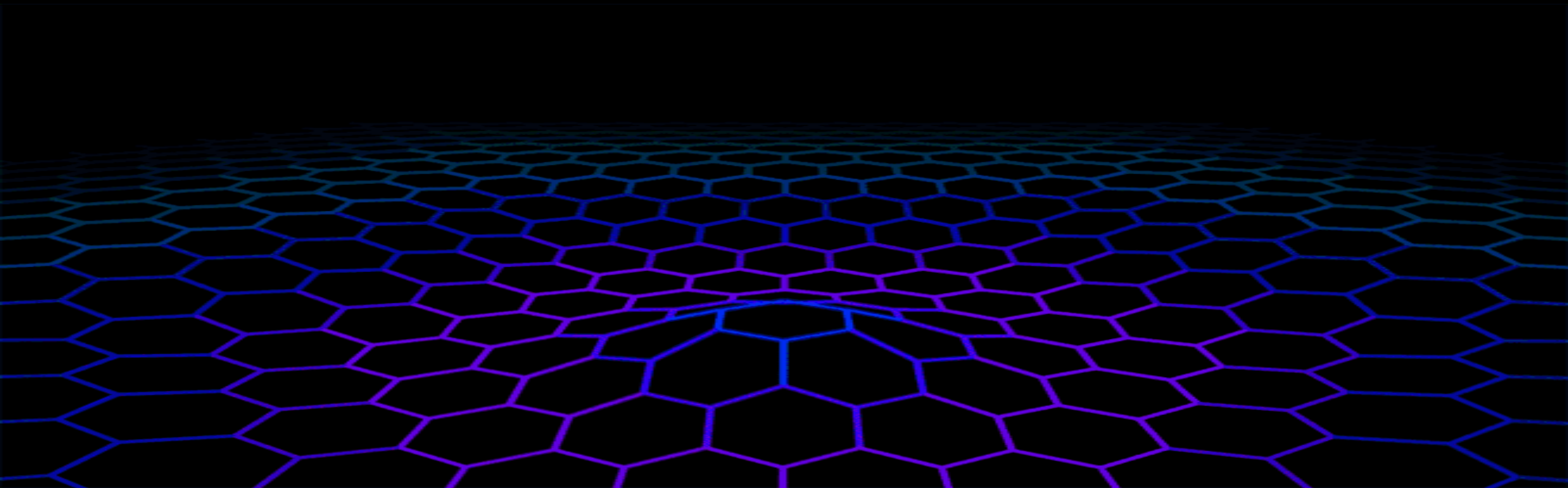


# Banco de Dados I



# Normalização



***EU SOU NORMAL***

***SARAIVA (FRANCISCO MILANI)***

# NORMALIZAÇÃO

Normalização de dados é o processo formal e passo a passo que examina os atributos de uma entidade, com o objetivo de evitar anomalias observadas na inclusão, exclusão e alteração de registros.

*Fonte: <http://www.luis.blog.br/normalizacao-de-dados-e-as-formas-normais.aspx>*



# NORMALIZAÇÃO

A normalização tem como objetivo modificar o conjunto de tabelas obtido, por transformação do modelo conceitual, num outro conjunto de tabelas equivalente **menos redundante** e **mais estável**.

# NORMALIZAÇÃO

- É um processo de depuração de um modelo de dados para reduzir sua redundância e aumentar sua estabilidade.
- Consiste em analisar o modelo e através de regras formais, reestruturar possíveis tabelas e atributos, reduzindo assim redundâncias e permitindo o crescimento do BD com o mínimo de efeito colateral.
- Consiste em diminuir redundância e anomalias de inserção, atualização e deleção.

# NORMALIZAÇÃO

- Consiste em analisar relações para satisfazer requisitos cada vez mais rigorosos acarretando agrupamentos cada vez melhores, mais estáveis e seguros.
- Realiza-se uma série de testes para certificar se a relação está ou não em uma determinada forma normal. O processo consiste em certificar e decompor.
- Fundamentado no conceito de **Dependência Funcional**.

# VANTAGENS

## Estabilidade do Modelo Lógico

- Ex.: Itens de repetição

## Integridade

## Economia

- Custo de manipulação dos dados. Backup, restauração, atualização (estr. adicionais)

## Fidelidade ao Ambiente Observado



# NORMALIZAÇÃO

Um modelo ER normalizado é convertido facilmente para um Banco de Dados relacional em tempo de projeto.

A **terceira forma normal** geralmente é aceita como boa para projeto de Banco de Dados sem redundância.

Existem formas normais de nível maior, mas que geralmente não são usadas.

# NORMALIZAÇÃO

Processo utilizado para:

- evitar a ocorrência das anomalias de atualização
- otimizar o processo de armazenamento de informações reduzindo a redundância e aumentando a confiabilidade do sistema

Este processo de projeto transforma um conjunto de especificações de entidades em outro conjunto, sendo que:

- Semântica original é preservada
- Dependências parciais e transitivação eliminadas
- Redundância de dados é minimizada

# Formas normais

1 FN (1o Forma Normal)

2 FN (2o Forma Normal)

**3 FN (3o Forma Normal) “Normalizado”**

---

4 FN (4o Forma Normal)

FNBC (Forma Normal de Boyce e Codd)

(Raymond F. **Boyce** e Edgar F. **Codd**)

5 FN (5o Forma Normal)

# NORMALIZAÇÃO

- Processo iterativo
- Processo decomposto em vários níveis de normalização, chamados de segunda forma normal, terceira forma normal e assim por diante
- Cada forma normal é mais restritiva que a anterior
- A forma normal de ordem **N + 1** está na forma normal de ordem **N** e não possui uma característica “não desejável” que a forma normal de ordem **N** possui
- A forma normal de ordem **N + 1** é mais desejável do que a forma normal de ordem **N**
- O processo de normalização é um processo de redução da redundância.

# MODELO NÃO NORMALIZADO

<u>codFornecedor</u>	nomeFornecedor	Tel1	Tel2	Endereco	<u>codProduto</u>	nomeProduto	precoUnitario	qtdPedida
1	Treichel	3232	5454	Rua X	100	Sabonete	6,00	50
1	Treichel	3232	5454	Rua X	200	Saboneteira	8,00	30
1	Treichel	3232	5454	Rua X	300	Talco	5,00	40
2	Krolow	6677	8899	Av. Y	100	Sabonete	6,00	30
2	Krolow	6677	8899	Av. Y	200	Saboneteira	8,00	15

## Problemas de Exclusão:

Caso sejam deletadas todas as solicitações de um fornecedor, seus dados cadastrais também serão apagados.

# MODELO NÃO NORMALIZADO

<u>codFornecedor</u>	nomeFornecedor	Tel1	Tel2	Endereco	<u>codProduto</u>	nomeProduto	precoUnitario	qtdPedida
1	Treichel	3232	5454	Rua X	100	Sabonete	6,00	50
1	Treichel	3232	5454	Rua X	200	Saboneteira	8,00	30
1	Treichel	3232	5454	Rua X	300	Talco	5,00	40
2	Krolow	6677	8899	Av. Y	100	Sabonete	6,00	30
2	Krolow	6677	8899	Av. Y	200	Saboneteira	8,00	15

## Problemas de **Inserção**:

- Só é possível inserir um novo fornecedor quando o mesmo vender produtos;
- Só é possível inserir um novo produto quando o mesmo for vendido por um fornecedor.

# MODELO NÃO NORMALIZADO

<u>codFornecedor</u>	nomeFornecedor	Tel1	Tel2	Endereco	<u>codProduto</u>	nomeProduto	precoUnitario	qtdPedida
1	Treichel	3232	5454	Rua X	100	Sabonete	6,00	50
1	Treichel	3232	5454	Rua X	200	Saboneteira	8,00	30
1	Treichel	3232	5454	Rua X	300	Talco	5,00	40
2	Krolow	6677	8899	Av. Y	100	Sabonete	6,00	30
2	Krolow	6677	8899	Av. Y	200	Saboneteira	8,00	15

## Problemas de **Atualização**:

- Para atualizar o endereço do fornecedor, todos os registros desse fornecedor deverão ser atualizados.
- Para atualizar o preço do produto, todos os registros desse produto deverão ser atualizados.

# MODELO NÃO NORMALIZADO

**A normalização permite eliminar atributos:**

- Com mais de um valor
- Duplicados ou repetidos
- Que contém dados derivados de outros atributos



# Dependências

## **Dependência Funcional** (Dependência Funcional Total/Completa)

Quando um atributo não identificador depende do(s) atributo(s) identificador(es).

## **Dependência Funcional Parcial**

Quando um atributo não identificador depende de parte dos atributos identificadores.

## **Dependência Funcional Transitiva**

Quando um atributo não identificador depende de outro atributo também não identificador.

# Dependência Funcional

Um relacionamento entre dois ou mais atributos de forma que o valor de um atributo identifique o valor para cada um dos outros atributos, ou seja, um atributo está relacionado a outro.

CLIENTE
<u>(PK) codigo</u>
nomeCliente
tipoLogradouro
logradouro
numero
complemento
bairro
cidade
UF

Para cada **codigo** temos somente um valor para **nomeCliente**, **tipoLogradouro**, **logradouro**, **numero**, **complemento**, **bairro**, **cidade** e **UF**.

Os atributos **nomeCliente**, **tipoLogradouro**, **logradouro**, **numeroComplemento**, **bairro**, **cidade** e **UF** são funcionalmente dependentes de **codigo**.

# Dependência Funcional Parcial

<u>(PK) matriculaAluno</u>	<u>(PK) semestre</u>	<u>(PK) codUnidade</u>	nomeUnidade	conceito
8784321	2	11	Banco de Dados 1	C
8784321	1	15	IoT	B
8784321	1	10	Organização de Computadores	A
8784321	1	11	Banco de Dados 1	B

O campo **nomeUnidade** depende somente de **codUnidade** e não depende de **matriculaAluno** e **semestre**.

# Dependência Funcional Transitiva

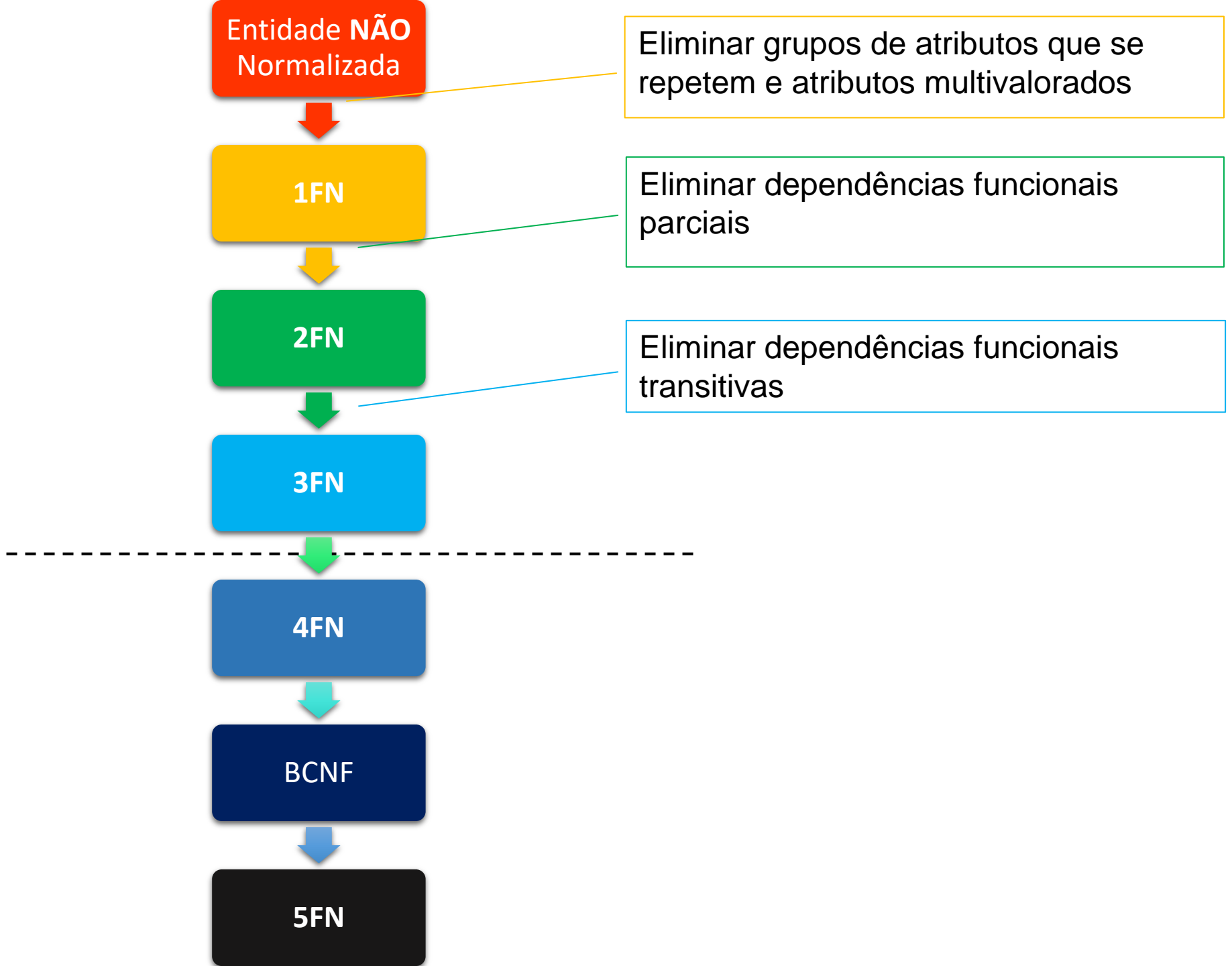
<u>(PK) matricula</u>	nomeFuncionario	codCargo	nomeCargo	salarioCargo
4432	Edécio	11	Professor	R\$ 985,00
3321	Angelo	22	Secretário	R\$ 400,00
2213	Cícero	33	Advogado	R\$ 9.999,99
1124	Kelly	44	Analista de Sistemas	R\$ 8.888,88
8940	Gladimir	11	Professor	R\$ 985,00

O campo matrícula determina apenas os atributos **nomeFuncionario** e **codCargo**.

O campo **codCargo** (que não é chave primária) determina **nomeCargo** e o **salarioCargo**.

Os campos **nomeCargo** e o **salarioCargo** não dependem diretamente da chave (**matricula**).

# PASSOS



# 1FN – Primeira Forma Normal

- Uma entidade está na primeira forma normal se não tem atributos com mais de um valor, nem atributos que ocorrem mais de uma vez.
- Uma relação está 1FN se e somente se todos os seus domínios só contêm valores atômicos;
- O modelo relacional exige que as relações estejam pelo menos na 1FN

## **Características:**

- Estrutura tabular;
- Princípio da atomicidade;
- Proibição de campos multivalorados.

# 1FN – Primeira Forma Normal

Os atributos da tabela não contêm grupos de repetição (tabelas aninhadas)

A tabela a seguir **NÃO** está na 1FN:

PROJETO						
<u>CodProj</u>	DescProj	<u>CodFunc</u>	NomeFunc	CargoFunc	SalFunc	DtInicio
11	Alfa	1001	Antonio	Analista Sr	1800	02/01/2005
		1004	Daniela	Analista Pl	1200	05/01/2005
12	Beta	1003	Claudio	Analista Sr	1800	10/02/2005

As tabelas a seguir estão na 1FN:

PROJETO	
<u>CodProj</u>	DescProj
11	Alfa
12	Beta

PROJFUNC					
<u>CodProj</u>	<u>CodFunc</u>	NomeFunc	CargoFunc	SalFunc	DtInicio
11	1001	Antonio	Analista Sr	1800	02/01/2005
11	1004	Daniela	Analista Pl	1200	05/01/2005
12	1003	Claudio	Analista Sr	1800	10/02/2005

# 1FN – Primeira Forma Normal

A tabela a seguir **NÃO** está na 1FN:

Tabela: Cliente			
ID	CPF	Nome	Telefone
1	987654321	José Antônio	(31) 3333-4444
2	987654321	José Antônio	(31) 9999-8888
3	123456789	Carlos Alberto	(31) 8979-5969
4	512346789	Ricardo Roberto	(31) 8889-6325

As tabelas a seguir estão na 1FN:

Tabela: Cliente		
ID	CPF	Nome
1	987654321	José Antônio
3	123456789	Carlos Alberto
4	512346789	Ricardo Roberto

Tabela: Telefone		
ID Telefone	ID Cliente	Telefone
1	1	(31) 3333-4444
2	1	(31) 9999-8888
3	3	(31) 8979-5969
4	4	(31) 8889-6325





# A SUA CARA

QUANDO O PROFESSOR TÁ EXPLICANDO MATÉRIA NOVA

# 2FN – Segunda Forma Normal

- Uma entidade está na segunda forma normal se está na primeira forma normal e todos os seus atributos não identificadores são dependentes do atributo identificador da entidade.
- Uma relação está na 2 FN se e somente se está na 1FN e todos os atributos não chave são **totalmente** dependentes da chave primária;
- Diz respeito às chaves primárias compostas;
- Se a chave primária de uma relação não é composta e a relação está na 1FN, ela está também na 2FN;
- Uma relação que está na 1 FN pode não estar na 2FN se sua chave for composta.

# 2FN – Segunda Forma Normal

Condição: Chave Primária (PK) composta por mais de uma coluna

Todas as colunas que não fazem parte da PK dependem de todas as colunas que compõem a PK.

A tabela a seguir **NÃO** está na 2FN:

PROJFUNC					
<u>CodProj</u>	<u>CodFunc</u>	NomeFunc	CargoFunc	SalFunc	DtInicio
11	1001	Antonio	Analista Sr	1800	02/01/2005
11	1004	Daniela	Analista Pl	1200	05/01/2005
12	1003	Claudio	Analista Sr	1800	10/02/2005

Passando para a 2FN:

FUNCIONARIO			
<u>CodFunc</u>	NomeFunc	CargoFunc	SalFunc
1001	Antonio	Analista Sr	1800
1004	Daniela	Analista Pl	1200
1003	Claudio	Analista Sr	1800

PROJFUNC		
<u>CodProj</u>	<u>CodFunc</u>	DtInicio
11	1001	02/01/2005
11	1004	05/01/2005
12	1003	10/02/2005

# 3FN – Terceira Forma Normal

- Uma entidade está na terceira forma normal se está na primeira e na segunda forma normal e não contém atributos não identificadores dependentes de outros atributos não identificadores
- Uma relação está na 3FN se e somente se ela está na 2FN e seus atributos não chave são mutuamente independentes;
- Os atributos são considerados mutuamente independentes quando nenhum atributo é funcionalmente dependente de nenhuma combinação dos outros, exceto da chave primária;
- A título de simplificação a terceira forma normal considera que a relação terá somente uma chave candidata, ou seja, a chave primária.

# 3FN – Terceira Forma Normal

- Os atributos `cod_cor` e `cor` são dependentes um do outro;
- Dependência indesejada durante as atualizações de dados;
- Não está na 3FN.

PRODUTO	
<u>codProduto</u> (PK)	
nomeProduto	
codCor	
nomeCor	
peso	

# 3FN – Terceira Forma Normal

Passos para normalização da relação:

- Eliminar o atributo `cod_cor`;
- Criar uma relação distinta para as informações sobre cor.

PRODUTO
<u>codProduto</u> (PK)
nomeProduto
codCor
peso

COR
<u>codCor</u> (PK)
nomeCor

# 3FN – Terceira Forma Normal

Não há dependências funcionais transitivas.

Cada coluna não PK depende DIRETAMENTE da PK.

A tabela a seguir **NÃO** está na 3FN:

FUNCIONARIO			
<u>CodFunc</u>	NomeFunc	CargoFunc	SalFunc
1001	Antonio	Analista Sr	1800
1004	Daniela	Analista Pl	1200
1003	Claudio	Analista Sr	1800

CodFunc → CargoFunc → SalFunc



Dependência Funcional Transitiva

Passando para a 3FN:

FUNCIONARIO		
<u>CodFunc</u>	NomeFunc	CargoFunc
1001	Antonio	Analista Sr
1004	Daniela	Analista Pl
1003	Claudio	Analista Sr

CARGOSALARIO	
<u>CargoFunc</u>	SalFunc
Analista Sr	1800
Analista Pl	1200

# RESUMÃO

## PRIMEIRA FORMA NORMAL

Uma entidade está na primeira forma normal se não tem atributos com mais de um valor, nem atributos que ocorrem mais de uma vez.



# RESUMÃO

## SEGUNDA FORMA NORMAL

- Uma entidade está na segunda forma normal se está na primeira forma normal e todos os seus atributos não identificadores são dependentes do atributo identificador da entidade.

# RESUMÃO

## TERCEIRA FORMA NORMAL

- Uma entidade está na terceira forma normal se está na primeira e na segunda forma normal e não contém atributos não identificadores dependentes de outros atributos não identificadores Observações
- Um modelo de E x R normalizado é convertido facilmente para um Banco de Dados relacional em tempo de projeto.
- A terceira forma normal geralmente é aceita como boa para projeto de Banco de Dados sem redundância.

# 4FN – Quarta Forma Normal

“Uma tabela está na 4FN se ela estiver na 3FN e não possuir DFs multivaloradas”

- tabelas com DFs multivaloradas devem ser desmembradas em tabelas que não possuam tais DFs

Tabelas que possuem CP composta por um ou dois atributos, ou que possuem atributos não chave estão automaticamente na 4FN

# 4FN – Quarta Forma Normal

## 3FN

**Estantes** (número, capacidade)

**Livros** (ISBN, título, ano)

**Autores** (codAutor, nome, nacionalidade)

**DistribuiçãoLivrosAutor**(número, ISBN, codAutor)

**DF Multivalorada:** ISBN -> codAutor



## 4FN

**Estantes** (número, capacidade)

**Livros** (ISBN, título, ano)

**Autores** (codAutor, nome, nacionalidade)

**DistribuiçãoLivros**(número, ISBN)

**Autoria**(ISBN, codAutor)

# Boyce and Codd Normal Form (BCNF)

- Uma relação está na BCNF se e somente se todo determinante é chave candidata;
- Um determinante é qualquer atributo do qual algum outro atributo é funcionalmente dependente;
- As anomalias resolvidas pela BCNF aparecem em relações com as seguintes características:
  - Múltiplas chaves candidatas;
  - Chaves candidatas são compostas;
  - Chaves candidatas possuem ao menos um atributo em comum.

# Boyce and Codd Normal Form (BCNF)

## Exemplo

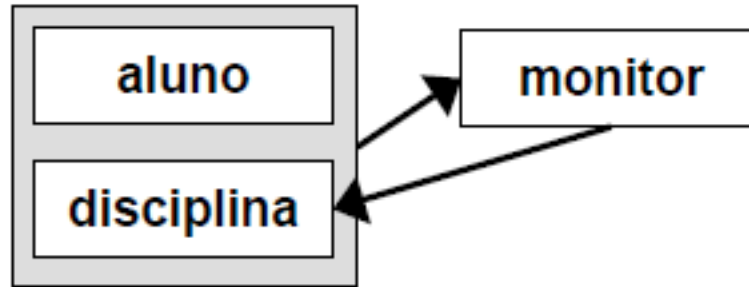
- Cada disciplina pode ter vários monitores
- Um monitor só monitora uma disciplina

<u>MONITOR</u>
aluno
disciplina
monitor

aluno	disciplina	monitor
Aline	Física II	Mário
Aline	Lógica	Marcos
Aline	Pesquisa	Ana
Carlos	Física II	Paula
João	Física II	Mário

# Boyce and Codd Normal Form (BCNF)

Dependências funcionais



**Está na 3FN, mas não está na BCNF.**

# Boyce and Codd Normal Form (BCNF)

## Solução

ALUNO_MONITOR	
aluno	monitor
Aline	Mário
Aline	Marcos
Aline	Ana
Carlos	Paula
João	Mário

MONITOR_DISCIPLINA	
A	
monitor	displina
Mário	Física II
Marcos	Lógica
Ana	Pesquisa
Paula	Física II

**Atende à BCNF.**



Crie os modelos lógicos (normalizados)  
para os dois enunciados a seguir

# Clube Esmeraldinos

O Clube Esmeraldinos é uma instituição sem fins lucrativos que pretende informatizar o seu serviço de secretaria.

O clube possui nas suas instalações: quatro quadras de tênis, uma piscina, um campo de futebol e um ginásio.

As suas atividades principais são o ensino das modalidades de tênis, ginástica e natação, além do aluguel dos suas áreas de esporte.

A secretaria do Esmeraldinos regista sempre que se inscreve um novo sócio os seguintes dados: código de sócio, nome, data de admissão, RG, data de nascimento, endereço e telefone.

Um sócio pode inscrever-se em qualquer modalidade ministrada (tênis, ginástica e natação) desde que haja vagas no horário pretendido e pagando por isso uma determinada mensalidade.

A secretaria aceita ainda marcações para qualquer espaço esportivo desde que o local esteja disponível, sendo o pagamento efetuado em simultâneo com a marcação.

Qualquer pagamento recebido origina o preenchimento e emissão de um recibo.

# Empresa “Sou usado mas sou limpinho”

A empresa de venda de automóveis usados “Sou usado mas sou limpinho”, pretende implementar um sistema para gerir a informação relativa aos reparos de oficina efetuados nos veículos dos seus clientes.

O sistema de informação deverá permitir manter um registo histórico de todos os reparos efetuados.

A empresa pretende manter dos clientes a seguinte informação: código, nome, endereço, e-mail e telefone.

A empresa pretende também saber os veículos que um dado cliente teve ou tem e as datas em que foram adquiridos.

Dos funcionários da oficina a empresa pretende manter a seguinte informação: matrícula, nome, endereço, e-mail, telefone e categoria.

O custo da hora de mão de obra depende da categoria do funcionário e é definido através de uma tabela que é atualizada regularmente.

A empresa pretende saber para cada reparo: qual o veículo, qual o cliente, a data em que foi efetuado e o custo total do reparo.

A empresa pretende também armazenar para cada reparo as peças que foram utilizadas e o seu preço, bem como o tempo de mão-de-obra gasto por cada funcionário e o respectivo custo.

A empresa pretende manter para as peças em armazém com a seguinte informação: código, descrição, custo unitário e quantidade em estoque.

Próxima aula

# Próxima aula

PostgreSQL



ORACLE®  
PL/SQL

