

TLBD I – Parte Teórica

Material de Apoio

Introdução

- A área de banco de dados é de grande importância no mundo da informática, uma vez que a informação é um bem precioso e deve ser armazenada de forma coerente e adequada.
- Atualmente, por menor e mais simples que seja um Sistema de Informação, ele precisará ter a capacidade de armazenar e recuperar dados rapidamente. Por exemplo, se você desenvolver um Sistema de Informação para um salão de beleza do seu bairro, este sistema terá que armazenar dados de clientes, produtos, valores, funcionários, serviços, etc. É para armazenar essas informações e recuperá-las rapidamente que utilizamos um sistema de banco de dados.

Introdução

- Para que possamos entender o que está compreendido na área de banco de dados, ou seja, quais os modelos, quais as ferramentas disponíveis, como modelar, implementar uma base de dados e como recuperar dados, é necessário compreender alguns conceitos básicos sobre o assunto.

O Que são dados? O que é
Informação? O Que é uma Base
de Dados?

Dados

- Dados são tudo que podemos inferir ou coletar sobre uma situação específica. Os dados podem ser úteis ou não. Por exemplo, em uma sala de aula a quantidade de carteiras, a cor da parede, o tipo do assoalho, as dimensões da sala, etc., fornecem-nos dados sobre o ambiente. No entanto, no desenvolvimento de uma aplicação, esses dados podem ser úteis ou não, dependendo do objetivo do projeto. Se, por exemplo, uma arquiteta tem a intenção de desenvolver um novo leiaute para aquela sala de aula, os dados acima mencionados serão úteis.

Informações

- Os dados úteis é o que chamamos de informação. E esses dados são o que armazenamos em uma base de dados. Por convenção, na área de banco de dados, os termos “informação” e “dado” significam a mesma coisa. Isso ocorre porque devemos armazenar apenas aquilo que é útil para a nossa aplicação. Sendo assim, daqui para frente, os dois termos serão usados como sinônimos.

O Que é uma Base de Dados?

- Uma base de dados é um local, ou espaço, onde informações estão armazenadas e de onde elas são recuperadas. Uma base de dados terá um nome, e este nome deverá representar o que aquela base armazena. Por exemplo, se a aplicação for uma agenda de contatos pessoal e profissional, o nome da base poderá ser `bd_Agenda`. Essa base de dados deverá armazenar todas as informações sobre os contatos pessoais e profissionais como, por exemplo, nome, endereço, tipo do endereço (residencial, comercial, de referência, etc.), telefone residencial, telefone celular, telefone comercial, nome da empresa em que trabalha, grau de parentesco, e-mail, etc. Uma base de dados permite que os dados fiquem centralizados e que se relacionem de forma coerente.

O Que É um Sistema Gerenciador de Banco de Dados (SGBD)?

- Um SGBD disponibiliza uma série de funcionalidades que permitem controlar e acompanhar melhor os dados armazenados. As principais características de um SGBD são:
- Permitir o acesso concorrente às bases de dados;
- Realizar o gerenciamento de transações;
- Permitir criar e aplicar regras de segurança às bases de dados;
- Permitir criar regras que garantam a integridade da base de dados.

Quem usa um banco de dados?

- Administrador do banco de dados (DBA): é o responsável por monitorar e gerenciar todas as bases de dados criadas no SGBD. Também é quem controla as permissões dos usuários, garante que os usuários tenham acesso aos dados, realiza backups, recupera os dados em caso de falhas, garante o melhor desempenho para o banco de dados, monitora serviços (Jobs) de usuários no banco de dados, etc. Normalmente, um DBA é responsável por um SGBD específico, e deve ter estudado e feito cursos para aquele SGBD (por exemplo: DBA Oracle ou DBA SQL Server)

Quem usa um banco de dados?

- Analistas de sistemas e programadores de aplicações: são responsáveis por modelar a base de dados e implementá-la no SGBD escolhido. Também são responsáveis por desenvolver a aplicação (programa escrito em uma linguagem de programação como: Java, PHP, C++, C#, etc.) e conectar essa aplicação à base de dados do sistema. Esse usuário precisa conhecer a área de banco de dados, saber modelar uma base de dados e também conhecer a linguagem SQL.

Quem usa um banco de dados?

- Usuários finais: os usuários finais são aquelas pessoas que vão trabalhar diariamente com as aplicações desenvolvidas. São eles os responsáveis pela entrada de dados no banco de dados e pelas alterações nos dados armazenados. Esses usuários não precisam ter nenhum conhecimento sobre banco de dados ou saber qual o SGBD utilizado. Para eles, o banco de dados é transparente, e só interessa que as informações estejam sendo salvas e possam ser recuperadas.

Fases para a criação de um Banco de Dados

1ª Fase - Modelagem Conceitual

- A modelagem conceitual refere-se ao desenvolvimento de um modelo inicial da base de dados que reflita as necessidades do usuário. Essa modelagem preocupa-se em descrever quais dados serão armazenados na base de dados e quais dados se relacionam. Para fazer o modelo conceitual, é necessário entender o que o usuário final espera que o sistema armazene e que informações este usuário espera que o sistema disponibilize (como por exemplo, relatórios). Para obter as informações necessárias para desenvolver a modelagem conceitual do sistema, deve-se realizar entrevistas com o usuário para entender os objetivos do sistema e as expectativas que o usuário tem em relação a ele. Um dos principais modelos desta etapa é o **Modelo de Entidade e Relacionamento**

2ª Fase – Modelagem Lógica

- A modelagem lógica compreende o processo de descrever como os dados serão armazenados no sistema e como irão se relacionar. Isso significa transformar o modelo conceitual obtido na primeira fase num modelo mais próximo da implementação. Para banco de dados relacionais, o modelo utilizado nessa fase é o modelo relacional. Também é necessário descrever o dicionário de dados da base de dados nessa etapa. Antes da fase de implementação, é necessário, ainda verificar se o modelo está normalizado e em caso negativo, deve-se normalizar o modelo.

3ª Fase – Implementação de Modelo Lógico

- Uma vez que toda a etapa de modelagem esteja concluída, será necessário implementar ou criar a base de dados no SGBD escolhido. Essa fase requer que o desenvolvedor conheça a Linguagem SQL e conheça o SGBD selecionado.

Importante

- A importância de uma boa modelagem se deve ao fato de que as aplicações que estarão acessando a base de dados deve estar de acordo com o modelo desenvolvido. É muito desanimador e trabalhoso perceber a necessidade de alterar o modelo de dados, depois da base de dados programada e da aplicação do usuário desenvolvida. A verdade é que isso gera um retrabalho, uma vez que não só a implementação da base terá que ser refeita, mas também os diagramas e a aplicação deverão ser atualizadas na maioria dos casos.

Importante

- A fase de modelagem é a principal etapa no desenvolvimento de uma base dados. Por isso é muito importante que se dedique tempo e esforço no desenvolvimento de uma boa modelagem da base de dados.

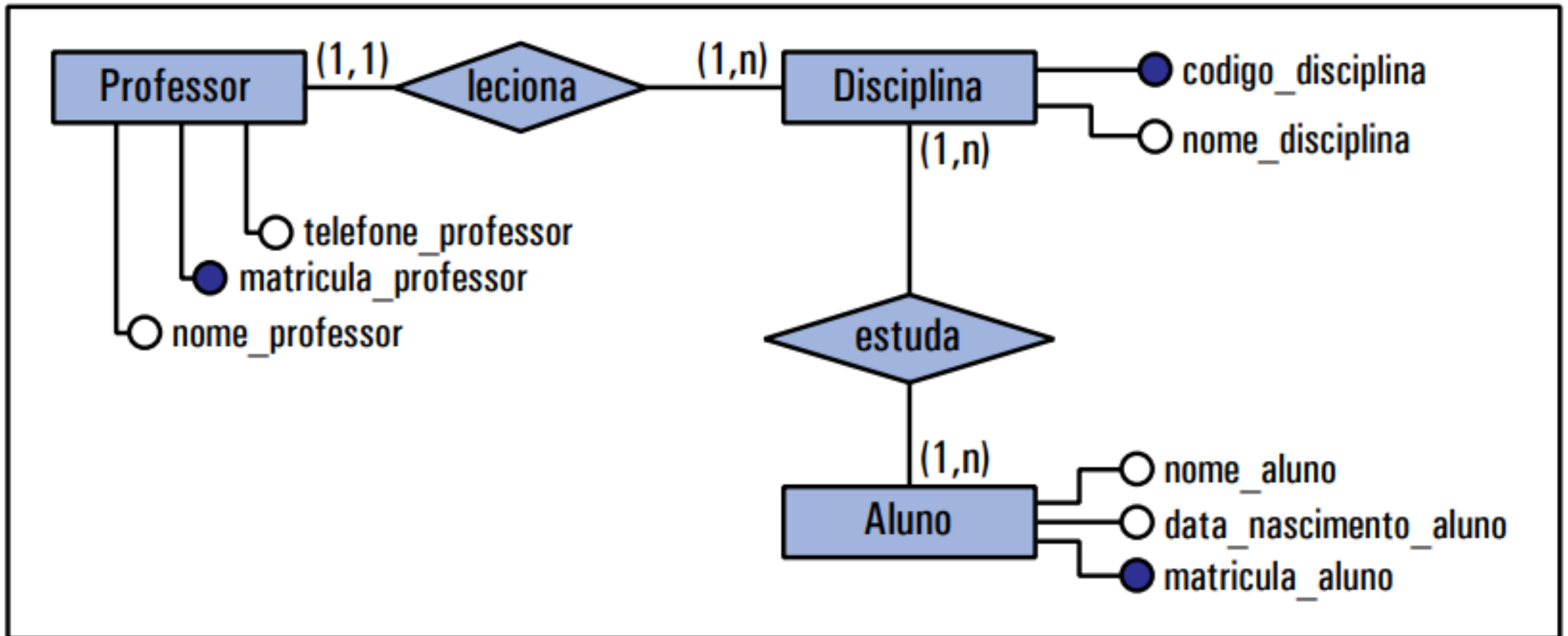
Introdução - DER

- O Diagrama de Entidade e Relacionamento (DER) é um modelo conceitual e deve estar o mais próximo possível da visão que o usuário tem dos dados, não se preocupando em representar como estes dados estarão realmente armazenados.
- Este modelo tem por objetivo descrever quais dados devem ser armazenados pela aplicação e quais desses dados se relacionam.

Introdução - DER

- Suponha que uma escola precise armazenar informações sobre seus alunos, professores, e disciplinas. O modelo da base de dados deve informar quais dados sobre alunos, professores e disciplinas são importantes para serem armazenados. Assim, sobre os alunos, pode-se armazenar informações como nome, data de nascimento e matrícula. Sobre os professores, armazenam-se informações como nome, telefone e matrícula. A respeito das disciplinas, pode-se armazenar informações como código da disciplina e o nome da disciplina. Além dessas informações, é necessário saber qual professor é responsável por qual disciplina e quais alunos fazem a disciplina

Resolução – DER - Exemplo



Introdução - DER

- Note que não estamos preocupados aqui em saber como os dados serão armazenados ou como eles devem ser implementados. A nossa preocupação é conseguir entender o que precisa ser armazenado e quais informações devem se relacionar.
- Observe que o DER informa que a base de dados contém informações sobre professores, alunos e disciplinas, mas não se preocupa em descrever o valor que esses dados armazenam.
- Um modelo de Entidade e Relacionamento consiste em um conjunto de objetos básicos chamados entidades e de relacionamentos entre as entidades.

Entidades e Atributos

- Você desenvolve um Sistema de Informação para resolver um problema do mundo real. Assim, toda a informação referente ao problema que se quer solucionar representa parte deste mundo.
- Uma **entidade** representa um conjunto de objetos do mesmo tipo do mundo real e sobre os quais se pretende armazenar dados. Por exemplo, ao desenvolver um Sistema de Informação para uma escola, as possíveis entidades desse sistema serão: professores, alunos, disciplinas, turmas e cursos.

Entidades e Atributos

- Uma entidade é representada graficamente por um retângulo com o nome da entidade dentro do retângulo. Por exemplo:



- Cada uma dessas entidades armazenará um conjunto de objetos do mesmo tipo. Ou seja, ter uma entidade denominada professor significa que vários professores serão cadastrados nessa entidade e cada professor representa, portanto, um objeto da entidade.

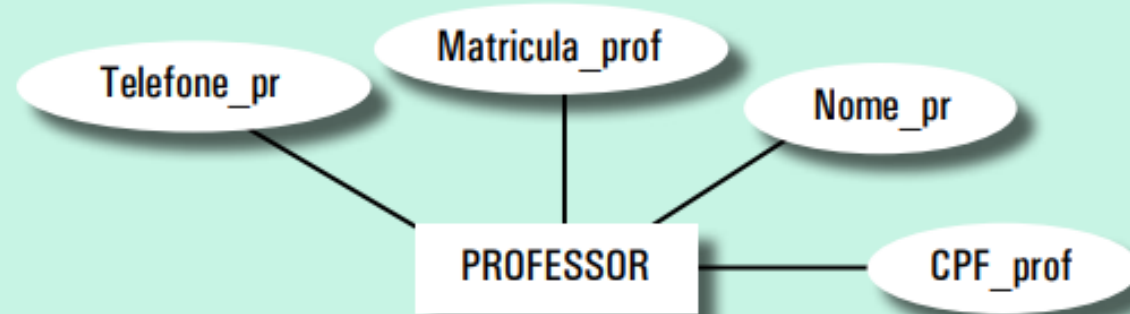
Entidades e Atributos

- Além de uma entidade representar objetos do mundo real, ela também deve possuir um conjunto de propriedades que a caracterize e a descreva, bem como aos seus objetos. A esse conjunto de propriedades dá-se o nome de atributos. Por exemplo, para a entidade Professor, é necessário armazenar dados como: CPF, nome, telefone, endereço, grau de escolaridade, número da matrícula, etc. Esses dados são os atributos da entidade Professor e são eles que identificam e caracterizam um objeto do tipo professor.

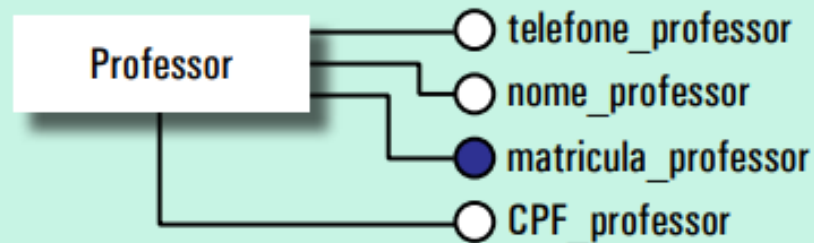
Entidades e Atributos

- Outro exemplo, para uma entidade chamada Cadeira, os possíveis atributos dessa entidade serão: número de pernas, cor, tamanho, peso, altura, tecido, etc. O nome dos atributos deve representar o que aquele atributo armazena.
- Um atributo pode ser representado graficamente por uma elipse com o nome do atributo dentro da elipse. A elipse é ligada à entidade por uma linha, conforme exemplo:

Entidades e Atributos



Outra forma de representação utilizada por algumas ferramentas é representar o atributo como uma bolinha ligada à entidade e com o nome do atributo ao lado, conforme exemplo:



Atributos

- Os atributos podem ainda ser divididos em 6 categorias: simples, compostos, monovalorado, multivalorado, derivado e nulo. É importante ressaltar que os atributos podem pertencer a mais de uma categoria ao mesmo tempo. Isso significa que é comum um único atributo ser simples, monovalorado e derivado ao mesmo tempo. A seguir, serão explicadas e exemplificadas cada uma das categorias

Atributos

- Atributo Simples: é o atributo indivisível, que não pode ou não deve ser decomposto. Por exemplo: “CPF”, “numero da matrícula”, “RG”, “preço do produto”, etc.
- Atributo composto: é o atributo que pode ser decomposto em outros atributos simples. Por exemplo, o atributo “endereço” pode ser decomposto em “nome da rua”, “número” e “complemento”.
- Atributo monovalorado: é o atributo que permite apenas o armazenamento de um valor por vez. Por exemplo, o atributo “CPF” é monovalorado porque uma pessoa possui apenas um número de CPF. Caso o CPF seja alterado ele é substituído pelo novo valor. Assim, uma pessoa nunca terá cadastrado mais de um CPF no mesmo campo

Atributos

- Atributo multivalorado: é o atributo que permite armazenar mais de um valor ao mesmo tempo no mesmo campo. Por exemplo, o atributo e-mail pode ser multivalorado uma vez que uma pessoa possui, normalmente, mais de um endereço de e-mail.
- Atributo nulo: é o atributo que permite que seja inserido um valor nulo para ele. Valor nulo representa a inexistência de um valor, ou seja, significa que o usuário não precisa cadastrar um valor para o atributo e pode deixá-lo vazio. Em algumas situações, é inevitável que permitamos valores nulos para os atributos. Vamos usar novamente o atributo “e-mail” como exemplo. Como nem todas as pessoas possuem e-mail, esse atributo deve permitir valores nulos, porque se ele não permitir algumas pessoas não poderão se cadastrar ou terão que criar um e-mail para poder efetivar o cadastro. Novamente é o usuário quem, muitas vezes, vai definir se um atributo é obrigatório ou não.

Atributos

- Atributo derivado: é o atributo cujo valor para ele deriva de outro(s) atributo(s). Por exemplo, suponha que a sua entidade se chame compra e que ela tenha os seguintes atributos: “número da compra”, “data da compra”, “valor da compra”, “percentual de desconto” e “valor da compra com o desconto”. O valor para este último atributo é calculado considerando-se o “valor da compra” e o “percentual de desconto”. Assim, esse atributo é derivado porque seu valor deriva dos valores de outros atributos e é calculado automaticamente pela aplicação ou pelo SGBD.

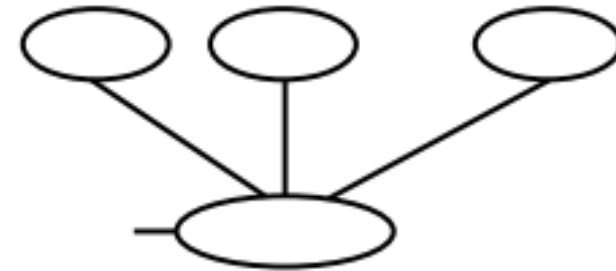
Simbologia - Atributos



ATRIBUTO SIMPLES
e MONOVALORADO



ATRIBUTO
CHAVE



ATRIBUTO
COMPOSTO



ATRIBUTO
MULTIVALORADO

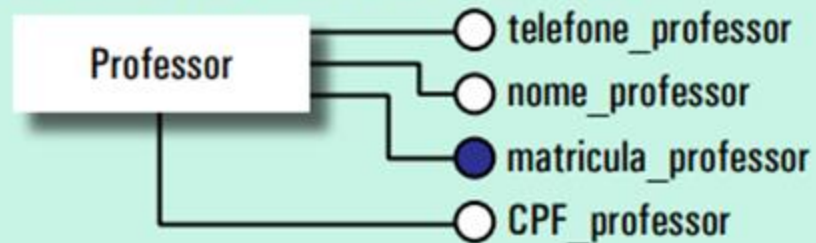


ATRIBUTO
DERIVADO

Atividade de Fixação

- Escolher três entidades de um sistema para Padaria.
- Para cada entidade, identificar 5 atributos.

Outra forma de representação utilizada por algumas ferramentas é representar o atributo como uma bolinha ligada à entidade e com o nome do atributo ao lado, conforme exemplo:



Chave Primária

- Um conceito importante no Modelo de Entidade e Relacionamento é o conceito de chave primária (ou Primary Key ou ainda PK). Uma chave primária é um atributo da entidade que identifica apenas um objeto dessa entidade. Portanto, o valor dentro de uma chave primária não poderá se repetir e também não poderá receber um valor nulo.
- Por exemplo, na entidade “Professor”, tanto o atributo “CPF” quanto o atributo “matrícula” não se repetem, uma vez que esses atributos são únicos para cada indivíduo. Nesse caso, qualquer um dos dois atributos poderia ser definido como uma chave primária.

Chave Primária

- A pergunta é: Qual deles eu devo definir como uma chave primária? Bem, se mais de um atributo for único para a entidade, a escolha da chave primária vai depender do escopo do problema e de como serão realizadas as consultas. Considerando-se que a maioria dos SGBD vincula um índice à chave primária, é interessante que essa chave primária seja o atributo mais utilizado nas consultas. Para o exemplo da entidade “Professor”, a maioria das consultas seriam feitas considerando-se a matrícula do professor na instituição e não o CPF. A opção por consultas que tenham como condição a matrícula do professor se deve por causa do escopo do problema, uma vez que a matrícula do professor é um atributo muito mais específico para a instituição de ensino. Sendo assim, o atributo `matricula_professor` seria a melhor opção para ser a chave primária.

Chave Primária

- Outro ponto importante a considerar durante a decisão de qual atributo deverá ser a chave primária é que se deve dar preferência a atributos numéricos (inteiros) em vez de atributos do tipo caractere, data ou hora. Como “CPF” é um atributo do tipo caractere, ele poderia ser descartado como chave primária por esse motivo também.

Tipos de Chave Primária

- Uma chave primária pode ser simples ou composta. Uma chave primária simples é aquela que será formada por apenas um atributo. Por exemplo: `matricula_professor`.
- Uma chave primária composta é formada por dois ou mais atributos. Por exemplo, imagine que tenhamos uma entidade chamada “Localização” e esta entidade tem os seguintes atributos: “nome da cidade”, “nome do estado”, “nome do país”. Cada um desses atributos sozinhos não pode ser chave primária porque eles se repetem, como mostra a figura a seguir:

nome_cidade	nome_estado	nome_país
Curitiba	Paraná	Brasil
Maringá	Paraná	Brasil
Campo Grande	Mato Grosso do Sul	Brasil
Campo Grande	Rio de Janeiro	Brasil

Como os atributos individualmente podem se repetir, vamos tentar encontrar uma chave primária composta. Sabemos que no Brasil um estado não tem duas cidades com o mesmo nome. Sendo assim, poderíamos criar uma chave primária composta do “nome da cidade” mais o “nome do estado”, porque o valor para esses dois atributos juntos nunca vai se repetir.

Tipos de Chave Primária

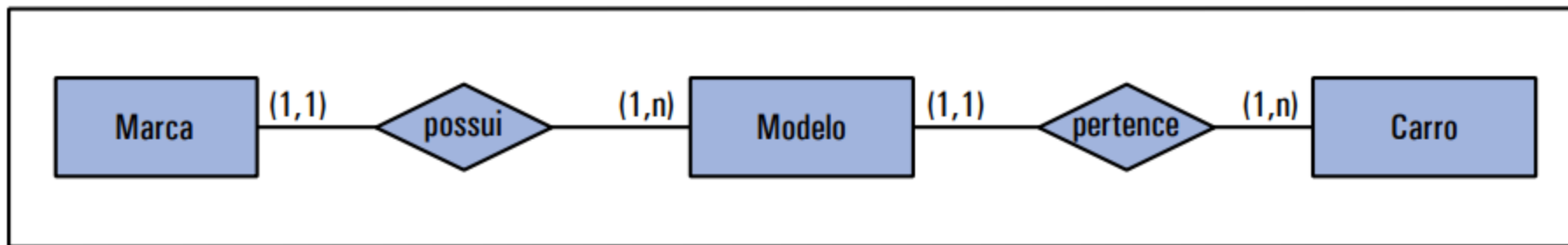
- Nesse caso, o atributo “nome da cidade” não é uma chave primária, e sim faz parte da chave primária.
- É importante destacar que não existe mais de uma chave primária por entidade. Essa chave primária poderá ser simples ou composta. Mesmo composta, é uma única chave primária composta de mais de um atributo.
- Simbologia:



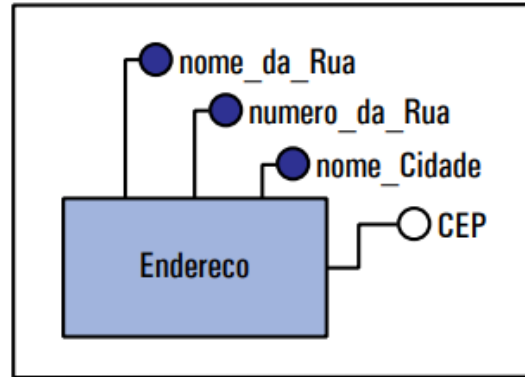
**ATRIBUTO
CHAVE**

Atividade de Fixação

1. Explique o que é uma chave primária e para que ela serve.
2. Com base nas entidades desenvolvidas na aula anterior, apresente 3 exemplos de atributos que poderiam ser chave primária na e explique o porquê.
3. Uma chave primária pode assumir o valor nulo? Justifique sua resposta.
4. Dado o diagrama de ER abaixo, coloque os atributos para cada entidade e marque as chaves primárias para cada entidade.



5. Dado o diagrama abaixo, pode-se afirmar que a entidade “Endereço” possui três chaves primárias? Justifique sua resposta.



6. Um DER pode mudar com frequência? Explique.

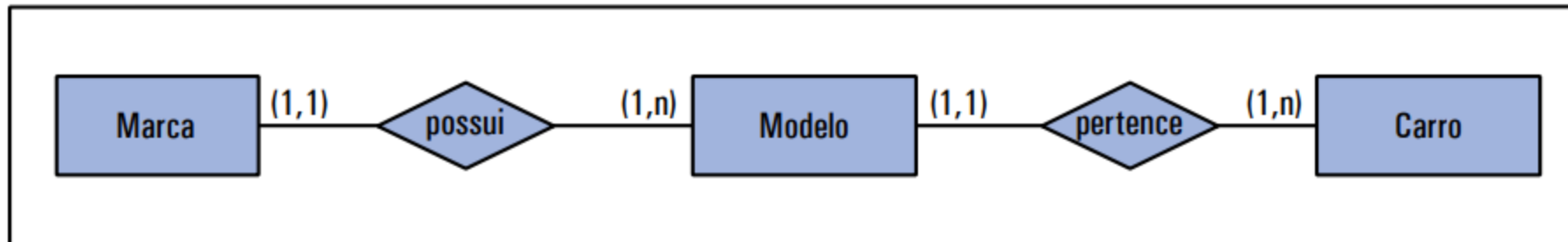
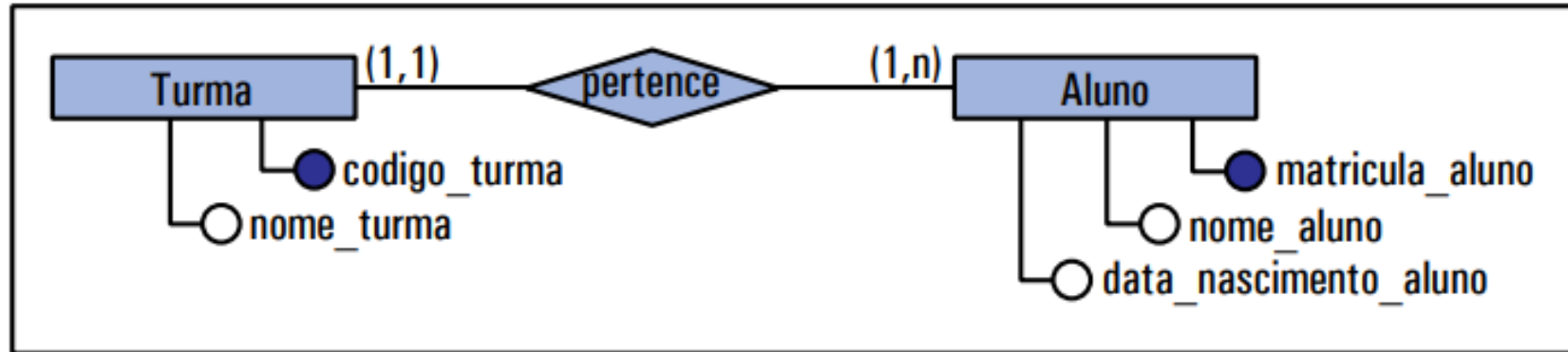
7. O que são atributos de uma entidade?

8. O que são objetos de uma entidade?

Relacionamento

- Um relacionamento é uma associação entre as entidades. Os dados devem ser armazenados e estarem relacionados na base de dados para que possamos utilizá-los eficientemente em um sistema, garantindo assim a integridade dos dados.
- Um relacionamento é representado por um losango com o nome do relacionamento no centro. O nome do relacionamento representa a relação que existe entre os objetos das entidades. O nome do relacionamento pode ser um verbo como, por exemplo: pertence, leciona, estuda, possui, etc.; ou também pode ser uma composição dos nomes das entidades como por exemplo “Aluno_Turma” em vez de “pertence”.

Relacionamento



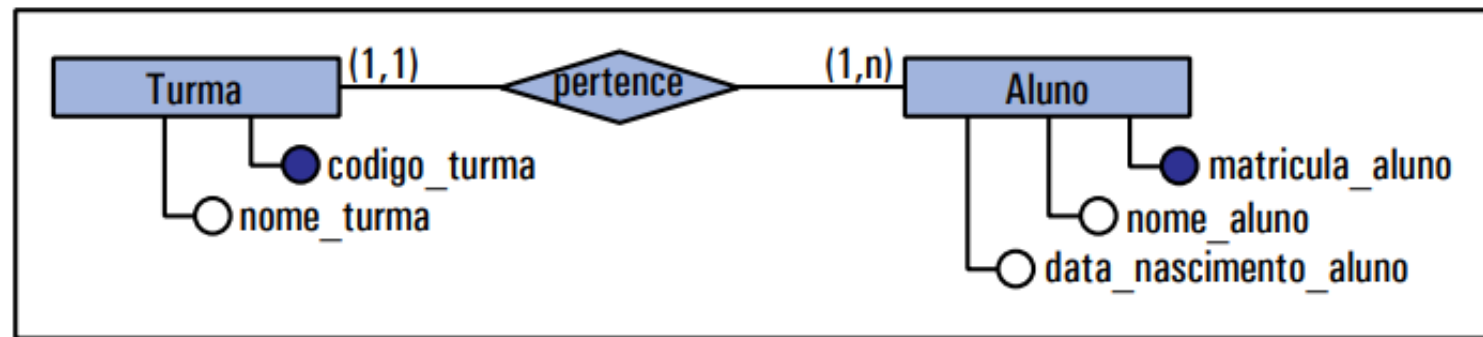
ANGELOTTI, E. S. Banco de Dados. Ed. Livro Técnico, 2012.

Cardinalidade

- Cardinalidade expressa o número de objetos de uma entidade ao qual outra entidade pode ser associada, via um relacionamento.
- Para descobrir a cardinalidade de um relacionamento, a pergunta que deve ser feita é: “Se eu pegar um único objeto da minha entidade X, a quantos objetos da entidade Y ele pode se associar?”
- Isto é, se eu pegar o objeto “Giovanni” da entidade “Professor”, esse objeto “Giovanni” poderá lecionar quantas disciplinas da entidade “Disciplina”? E se eu pegar o objeto “TLBD” da entidade “Disciplina”, a quantos professores ele estará relacionado?

Cardinalidade

- Esse número de associações entre objetos pode ser 0, 1 ou vários (representado por N).

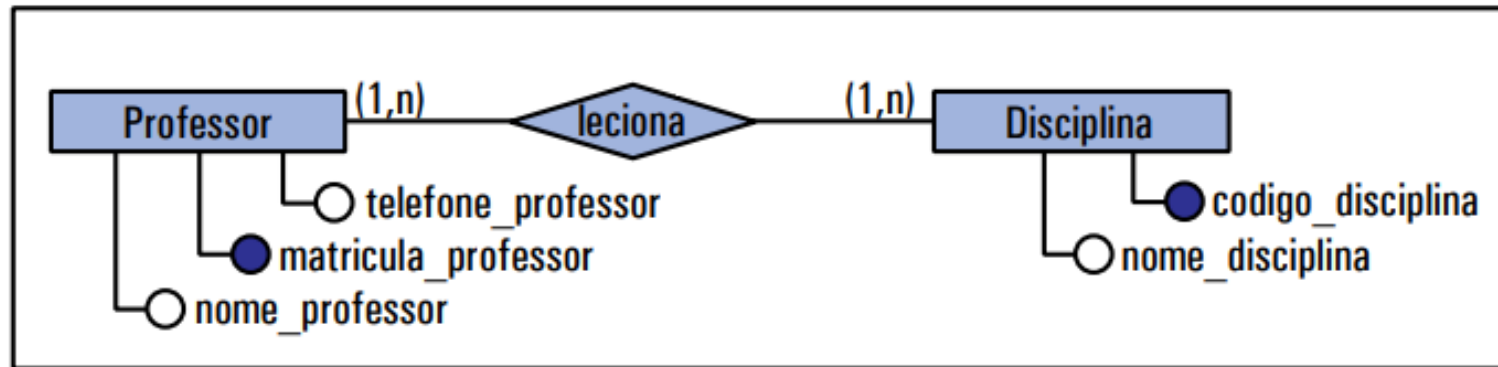


ANGELOTTI, E. S. Banco de Dados. Ed. Livro Técnico, 2012.

- Neste diagrama temos que uma turma pode ter um ou vários alunos e que um aluno pode pertencer a uma e apenas uma turma. Assim, não teremos o mesmo aluno em duas turmas diferentes.

Cardinalidade

- No diagrama abaixo, a cardinalidade do modelo é N para N, permitindo que um professor leciona várias disciplinas e que uma disciplina seja ministrada por mais de um professor. Por exemplo, a disciplina de TLBD poderia ser ministrada pelo professor Giovanni Guarnieri e pelo professor Felipe Roberto. Além de TLBD, o professor Giovanni poderia lecionar também, a disciplina de Análise de Projetos.



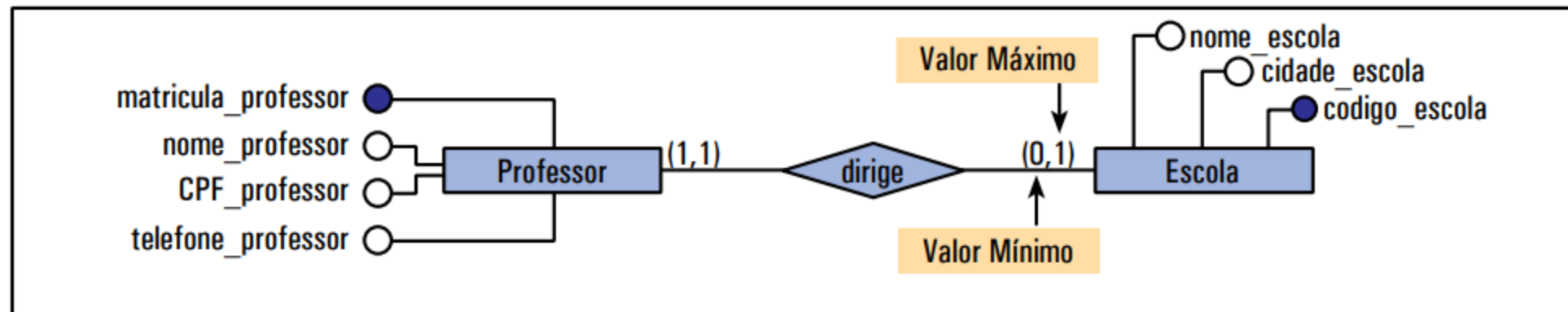
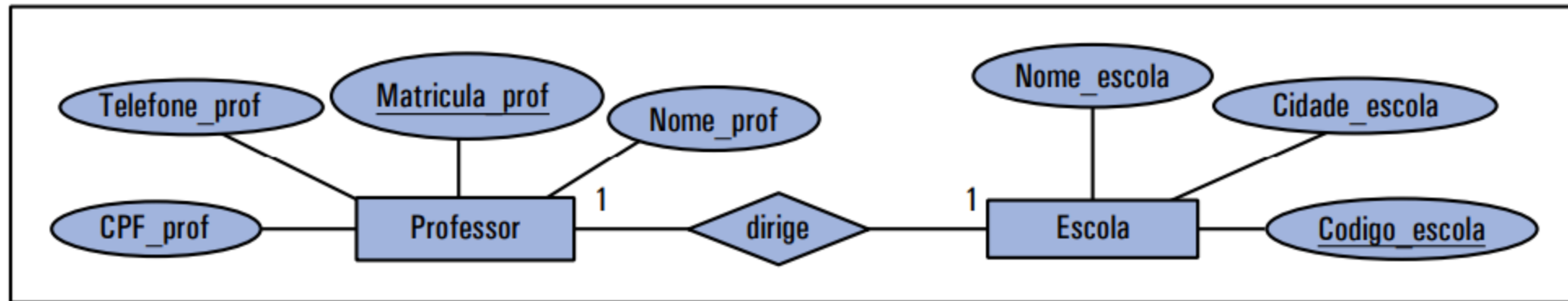
ANGELOTTI, E. S. Banco de Dados. Ed. Livro Técnico, 2012.

- A cardinalidade dos relacionamentos podem ser de 3 tipos: Um para Um; Um para Muitos; e Muitos para Muitos

Cardinalidade – 1 para 1

- A cardinalidade Um para Um (1:1) ocorre quando um objeto de uma entidade pode se relacionar a apenas um objeto de outra entidade e vice-versa. Imagine que você esteja desenvolvendo um sistema para uma rede de escolas. Cada escola terá um diretor (que é um professor da escola), e este diretor só poderá dirigir uma escola. Sendo assim, o relacionamento “dirige” entre as entidades “Escola” e “Professor” será 1:1

Cardinalidade – 1 para 1



Cardinalidade – 1 para N

- A cardinalidade Um para Muitos ocorre quando um objeto de uma entidade pode se relacionar a vários objetos da outra entidade, mas o contrário não é verdadeiro. Imagine que um professor possa trabalhar em apenas uma escola da rede municipal e que uma escola possa ter vários professores.

Cardinalidade – N para N

- A cardinalidade Muitos para Muitos (N:N) ocorre quando um objeto de uma entidade pode se relacionar a vários objetos da outra entidade e vice-versa. Imagine que um professor possa ministrar aulas em várias disciplinas e que uma disciplina possa ser ministrada por vários professores. Nesse caso, temos um relacionamento N:N, como mostra a figura 3.6.

Atividade de Fixação – Cardinalidade

- Desenvolva o Diagrama Entidade/Relacionamento para as seguintes situações:
 1. Um aluno realiza vários trabalhos. Um trabalho é realizado por um ou mais alunos.
 2. Um diretor dirige no máximo um departamento. Um departamento tem no máximo um diretor.
 3. Um autor escreve vários livros. Um livro pode ser escrito por vários autores.
 4. Uma equipe é composta por vários jogadores. Um jogador joga apenas em uma equipe.
 5. Um cliente realiza várias encomendas. Uma encomenda diz respeito apenas a um cliente.

Atividade de Fixação

- Sistema Simples para uma Biblioteca.

Considerar as seguintes entidades: Editora, Livro, Aluno, Funcionário e Locação.

- Uma editora pode conter vários Livros, um Livro só pode pertencer a uma editora
- Um aluno pode efetuar diversas locações, uma locação só pode ser efetuada por um aluno.
- Uma locação pode conter diversos livros, um livro pode ser alugado diversas vezes.
- Um funcionário pode efetuar diversas locações, uma locação só pode ser realizada por um funcionário.
- Com base nestas entidades, desenvolver um DER contendo atributos e relacionamento com suas devidas cardinalidades.

Referência Bibliográfica

- DATE, C. J. Introdução a Sistemas de Banco de Dados. Rio de Janeiro: Elsevier, 2003.
- ELMASRI, R.; NAVATHE, S. B. Sistema de Banco de Dados. Ed. Pearson, 2005.
- ANGELOTTI, E. S. Banco de Dados. Ed. Livro Técnico, 2012.
- HEUSER, C. A. Projeto de **Banco** de Dados. 6. ed. Editora Bookman, 2009.
- <<http://pronatec.ifpr.edu.br/wp-content/uploads/2012/07/admbd.pdf>> Márcia Cristina Dadalto Pascutti. ADMINISTRADOR DE BANCO DE DADOS
- KRIEGEL, A.; TRUKHNOV, B. M. SQL Bible. Indianápolis: Wiley, 2003.
- SILBERCHATZ, A. et al. Sistema de Banco de Dados. 3. ed. Makron Books, 1999.