

# SISTEMAS DE BANCO DE DADOS 2

## AULA 5

### Modelo Relacional de Dados e Controle de Consistência

Vandor Roberto Vilardi Rissoli



# APRESENTAÇÃO

- Modelo de Dados Relacional (MR)
- Chave e Restrições de Integridade
- Mapeamento do ME-R para MR
- Controle de Consistência e Dependência Funcional
- Normalização
- Referências

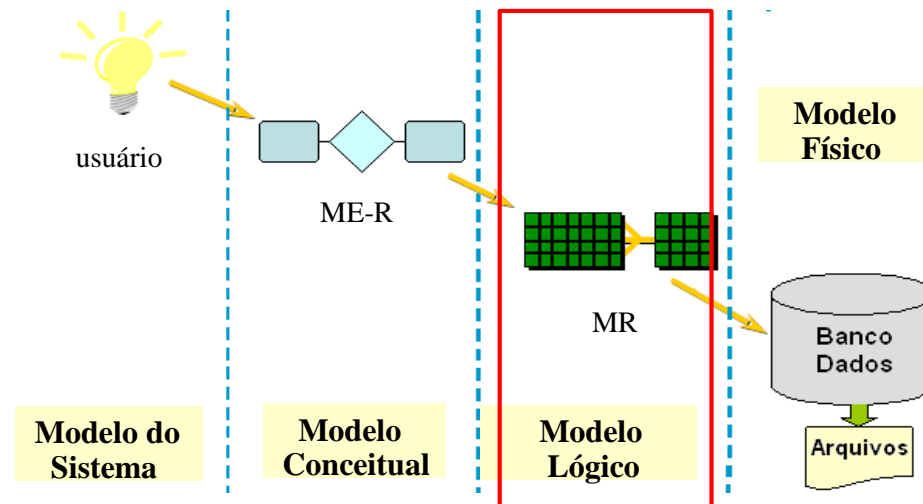


# Modelo de Dados (revisão)

- Conjunto de conceitos utilizados para descrever a estrutura de um banco de dados
- Principal ferramenta no fornecimento de informações sobre a abstração efetuada

## Modo de Implementação (mapeamentos)

... Modelo Conceitual → **Modelo Lógico** → Modelo Físico



# Esquema e Instância

## ESQUEMA

- A descrição da organização dos dados de um BD.
- Um esquema de BD é especificado durante o projeto do BD e não é frequentemente modificado

## INSTÂNCIA

- Os dados armazenados em um BD, em um momento específico, são denominados *instâncias* do BD - (seria a fotografia do BD em um instante)



# Modelo de Dados Relacional

O Modelo Relacional representa os dados em um BD, por meio de um conjunto de relações (tabelas).

Estas relações contém informações sobre entidades ou relacionamentos existentes no domínio da aplicação utilizada como alvo da modelagem.

Informalmente, uma relação pode ser considerada como uma tabela de valores, onde cada linha desta tabela representa uma coleção de valores de dados inter-relacionados.



# Modelo de Dados Relacional

Os nomes fornecidos às tabelas e às suas colunas podem auxiliar na compreensão do significado dos valores armazenados em cada uma de suas linhas (os registros).

Na terminologia do Modelo Relacional, cada linha da tabela é chamada de **TUPLA**, a tabela é denominada **RELAÇÃO**, o nome da coluna é **ATRIBUTO** da relação, e o conjunto de valores que cada atributo pode assumir em uma determinada relação corresponde ao seu **DOMÍNIO**.



# Modelo de Dados Relacional

O domínio consiste do grupo de valores atômicos a partir dos quais um ou mais atributos obtêm seus valores reais.

O esquema de uma relação é o conjunto de atributos que descreve as características dos elementos a serem modelados. A quantidade desses atributos é denominado Grau da Relação, por exemplo, **FUNCIONÁRIO** abaixo tem grau 4.

**FUNCIONÁRIO**

Código	Nome	Categoria	Cidade
F1	Paulo	20	Lins
F2	César	10	Palmas
F3	Carlos	30	Lins
F4	Matilde	20	Brasília

domínio

Esquema

Relação

Instâncias

tupla

# Modelo de Dados Relacional

A instância de uma relação consiste no conjunto de valores que cada atributo, definido no esquema, assume em um determinado instante, formando o conjunto de tuplas.

As instâncias das relações formam os dados que são armazenados no BD.

Observe no exemplo anterior:

- O domínio do atributo **Cidade** é o conjunto de todos os nomes válidos de cidades nacionais (Lins, Palmas, etc.).
- O esquema de **FUNCIONÁRIO** é definido como *código*, *nome*, *categoria* e *cidade*, possuindo grau 4 (4 atributos).
- As instâncias são os dados que são armazenados no BD.





# Modelo de Dados Relacional

## Características das Relações (tabelas)

- Não há tuplas duplicadas em uma relação.
- A ordem das tuplas não é relevante para diferenciar uma relação de outra.
- Existe ordem dos valores nas tuplas, uma vez que o cabeçalho da relação (ou esquema) é definido como um conjunto de atributos.
- Os valores dos atributos devem ser atômicos, não sendo divisíveis em componentes.
  - Atributos **multivalorados** são representados por meio de uma outra relação;
  - Atributos **compostos** são pelos seus componentes.



# Modelo de Dados Relacional

## Chaves e Restrições de Integridade Relacional

- Como não pode haver uma tupla repetida (duplicada) em uma instância da relação, é necessário identificar uma atributo ou mais que terá a responsabilidade de identificar **unicamente cada tupla**;
- Este atributo, ou conjunto de atributos, é chamado de chave da relação;
- A definição dessa chave será identificada como **chave primária** da relação;
- A chave primária composta por mais que um atributo é chamada de **chave primária composta**.



# Modelo de Dados Relacional

## Restrições de Integridade

Restrições de Chaves: cada atributo das chaves candidatas deve possuir **valor único** em todas as tuplas da relação.

Restrição de Integridade de Entidade: uma chave primária **não** pode assumir valor **nulo** nas tuplas da relação.

Restrição de Integridade Referencial: inclusão de um atributo de uma relação que referência outra tupla em outra relação, surgindo um **novo tipo de chave**.

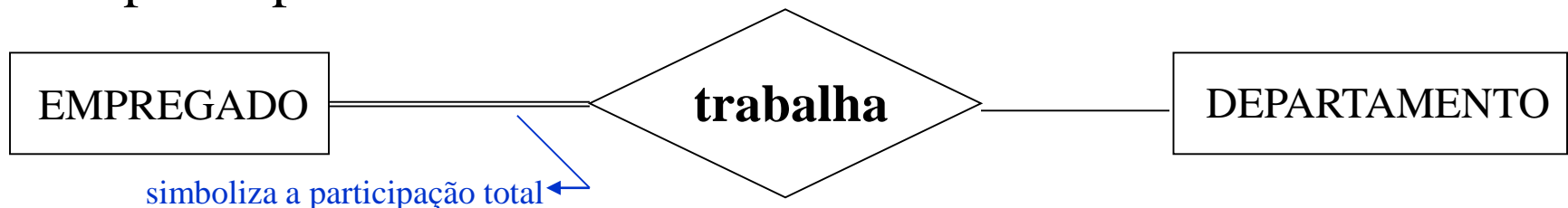
Restrições de Integridade Semântica: se referem mais especificamente sobre **valores ou características que determinados atributos podem assumir** no contexto de uma determinada aplicação (por exemplo sexo).

# Modelo de Dados Relacional

## Restrição de Participação

A restrição de participação determina que a existência de uma entidade depende de sua participação em um tipo de relacionamento. Existem dois tipos de restrições de participação: *total* e *parcial*. A participação total está estreitamente relacionada a **dependência de existência**.

a) Participação **Total**: Suponha que um empregado deve trabalhar em um departamento, ou seja, não existem empregados que não estejam vinculados a algum departamento. Assim, uma entidade do tipo **EMPREGADO** existe somente se ela participa em uma instância de relacionamento **trabalha**.



# Modelo de Dados Relacional

- b) Participação *Parcial*: o tipo de relacionamento **gerencia**, o qual envolve os tipos de entidade **EMPREGADO** e **DEPARTAMENTO**, a participação do tipo de entidade **EMPREGADO** é *parcial*, pois somente alguns empregados gerenciam departamentos, o que não limita a existência dos demais empregados.





# Modelo de Dados Relacional

## Regras para o Mapeamento

Os passos necessários, a partir de um projeto conceitual (ME-R), para realizar a modelagem do projeto lógico de implementação do BD por meio do Modelo Relacional é realizado seguindo as seguintes regras:

1. Todas as entidades são mapeadas para uma relação contendo os mesmos atributos do ME-R.
2. Para entidade fraca é criada a relação contendo todos os seus atributos, tendo acrescido, como chave estrangeira, a chave primária da entidade forte (pai).



# Modelo de Dados Relacional

3. Para relacionamentos 1:1 - dentre as relações que mapeiam as entidades participantes escolha uma delas (a que possuir participação total) e inclua como chave estrangeira a chave primária da outra.
4. Para relacionamentos 1:N – escolha a relação que representa a entidade presente no lado N, e acrescente como chave estrangeira a chave primária da entidade do lado 1 (esta regra não se aplica para os relacionamentos de identificação - fracos).
5. Para relacionamentos N:M – é criada uma nova relação contendo como chaves estrangeiras as chaves primárias das entidades participantes, mais os atributos do relacionamento.



# Modelo de Dados Relacional

6. Os atributos multivalorados ( $A$ ) - é criada uma relação  $R$  que terá como atributos os mesmos de  $A$ , mais a chave primária da entidade (ou relacionamento) no qual  $A$  é atributo. Se o atributo multivalorado é composto, será incluído seus componentes.
7. Para relacionamentos triplos (ou **ternários**) o mapeamento ocorre de forma semelhante ao descrito pela regra 5, apenas considerando que se uma das entidades apresentar um limite máximo de participação igual a 1, a chave desta entidade pode determinar a relação montada.



# Modelo de Dados Relacional

## Diagrama de Esquemas

Uma representação simplificada que identifica “bem” um esquema e suas inter-relações é o diagrama de esquema demonstrado a seguir, com uma representação simplificada do gráfico de *Gantt*:

### ALUNO

1	<u>matricula</u>	rg	nome	telefone	idCurso
---	------------------	----	------	----------	---------

### CURSO

2	<u>idCurso</u>	nome	periodo
---	----------------	------	---------

(n:1)



# Modelo de Dados Relacional

## Operações sobre as Relações

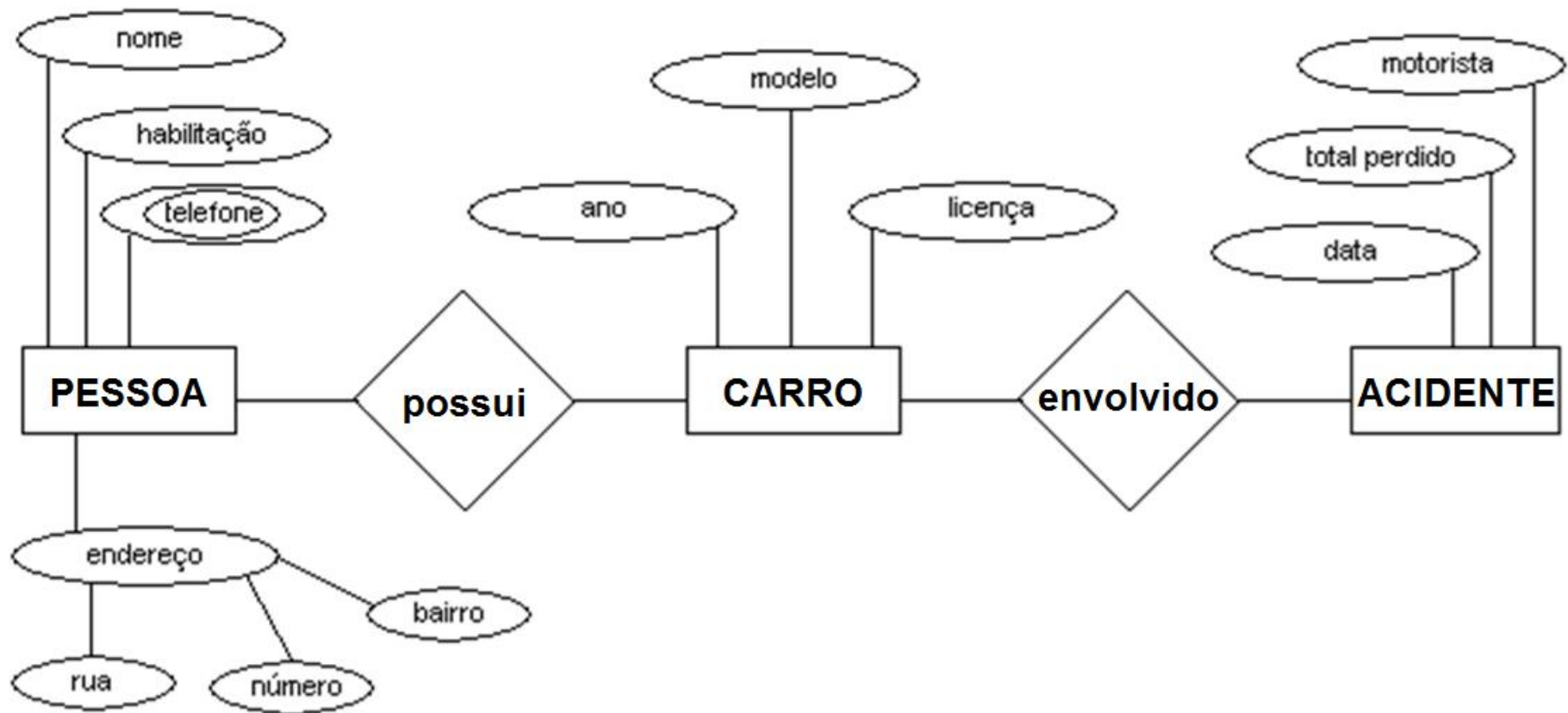
Podem ser definidas as operações de inserção, remoção e atualização das tuplas de uma determinada relação. Todas estas operações devem respeitar as restrições de integridade apresentadas anteriormente.

Quando algumas destas restrições for violar uma das operações citadas anteriormente, o SGBD poderá executar algumas ações para manter as restrições ou para notificar a violação ocorrida.



# Exercício de Fixação

- 1) Projete o BD relacional correspondente ao Diagrama de Entidade-Relacionamento (DE-R), completando no diagrama, as informações referentes aos relacionamentos que envolvem a lógica que vai ser implementada.



# Exercício de Fixação (continuação)

... continuação do exercício 1

Faça a especificação do projeto (conte uma história do problema) e descreva as restrições de integridade que serão necessárias no projeto que estará sendo proposto, identificando também as chaves envolvidas.

Efetue ainda o correto mapeamento representando todos os esquemas necessários para atender essa proposta de projeto.



# Modelo de Dados Relacional

## Álgebra Relacional

Consiste em operadores que realização ações sobre as relações. Os operadores da álgebra relacional podem ser divididos em duas categorias:

- **Operadores de Conjuntos**: operadores típicos definidos pela álgebra de conjunto, como: **união**, **interseção**, **diferença** e **produto cartesiano**.
- **Operadores Relacionais**: são operadores especiais definidos especialmente para manipulação de tuplas, tais como a **seleção**, **junção** e **projeção**.



# Controle de Consistência

No Modelo Relacional, cada esquema de uma relação possui um número de atributos (**grau da relação**), enquanto que o esquema de um BD relacional consiste de uma série de esquemas de relações (tabelas).

A avaliação de um mapeamento proposto por um projetista, ou resultante de um ME-R, deve respeitar algumas características que o conceitue como um mapeamento eficiente.

A indicação de que um esquema relacional está eficiente é realizada quando as relações produzidas representam os dados de forma consistente e com o mínimo de redundância.



# Controle de Consistência

No intuito de orientar o projetista na definição e avaliação de um esquema relacional é aplicado um conjunto de critérios informais que o auxiliem, sendo alguns destes critérios apresentados a seguir:

## A) Semântica dos atributos de uma relação

Sempre que um conjunto de atributos é agrupado em uma relação, um determinado significado é associado a eles. Este significado (ou semântica) identifica como devem ser interpretados os atributos pertencentes a cada tupla da relação.

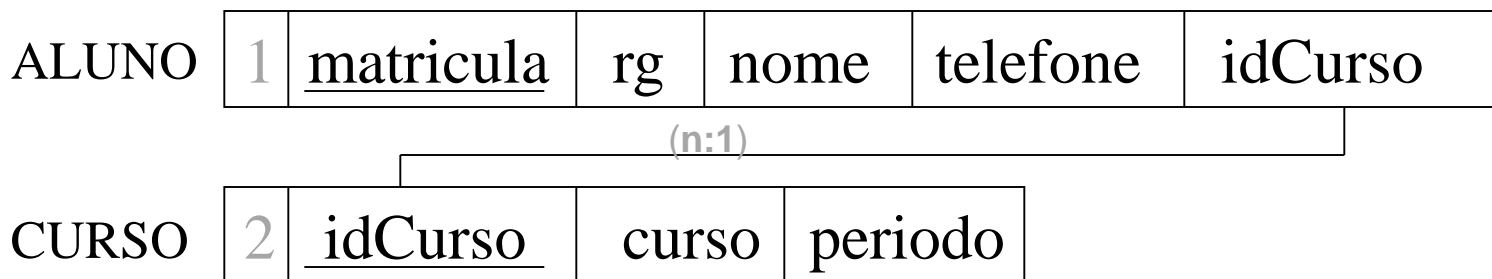
Por exemplo a relação ALUNO\_CURSO representada a seguir:

ALUNO\_CURSO (nome,rg,matrícula,telefone,curso,período,id\_curso)



# Controle de Consistência

O esquema anterior (ALUNO\_CURSO), apesar de representar claramente o seu significado (os alunos que fazem um curso), não é aconselhado, pois possui atributos de duas entidades. A representação mais correta, por meio de esquemas seria:



# Controle de Consistência

## B) Redundância de valores

Sempre deve-se **evitar** a definição de dados em mais que um local (esquema) no BD, ou seja, a duplicação (ou **redundância**) das dados no BD.

Como exemplo, considere a relação ALUNO\_CURSO contendo os alunos e os seus respectivos cursos. Se mais de um aluno estudar em um mesmo curso, os valores dos atributos do curso deverão aparecer duplicados para cada tupla de aluno cadastrado.

Com isso podem ocorrer alguns problemas (**anomalias**) nas operações de atualização, tais como:



# Controle de Consistência

## ANOMALIAS

**Inserção**: para colocar um novo aluno deve-se preencher os atributos do curso que ele faz, ou colocar atributos nulos, quando ele não fizer nenhum curso (for um pesquisador). Para possuir um curso sem alunos, torna-se necessário colocar valores nulos para os atributos do aluno, o que significa que a chave da tupla será nula, violando a restrição de integridade de chave.

**Remoção**: quando for removido o último aluno de um determinado curso, todos os dados referentes a este curso serão perdidos (não existirá o curso na base de dados).

**Atualização**: a alteração do atributo de período de um curso deverá ser feita em todas as tuplas da relação, onde um aluno faça este curso, caso contrário a base de dados se tornará inconsistente.

# Controle de Consistência

## C) Valores nulos em tuplas

Um atributo que possua valor nulo permite **diferentes interpretações**, tais como:

- 1- atributo não se aplica para esta tupla
- 2- valor do atributo para esta tupla é desconhecido
- 3- valor é conhecido, mas encontra-se ausente (não inserido)

**Sempre procure evitar atributos que possam ser nulos.**  
Quando for inevitável, verifique se estas tuplas acontecem em pequeno número na relação.



# Controle de Consistência

Uma das importâncias fundamentais na definição de esquemas relacionais é manter a consistência dos dados em todo a base de dados.

O **controle de consistência** dos dados pode ser exercido em três níveis:

- pelo gerenciador (SGBD);
- pelo aplicativo;
- pela própria construção de um sistema.

De forma geral, o controle de consistência obtido pela própria construção do sistema é mais eficiente, pois normalmente não implica em perda de desempenho do SGBD durante a sua execução.



# Controle de Consistência

O controle por meio da própria construção do sistema é obtido, no Modelo Relacional, construindo-se as relações segundo regras que garantem a manutenção de certas propriedades (**respeito as regras de negócio** existentes no problema existente).

As relações que atendem a determinadas regras diz-se estarem em uma determinada “Forma Normal”.

O processo de **Normalização** pode ser considerado como a aplicação de uma série de regras, que constituem as Formas Normais, sobre cada uma das relações (tabelas) do projeto proposto, lhe assegurando um nível de implementação para o **controle da redundância** e da **consistência** da base de dados.



# Controle de Consistência

O processo de Normalização irá provocar a decomposição de esquemas insatisfatórios, de algumas relações, em novas relações que tenham cuidados com as situações já apresentadas para **evitar a inconsistência** e a **redundância**.

Este processo ainda permite ao projetista da base de dados controlar o **quanto de consistência seria garantida** pela maneira de construção do sistema (estrutura), e quanto deve ser de responsabilidade dos aplicativos e/ou do SGBD.

Porém, deve-se analisar até que ponto a Normalização deve chegar, pois **normalizar demais** pode diminuir a eficiência dos aplicativos, e **de menos**, abre a possibilidade das inconsistências.



# Dependência Funcional

A Dependência Funcional está baseada no reconhecimento de que os valores de alguns atributos contribuem na identificação de outros.

Uma dependência funcional significa que ao se conhecer o valor de um atributo, então pode-se sempre determinar o valor de outro. A notação usada na teoria relacional é:

$A \rightarrow B$  onde se lê que “A **determina funcionalmente** B”,  
ou que “B **depende de** A”.

Exemplo: conhecendo a **matrícula** de um aluno, então posso determinar o seu **nome**. Sabendo o **código** de um produto, então posso determinar o seu **preço**.





# Dependência Funcional

Uma **Dependência Multivalorada** significa que ao se conhecer o valor de um atributo, então pode-se sempre determinar os valores de um conjunto de outros atributos. A notação usada na teoria relacional é:

$A \twoheadrightarrow B$  onde se lê “A **determina muitos** B’s”,  
ou que “B **é multidependente de** A”

Exemplo: conhecendo um curso posso determinar a lista de seus alunos.

Uma das maneiras de controlar a consistência é por meio das *Dependências Funcionais* existentes entre os atributos armazenados.



# Normalização

As Formas Normais, processo chamado de Normalização, são identificadas por meio de números, onde serão usadas a primeira, segunda e terceira forma normal no decorrer da disciplina.

No Modelo Relacional a normalização principal é a primeira, pois é considerada parte da própria definição de uma relação.

## PRIMEIRA - (1 FN)

Uma relação se encontra na Primeira Forma Normal se todos os domínios de atributos possuem apenas valores atômicos (simples e indivisíveis). Com isso a relação é construída sem atributos compostos e multivalorados em suas tuplas.

Exemplo:

CURSO(curso, periodo, idCurso, {nome, matricula} ) – não 1FN

# Normalização

Para levar uma relação a atender as exigências da primeira forma normal (normalizar) tem-se que realizar a aplicação de algumas “**regras**” para as seguintes situações:

1- Cada um dos **atributos compostos** (ou não atômicos) devem ser “divididos” em seus componentes.

2- Para os **atributos multivalorados** tem-se duas alternativas:

**A) Pequena quantidade de valores possíveis**, substitui-se o atributo multivalorado por um conjunto de atributos de mesmo domínio, cada um monovalorado representando uma ocorrência do valor;

**B) Quantidade de possíveis valores desconhecida**, grande ou muito variada, retira-se da relação o atributo multivalorado e cria-se uma nova relação que tem o mesmo conjunto de atributos chave, mais o atributo multivalorado também como chave, mas tomado como monovalorado.

# Normalização

A 1 FN é uma das maneiras de controlar a consistência por meio da própria estrutura do sistema, sendo também fundamental para a conceituação do sistema.

## SEGUNDA - (2 FN)

Uma relação esta na Segunda Forma Normal quando ela está na 1 FN e todos os atributos que não participam da chave primária são dependentes diretos de toda a chave primária.

Para se normalizar uma relação para a 2 FN as “**regras**” seguintes devem ser seguidas.



# Normalização

A normalização para um relação estar na 2 FN deve:

- 1- Verificar os grupos de atributos que dependem da mesma parte da chave primária.
- 2- Retirar da relação todos os atributos de um desses grupos (separá-los em relações diferentes).
- 3- Criar uma nova relação contendo esse grupo como atributo não chave, e os atributos que determinam esse grupo como chave primária de cada nova relação.
- 4- Repetir o procedimento para cada grupo, até que cada relação tenha somente atributos que dependam da chave primária definida para cada relação.



# Normalização

A normalização com a 2 FN evita a inconsistência devido a duplicidade de informações, além da perda de dados em operações de remoção ou de alterações (**anomalias**) na relação.

Note que a normalização de relações é realizada, na grande maioria das vezes, decompondo-se uma relação em duas ou mais relações que protejam a consistência e controlem a redundância diminuindo-a.

Exemplo: não está na 2 FN

**EMPREGADO\_PROJETO** (matrEmpr, codProj, horas,  
nomeEmpr, nomeProj, localProj)



# Normalização

continuação do exemplo...

**EMPREGADO\_PROJETO** (matrEmpr, codProj, horas,  
nomeEmpr, nomeProj, localProj)

- Dependências funcionais parciais em relação à chave primária:  
matrEmpr  $\rightarrow$  nomeEmpr  
codProj  $\rightarrow$  nomeProj, localProj
- Dependências funcionais totais em relação à chave primária:  
(matrEmpr, codProj)  $\rightarrow$  horas

**Esquema que atende a 2 FN:**

EMPREGADO (matrEmpr, nomeEmpr)

PROJETO (codProj, nomeProj, localProj)

EMPREGADO\_PROJETO (matrEmpr, codProj,  
horas)

# Normalização

## TERCEIRA - (3 FN)

Uma tabela está na Terceira Forma Normal se estiver na 2 FN e não possuir dependências transitivas (ou indireta). Uma dependência transitiva ocorre quando

$$A \rightarrow B \quad \text{e} \quad B \rightarrow C \quad \text{então} \quad A \rightarrow C.$$

Em outras palavras, deve-se evitar que qualquer atributo não chave seja dependente funcional de outro atributo não chave.

Exemplo:    não está na 3 FN

**EMPREGADO\_DEPTO** (matrEmpr, nomeEmpr, dtNasc,  
codDep, nomeDep)





# Normalização

continuação do exemplo...

**EMPREGADO\_DEPTO** (matrEmpr, nomeEmpr, dtNasc,  
codDep, nomeDep)

Dependências transitivas (indiretas):

matrEmpr  $\rightarrow$  codDep  $\rightarrow$  nomeDep

**Esquema que atende a 3 FN:**

EMPREGADO(matrEmpr, nomeEmpr, dtNasc, **codDep**)

DEPTO (codDep, nomeDep)

Restrição **EMPREGADO\_DEPTO\_FK** de chave estrangeira (**codDep**)  
que Referencia DEPTO(**codDep**)

→ Da mesma forma que a 2 FN, na normalização para a 3 FN evita a inconsistência devido a duplicidade de dados, além da perda de dados em operações de remoção ou de alterações na relação.

# Normalização

Um outro exemplo no processo de normalização envolvendo, diretamente, a dependência transitiva seria:

**ENFERMEIRO** (codigoFuncional, nome, dtFormacao, dtNascimento, sexo, idade)

Exemplo: não está na 3 FN

**ENFERMEIRO** (codigoFuncional, nome, dtFormacao, dtNascimento, sexo, ~~idade~~)

O atributo **idade** tem dependência direta em relação ao atributo **dtNascimento** e poderia ser responsável pela inconsistência na base de dados se um enfermeiro tivesse nascido a exatos 20 anos atrás, mas a idade armazenada fosse 51.

Qual idade em anos completos seria a correta para o enfermeiro?

# Normalização

RELAÇÃO (ou Tabela)  
**NÃO** NORMALIZADA

Normalizar Projeto Lógico  
Banco de Dados

Remoção dos Atributos  
Multivalorados e Compostos

1ª FN

Remoção das Dependências PARCIAIS  
(atributos dependentes da PK)

2ª FN

Remoção das Dependências TRANSITIVAS  
(atributos dependentes da PK)

3ª FN



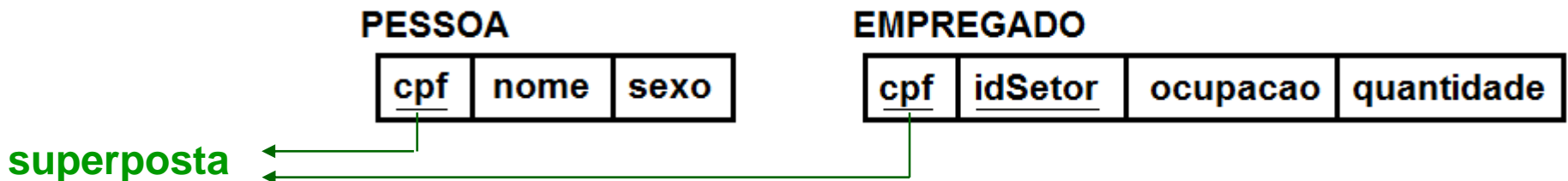
# Normalização

## BOYCE CODD - (FNBC)

A Forma Normal de Boyce Codd (FNBC) consiste na 3FN mais “forte” ou rigorosa, a fim de evitar algumas **anomalias** na base de dados.

Na normalização a FNBC deve ser aplicada sobre as tabelas que já estão respeitando a 3 FN e possuam mais de uma chave candidata (**primária também é candidata antes de ser PK**), sendo pelo menos uma dessas chaves composta e com **superposição**.

**CHAVE SUPERPOSTA**: duas chaves são superpostas quando pelo menos uma delas é composta e existe ao menos um atributo em comum entre elas.



# Normalização

Assim, para uma tabela estar na FNBC ela deverá estar na **3FN** e sem possuir nenhum outro atributo que seja determinado por outro(s) atributo(s) que **NÃO** é (sejam) uma chave candidata da tabela.

Se isso acontecer a solução será conduzir a tabela para a **3FN** e levar esses atributos para uma outra tabela, aplicando o conceito de **decomposição sem perdas**, por exemplo:

CLIENTE	AGENCIA	GERENTE
00185	Norte	Paulo
01018	Norte	Ana
05270	Norte	Paulo
04857	Norte	Ana
00854	Central	Maria
05879	Central	Carlos

- A tabela está na 3 FN;
- Note que existe outra chave candidata e que ela pode ser superposta à chave primária;
- **CLIENTE** estará nas duas chaves candidatas compostas, efetuando a supersposição (**sem FNBC**).



CLIENTE	AGENCIA
---------	---------

CLIENTE	GERENTE
---------	---------

→ **Anomalia na atualização**, pois a agência se repete para o mesmo gerente

# Normalização

Chave primária { ~~CLIENTE~~, AGENCIA } → { GERENTE }

Chave candidata { ~~CLIENTE~~, ~~GERENTE~~ } → { AGENCIA }

Dessa forma, será necessário decompor a tabela para que ela esteja na FNBC, no qual o atributo determinado por um campo que **NÃO** é chave candidata vai para uma outra tabela.

- Note que há outro **determinante** que **NÃO** é chave candidata, sendo superposta e fazendo com que esta solução **respeite a FNBC**.

GERENTE	AGENCIA
Paulo	Norte
Ana	Norte
Maria	Central
Carlos	Central

CLIENTE	GERENTE
00185	Paulo
01018	Ana
05270	Paulo
04857	Ana
00854	Maria
05879	Carlos

CLIENTE	AGENCIA	GERENTE
00185	Norte	Paulo
01018	Norte	Ana
05270	Norte	Paulo
04857	Norte	Ana
00854	Central	Maria
05879	Central	Carlos

- Tabelas na **3FN** que **NÃO** possuem superposição de chaves já estão ou respeitam a **FNBC**.

# Normalização

## QUARTA - (4 FN)

A Quarta Forma Normal (4 FN) faz uso de dependências funcionais **multivaloradas**.

Considere a relação **DEPTO\_PROJETO\_PECAS** a seguir, onde:

Deppto  $\twoheadrightarrow$  Proj

e

Deppto  $\twoheadrightarrow$  Peca

Deppto	Proj	Peca
D1	J1	P1
D1	J1	P2
D1	J2	P1
D1	J2	P2
D2	J3	P2
D2	J3	P4
D2	J4	P2
D2	J4	P4
D2	J5	P2
D2	J5	P4
D3	J2	P5
D3	J2	P6

→ A dependência multivalorada também causa anomalia de atualização.

# Normalização

De forma simples, se diz que uma tabela está em 4 FN quando ela está na FNBC e não possui duas ou mais Dependências Multivaloradas (DM) independentes (não considerando as Dependências Funcionais (DF) que são casos também de DM) de um mesmo **multideterminante**.

Observe que para adicionar (inserir) uma **Peça** para um **Departamento** se deve criar uma nova linha (tupla). Remover uma **Peça** ou um **Projeto** de uma linha pode resultar em perda de informação. A atualização de uma **Peça** ou de um **Projeto** pode requerer que muitas linhas sejam atualizadas (**possíveis anomalias**).

Por isso a solução é **decompor** esta relação em duas relações, uma com **DEPTO\_PROJ** e outra **DEPTO\_PECA**.




# Normalização


**Deppto**  $\rightarrow\rightarrow$  **Proj**      e      **Deppto**  $\rightarrow\rightarrow$  **Peca**

**DEPTO\_PROJETO\_PECAS**

Deppto	Proj	Peca
D1	J1	P1
D1	J1	P2
D1	J2	P1
D1	J2	P2
D2	J3	P2
D2	J3	P4
D2	J4	P2
D2	J4	P4
D2	J5	P2
D2	J5	P4
D3	J2	P5
D3	J2	P6



Deppto	Proj
D1	J1
D1	J2
D2	J3
D2	J4
D2	J5
D3	J2



Deppto	Peca
D1	P1
D1	P2
D2	P2
D2	P4
D3	P5
D3	P6



# Normalização

Alguns autores **confundem** a 4 FN com relacionamentos *binários* do tipo “*muitos para muitos*” (**n:m**) por causa da aplicação da Dependência Multivalorada.

No entanto, esta confusão não deveria acontecer porque a dependência multivalorada ocorre “**dentro**” de uma tabela, ao contrário da cardinalidade que acontece sobre os relacionamentos (entre relações).

A 4 FN **só é necessária** se a tabela analisada possuir:

- Um multideterminante que aponte para mais de um multidependente;
- Independência entre esses multidependentes, o que seria como se uma tabela contivesse duas outras tabelas (ou mais) que não possuísem relação entre si.

# Normalização

## Realizando a Normalização:

- A normalização é aplicada em uma relação por vez;
- Uma relação vai sendo dividida criando outras relações;
- Inicia-se das formas normais menos rígidas (1 FN→2 FN→...) até chegar em uma normalização considerada satisfatória.

A decisão de normalizar ou não uma relação é um compromisso entre garantir a eliminação de inconsistências na base de dados e a eficiência de acesso. A normalização para formas apoiadas em dependências funcionais evita inconsistências, usando para isso a própria construção da base de dados.



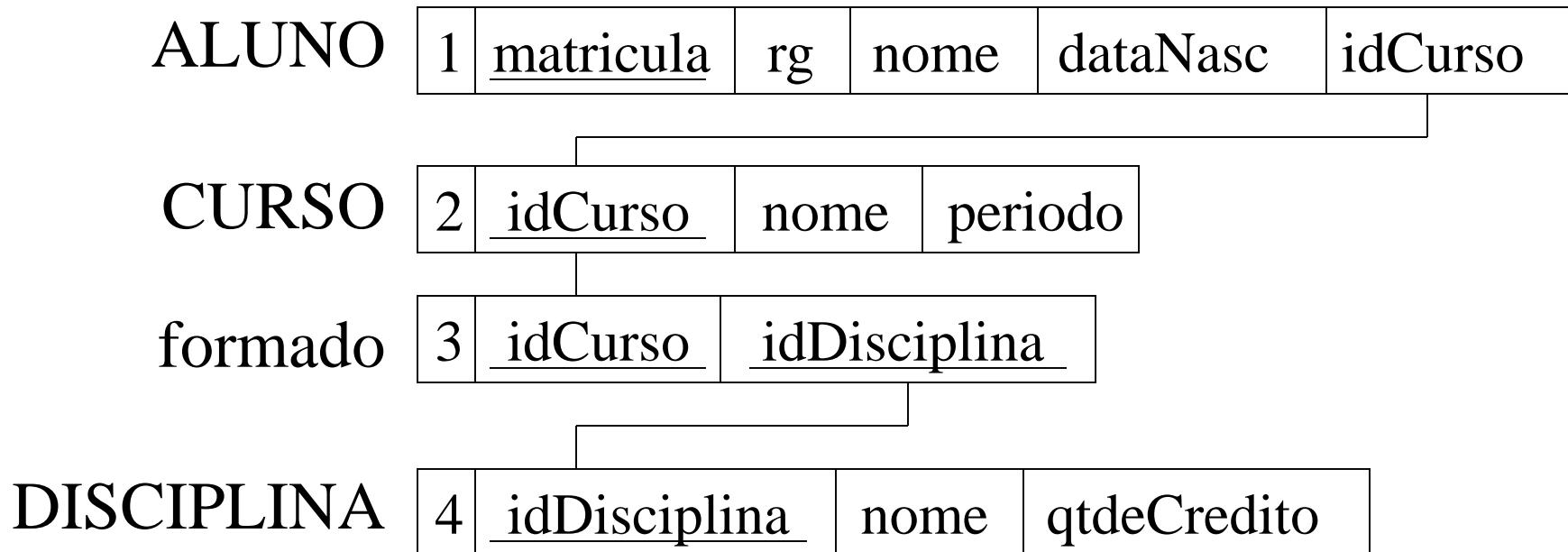
# Normalização

- Existem outras formas normais, porém a **quarta**, **quinta**, **sexta** e outras não tem o mesmo foco de controle da redundância como as três primeiras;
- O foco das formas normais chamadas superiores (a partir da quarta) está no custo e no desempenho da base de dados;
- Assim, se a inconsistência **NÃO** for um fator relevante e fundamental para a base de dados que esteja sendo projetada pode-se abrir mão da normalização.



# Exercícios de Fixação

2) De acordo com o Diagrama de Esquemas a seguir faça as descrições da relações que serão implementadas neste banco de dados, descrevendo corretamente as restrições de integridade de entidade, referencial e colocando como comentário ("--") os valores para a integridade semântica quando ela existir sobre algum atributo.



# Exercícios de Fixação

3) A Farmácia **Dodói** fez a aquisição de alguns equipamentos de informática para conseguir gerenciar melhor os dados que são necessários ao seu funcionamento. Para o desenvolvimento de um banco de dados você foi contratado, e a especificação do problema foi sintetizada a seguir.

Os produtos comercializados nesta farmácia são adquiridos de fornecedores únicos para cada produto, apesar de serem diversos os fornecedores (ou fabricantes). Cada fabricante é identificado por meio de seu nome, CGC, endereço completo do distribuidor e telefones de distribuição.

Os produtos comercializados possuem código de controle, nome comercial, tipo de embalagem, quantidade e preço unitário. A farmácia comercializa dois tipos de produtos: os medicamentos que apresentam uma fórmula e uma tarja e os itens de perfumaria que tem tipo e fragrância.



# Exercícios de Fixação (continuação do 3)

A venda destes produtos também deverá ser acompanhada pelo banco de dados, onde cada venda deverá armazenar a data de compra, número da nota, nome do cliente, quantidade, preço total e o imposto a ser recolhido.

Somente para os medicamentos é necessário a apresentação da receita médica que terá o CRM do médico, a identificação do medicamento e a data da emissão da mesma.

Baseado na especificação anterior elabore o Diagrama Entidade-Relacionamento identificando cada item necessário ao projeto, além das cardinalidades e participações existentes para este diagrama.

De acordo com o projeto elaborado realize o mapeamento deixando os esquemas dentro da terceira forma normal no Diagrama de Esquema e descreva as principais restrições que devem ser observadas.



# Referência de Criação e Apoio ao Estudo

## Material para Consulta e Apoio ao Conteúdo

- ELMASRI, R. e Navathe, S. B., Fundamentals of Database Systems, Addison-Wesley, 3rd edition, 2000
  - Capítulo 7, 14
- SILBERSCHATZ, A. & Korth, H. F., Sistemas de Banco de Dados.
  - Capítulo 3, 7
- HEUSER, C. A., Projeto de Banco Dados, 2001.
  - Capítulo 4, 5, 6
- Universidade de Brasília (UnB Gama)
  - <https://sae.unb.br/cae/conteudo/unbfga>  
(escolha no menu superior **Sistemas Banco dados 1**  
seguida da opção **Mod. Entid. Relacionamento** )

