



## Experiência Prática 3 – Modelo Relacional e Normalização

### 1. Identificação do Projeto

**Título do Projeto:**

**Tecnologia da Informação Aplicada na Gestão de Abrigo de Animais em ONGs**

### 2. Panorama e Objetivos

#### 2.1. Resumo

Gerir uma ONG, como para abrigo de animais, além de boa vontade e dedicação, requer boa tomada de decisão, administração dos recursos, visão e conformidade com as leis e boas práticas para garantir o bem-estar e proteção dos animais e continuidade de suas atividades. A tecnologia para o controle do fluxo da informação torna-se então uma ferramenta de grande auxílio.

#### 2.2. Objetivos Principais

Este trabalho apresenta a transformação para o modelo lógico normalizado a partir do modelo conceitual construído anteriormente referente a uma instituição de adoção de pets. São descritas as etapas de identificação de entidades e atributos, aplicação da primeira forma normal (1FN) com decomposição de atributos compostos e multivalorados, e a normalização até a terceira forma normal (3FN). Ao final é feita uma verificação para avaliação de conformidade da estrutura com as regras de integridade e normalização para posterior implementação utilizando SGBD.

### 3. Situação inicial (pré-normalização)

#### 3.1. Descrição

O Diagrama Entidade-Relacionamento no documento anterior apresenta entidades com atributos compostos, redundâncias, associações N:M e potenciais multivalorados, como “dados\_endereco” e “dados\_contato”.

#### 3.2 Descrição das entidades e seus atributos pelo modelo anterior

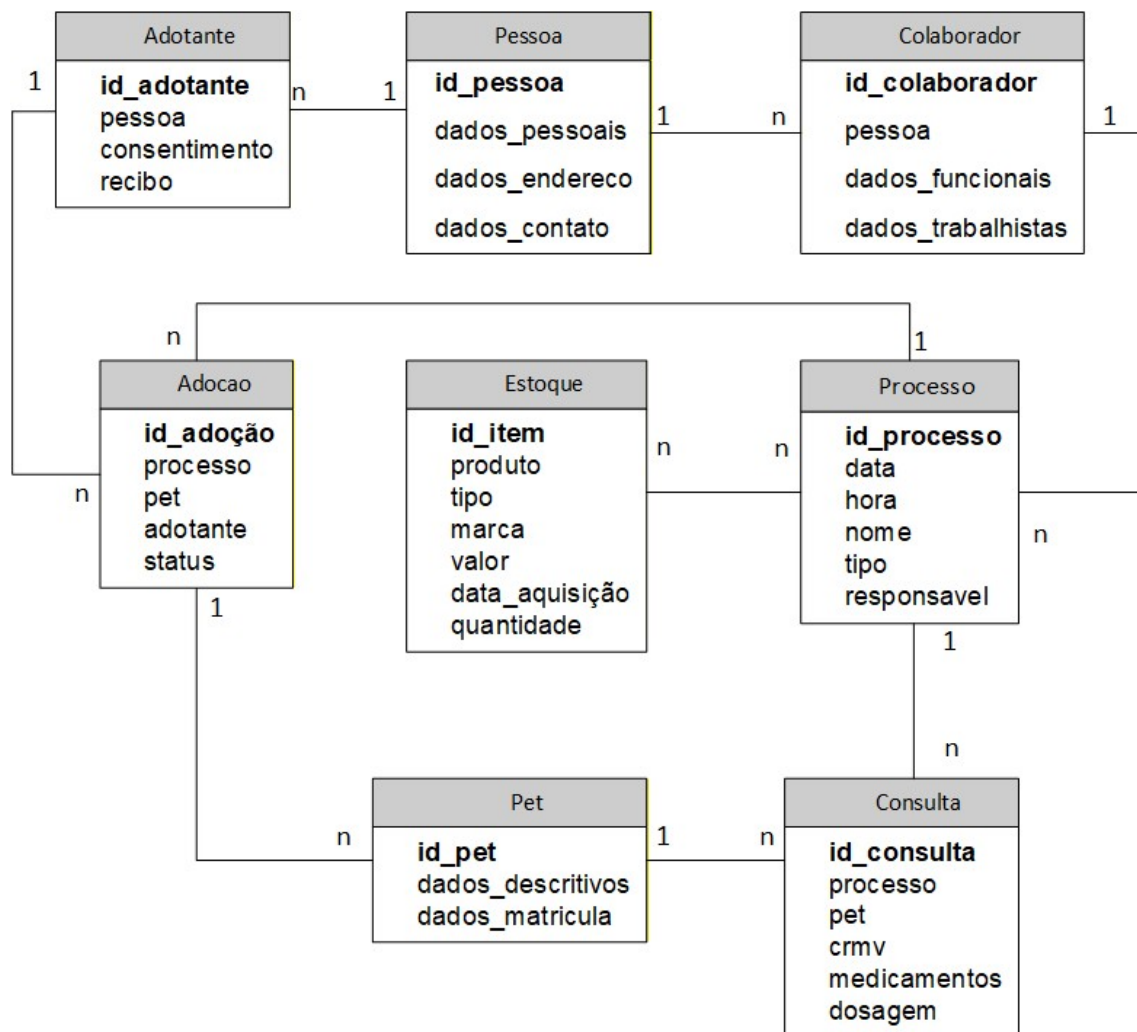
A seguir são apresentadas as definições das entidades e atributos do DER inicial (atributos-chave em negrito):

1. pessoa = {**id\_pessoa**, dados\_pessoais, dados\_endereco, dados\_contato};



2. colaborador = {**id\_colaborador**, pessoa, dados\_funcionais, dados\_trabalhistas};
3. adotante = {**id\_adotante**, pessoa, consentimento, recibo};
4. pet = {**id\_pet**, dados\_descritivos, dados\_matricula};
5. processo = {**id\_processo**, data, hora, nome, tipo, responsavel};
6. consulta = {**id\_consulta**, processo, pet, crmv, medicamentos, dosagem};
7. adoção = {**id\_adoção**, processo, pet, adotante, status};
8. estoque = {**id\_item**, Produto, tipo, marca, valor, data de aquisição, quantidade}.

### 3.3. Diagrama Entidade-Relacionamento (pré-normalização)





## 4. Normalização e framework de Conformidade com Regras de Integridade

### 4.1. Verificação: Primeira Forma Normal (1FN)

**Objetivo:** Avaliar se a estrutura de uma entidade/tabela de banco de dados está em conformidade com as regras da Primeira Forma Normal (1FN).

**Instruções:** Para cada afirmação abaixo, marque a opção que melhor descreve a tabela que está sendo analisada.

- Atende: A afirmação é totalmente verdadeira para a tabela, assinalando (V);
- Atende Parcialmente: A tabela cumpre a afirmação na maioria dos casos, mas existem exceções ou pontos de melhoria, assinalando (P);
- Não Atende: A afirmação é falsa para a tabela, assinalando (F).

	Afirmação	Entidades							
		1	2	3	4	5	6	7	8
1	Cada célula (interseção de linha e coluna) da tabela contém apenas um único valor (valor atômico).	F	F	F	F	V	V	V	V
2	A tabela não possui colunas que contenham listas, <i>arrays</i> ou conjuntos de valores.	F	F	F	F	V	V	V	V
3	Não existem grupos de colunas repetidas para armazenar itens semelhantes (ex.: Telefone1, Telefone2, Telefone3).	F	F	F	V	V	V	F	V
4	Cada linha na tabela é única e pode ser identificada por uma chave primária.	V	V	V	V	V	V	V	V
5	Os valores em uma mesma coluna são todos do mesmo tipo de dado (ex.:	F	F	F	F	V	V	V	V



	todos são textos, ou todos são números inteiros).								
6	Não é necessário decompor os dados de nenhuma coluna para obter informações individuais (ex.: uma coluna "Endereço" que armazena rua, número e cidade juntos).	F	F	F	F	V	V	V	V
7	A ordem das linhas não afeta a interpretação ou a unicidade dos dados.	V	V	V	V	V	V	V	V
8	A ordem das colunas não altera o significado dos dados armazenados na tabela.	V	V	V	V	V	V	V	V
9	A tabela possui uma chave primária claramente definida, seja ela simples (uma coluna) ou composta (múltiplas colunas).	V	V	V	V	V	V	V	V
10	A estrutura da tabela evita a necessidade de consultas complexas para extrair um único pedaço de informação de um campo multivalorado.	F	V	F	F	F	F	F	F

## 4.2. Regras aplicadas

### 4.2.1. Decomposição dos atributos compostos (detalhamento e justificativa)

A partir das definições originais, procede-se à decomposição:

Entidade 1: Pessoa (original: **id\_pessoa**, dados\_pessoais, dados\_endereco, dados\_contato)

- Decomposição:

- Atributo 1: dados\_pessoais → decompõe-se: nome, sobrenome, data\_nascimento, cpf



- Atributo 2: dados\_endereco → decompõe-se: logradouro, numero, complemento, bairro, cidade, uf, cep
- Atributo 3: dados\_contato → decompõe-se: celular, telefone, email.
- Justificativa: Separar nome em partes facilita buscas; separar endereço em componentes evita repetição de blocos de texto e permite indexação; contatos podem ser multivalorados (vários telefones) — assim, criou-se uma entidade Contato se houver necessidade de múltiplos contatos por pessoa.

Entidade 2: Colaborador (original: **id\_colaborador**, pessoa, dados\_funcionais, dados\_trabalhistas}

- Decomposição

- Atributo 1: pessoa → refere-se a pessoa (id\_pessoa FK)
- Atributo 2: dados\_funcionais → decompõe-se: cargo, setor, status\_funcional
- Atributo 3: dados\_trabalhistas → decompõe-se: tipo\_contrato, data\_admissao, salario.

- Justificativa: pessoa como chave estrangeira assegura integridade referencial e a decomposição facilita buscas.

Entidade 3: Processo (original: **id\_processo**, data, hora, nome, tipo, responsavel)

- Alterações:

- Atributo 1: data e hora → generalização em um campo único data\_hora para representar o momento
- Atributo 2: nome → nome\_processo
- Atributo 3: tipo → tipo\_processo (ex.: 'Consulta', 'Adoção', 'Limpeza')
- Atributo 4: responsavel → refere-se a colaborador (id\_colaborador FK)

- Justificativa: data e hora combinados simplificam ordenação; responsavel como FK assegura integridade referencial; tipo orienta subclasses (entidades: Consulta e Adoção).

Entidade 4: Adotante (original: **id\_adotante**, pessoa, consentimento, recibo)

- Alteração: Atributo 1: pessoa → refere-se a pessoa (id\_pessoa FK)

- Decomposição:

- Atributo 2: consentimento → decompõe-se: documento, data\_assinatura





- Atributo 3: recibo → decompõe-se: data\_recibo, valor
- Justificativa: A decomposição em entidade recibo e consentimento é para a necessidade de múltiplos documentos de consentimento (ex: orientações, cuidados e responsabilidade com pet, tratamento de dados) e recibos (pagamento de taxas se aplicáveis)

Entidade 5: Pet (original: **id\_pet**, dados\_descritivos, dados\_matricula)

- Decomposição:
- Atributo 1: dados\_descritivos → decompõe-se: nome, especie (ex: “cão”, “gato”), raca, idade, aparencia, comportamento, status\_pet (ex: “adotado”, “divulgado para adoção”, “sob cuidados”)
  - Atributo 2: dados\_matricula → decompõe-se: data\_entrada, tipo\_entrada (ex: “doação”, “resgate”), local\_origem

Entidade 6: Adoção (original: **id\_adocao**, processo, pet, adotante, status)

- Alterações:
- Atributo 1: processo → id\_processo (FK)
  - Atributo 2: pet → id\_pet (FK)
  - Atributo 3: status → atributo atômico status\_adocao (ex.: ‘Em andamento’, ‘Concluída’, ‘Cancelada’)
- Justificativa: Uma adoção pode ter múltiplos pets;

Entidade 7: Estoque {**id\_item PK**, nome\_produto, tipo, marca, valor, data\_aquisicao, quantidade}

Entidade 8: Consulta {**id\_consulta PK**, id\_processo FK, id\_pet FK, medicamento, dosagem, observação}

#### 4.2.2. Tabelas associativas

Em associações N:M observadas são criadas novas tabelas associativas:

Relacionamento Pet e Adoção (N:M) → tabela associativa Pet\_Adocao

Relacionamento Estoque e Processo (N:M) → tabela associativa Item\_processo



Na entidade Consulta, o Atributo medicamento é possivelmente multivalorado (vários medicamentos por consulta) → tabela Consulta\_Medicamento.

**4.3. Verificação: Segunda Forma Normal (2FN)**

**Objetivo:** Avaliar se a estrutura de uma entidade/tabela de banco de dados está em conformidade com as regras da Segunda Forma Normal (2FN).

**Pré-requisito:** A tabela já deve atender à Primeira Forma Normal (1FN).

**Instruções:** Para cada afirmação abaixo, marque a opção que melhor descreve a tabela que está sendo analisada.

- Atende: A afirmação é totalmente verdadeira para a tabela, assinalado com “V”.
- Atende Parcialmente: A tabela cumpre a afirmação na maioria dos casos, mas existem exceções ou pontos de melhoria, assinalado com “P”;
- Não Atende: A afirmação é falsa para a tabela, assinalado com “F”;
- N/A (Não se Aplica): A afirmação não é relevante (geralmente para tabelas com chave primária simples).

Numeração das Entidades: 1 – Pessoa; 2 – Colaborador; 3 – Adotante; 4 – Pet; 5 – Processo; 6 – Consulta; 7 – Adoção; 8 – Estoque; 9 – Endereço; 10 – Contato; 11- – Consentimento; 12 – Recibo; 13 - Pet\_adocao; 14 - item\_processo; 15 – Medica-mento e 16 - Consulta\_Medicamento.

	Afirmação	Entidade							
		1	2	3	4	5	6	7	8
1	A tabela está em conformidade com a Primeira Forma Normal (1FN).	V	V	V	V	V	V	V	V
2	Se a tabela possui uma chave primária simples (composta por	V	V	V	V	V	V	V	V







	funcional de pelo menos um atributo não-chave.								
10	A estrutura evita anomalias de atualização, inserção e exclusão relacionadas a dependências parciais da chave.	V	V	V	V	V	V	V	V

	Afirmação	Entidade							
		9	10	11	12	13	14	15	16
1	A tabela está em conformidade com a Primeira Forma Normal (1FN).	V	V	V	V	V	V	V	V
2	Se a tabela possui uma chave primária simples (composta por uma única coluna), ela automaticamente atende à 2FN.	V	V	V	V	V	V	V	V
3	A tabela possui uma chave primária composta (formada por duas ou mais colunas).	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4	Todos os atributos (colunas) que não fazem parte da chave primária dependem da chave primária em sua totalidade.	V	V	V	V	V	V	V	V
5	Não existem colunas na tabela que dependam de apenas uma parte da chave primária composta.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
6	Se uma coluna não-chave fosse movida para outra tabela junto com a parte da chave da qual	F	F	F	F	F	F	F	F



	ela depende, não haveria perda de informação.								
7	Não há redundância de dados causada por uma coluna que descreve um atributo de apenas um dos componentes da chave primária.	V	V	V	V	V	V	V	V
8	Todas as colunas não-chave descrevem o objeto ou evento identificado pela combinação completa das colunas da chave primária.	V	V	V	V	V	V	V	V
9	A remoção de qualquer coluna da chave primária composta quebraria a dependência funcional de pelo menos um atributo não-chave.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10	A estrutura evita anomalias de atualização, inserção e exclusão relacionadas a dependências parciais da chave.	V	V	V	V	V	V	V	V

#### 4.3.2. Regras aplicadas

A segunda forma normal - 2FN - exige que a tabela esteja em 1FN e que todos os atributos não-chave dependam totalmente de qualquer chave primária composta (remover dependências parciais).

#### 4.3.3 Aplicação

- Identificação de tabelas com chave composta: Pet\_Adocao, Item\_Processo e Medicamento\_Consulta.
- Verificação: Não há dependências parciais



**4.3.4. Conclusão:** O projeto já adere à 2FN após 1FN por separação de atributos e criação de entidades associativas.

**4.4. Verificação: Terceira Forma Normal (3FN)**

**Objetivo:** Avaliar se a estrutura de uma entidade/tabela de banco de dados está em conformidade com as regras da Terceira Forma Normal (3FN).

**Pré-requisito:** A tabela já deve atender à Segunda Forma Normal (2FN).

**Instruções:** Para cada afirmação abaixo, marque a opção que melhor descreve a tabela que está sendo analisada.

- Atende: A afirmação é totalmente verdadeira para a tabela, assinalado com “V”.
- Atende Parcialmente: A tabela cumpre a afirmação na maioria dos casos, mas existem exceções ou pontos de melhoria, assinalado com “P”;
- Não Atende: A afirmação é falsa para a tabela, assinalado com “F”;
- N/A (Não se Aplica): A afirmação não é relevante (geralmente para tabelas com chave primária simples).

**Numeração das Entidades:** 1 – Pessoa; 2 – Colaborador; 3 – Adotante; 4 – Pet; 5 – Processo; 6 – Consulta; 7 – Adoção; 8 – Estoque; 9 – Endereço; 10 – Contato; 11— Consentimento; 12 – Recibo; 13 - Pet\_adocao; 14 - item\_processo; 15 – Medicamento e 16 - Consulta\_Medicamento.

	Afirmação	Entidades							
		1	2	3	4	5	6	7	8
1	A tabela já atende plenamente à Segunda Forma Normal (2FN).	V	V	V	V	V	V	V	V
2	Nenhum atributo (coluna) que não faz parte da chave primária depende de outro atributo que também não faz parte da chave primária.	V	V	V	V	V	V	V	V



3	Não existem dependências transitivas na tabela (um atributo não-chave A depende de um atributo não-chave B, que por sua vez depende da chave primária).	V	V	V	V	V	V	V	V
4	Todas as colunas da tabela dependem exclusivamente da chave primária, e de nada mais.	V	V	V	V	V	V	V	V
5	Não há colunas na tabela que sejam atributos de outras colunas não-chave (ex.: em uma tabela de Pedidos, ter colunas como NomeCliente e EnderecoCliente, que são atributos do Cliente e não do Pedido).	V	V	V	V	V	V	V	V
6	Se o valor de um atributo não-chave for alterado, isso não exigirá a alteração de nenhum outro atributo não-chave na mesma tabela.	V	V	V	V	V	V	V	V
7	Cada coluna não-chave fornece uma informação sobre a "chave, a chave inteira e nada mais que a chave".	V	V	V	V	V	V	V	V
8	A estrutura evita redundância de dados onde a mesma informação (que não faz parte da chave) é repetida em múltiplas linhas porque depende de outro atributo não-chave.	V	V	V	V	V	V	V	V
9	Para inserir um novo dado em uma coluna não-chave, não é necessário primeiro ter o valor de outra coluna não-chave da qual ela depende.	V	V	V	V	V	V	V	V



10	A estrutura evita anomalias de atualização, inserção e exclusão relacionadas a dependências entre atributos não-chave.	V	V	V	V	V	V	V	V
----	--	---	---	---	---	---	---	---	---

	Afirmação	Entidades							
		9	10	11	12	13	14	15	16
1	A tabela já atende plenamente à Segunda Forma Normal (2FN).	V	V	V	V	V	V	V	V
2	Nenhum atributo (coluna) que não faz parte da chave primária depende de outro atributo que também não faz parte da chave primária.	V	V	V	V	V	V	V	V
3	Não existem dependências transitivas na tabela (um atributo não-chave A depende de um atributo não-chave B, que por sua vez depende da chave primária).	V	V	V	V	V	V	V	V
4	Todas as colunas da tabela dependem exclusivamente da chave primária, e de nada mais.	V	V	V	V	V	V	V	V
5	Não há colunas na tabela que sejam atributos de outras colunas não-chave (ex.: em uma tabela de Pedidos, ter colunas como NomeCliente e EnderecoCliente, que são atributos do Cliente e não do Pedido).	V	V	V	V	V	V	V	V
6	Se o valor de um atributo não-chave for alterado, isso não exigirá a alteração de	V	V	V	V	V	V	V	V





	nenhum outro atributo não-chave na mesma tabela.								
7	Cada coluna não-chave fornece uma informação sobre a "chave, a chave inteira e nada mais que a chave".	V	V	V	V	V	V	V	V
8	A estrutura evita redundância de dados onde a mesma informação (que não faz parte da chave) é repetida em múltiplas linhas porque depende de outro atributo não-chave.	V	V	V	V	V	V	V	V
9	Para inserir um novo dado em uma coluna não-chave, não é necessário primeiro ter o valor de outra coluna não-chave da qual ela depende.	V	V	V	V	V	V	V	V
10	A estrutura evita anomalias de atualização, inserção e exclusão relacionadas a dependências entre atributos não-chave.	<u>V</u>	V	V	V	V	V	V	V

4.4.2 Regras aplicadas

A terceira forma normal - 3FN - exige que esteja em 2FN e que nenhum atributo não-chave dependa transitivamente de uma chave primária — ou seja, remover dependências transitivas.

4.4.3 Aplicação

Após revisão, o esquema proposto encontra-se em 3FN.



## 5. Diagrama Entidade-Relacionamento (DER) – Revisado

### 5.1 Descrição:

- O diagrama abaixo apresenta o modelo lógico do minimundo.
- Todas as entidades relevantes foram identificadas.
- Os atributos principais de cada entidade estão detalhados.
- Os relacionamentos entre as entidades foram estabelecidos.
- As cardinalidades de todos os relacionamentos estão definidas.
- Aplicação e revisão das formas normais





## 6. Quadro-resumo

Descrição: comparação entre o modelo relacional antes e após o processo de normalização e aplicação de regras de integridade.

Pré-normalização	Modelo revisado
<ul style="list-style-type: none"><li>. atributos compostos com potenciais multivalorados</li><li>. apresentou redundâncias massivas</li><li>. dependências funcionais parciais</li><li>. dados inconsistentes</li><li>. relacionamentos muitos para muitos sem tabelas associativas</li></ul>	<ul style="list-style-type: none"><li>. eliminação de anomalias de inserção, atualização e exclusão</li><li>. manutenibilidade melhorada</li><li>. maior consistência</li><li>. performance melhorada</li></ul>

## 7. Conclusão

Criado por Edgar Frank Codd, o processo de normalização permite que o modelo relacional possa ser aplicado de forma ampla, eficiente e adaptada ao contexto de aplicação. O conhecimento teórico, técnico e de frameworks para estruturação, alinhados a uma boa definição do escopo do projeto, identificando detalhadamente suas características e restrições são muito importantes para a construção de um banco de dados eficiente e eficaz.

