

NORMALIZAÇÃO

Profa. Leticia Zoby

Introdução

- O processo de normalização pode ser visto como processo no qual são eliminados esquemas de relações (tabelas) não satisfatórios, decompondo-os, através da separação de seus atributos em esquemas de relações menos complexas mas que satisfaçam as propriedades desejadas.
- A Dependência Funcional é a principal ferramenta para medir o “quão apropriado” é o agrupamento de atributos numa relação.

Introdução

- **Análise de Dependência Funcional**

- Dependência Funcional

O atributo B é funcionalmente dependente do atributo A se, em qualquer instante, um valor em A determina, de modo único, o valor correspondente em B, na mesma relação

Exemplo:

Num-Emp \rightarrow Nome-Emp

O Nome-Emp é funcionalmente dependente do Num-Emp, pelo fato de cada Num-Emp está associado sempre ao mesmo Nome-Emp.

EMPREGADO		
#Num-Emp	Nome-Emp	Vir-Sal-Emp

Introdução

- Análise de Dependência Funcional
 - Dependência Funcional
- $A \rightarrow B$, lê - se:
 - *A funcionalmente determina B*
 - *B é funcionalmente dependente de A*
 - *B é função de A*
- Para cada valor de A só existe um valor de B
- $A \not\rightarrow B$, negação de $A \rightarrow B$.
A ou B podem ser um conjunto de atributos.

Introdução

- Dependência Funcional

Propõe três tipos de dependência entre os atributos de uma tabela

- Dependência Funcional Total
- Dependência Funcional Parcial
- Dependência Funcional Transitiva

Introdução

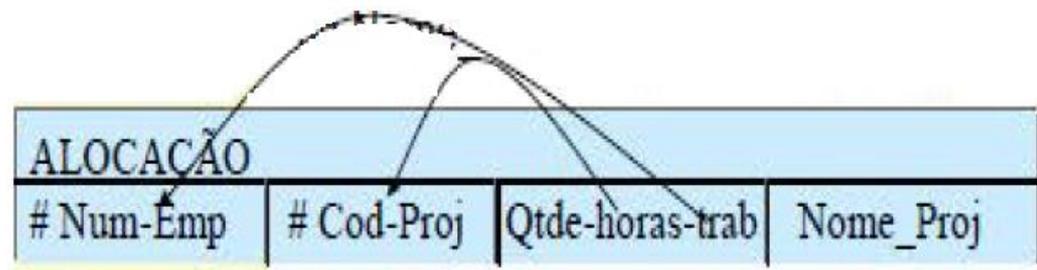
• Dependência Funcional Total

- Os atributos de uma tabela tem que depender da chave primária e somente da chave primária
- Um atributo C é totalmente funcionalmente dependente da chave primária composta pelo atributos A e B, quando for funcionalmente dependente de A e B não dependente funcionalmente de qualquer parte da chave primária

Exemplo:

Qtde_horas depende da chave primária totalmente.

Já Nome_Proj somente de cod_proj.



Introdução

- Dependência Funcional Parcial
 - O atributo C é parcialmente funcionalmente dependente da chave primária composta pelos atributos A e B quando for funcionalmente dependente de A ou B e não de ambos A e B.
 - A dependência funcional parcial ocorre quando a chave primária da relação é composta e se constitui numa anomalia que se deve ser evitada

ALOCACÃO			
# Num-Emp	# Cod-Proj	Qtde-horas-trab	Nome_Proj



The diagram illustrates a partial functional dependency. A curved arrow originates from the 'Nome_Proj' attribute and points to the '# Cod-Proj' attribute, indicating that 'Nome_Proj' is partially functionally dependent on the composite primary key consisting of '# Num-Emp' and '# Cod-Proj'.

Introdução

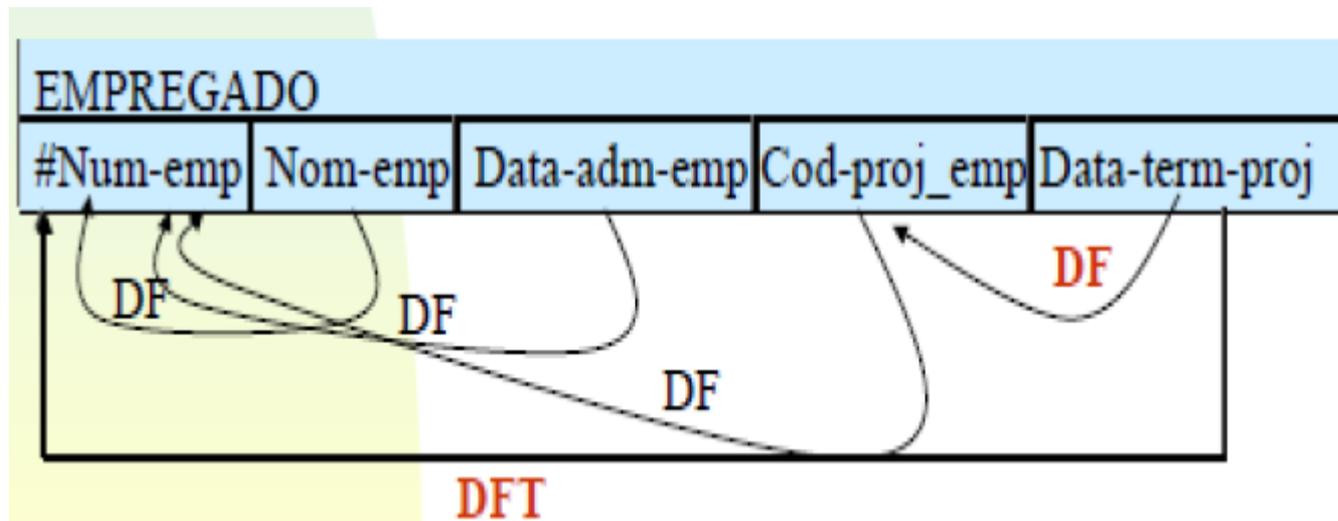
- Dependência Funcional Parcial
 - Solução: criação de uma nova relação composta pelo atributo ou atributos que dependem de parte da chave e a chave que determine, de modo único, estes atributos

ALOCAÇÃO		
# Num-Emp	# Cod-Proj	Qtde-horas-trab

PROJETO	
# Cod-Proj	Nome_Proj

Introdução

- Dependência Funcional Transitiva
 - O atributo C é dependente funcional transitivo (DFT) de A se C é funcionalmente dependente de A, na mesma relação
 - A DFT constitui uma anomalia que deve ser evitada



Introdução

- Dependência Funcional Transitiva
 - Solução: criação de uma nova relação que será composta pelo atributo ou atributos que são dependentes funcionais transitivos tendo como chave primária o atributo que determina a transitividade

EMPREGADO			
#Num-emp	Nom-emp	Data-adm-emp	Cod-proj_emp

PROJETO	
# Cod-Proj_emp	Data-term-proj

Introdução

- A normalização é importante para identificar um bom projeto relacional.
- Um bom MER e sua consequente conversão para um MR, praticamente deixa o esquema relacional *normalizado*.
- Utiliza-se a normalização somente para *validar* um projeto relacional.

Redundância de dados

Identidade	Nome	Endereço	Habilidade
8795835	Édson Arantes	Ponta da Praia	Futebol
8795835	Édson Arantes	Ponta da Praia	Voleibol
8795835	Édson Arantes	Ponta da Praia	Basquete
8795835	Édson Arantes	Ponta da Praia	Atletismo
8795835	Édson Arantes	Ponta da Praia	Tênis

Se Pelé mudar de endereço ?

Um novo esporte para Pelé ?

Problemas com atualização

Problemas com inclusão

Idealmente

Identidade	Nome	Endereço	Habilidade
8795835	Édson Arantes	Ponta da Praia	{Futebol, Voleibol, Basquete, Atletismo, Tênis}

Mas isto não é uma tabela (atributo *habilidade* não é atômico)!

Identidade	Nome	Endereço
8795835	Édson Arantes	Ponta da Praia
...

Identidade	Esporte
8795835	Futebol
8795835	Voleibol
8795835	Basquetebol
8795835	Atletismo
8795835	Tênis

A repetição da coluna Identidade é uma redundância necessária

Identidade	Nome	Endereço	Habilidade
8795835	Édson Arantes	Ponta da Praia	Futebol
8795835	Édson Arantes	Ponta da Praia	Voleibol
8795835	Édson Arantes	Ponta da Praia	Basquete
8795835	Édson Arantes	Ponta da Praia	Atletismo
8795835	Édson Arantes	Ponta da Praia	Tênis

Identidade \rightarrow Nome

Nome $\neg \rightarrow$ Identidade

Identidade \rightarrow Endereço

Habilidade $\neg \rightarrow$ Identidade

Identidade $\neg \rightarrow$ Habilidade

Identidade \rightarrow Nome, Endereço

Introdução

- O processo de normalização como foi proposto inicialmente por Codd conduz um esquema de relação através de uma bateria de testes para certificar se o mesmo está na 1ª, 2ª e 3ª Formas Normais.
- Essas três Formas Normais são baseadas em dependências funcionais dos atributos do esquema de relação.

Primeira Forma Normal (1FN)

- Toda tabela deve ser “minimamente” normalizada (1FN).
- Uma relação estará na 1FN se não houver atributo representando agrupamento (não atômico) e nem atributo repetitivo (multivalorado).
- Tabela em 1FN: O valor de uma coluna de uma tabela é indivisível.

Primeira Forma Normal (1FN)

- Exemplo: **Cliente**

1-Código

2- {Telefone}

3- Endereço (Rua, Número, Cidade)

Gerando a tabela resultante:

Cliente	<u>Código</u>	Telefone 1	Endereço		
		Telefone <i>n</i>	Rua	Número	Cidade

Sendo que a mesma não está na 1FN pois seus atributos não são atômicos. Para que a tabela fique na 1FN temos que eliminar os atributos não atômicos, gerando as seguintes tabelas como resultado:

Cliente	<u>Código</u>	Rua	Número	Cidade
----------------	---------------	-----	--------	--------

Cliente_Telefone	<u>Código Cliente</u>	<u>Telefone Cliente</u>
-------------------------	-----------------------	-------------------------

Matrícula	Nome	Cod Cargo	NomeCargo	CodProj	DataFim	Horas
120	João	1	Programador	01	17/07/95	37
120	João	1	Programador	08	12/01/96	12
121	Hélio	1	Programador	01	17/07/95	45
121	Hélio	1	Programador	08	12/01/96	21
121	Hélio	1	Programador	12	21/03/96	107
270	Gabriel	2	Analista	08	12/01/96	10
270	Gabriel	2	Analista	12	21/03/96	38
273	Silva	3	Projetista	01	17/07/95	22
274	Abraão	2	Analista	12	21/03/96	31
279	Carla	1	Programador	01	17/07/96	27
279	Carla	1	Programador	08	12/01/96	20
279	Carla	1	Programador	12	21/03/96	51
301	Ana	1	Programador	12	21/03/96	16
306	Manoel	3	Projetista	17	21/03/96	67

Redundância de dados

Problemas de inserção, remoção e atualização

- Uma tabela em 1FN continua com problemas de redundância de dados, inclusão, atualização, e remoção.
- É necessário a aplicação de outras formas normais:
 - 2FN
 - 3FN

Segunda Forma Normal 2FN

*Uma tabela está na Segunda Forma Normal (2FN) se ela está 1FN e todo atributo do complemento de uma chave candidata é **totalmente funcionalmente dependente** daquela chave.*

$A, B, C \rightarrow D$ (D é *totalmente funcionalmente dependente* de $\{A, B, C\}$) se para todo valor de $\{A, B, C\}$ só existe um valor de D , e se D não é funcionalmente dependente de A , ou B , ou C .

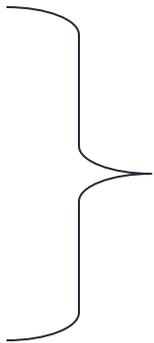
E #	Enome	Sexo	Idade	D #	Dnome	Opinião
E 1	João	M	25	D 1	Mat	Boa
E 1	João	M	25	D 2	Quim	Má
E 1	João	M	25	D 3	Fis	Boa
E 2	Maria	F	22	D 2	Quim	Satisf.
E 2	Maria	F	22	D 3	Fis	Satisf.
E 2	Maria	F	22	D 4	Est	Má
E 3	João	M	27	D 2	Quim	Boa
E 3	João	M	27	D3	Fis	Boa

Chaves Candidatas	Complementos da Chave
{E#, D#}	Enome, Sexo, Idade, Dnome, Opinião

$\{E\#, D\#\} \multimap \rightarrow \text{Enome} \quad (E\# \rightarrow \text{Enome})$

$\{E\#, D\#\} \multimap \rightarrow \text{Sexo} \quad (E\# \rightarrow \text{Sexo})$

$\{E\#, D\#\} \multimap \rightarrow \text{Idade} \quad (E\# \rightarrow \text{Idade})$

 Tabela 1

$\{E\#, D\#\} \multimap \rightarrow \text{Dnome} \quad (D\# \rightarrow \text{Dnome})$

\longrightarrow Tabela 2

$\{E\#, D\#\} \rightarrow \text{Opini\~{a}o}$

\longrightarrow

Tabela 3

E #	Enome	Sexo	Idade
E1	João	M	25
E2	Maria	F	22
E3	João	M	27

D #	Dnome
D1	Mat
D2	Quim
D3	Fis
D4	Est

E #	D #	Opinião
E1	D1	Boa
E1	D2	Pobre
E1	D3	Boa
E2	D2	Satisfatória
E2	D3	Satisfatória
E2	D4	Pobre
E3	D2	Boa
E3	D3	Boa

Segunda Forma Normal 2FN

- Uma tabela T está na 2FN se estiver na 1FN e todos os seus atributos não chaves forem totalmente funcionalmente dependente da chave primária C.
- Se uma tabela não está na 2FN a mesma pode ser normalizada gerando outras tabelas cujos atributos que não façam parte da chave primária sejam totalmente funcionalmente dependente da mesma, ficando a tabela na 2FN.

Terceira Forma Normal 3FN

Uma relação está em 3FN se, e somente se, estiver em 2FN e todos os atributos não-chave forem **dependentes não-transitivos** da chave primária.

Matr	Nome	codCargo	nomeCargo	codPro	DataFim	horas
120	João	1	Programador	01	17/07/95	37
120	João	1	Programador	08	12/01/96	12
121	Hélio	1	Programador	01	17/07/95	45
121	Hélio	1	Programador	08	12/01/96	21
121	Hélio	1	Programador	12	21/03/96	107
270	Gabriel	2	Analista	08	12/01/96	10
270	Gabriel	2	Analista	12	21/03/96	38
273	Silva	3	Projetista	01	17/07/95	22
274	Abraão	2	Analista	12	21/03/96	31
279	Carla	1	Programador	01	17/07/96	27
279	Carla	1	Programador	08	12/01/96	20
279	Carla	1	Programador	12	21/03/96	51
301	Ana	1	Programador	12	21/03/96	16

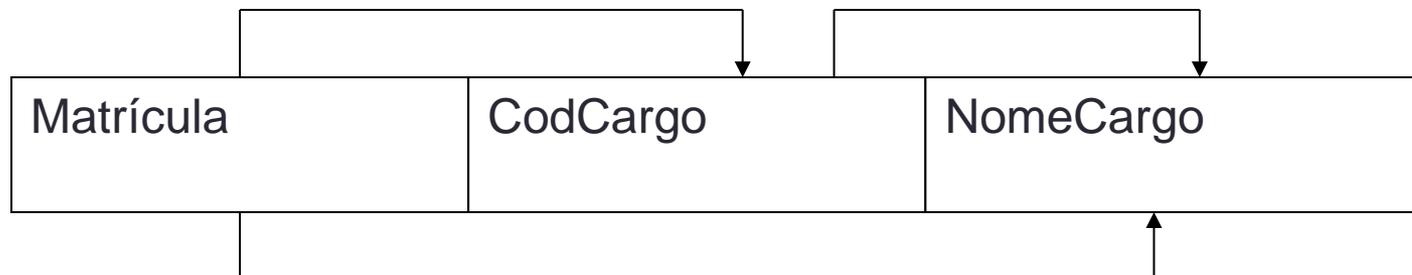
2FN

Matr	Nome	codCargo	nomeCargo
120	João	1	Programador
121	Hélio	1	Programador
270	Gabriel	2	Analista
273	Silva	3	Projetista
274	Abraão	2	Analista
279	Carla	1	Programador
301	Ana	1	Programador
306	Manuel	3	Projetista

codProj	dataFim
01	17/07/95
08	12/01/96
12	21/03/96

Matr	codPro	horas
120	01	37
120	08	12
121	01	45
121	08	21
121	12	107
270	08	10
270	12	78
273	01	22
274	12	31
279	01	27
279	08	20
279	12	51
301	01	16
301	12	85
306	12	67

Problemas de inclusão, remoção e atualização



NomeCargo é dependente transitivo de Matrícula.

Matr	Nome	codCargo
120	João	1
121	Hélio	1
270	Gabriel	2
273	Silva	3
274	Abraão	2
279	Carla	1
301	Ana	1

codCargo	nome
1	Programador
2	Analista
3	Projetista

- Exemplo 2:

CódProj	Tipo	Descr	Emp					
			CodEmp	Nome	Cat	Sal	DataIni	TempAI
LSC001	Novo Desenv.	Sistema de Estoque	2146	João	A1	4	1/11/91	24
			3145	Silvio	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

ProjEmp (CodProj, Tipo, Descr, CodEmp, Nome, Cat, Sal, DataIni, TempAI)

Proj (CodProj, Tipo, Descr)

ProjEmp (CodProj, CodEmp, Nome, Cat, Sal, DataIni, TempAI)

- Exemplo 2:

Proj (CódProj, Tipo, Descr)

ProjEmp (CódProj, CodEmp, Nome, Cat, Sal, DataIni, TempAl)

1FN

Proj:

CódProj	Tipo	Descr
LSC001	Novo Desenv.	Sistema de Estoque
PAG02	Manutenção	Sistema de RH

ProjEmp:

CódProj	CodEmp	Nome	Cat	Sal	DataIni	TempAl
LSC001	2146	João	A1	4	1/11/91	24
LSC001	3145	Sílvio	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
PAG02	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

- Exemplo 2:

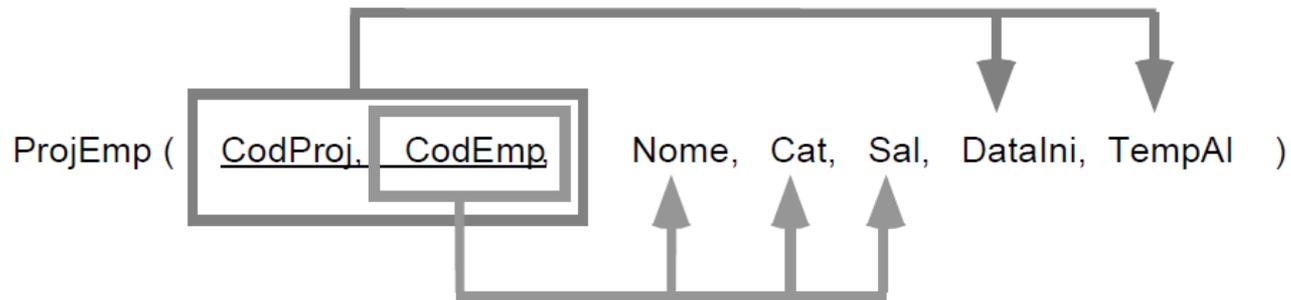
Para passar à 2FN, ou seja, eliminar as dependências de parte da chave primária é necessário dividir a tabela ProjEmp em duas tabelas como:

ProjEmp (CodProj, CodEmp, DataIni, TempAl)

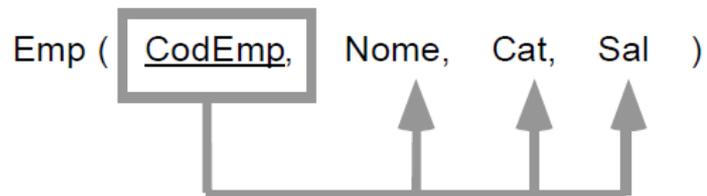
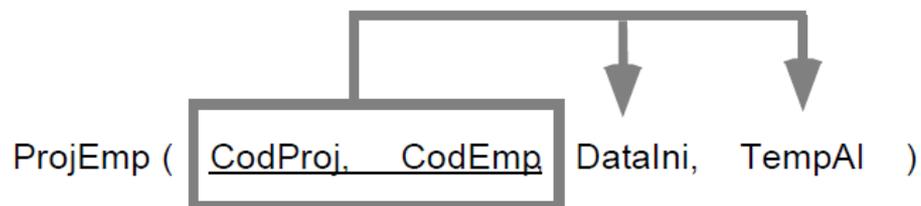
Emp (CodEmp, Nome, Cat, Sal)

• Exemplo 2:

Tabela na 1FN e dependências funcionais



Tabelas na 2FN e dependências funcionais



- Exemplo 2:

Proj (CodProj, Tipo, Descr)

ProjEmp (CodProj, CodEmp, DataIni, TempAl)

Emp (CodEmp, Nome, Cat, Sal)

} 2FN

Proj:

CódProj	Tipo	Descr
LSC001	Novo Desenv.	Sistema de Estoque
PAG02	Manutenção	Sistema de RH

Emp:

CodEmp	Nome	Cat	Sal
2146	João	A1	4
3145	Sílvio	A2	4
6126	José	B1	9
1214	Carlos	A2	4
8191	Mário	A1	4
8191	Mário	A1	4
4112	João	A2	4
6126	José	B1	9

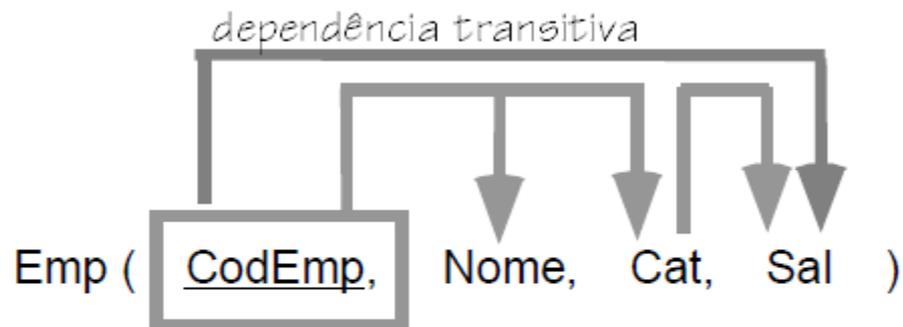
ProjEmp:

CódProj	CodEmp	DataIni	TempAl
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
PAG02	8191	1/05/93	12
PAG02	4112	4/01/91	24
PAG02	6126	1/11/92	12

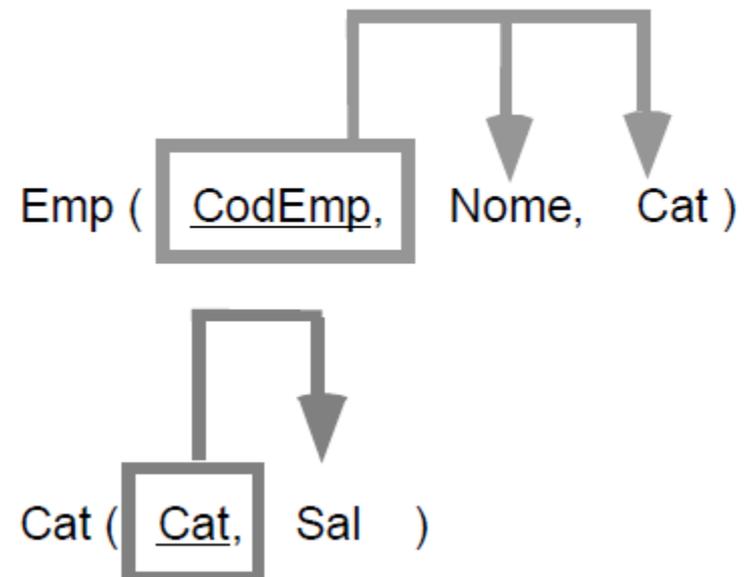
- Exemplo 2:

Uma tabela encontra-se na 3FN quando, além de estar na 2FN, toda coluna não chave depende diretamente da chave primária, ie, quando não há dependência funcional transitiva ou indiretas.

Tabela na segunda forma normal (2FN)



Tabelas na terceira forma normal (3FN)



- Exemplo 2:

Proj (CodProj, Tipo, Descr)

ProjEmp (CodProj, CodEmp, DataIni, TempAl)

Emp (CodEmp, Nome, Cat)

Cat (Cat, Sal)

} 3FN

Proj:

CódProj	Tipo	Descr
LSC001	Novo Desenv.	Sistema de Estoque
PAG02	Manutenção	Sistema de RH

Emp:

NúmEmp	Nome	Cat
2146	João	A1
3145	Sílvio	A2
6126	José	B1
1214	Carlos	A2
8191	Mário	A1
8191	Mário	A1
4112	João	A2
6126	José	B1

ProjEmp:

CódProj	NúmEmp	DataIni	TempAl
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
PAG02	8191	1/05/93	12
PAG02	4112	4/01/91	24
PAG02	6126	1/11/92	12

Cat:

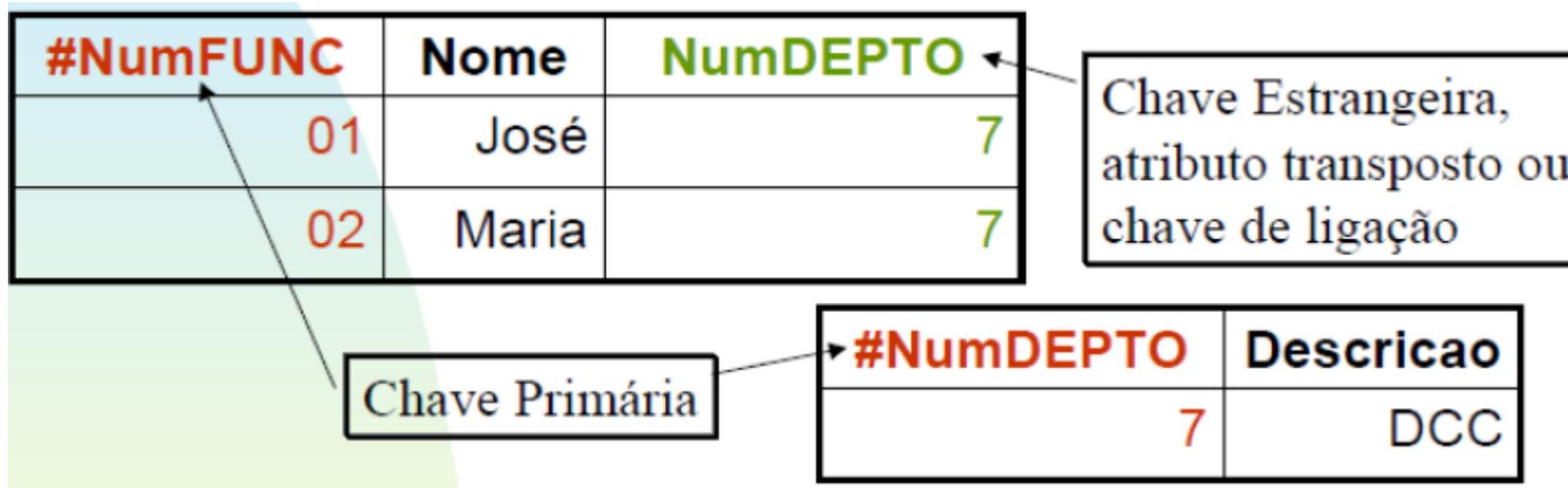
Cat	Sal
A1	4
A2	4
B1	9

Aspecto de Integridade

- Restrição de Chave
- Uma relação deve ter pelo menos uma chave.
- Definição de chave: uma chave é um atributo ou conjunto de atributos cujo valor ou combinação de valores deve ser distinto em qualquer instância da relação.
- A existência de pelo menos uma chave é uma condição obrigatória, tendo em vista que uma relação é definida como um conjunto de tuplas e, todas as tuplas devem ter um valor distinto.

Aspecto de Integridade

- Restrição de Entidade
- Especifica que nenhum valor de chave primária pode ser nulo.
- Uma tupla em uma tabela que se refere a uma outra relação deve referenciar uma tupla existente naquela relação.



Aspecto de Integridade

- Outros Tipos de Restrições
- As restrições verificadas tratam somente das restrições estruturais.
- Existem as restrições de integridade semântica ou regras de negócio.
- São implementadas por meio de **gatilhos** ou **stored procedures** (procedimentos armazenados).
- Exemplo:
 - o salário de uma atriz não pode ser maior que $1/3$ do orçamento total do filme.

Restrições de Integridade no Modelo Relacional

- Integridade Lógica
 - Conjunto de regras que existem para o modelo de dados, assim como um conjunto de regras de negócio, que regem a manipulação do BD, de forma a não ferir nenhuma destas regras estabelecidas.
- Integridade Física
 - Conjunto de procedimentos operacionais que garantem a integridade do BD, mesmo em situações de falha de algum componente do ambiente onde o BD é manipulado.
 - Exemplo:
 - Valor mínimo de depósito para abertura de uma conta R\$10.000,00
 - Conta corrente sem movimento há 180 dias será encerrada.

Especificação de Banco de Dados Relacionais

- A especificação de um banco de dados relacional (chamada de esquema do banco de dados) deve conter no mínimo a definição do seguinte:
 - Tabelas que formam o banco de dados
 - Colunas que as tabelas possuem
 - Restrições de integridade

Fixando

- Formas Normais
 - 1ª Forma Normal (1FN): toda relação deve ter uma chave primária e deve-se garantir que todo atributo seja atômico. Atributos compostos devem ser separados.
 - 2ª Forma Normal (2FN): toda relação deve estar na 1FN e devem-se eliminar dependências funcionais parciais, ou seja, todo atributo não chave deve ser totalmente dependente da chave primária.
 - 3ª Forma Normal (3FN): Toda relação deve estar na 2FN e devem-se eliminar dependências funcionais transitivas, ou seja, todo atributo não chave deve ser mutuamente independente.

Atividades

- Estudar
 - Capítulo 10 do livro
 - ELMASRI, R., NAVATHE, S. B., Sistemas de Banco de Dados
 - Capítulo 6 do livro
 - HEUSER, C. A. , Projeto de banco de Dados.

Exercício

- Considere a seguinte tabela:

Trecho(CodEst,CodLocIni,CodLocFim, Distancia, NomeEst,
NomeLocIni, SiglaUFIni, NomeUFIni)

- As dependências funcionais, podendo incluir dependências transitivas, que existem nesta tabela são as seguintes:
 - CodEst -> NomeEst
 - CodLocIni -> NomeLocIni
 - CodLocIni -> SiglaUFIni
 - CodLocIni -> NomeUFIni
 - SiglaUFIni -> NomeUFIni
 - (CodEst, CodLocIni, CodLocFim) -> Distancia
- A tabela encontra-se em qual forma normal? Se a tabela não se encontrar na terceira forma normal, mostre a transformação da tabela para a terceira forma normal e cada forma normal intermediária (entre a que a tabela se encontra e a terceira forma normal).