

# Banco de Dados

Profa. Elaine J.C.Brito

## **INTRODUÇÃO**

Estamos vivendo “A Era da informação” na qual as informações valem muito mais que as máquinas. Podemos notar uma busca cada vez maior de meios para que o armazenamento, a transmissão e a recuperação de informação sejam cada vez mais rápidos e seguros.

Um desses meios é o banco de dados, que é um conjunto de informações manipuláveis de mesma natureza inseridas em um mesmo local, obedecendo a um padrão de armazenamento. Alguns exemplos são: agenda telefônica, lista de compras, dicionário, entre outros.

Os exemplos anteriores são bancos de dados manuais, o que estudaremos é a informatização deles.

O banco de dados ou base de dados é basicamente um sistema de manutenção de informações por computador, capaz de manter as informações organizadas e torna-las disponíveis quando solicitadas, com rapidez e confiabilidade. Existem várias ferramentas para este fim, como gerenciadores de bancos de dados: Access, MySQL que estudaremos nesta disciplina.

## **POR QUE BANCO DE DADOS?**

- O banco de dados é compacto, pois elimina o volume de arquivos de papéis.
- É rápido, pois pode recuperar e modificar os dados muito mais rapidamente do que o ser humano.
- Permite integração e evita redundância. O sistema de banco de dados proporciona ao usuário o controle centralizado de seus dados operacionais, que contrasta nitidamente com o que pode ser observado em uma empresa sem um sistema automático, em que cada aplicação dispõe de seus próprios arquivos de tal forma que os dados operacionais são muito dispersos, dificultando o controle sistemático.
- Compartilhamento. Parcelas isoladas de dados podem ser compartilhadas por diversos usuários num banco de dados. É o fluxo corrente da disponibilidade de informações certas e atualizadas.
- Acesso com restrições de segurança.
- Padrões bem-definidos, evitando falta de informações ou informações insatisfatórias no banco de dados.

## **SISTEMA GERENCIADOR DE BASE DE DADOS (SGBD)**

Um Sistema Gerenciador de Base de Dados (SGBD) é uma coleção de programas que permitem aos usuários criarem e manipularem uma base de dados. Um SGBD é, assim, um sistema de software de propósito geral que facilita o processo de definir, construir e manipular bases de dados de diversas aplicações.

Definir uma base de dados envolve a especificação de tipos de dados a serem armazenados na base de dados.

Construir uma base de dados é o processo de armazenar os dados em algum meio que seja controlado pelo SGBD.

Manipular uma base de dados indica a utilização de funções como a de consulta, para recuperar dados específicos, modificação da base de dados para refletir mudanças no mundo (inserções, atualizações e remoções), e geração de relatórios.

A base de dados e o software de gerenciamento da base de dados compõem o chamado Sistema de Base de Dados.

## MODELOS DE DADOS

O primeiro Sistema Gerenciador de Banco de Dados (SGBD) comercial surgiu no início da década de 60 com base nos sistemas de arquivos disponíveis na época, os quais não controlavam o acesso concorrente por vários usuários ou processos. Os SGBDs evoluíram e passaram a utilizar diferentes formas de representação, ou modelos de dados, para descrever a estrutura das informações contidas em seus bancos de dados. Atualmente, os seguintes modelos de dados são utilizados pelos SGBD's: modelo hierárquico, modelo em redes, modelo relacional (amplamente usado e que estudaremos nesta disciplina) e o modelo orientado a objetos.

## VISÃO DOS DADOS

O maior benefício de um banco de dados é proporcionar ao usuário uma visão abstrata dos dados. Isto é, o sistema acaba por ocultar determinados detalhes sobre a forma de armazenamento e manutenção desses dados.

## ABSTRAÇÃO DE DADOS

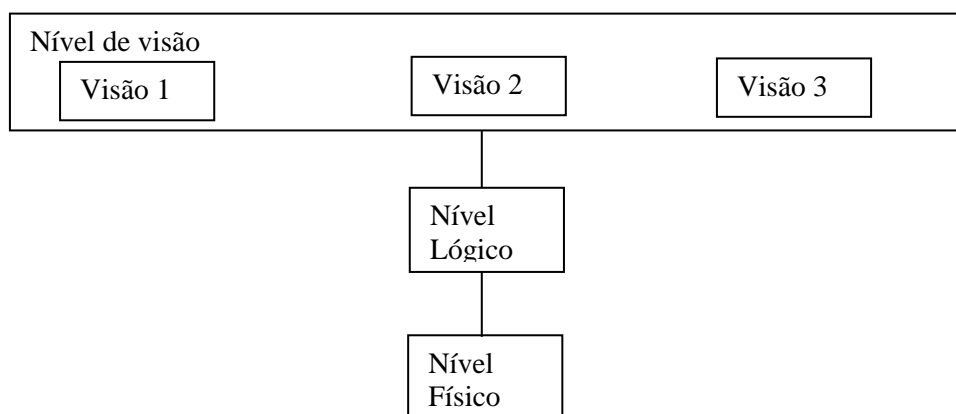
Para que se possa usar um sistema, ele precisa ser eficiente na recuperação das informações. Esta eficiência está relacionada à forma pela qual foram projetadas as complexas estruturas de representação desses dados no banco de dados. Já que muitos dos usuários dos sistemas de banco de dados não são treinados em computação, os técnicos em desenvolvimento de sistemas omitem essa complexidade desses usuários por meio dos diversos níveis de abstração, de modo a facilitar a interação dos usuários com o sistema:

**Nível físico.** É o mais baixo nível de abstração que descreve como esses dados estão de fato armazenados. No nível físico, estruturas de dados complexas de nível baixo são descritas em detalhes.

**Nível lógico.** Este nível médio de abstração descreve quais dados estão armazenados no banco de dados e quais os inter-relacionamentos entre eles. Assim, o banco de dados como um todo é descrito em termos de um número relativamente pequeno de estruturas simples. Embora a implementação dessas estruturas simples no nível lógico possa envolver estruturas complexas no nível físico, o usuário do nível lógico não necessariamente precisa estar familiarizado com essa complexidade. O nível lógico de abstração é utilizado pelos administradores do banco de dados que precisam decidir quais informações devem pertencer ao banco de dados.

**Nível de visão.** O mais alto nível de abstração descreve apenas parte do banco de dados. Apesar das estruturas simples do nível lógico, alguma complexidade permanece devido ao tamanho dos bancos de dados. Muitos dos usuários de banco de dados não precisam conhecer todas as suas informações. Pelo contrário, os usuários normalmente utilizam apenas parte do banco de dados. Assim, para que estas interações sejam simplificadas, um nível de visão é definido. O sistema pode proporcionar diversas visões do mesmo banco de dados. Cada visão descreve, tipicamente, a parte da base de dados que um particular grupo de usuários está interessado e esconde deste o restante da base de dados.

O inter-relacionamento entre esses três níveis de abstração está ilustrado na figura a seguir :



## MODELO ENTIDADE-RELACIONAMENTO (MER)

O MER é um modelo de dados conceitual de alto-nível, ou seja, seus conceitos foram projetados para serem compreensíveis a usuários, descartando detalhes de como os dados são armazenados. Esse modelo pode ser expresso graficamente por meio do diagrama E-R, assim cada elemento do MER tem uma notação própria com a qual ele é representado no DER.

### CONJUNTO DE ENTIDADES

Uma **entidade** é uma "coisa" ou um "objeto" do mundo real que pode ser identificado por outros objetos. Por exemplo, cada pessoa é uma entidade, cada conta de cliente de um banco também podem ser considerada uma entidade. Ao conjunto de todas as contas do banco chamamos de conjunto de entidades. Os conjuntos de entidades modelados no MER serão tabelas dos bancos de dados. Cada entidade será uma linha da tabela ou registro ou ainda tupla.

Conjuntos de entidades são representados por Retângulos no DER.



## ATRIBUTOS

As entidades são descritas no banco de dados por meio de seus **atributos**. Por exemplo, os atributos número conta e saldo descrevem uma conta bancária em particular. Os atributos serão as colunas das tabelas.

Os atributos são representados no DER por ovais:



## Exercícios

Identifique entidades com seus atributos para :

- a) controlar aniversários e e-mail dos amigos
- b) controlar uma coleção de livros
- c) controlar pacientes de um consultório odontológico
- d) controlar uma escola
- e) controlar uma loja
- f) controlar um consultório médico

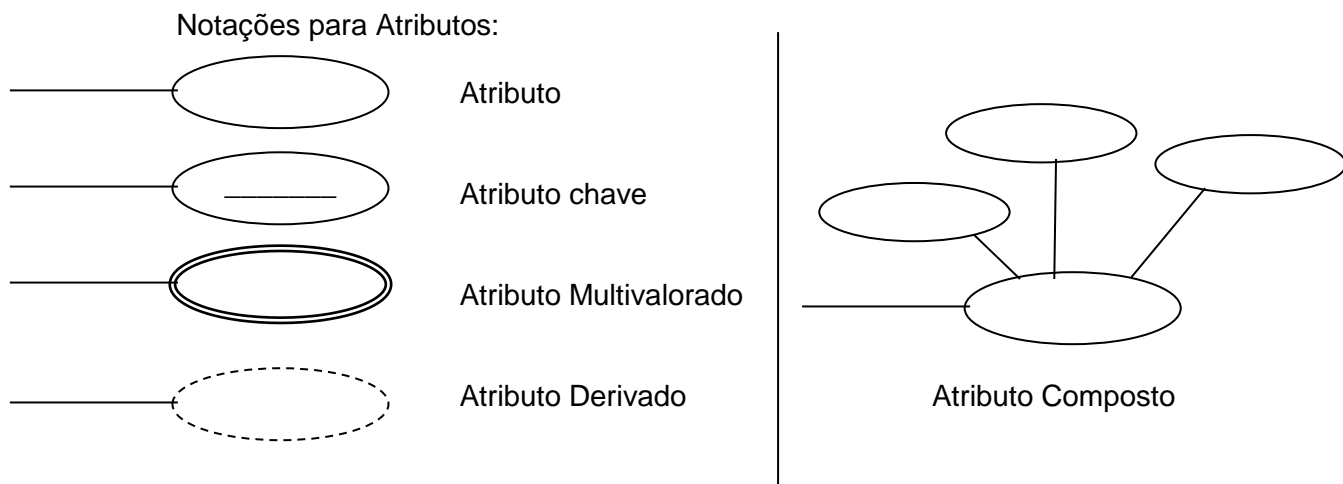
Um atributo, como é usado no MER, pode ser caracterizado pelos seguintes tipos:

**Atributos simples** (não podem ser divididos) ou **compostos** (podem ser divididos em partes exemplo, Endereço pode ser dividido em Nome da Rua, Cidade, Estado e CEP) Atributos compostos são úteis quando os usuários referenciam o atributo composto como uma unidade e, em outros momentos, referenciam especificamente a seus componentes. Se o atributo composto for sempre referenciado como um todo, não existe razão para subdividi-lo em componentes elementares.

**Atributos monovalorados ou multivalorados** Monovalorados são atributos que têm apenas um único valor (exemplo, Data de nascimento). Em outros casos, um atributo pode ter um conjunto de valores e são chamados de atributos multivalorados (exemplo, Telefone residencial ). Atributos multivalorados podem possuir uma multiplicidade, indicando as quantidades mínima e máxima de valores.

**Atributos derivados** Em alguns casos, dois ou mais atributos são relacionados. Por exemplo, Idade e Data de Nascimento de uma pessoa. Para uma entidade pessoa em particular, a Idade pode ser determinada a partir da data atual e da Data de Nascimento. Atributos como Idade são chamados atributos derivados.

**Atributos nulos** Uma entidade pode não ter quaisquer valores para um atributo. Por exemplo, o atributo Apartamento aplica-se somente à quem reside em algum prédio. Para tais situações, um valor especial chamado null é criado. O valor null pode também ser utilizado para denotar que o valor é desconhecido, como por exemplo, quando o cliente em um cadastro não responde o número do CEP da rua onde reside. O significado para o primeiro uso do null é “não aplicável” e, para o segundo, “desconhecido”.



## TIPOS DE ATRIBUTOS

Cada atributo deve conter um tipo de informação específico segundo os dados que ele receberá. Eles podem ser texto(até 255 caracteres), memorando (até 65.535 caracteres), número, data/hora, moeda, sim/não, etc. Veremos em detalhes os tipos de atributos segundo o Access na Parte 2 da apostila.

## CHAVES

Dentro de um conjunto de entidades cada entidade deve ser diferenciada de outra através de seus atributos. Os atributos utilizados para fazer essa distinção são as chaves. Uma chave nunca pode ser repetida, ou seja, o conjunto de valores dos atributos constituintes de uma chave deve ser único para cada registro dentro de uma determinada tabela.

## SUPERCHAVE

É um conjunto de um ou mais atributos que permite identificar univocamente uma entidade no conjunto de entidades.

Exemplo: Para o conjunto de entidades Cliente o atributo CPF é suficiente para diferenciar um cliente de outro, assim o CPF é uma superchave. A combinação dos atributos CPF e nome também é superchave pois também identifica um único cliente, já o atributo nome não é superchave pois não identifica um único cliente pois podem ter duas pessoas com mesmo nome.

Nosso interesse é por superchaves para as quais nenhum subconjunto possa ser superchave. Essas superchaves são chamadas chaves candidatas.

## CHAVE CANDIDATA

É uma superchave que não tenha nenhum subconjunto próprio que seja uma superchave.

Imagine que uma combinação nome, data nascimento e nome da mãe seja suficiente para identificar um único cliente. Então, (CPF) e (nome, data nascimento e nome da mãe) são chaves candidatas. Embora os atributos CPF e nome, juntos, possam identificar um único cliente, sua combinação não forma uma chave candidata, pois apenas CPF já é uma chave candidata.

O projetista do banco de dados escolhe uma chave candidata dentre tantas possíveis para a identificação de uma entidade em um conjunto de entidades. Essa chave candidata escolhida recebe o nome de chave primária.

## CHAVE PRIMÁRIA

É o conjunto de atributos que melhor identifica univocamente uma entidade dentro de um conjunto.

Ao escolher uma chave primária, você deve estar atento a alguns detalhes:

- A chave primária nunca deve ser repetida, portanto não escolha campos que não satisfaçam essa condição;
- O tamanho da chave primária afeta a velocidade das operações do seu banco de dados. Para um melhor desempenho, use o menor tamanho que acomode os valores necessários.
- O campo chave primária sempre vai estar indexado, ou seja, ordenado.

Quando apenas um atributo compõe a chave primária, ela é chamada de chave primária simples. Quando mais de um atributo compõe a chave primária, ela é chamada chave primária composta. Neste caso, o conjunto de atributos é que não deve ser repetido. E cada atributo pode ser repetido individualmente, dependendo do caso. Ex. O número de uma conta corrente normalmente é formado por dois atributos: número da agência e número da conta.

## CHAVE ÚNICA

Algumas vezes a chave primária é um atributo gerado pelo sistema como por exemplo o RA de um aluno. A escola identifica esse aluno pelo seu RA que normalmente é um número. O RA é a chave primária. Em toda a escola só existe um aluno com esse RA. Se por algum equívoco o aluno se cadastrasse duas vezes seriam gerados dois RAs e com certeza isso acarretaria problemas.

Uma chave única é um meio que utilizamos quando um determinado campo não deve ser repetido e não foi escolhido como chave primária. Com esse método, damos mais consistência ao banco de dados. Por exemplo, podemos optar por usar o RG como chave única, assim evitaremos que o aluno seja cadastrado duas vezes.



## Exercícios

Identifique as chaves para as entidades identificadas nos exercícios anteriores.

## CHAVE ESTRANGEIRA

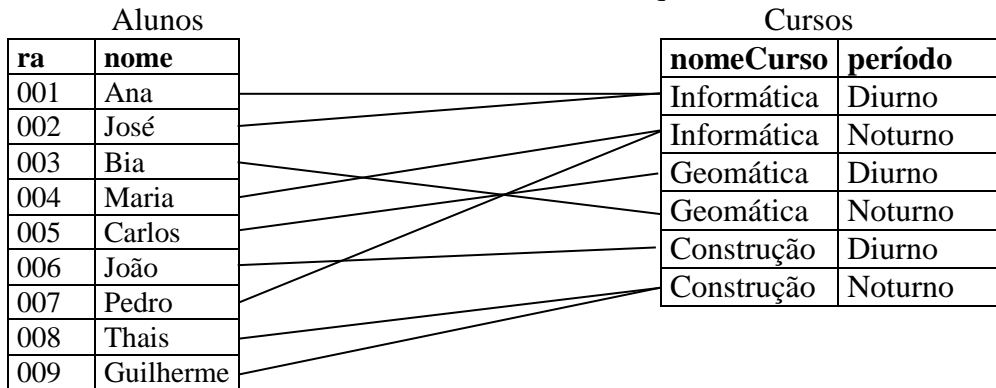
A chave estrangeira é utilizada quando queremos que o valor de um atributo seja validado a partir do valor de atributo de uma outra tabela. Nesse caso há um vínculo entre as tabelas.

Exemplo: Antes de efetuar o cadastro de um pedido de venda, devemos nos certificar de que o cliente em questão consta no cadastro de clientes. O campo código do cliente na tabela de clientes é uma chave primária. Esse mesmo valor na tabela de pedidos é uma chave estrangeira.

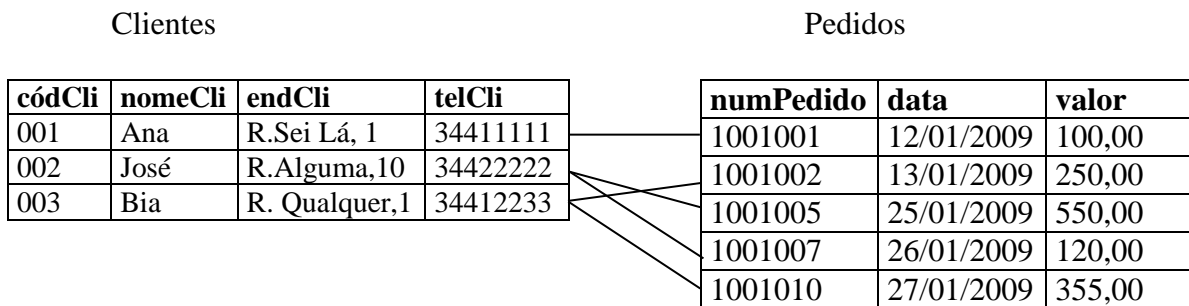
Para compreender melhor a chave estrangeira precisamos definir relacionamento

## RELACIONAMENTOS

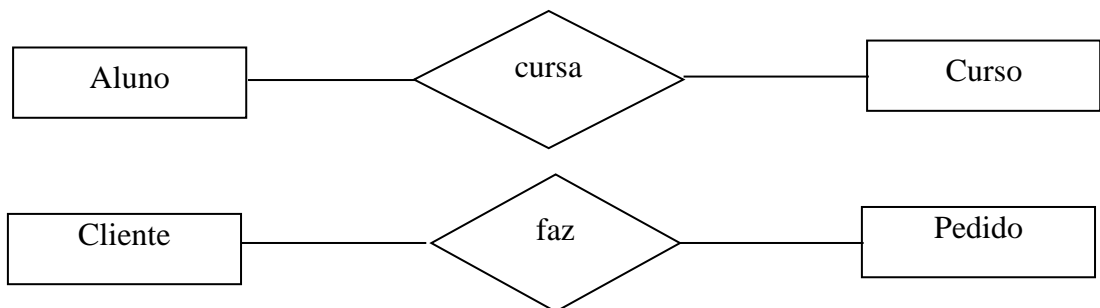
Um relacionamento é uma associação entre entidades. Por exemplo, na escola um relacionamento cursa associa um aluno ao curso que ele está matriculado.



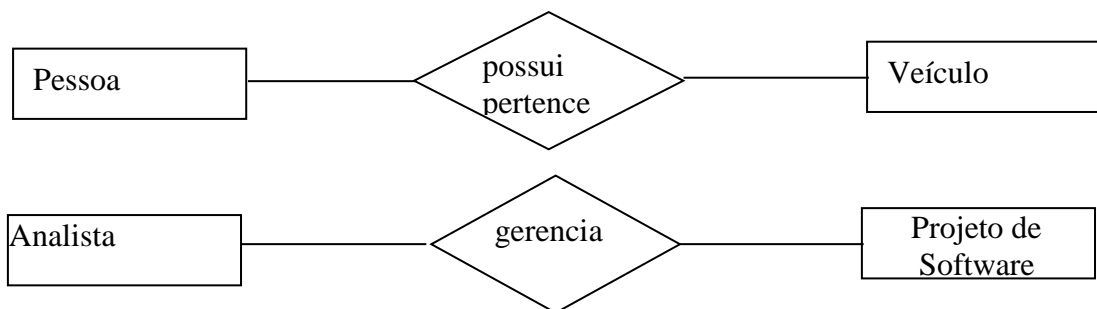
Outro Exemplo: Podemos associar um determinado cliente a um pedido.



Relacionamentos são representados por losângos no DER.

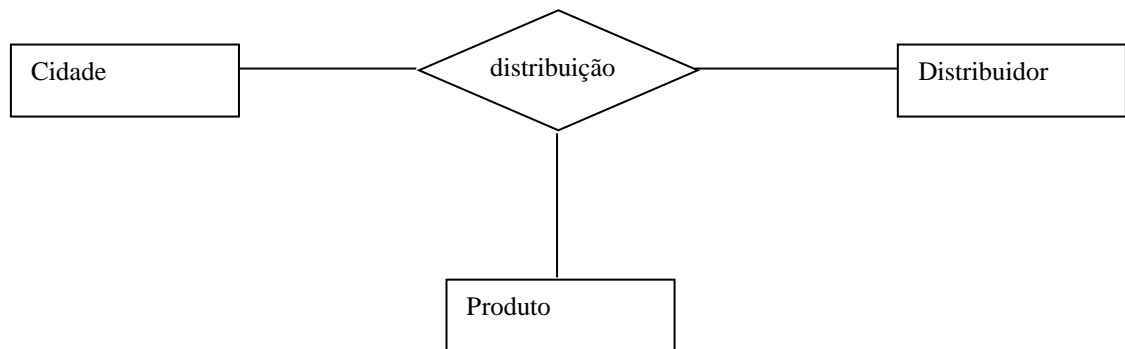


**Relacionamento binário** : Representa o relacionamento entre duas entidades.

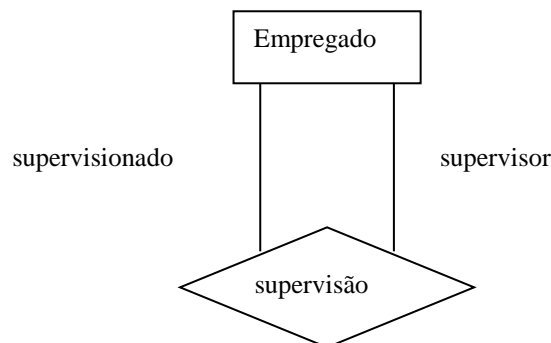




**Relacionamento Ternário:** Representa o relacionamento entre três entidades distintas.



**Auto-relacionamento:** Relacionamento entre ocorrência de uma mesma entidade. Neste caso, torna-se necessário especificar o papel da entidade no relacionamento.

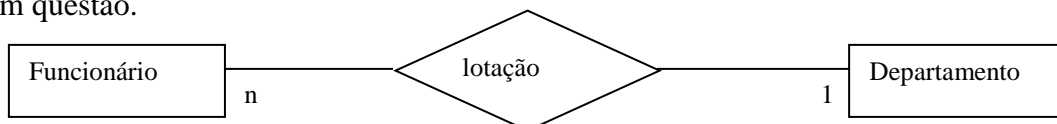


Além das entidades e dos relacionamentos, o MER representa certas regras, as quais o conteúdo do banco de dados precisa respeitar. Uma regra importante é o mapeamento das cardinalidades, as quais expressam o número de entidades às quais a outra entidade se relaciona por meio daquele conjunto de relacionamentos.

A cardinalidade se classifica em mínima e máxima.

### Cardinalidade máxima

Expressa o **número máximo de ocorrências** de outra entidade associadas a entidade em questão.



O modelo acima mostra que:

- A entidade Funcionário tem cardinalidade máxima 1 no relacionamento Lotação, porque uma ocorrência Funcionário pode estar associado no máximo a uma ocorrência de Departamento.
- A entidade Departamento tem cardinalidade máxima N no relacionamento Lotação, porque cada ocorrência de Departamento pode estar associada a várias ocorrências de Funcionário.

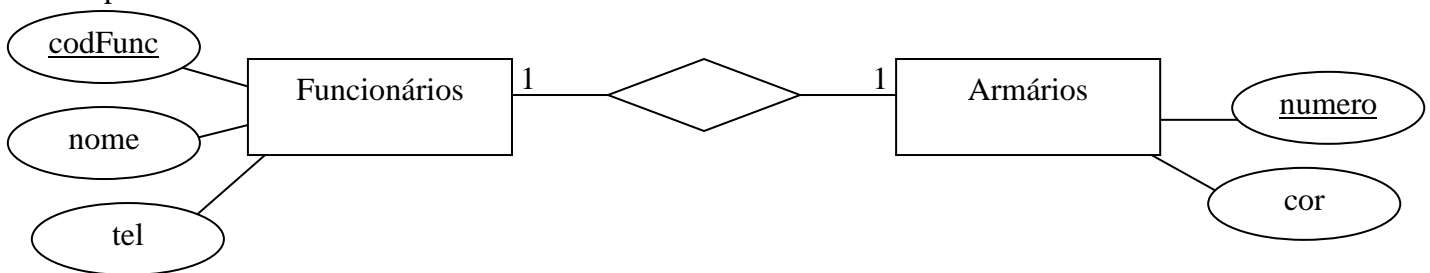
A **cardinalidade máxima** é utilizada para classificar os **relacionamentos binários** em:

- 1 : 1** - lê-se um-para-um
- 1 : n** - lê-se um-para-muitos
- n : n** - lê-se muitos-para-muitos

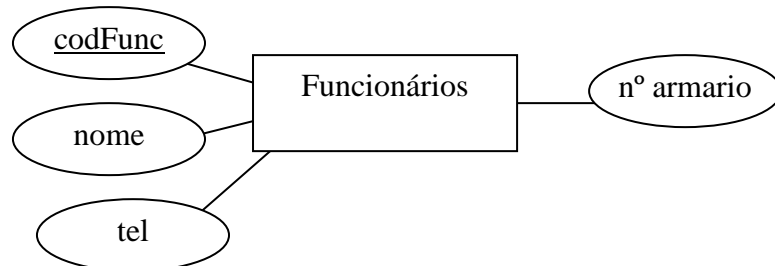
### RELACIONAMENTO UM-PARA-UM

Em um relacionamento um-para-um, cada registro na tabela X pode ter somente um registro coincidente na tabela Y, e cada registro na tabela Y pode ter somente um registro coincidente na tabela X.

Exemplo: Vamos criar uma entidade funcionários e outra armários. Um funcionário só pode ter um armário e vice-versa.



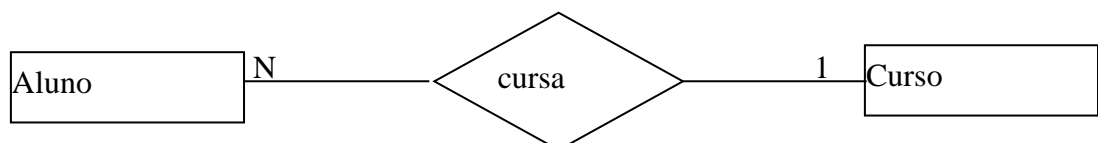
Esse tipo de relacionamento é incomum, pois a maioria das informações assim relacionadas estaria em uma só tabela. A utilização de um relacionamento um-para-um é recomendada quando você deseja dividir uma tabela com muitos campos, isolar parte de uma tabela por segurança ou armazenar informações que se apliquem somente a um subconjunto da tabela principal.



### RELACIONAMENTO UM-PARA-MUITOS

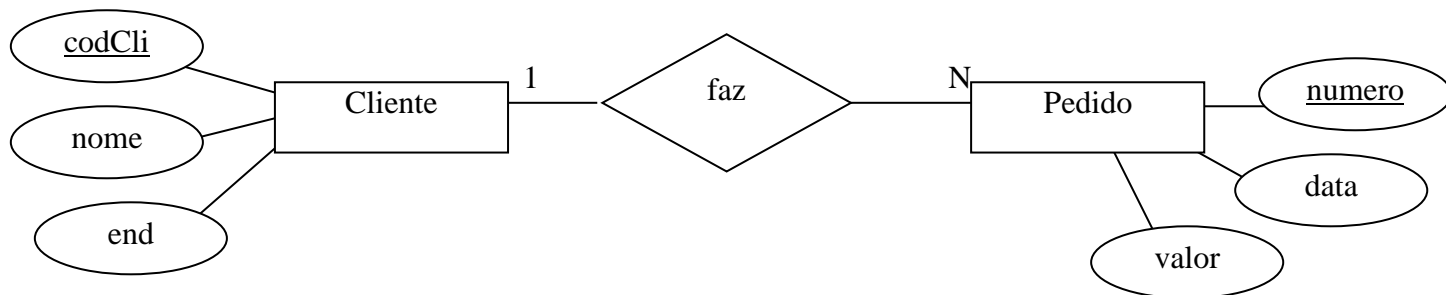
Um relacionamento um-para-muitos estabelece que um registro em uma tabela X pode ter vários registros associados em uma tabela Y. Este é o mais comum dos relacionamentos.

Exemplo: vamos relacionar um aluno a um curso



Neste exemplo um curso pode ter vários alunos, mas um aluno só pode cursar um curso.

Outro Exemplo: vamos relacionar um cliente a um determinado pedido.

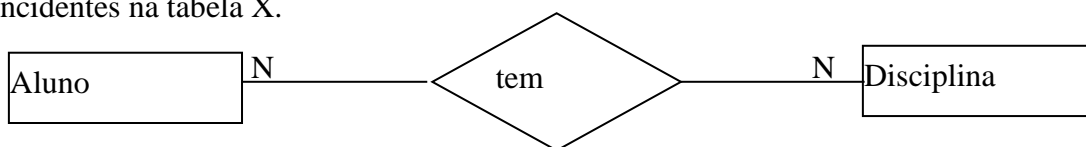


Neste exemplo um cliente pode efetuar vários pedidos, mas um pedido só pode ter um cliente.

Observe que um cliente que faz muitos pedidos não precisa ter seu endereço repetido em cada pedido, essa informação é obtida através do relacionamento.

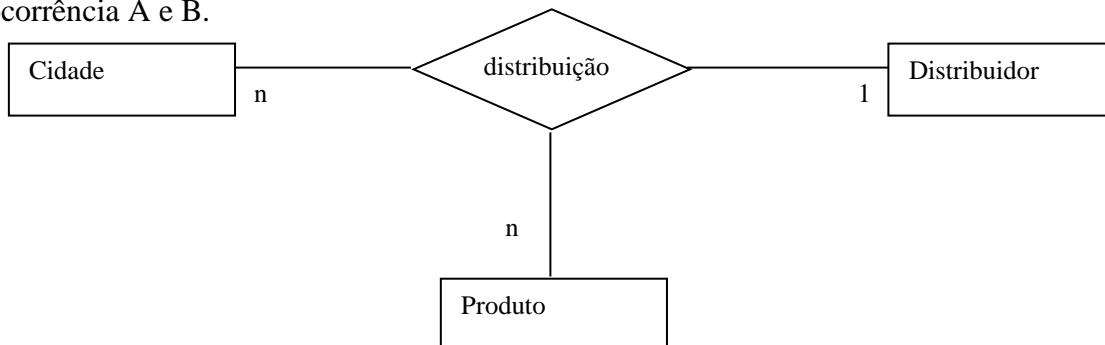
### RELACIONAMENTO MUITOS-PARA-MUITOS

Em um relacionamento muitos-para-muitos, um registro na tabela X pode ter vários registros coincidentes na tabela Y, e um registro na tabela Y pode ter vários registros coincidentes na tabela X.



Neste exemplo um aluno tem muitas disciplinas e uma disciplina tem muitos alunos.

No caso do **relacionamento ternário**, a **cardinalidade** refere-se a **pares de entidades**. Em um relacionamento R entre três entidades A, B, C a cardinalidade máxima de A e B dentro de R indica quantas ocorrências de C podem estar associadas a um par de ocorrência A e B.



O modelo acima expressa que:

- A um par (cidade, produto) está associado a no máximo um distribuidor.
- A um par (cidade, distribuidor) podem estar associados muitos produtos.
- A um par (produto, distribuidor) podem estar associadas muitas cidades.



### Exercícios:

Represente as seguintes situações:

1. Todo empregado de uma empresa é alocado a uma mesa e todas as mesas tem seu Empregado alocado a ela.
2. Atleta pode se inscrever em uma única modalidade esportiva, a qual pode ter muitos atletas inscritos.
3. Um departamento do colégio pode ser responsável por ministrar várias disciplinas e cada disciplina tem um único departamento responsável.

### Cardinalidade mínima

A cardinalidade mínima expressa o número mínimo de ocorrências de outra entidade associada a entidade em questão através do relacionamento.

São considerados duas cardinalidades mínimas:

**Cardinalidade mínima 1 ou associação obrigatória** - significa que o relacionamento entidade deve obrigatoriamente associar uma ocorrência de entidade a cada ocorrência da entidade em questão.

**Cardinalidade mínima 0 ou associação opcional** - significa que uma ocorrência da entidade em questão pode existir sem a necessidade de se associar a uma ocorrência de outra entidade.

A cardinalidade mínima é anotada no diagrama junto à cardinalidade máxima através dos pares **(0,x)** e **(1,x)** onde:

- **0** - representa a cardinalidade mínima opcional;
- **1** - a cardinalidade mínima obrigatória ;
- **x** - a cardinalidade máxima.

Observação: A **classe do relacionamento** é identificada pela **cardinalidade máxima** obtida no par.

### Relacionamento Parcial (0,x) e Total (1,x)

Quando a cardinalidade mínima é 0 (zero) o relacionamento é parcial em relação a entidade em questão.

Quando a cardinalidade mínima é 1 (um) o relacionamento é total em relação a entidade em questão.



Com base no que discutimos podemos fazer a seguinte leitura do modelo acima:

1 . Trata-se de um relacionamento binário cuja classe do relacionamento é 1:n (um-para-muitos)

2 . Cada departamento pode lotar **nenhum** ou vários funcionários. Logo, o relacionamento **Lotação é parcial em relação a entidade Departamento**, já que vai existir Departamento que não vai se associar a funcionário através de Lotação.

3. Cada funcionário deve estar lotado a **um** e somente um departamento. Neste caso, o relacionamento **Lotação é total em relação a entidade Funcionário**, pois para cada ocorrência de Funcionário vai existir um associação a um ocorrência de Departamento através do relacionamento (A restrição de participação total é algumas vezes chamada dependência existencial).

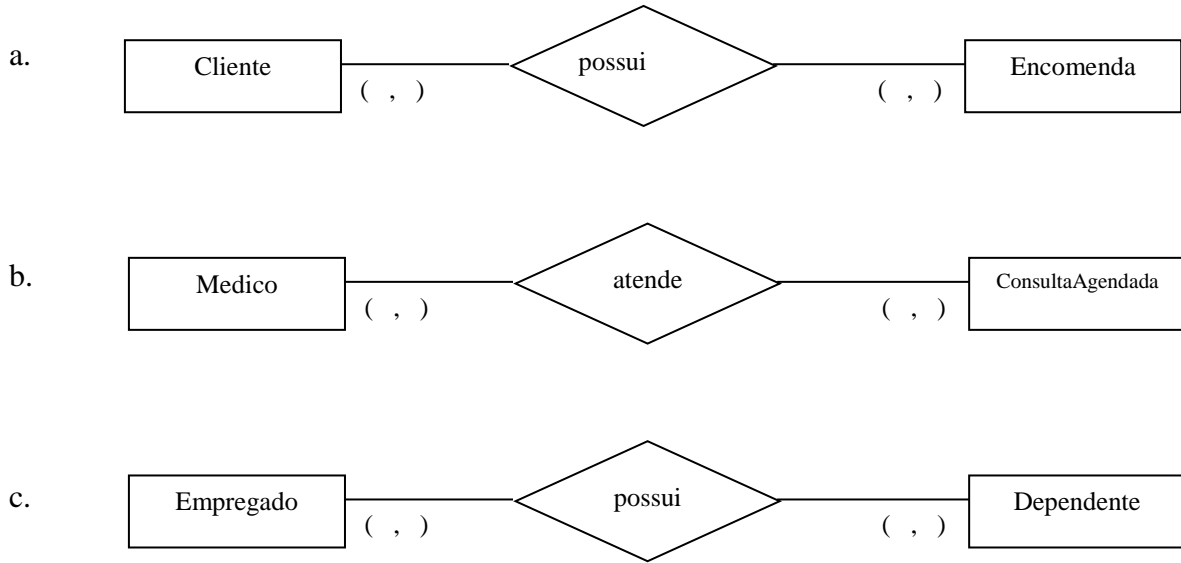
### Exercícios:



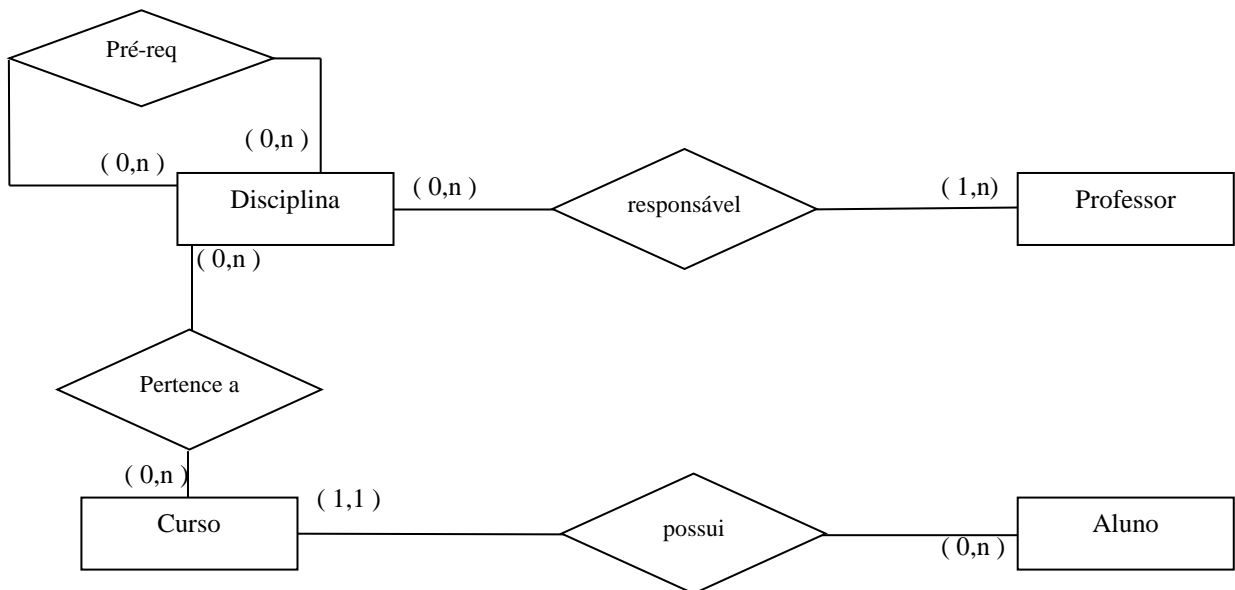
1. Identifique a cardinalidade dos relacionamentos exibindo os passos conforme o exemplo:



Um aluno pode não estar inscrito em nenhum curso(mínimo 0) ou estar inscrito em um curso (máximo 1). Um curso pode ter zero ou muitos alunos inscritos



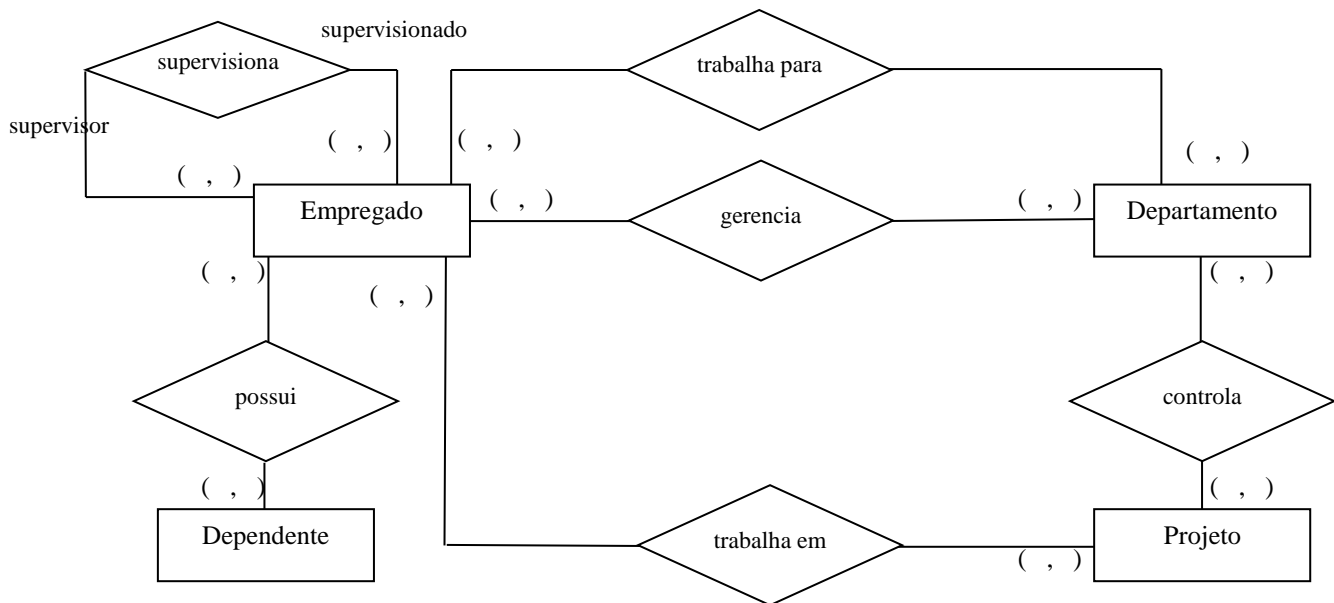
2. Observe o modelo e responda as questões a seguir:



- Identifique as entidades do modelo
- Identifique os relacionamentos do modelo
- Interprete cada um dos relacionamentos identificando o tipo de cardinalidade.

3. Analise o texto a seguir, que exhibe as regras de negócio e defina o grau dos relacionamentos do modelo:

Uma companhia é organizada em departamentos. Um departamento pode possuir um gerente e pode controlar inúmeros projetos. Empregado deve estar associado a um único departamento, mas pode trabalhar em diversos projetos, que não são necessariamente controlados pelo mesmo departamento. Todo empregado responde a um supervisor.

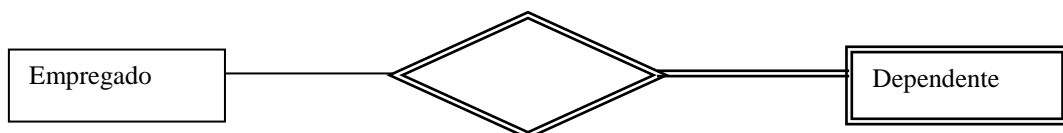


### TIPO DE ENTIDADE-FRACA

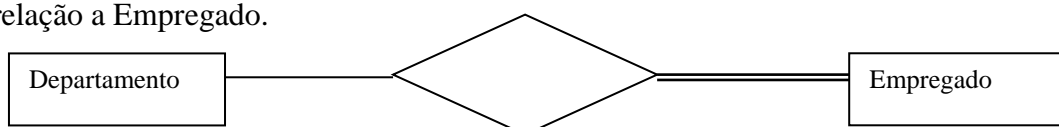
Alguns tipos de entidades podem não ter quaisquer atributos-chaves. Isto implica que não se pode distinguir as entidades porque a combinação dos valores de atributos podem ser idênticas. Tais tipos de entidades são chamadas tipos de entidades-fracas. Entidades que pertencem a um tipo de entidade-fraca são identificadas por estarem associadas a entidades específicas de um outro tipo de entidade em combinação com alguns de seus valores de atributos. Este outro tipo de entidade é denominado proprietário da identificação, e o tipo de relacionamento que relaciona um tipo de entidade-fraca com o proprietário da identificação é chamado relacionamento de identificação do tipo de entidade-fraca. Um tipo de entidade fraca sempre tem uma restrição de participação total (dependência existencial) com respeito ao seu relacionamento de identificação, porque não é possível identificar uma entidade-fraca sem a correspondente entidade proprietária.

Por exemplo, considere o tipo de entidade **DEPENDENTE**, relacionado a **EMPREGADO**, que é usado para representar os dependentes de cada empregado através do relacionamento 1:N.

Os atributos de **DEPENDENTE** são Nome (apenas o primeiro nome do dependente), DataNasc, Sexo e Relação com o empregado (esposa, marido, filho, sogra, etc.). Dois dependentes de empregados distintos podem ter os mesmos valores para os atributos, mesmo assim eles ainda serão entidades distintas. Os dependentes serão identificados como entidades distintas após a determinação da entidade empregado com a qual cada um está relacionado.

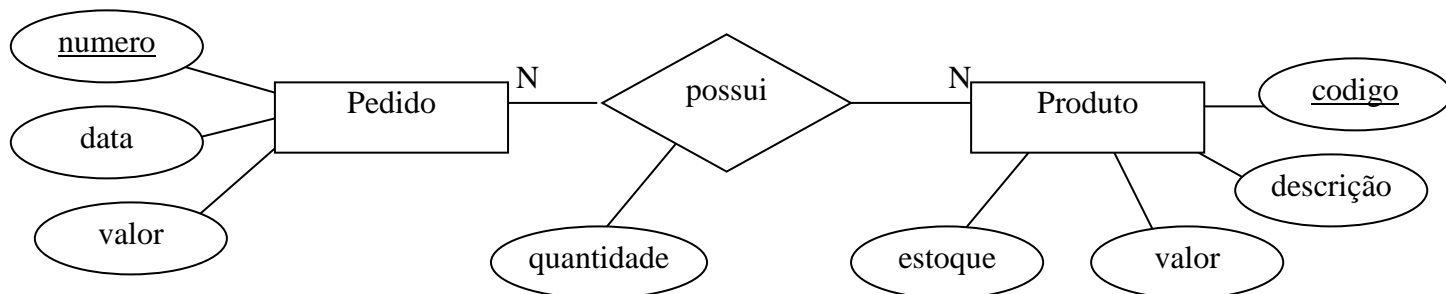


**Dependência de existência** ocorre quando a existência de uma determinada entidade está condicionada à existência de uma outra entidade a ela relacionada mas ambas possuem atributo chave. Exemplo. O relacionamento a seguir tem uma restrição de participação total em relação a **Empregado**.



## ATRIBUTOS DE RELACIONAMENTO

Relacionamentos podem ter atributos, e nesse caso se liga o atributo ao losango do relacionamento.



### Exercícios:

Crie os DER para cada realidade descrita a seguir:

1. Um consultório dentário deseja cadastrar os pacientes, os dentistas e controlar o agendamento de consultas.
2. O departamento de informática deseja cadastrar os alunos do curso de informática para controlar o acesso dos mesmos aos laboratórios. Haverá uma catraca na porta de cada laboratório. Ao entrar o aluno passa o RA e o sistema libera o acesso registrando a entrada daquele aluno, naquele laboratório, naquela data e horário.
3. Para controlar a presença dos alunos nas palestras da Semana de Informática deve haver um cadastro de alunos inscritos e de palestras agendadas. É importante saber o RA, nome e turma do aluno. Sobre a palestra interessa saber data, hora de início, duração prevista, e título da palestra, além do nome do palestrante, seu endereço, telefone e formação acadêmica.
4. Controlar jogos definindo os 2 times envolvidos, a data e a hora do jogo. Antes de controlar o horário dos jogos devem ser cadastrados todos os times com código do time, cor da camiseta, e nome do capitão.
5. Uma escola de informática deseja manter o cadastro de seus cursos, sendo importante cadastrar além do nome do curso a carga horária do mesmo. Sabe-se que a escola abre turmas diferentes do mesmo curso de acordo com a procura dos alunos. Cada turma tem aulas em determinado horário e sala e tem um único professor. A escola mantém um cadastro de seus professores. Ao se matricular, o aluno escolhe a turma que frequentará de acordo com sua disponibilidade de horário.
6. Uma loja de artesanato trabalha sobre encomendas. Sempre que um cliente faz uma encomenda se mantém armazenados: data da encomenda, valor total e data prevista de entrega. Uma encomenda pode solicitar muitos produtos, cada um deles com sua quantidade especificada. Todos os vendedores da loja são cadastrados com seus dados pessoais e como esses vendedores trabalham com comissão, é importante controlar qual vendedor obteve cada encomenda. O cliente deve ser cadastrado também com seu telefone e e-mail para ser informado quando a encomenda estiver pronta para retirada. Os produtos são cadastrados com código, descrição e valor unitário.

7. Existem os Municípios, e onde cada Município pertence a um único Estado do País. Do Estado me interessa a sigla e o nome por extenso. Do Município me interessa o nome, a área, a população e qual o partido que governa a municipalidade (prefeitura), mas há um problema que existe casos onde há um mesmo nome de município em estados diferentes. Também me interessam os partidos políticos, onde me interessa cadastrar o seu número no TSE, sua sigla e seu nome. Perceba então que além de se relacionar com o estado a que pertence, o município também possui uma relação com o partido do seu prefeito.
8. Um Banco deseja controlar o movimento das suas contas correntes. O banco possui agências espalhadas por várias cidades; o cadastro das agências contém informações sobre número da agência, nome e endereço onde a agência é localizada. No cadastro de clientes estão armazenados os dados de nome, CPF e endereço de cada cliente. Para identificar uma conta corrente é necessário o número do banco, o número da agência e o número da conta. É necessário manter o saldo atualizado das mesmas.
9. Uma agenciadora de mão-de-obra pretende informatizar o cadastro de candidatos a empregos temporários, objetivando construir um banco de dados onde possa manter os dados cadastrais dos candidatos e seus contratos temporários com as empresas que necessitam fazer contratações. O candidato é identificado por um número de controle e, além dessa informação, ficam registrados o seu nome, endereço, nascimento e profissão. Os contratos de mão-de-obra temporária são feitos individualmente, ou seja, um contrato para cada candidato contratado pela empresa. Cada contrato é identificado por um número único e nele são registrados a empresa contratante, a vigência do contrato (data de início e de término) e o valor pago por hora trabalhada. As empresas são identificadas por um CNPJ e possuem um nome e endereço.
10. A empresa é organizada em departamentos. Cada departamento tem um nome, um número e um funcionário que gerencia o departamento. Armazena-se a data de início que o funcionário começou a gerenciar o departamento. Um departamento pode ter diversas localizações. Um departamento controla inúmeros projetos, sendo que cada um tem um nome, um número e uma localização. Do funcionário armazena-se o nome, o RG, endereço, salário, sexo e data de nascimento. Todo funcionário está associado a um departamento, mas pode trabalhar em diversos projetos, que não são necessariamente controlados pelo mesmo departamento. Armazena-se também o número de horas que o funcionário trabalha em cada projeto. Mantém-se, ainda, a indicação do supervisor direto de cada funcionário. Os dependentes de cada funcionário são armazenados para o propósito de garantir os benefícios do seguro. Para cada dependente será armazenado o nome, sexo, data de nascimento e o parentesco com o funcionário.
11. A secretaria de uma determinada universidade precisa gerenciar informações sobre suas disciplinas, estudantes e professores. Um aluno pode estar cursando uma ou mais disciplinas, que por sua vez é ministrada por um professor. Para cada disciplina guarda-se o seu nome, as salas onde podem ser ministradas e as disciplinas que lhe são pré-requisitos. Para cada alunos temos o nome, o RA e o curso que está fazendo. Para cada professor, é de interesse da secretaria o seu nome, telefone e o departamento onde está alocado.



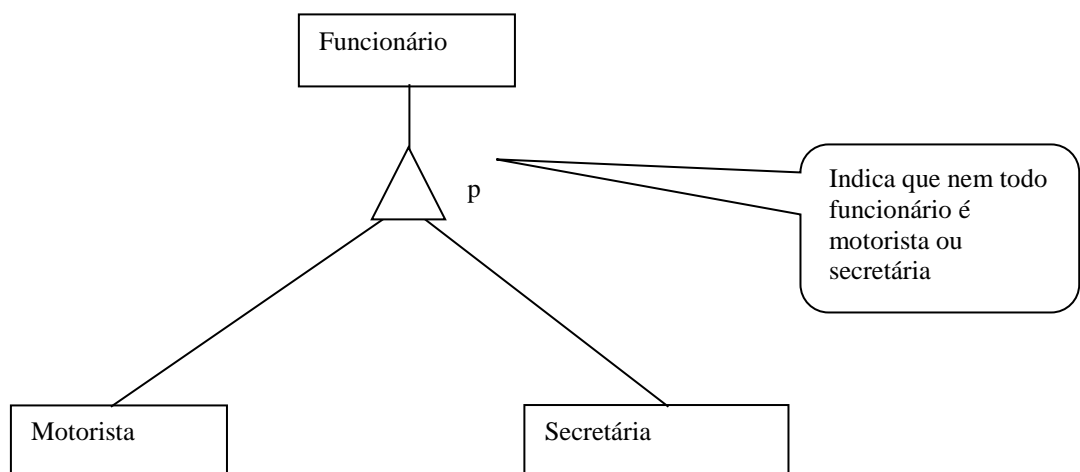
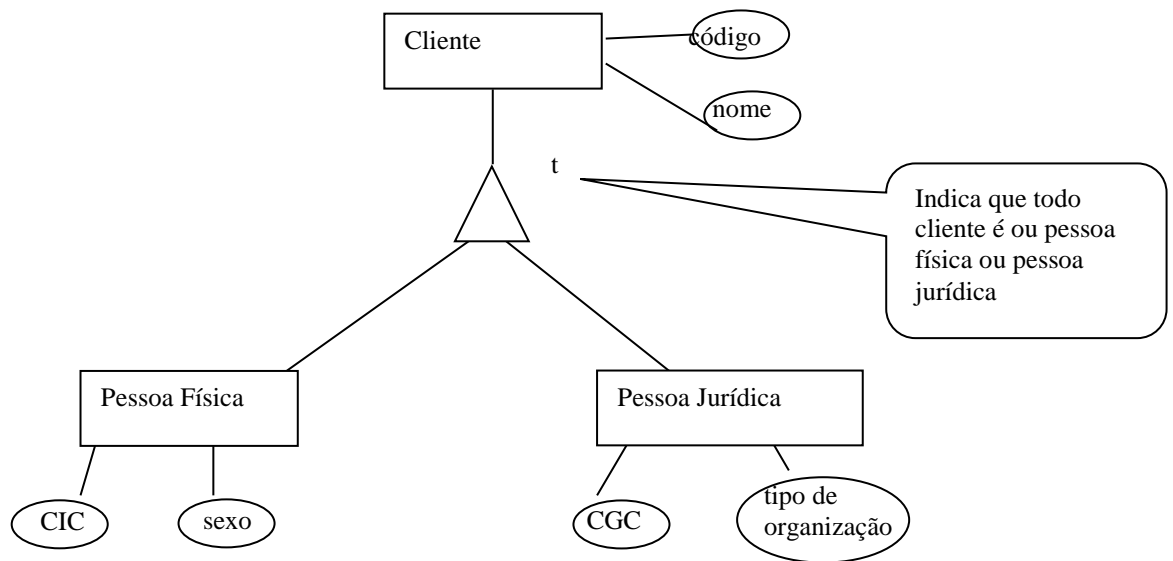
## GENERALIZAÇÃO/ESPECIALIZAÇÃO

Existem muitos casos práticos em que um conjunto de entidades representa elementos do mundo real que se subdividem em entidades especializadas com atributos parcialmente distintos. A entidade genérica possui os atributos comuns às entidades especializadas. Cada entidade especializada possui, além de suas próprias propriedades (atributos, relacionamentos e generalizações/especializações), também as propriedades da ocorrência da entidade genérica correspondente.

A generalização/especialização pode ser classificada em dois tipos: **Total** ou **Parcial**

Uma **generalização/especialização é total** se para cada ocorrência da entidade genérica existe sempre uma ocorrência em uma das entidades especializadas.

Uma **generalização/especialização é parcial** quando há ocorrências na entidade genérica sem correspondência nas entidades especializadas.

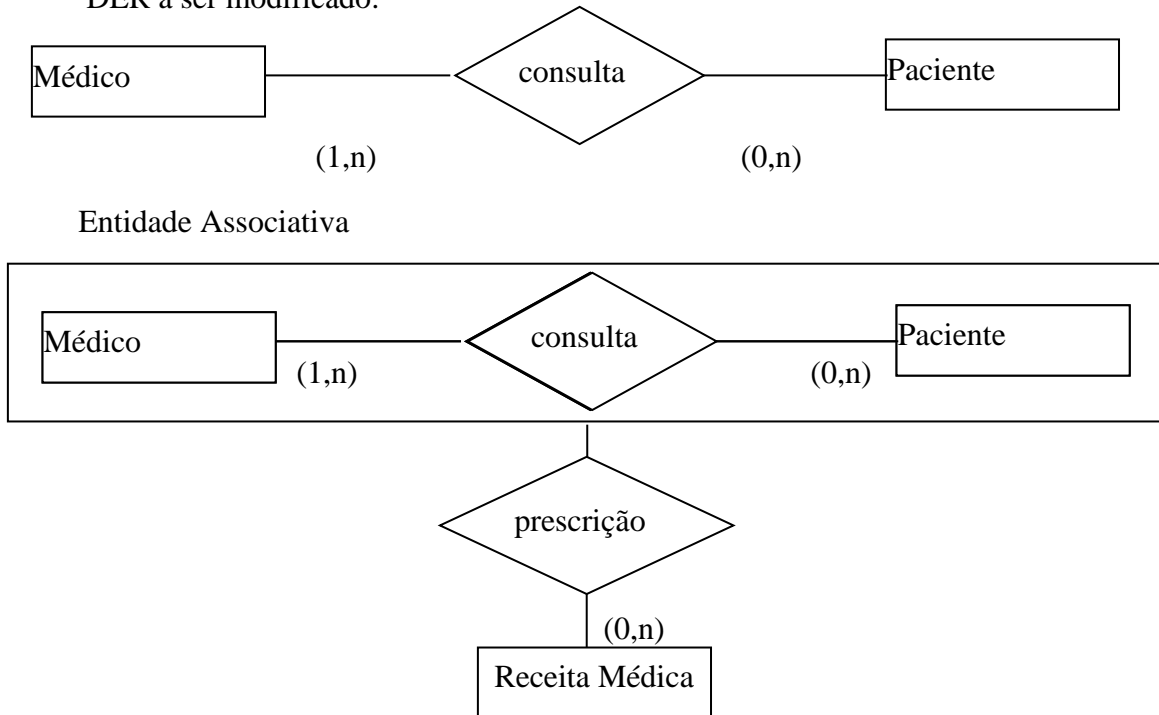


## ENTIDADE ASSOCIATIVA

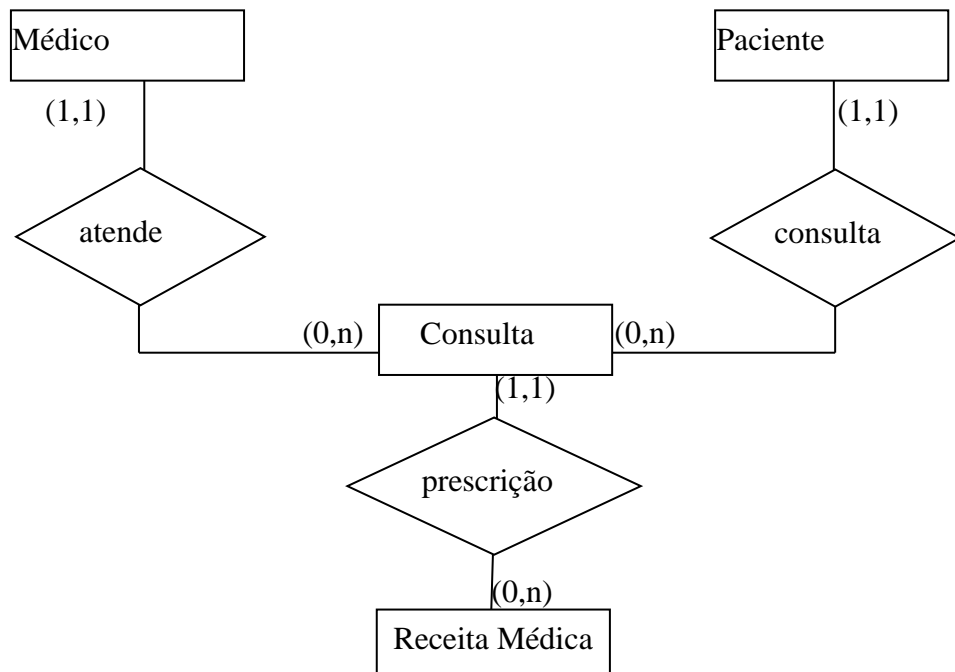
Um relacionamento é uma associação entre entidades. Na modelagem ER não foi prevista a possibilidade de associar uma entidade com um relacionamento ou então de associar dois relacionamentos entre si. Para resolver este problema é feita uma redefinição de um relacionamento, que passa a ser tratado como se fosse também uma entidade, permitindo associá-lo com outras entidades.

Por exemplo, suponha que no modelo abaixo, seja necessário saber que medicamentos foram prescritos em cada consulta. Como representar isso?

DER a ser modificado:



No exemplo anterior, caso não se desejasse usar o conceito de entidade associativa, seria necessário transformar o relacionamento Consulta em uma entidade, que então poderia ser relacionada a Medicamento, conforme mostra o modelo a seguir:



**Exercícios:**

Crie os DER para cada realidade descrita a seguir:

1. Os empregados de uma empresa pertencem a um dos três cargos: engenheiro, digitador ou motorista. Precisa-se cadastrar os seguintes dados: para os engenheiros (CREA, nome, matrícula e endereço), para os digitadores (nome, matrícula, endereço e idioma) e motorista (nome, matrícula, endereço e número da carteira de motorista)
2. Uma empresa tem dois tipos de empregados mensalistas e diaristas. Para gerenciar cada empregado são necessários sua matrícula, seu nome, para os mensalistas precisa-se seu salário mensal e para os diaristas o valor da diária e a quantidade de horas trabalhadas no mês. Apenas os diaristas pertencem a um sindicato. Cada empregado está lotado a um departamento.
3. Um funcionário pode ser alocado a vários projetos a partir de certa data. Em um projeto podem estar alocados vários funcionários. Um funcionário ao ser alocado a um projeto pode usar várias máquinas a partir de uma data e durante um certo tempo, assim como uma máquina pode ser utilizada em vários projetos por vários funcionários.

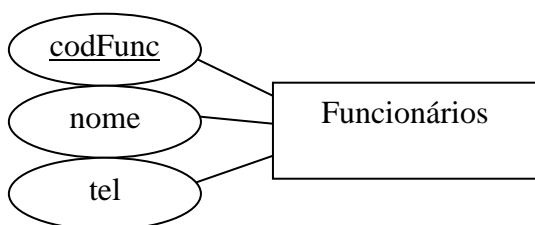
**REDUÇÃO DE UM ESQUEMA E-R A TABELAS**

Nos projetos de banco de dados iniciamos com uma modelagem de alto nível gerando os esquemas de visões através de um DER. O passo seguinte é o mapeamento para um modelo de dados de implementação. Trabalharemos nessa disciplina com o Modelo Relacional.

Tanto o modelo E-R quanto o modelo relacional empregam princípios de projetos similares, então podemos converter o projeto E-R em projeto relacional representando um esquema E-R por tabelas. Para essa conversão seguiremos alguns passos:

Passo1:

Cada **conjunto de entidade forte** será uma tabela que terá uma coluna para cada atributo do conjunto de entidades. Exemplo:



Funcionários

codFunc	nome	tel

**Funcionários (codFunc, nome, tel)**

OBS: A notação a ser seguida é:

cada relação ou tabela - nome em letra maiúscula

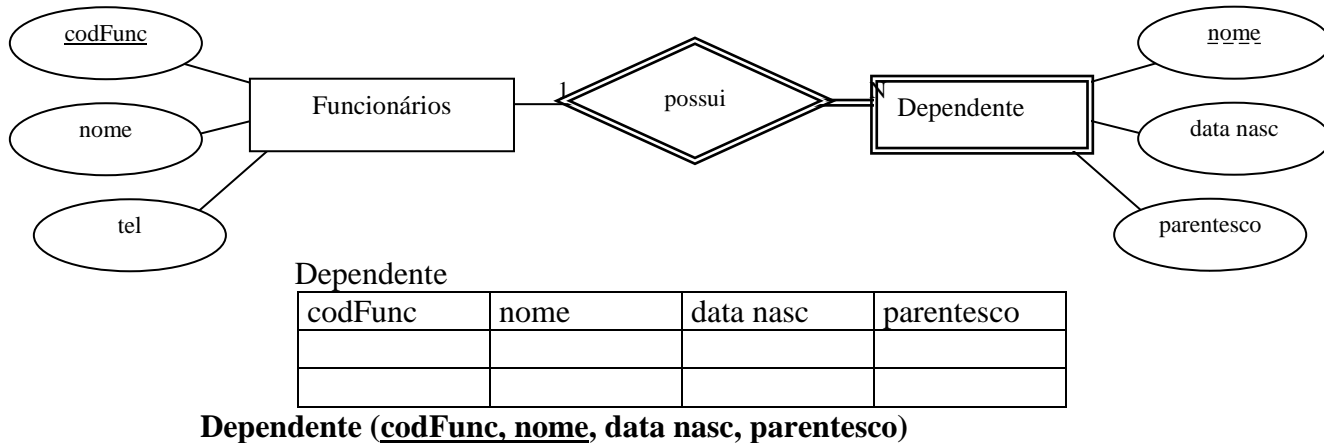
atributos ou colunas - entre parêntesis e em letras minúsculas

chave primária – sublinhada

chave estrangeira – sublinhada tracejada

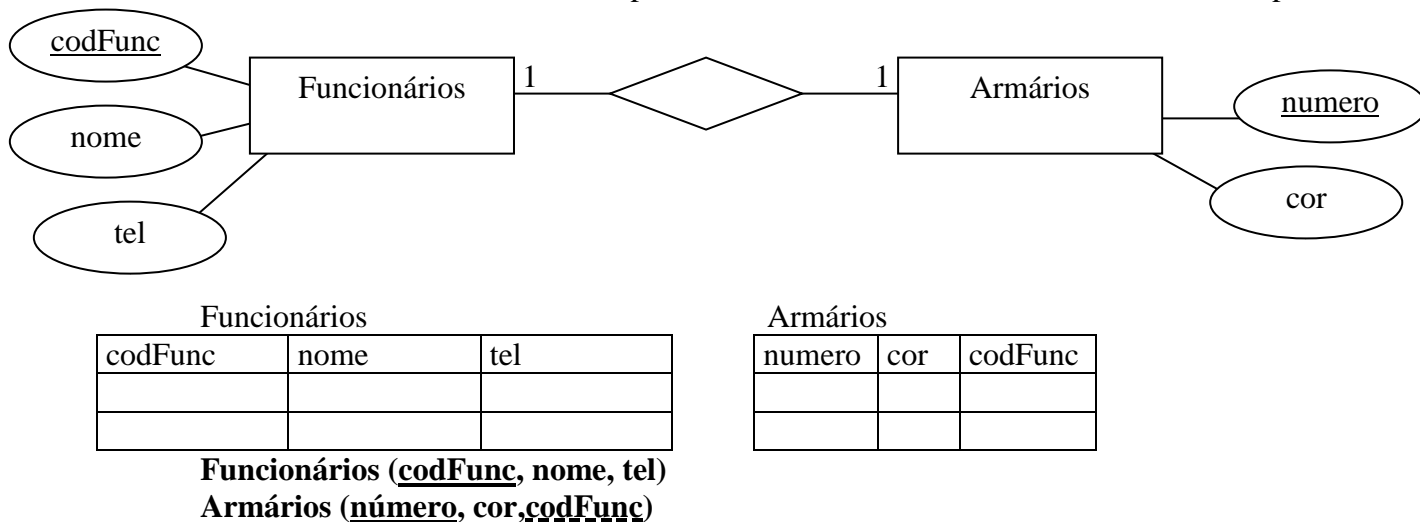
Passo2:

Cada **conjunto de entidade fraca** será uma tabela que terá uma coluna para cada atributo do conjunto de entidade fraca mais o atributo chave do conjunto de entidades forte do qual é dependente. Exemplo:



Passo 3:

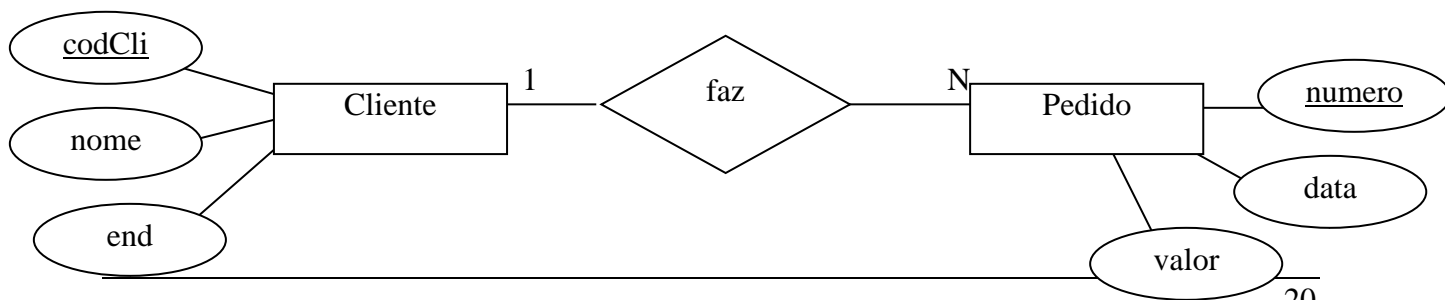
Para **conjunto de relacionamento binário 1:1** escolher uma das tabelas envolvidas no relacionamento e incluir nela a chave primária da outra tabela do relacionamento. Exemplo



Na tabela Armários dizemos que número é chave primária e codfunc é chave estrangeira.

Passo 4

Para **conjunto de relacionamento binário 1:N (não fraca)** identificar a tabela do lado N do relacionamento e incluir nela o atributo chave primária da outra tabela do relacionamento ( a do lado 1 do relacionamento) Exemplo:



Ciente

codCli	nome	end

Pedido

numero	data	valor	codCli

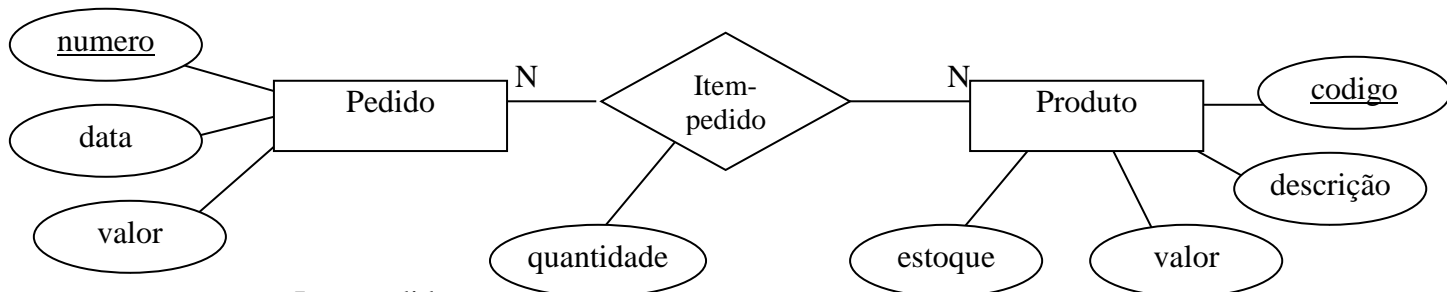
**Ciente** (codCli, nome, end)

**Pedido** (número, data, valor, codCli)

Na tabela Pedido dizemos que número é chave primária e codCli é chave estrangeira.

#### Passo 5

Para **conjunto de relacionamento binário N:N** criar uma tabela que terá uma coluna para cada atributo chave do conjunto de entidades envolvidos no relacionamento e mais uma coluna para cada atributo do relacionamento. Exemplo:



Item-pedido

numeroPedido	codigoProduto	quantidade

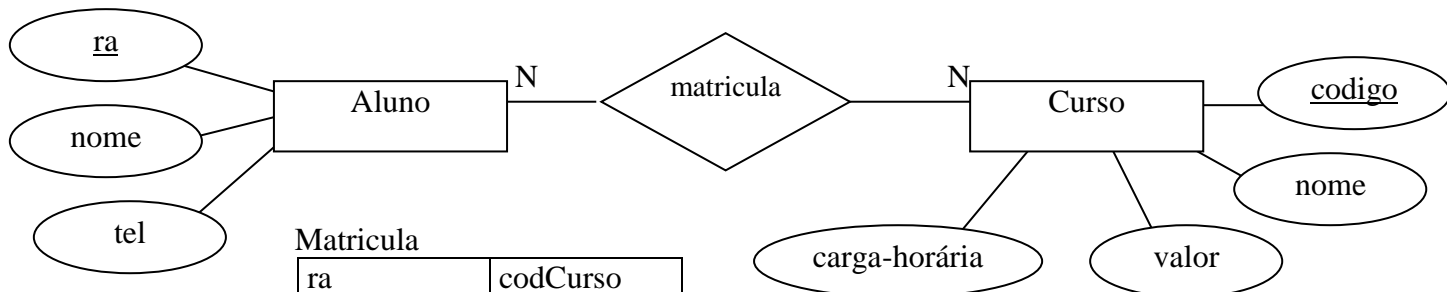
**Pedido** (número, data, valor)

**Produto** (código, descrição, valor, estoque)

**ItemPedido** (numPed, codProd, quantidade)

Na tabela Item-Pedido numPed e codProd compõem a chave primária e são estrangeiras. Observe que esses atributos tem nomes diferentes dos que tinham na tabela original

Caso o conjunto de relacionamento não tenha atributos de relacionamento a tabela será composta apenas pelos atributos chaves das entidades envolvidas nos relacionamentos.



Matricula

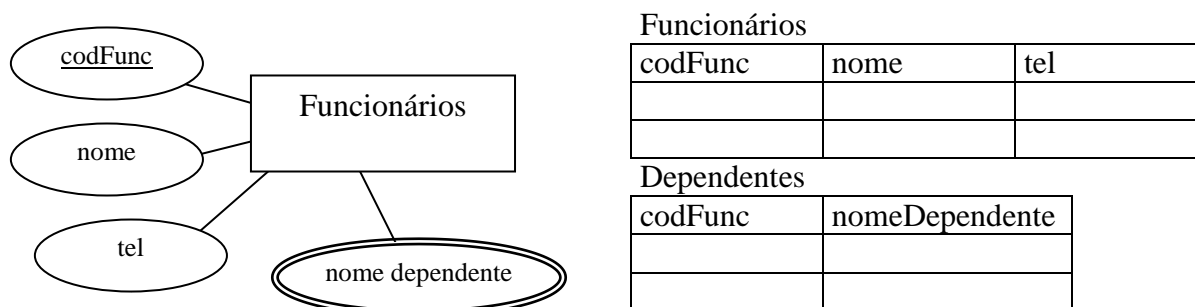
ra	codCurso

**Matricula** (ra, codCurso)

-----

## Passo 6

Para cada atributo multivalorado, como esse atributo não poderá ser uma coluna da tabela, devemos criar uma nova tabela com esse atributo e o atributo chave do conjunto de entidades. Exemplo:

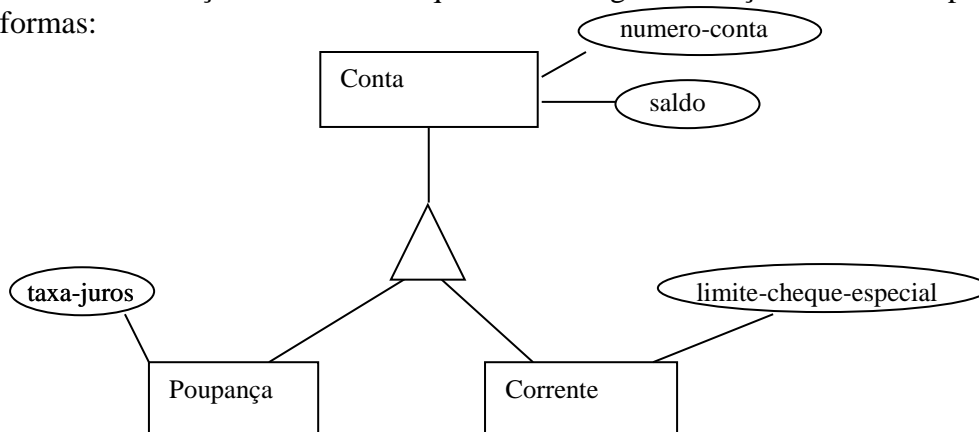


**Funcionários** (codFunc, nome, tel)  
**Dependente** (codFunc, nomeDependente)

Na tabela Dependentes, CodFunc e nome dependente compõem a chave primária.

## Passo 7

A transformação de um DER que contenha generalização em tabelas pode se dar de duas formas:



1. Criar a tabela para o conjunto de entidades de nível superior. Para cada conjunto de entidades de nível inferior, criar uma tabela que inclua uma coluna para cada um dos atributos daquele conjunto de entidades mais uma coluna para cada atributo chave primária do conjunto de entidades de nível superior. Assim teremos três tabelas:

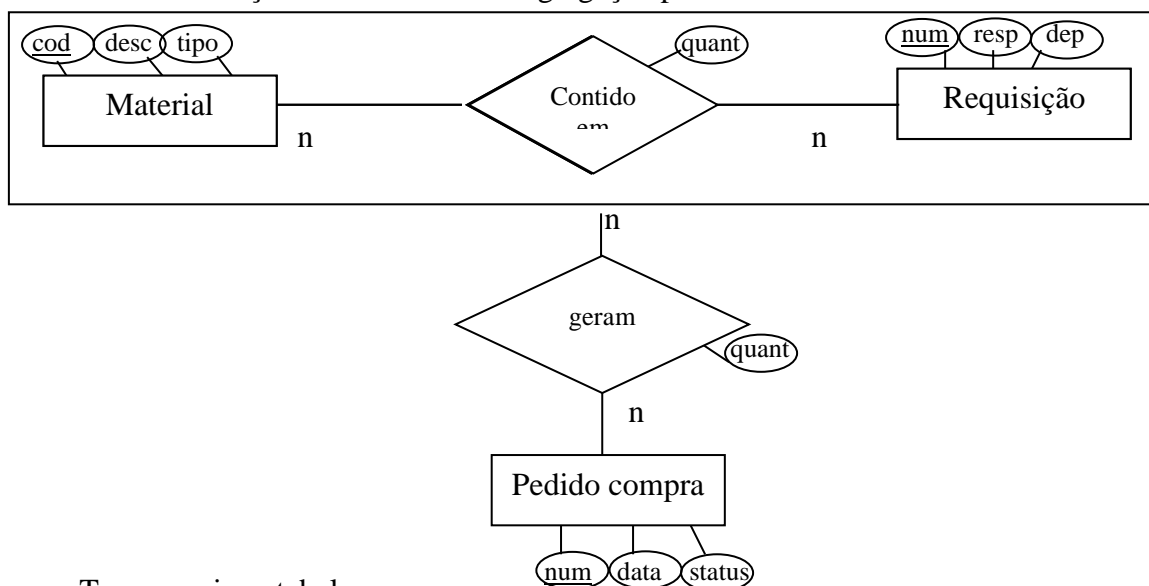
**Conta** (numero-conta, saldo)  
**Poupança** (numero-conta, taxa-juros)  
**Corrente** (numero-conta, limite-cheque especial)

2. Se a generalização é mutuamente exclusiva e total – isto é, se nenhuma entidade é membro de mais de um conjunto de entidades de nível inferior e se todas as entidades do conjunto de entidades de nível superior são membros também de um conjunto de entidades de nível inferior-, então uma outra alternativa é possível. Para cada conjunto de entidades de nível inferior é criada uma tabela contendo uma coluna para cada um de seus atributos mais uma coluna para cada atributo do conjunto de entidades de nível superior. Assim teremos duas tabelas:

**Poupança** (numero-conta, saldo, taxa-juros)  
**Corrente** (numero-conta, saldo, limite-cheque especial)

Passo 8

A transformação de um DER com agregação para tabelas é direta. Observe:



Teremos cinco tabelas:

O conjunto de entidades Material gera a tabela

**Material (cód, desc, tipo)**

O conjunto de entidades Requisição gera a tabela

**Requisição (num, resp, dep)**

O conjunto de relacionamentos Contido em gera a tabela

**ItemRequisição (NumReq, códMat, quant)**

O conjunto de entidades Pedido compra gera a tabela

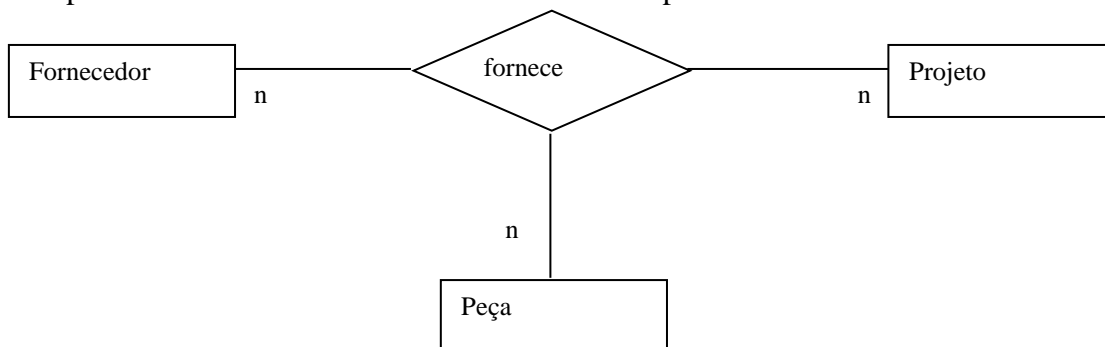
**Pedido (num, data, status)**

O conjunto de relacionamentos Geram em gera a tabela

**ItemPedido (Num Ped, NumReq, códMat, quant)**

Passo 9

Para cada tipo de relacionamento n-ário, criar uma nova tabela para representar o relacionamento. Incluir como chave estrangeira dessa tabela as chaves primárias das tabelas participantes e os atributos do relacionamento. Exemplo:



Fornecedor	
codForn	nome

Projeto	
códProj	dataIni

Peça	
codPeça	desc

Fornece			
codForn	codProj	codPeça	quant



### **Exercícios:**

Crie os mapeamentos dos MER dos exercícios 1 a 12 das páginas 15 a 17 para o Modelo Relacional (MR).

## **INTEGRIDADE DE DADOS**

### **INTEGRIDADE DE ENTIDADE**

Impõe que os valores dos atributos que correspondem à chave primária de uma entidade não podem ser nulos nem iguais a outros já existentes na tabela. A introdução de um valor nulo numa chave primária ou a inclusão nessa chave de um valor igual a outro já existente, faria com que a chave deixasse de identificar de modo unívoco os registros da tabela, portanto, deixaria de poder funcionar como chave.

### **INTEGRIDADE DE DOMÍNIO**

Esse tipo de integridade zela pelos valores ideais e necessários para um atributo. Para isso, definimos algumas regras de validação por meio de expressões compostas de valores constantes. Exemplo: Não permitir estoque negativo, impedir data de nascimento maior que a atual, não permitir preço menor que zero.

### **INTEGRIDADE REFERENCIAL**

A regra da integridade referencial zela pela consistência dos registros de uma entidade a partir de valores provenientes de outras entidades (chaves estrangeiras) Ou seja, determinado registro vai depender diretamente de um registro em outra tabela.

Exemplo: Se um pedido deve possuir o cliente que o solicitou, Pedido tem como chave estrangeira o código do cliente, e portanto para efetuar um pedido o cliente deve estar cadastrado na tabela clientes. Não podemos inserir um código de cliente em pedido se este não existir em Clientes.

Para impor integridade referencial a um relacionamento é necessário que os dois campos relacionados tenham o mesmo tipo de dados e que pelo menos um dos campos relacionados seja uma chave primária.

Quando impomos a integridade referencial devemos definir regras para atualização dos campos relacionados como propagar atualização que faz com que todas as alterações feitas nos registros da tabela primária sejam automaticamente atualizados para a tabela relacionada. Por exemplo, se mudarmos o código de um cliente que já efetuou pedidos, será modificado automaticamente o código do cliente para o novo código em todos os pedidos que ele já tiver efetuado.

Outro aspecto importante é o tratamento da exclusão de elementos de uma relação. Por exemplo, o que deve acontecer com os pedidos já existentes se o cliente for eliminado da relação de clientes? Uma alternativa é proibir a eliminação do elemento de Clientes enquanto ele estiver em uso como valor de uma chave estrangeira. Uma outra alternativa é eliminar de forma automática todos os pedidos daquele cliente (este comportamento é chamado de exclusão em cascata).

### **Exercícios:**



1. Para um sistema desenvolvido para uma clínica de vacinação, onde se tem um cadastro de vacinas das quais se controla o estoque, um cadastro de clientes e um controle de vacinações (vacinas tomadas pelos pacientes),



além de um controle de datas de reforço dessas vacinações ( quando toma a primeira dose, já ficam agendadas as datas dos reforços dessa vacina), estabeleça:

- a. Uma integridade de domínio:
  - b. Uma integridade de entidade:
  - c. Uma integridade referencial:
2. Para um sistema desenvolvido para a semana de informática, tem-se um cadastro de alunos contendo o RA, nome, endereço e turma, um cadastro de palestrantes contendo nome, RG, CPF (para efetuar o pagamento e emitir recibo), formação, endereço, telefone, e\_mail, um cadastro de palestras contendo data, hora, tema, RG do palestrante, duração estimada e custo. São cadastrados também os patrocinadores que colaboraram financeiramente com o evento, sendo cadastrados os seguintes dados: nome do patrocinador, endereço e valor da contribuição. Para esse sistema, estabeleça:
- a. Uma integridade de domínio:
  - b. Uma integridade de entidade:
  - c. Uma integridade referencial:

## **NORMALIZAÇÃO**

A Normalização é uma Técnica, que consiste de um Processo Matemático Formal fundamentado na Teoria dos Conjuntos;

Através do Processo Matemático da Normalização, pode-se substituir, gradativamente, um conjunto de Entidades e Relacionamentos por um outro, mais “adequado”, em relação a Anomalias de Atualização (Inclusão, Alteração e Exclusão);

Anomalias de Atualização em um Banco de Dados podem causar problemas tais como a ocorrência de: grupos repetitivos de dados; dependências parciais de chave; redundâncias desnecessárias de dados; perdas acidentais de informações; dificuldades de representações de fatos da realidade (modelos); e dependências transitivas entre atributos;

Todas essas dificuldades podem ser reduzidas ou minimizadas através do uso da Técnica de Normalização. A utilização da Técnica de Normalização torna o Modelo de Dados, que se estiver utilizando, bastante estável, isto é, sujeito a poucas Manutenções.

Para fazer a normalização é necessário analisar os atributos em relação à chave primária de sua entidade, verificando-se uma série de requisitos que devem ser cumpridos para que se tenha um modelo de dados correto.

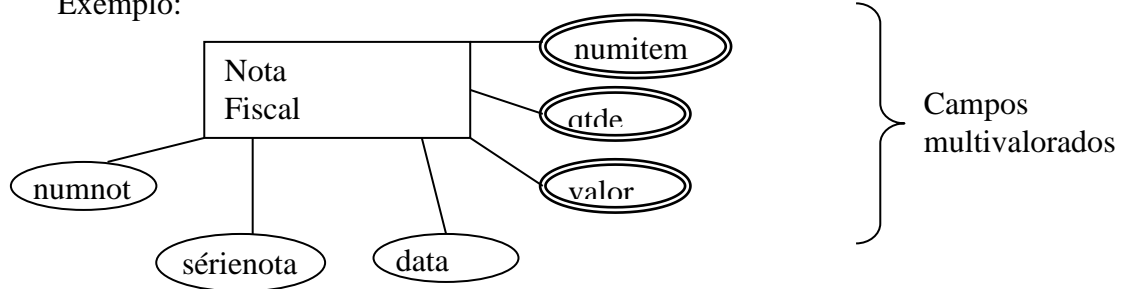
Segundo os ideais de normalização, dependendo do nível de refinamento em que um arquivo se encontre, ele estará numa determinada forma normal. A normalização envolve diversas regras, das quais as mais importantes são as regras das três formas normais, que estudaremos a seguir.

### **PRIMEIRA FORMA NORMAL (1FN)**

Uma entidade está na 1FN se o relacionamento entre a chave e os atributos for unívoca, ou seja, para cada chave há a ocorrência de uma e somente uma informação de cada atributo. Em outras palavras, uma entidade está na 1FN se não houver itens repetitivos dentro da mesma. Está na 1FN quando nenhum campo for multivalorado.

Caso não esteja na 1FN, uma nova entidade deverá ser criada, contendo a chave da primeira, e os itens que se repetem. Destes itens, deve ser escolhido um ou mais que será o discriminador, ou seja irá compor, juntamente com a chave primária da primeira entidade, a chave primária da nova entidade.

Exemplo:



Colocando na 1FN, a situação é a seguinte:

Nota\_Fiscal(numnota, sérienota, data)

Item\_Nota\_Fiscal (numnota, numitem, qtde, valor)

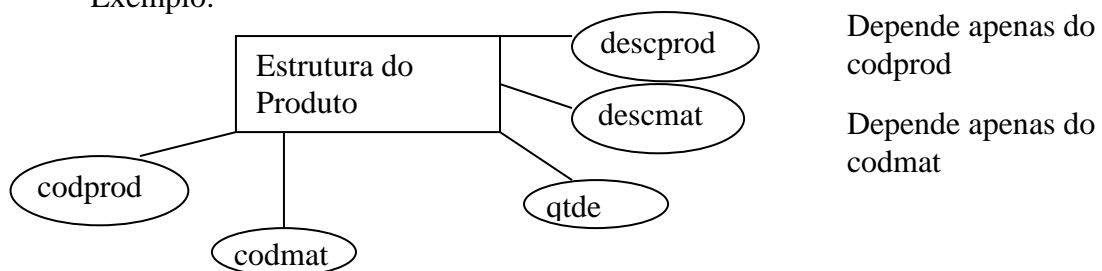
Todas as vezes que houver uma normalização para colocar as entidades na 1FN, ocorrerá a caracterização da dependência de existência, ou seja, a entidade item\_nota\_fiscal depende da existência da entidade Nota\_fiscal para existir. Portanto pode-se dizer que a Nota\_fiscal é uma entidade forte, porque não depende de outra entidade para existir, e a entidade Item\_nota\_fiscal é uma entidade fraca de Nota\_fiscal, já que depende desta para existir.

## SEGUNDA FORMA NORMAL (2FN)

Uma entidade está na 2FN se estiver na 1FN e cada atributo dependente (não chave) é funcionalmente dependente da chave, sem ser funcionalmente dependente de qualquer sub-conjunto da chave (dependência parcial da chave).

Se a entidade não possui chave composta, ou seja, com mais de um atributo, e estiver na 1FN ela estará automaticamente na 2FN. Se ela tiver chave composta e um dos mais de seus atributos dependerem de parte da chave, esta entidade deverá ser normalizada, criando uma ou mais novas entidades, cuja chave será apenas o sub-conjunto da chave primária da primeira entidade da qual dependem os outros atributos.

Exemplo:



Colocando na 2FN, a situação é a seguinte:

Estrutura\_do\_produto ( codprod, codmat, qtde)

Produto (codprod, descprod)

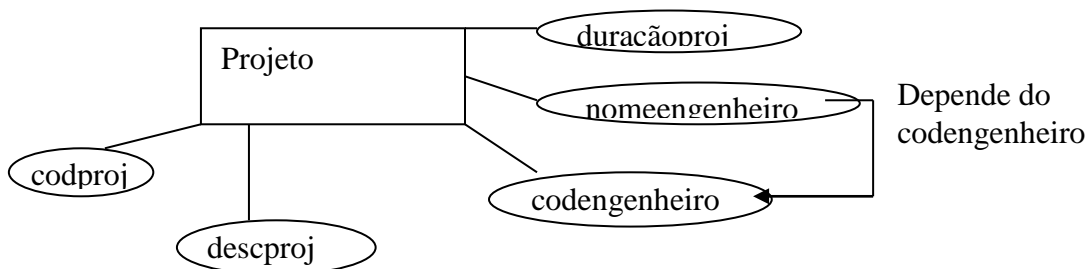
Material (codmat, descmat)

### TERCEIRA FORMA NORMAL (3FN)

Uma entidade está na 3FN se estiver na 2FN e os atributos dependentes não são dependentes transitivos da chave. Pó dependência transitiva entende-se a dependência indireta da chave, ou seja, um atributo depende de outro atributo (que não é a chave primária) que por sua vez depende da chave primária da entidade.

Se houver dependência transitiva, é necessário, para normalização da 3FN, a criação de uma nova entidade, cuja chave será o atributo do qual outros atributos dependem e estes serão atributos da nova entidade.

Exemplo:



Colocando na 3FN, a situação é a seguinte:

Projeto (codproj, descproj, codengenheiro, duraçãoproj)

Engenheiro (codengenheiro, nomeengenheiro)

**Exemplo:** Dada a tabela Funcionário a seguir, aplique as três formas normais:

<u>Matrícula</u>	Nome	Salário	Data Admissão	Cargo	Dependentes
12223	José	R\$ 5500,00	05/08/1972	Engenheiro	Mário, Ana
12334	Maria	R\$ 1200,00	06/06/1984	Desenhista	Roberto
12455	Ana	R\$ 1200,00	07/06/1980	Desenhista	Guilherme, Thais

#### 1FN

<u>Matrícula</u>	Nome	Salário	Data Admissão	Cargo
12223	José	R\$ 5500,00	05/08/1972	Engenheiro
12334	Maria	R\$ 1200,00	06/06/1984	Desenhista
12455	Ana	R\$ 1200,00	07/06/1980	Desenhista

<u>Matrícula</u>	<u>Dependentes</u>
12223	Mário
12223	Ana
12334	Roberto
12455	Guilherme
12455	Thais

**2FN** – As duas tabelas estão na 2FN

#### 3FN

<u>Matrícula</u>	Nome	Data Admissão	Cargo
12223	José	05/08/1972	Engenheiro
12334	Maria	06/06/1984	Desenhista
12455	Ana	07/06/1980	Desenhista

<u>Cargo</u>	Salário
Engenheiro	R\$ 5500,00
Desenhista	R\$ 1200,00

**Exercícios:**

Faça a normalização das tabelas a seguir:

1.

CodVda	CodCli	NomeCli	DataVda	ValorTotal	CodProd	Descr	Qtd	ValorUnit	TipoProd
1	1	José	01/02/1998	900,00	1	TV Sony	1	350,00	1
					10	Rack p/ Tv	1	550,00	3
2	55	Maria	10/01/1998	300,00	140	Cobertor	5	60,00	5

2.

CodVda	CodCli	NomeCli	DataVda	NrParc	VencParc	ValParc	CodProd	Descr	Qtd	ValorUnit	TipoProd
1	1	José	01/02/1998	0	01/02/1998	900,00	1	TV Sony	1	350,00	1
							10	Rack p/ Tv	1	550,00	3
2	55	Maria	10/01/1998	3	10/02/1998	100,00	140	Cobertor	5	60,00	5
					10/03/1998	100,00					
					10/04/1998	100,00					

3.

Cod	Categoria	Descrição	Marca	Série	Fornecedor	Capacidade	Precisão	Responsável
3	Paq	Paquímetro	MITUTOYO	2145440	Andorinha Ferramentas		0,05 1/128	Alexandre
4	Paq	Paquímetro	DIGIMESS	5478850	Casa Agrícola	150-6"	0,02 1/128	Eduardo
5	Mic	Micrometro	MITUTOYO	3548879	Casa Agrícola	0-25	0,001	Rodrigo
6	Esc	Escala	DIGIMESS	4564564	Andorinha Ferramentas	1000-40"	0,5	Willian
7	Esc	Escala	MITUTOYO	1251251	Casa Agrícola	150-6"	0,5	Alexandre

4. Normalize:

- Disciplina(código, nome, num\_aulas, cod\_curso, nome\_curso, cod\_prof, nome\_prof, Ra, nome\_aluno, data\_matrícula)
- Participação(codEmpregado, codProjeto, horasTrabalhadas, nomeEmpregado, nomeProjeto, localProjeto)
- Cliente(codCli, nomeCli, end, bairro, CEP, tel RG, codProduto, descProduto, marcaProduto, valorUnit, qtde, codFornecedor, nomeFornecedor, telFornecedor)

5. Observe a ficha de inscrição abaixo. Refere-se a inscrição de processo seletivo de uma instituição.

Ficha de Inscrição Processo Seletivo							
Inscrição :	00010	Candidato :	Robson Vieira	Data Inscrição:	03/06/2008		
Rua :					Bairro		
Cidade					Cep:		
Telefone :			Celular :			email:	
Código Curso:		Nome do Curso:		Código Turno:		Nome Turno:	
Responsável (caso seja menor de idade):							

Aplicar as 3 Formas Normais cabíveis, passo a passo, caso se faça necessário. Caso não utilize as três formas normais, indique qual ou quais foram usadas.

6. Analise o histórico de um dos alunos de uma faculdade:

Universidade Estadual - Centro de Tecnologia

Curso de : Análise de Sistemas

Código do Curso: 0037

Aluno: Victor Alexandre Costa

Matricula: 007043

Status: Regular

Histórico

Disciplina	Professor	Nota	Faltas	Situação
Análise de sistemas	Roberto Carlos	7,5	7	Aprovado
Matemática	Jandira	8.0	4	Aprovado
Inglês	Junior Villas	4.5	0	Reprovado

Aplicar as 3 Formas Normais cabíveis, passo a passo, caso se faça necessário. Caso não utilize as três formas normais, indique qual ou quais foram usadas.