



Fundamentos de Business Intelligence e Análise de Dados

BLOCO: B.I. E ANÁLISE DE DADOS

PROF. RODRIGO EIRAS, M.SC.

[ETAPA 3] AULA 2 – MODELAGEM RELACIONAL E NORMALIZAÇÃO



Na aula anterior...

- Abstração
- Modelos e modelagem
- Etapas da modelagem relacional
- Finalidade da modelagem a nível conceitual, lógico e físico
- Entidades, atributos e relacionamentos
- Exercício de modelagem



Agenda

- Correção do exercício extra
- Mais sobre relacionamentos na modelagem relacional
- Normalização

A partir do cenário abaixo, identifique as entidades, os atributos e os relacionamentos.

Cenário: Sistema de Vendas Simples

Uma firma vende produtos de limpeza, e deseja melhor controlar os produtos que vende, seus clientes e os pedidos. Cada produto é caracterizado por um código, nome do produto, categoria (ex. detergente, sabão em pó, sabonete, etc.), e seu preço.

A categoria é uma classificação criada pela própria firma. A firma possui informações sobre todos seus clientes. Cada cliente é identificado por um código, nome, endereço, telefone, status ("bom", "médio", "ruim") e o seu limite de crédito.

Guarda-se igualmente a informação dos pedidos feitos pelos clientes. Cada pedido possui um número e guarda-se a data de elaboração do pedido. Cada pedido pode envolver de um a vários produtos, e para cada produto, indica-se a quantidade deste pedido.

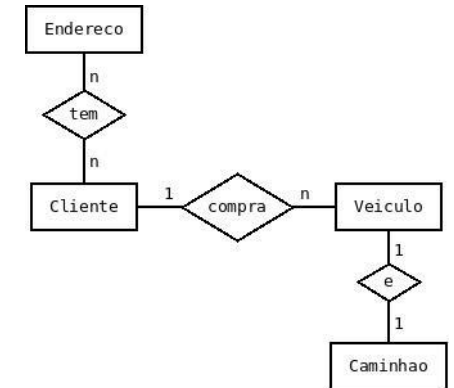
Exercício Extra de Modelagem Conceitual – Correção



Modelagem Relacional

Sobre relacionamentos entre entidades, devemos observar alguns aspectos importantes que sinalizam erros na modelagem:

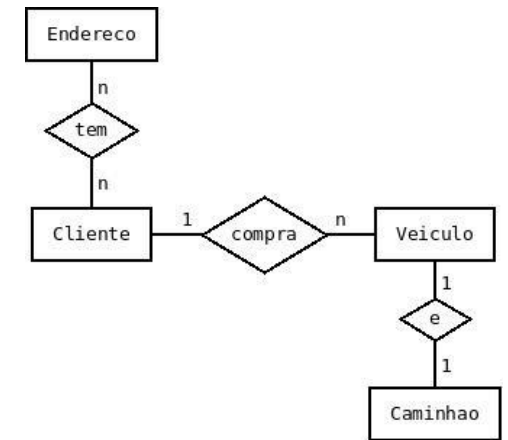
- Quando "sobram" entidades sem relacionamentos;
- Quando ocorrem "ciclos fechados" de relacionamentos;
 - Exemplo: Usuário relaciona-se com Empréstimo que relaciona-se com Livro que relaciona-se com Usuário que relaciona-se com Empréstimo, etc...
- Entidades com muitos atributos relacionando-se com entidades com apenas alguns atributos;
- Muitas entidades relacionando-se à uma mesma entidade;



Relacionamento entre entidades

Determinados os relacionamentos, temos que verificar o número de referências de uma entidade em outra, ou seja, agora vamos verificar a CARDINALIDADE dos relacionamentos.

- Relacionamento Um-Para-Um (1:1) - Uma instância da entidade A relaciona-se a uma instância da entidade B
- Relacionamento Um-Para-Vários (1:N) - Uma instância da entidade A relaciona-se a várias instâncias da entidade B
- Relacionamento Vários-Para-vários (N:M) - Várias instâncias da entidade A relacionam-se a várias instâncias da entidade B

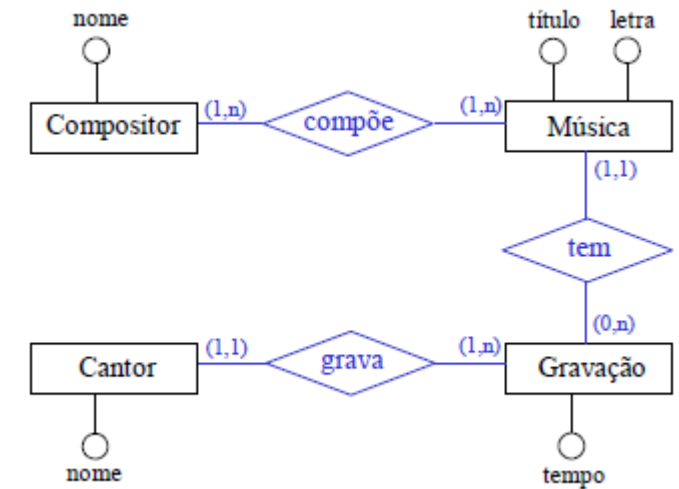


Cardinalidade Máxima

- Relacionamento Um-Para-Vários (0,1:1,N) -Nenhuma ou uma instância da entidade A relaciona-se á uma ou várias instancias da entidade B.

- Exemplo de relacionamento Um-Para-Vários com cardinalidade mínima:
Usuário - Empréstimo (1,1:0,N)

- Um usuário pode estar relacionado a nenhum ou a vários empréstimos.
 - Um empréstimo deve estar relacionado a somente um usuário.
- Conclusão:
 - Pode ser que um usuário nunca faça empréstimos, assim como pode haver usuário que faça vários empréstimos, porém um empréstimo obrigatoriamente tem que ser feito por um único usuário.



Cardinalidade Mínima

Exemplo de relacionamento Vários-Para-Vários com cardinalidade mínima:

- Empréstimo - Livro (0,N:1,N)
- Um empréstimo pode estar relacionado a um ou a vários livros.
- Um livro pode estar relacionado a nenhum ou a vários empréstimos.

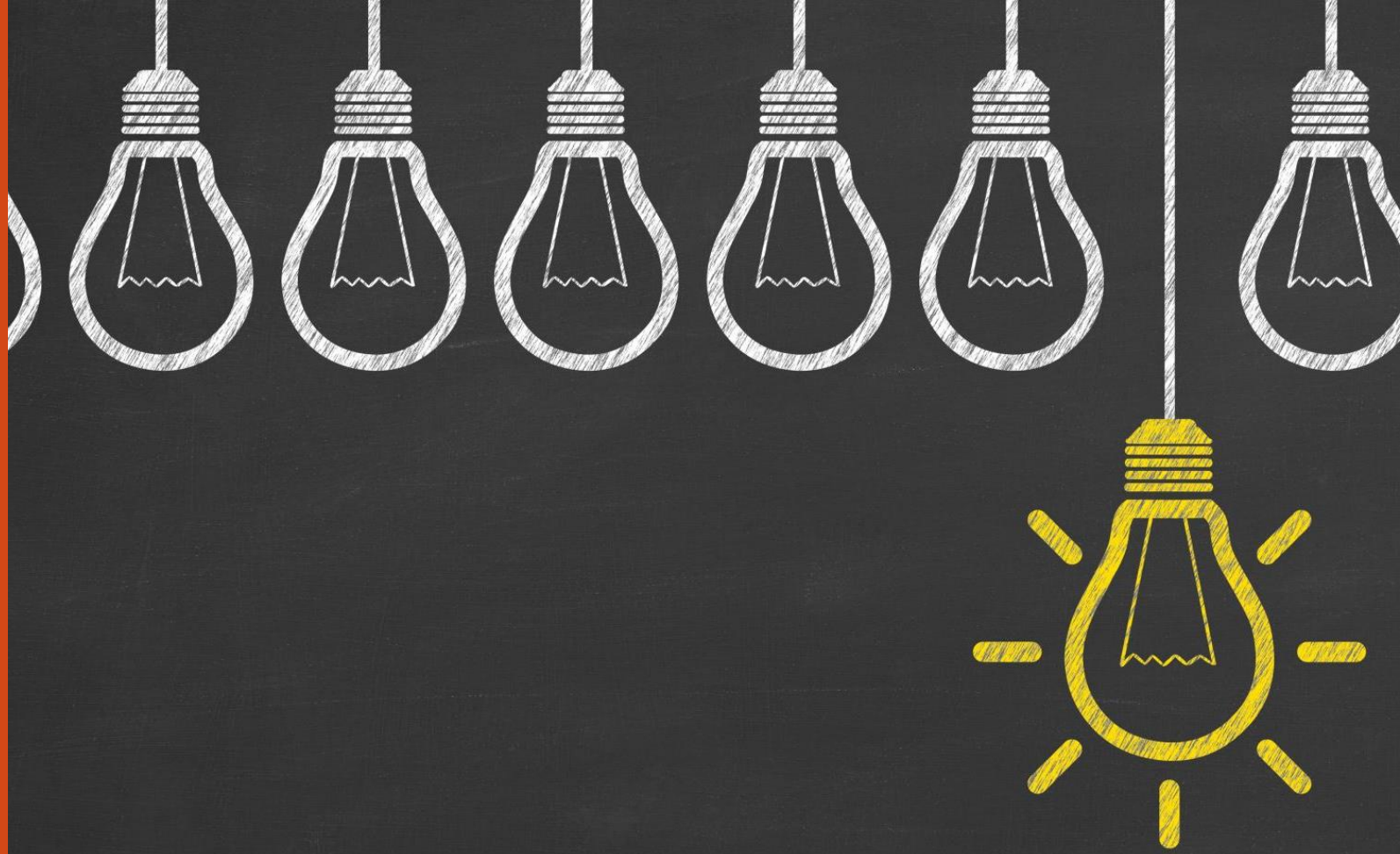
Conclusão:

- Pode ser que um livro nunca seja emprestado, assim como pode haver livros que tenham sido emprestado várias vezes, porém um empréstimo deve conter pelo menos um livro ou pode conter vários.

A Cardinalidade Mínima pode ser incluída no Modelo Lógico, mas é pouco utilizada por ser, muitas vezes, redundante e óbvia, mas muito é útil no que refere à expor a clareza dos relacionamentos entre entidades.

Cardinalidade Mínima (+)

MER – Metodologia



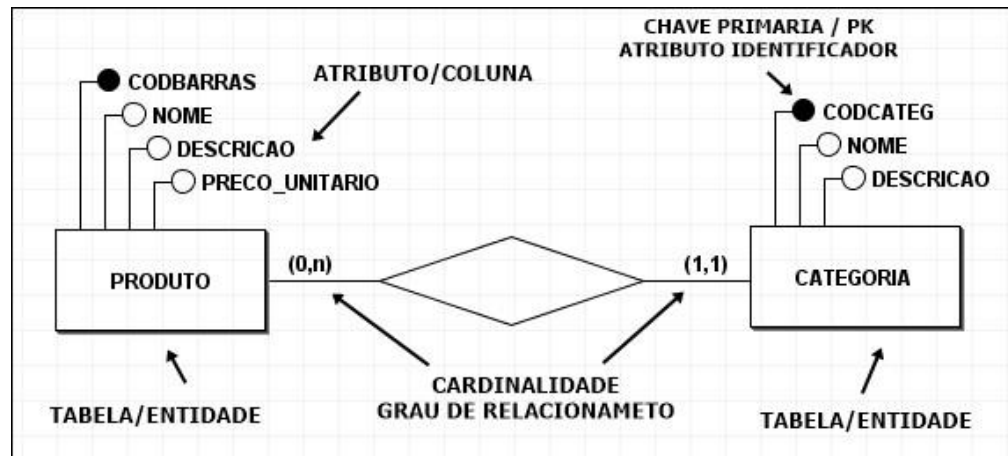
Construir o modelo conceptual de dados:

- Identificar todas as entidades e todos os relacionamentos importantes para a situação a tratar;
- Construir o diagrama de Entidade-Relacionamento (DER);
- Identificar todos os atributos relevantes e associa-los a uma das entidades preliminares já definidas (ou a relacionamentos existentes entre estas);

Construção do modelo de dados relacional pelo método de E-R – Primeira Etapa

Derivar o modelo de dados relacional:

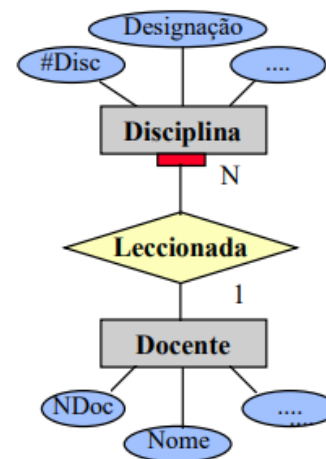
- Identificar as chaves do conjunto de entidades preliminares;
- Aplicar as regras de derivação do modelo relacional;
- Verificar o resultado aplicando a teoria da normalização.



Construção do modelo de dados relacional pelo método de E-R – Segunda Etapa

Principais factores com influência nas regras de derivação:

- Grau de relacionamento (1:1, 1:N, N:M);
- Tipo de participação das entidades no relacionamento (obrigatória/opcional);
- Tipo de relacionamento (binário, n_ário, recursivo, etc.);
- Tipo de entidade.



Disciplina

<u>#Disc</u>	Designação	NDoc

Docente

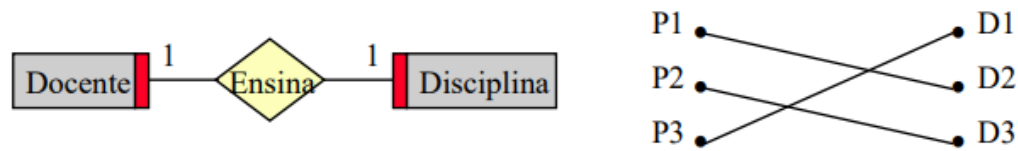
<u>NDoc</u>	Nome

Regras de derivação do modelo relacional

Relacionamento binário de grau 1:1

Caso 1 (1:1) - participação obrigatória das duas entidades:

- Todos os docentes têm de leccionar uma só disciplina;
- Cada disciplina tem de ser assegurada por um docente.



Docente (Ndoc, Nome, Tel, #Disc, Prereq)

NDoc	Nome	Tel	#Disc	Prereq
1001	Couto	721334	Inf2	Inf1
1662	Nunes	776188	SOC	LP
977	Peixoto	722876	Inf1	Nenhum

Regra 1

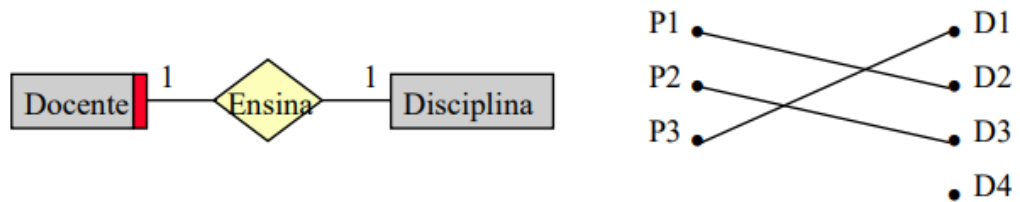
Relacionamento binário de grau 1:1 e participação obrigatória de ambas as entidades.

- É apenas necessário uma entidade;
- A chave primária dessa relação pode ser a chave primária de qualquer das entidades.

Relacionamento binário de grau 1:1

Caso 2 (1:1) - participação obrigatória de apenas uma das entidades:

- Todos os docentes têm de leccionar uma só disciplina.



NDoc	Nome	Tel	#Disc	Prereq
1001	Couto	721334	Inf2	Inf1
?	?	?	IG2	IG1

Relacionamento binário de grau 1:1

Regra 2

Relacionamento binário de grau 1:1 e participação obrigatória de apenas uma das entidades.

- São necessárias duas entidades;
- A chave primária de cada entidade serve de chave primária na entidade correspondente;
- A chave primária da entidade com participação não obrigatória tem de ser usada como atributo na entidade correspondente à entidade cuja participação é obrigatória.

NDoc	Nome	Tel	#Disc	Prereq
1001	Couto	721334	Inf2	Inf1
?	?	?	IG2	IG1



Docente (Ndoc, Nome, Tel, #Disc)

NDoc	Nome	Tel	#Disc
1001	Couto	721334	Inf2

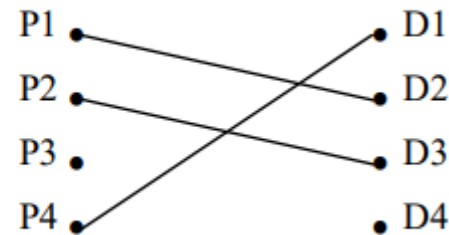
Disciplina (#Disc, Prereq)

#Disc	Prereq
Inf2	Inf1
IG2	IG1

Relacionamento binário de grau 1:1

Caso 3 (1:1) - sem participação obrigatória em ambas as entidades:

- Os docentes leccionam uma só disciplina, se não estiverem dispensados do serviço docente;
- Cada disciplina é assegurada por um docente, excepto se for opcional e se o número de inscrições for inferior a 15 alunos.



NDoc	Nome	Tel	#Disc	Prereq
1001	Couto	721334	Inf2	Inf1
1662	Nunes	776188	SOC	LP
?	?	?	IG2	IG1
1056	Martins	734976	?	?

Com a utilização de uma só entidade, surgem valores nulos quer para as disciplinas que ainda não têm docente, quer para os docentes que não leccionam nenhuma disciplina.

Relacionamento binário de grau 1:1

Docente (Ndoc, Nome, Tel, #Disc)

NDoc	Nome	Tel	#Disc
1001	Couto	721334	Inf2
1662	Nunes	776188	SOC
1056	Martins	734976	?

Disciplina (#Disc, Prereq, NDoc)

#Disc	Prereq	NDoc
Inf2	Inf1	1001
SOC	LP	1662
IG2	IG1	?

A subdivisão da entidade em duas, segundo solução análoga à regra 2, também origina valores nulos.

Relacionamento binário de grau 1:1

Docente

(Ndoc, Nome, Tel)

NDoc	Nome	Tel
1001	Couto	721334
1662	Nunes	776188
1056	Martins	734976

Disciplina

(#Disc, Prereq)

#Disc	Prereq
Inf2	Inf1
SOC	LP
IG2	IG1

Leccionar

(Ndoc, #Disc)

NDoc	#Disc
1001	Inf2
1662	SOC

Obs.: A chave da entidade Leccionar pode ser NDoc ou #Disc.

Regra 3

Relacionamento binário de grau 1:1 e participação não obrigatória em ambas as entidades.

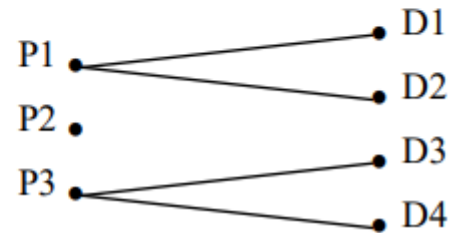
- São necessárias três entidades, uma para cada entidade e a terceira para o relacionamento;
- A chave primária de cada entidade serve de chave primária na entidade correspondente;
- A entidade correspondente ao relacionamento terá entre os seus atributos as chaves primárias das duas entidades.

Relacionamento binário de grau 1:N

Caso 4 (1:N) - participação obrigatória do lado N:

(a participação obrigatória no lado 1 não afecta resultado)

- Os docentes podem leccionar várias disciplinas;
- Cada disciplina têm de ser assegurada por um só docente.



NDoc	Nome	Tel	#Disc	Prereq
1662	Nunes	776188	SOC	LP
1662	Nunes	776188	SDP	SOC
1056	Martins	734976	?	?

Relacionamento binário de grau 1:N

Regra 4

Relacionamento binário de grau 1:N e participação obrigatória do lado N.

- São necessárias duas entidades;
- A chave primária de cada entidade serve de chave primária na entidade correspondente;
- A chave primária da entidade do lado 1 tem de ser usada como atributo na entidade correspondente à entidade do lado N.

NDoc	Nome	Tel	#Disc	Prereq
1662	Nunes	776188	SOC	LP
1662	Nunes	776188	SDP	SOC
1056	Martins	734976	?	?



Docente (Ndoc, Nome, Tel)

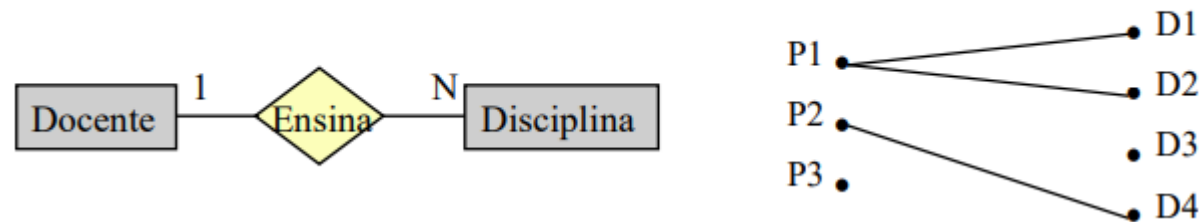
NDoc	Nome	Tel
1662	Nunes	776188
1056	Martins	734976

Disciplina (#Disc, Prereq, **NDoc**)

#Disc	Prereq	NDoc
SOC	LP	1662
SDP	SOC	1662

Relacionamento binário de grau 1:N

Caso 5 (1:N) - participação não obrigatória do lado N.



NDoc	Nome	Tel	#Disc	Prereq
1662	Nunes	776188	SOC	LP
1662	Nunes	776188	SDP	SOC
1056	Martins	734976	?	?
?	?	?	IG2	IG1

Relacionamento binário de grau 1:N

Regra 5

Relacionamento binário de grau 1:N e participação não obrigatória do lado N.

- São necessárias três entidades, uma para cada entidade e a terceira para o relacionamento;
- A chave primária de cada entidade serve de chave primária na entidade correspondente;
- A entidade relativa ao relacionamento terá de ter entre os seus atributos as chaves primárias de cada uma das entidades.

NDoc	Nome	Tel	#Disc	Prereq
1662	Nunes	776188	SOC	LP
1662	Nunes	776188	SDP	SOC
1056	Martins	734976	?	?
?	?	?	IG2	IG1



Docente

(Ndoc, Nome, Tel)

NDoc	Nome	Tel
1662	Nunes	776188
1056	Martins	734976

Disciplina

(#Disc, Prereq)

#Disc	Prereq
SOC	LP
SDP	SOC
IG2	IG1

Leccionar

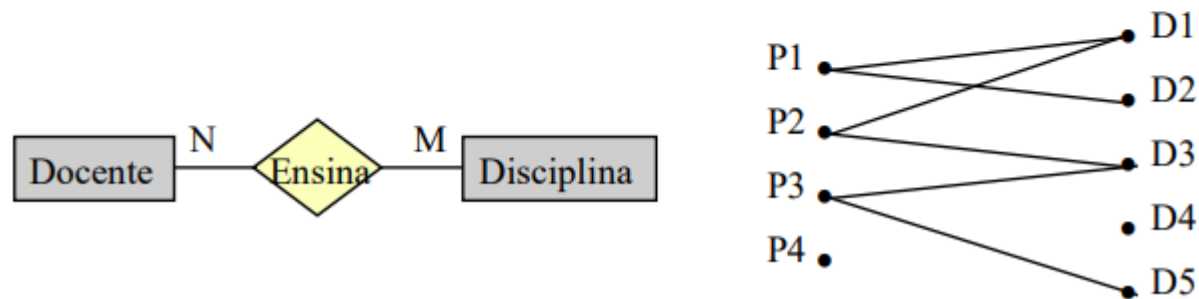
(#Disc, NDoc)

#Disc	NDoc
SOC	1662
SDP	1662

Relacionamento binário de grau N:M

Caso 6 (N:M) - Quando o grau de relacionamento binário é N:M, independentemente do tipo de participação, são sempre necessárias 3 entidades.

- Um docente pode leccionar várias disciplinas
- Uma disciplina pode ser leccionada por vários docentes



NDoc	Nome	Tel	#Disc	Prereq
1001	Couto	721334	Inf2	Inf1
1662	Nunes	776188	SOC	LP
1662	Nunes	776188	SDP	SOC
1033	Reis	716633	IA	LP
1052	Neves	714356	IA	LP
1056	Martins	734976	?	?
?	?	?	IG2	IG1

Com a utilização de uma só entidade, surgem valores nulos quer para as disciplinas que ainda não têm docente, quer para os docentes que não leccionam nenhuma disciplina.

Relacionamento binário de grau N:M

Regra 6

Relacionamento binário de grau N:M.

- São sempre necessárias três entidades, uma para cada entidade e uma terceira para o relacionamento;
- A chave primária de cada entidade serve de chave primária na entidade correspondente;
- A entidade relativa ao relacionamento terá de ter entre os seus atributos as chaves primárias de cada uma das entidades.

Docente

(Ndoc, Nome, Tel)

NDoc	Nome	Tel
1001	Couto	721334
1662	Nunes	776188
1033	Reis	716623
1052	Neves	714356
1056	Martins	734976

Disciplina

(#Disc, Prereq)

#Disc	Prereq
Inf2	Inf1
SOC	LP
SDP	SOC
IA	LP
IG2	IG1

Leccionar

(#Disc, NDoc)

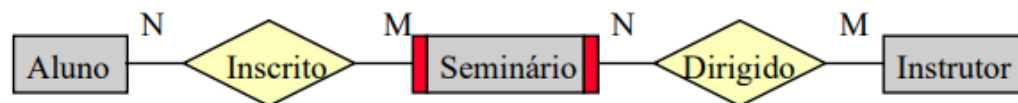
#Disc	NDoc
Inf2	1001
SOC	1662
SDP	1662
IA	1033
IA	1052

Relacionamento binário múltiplos

Na maioria dos casos, uma entidade pode ter relacionamentos binários com diversas entidades, ou seja, relacionamentos binários múltiplos.

Exemplo:

- Um aluno pode inscrever-se em vários seminários;
- Um seminário é dirigido por vários instrutores;
- Um instrutor dirige vários seminários.



3 entidades

3 entidades

Aluno (N aluno,)

Seminário (N Seminário,)

Inscrição (N Seminário, N aluno, ...)

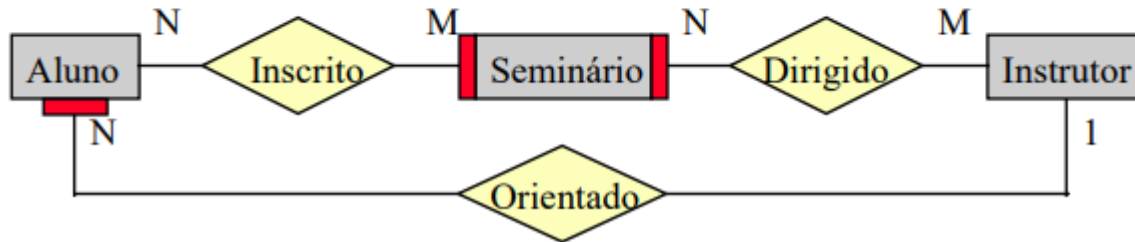
Instrutor (N Instrutor,)

Seminário (\Leftarrow)

Direcção (N seminário, N Instrutor,..)

Relacionamento binário múltiplos

Supondo que um aluno tem de ser orientado por um instrutor nos vários seminários (a intervenção do aluno é sempre a mesma, mas ocorre em vários seminários), seria necessário acrescentar outro relacionamento, obtendo-se:



O novo relacionamento dá origem às seguintes entidades:

Aluno (N_aluno, ..., N_instrutor, ...)

Instrutor (N_Instrutor,)

O Modelo final seria:

Aluno (N_aluno, ..., N_instrutor, ...)

Instrutor (N_Instrutor,)

Seminário (N_Seminário,)

Inscrição (N_Seminário, N_aluno,)

Direcção (N_seminário, N_Instrutor,...)

Relacionamento binário múltiplos

Supondo ainda, que o mesmo aluno pode ter vários instrutores, que poderão ser diferentes consoante o seminário (intervenções diferentes em cada seminário), o relacionamento “orientado” passaria a ser do tipo N:M.

O modelo final passaria a ser:

Aluno (N_aluno,)

Instrutor(N_Instrutor,)

Seminário (N_Seminário,)

Inscrição(N_Seminário, N_aluno,)

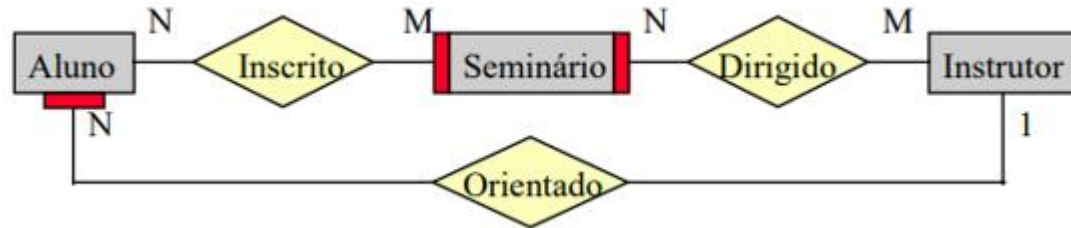
Direcção (N_seminário, N_Instrutor,...)

Orientação (N_aluno, N_Instrutor,...)

Relacionamento binário múltiplos

Questão:

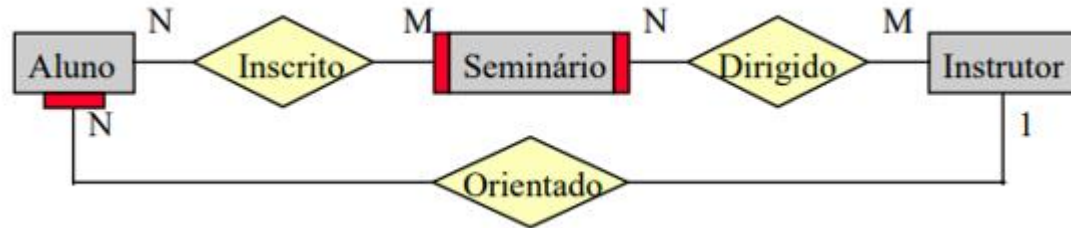
Quem é(são) o(s) orientador(es) de um aluno num dado seminário ?



Relacionamento binário múltiplos

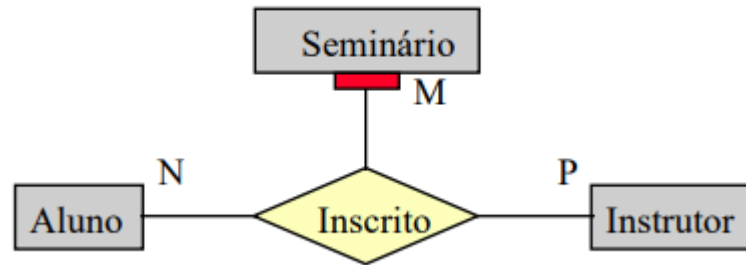
Questão:

Quem é(são) o(s) orientador(es) de um aluno num dado seminário ?



⇒ só é possível determinar quais são os instrutores de um seminário e quais são os orientadores de um dado aluno.

Relacionamento ternário

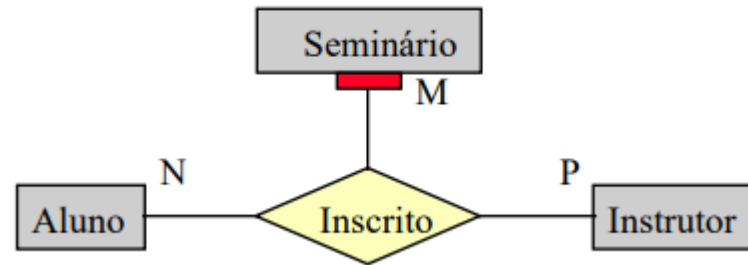


Regra 7

Relacionamento ternário (e superior)

- São sempre necessárias quatro entidades, uma para cada entidade e uma quarta para o relacionamento;
- A chave primária de cada entidade serve de chave primária na entidade correspondente;
- A entidade relativa ao relacionamento terá de ter entre os seus atributos as chaves primárias de cada uma das entidades;

Relacionamento ternário



O modelo final passaria a ser:

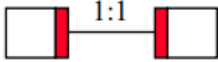
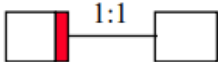
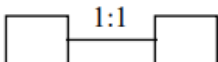
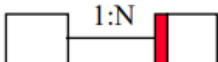
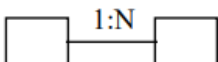
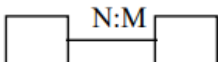
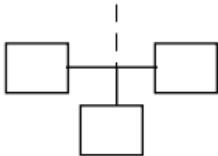
Aluno (N_aluno,)

Instrutor(N_Instrutor,)

Seminário (N_Seminário,)

Inscrição(N_Seminário, N_aluno, N_instrutor,....)

Obs.: Se cada aluno tiver um só instrutor num dado seminário, a chave primária da entidade seria somente N_Aluno, N_seminário.

Relacionamento	Nº Entidades	Observações
	1	A Chave primária pode ser a chave de qualquer das entidades.
	2	A Chave da entidade c/ participação <u>não</u> obrigatória tem de ser atributo na outra.
	3	A entidade do relacionamento terá como atributos as chaves de ambas as entidades
	2	A Chave da entidade do lado 1 tem de ser atributo na entidade do lado N.
	3	A entidade do relacionamento terá como atributos as chaves de ambas as entidades.
	3	A entidade do relacionamento terá como atributos as chaves de ambas as entidades.
	N+1	A entidade do relacionamento terá como atributos as chaves de <u>todas</u> as entidades.

Resumo!



Normalização de bancos de dados

Normalização

- Com o surgimento e o aperfeiçoamento do sistema relacional na década de 70, várias regras foram definidas para a simplificação de tabelas, recebendo o nome de formas normais.
- Cada uma destas regras apresenta um critério de adequação, e o processo de adequação de tabelas a estas regras práticas chama-se normalização.

Itens do pedido				
Pedido	Item	Preço	Quantidade	Total
15	102	9,25	2	18,5
15	132	1,3	5	6,5

Normalização

Id	Nome	Telefones
1	João	3322-1122
1	João	9999-8877
2	Maria	8888-4433
3	José	2211-5566
3	José	9765-1234

- É um processo formal, passo a passo, de análise dos atributos de uma relação
- Objetivo:
 - evitar redundância,
 - inconsistência e
 - perda de informação no banco de dados
- Teoria proposta por Codd no início dos anos 70.

Inclusão

Exclusão

Modificação

Ex: Quais problemas são decorrentes da relação Vendas?

nomeC	<u>CPF</u>	endereco	fone	<u>codPr</u>	nomePr	Vunit	qtd	total
Zé	111	ABC	9923123	A	Lápis	0,50	2	1,00
Ana	222	XYZ	9111456	B	Caneta	1,00	3	3,00
Zé	111	ABC	9912123	C	Régua	1,00	2	2,00
Pedro	444	KZZ	Null	A	Lápis	0,50	20	10,00

Anomalias de atualização

Anomalias de atualização

nomeC	<u>CPF</u>	endereco	fone	<u>codPr</u>	nomePr	Vunit	qtd	total
Zé	111	ABC	9923123	A	Lápis	0,50	2	1,00
Ana	222	XYZ	9111456	B	Caneta	1,00	3	3,00
Zé	111	ABC	9912123	C	Régua	1,00	2	2,00
Pedro	444	KZZ	Null	A	Lápis	0,50	20	10,00

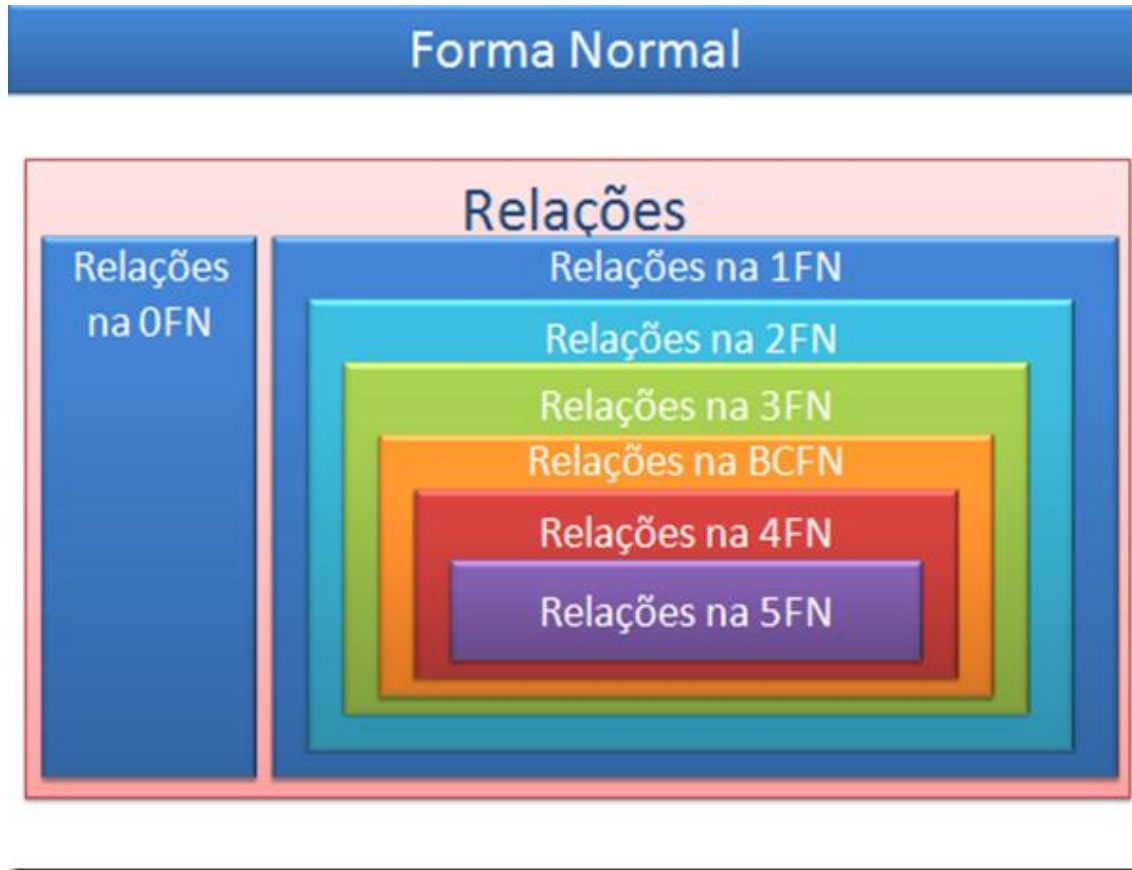
- Modificação/Atualização:
 - Uma mudança na descrição da peça A requer várias mudanças
- Inconsistência:
 - Não há nada no projeto impedindo que o produto A tenha duas ou mais descrições diferentes no BD

Anomalias de atualização

nomeC	<u>CPF</u>	endereco	fone	<u>codPr</u>	nomePr	Vunit	qtd	total
Zé	111	ABC	9923123	A	Lápis	0,50	2	1,00
Ana	222	XYZ	9111456	B	Caneta	1,00	3	3,00
Zé	111	ABC	9912123	C	Régua	1,00	2	2,00
Pedro	444	KZZ	Null	A	Lápis	0,50	20	10,00

- Inclusão:
 - Redundância do cliente e do produto
- Exclusão:
 - Se o cliente ANA fosse eliminado seria perdida a informação de que o produto B é chamado caneta e custa R\$ 1,00

Normalização de dados



Como evitar os problemas na criação de um novo banco de dados?

- Elaborando um bom modelo conceitual de dados
- Aplicando corretamente o projeto lógico de bancos de dados

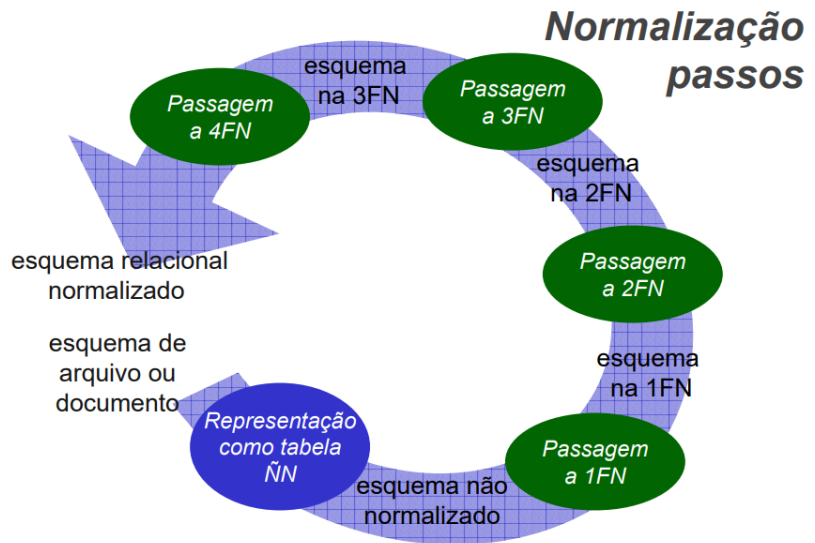
Processo de Normalização de Dados

Forma normal

- Regra que uma tabela deve obedecer para ser considerada “bem projetada”
- Há diversas formas normais, cada vez mais rígidas, para verificar tabelas relacionais
- Aqui tratadas
 - primeira forma normal (1FN)
 - segunda forma normal (2FN)
 - terceira forma normal (3FN)

Definindo
então!

Processo de Normalização



- Inicia com uma relação ou coleção de relações
- Produz uma nova coleção de relações:
 - Equivalente a coleção original (representa a mesma informação)
 - Livre de problemas
- Propriedades a serem alcançadas:
 - Minimização da redundância
 - Minimização de anomalias de inserção, exclusão e atualização de dados
 - Ex.: Florianópolis, Florianópolis, Floripa, Flópolis
- Significado: as novas relações estarão, pelo menos na 3FN
- Cada forma normal é mais restritiva que a anterior e busca a redução da redundância



Tipos de Formas Normais

Primeira Forma Normal (1FN)

- Domínio multivalorado
- Atributo determinante ou chave

Segunda Forma Normal (2FN)

- Dependência funcional

Terceira Forma Normal (3FN)

- Dependência funcional transitiva

- Tabela *não-normalizada* ou
- *tabela não-primeira-forma-normal*
 - possui uma ou mais *tabelas aninhadas*
 - tabela aninhada (ou grupo repetido ou coluna multi-valorada ou coluna não atômica)
 - coluna que ao invés de conter valores atômicos, contém tabelas aninhadas
- Abreviatura: ÑN

CódProj	Tipo	Descr	Emp					
			CodEmp	Nome	Cat	Sal	DataIni	TempAl
LSC001	Novo Desenv.	Sistema de Estoque	2146	João	A1	4	1/11/91	24
			3145	Sílvio	A2	4	2/10/91	24
			6126	José	B1	9	3/10/92	18
			1214	Carlos	A2	4	4/10/92	18
			8191	Mário	A1	4	1/11/92	12
PAG02	Manutenção	Sistema de RH	8191	Mário	A1	4	1/05/93	12
			4112	João	A2	4	4/01/91	24
			6126	José	B1	9	1/11/92	12

Tabela aninhada em uma linha de projeto

Exemplo de tabela não normalizada

<u>CodProj</u>	<u>Tipo</u>	<u>Descr</u>	<u>CodEmp</u>	<u>Nome</u>	<u>Cat</u>	<u>Sal</u>	<u>DatIni</u>	<u>TempAl</u>
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque	2146	João	A1	4	1/11/91	24
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque	3145	Silvio	A2	4	2/10/91	24
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque	6126	José	B1	9	3/10/92	18
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque	1214	Carlos	A2	4	4/10/92	18
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque	8191	Mário	A1	4	1/11/92	12
PAG02	Manutenção	Sistema de RH	8191	Mário	A1	4	1/05/93	12
PAG02	Manutenção	Sistema de RH	4112	João	A2	4	4/01/91	24
PAG02	Manutenção	Sistema de RH	6126	José	B1	9	1/11/92	12

1FN

- Cria-se uma tabela na 1FN referente à tabela Não Normalizada e que contém apenas colunas com valores atômicos, isto é, sem as tabelas aninhadas;
- Para cada tabela aninhada, cria-se uma tabela na 1FN compostas pelas seguintes colunas:
 - A chave primária de uma das tabelas na qual a tabela em questão está aninhada
 - As colunas da própria tabela
- São definidas as chaves primárias das tabelas na 1FN que correspondem a tabelas aninhadas

1FN

<u>CodProj</u>	<u>Tipo</u>	<u>Descr</u>	<u>CodEmp</u>	<u>Nome</u>	<u>Cat</u>	<u>Sal</u>	<u>DatIni</u>	<u>TempAl</u>
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque	2146	João	A1	4	1/11/91	24
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque	3145	Silvio	A2	4	2/10/91	24
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque	6126	José	B1	9	3/10/92	18
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque	1214	Carlos	A2	4	4/10/92	18
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque	8191	Mário	A1	4	1/11/92	12
PAG02	Manutenção	Sistema de RH	8191	Mário	A1	4	1/05/93	12
PAG02	Manutenção	Sistema de RH	4112	João	A2	4	4/01/91	24
PAG02	Manutenção	Sistema de RH	6126	José	B1	9	1/11/92	12

Segundo a definição da 1FN, para normalizar a tabela acima, é necessário decompô-la em duas:

1FN

Proj(CodProj, Tipo, Descr);

Emp(CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl);

Proj(CodProj, Tipo, Descr);

Emp(CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl);

Proj		
<u>CodProj</u>	Tipo	<u>Descr</u>
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque
PAG02	Manutenção	Sistema RH

ProjEmp						
<u>CodProj</u>	<u>CodEmp</u>	Nome	<u>Cat</u>	Sal	<u>DataIni</u>	<u>TempAl</u>
LSC001	2146	João	A1	4	1/11/91	24
LSC001	3145	Silvio	A2	4	2/10/91	24
LSC001	6126	José	B1	9	3/10/92	18
LSC001	1214	Carlos	A2	4	4/10/92	18
LSC001	8191	Mário	A1	4	1/11/92	12
LSC001	8191	Mário	A1	4	1/05/93	12
PAG02	4112	João	A2	4	4/01/91	24
PAG02	6126	José	B1	9	1/11/92	12

1FN

Quando encontrarmos um atributo multivalorado, deve-se criar um novo atributo que individualize a informação que esta multivalorada:

BOLETIM (matricula-aluno, materia, notas)

No caso acima, cada nota seria individualizada identificando a prova a qual aquela nota se refere:

BOLETIM (matricula-aluno, materia, numero-prova, nota)

1FN

Quando encontrarmos um atributo não atômico, deve-se dividi-lo em outros atributos que sejam atômicos:

PESSOA (CPE, nome-completo)

Vamos supor que, para a aplicação que utilizará esta relação, o atributo nome-completo não é atômico, a solução então será:

PESSOA (CPE, nome, sobrenome)

1FN

Uma tabela encontra-se na segunda forma normal, quando, além de estar na 1FN, não contem dependências parciais.

Dependência parcial -> Uma dependência parcial ocorre quando uma coluna depende apenas de parte de uma chave primária composta.

2FN

ProjEmp

CodProj	CodEmp	Nome	Cat	Sal	DataIni	TempAl
LSC001	2146	João	A1	4	1/11/91	24
LSC001	3145	Silvio	A2	4	2/10/91	24
LSC001	6126	José	B1	9	3/10/92	18
LSC001	1214	Carlos	A2	4	4/10/92	18
LSC001	8191	Mário	A1	4	1/11/92	12
LSC001	8191	Mário	A1	4	1/05/93	12
PAG02	4112	João	A2	4	4/01/91	24
PAG02	6126	José	B1	9	1/11/92	12

Depende apenas de CodEmp e não da
Chave Primária (CodProj, CodEmp)

2FN

- Copiar para a 2FN cada tabela que tenha chave primária simples ou que não tenha colunas além da chave.
- Para cada tabela com chave primária composta e com pelo menos uma coluna não chave:
 - Criar na 2FN uma tabela com as chaves primárias da tabela na 1FN
 - Para cada coluna não chave fazer a seguinte pergunta:
"a coluna depende de toda a chave ou de apenas parte dela"
 - Caso a coluna dependa de toda a chave
 - Criar a coluna correspondente na tabela com a chave completa na 2FN
 - Caso a coluna não dependa apenas de parte da chave
 - Criar, caso ainda não existir, uma tabela na 2FN que tenha como chave primária a parte da chave que é determinante da coluna em questão
 - Criar a coluna dependente dentro da tabela na 2FN

2FN

Proj

<u>CodProj</u>	Tipo	<u>Descr</u>
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque
PAG02	Manutenção	Sistema RH

ProjEmp

<u>CodProj</u>	<u>CodEmp</u>	<u>DataIni</u>	<u>TempAl</u>
LSC001	2146	1/11/91	24
LSC001	3145	2/10/91	24
LSC001	6126	3/10/92	18
LSC001	1214	4/10/92	18
LSC001	8191	1/11/92	12
LSC001	8191	1/05/93	12
PAG02	4112	4/01/91	24
PAG02	6126	1/11/92	12

Emp

<u>CodEmp</u>	Nome	<u>Cat</u>	Sal
2146	João	A1	4
3145	Silvio	A2	4
6126	José	B1	9
1214	Carlos	A2	4
8191	Mário	A1	4

2FN

Observe a relação abaixo:

BOLETIM

(matricula-aluno, codigo-materia, numero-prova, nota, data-da-prova, nome-aluno, endereço-aluno, nome-materia)

Fazendo a análise da dependência funcional de cada atributo primo, chegamos às seguintes dependências funcionais:

matricula-aluno, codigo-materia, numero-prova -> nota

codigo-materia, numero-prova -> data-da-prova

matricula-aluno -> nome-aluno, endereço-aluno

codigo-materia -> nome-materia

2FN

Concluímos então que apenas o atributo primo nota depende totalmente de toda chave primária. Para que toda a relação seja passada para a segunda forma normal, deve-se criar novas relações, agrupando os atributos de acordo com suas dependências funcionais:

BOLETIM (matricula-aluno, codigo-materia, numero-prova, nota)

PROVA (codigo-materia, numero-prova, data-da-prova)

ALUNO (matricula-aluno, nome-aluno, endereço-aluno)

MATERIA (codigo-materia, nome-materia)

2FN

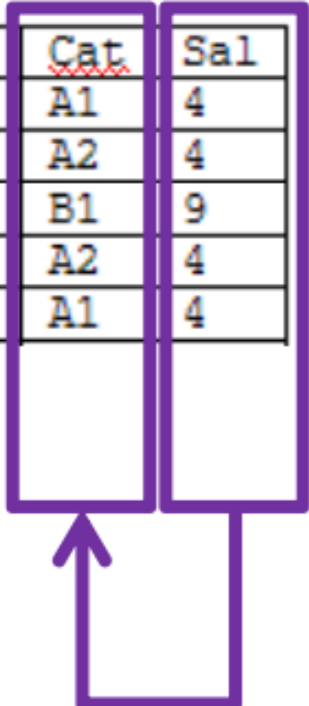
Uma tabela encontra-se na terceira forma normal, quando, além de estar na 2FN, não contém dependências transitivas

Dependência transitiva -> Uma dependência funcional transitiva ocorre quando uma coluna, além de depender da chave primária da tabela, depende de outra coluna ou conjunto de colunas da tabela

3FN

Emp

<u>CodEmp</u>	Nome	<u>Cat</u>	Sal
2146	João	A1	4
3145	Silvio	A2	4
6126	José	B1	9
1214	Carlos	A2	4
8191	Mário	A1	4



Supondo que o salário de um empregado é determinado por sua categoria funcional (Cat)

3FN

- Copiar para o esquema da 3FN cada tabela que tenha menos de duas colunas não chave, pois neste caso não há como haver dependências transitivas
- Para tabelas com duas ou mais colunas não chaves, fazer a seguinte pergunta: *"a coluna depende de alguma outra coluna não chave?"*
 - Caso dependa apenas da chave
 - Copiar a coluna para a tabela na 3FN
 - Caso a coluna depender de outra coluna
 - Criar, caso ainda não exista, uma tabela no esquema na 3FN que tenha como chave primária a coluna na qual há a dependência indireta
 - Copiar a coluna dependente para a tabela criada
 - A coluna determinante deve permanecer também na tabela original

3FN

A tabela Emp obtida pela aplicação da 2FN, que contém os dados redundantes, pode ser decomposta em outras duas (Emp e Cat)

Emp		
CodEmp	Nome	Cat
2146	João	
3145	Silvio	
6126	José	
1214	Carlos	
8191	Mário	

Emp			
CodEmp	Nome	Cat	Sal
2146	João	A1	4
3145	Silvio	A2	4
6126	José	B1	9
1214	Carlos	A2	4
8191	Mário	A1	4

Cat	
Cat	Sal
A1	4
A2	4
B1	9

3FN

<u>CodProj</u>	<u>Tipo</u>	<u>Descr</u>	<u>CodEmp</u>	<u>Nome</u>	<u>Cat</u>	<u>Sal</u>	<u>DatIni</u>	<u>TempAl</u>
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque	2146	João	A1	4	1/11/91	24
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque	3145	Silvio	A2	4	2/10/91	24
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque	6126	José	B1	9	3/10/92	18
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque	1214	Carlos	A2	4	4/10/92	18
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque	8191	Mário	A1	4	1/11/92	12
PAG02	Manutenção	Sistema de RH	8191	Mário	A1	4	1/05/93	12
PAG02	Manutenção	Sistema de RH	4112	João	A2	4	4/01/91	24
PAG02	Manutenção	Sistema de RH	6126	José	B1	9	1/11/92	12

3FN

Proj

<u>CodProj</u>	Tipo	<u>Descr</u>
LSC001	Novo <u>Desenv.</u>	Sistema de Estoque
PAG02	Manutenção	Sistema RH

Emp

<u>CodEmp</u>	Nome	<u>Cat</u>
2146	João	
3145	Silvio	
6126	José	
1214	Carlos	
8191	Mário	

ProjEmp

<u>CodProj</u>	<u>CodEmp</u>	<u>DataIni</u>	<u>TempAl</u>
LSC001	2146	1/11/91	24
LSC001	3145	2/10/91	24
LSC001	6126	3/10/92	18
LSC001	1214	4/10/92	18
LSC001	8191	1/11/92	12
LSC001	8191	1/05/93	12
PAG02	4112	4/01/91	24
PAG02	6126	1/11/92	12

Cat

<u>Cat</u>	Sal
A1	4
A2	4
B1	9

3FN

Observe a relação abaixo:

PEDIDO(numero-pedido, codigo-cliente, data-pedido, nome-cliente, codigo-cidade-cliente, nome-cidade-cliente)

Fazendo a análise da dependência funcional de cada atributo primo, chegamos às seguintes dependências funcionais:

numero-pedido -> codigo-cliente

numero-pedido -> data-pedido

codigo-cliente -> nome-cliente

codigo-cliente -> codigo-cidade-cliente

codigo-cidade-cliente -> nome-cidade-cliente

3FN

Concluimos então que apenas os atributos primos `codigo-cliente` e `data-pedido` dependem não transitivamente totalmente de toda chave primária.

Observe que:

`numero-pedido` -> `codigo-cliente` -> `nome-cliente`

`numero-pedido` -> `codigo-cliente` -> `codigo-cidade-cliente`

`numero-pedido` -> `codigo-cliente` -> `codigo-cidade-cliente` ->
`nome-cidade-cliente`

3FN

Devemos resolver inicialmente as dependências mais simples, criando uma nova relação onde `codigo-cliente` é a chave, o `codigo-cliente` continuará na relação PEDIDO como atributo primo, porém, os atributos que dependem dele devem ser transferidos para a nova relação:

PEDIDO (numero-pedido, `codigo-cliente`, `data-pedido`)

CLIENTE (codigo-cliente, `nome-cliente`, `codigo-cidade-cliente`, `nome-cidade-cliente`)

3FN

As dependências transitivas da relação PEDIDO foram eliminadas, porém ainda devemos analisar a nova relação CLIENTE:

codigo-cliente -> codigo-cidade-cliente -> nome-cidade-cliente

Observe que o nome-cidade-cliente continua com uma dependência transitiva, vamos resolvê-la da mesma maneira :

PEDIDO (numero-pedido, codigo-cliente, data-pedido)

CLIENTE(codigo-cliente, nome-cliente, codigo-cidade-cliente)

CIDADE (codigo-cidade-cliente, nome-cidade-cliente)

3FN

Para pensar...



Num banco de dados, em quais casos redundância deve ser evitada? Em quais casos ela é necessária? Dê exemplos de aplicações em cada caso.

Na próxima aula...

Falaremos sobre consultas a bases de dados e linguagem SQL.

