

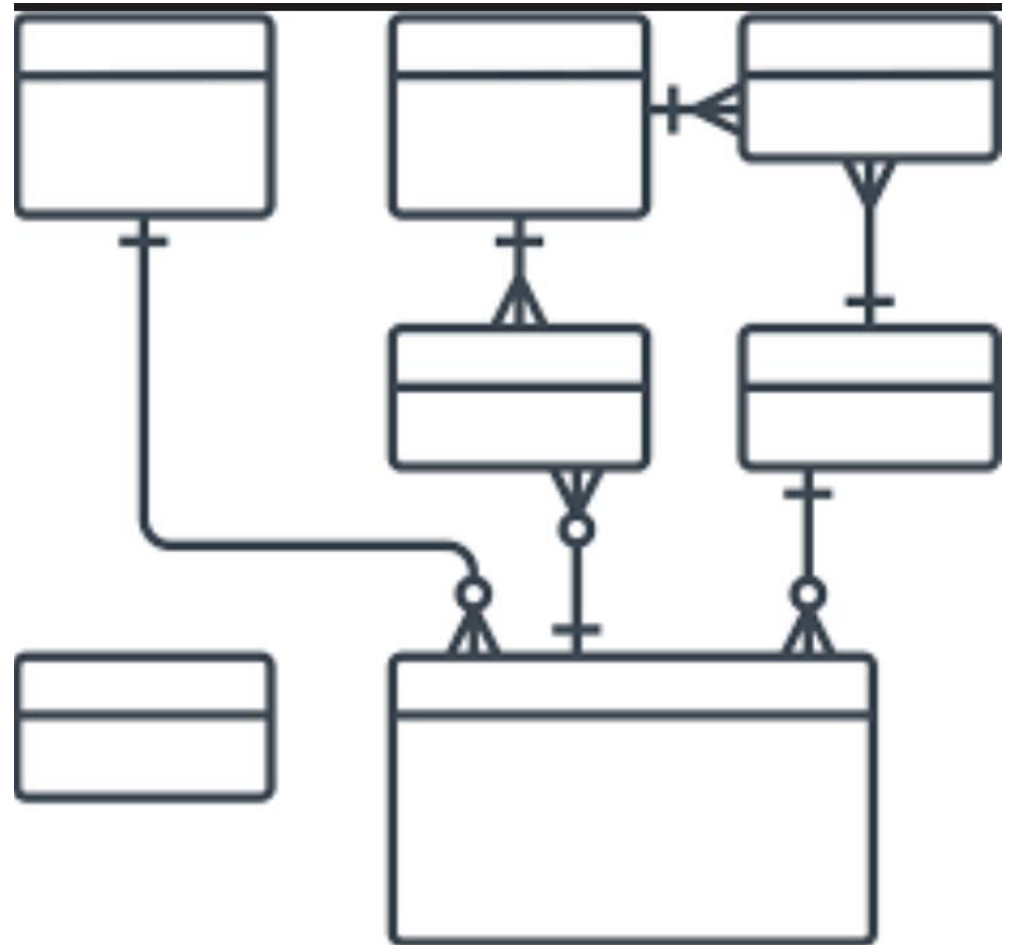
FIAP GRADUAÇÃO

SISTEMAS DE INFORMAÇÃO

Designer e Desenvolvimento de Banco de Dados

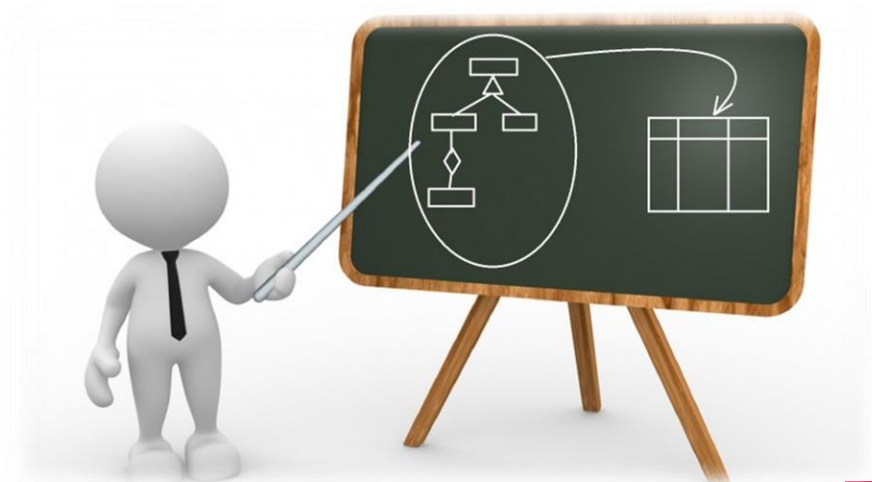
PROF. Luciano Melo

MER - Normalização



Agenda

- ✓ Definição e Objetivo
- ✓ 1ª Forma Normal (1FN)
- ✓ 2ª Forma Normal (2FN)
- ✓ 3ª Forma Normal (3FN)
- ✓ Conclusão



Definição e Objetivo



Normalização em um projeto de banco de dados relacional é o processo que **examina os atributos de uma entidade** com o objetivo de evitar redundância e que o modelo apresente anomalias relacionadas a inserção, alteração e exclusão de dados

A normalização é, portanto, o instrumento utilizado para garantir a **qualidade do projeto de banco de dados relacional**

Consiste da aplicação de regras ao modelo de dados que irão garantir a exclusão de redundância e de anomalias causadas por tais redundância. Essas regras são chamadas de **FORMAS NORMAIS**

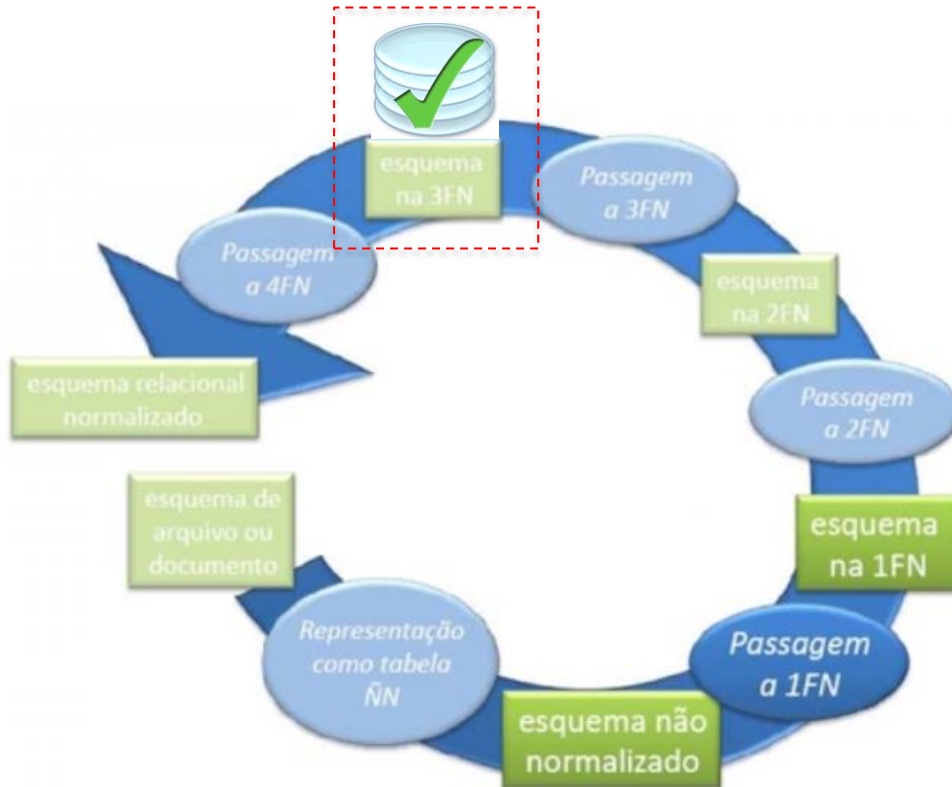
Formas Normais (FN)

- As Formas Normais são um conjunto de regras ou restrições as quais os dados devem satisfazer.
- Foi Introduzida pelo Dr. Edgard F. Codd em 1970
- Apresenta técnicas bem definidas para remover anomalias do modelo relacional
- É aplicada a cada entidade analisando os atributos da mesma, de forma que seja considerada bem projetada no modelo
- Existem 6 Formas Normais (conjunto de regras a serem aplicadas), mas a aplicação da 3ª Forma Normal já é suficiente para garantir a estabilidade e uma boa qualidade do modelo de dados



Estudaremos neste curso apenas até a 3FN

Processo de Normalização



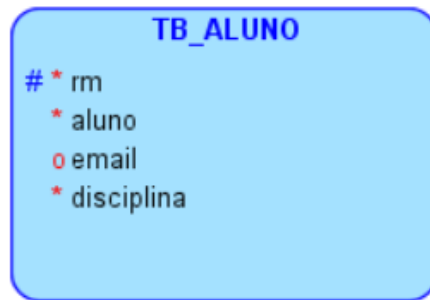
- O processo de normalização consiste na análise dos atributos de cada entidade
- Para eliminar redundâncias e problemas de anomalias, identifica-se qual o nível de normalização da entidade e aplica-se regras para torná-la mais normalizada

- Como regra geral para normalizar uma entidade, uma ou mais entidades devem ser criadas e relacionadas

ANOMALIAS

Anomalias são problemas que podem acontecer em operações de **inclusão**, **alteração** e **remoção** de dados, como integridade e redundância.

Exemplo:



Vamos analisar instâncias dessa entidade (como se fossem linhas da tabela)

RM	Aluno	email	Disciplina
12345	José de Alencar	jalecar@mymail.com.br	Algoritimos
12345	José de Alencar	jalecar@mymail.com.br	Banco de Dados I
12345	José de Alencar	jalecar@mymail.com.br	Tecnicas de Programação
12345	José de Alencar	jalecar@mymail.com.br	Gestão de TI

Esta Entidade está mal projetada! Vejamos ...

ANOMALIAS

RM	Aluno	email	Disciplina
12345	José de Alencar	jalecar@mymail.com.br	Algoritimos
12345	José de Alencar	jalecar@mymail.com.br	Banco de Dados I
12345	José de Alencar	jalecar@mymail.com.br	Tecnicas de Programação
12345	José de Alencar	jalecar@mymail.com.br	Gestão de TI
12390	Machado Assis	massis@myemail.com.br	Banco de Dados I

1. E se José iniciar uma nova disciplina? ➡ Anomalia de Inclusão

Vou ter que adicionar todos os dados novamente?

2. E se José mudar o e-mail dele? ➡ Anomalia de atualização

Quantas linhas devem ser atualizadas?

3. E se José for removido? ➡ Anomalia de Exclusão

As disciplinas de Algoritmo, Tecnica de Programação e Gestão de TI deixariam de existir na minha base de dados

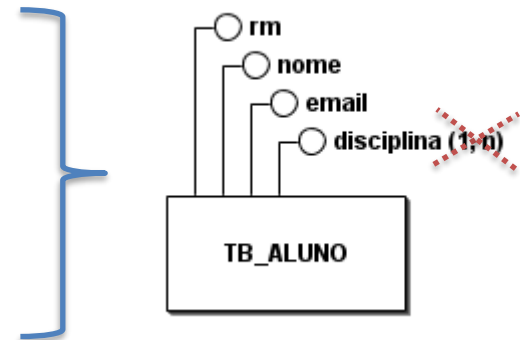
ANOMALIAS

Para a tabela anterior, por exemplo, poderíamos remover a redundância da seguinte maneira

RM	Aluno	email	Disciplina
12345	José de Alencar	jalecar@mymail.com.br	{ Algoritimos, Banco de Dados I, Tecnicas de Programação, Gestão de TI }



O problema é que o atributo Disciplina passa a ter vários valores possíveis, portanto, é um atributo multivalorado. Além disso, não resolveríamos a anomalia de exclusão



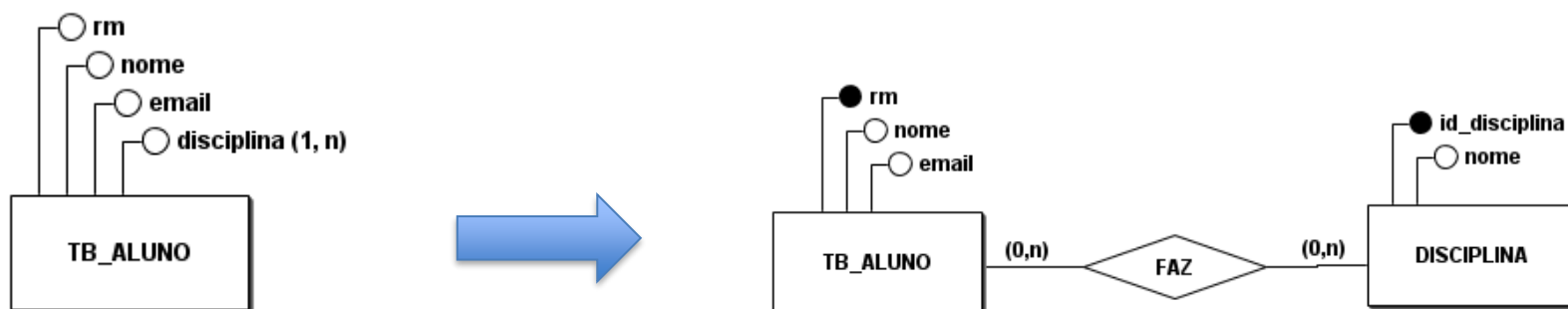
➔ Precisamos NORMALIZAR esta entidade!!!

1ª Forma Normal (1FN)

Uma ENTIDADE está na 1FN se, e somente se, não possuir atributos multivalorados.

→ Ou seja, não se admite na tabela atributos que possam ter mais de um valor na mesma tupla (Atributos multivalorados)

1º passo: Atributos multi-valorados devem ser transformados em ENTIDADES e relacionados com a cardinalidade adequada



1ª Forma Normal (1FN)

→ Vamos usar a entidade TB_FUNCIONARIO criada por um analista para armazenar que um empregado trabalha em N projetos e um projeto tem N empregado.

Percebemos que ela está na 1FN, pois todos seus atributos são monovalorados.

TB_FUNCIONARIO	
# *	id_empregado
*	nome
*	cod_cargo
*	nome_cargo
# *	cod_projeto
*	nome_projeto
*	horas

ID_EMPREGADO	NOME	COD_CARGO	NOME_CARGO	COD_PROJETO	Nome_projeto	DATA_PROJ	HORAS
120	João	1	Programador	1	Imobiliária	17/07/1995	37
120	João	1	Programador	8	Hotelaria	12/1/1996	12
121	Hélio	1	Programador	1	Imobiliária	17/07/1995	45
121	Hélio	1	Programador	8	Hotelaria	12/1/1996	21
121	Hélio	1	Programador	12	Vendas	21/03/1996	107
270	Gabriel	2	Analista	8	Hotelaria	12/1/1996	10
270	Gabriel	2	Analista	12	Vendas	21/03/1996	38
273	Silva	3	Projetista	1	Imobiliária	17/07/1995	22
274	Abraão	2	Analista	13	Estoque	21/03/1996	31

Instâncias (tuplas)

1ª Forma Normal (1FN)

Conclusão da 1FN

- ✚ Uma tabela normalizada na 1FN não evita anomalias de inclusão, alteração e remoção.
- ✚ É necessário um nível mais apurado de normalização, aplicando outras formas normais (2FN e 3FN, por exemplo).
- ✚ Para continuar normalizando uma tabela que está na primeira forma normal, são necessários aplicar as regras das próximas Formas Normais, o que implica na criação de novas entidades

2ª Forma Normal (2FN)

Uma relação está na 2FN se, e somente se, estiver na 1FN e cada atributo não-chave for dependente da chave primária inteira, isto é, cada atributo não-chave não poderá ser dependente de apenas parte da chave..

Regras:

1. Estar na 1FN
2. Não possui dependência parcial da chave

*Aplica-se apenas a **tabelas com chaves primárias compostas**. Nestes casos, se um atributo depende apenas de uma parte da chave primária, então esse atributo deve ser colocado em outra tabela*

Passando para 2FN

EMPREGADO { id_empregado, cod_projeto, nome, cod_cargo, cargo, nome_proejto, data_proj, horas }

1 FN

Analizando os atributos

id_empregado → nome

id_empregado → cod_cargo

cod_prjeto → data_proj

Cod_projeto → nome_proejto

cod_cargo → nome_cargo

id_empregado, cod_projeto → horas

ou

id_empregado → nome

id_empregado → cod_cargo → nome_cargo

cod_prjeto → data_proj

cod_projeto → nome_projeto

id_empregado, cod_projeto → horas

Normalizando

EMPREGADO { id_empregado, nome, cod_cargo, nome_cargo }

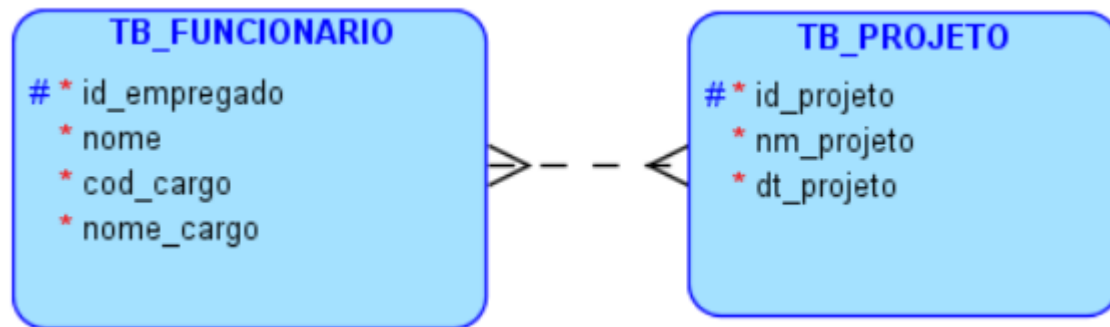
PROJETO { cod_projeto, nome_projeto, data_projeto }

ALOCACAO { id_empregado, cod_projeto, horas }

2 FN

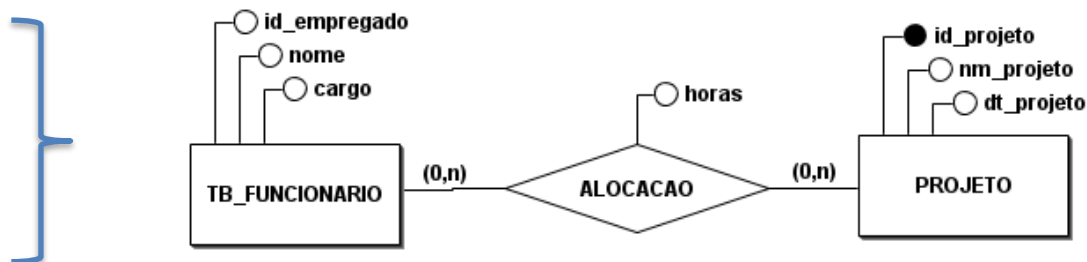
2ª Forma Normal (2FN)

Notem que ALOCACAO é a associação de FUNCIONARIO com PROJETO e em todas as entidades, os atributos dependem da chave primária como um todo

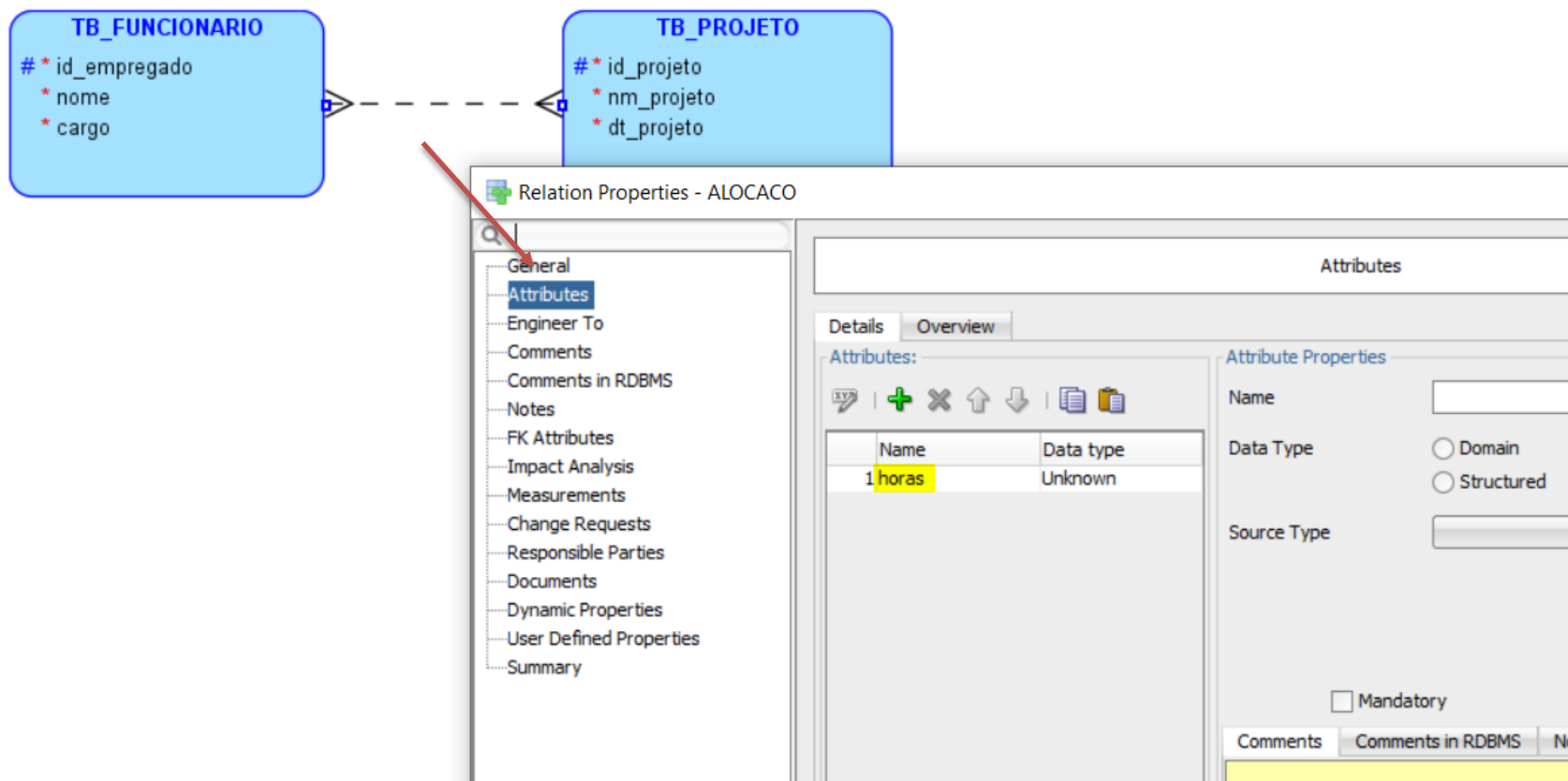


→ Ou seja, surge do relacionamento M:N. A quantidade de horas que cada funcionário atua em cada projeto é um atributo do relacionamento

Seria assim na notação
De Peter-Chen



No data Modeler



2ª Forma Normal (2FN)

Exemplo das três “tabelas” geradas pela Normalização

EMPREGADO

ID_EMPREGADO	NOME	COD_CARGO	NOME_CARGO
120	João	1	Programador
121	Hélio	1	Programador
270	Gabriel	2	Analista
273	Silva	3	Projetista
274	Abraão	2	Analista

PROJETO

COD_PROJETO	Nome_projeto	DATA_PROJ
1	Imobiliária	17/07/1995
8	Hotelaria	12/1/1996
12	Vendas	21/01/1996
13	Estoque	15/05/2000

ALOCACAO

ID_EMPREGADO	COD_PROJETO	HORAS
120	1	37
120	8	12
121	1	45
121	8	21
121	12	107
270	8	10
270	12	38
273	1	22
274	13	31

| Anomalias da 2FN

Já está bem melhor, mas ainda podemos ter anomalias quando uma entidade está na 2FN

Do exemplo anterior temos:

- Anomalia de Inclusão:** Só se pode criar um novo cargo se houver um empregado designado para ele
- Anomalia de Atualização:** Se um cargo muda de nome, será necessário alterar todas as tuplas que este nome aparece.
- Anomalia de Remoção:** Se removermos um empregado que ocupa unicamente um cargo, as informações deste cargo serão perdidas.

3ª Forma Normal (3FN)


Uma relação está na 3FN se, e somente se, estiver na 2FN e cada atributo não-chave não possuir dependência transitiva.

Regras:

1. Estar na 2FN
2. Não existir dependência indireta

Exemplo:

EMPREGADO{ id_empregado, nome, cod_cargo, nome_cargo }



The diagram shows two red curved arrows. The first arrow starts from the underlined attribute 'id_empregado' and points to 'cod_cargo'. The second arrow starts from 'cod_cargo' and points to 'nome_cargo', illustrating a transitive dependency.

→ Note que o atributo nome_cargo é dependente transitivamente da chave primária, logo, a relação **NÃO está na 3FN**.



Passando para a 3FN

- ✓ Geração de novas tabelas de tal maneira que sejam removidas as dependência indiretadas.

Passando para 3FN

EMPREGADO{ id_employado, nome, cod_cargo, nome_cargo }

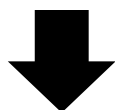
2 FN

id_employado → nome
id_employado → cod_cargo
cod_cargo → nome_cargo

ou

id_employado → nome
id_employado → cod_cargo → nome_cargo

Dependencia
transitiva



Normalizando

EMPREGADO{ id_employado, nome, cod_cargo }
CARGO { cod_cargo, nome_cargo }

3 FN

Tabelas na 3FN

Após o processo de normalização, consegue-se as seguintes relações na terceira forma normal (3FN)

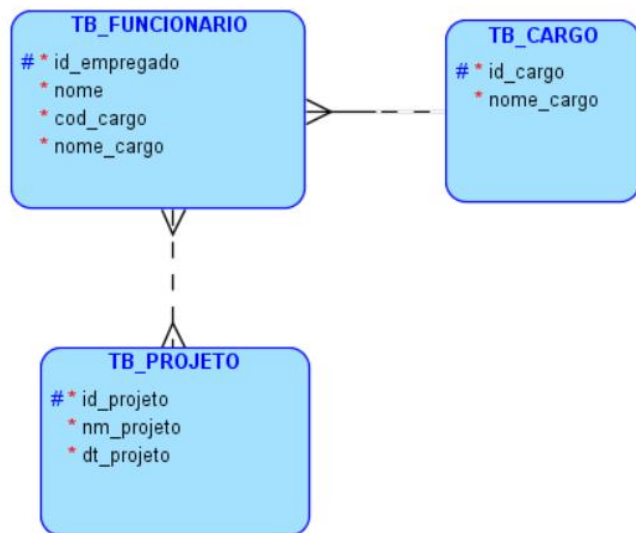
EMPREGADO{ id_employado, nome, cod_cargo}

CARGO {cod_cargo, nome_cargo}

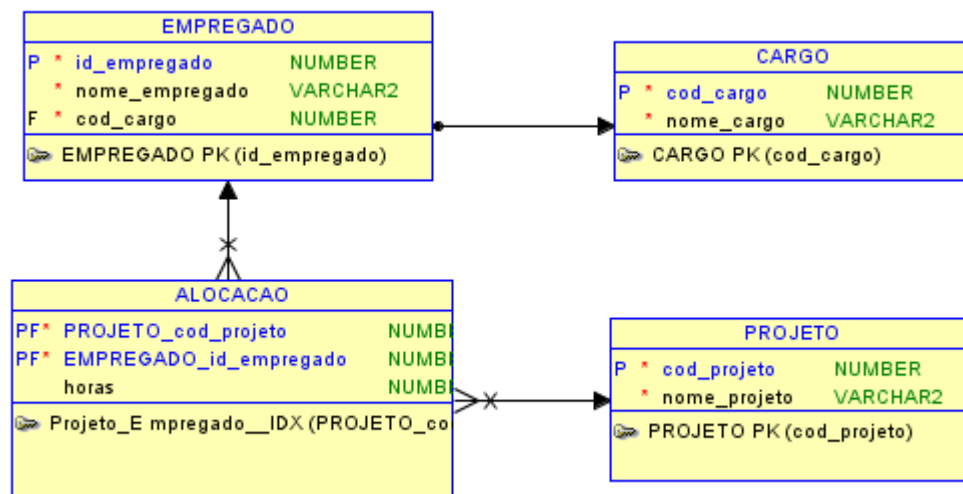
PROJETO {cod_projeto, nome_proeito, data_projeto}

ALOCACAO { id_employado, cod_projeto, horas }

Modelo Relacional Lógico



Modelo Relacional



Tabelas na 3FN

Não vemos mais
anomalias

EMPREGADO

ID_EMPREGADO	NOME	COD_CARGO
120	João	1
121	Hélio	1
270	Gabriel	2
273	Silva	3
274	Abraão	2

PROJETO

COD_PROJETO	Nome_projeto	DATA_PROJ
1	Imobiliária	17/07/1995
8	Hotelaria	12/1/1996
12	Vendas	21/01/1996
13	Estoque	15/05/2000

CARGO

COD_CARGO	NOME_CARGO
1	Programador
2	Analista
3	Projetista

ALOCAÇÃO

ID_EMPREGADO	COD_PROJETO	HORAS
120	1	37
120	8	12
121	1	45
121	8	21
121	12	107
270	8	10
270	12	38
273	1	22
274	13	31

Conclusão

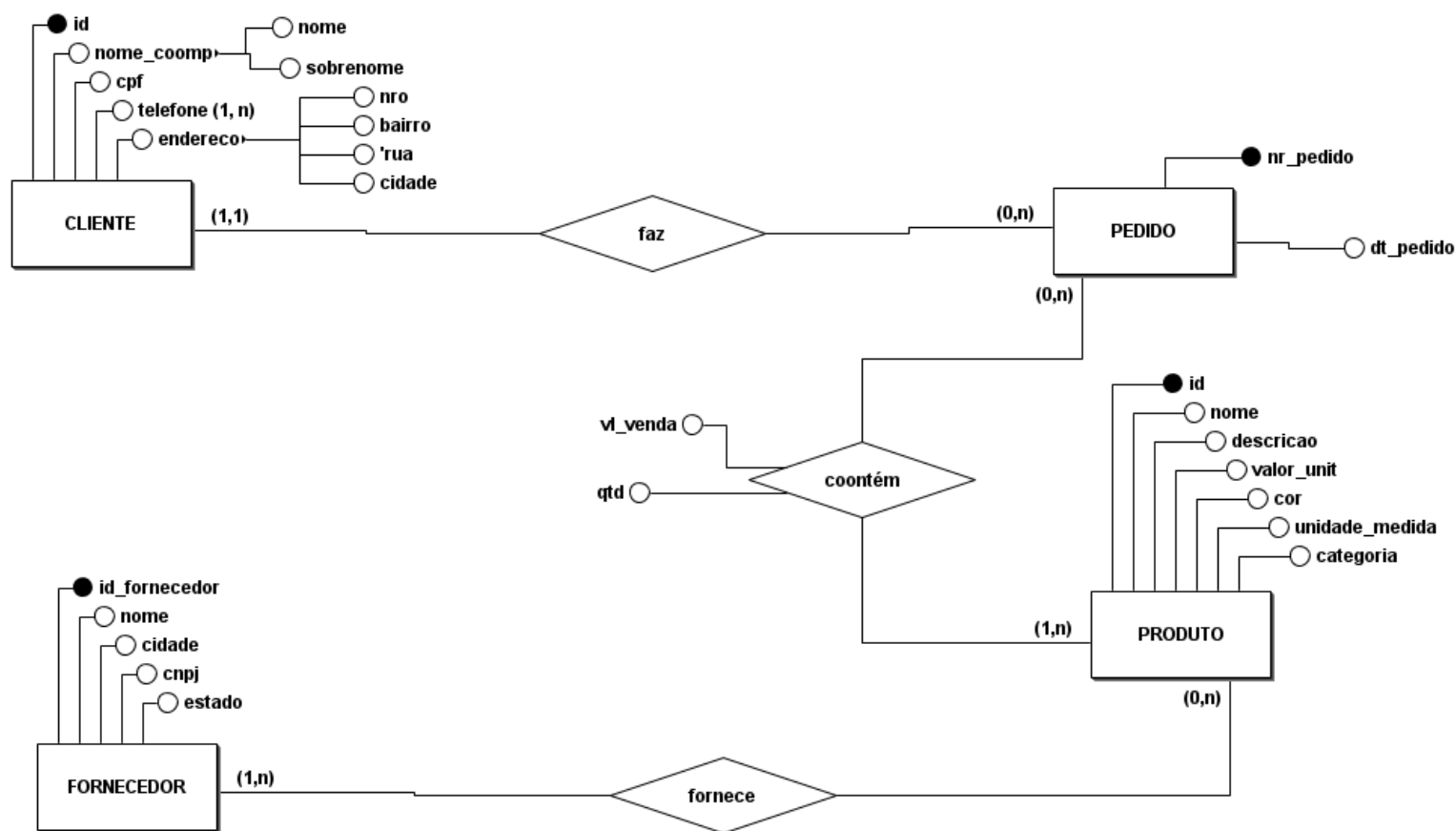
- ✚ Uma relação está em 3FN se todas as colunas da tabela são funcionalmente dependentes da chave inteira e nada além da chave
- ✚ A 3FN elimina as características mais potencialmente indesejáveis dos dados que estão em 2FN ou 1FN, evitando grande parte das anomalias possíveis.
- ✚ Existem outros casos especiais que requerem mais níveis de normalização: Boyce-Codd, 4FN e 5FN. Consultem bibliografias a respeito para mais detalhes sobre elas

Dúvidas



Hands on

No MER abaixo (usado em exemplo de aulas anteriores), podemos notar que existem entidades que não estão na 3FN. Vamos melhorá-lo e passar todas as entidades para 3FN!



Avaliação de Checkpoint

- Nossa segunda avaliação de checkpoint é prática
- Você deverá projetar o modelo lógico de um banco de dados para um pet shop (PAS – *Pet Application System*)
- O trabalho é em DUPLA.
- Vide documento fornecido pelo professor



■ Próximo Capítulo

Aspectos avançados do MER

Alguns projetos podem requerem aspectos mais avançados como associações (entidades associativas), especialização ou generalização



REFERÊNCIAS



NAVATHE, Shamkant B., ELMASRI, Ramez E.
Sistemas de banco de dados. São Paulo:
Pearson, 2005.

[http://pt.wikipedia.org/wiki/Normaliza%C3%A7%C3%A3o de dados](http://pt.wikipedia.org/wiki/Normaliza%C3%A7%C3%A3o_de_dados)

Complementar

PUGA, Sandra; FRANÇA Edson; GOYA Milton.
Banco de Dados – Implementação em SQL
PL/SQL e Oracle 11g. São Paulo: Pearson, 2014

Copyright © Prof. Luciano Melo

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).