórica ou Prática
Data de Término

É sobre

Instrutores
Disciplinas
Salas
Turmas

7.2 O MODELO DE ENTIDADE-RELACIONAMENTO (MER)

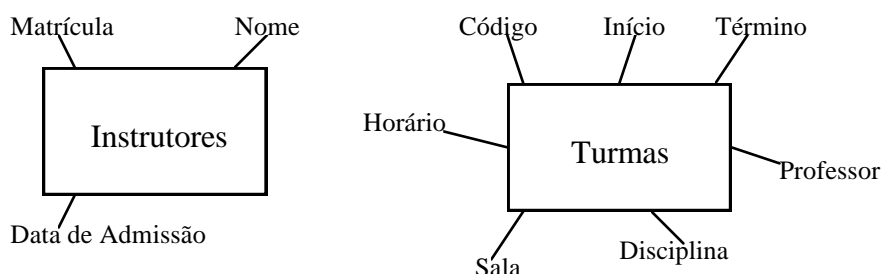
O MER é o modelo escolhido para representar o modelo conceitual do nosso banco de dados. Ele utiliza-se de uma linguagem gráfica e é o mais aceito pelos projetistas de bancos de dados. Esta linguagem é chamada de Diagrama de Entidades e Relacionamentos (DER).

7.2.1 Entidades, Atributos e Conjuntos de Valores

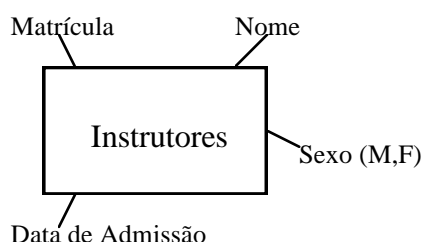
Os conjuntos de informação darão origem as entidades do MER. Cada entidade irá representar todos os elementos de um conjunto de informação. Elas serão modeladas com retângulos e com o nome do conjunto escrito dentro dele no plural porque a entidade representa todos os elementos daquele conjunto. Por exemplo: a entidade Instrutores não representa somente o instrutor Robson e sim todos os instrutores do SENAC.

Instrutores

Os atributos são as informações de uma determinada entidade. A entidade instrutores, ou seja todos os instrutores podem ter matrícula, nome e data de admissão. Turmas podem ter código, data de início, data de término, horário, professor, sala e disciplina.



Percebam que o atributo representa todos os possíveis valores que aquela informação pode assumir. O atributo nome, por exemplo, representa todos os nomes de instrutores do SENAC. Ou seja, o atributo nome representa o conjunto de valores nome de instrutores. Em alguns casos, o conjunto de valores que um atributo pode referenciar é limitado e pode ser definido. Se na entidade instrutores tivéssemos um atributo sexo. O conjunto de valores que este atributo pode assumir é M ou F.



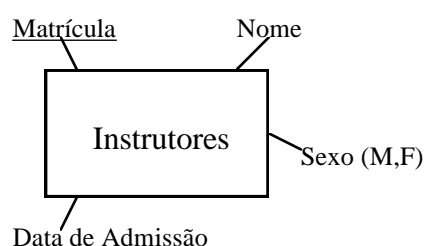
Como a entidade representa todos os elementos de um conjunto de informações, chamaremos de instância da entidade um determinado elemento dela. Uma instância da entidade instrutores poderia ser:

Matrícula	Nome	Admissão	Sexo
1230	NICHOLAS SERGEY	01/01/93	M

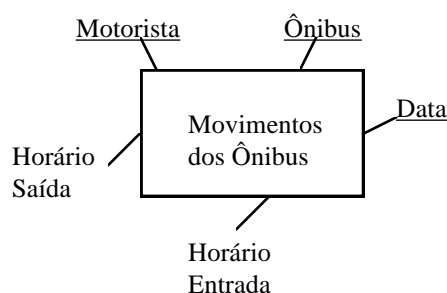
7.2.2 Atributo Determinante e Atributo Multivalorado

Em cada entidade temos que ter um atributo (ou uma junção de atributos) que tenha um valor único para cada instância. Ou seja, este valor aparecerá somente em uma instância e, para cada outra instância, teremos também valores que são exclusivos dela. Damos o nome de atributo determinante a este, ou estes, atributo. Ele recebe este nome porque através dele nós determinamos os outros atributos. Isto é chamado dependência funcional, ou seja, os outros atributos

dependem do atributo determinante. No caso da entidade instrutores, o atributo determinante é Matrícula, pois cada instrutor tem seu número de matrícula que é exclusivo dele. No caso da turma é o código. Cada turma do SENAC tem seu código e ele não será usado para nenhuma outra. Para identificar quais atributos são determinantes, sublinharemos eles.



Em alguns casos, para termos um valor único que identifique somente uma instância da entidade, teremos que concatenar dois ou mais atributos. No caso de uma empresa de ônibus que queira saber qual motorista estava responsável pelo ônibus em determinado dia. Sabendo que cada ônibus somente pode ser usado por um motorista a cada dia, abaixo teríamos um modelo para armazenar essas informações:

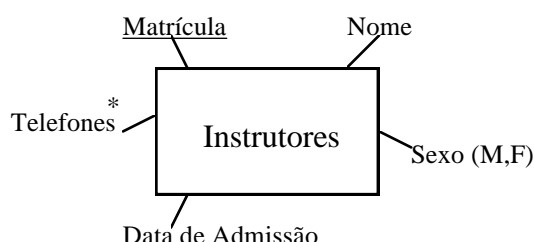


Neste caso, para termos um valor que fosse exclusivo para cada instância, nós concatenamos três atributos. Pela tabela abaixo, perceba que essa foi uma boa saída:

Motorista	Ônibus	Data	Saída	Entrada
0989	0010/92	05/06/97	08:00	16:00
0989	0010/92	06/06/97	07:30	15:00
0712	0032/93	05/06/97	07:00	16:00
0809	0032/93	06/06/97	07:00	16:00

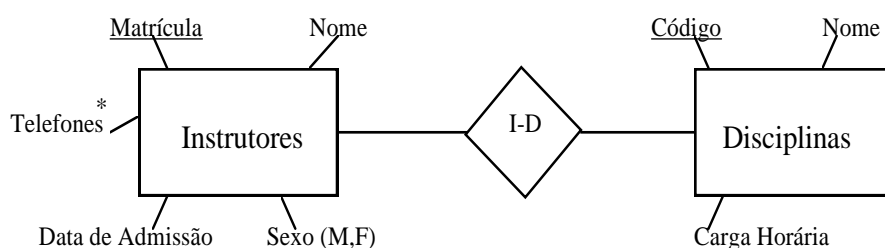
Existe a possibilidade de com os atributos naturais de uma entidade você não encontrar um que possa ser determinante. Neste caso o projetista tem que criar um atributo artificial que será determinante. Geralmente um código ou coisa semelhante.

Um atributo é dito multivalorado quando ele pode conter diversos valores. Imagine que nós resolvamos armazenar, para cada instrutor, os números de telefone. Muitos têm o telefone residencial, telefone celular e telefone do escritório (ou de outro trabalho). Cada atributo somente pode armazenar um valor. Para poder armazenar vários valores em um atributo ele tem que ser do tipo multivalorado. Para indicar que um atributo é multivalorado usamos um asterisco(*).



7.2.3 Relacionamentos

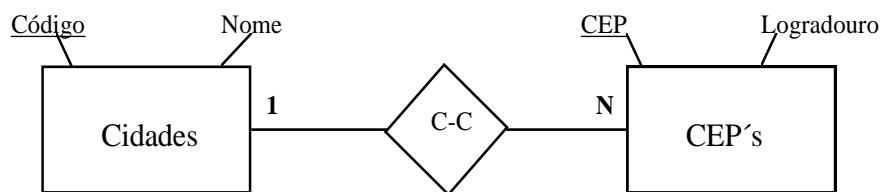
Em um sistema, as entidades não ficam isoladas sem nenhuma ligação com outros conjuntos de informação. A não ser que o sistema só tenha um conjunto. As entidades em um sistema estão ligadas através de relacionamentos. Um instrutor está habilitado a ministrar disciplinas. As turmas estão localizadas em salas. Uma turma tem um professor. No MER nós vamos mostrar estes relacionamentos. Um relacionamento é representado por um losango que é colocado entre as duas entidades, ou mais, que ele liga. Dentro do losango pode ser escrito um nome para o relacionamento, mas isso não é tão importante. Muitos projetistas colocam as iniciais das entidades ligadas.



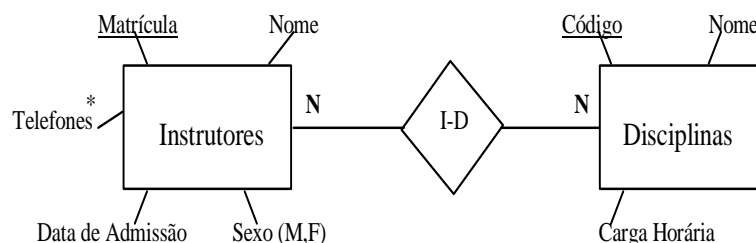
7.2.3.1 Classes de Relacionamentos

Não basta dizer que existe um relacionamento entre determinadas entidades. É preciso dizer também como os elementos de uma entidade se ligam aos elementos de outra. Por exemplo, um elemento do conjunto professores pode ligar-se a quantos do disciplinas? E um do disciplinas pode relacionar-se a quantos de instrutores? A resposta destas perguntas irá resultar na classe do relacionamento.

As classes de relacionamento principais são a 1:N e a N:N. Traduzindo: um para vários (ou vários para um) e vários para vários. Todas as outras classes são simplificações destas duas. Na 1:N temos que um elemento de uma entidade X liga-se a vários de outra entidade Y relacionada, e que um elemento da entidade Y relacionada, liga-se a somente um da entidade X. Como exemplo temos: uma cidade do Brasil pode ter vários CEP's, porém um CEP só está relacionado a uma cidade.



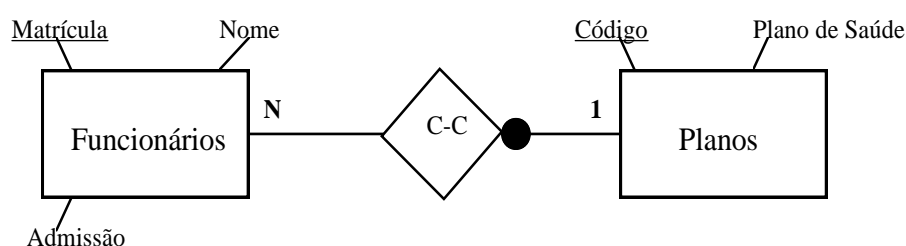
Já no relacionamento N:N, um elemento de uma entidade X liga-se a vários elementos da entidade Y, e um elemento de Y liga-se a vários de X. No exemplo dado acima do relacionamento entre instrutores e disciplinas, temos um relacionamento N:N.



Alguns projetistas gostam de criar simplificações e/ou limitações das classes, tais como: relacionamentos 1:3, 2:5, mas no nosso entender isso gera problemas. Ao definir, por exemplo, que em uma locadora cada usuário tem no máximo 3 dependentes, mesmo que esta seja a política atual, você estará limitando o seu sistema.

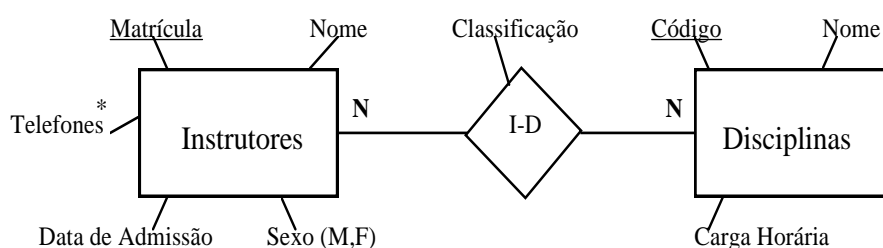
7.2.3.2 Relacionamentos Totais e Parciais

Quando estabelecemos um relacionamento dizemos que existem elementos de uma entidade que estão ligados a outra, mas será que todos os elementos tem ligação obrigatoriamente. Imagine que uma empresa tem um sistema que informa a qual plano de saúde o funcionário está associado. Alguns funcionários não têm plano de saúde e a empresa não os obriga a ter. Neste caso, nem todos os elementos de funcionários estarão ligados a elementos da entidade planos de saúde. Assim sendo, temos um relacionamento parcial no sentido funcionários-planos. Já no sentido contrário, podemos colocar o relacionamento como total. Somente registraremos planos que tenham ligação com a empresa. Para representar relacionamentos totais, colocamos uma “bolinha” preta ao lado da entidade que dá origem a um relacionamento total.



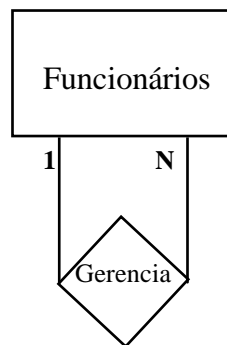
7.2.3.3 Atributos de Relacionamentos

Comentamos anteriormente sobre atributos de entidades, porém os relacionamentos também têm suas informações. Informações que não aparecem nas entidades separadamente. Somente quando você liga elas é que vão surgir informações da ligação. Para exemplificar, imagine que cada instrutor do SENAC recebe uma classificação de acordo com cada disciplina que ele está habilitado a lecionar. Ele pode ser Bom, Ótimo ou Excepcional. Percebam que um instrutor pode ser bom em uma disciplina e excepcional em outra. Temos então uma informação dos instrutores, pois a classificação não é do instrutor e sim do seu conhecimento de uma determinada disciplina. Nem de disciplinas, pois a classificação também não é dela. A classificação é um atributo do relacionamento.



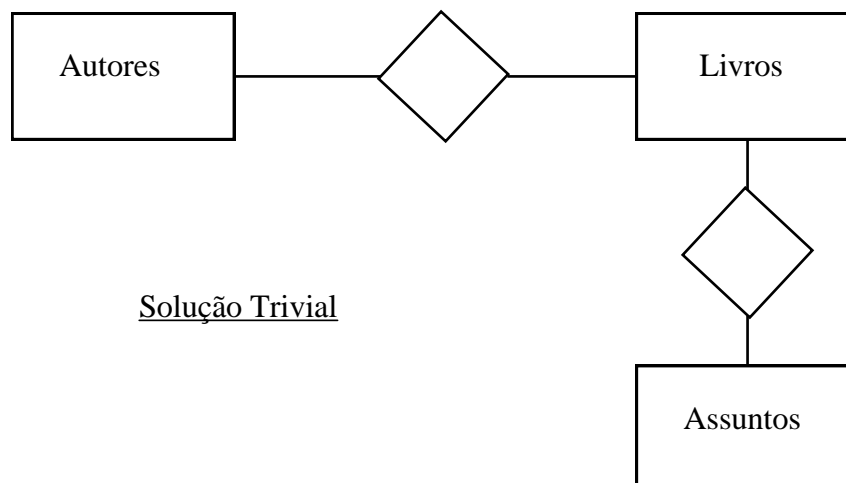
7.2.3.4 Auto-relacionamento

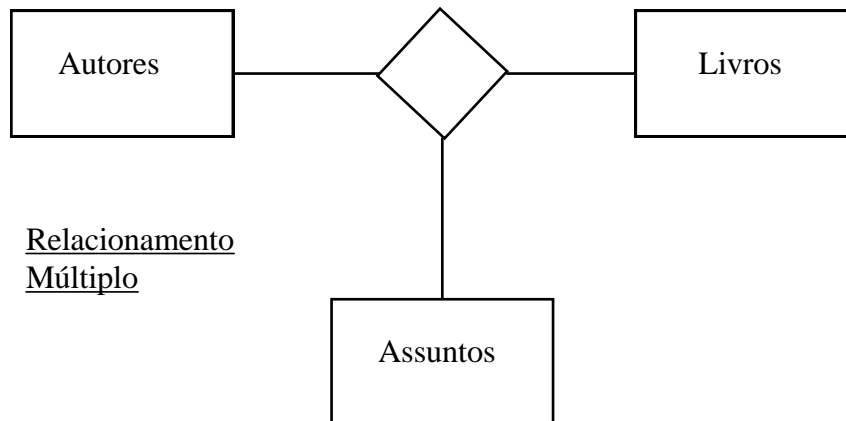
Uma entidade pode relacionar-se com ela mesma. O caso mais trivial é o de funcionários que gerenciam outros: um pode gerenciar vários e pode ser gerenciado diretamente por um. Nesse caso, temos um auto-relacionamento 1:N.

7.2.3.5 Relacionamentos Múltiplos

18

Nos relacionamentos vistos até agora, a ligação é sempre dupla, uma entidade relaciona-se com outra. Este é o tipo mais comum de ligação. Porém com alguma frequência você encontra nos sistemas ligações triplas e até quádruplas. Nestes casos, nós dizemos que existe um relacionamento múltiplo. No exemplo abaixo, temos um caso de ligação múltipla. Os autores escrevem livros. E geralmente nós associamos os assuntos aos livros. Porém imagine que em uma biblioteca você quisesse saber sobre que assuntos um autor já escreveu. Teríamos então uma tripla: autor, livro e assunto. Abaixo os modelos de duas soluções, um com a solução geralmente dada e outro com o relacionamento múltiplo.

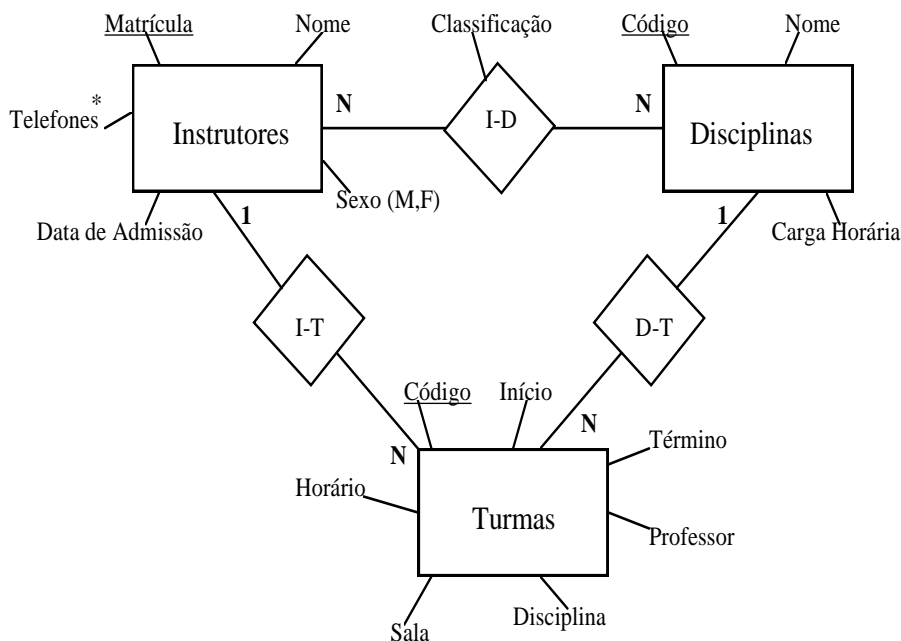




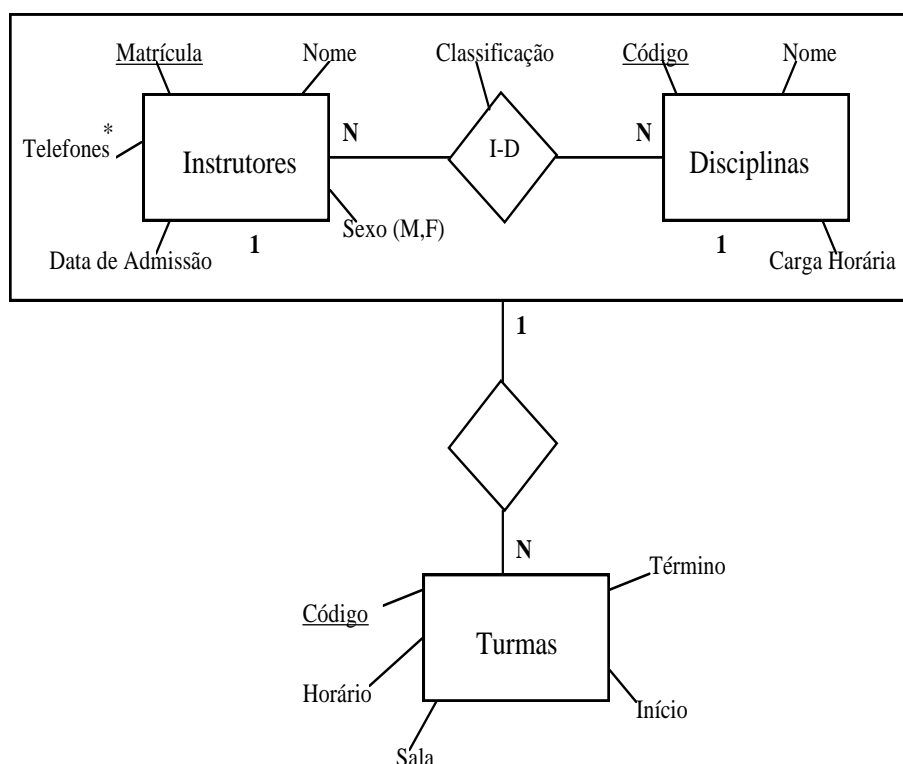
Perceba que pelas duas soluções eu poderia descobrir quais os assuntos já abordados por determinado autor, mas a solução com relacionamento múltiplo é a mais eficiente porque evita duplicação de informação e a recupera.

7.2.3.6 Agregações

Em alguns casos de relacionamentos, temos uma ligação de uma entidade com outro relacionamento. A entidade não relaciona-se com outra mas com uma ligação entre duas entidades. Voltemos ao exemplo do SENAC. Sabemos que uma turma tem um instrutor e uma disciplina. Uma maneira de modelar isto seria:

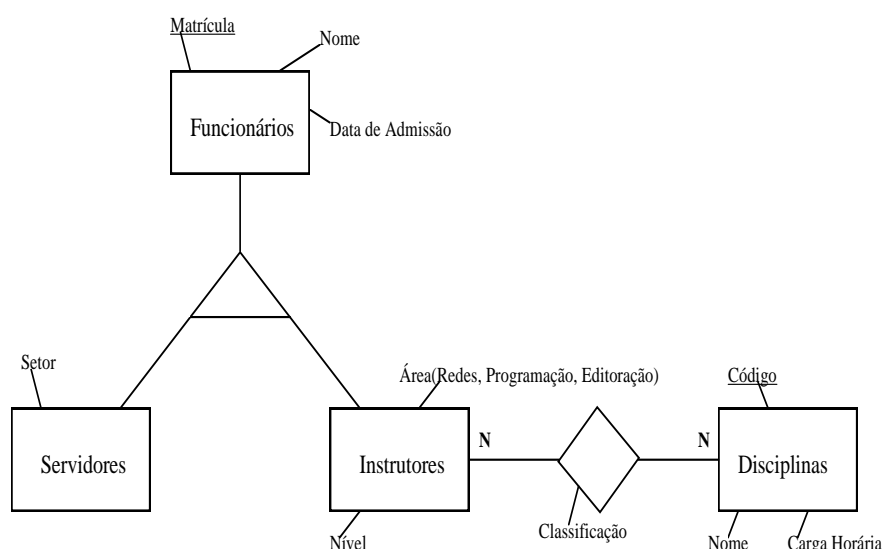


Às vezes, um conjunto de informações pode conter subconjuntos. Por exemplo, uma entidade funcionários de uma empresa pode conter os subconjuntos motoristas, vendedores, secretárias, etc. Funcionários não deixa de ser uma entidade, porém percebe-se que, para cada subconjunto, nós teremos atributos e relacionamentos específicos. Para as secretárias pode interessar o registro de quais idiomas ela fala, mas para motoristas, não. Para o vendedor registraremos qual seu setor de vendas, mas para secretárias, não. Motoristas estão relacionados a veículos, mas secretária e vendedores, não. Para evitar este problema, a entidade é particionada e os atributos e relacionamentos específicos de cada subconjunto são alocados a ele.



Para ilustrar este tipo de ligação, vamos usar um exemplo baseado no SENAC. Neste existem dois tipos de funcionários: servidores e instrutores. Os servidores são o pessoal de secretária, serviços gerais, coordenação e

administração. Para a empresa interessa registrar qual o setor de cada servidor, mas para instrutores, não. Cada instrutor está relacionado às disciplinas que ele pode ministrar e tem registrada qual sua área de atuação e nível salarial, porém os servidores não.

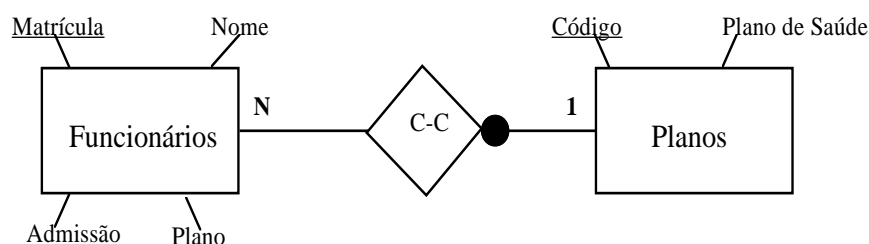


7.2.4 Decomposição de Relacionamentos

Comentamos anteriormente que não tínhamos métodos formais de passagem do mundo real para o modelo descritivo e deste para o conceitual, mas na passagem do nível conceitual ao operacional, temos um processo eficiente chamado decomposição. O objetivo deste método é deixar as estruturas de informação no “ponto” de virarem bancos de dados automatizados.

7.2.4.1 Relacionamentos 1:N

Nos relacionamentos 1:N somente precisamos acrescentar na entidade que fica do lado N do relacionamento um atributo de ligação. Lembre-se de que a entidade que está do lado N está ligada a somente um elemento do outro conjunto. O valor do atributo de ligação é justamente o atributo determinante da entidade do lado 1. Perceba que o atributo de ligação é necessário para informar a qual elemento do conjunto do lado 1, o elemento do conjunto do lado N está ligado. Como exemplo, usaremos o relacionamento funcionários / planos de saúde, já abordado anteriormente.

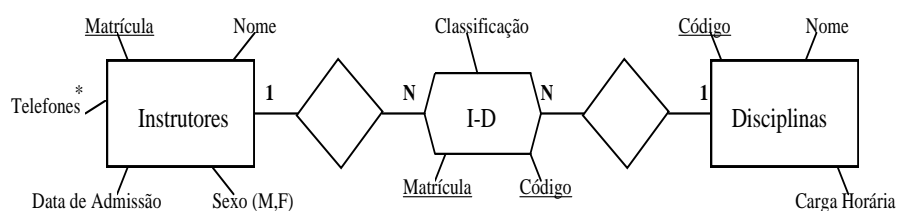


Para resolver este problema, criamos um atributo de ligação na entidade funcionários chamado plano. Este atributo irá armazenar o código do plano de saúde ao qual o funcionário está relacionado.

7.2.4.2 Relacionamentos N:N

Na decomposição de um relacionamento N:N surgirá um conjunto auxiliar que nós representaremos por um hexágono para não confundir com as entidades naturais do sistema. Essa entidade auxiliar conterá os atributos determinantes das duas entidades relacionadas além dos atributos do relacionamento (se eles existirem). A entidade auxiliar estará relacionada com as duas entidades naturais através de relacionamentos 1:N.

Para exemplificar, vamos decompor o relacionamento instrutores com disciplinas.



Perceba que os atributos determinantes herdados das entidades naturais serão atributos determinantes da entidade auxiliar. Para melhorar o entendimento dessa decomposição, vamos ver exemplos de instâncias das três entidades.

Instrutores

0988	MAX	01/06/92	M
1450	BARTIRA	06/06/96	F
1340	ANDREI	01/01/96	M

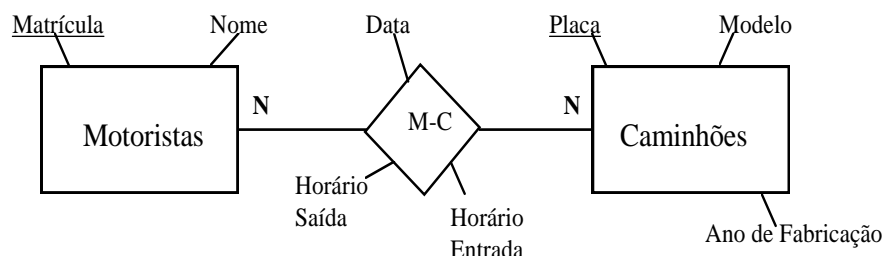
Disciplinas

W95	WINDOWS 95	40
ACC	ACCESS 7	40
IDS	INT. AO DES. DE SISTEMAS	28
BDR	BANCO DE DADOS RELACIONAL	50

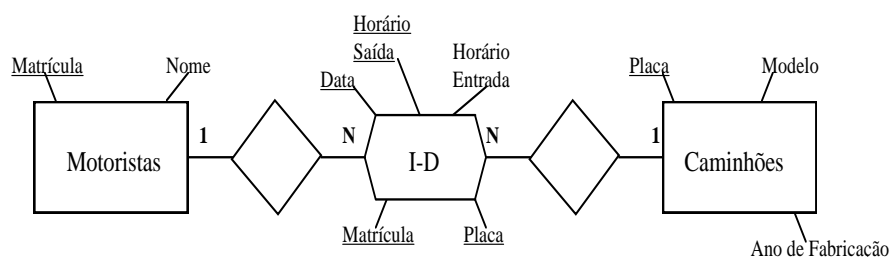
I-D

0988	W95	BOM
1450	W95	ÓTIMO
0988	BDR	EXCELENTE
1340	IDS	BOM
1450	ACC	BOM

Agora é bom ressaltar que os atributos herdados farão parte da determinação da entidade auxiliar, mas podemos ter atributos do relacionamento fazendo parte da determinação. Veja abaixo o exemplo do controle de caminhões de uma empresa que presta serviços de limpeza. Um mesmo caminhão pode ser usado por diferentes motoristas em um mesmo dia, porém eles pegam o caminhão na garagem da empresa em horários diferentes. Primeiro vamos ver o modelo conceitual.



Agora a decomposição.

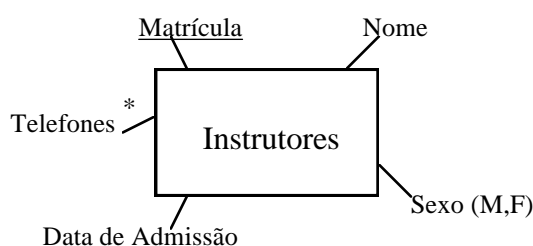


A determinação do conjunto auxiliar é dada pela junção de matrícula, placa, data e horário de saída. Isto porque a combinação matrícula e placa não seriam únicos, pois o motorista poderia vir a usar o mesmo caminhão em outro dia. A junção matrícula, placa e data também não é suficiente, pois o motorista pode vir a usar o mesmo caminhão no mesmo dia em horários de saída diferentes.

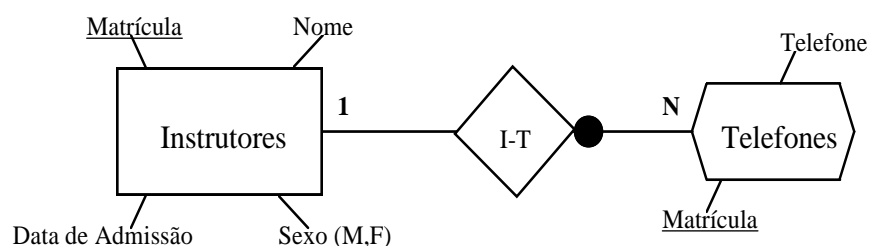
7.2.4.3 Atributos Multivalorados

Alguns Sistemas Gerenciadores de Bancos de Dados (SGBD) implementam atributos multivalorados diretamente sem precisar de nenhuma adequação, porém a maioria deles não tem este recurso. Por isso, nós também vamos decompor os atributos multivalorados.

Para decompor um atributo multivalorado, criamos uma entidade auxiliar relacionada à entidade natural. Nesta entidade criada, nós teremos como atributos o atributo determinante da entidade natural, estabelecendo uma ligação entre os dois conjuntos, e o atributo que era multivalorado no conjunto natural e que na entidade auxiliar não é. Para exemplificar, vamos usar a entidade instrutores com o atributo telefones multivalorado. Abaixo vemos o modelo conceitual.



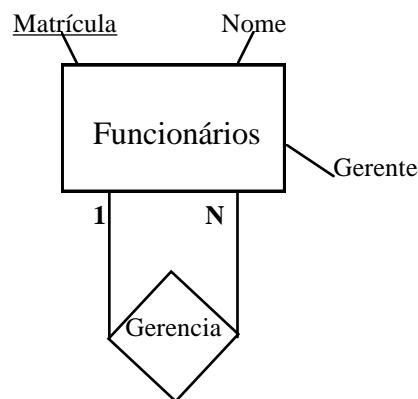
E agora a decomposição:



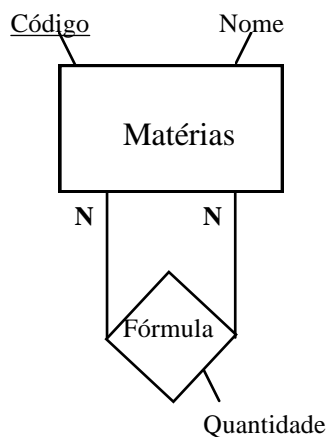
Percebam que o relacionamento criado é necessariamente 1:N e total no sentido entidade auxiliar / entidade natural.

7.2.4.4 Auto-relacionamentos 1:N e N:N

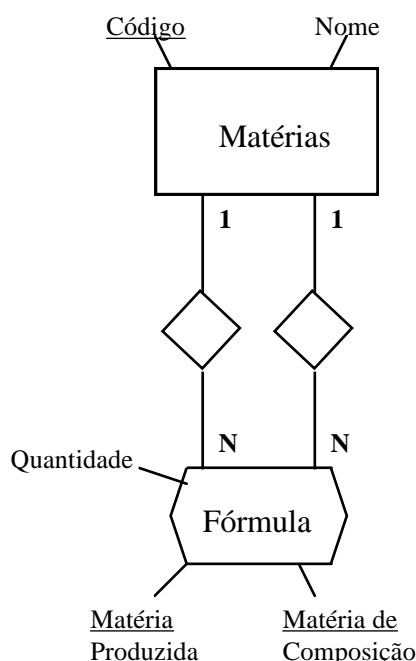
Primeiro vamos ver como vamos decompor um auto-relacionamento 1:N. Vamos usar um exemplo já visto, o de funcionários que gerenciam e são gerenciados por outros funcionários. Percebam que cada funcionário só é gerenciado diretamente por um outro. Neste caso, criamos um atributo de ligação na entidade funcionários chamado gerente que estabelecerá o relacionamento. O atributo de ligação irá conter o número de matrícula do gerente do funcionário.



Já em auto-relacionamentos N:N existe a necessidade de criarmos uma entidade auxiliar. Vejamos, primeiro, um exemplo de auto-relacionamento N:N. Em uma indústria farmacêutica algumas matérias-primas são compradas a terceiros, mas a maioria da matéria-prima da empresa é fabricada nela mesma a partir de outras matérias, ou seja, uma matéria é composta da manipulação de outras e uma matéria pode compor a fórmula de várias outras. Abaixo o modelo deste auto-relacionamento.



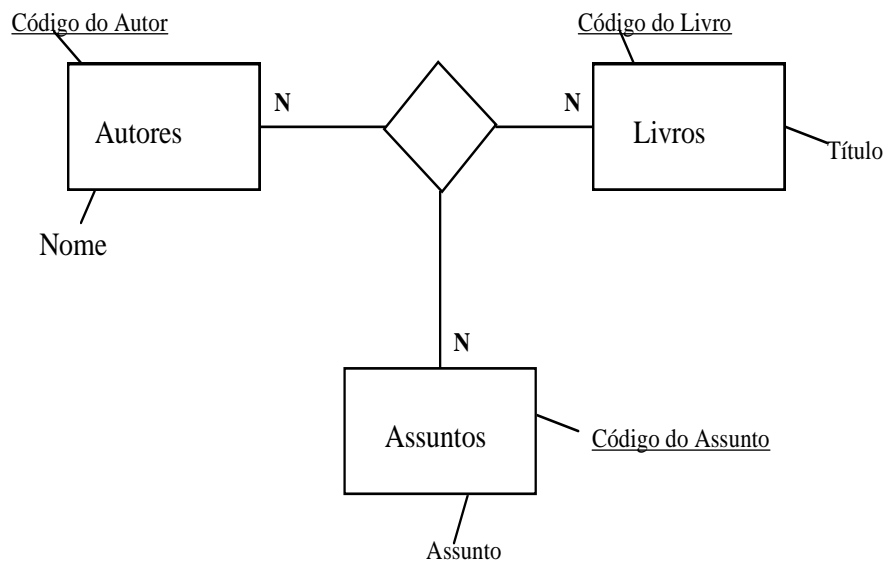
Acrescentamos um atributo ao relacionamento para registro da quantidade utilizada na formulação. Agora vamos decompor.



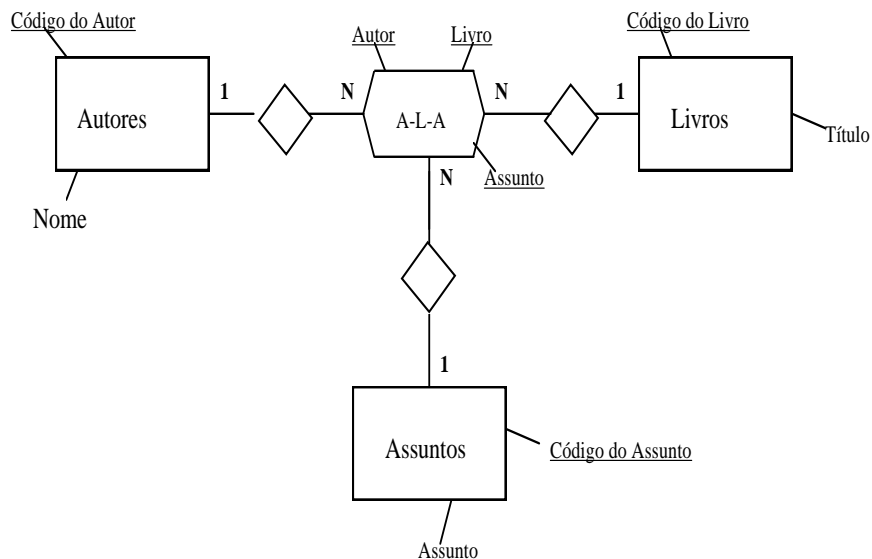
Perceba que existe similaridade entre a decomposição de auto-relacionamento 1:N com relacionamento 1:N e de auto-relacionamento N:N com relacionamento N:N.

7.2.4.5 Relacionamentos Múltiplos

Seguindo o mesmo raciocínio de quando decompomos relacionamentos N:N, vamos criar uma entidade auxiliar. No caso de relacionamento N:N, a entidade registrava os pares de ligação. Em relacionamento múltiplos, iremos registrar triplas, quadras ou quinas, de acordo com a quantidade de entidades envolvidas na ligação. O conjunto auxiliar herda os atributos determinantes de todas as entidades naturais e elas farão parte da determinação da entidade auxiliar. Vejamos primeiro um MER com relacionamento múltiplo.

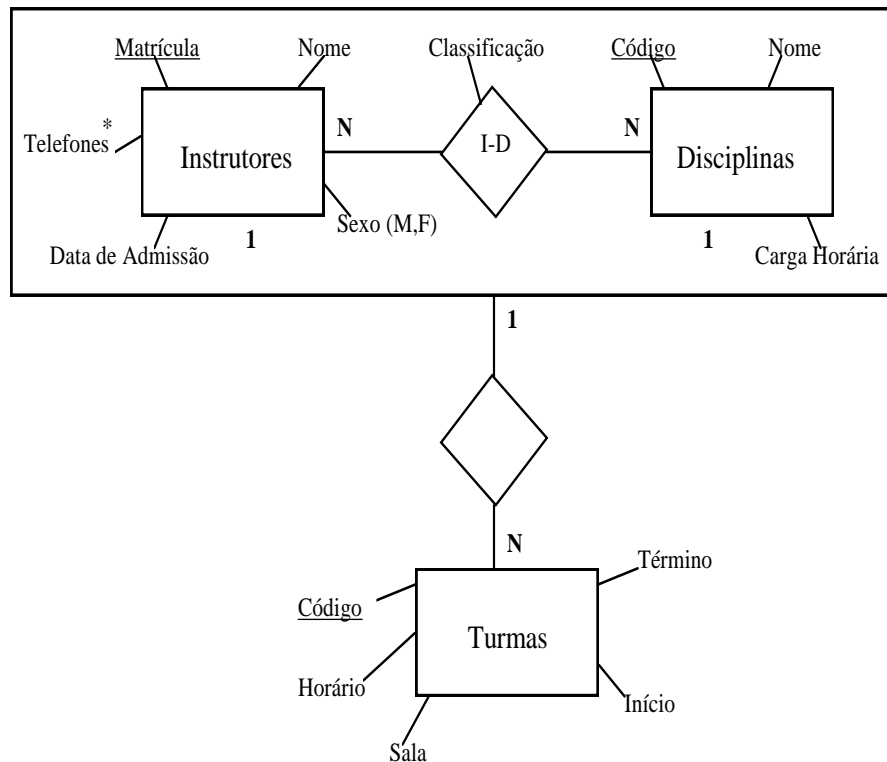


E agora a decomposição dele.

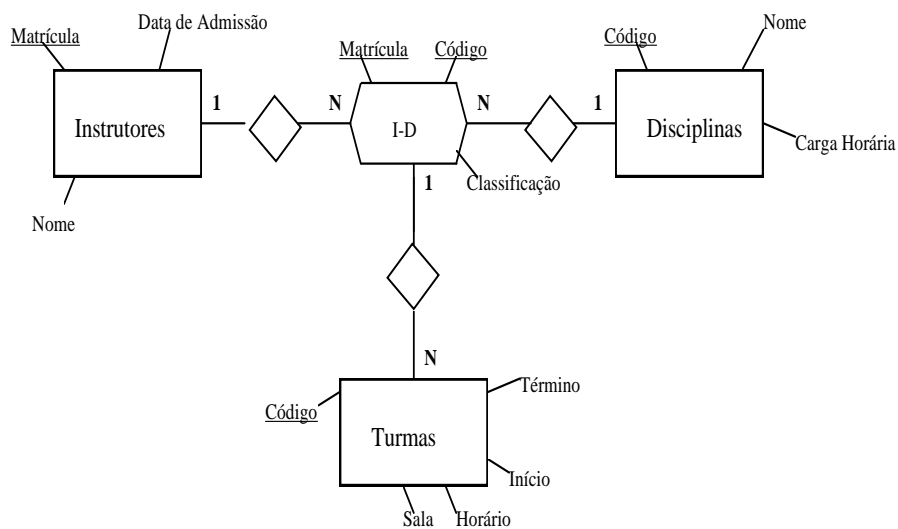


7.2.4.6 Agregações

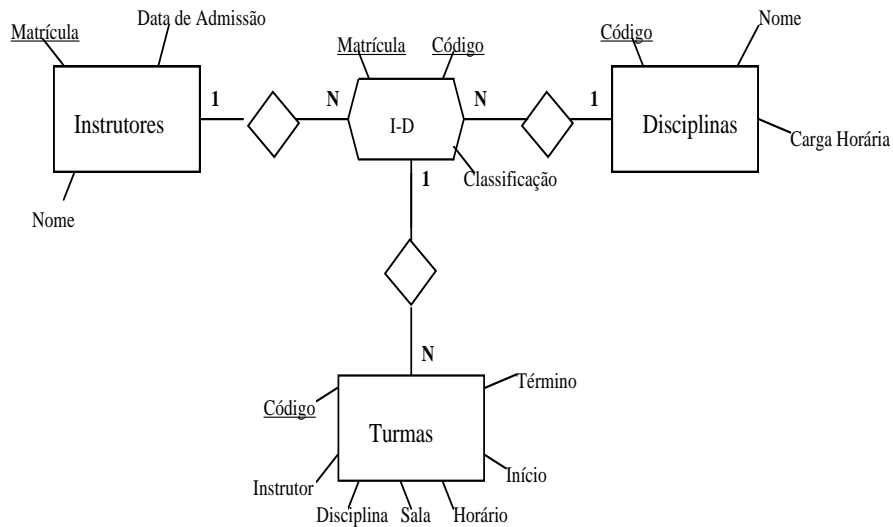
Lembrando que, nas agregações temos uma entidade que liga-se a um relacionamento, então, primeiramente temos que decompor este relacionamento. Vamos usar o exemplos mostrados anteriormente de instrutores e disciplinas agregados e relacionados a turmas.



Agora vamos decompor o relacionamento instrutores / disciplinas.

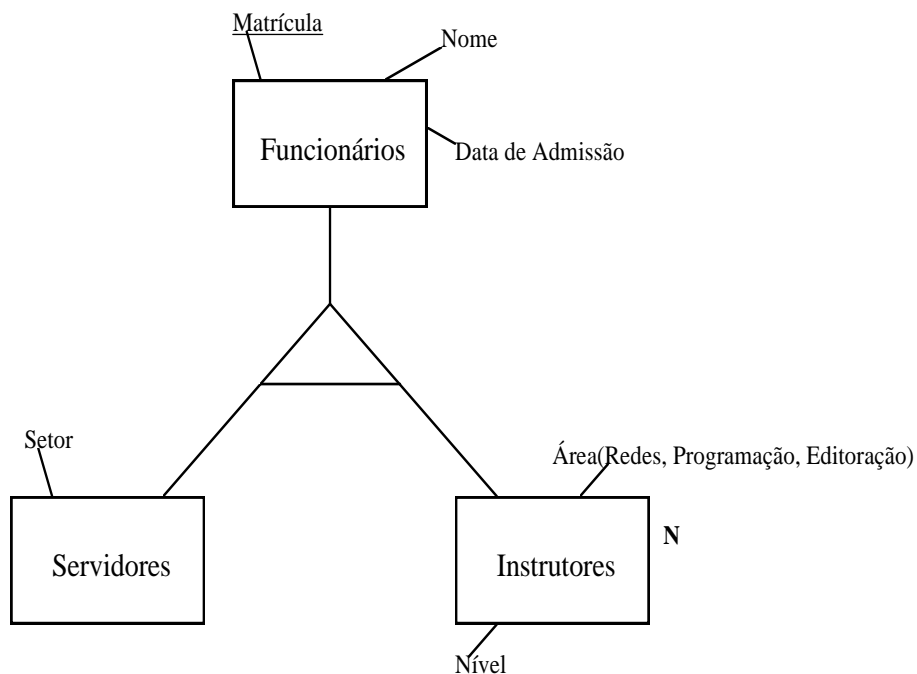


Aparece agora entre a entidade auxiliar I-D e turmas um relacionamento 1:N. Quando isto acontece criamos atributos de ligação na entidade que está do lado N, neste caso turmas. Abaixo o modelo decomposto final.

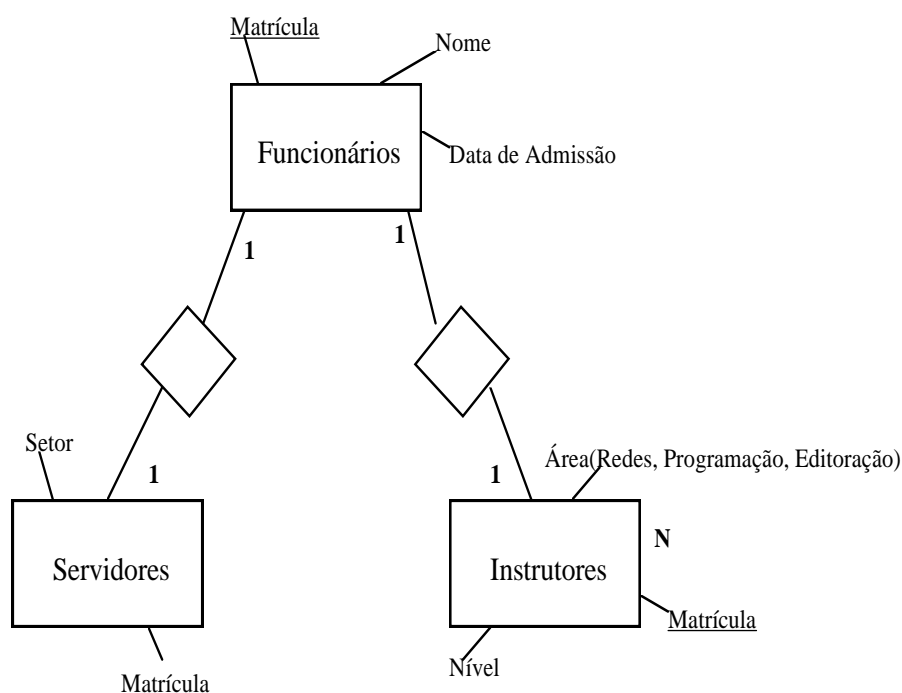


7.2.4.7 Particionamentos

Se você prestar atenção em um particionamento, verá que na realidade o que temos ali é um relacionamento 1:1. O relacionamento 1:1 é uma simplificação do 1:N. Então, a solução é a mesma. Exportar o atributo determinante para a entidade relacionada. Vejamos um exemplo.



Vejam agora a decomposição.



7.3 VALIDAÇÃO DO MER

Terminada a decomposição do nosso modelo conceitual, que garantias nós temos de que realmente nosso modelo obedece aos requisitos de um bom projeto de banco de dados? Será que existe alguma redundância desnecessária? Para dar maiores garantias ao nosso modelo, veremos a partir de agora algumas técnicas de verificar a qualidade do nosso modelo. Propomos aqui uma abordagem diferente da geralmente adotada pelos projetistas. Vamos utilizar um processo chamado normalização para validar nosso DER decomposto. Geralmente, a normalização é utilizada antes de se escreverem os diagramas e é baseada em uma ferramenta chamada dicionário de dados. Esta ferramenta é usada pelos analistas de sistemas para levantar as informações de um sistema. Como muitas vezes, nós projetamos os BD's sem ter usado o dicionário de dados, pensamos em usar a normalização como um método de validação final. Por ter regras rigorosas, ela presta-se muito bem a este serviço.

7.3.1 Normalização

A normalização, como o próprio nome já diz, é o processo de adequar um objeto a determinadas normas. Ela consiste de seis normas chamadas formas normais. Cada forma normal tem um requisito a ser atingido, visando eliminar redundâncias da sua estrutura de BD. Apesar de não existir uma seqüência de aplicação das normas, é interessante seguir a ordem apresentada aqui para obter um melhor resultado.

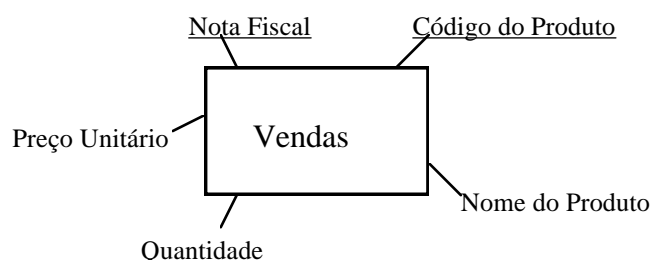
7.3.2 AS FORMAS NORMAIS

7.3.2.1 Primeira Forma Normal

Esta é a forma normal que “detona” os atributos multivalorados. Ela simplesmente exige que não exista repetição de informação em uma instância de um conjunto de informações. Aplique esta exigência em todas as suas entidades e verifique se não foi esquecida a decomposição de algum atributo multivalorado.

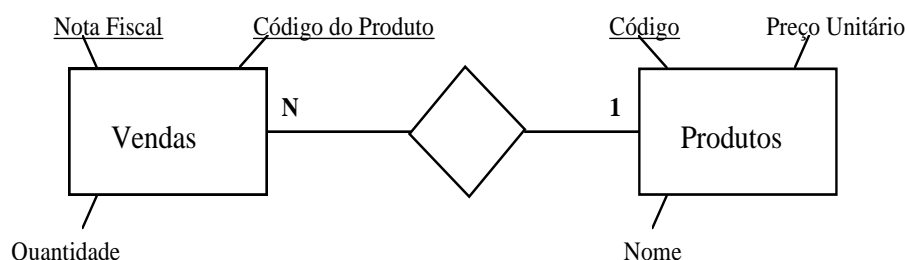
7.3.2.2 Segunda forma Normal

Essa norma exige que, em nossas entidades, não tenhamos informações que dependam parcialmente dos atributos determinantes. Para exemplificar, vejamos o exemplo de um conjunto que não está na 2ª forma normal. A entidade abaixo representa as vendas de uma loja. Em cada venda é registrada o número da nota, código do produto, nome do produto, preço unitário e quantidade vendida.



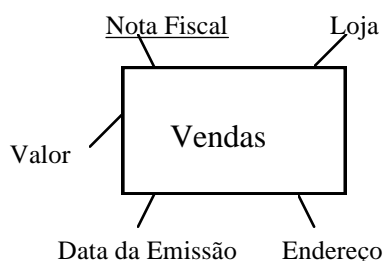
Percebam que nome do produto e preço unitário não dependem de nota fiscal, e sim de código do produto. Temos aí uma dependência parcial em relação aos atributos determinantes. Já a quantidade está correta pois depende de nota

fiscal e código. Isso indica que temos um conjunto de informação que não foi identificado pelo projetista: o conjunto produtos. Vamos ver como resolvemos isso.

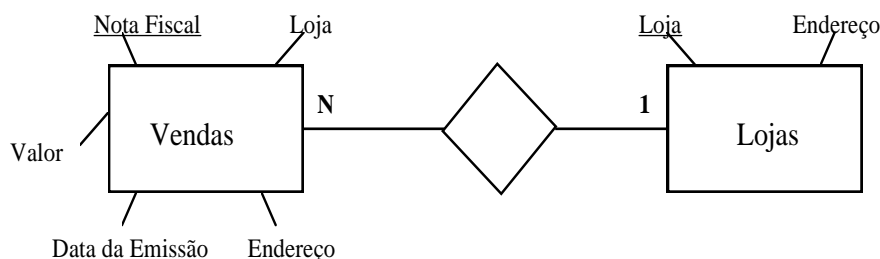


7.3.2.3 Terceira Forma Normal

Na 3ª forma temos que obedecer ao seguinte critério: não podemos ter, em nossos conjuntos, informações que dependam de um atributo que não é determinante. Vejamos o exemplo de um conjunto de informações de faturamento de uma rede lojas. Para cada venda é registrado: número da nota, data de emissão, loja, endereço da loja e valor.



O atributo determinante é o número da nota, porém o endereço da loja não depende dele. O endereço depende do atributo loja. Isso indica que você esqueceu de modelar um conjunto de informações, que nesse caso é lojas.



Bibliografia

SETZER, Valdemar W. " Bancos de Dados", Editora Edgard Blucher LTDA, 1989.

SILBERSCHATZ, Abraham.Horth, Henry F., Sudarshan. S.Sistema de Bancos de Dados. Makron Books.