Página 1 de 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Curso: Bacharelado em Sistemas de Informação** | | | | |
| **Lista de Exercícios (3 Questões)** | | | | |
| **Disciplina:**  Banco de Dados II | **Semestre:**  V | **Professor:**  Pablo Freire Matos | |
| **Assunto:**  Normalização | | | | |
| **Aluno (a): Geison de Oliveira Lemos Ferreira** |  | **Turma:**  Única | **Turno:**  Noturno |

Normalize as relações de cada minimundo a seguir, de forma que todas as relações resultantes estejam na forma normal (FN) mais restrita. A normalização deve ser realizada passo a passo (a 1FN, a 2FN e a 3FN). Para cada forma normal:

1. Identifique quais as dependências funcionais que se aplicam sobre a relação (menos para a 1FN);
2. Identifique e justifique se a relação encontra-se ou não na forma normal em questão; e
3. Caso a relação não se encontre na forma normal em questão, informe qual(is) dependência(s) funcional(is) e atributo(s) fere(m) a FN e, em seguida, normalize a relação, especificando as relações originadas;
4. Por fim, mostre a relação não normalizada e as relações normalizadas identificando as chaves primárias e estrangeiras.

# Minimundo Venda de Carro

VendaCarro (cod\_cliente, nome\_cliente, telefone\_cliente, {Carro (cod\_carro, nome\_carro, preco\_carro, cod\_fabricante, nome\_fabricante, {Venda (data\_venda, preco\_venda)} )})

Regras de Negócio:

* + Um cliente possui um nome e um telefone identificado pelo código do cliente.
  + Um carro possui um nome e um preço inicial identificado pelo código do carro.
  + Um fabricante possui um nome identificado pelo código do fabricante.
  + Um carro só pode ser fabricado por um único fabricante e um fabricante pode fabricar diversos carros.
  + Um cliente pode comprar vários carros, e o mesmo carro pode ser vendido para diferentes clientes em datas distintas. Além disso, a venda pode incluir um desconto, resultando em um preço de venda diferente do preço inicial.

**R.** **1FN:** A relação vendaCarro fere a 1FN porque há atributos multivalorados (Carro e Venda).

Normalizando:

Solução: VendaCarro(cod\_cliente, nome\_cliente, telefone\_cliente)

Carro(cod\_cliente, cod\_Carro, data\_venda, nome\_carro, preco\_carro, cod\_fabricante, nome\_fabricante, preco\_venda)

*Cod\_cliente referencia VendaCarro*

**2FN:**

Dependencias funcionais:

DF1: cod\_cliente → nome\_cliente, telefone\_cliente

DF2: cod\_carro → nome\_carro, preco\_carro

DF3: cod\_fabricante → nome\_fabricante

DF4: cod\_carro → cod\_fabricante

DF5: { cod\_cliente, cod\_carro, data\_venda} → preco\_venda

Mesclando as DFs:

DF1: cod\_cliente → nome\_cliente, telefone\_cliente

DF2: cod\_carro → nome\_carro, preco\_carro, cod\_fabricante, nome\_fabricante

DF3: { cod\_cliente, cod\_carro, data\_venda} → preco\_venda

DF1 e DF2 ferem a 2FN porque os atributos não chave (nome\_cliente, telefone\_cliente, e nome\_carro, preco\_carro, cod\_fabricante, nome\_fabricante dependem de parte da chave, respectivamente cod\_cliente e cod\_carro)

Normalizando:

VendaCarro(cod\_cliente, nome\_cliente, telefone\_cliente)

Carro(cod\_carro, nome\_carro, preco\_carro, cod\_fabricante, nome\_fabricante)

Venda(cod\_cliente, cod\_carro, data\_venda, preco\_venda)

*Cod\_cliente referencia VendaCarro*

*Cod\_carro referencia Carro*

**3FN:**

Dependencia Funcional:

DF2: cod\_carro → nome\_carro, preco\_carro, cod\_fabricante, nome\_fabricante

A DF2 fere a 3FN porque o atributo não chave (nome\_fabricante) depende de outro atributo não chave (cod\_fabricante) que por sua vez depende de um chave (cod\_carro) o que resulta em uma dependencia transitiva (cod\_carro → nome\_fabricante).

Separando as DFs:

DF1: cod\_carro → nome\_carro, preco\_carro, cod\_fabricante

DF2: cod\_fabricante → nome\_fabricante

Normalizando:

Fabricante(cod\_fabricante, nome\_fabricante)

Carro(cod\_carro, cod\_fabricante, nome\_carro, preco\_carro)

*Cod\_fabricante referencia fabricante*

**Relação não normalizada:**

VendaCarro (cod\_cliente, nome\_cliente, telefone\_cliente, {Carro (cod\_carro, nome\_carro, preco\_carro, cod\_fabricante, nome\_fabricante, {Venda (data\_venda, preco\_venda)} )})

**Relações normalizadas até a 3FN:**

Fabricante(cod\_fabricante, nome\_fabricante)

Carro(cod\_carro, cod\_fabricante, nome\_carro, preco\_carro)

*Cod\_fabricante referencia fabricante*

VendaCarro(cod\_cliente, nome\_cliente, telefone\_cliente)

Venda(cod\_cliente, cod\_carro, data\_venda, preco\_venda)

*Cod\_cliente referencia VendaCarro*

*Cod\_carro referencia Carro*

# Minimundo Atendimento

Vendedor ( nro\_vend, nome\_vend, {Cliente (cpf\_cli, nome\_cli, {Atendimento (data\_atendimento)})} )

Regras de Negócio:

* + Um vendedor tem um nome e é identificado pelo número do vendedor.
  + Um cliente tem um nome que é identificado pelo CPF do cliente.
  + Um vendedor pode atender diversos clientes, e um cliente pode ser atendido por diversos vendedores, e, em ambas as ocasiões, a data de atendimento deve ser registrada. Ressalta-se que um cliente pode ser atendido pelo mesmo vendedor mais de uma vez.

**R.** **1FN:** A relação Vendedor fere a 1FN porque há atributos multivalorados (Cliente e atendimento).

Normalizando:

Solução: Vendedor(nro\_vend, nome\_vend)

Cliente(cpf\_cli, nro\_vend, data\_atendimento, nome\_cli)

*Nro\_vend referencia Vendedor*

**2FN:**

Dependencias funcionais:

DF1: nro\_vend → nome\_vend

DF2: cpf\_cli → nome\_cli

DF3: { nro\_vend, cpf\_cli, data\_atendimento} →

DF1 e DF2 ferem a 2FN porque os atributos não chave (nome\_vend, e nome\_cli) dependem de parte da chave, respectivamente nro\_vend e cpf\_cli.

Normalizando:

Vendedor(nro\_vend, nome\_vend)

Cliente(cpf\_cli, nome\_cli)

Venda(nro\_vend, cpf\_cli, data\_atendimento)

*Nro\_vend referencia vendedor*

*Cpf\_cli referencia cliente*

**\*\*\*Não fere a 3FN\*\*\***

**Relação não normalizada:**

Vendedor ( nro\_vend, nome\_vend, {Cliente (cpf\_cli, nome\_cli, {Atendimento (data\_atendimento)})} )

**Relações normalizadas até a 3FN:**

Vendedor(nro\_vend, nome\_vend)

Cliente(cpf\_cli, nome\_cli)

Venda(nro\_vend, cpf\_cli, data\_atendimento)

*Nro\_vend referencia vendedor*

*Cpf\_cli referencia cliente*

# Minimundo Certificado

Certificado (cpf\_participante, nome\_participante, data\_nascimento, {Participacao (cod\_atividade, nome\_atividade, ch\_atividade, cod\_evento, nome\_evento, sigla\_evento, ch\_participacao)})

Regras de Negócio:

* + Um participante possui um nome e uma data de nascimento e é identificado pelo CPF.
  + Uma atividade possui um nome e uma carga horária e é identificada pelo código.
  + Um evento possui um nome, uma sigla e é identificado pelo código.
  + Uma atividade só pode estar vinculada a um único evento e um evento possui várias atividades.
  + Um participante pode participar de várias atividades e uma atividade pode ter vários participantes. Ao participar de uma atividade, cada participante tem uma carga horaria de participação que é menor ou igual a carga horária da atividade.

**R.** **1FN:** A relação Certificado fere a 1FN porque há atributos multivalorados (Participacao).

Normalizando:

Solução: Certificado(cpf\_participante, nome\_participante, data\_nascimento)

Participacao(cpf\_participante, cod\_atividade, nome\_atividade, ch\_atividade, cod\_evento, nome\_evento, sigla\_evento, ch\_participacao)

*Cpf\_participante referencia Certificado*

**2FN:**

Dependencias funcionais:

DF1: cpf\_participante → nome\_participante, data\_nascimento

DF2: cod\_atividade → nome\_atividade, ch\_atividade

DF3: cod\_evento → nome\_evento, sigla\_evento

DF4: cod\_atividade → cod\_evento

DF5: { cpf\_participante, cod\_atividade, ch\_participacao} →

Mesclando as DFs:

DF1: cpf\_participante → nome\_participante, data\_nascimento

DF2: cod\_atividade → nome\_atividade, ch\_atividade, cod\_evento, nome\_evento, sigla\_evento

DF3: { cpf\_participante, cod\_atividade, ch\_participacao } →

DF1 e DF2 ferem a 2FN porque os atributos não chave (nome\_participante, data\_nascimento, e nome\_atividade, ch\_atividade, cod\_evento, nome\_evento, sigla\_evento dependem de parte da chave, respectivamente cpf\_participante e cod\_atividade)

Normalizando:

Certificado(cpf\_participante, nome\_participante, data\_nascimento)

Atividade(cpf\_participante, cod\_atividade, nome\_atividade, ch\_atividade, cod\_evento, nome\_evento, sigla\_evento)

*Cpf\_participante referencia Certificado*

Participacao(cpf\_participante, cod\_atividade, ch\_participacao)

*Cpf\_participante referencia Certificado*

*Ch\_participacao referencia Atividade*

**3FN:**

Dependencia Funcional:

DF2: cod\_atividade → nome\_atividade, ch\_atividade, cod\_evento, nome\_evento, sigla\_evento

A DF2 fere a 3FN porque os atributos não chaves (nome\_evento, sigla\_evento) dependem de outro atributo não chave (cod\_evento) que por sua vez depende de um chave (cod\_atividade) o que resulta em uma dependencia transitiva (cod\_atividade → nome\_evento, sigla\_evento).

Separando as DFs:

DF1: cod\_atividade → nome\_atividade, ch\_atividade, cod\_evento

DF2: cod\_evento → nome\_evento, sigla\_evento

Normalizando:

Evento(cod\_evento, nome\_evento, sigla\_evento)

Atividade(cpf\_participante, cod\_atividade, cod\_evento, nome\_atividade, ch\_atividade)

*Cpf\_participante referencia Certificado*

*Cod\_evento referencia Evento*

**Relação não normalizada:**

Certificado (cpf\_participante, nome\_participante, data\_nascimento, {Participacao (cod\_atividade, nome\_atividade, ch\_atividade, cod\_evento, nome\_evento, sigla\_evento, ch\_participacao)})

**Relações normalizadas até a 3FN:**

Certificado(cpf\_participante, nome\_participante, data\_nascimento)

Evento(cod\_evento, nome\_evento, sigla\_evento)

Atividade(cpf\_participante, cod\_atividade, cod\_evento, nome\_atividade, ch\_atividade)

*Cpf\_participante referencia Certificado*

*Cod\_evento referencia Evento*

Participacao(cpf\_participante, cod\_atividade, ch\_participacao)

*Cpf\_participante referencia Certificado*

*Ch\_participacao referencia Atividade*