



## Experiência Prática 3 – Documento Modelo

### 1. Identificação do Projeto

**Nome do Projeto:** Preencha com o nome do seu projeto

### 2. Descrição do Minimundo

Use a mesma descrição utilizada no Documento Modelo da Experiência Prática 1.

### 3. Entregáveis

Nesta seção, você deve incluir os seus entregáveis.

#### 3.1. 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.
- Atende Parcialmente: A tabela cumpre a afirmação na maioria dos casos, mas existem exceções ou pontos de melhoria.
- Não Atende: A afirmação é falsa para a tabela.

#### Entidade / Tabela: TODAS

(decidi aqui utilizar essa tabela para representar todas as minhas tabelas, pois, para meu conhecimento, todas atendem todos requisitos)

	Afirmação	Atende	Atende Parcialmente	Não Atende
1	Cada célula (interseção de linha e coluna) da tabela contém apenas um único valor (valor atômico).	X		



2	A tabela não possui colunas que contenham listas, <i>arrays</i> ou conjuntos de valores.	X		
3	Não existem grupos de colunas repetidas para armazenar itens semelhantes (ex.: Telefone1, Telefone2, Telefone3).	X		
4	Cada linha na tabela é única e pode ser identificada por uma chave primária.	X		
5	Os valores em uma mesma coluna são todos do mesmo tipo de dado (ex.: todos são textos, ou todos são números inteiros).	X		
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).	X		
7	A ordem das linhas não afeta a interpretação ou a unicidade dos dados.	X		
8	A ordem das colunas não altera o significado dos dados armazenados na tabela.	X		
9	A tabela possui uma chave primária claramente definida, seja ela simples (uma coluna) ou composta (múltiplas colunas).	X		
10	A estrutura da tabela evita a necessidade de consultas complexas para extrair um único pedaço de	X		



	informação de um campo multivalorado.			
--	---------------------------------------	--	--	--

### 3.2. 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.
- Atende Parcialmente: A tabela cumpre a afirmação na maioria dos casos, mas existem exceções ou pontos de melhoria.
- Não Atende: A afirmação é falsa para a tabela.
- N/A (Não se Aplica): A afirmação não é relevante (geralmente para tabelas com chave primária simples).

**Entidade / Tabela: TODAS**

(decidi aqui utilizar essa tabela para representar todas as minhas tabelas, pois, para meu conhecimento, todas atendem todos requisitos)

	Afirmação	Atende	Atende Parcialmente	Não Atende	N/A
1	A tabela está em conformidade com a Primeira Forma Normal (1FN).	X			
2	Se a tabela possui uma chave primária simples (composta por uma única coluna), ela automaticamente atende à 2FN.	X			



<b>3</b>	A tabela possui uma chave primária composta (formada por duas ou mais colunas).	X			
<b>4</b>	Todos os atributos (colunas) que não fazem parte da chave primária dependem da chave primária em sua totalidade.	X			
<b>5</b>	Não existem colunas na tabela que dependam de apenas uma parte da chave primária composta.	X			
<b>6</b>	Se uma coluna não-chave fosse movida para outra tabela junto com a parte da chave da qual ela depende, não haveria perda de informação.	X			
<b>7</b>	Não há redundância de dados causada por uma coluna que descreve um atributo de apenas um dos componentes da chave primária.	X			
<b>8</b>	Todas as colunas não-chave descrevem o objeto ou evento identificado pela combinação completa das colunas da chave primária.	X			
<b>9</b>	A remoção de qualquer coluna da chave primária composta quebraria a dependência funcional de pelo menos um atributo não-chave.	X			



10	A estrutura evita anomalias de atualização, inserção e exclusão relacionadas a dependências parciais da chave.	X			
----	--	---	--	--	--

### 3.3. 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.
- Atende Parcialmente: A tabela cumpre a afirmação na maioria dos casos, mas existem exceções ou pontos de melhoria.
- Não Atende: A afirmação é falsa para a tabela.

#### Entidade / Tabela: TODAS

(decidi aqui utilizar essa tabela para representar todas as minhas tabelas, pois, para meu conhecimento, todas atendem os mesmos requisitos)

	Afirmação	Atende	Atende Parcialmente	Não Atende
1	A tabela já atende plenamente à Segunda Forma Normal (2FN).	X		
2	Nenhum atributo (coluna) que não faz parte da chave primária depende de outro atributo que também não faz parte da chave primária.	X		



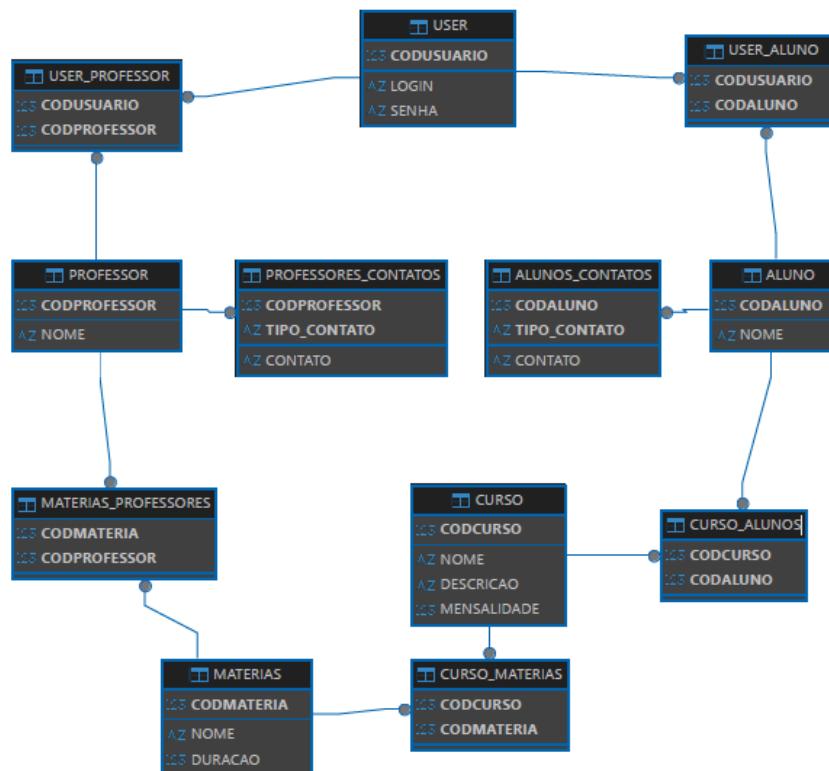
<b>3</b>	Não existem dependências transitivas na tabela (um atributo não-chave A depende de um atributo não-chave B, que por sua vez depende da chave primária).	X		
<b>4</b>	Todas as colunas da tabela dependem exclusivamente da chave primária, e de nada mais.	X		
<b>5</b>	Não há colunas na tabela que 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).	X		
<b>6</b>	Se o valor de um atributo não-chave for alterado, isso não exigirá a alteração de nenhum outro atributo não-chave na mesma tabela.	X		
<b>7</b>	Cada coluna não-chave fornece uma informação sobre a "chave, a chave inteira e nada mais que a chave".	X		
<b>8</b>	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.	X		
<b>9</b>	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.	X		
<b>10</b>	A estrutura evita anomalias de	X		

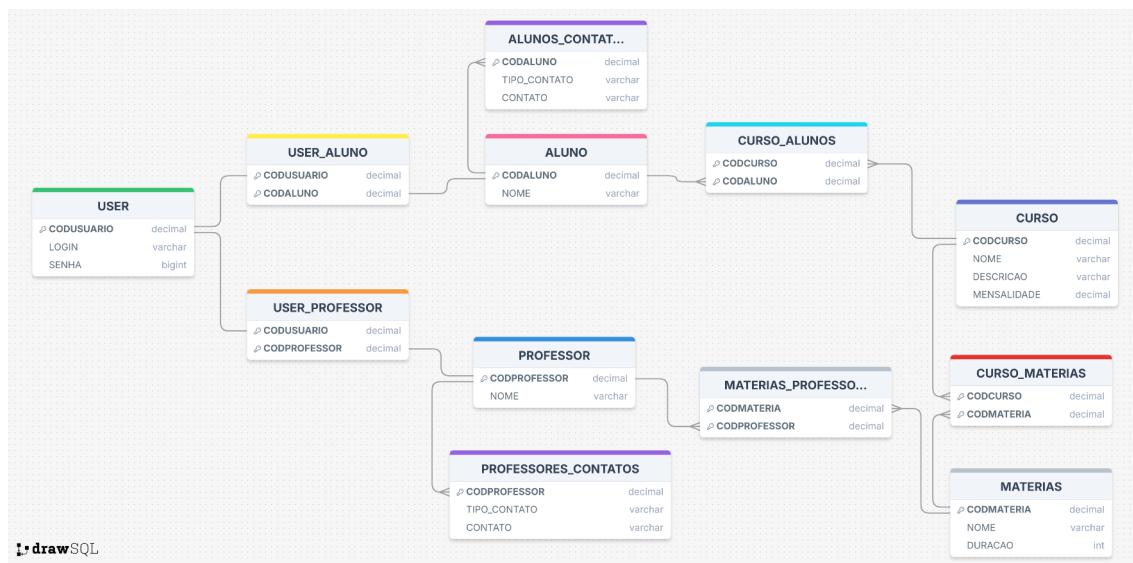


	atualização, inserção e exclusão relacionadas a dependências entre atributos não-chave.			
--	---	--	--	--

### 3.4. Diagrama Entidade-Relacionamento (DER) - Revisado

- Revisar e desenvolver o Modelo Lógico aplicando as formas normais 1FN, 2FN e 3FN.
- 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





Deixo abaixo o script de criação das tabelas, para auxiliar no entendimento, já que não achei um programa onde conseguisse criar um DML satisfatório

```

CREATE TABLE ALUNO (
    CODALUNO INTEGER PRIMARY KEY,
    NOME VARCHAR(100) NOT NULL
);

CREATE TABLE ALUNOS_CONTATOS (
    CODALUNO INTEGER NOT NULL,
    TIPO_CONTATO VARCHAR(2) NOT NULL,
    CONTATO VARCHAR(100) NOT NULL,
    PRIMARY KEY (CODALUNO, TIPO_CONTATO),
    FOREIGN KEY (CODALUNO) REFERENCES ALUNO(CODALUNO)
);

CREATE TABLE PROFESSOR (
    CODPROFESSOR INTEGER PRIMARY KEY,
    NOME VARCHAR(100) NOT NULL
);

CREATE TABLE PROFISSORES_CONTATOS (
    CODPROFESSOR INTEGER NOT NULL,
    TIPO_CONTATO VARCHAR(2) NOT NULL,
    CONTATO VARCHAR(100) NOT NULL,
    PRIMARY KEY (CODPROFESSOR, TIPO_CONTATO),
    FOREIGN KEY (CODPROFESSOR) REFERENCES PROFESSOR(CODPROFESSOR)
);

```



```
TIPO_CONTATO VARCHAR(2) NOT NULL,  
CONTATO VARCHAR(100) NOT NULL,  
PRIMARY KEY (CODPROFESSOR, TIPO_CONTATO),  
FOREIGN KEY (CODPROFESSOR) REFERENCES PROFESSOR(CODPROFESSOR)  
);  
  
CREATE TABLE CURSO (  
    CODCURSO INTEGER PRIMARY KEY,  
    NOME VARCHAR(100) NOT NULL,  
    DESCRICAO VARCHAR(999) NOT NULL,  
    MENSALIDADE DECIMAL(12,8) NOT NULL  
);  
  
CREATE TABLE MATERIAS (  
    CODMATERIA INTEGER PRIMARY KEY,  
    NOME VARCHAR(100) NOT NULL,  
    DURACAO INTEGER NOT NULL  
);  
  
CREATE TABLE CURSO_ALUNOS (  
    CODCURSO INTEGER NOT NULL,  
    CODALUNO INTEGER NOT NULL,  
    PRIMARY KEY (CODCURSO, CODALUNO),  
    FOREIGN KEY (CODCURSO) REFERENCES CURSO(CODCURSO),  
    FOREIGN KEY (CODALUNO) REFERENCES ALUNO(CODALUNO)  
);  
  
CREATE TABLE CURSO_MATERIAS (  
    CODCURSO INTEGER NOT NULL,  
    CODMATERIA INTEGER NOT NULL,  
    PRIMARY KEY (CODCURSO, CODMATERIA),  
    FOREIGN KEY (CODCURSO) REFERENCES CURSO(CODCURSO),  
    FOREIGN KEY (CODMATERIA) REFERENCES MATERIAS(CODMATERIA)  
);  
  
CREATE TABLE MATERIAS_PROFESSORES (  
    CODMATERIA INTEGER NOT NULL,  
    CODPROFESSOR INTEGER NOT NULL,  
    PRIMARY KEY (CODMATERIA, CODPROFESSOR),  
    FOREIGN KEY (CODMATERIA) REFERENCES MATERIAS(CODMATERIA),  
    FOREIGN KEY (CODPROFESSOR) REFERENCES PROFESSOR(CODPROFESSOR)  
);
```



```
CREATE TABLE USER (
    CODUSUARIO INTEGER PRIMARY KEY,
    LOGIN VARCHAR(50) NOT NULL,
    SENHA VARCHAR(50) NOT NULL
);

CREATE TABLE USER_ALUNO (
    CODUSUARIO INTEGER NOT NULL,
    CODALUNO INTEGER NOT NULL,
    PRIMARY KEY (CODUSUARIO, CODALUNO),
    FOREIGN KEY (CODUSUARIO) REFERENCES USER(CODUSUARIO),
    FOREIGN KEY (CODALUNO) REFERENCES ALUNO(CODALUNO)
);

CREATE TABLE USER_PROFESSOR (
    CODUSUARIO INTEGER NOT NULL,
    CODPROFESSOR INTEGER NOT NULL,
    PRIMARY KEY (CODUSUARIO, CODPROFESSOR),
    FOREIGN KEY (CODUSUARIO) REFERENCES USER(CODUSUARIO),
    FOREIGN KEY (CODPROFESSOR) REFERENCES PROFESSOR(CODPROFESSOR)
);
```