
MODELAGEM DE DADOS

Apostila do Aluno

Ensino Médio Técnico em Informática

MySQL / MariaDB

SUMÁRIO

Unidade 1 Introdução à Modelagem de Dados

Unidade 2 Modelo Entidade-Relacionamento (MER)

Unidade 3 Modelo Relacional (MR)

Unidade 4 Generalização e Especialização

Unidade 5 Entidade Associativa

Glossário

Como usar esta apostila

- Esta apostila acompanha as aulas. Leia o conteúdo antes de cada aula para aproveitar melhor o tempo em sala.
- Os exemplos estão prontos para serem executados no MySQL. Use o MySQL Workbench ou o terminal.
- Faça os exercícios antes de verificar a solução com o professor — é a melhor forma de aprender modelagem.
- Consulte o Glossário ao final para rever definições rapidamente.

UNIDADE 1

Introdução à Modelagem de Dados

1.1 Dado e Informação

No dia a dia usamos constantemente os termos dado e informação como se fossem sinônimos, mas em banco de dados eles têm significados distintos.

Definição: Dado

Elemento bruto, sem significado por si só. Exemplos: 15, 'Maria', '2024-03-10'.

Definição: Informação

Dado processado e contextualizado que possui significado. Exemplo: 'Maria tem 15 anos e nasceu em 10/03/2024'.

Dica

Um banco de dados armazena dados e os transforma em informação por meio de consultas e relatórios.

1.2 O que é um Banco de Dados?

Um Banco de Dados (BD) é uma coleção organizada de dados relacionados entre si, representando aspectos do mundo real (mini-mundo), com um propósito específico e um grupo de usuários interessado.

Comparação: Arquivos Avulsos × Banco de Dados

Aspecto	Arquivos Avulsos	Banco de Dados
Redundância	Alta (dados duplicados)	Baixa (dados centralizados)
Inconsistência	Frequente	Controlada
Acesso	Limitado	Multiusuário
Segurança	Difícil	Controle de acesso integrado
Integridade	Manual	Regras automáticas

1.3 Sistema Gerenciador de Banco de Dados (SGBD)

O SGBD é o software responsável por criar, manter e controlar o acesso a um banco de dados. Ele age como intermediário entre o usuário/aplicação e o banco de dados físico.

Principais SGBDs:

- MySQL / MariaDB — open source, amplamente usado em aplicações web.
- PostgreSQL — open source, muito poderoso, com suporte a funcionalidades avançadas.
- Oracle Database — robusto, muito usado em grandes empresas.
- Microsoft SQL Server — integrado ao ecossistema Microsoft.

- SQLite — leve, embutido em aplicações mobile e desktop.

1.4 As Fases do Projeto de Banco de Dados

Antes de criar qualquer tabela no computador, é necessário planejar o banco de dados em etapas. Esse processo chama-se Projeto de Banco de Dados e possui três fases principais:

Fase 1 — Modelo Conceitual (Alto Nível)

- Representa o mini-mundo de forma independente de qualquer SGBD.
- Linguagem próxima do usuário (não técnica).
- Principal ferramenta: Diagrama Entidade-Relacionamento (DER).
- Pergunta-chave: O QUE o banco precisa representar?

Fase 2 — Modelo Lógico

- Adapta o modelo conceitual ao tipo de banco (relacional, orientado a objetos, etc.).
- Para bancos relacionais: Diagrama de Tabelas Relacionais (DTR).
- Pergunta-chave: COMO os dados serão estruturados logicamente?

Fase 3 — Modelo Físico

- Específico para um SGBD (MySQL, Oracle, etc.).
- Contém tipos de dados, índices, particionamento e otimizações.
- Resultado: scripts SQL (CREATE TABLE, etc.).
- Pergunta-chave: COMO o banco será implementado no SGBD?

✦ Atenção

A ordem SEMPRE deve ser: Conceitual → Lógico → Físico. Pular etapas aumenta o risco de erros que serão difíceis de corrigir depois.

📎 EXERCÍCIOS

1. Explique com suas palavras a diferença entre dado e informação. Dê dois exemplos de cada.
2. Cite três vantagens de usar um SGBD em vez de arquivos de texto.
3. Um sistema de controle de estoque de uma loja será desenvolvido. Classifique cada item abaixo no modelo correto (Conceitual, Lógico ou Físico): (a) Entidade PRODUTO com atributo Nome; (b) Tabela PRODUTO(id INT PRIMARY KEY, nome VARCHAR(100)); (c) Relação PRODUTO(id, nome, preco, estoque) com PK id.
4. Pesquise: qual é a diferença entre MySQL e MariaDB?

UNIDADE 2

Modelo Entidade-Relacionamento (MER)

2.1 Características e Finalidades

O Modelo Entidade-Relacionamento (MER) foi proposto por Peter Chen em 1976 e é o mais utilizado para a fase de modelagem conceitual. Seu objetivo é representar graficamente a estrutura de dados de um mini-mundo, de forma compreensível tanto para técnicos quanto para usuários leigos.

O resultado visual do MER chama-se Diagrama Entidade-Relacionamento (DER).

Elementos Básicos do DER

Elemento	Símbolo	Descrição
Entidade	Retângulo	Objeto do mundo real com existência independente
Relacionamento	Losango	Associação entre duas ou mais entidades
Atributo	Elipse	Propriedade ou característica de uma entidade
Linha	Linha reta	Conecta entidade ao relacionamento ou atributo

2.2 Tipos de Entidade

Definição: Entidade Forte

Possui existência independente — não depende de nenhuma outra entidade para existir. Representada por retângulo simples. Ex: ALUNO, PRODUTO, CLIENTE.

Definição: Entidade Fraca

Depende de uma entidade forte para existir. Não possui atributo identificador próprio completo. Representada por retângulo duplo. Ex: DEPENDENTE (de FUNCIONARIO), ITEM_PEDIDO (de PEDIDO).

Dica

Para identificar se é fraca: pergunte 'essa entidade pode existir sem a outra?' Se não → entidade fraca.

2.3 Tipos de Atributos

Atributo Simples vs. Composto

Atributo Simples: não pode ser subdividido. Ex: CPF, Salário.

Atributo Composto: pode ser dividido em partes menores com significado próprio. Ex: Endereço → Rua + Número + Bairro + Cidade + CEP.

Atributo Monovalorado vs. Multivalorado

Monovalorado: assume apenas um valor por ocorrência. Ex: Data_Nascimento.

Multivalorado: pode ter mais de um valor. Representado por elipse dupla. Ex: Telefone (pode ter celular, residencial, comercial).

Atributo Derivado

Seu valor pode ser calculado a partir de outro atributo. Representado por elipse tracejada. Ex: Idade (derivada de Data_Nascimento).

Atributo Identificador (Chave)

Identifica unicamente cada ocorrência da entidade. Representado com o nome sublinhado. Ex: CPF, Matricula, CodProduto.

Atributo Identificador Parcial (entidade fraca): sublinhado tracejado. Ex: NumItem em ITEM_PEDIDO.

Resumo dos Tipos de Atributos

Tipo	Representação no DER	Exemplo
Simples	Elipse normal	CPF, Salário
Composto	Elipse com sub-elipses	Endereço
Monovalorado	Elipse normal (padrão)	Data_Nascimento
Multivalorado	Elipse dupla	Telefone
Derivado	Elipse tracejada	Idade
Identificador (forte)	Elipse com nome sublinhado	Matricula
Identificador parcial (fraca)	Elipse com nome sublinhado tracejado	NumItem

2.4 Tipos de Relacionamento

Grau do Relacionamento

- **Binário:** envolve duas entidades (mais comum). Ex: ALUNO cursa DISCIPLINA.
- **Ternário:** envolve três entidades. Ex: MEDICO prescreve MEDICAMENTO para PACIENTE.
- **Autorelacionamento (Unário):** uma entidade se relaciona com ela mesma. Ex: FUNCIONARIO supervisiona FUNCIONARIO.

2.5 Cardinalidade Máxima

A cardinalidade máxima define quantas ocorrências de uma entidade podem se associar a cada ocorrência da outra entidade.

Tipos de Cardinalidade Máxima

Cardinalidade	Significado	Exemplo
1:1 (um para um)	Cada ocorrência de A associa-se a no máximo 1 de B e vice-versa	FUNCIONARIO gerencia DEPARTAMENTO
1:N (um para muitos)	Cada ocorrência de A associa-se a várias de B; cada B a no máximo 1 de A	DEPARTAMENTO possui FUNCIONARIOS
M:N (muitos para muitos)	Cada ocorrência de A associa-se a várias de B e vice-versa	ALUNO cursa DISCIPLINA

Dica

Para determinar a cardinalidade, faça as perguntas nos dois sentidos: 'Um(a) X pode se relacionar com quantos(as) Y? E um(a) Y com quantos(as) X?'

2.6 Cardinalidade Mínima e Participação

A cardinalidade mínima define se a participação de uma entidade em um relacionamento é obrigatória ou opcional.

Definição: Participação Total (obrigatória)

Toda ocorrência da entidade DEVE participar do relacionamento. Cardinalidade mínima = 1. Representada por linha dupla no DER.

Definição: Participação Parcial (opcional)

Uma ocorrência da entidade PODE não participar do relacionamento. Cardinalidade mínima = 0. Representada por linha simples no DER.

A notação completa (min, max) é escrita ao lado de cada entidade na linha do relacionamento:

Exemplos de Notação (min,max)

Notação	Significado
(1,1)	Participa de exatamente 1 — obrigatório e único
(0,1)	Participa de 0 ou 1 — opcional e único
(1,N)	Participa de 1 ou mais — obrigatório e múltiplo
(0,N)	Participa de 0 ou mais — opcional e múltiplo

Exemplo Completo — Mini-mundo: Escola

ALUNO(0,N) —cursa— (1,N)DISCIPLINA

- Um aluno pode cursar de 0 a N disciplinas (pode estar sem matrícula).
- Uma disciplina deve ter pelo menos 1 aluno cursando.

TURMA(1,1) —tem— (1,N)ALUNO

- Um aluno pertence a exatamente 1 turma.
- Uma turma tem 1 ou mais alunos.

EXERCÍCIOS

- Classifique os atributos de FUNCIONARIO: (a) CPF; (b) Nome_Completo (Primeiro + Último); (c) Idade; (d) Telefone (pode ter vários); (e) Código.
- Determine a cardinalidade máxima dos relacionamentos: (a) MÉDICO atende PACIENTE; (b) PESSOA possui CPF; (c) ATOR atua em FILME.
- Dado o mini-mundo: 'Uma escola possui turmas. Cada turma tem exatamente um professor responsável. Um professor pode ser responsável por no máximo uma turma. Todo aluno deve estar matriculado em uma turma.' — Construa o DER com as cardinalidades.
- Diferencie entidade forte de entidade fraca. Cite dois exemplos de cada.
- Construa o DER completo para o mini-mundo: 'Uma clínica possui médicos (CRM, Nome, Especialidade) e pacientes (CPF, Nome, Data_Nascimento). Um paciente pode consultar vários médicos e um médico atende vários pacientes. Cada consulta tem data e um diagnóstico.'

UNIDADE 3

Modelo Relacional (MR)

3.1 Fundamentos Teóricos

O Modelo Relacional foi proposto por Edgar F. Codd em 1970, no artigo 'A Relational Model of Data for Large Shared Data Banks'. É o modelo mais utilizado nos SGBDs comerciais e o fundamento teórico do SQL.

Terminologia do Modelo Relacional

Termo Formal	Sinônimo Comum	Descrição
Relação	Tabela	Conjunto de tuplas com o mesmo esquema
Tupla	Linha / Registro	Uma ocorrência de dados na relação
Atributo	Coluna / Campo	Propriedade da relação
Domínio	Tipo de dado	Conjunto de valores permitidos para um atributo
Grau	Número de colunas	Quantidade de atributos da relação
Cardinalidade	Número de linhas	Quantidade de tuplas da relação

Propriedades das Relações

- Não existem tuplas duplicadas.
- A ordem das tuplas não tem significado.
- A ordem dos atributos não tem significado.
- Cada valor de atributo é atômico (indivisível — 1ª Forma Normal).

Definição: Esquema de Relação

Notação formal: NOME_RELACÃO(atrib1, atrib2, ...). Ex: FUNCIONARIO(Matricula, Nome, Salario, CodDepto).

3.2 Domínios e Tipos de Dados no MySQL

O domínio de um atributo define quais valores são válidos. No MySQL, os domínios são implementados por tipos de dados:

Principais Tipos de Dados no MySQL

Tipo	Uso	Exemplo
INT	Números inteiros	Idade, Quantidade
DECIMAL(p,e)	Números decimais precisos	Salário: DECIMAL(10,2)

VARCHAR(n)	Texto de tamanho variável (até n)	Nome VARCHAR(100)
CHAR(n)	Texto de tamanho fixo	CPF CHAR(11)
DATE	Data (YYYY-MM-DD)	Data_Nascimento
DATETIME	Data e hora	Data_Hora_Consulta
BOOLEAN / TINYINT(1)	Verdadeiro/Falso	Ativo
TEXT	Texto longo	Descrição, Observação

Valores Nulos (NULL)

Definição: NULL

Representa ausência de valor — seja porque o valor é desconhecido, inaplicável ou inexistente. NULL ≠ zero ≠ string vazia.

Atenção

Cuidado: NULL em cálculos resulta em NULL (ex: 10 + NULL = NULL). Em comparações, NULL = NULL é falso — use IS NULL.

Exemplo SQL (MySQL)

```
-- Criando tabela com restrição NOT NULL:
CREATE TABLE FUNCIONARIO (
  Matricula    INT           PRIMARY KEY,
  Nome         VARCHAR(100)  NOT NULL,
  Salario      DECIMAL(10,2) NOT NULL,
  Supervisor   INT           NULL  -- pode ser nulo (sem supervisor)
);
```

3.3 Tipos de Chaves

Definição: Superchave

Conjunto de um ou mais atributos que identifica unicamente uma tupla.

Definição: Chave Candidata

Superchave mínima — sem atributos desnecessários.

Definição: Chave Primária (PK)

Chave candidata escolhida como identificador principal. Não pode ser NULL; deve ser única.

Definição: Chave Alternativa

Chave candidata que não foi escolhida como PK.

🔑 Definição: Chave Composta

Chave formada por dois ou mais atributos juntos.

🔑 Definição: Chave Estrangeira (FK)

Atributo(s) em uma relação que referencia(m) a PK de outra relação. Implementa o relacionamento entre tabelas.

Exemplo de Chaves

FUNCIONARIO(Matricula, CPF, Nome, CodDepto)

- Superchaves: {Matricula}, {CPF}, {Matricula, CPF}, {Matricula, Nome}, ...
- Chaves candidatas: {Matricula}, {CPF} ← ambas são mínimas
- Chave primária (escolhida): Matricula
- Chave alternativa: CPF
- CodDepto: FK que referencia DEPARTAMENTO(CodDepto)

3.4 Regras de Integridade**🔑 Definição: Integridade de Entidade**

A chave primária de uma relação NUNCA pode ser NULL. Todo valor de PK deve ser único.

🔑 Definição: Integridade Referencial

O valor de uma FK deve ser igual a algum valor de PK existente na relação referenciada, OU ser NULL (se permitido).

Ações referenciais — o que acontece quando a tupla referenciada é excluída ou atualizada:

Ações Referenciais no MySQL

Ação	ON DELETE	ON UPDATE
RESTRICT	Impede a exclusão se houver FK referenciando	Impede a atualização
CASCADE	Exclui as tuplas dependentes automaticamente	Atualiza as FKs automaticamente
SET NULL	Define as FKs como NULL	Define as FKs como NULL
NO ACTION	Igual a RESTRICT (verificado ao final da transação)	Igual a RESTRICT

Exemplo SQL (MySQL)

```
CREATE TABLE DEPTO (
  CodDepto  INT          PRIMARY KEY,
  Nome      VARCHAR(80)  NOT NULL
);

CREATE TABLE FUNCIONARIO (
  Matricula INT          PRIMARY KEY,
  Nome      VARCHAR(100) NOT NULL,
  CodDepto  INT,
  FOREIGN KEY (CodDepto) REFERENCES DEPTO(CodDepto)
    ON DELETE SET NULL
    ON UPDATE CASCADE
);
```

3.5 Migração do DER para o DTR

A migração converte o modelo conceitual (DER) em modelo lógico relacional (DTR = Diagrama de Tabelas Relacionais).

Regra 1 — Entidade Forte

Cada entidade forte → uma tabela. Atributos simples → colunas. Atributo identificador → PK. Atributo composto → colunas dos sub-atributos. Atributo multivalorado → tabela separada.

Regra 2 — Entidade Fraca

Entidade fraca → tabela com PK composta: (PK da entidade forte + identificador parcial). Adicionar FK para a entidade forte.

Regra 3 — Relacionamento 1:1

Migrar a PK de uma entidade como FK na outra. Preferência: migrar para a entidade de participação total (obrigatória). Atributos do relacionamento acompanham a FK.

Regra 4 — Relacionamento 1:N

Migrar a PK da entidade do lado '1' como FK na tabela do lado 'N'. Atributos do relacionamento também vão para o lado 'N'.

Regra 5 — Relacionamento M:N

Criar uma nova tabela (tabela associativa). Sua PK é composta pelas PKs das duas entidades. Atributos do relacionamento tornam-se colunas dessa nova tabela.

Regra 6 — Autorelacionamento

1:N → adicionar FK autorreferencial na mesma tabela. M:N → criar tabela associativa com duas FKs para a mesma tabela (mas com nomes diferentes).

Exemplo Completo de Migração

DER: DEPARTAMENTO(CodDepto, Nome) —1:N— FUNCIONARIO(Matricula, Nome, Salario)

DTR (lógico):

DEPARTAMENTO(CodDepto*, Nome)

FUNCIONARIO(Matricula*, Nome, Salario, CodDepto**)

Onde: * = PK ** = FK

DER: ALUNO —M:N cursa— DISCIPLINA [atributo: Nota]

DTR:

ALUNO(RA*, Nome)

DISCIPLINA(CodDisc*, Descricao)

CURSA(RA**, CodDisc**, Nota) ← PK composta: (RA, CodDisc)

EXERCÍCIOS

10. Explique o que é integridade referencial e por que ela é importante.
11. Dada a relação PEDIDO(NroPedido, DataPedido, CodCliente, CodVendedor), identifique: (a) PK; (b) FKs prováveis; (c) atributos que podem ser NOT NULL.
12. Migre o seguinte DER para DTR: AUTOR(CodAutor, Nome) —M:N escreve— LIVRO(ISBN, Titulo, AnoPublicacao) [atributo do relacionamento: OrdemAutoria]
13. Explique a diferença entre ON DELETE CASCADE e ON DELETE RESTRICT. Em qual situação você usaria cada um?
14. Crie o script SQL (MySQL) para implementar o DTR do exercício anterior (Autor e Livro), incluindo as restrições de integridade.

UNIDADE 4

Generalização e Especialização

4.1 Conceitos Fundamentais

A Generalização e Especialização (G/E) permite modelar hierarquias de tipos no MER, representando situações onde um conjunto de entidades pode ser categorizado em subgrupos com características próprias.

Definição: Generalização

Processo de abstração: agrupamos entidades com características comuns em uma superentidade (supertipo). Processo bottom-up (de baixo para cima).

Definição: Especialização

Processo inverso: a partir de uma entidade genérica, definimos subentidades com atributos ou relacionamentos específicos. Processo top-down (de cima para baixo).

Definição: Herança

Subentidade herda automaticamente TODOS os atributos e relacionamentos da superentidade, além de poder ter os seus próprios.

Exemplo Clássico — PESSOA

PESSOA (superentidade): CPF, Nome, Data_Nascimento, Endereço

└ ALUNO (subentidade): herda tudo de PESSOA + Matricula, Curso

└ PROFESSOR (subentidade): herda tudo de PESSOA + SIAPE, Titulacao

Notação no DER: triângulo com 'ISA' (is-a = é um tipo de)

Leitura: 'ALUNO é um tipo de PESSOA'

4.2 Tipos de Generalização/Especialização

Quanto à Totalidade

Definição: Especialização Total (t)

TODA ocorrência da superentidade pertence a pelo menos uma subentidade. Não existe uma ocorrência genérica 'solta'. Ex: todo ANIMAL é ou MAMIFERO ou REPTIL ou AVE (sem 'animal genérico').

Definição: Especialização Parcial (p)

Uma ocorrência da superentidade PODE não pertencer a nenhuma subentidade. Ex: PESSOA pode ser apenas PESSOA, sem ser ALUNO nem PROFESSOR.

Quanto à Sobreposição

🔑 Definição: Especialização Exclusiva (x ou d — disjoint)

Uma ocorrência pode pertencer a NO MÁXIMO uma subentidade. As subentidades são disjuntas. Ex: um VEÍCULO é CARRO ou MOTO, não ambos.

🔑 Definição: Especialização Compartilhada (s ou o — overlapping)

Uma ocorrência PODE pertencer a mais de uma subentidade simultaneamente. Ex: uma PESSOA pode ser ALUNO e FUNCIONÁRIO ao mesmo tempo.

Combinações de G/E

Tipo	Notação	Significado	Exemplo
Total Exclusiva	{t, x}	Toda ocorrência pertence a exatamente 1 subentidade	ANIMAL → {MAMIFERO, REPTIL, AVE}
Total Compartilhada	{t, s}	Toda ocorrência pertence a pelo menos 1 subentidade, podendo ser mais	PRODUTO → {PERECIVEL, FRAGIL} (pode ser ambos)
Parcial Exclusiva	{p, x}	Pode não pertencer a nenhuma; se pertencer, a no máximo 1	PESSOA → {ALUNO, PROFESSOR}
Parcial Compartilhada	{p, s}	Pode não pertencer a nenhuma; se pertencer, pode ser a várias	PESSOA → {CLIENTE, FORNECEDOR}

4.3 Mapeamento de G/E para o Modelo Relacional

Existem três estratégias para converter hierarquias de G/E em tabelas relacionais:

Estratégia 1 — Tabela Única (Table per Hierarchy)

Uma só tabela para toda a hierarquia.

Inclui todos os atributos da super + de todas as sub.

Um atributo discriminador indica o subtipo.

Atributos de subentidades ficam NULL para as demais.



Vantagem: consultas simples, sem JOIN.



Desvantagem: muitos valores NULL; pode violar integridade.

Estratégia 2 — Tabela por Tipo Concreto (Table per Concrete Type)

Uma tabela para cada subentidade, repetindo os atributos da super.
Não há tabela para a superentidade.

✅ Vantagem: não há NULLs.

❌ Desvantagem: redundância de atributos; difícil de manter.

Estratégia 3 — Tabela por Tipo (Table per Type) ← RECOMENDADA

Tabela para a superentidade + uma tabela para cada subentidade.

As tabelas das subentidades referenciam a PK da superentidade.

Não há repetição de atributos.

✅ Vantagem: sem redundância; fiel ao modelo conceitual.

❌ Desvantagem: JOINS necessários para recuperar dados completos.

Exemplo SQL (MySQL)

-- Estratégia 3: Tabela por Tipo

```
CREATE TABLE PESSOA (
    CPF          CHAR(11)      PRIMARY KEY,
    Nome         VARCHAR(100)  NOT NULL,
    DtNasc       DATE
);

CREATE TABLE ALUNO (
    CPF          CHAR(11)      PRIMARY KEY,
    Matricula    VARCHAR(20)   UNIQUE NOT NULL,
    Curso        VARCHAR(80),
    FOREIGN KEY (CPF) REFERENCES PESSOA(CPF) ON DELETE CASCADE
);

CREATE TABLE PROFESSOR (
    CPF          CHAR(11)      PRIMARY KEY,
    SIAPE        VARCHAR(20)   UNIQUE NOT NULL,
    Titulacao    VARCHAR(50),
    FOREIGN KEY (CPF) REFERENCES PESSOA(CPF) ON DELETE CASCADE
);
```

EXERCÍCIOS

15. Classifique o tipo de G/E em cada cenário: (a) VEICULO se especializa em CARRO e MOTO, e todo veículo é de um tipo; (b) PESSOA pode ser CLIENT, FUNCIONARIO, ou ambos, e nem toda pessoa precisa ser uma das duas; (c) CONTA_BANCARIA se divide em CORRENTE e POUPANÇA, sendo que toda conta é de um único tipo.
16. Dado o mini-mundo: 'Um hospital tem funcionários que podem ser médicos ou enfermeiros ou recepcionistas. Toda funcionário é de exatamente um desses tipos.' — Modele a hierarquia de G/E no DER com tipo correto.

17. Para a hierarquia do exercício anterior, implemente a Estratégia 3 (Tabela por Tipo) em SQL (MySQL). Inclua os atributos: FUNCIONARIO(CPF, Nome, Salário); MEDICO(CRM, Especialidade); ENFERMEIRO(COREN, Turno); RECEPCIONISTA(Setor).
18. Qual estratégia de mapeamento você usaria para: (a) uma hierarquia {t,x} com poucas subentidades e muitos atributos em comum? (b) uma hierarquia {p,s} com subentidades muito diferentes? Justifique.

UNIDADE 5**Entidade Associativa****5.1 Motivação e Conceito**

Em um DER, relacionamentos servem para conectar entidades. Mas o que acontece quando precisamos associar OUTRO relacionamento ou entidade a um relacionamento M:N existente?

Considere o exemplo: MÉDICO prescreve MEDICAMENTO para PACIENTE. Cada prescrição tem uma dosagem e uma duração. Queremos também registrar EXAMES realizados a partir de uma prescrição.

Nesse caso, o relacionamento 'prescreve' precisa se relacionar com EXAME — e para isso, ele deve se tornar uma Entidade Associativa.

🔑 Definição: Entidade Associativa

Um relacionamento (geralmente M:N) que é 'elevado' ao nível de entidade, podendo, portanto, participar de outros relacionamentos e ter atributos. Representada no DER por um retângulo ao redor do losango.

5.2 Quando Usar

Use Entidade Associativa quando:

- O relacionamento M:N precisa se relacionar com outra entidade.
- O relacionamento possui atributos E esses atributos têm relacionamentos próprios.
- Conceitualmente, o relacionamento em si representa um 'objeto' importante do negócio (ex: Matrícula, Pedido, Consulta, Contrato).

💡 Atenção

Atenção: se o relacionamento M:N tem apenas atributos simples (sem relacionamentos com outras entidades), NÃO é necessário transformá-lo em Entidade Associativa — basta manter como relacionamento com atributos.

5.3 Exemplos Detalhados**Exemplo 1 — MATRÍCULA****Contexto**

ALUNO e DISCIPLINA têm um relacionamento M:N 'cursa' com atributo Nota. Além disso, cada 'cursada' (matrícula) pode ter várias AVALIAÇÕES associadas.

Solução: elevar 'cursa' para a Entidade Associativa MATRICULA.

DER:

ALUNO —M:N cursa —DISCIPLINA

[cursa] → MATRICULA (Entidade Associativa)

MATRICULA —1:N possui— AVALIACAO(TipoAvaliacao, Nota, Data)

Exemplo 2 — ITEM DE PEDIDO

Contexto

PEDIDO e PRODUTO têm relacionamento M:N 'contém' com atributos Quantidade e Preço_Unitario.

Aqui, como não há outro relacionamento, poderia ser apenas um relacionamento com atributos.

Porém, se ITEM_PEDIDO precisar se relacionar com DEVOLUCAO, elevamos para Entidade Associativa.

DTR resultante (Estratégia 5 de migração):

PEDIDO(NroPedido*, DataPedido, CodCliente**)

PRODUTO(CodProduto*, Descricao, Preço)

ITEM_PEDIDO(NroPedido**, CodProduto**, Quantidade, Preço_Unitario)

DEVOLUCAO(CodDevolucao*, NroPedido**, CodProduto**, Motivo, DataDev)

5.4 Mapeamento para o Modelo Relacional

A Entidade Associativa é mapeada como uma tabela com:

- PK composta pelas PKs das entidades participantes do relacionamento original (ou uma PK surrogate — chave artificial).
- Atributos próprios do relacionamento como colunas.
- FKs para as entidades com as quais a Entidade Associativa se relaciona.



Exemplo SQL (MySQL)

```
-- Exemplo: MATRICULA como Entidade Associativa
CREATE TABLE ALUNO (
  RA      INT          PRIMARY KEY,
  Nome    VARCHAR(100) NOT NULL
);

CREATE TABLE DISCIPLINA (
  CodDisc  INT          PRIMARY KEY,
  Descricao VARCHAR(100) NOT NULL
);

CREATE TABLE MATRICULA (
  RA        INT NOT NULL,
  CodDisc   INT NOT NULL,
  Semestre  CHAR(6) NOT NULL,      -- ex: '2024-1'
  PRIMARY KEY (RA, CodDisc, Semestre),
  FOREIGN KEY (RA) REFERENCES ALUNO(RA),
  FOREIGN KEY (CodDisc) REFERENCES DISCIPLINA(CodDisc)
);

CREATE TABLE AVALIACAO (
```

```

CodAval    INT AUTO_INCREMENT PRIMARY KEY,
RA         INT NOT NULL,
CodDisc    INT NOT NULL,
Semestre   CHAR(6) NOT NULL,
Tipo       VARCHAR(50),
Nota       DECIMAL(4,2),
FOREIGN KEY (RA, CodDisc, Semestre)
  REFERENCES MATRICULA(RA, CodDisc, Semestre)
);

```

EXERCÍCIOS

19. Explique com suas palavras a diferença entre: (a) relacionamento M:N com atributos; (b) Entidade Associativa.
20. Dado o seguinte mini-mundo: 'Médicos prescrevem medicamentos para pacientes. Uma prescrição tem data, posologia e duração. Para cada prescrição, podem ser solicitados vários exames.' — Identifique a Entidade Associativa, modele o DER completo e converta para DTR.
21. Construa o DER e o DTR completo para o seguinte mini-mundo: 'Uma plataforma de cursos online tem alunos e cursos. Um aluno pode se matricular em vários cursos e cada curso tem vários alunos. A matrícula registra a data de início. Para cada matrícula, há uma ou mais avaliações de progresso.'
22. **Exercício Integrador:** Mini-mundo — Sistema de Locadora de Veículos Uma locadora tem clientes (CPF, Nome, CNH, Endereço), veículos (Placa, Marca, Modelo, Ano, Categoria: Básico/Intermediário/Luxo) e funcionários (Matricula, Nome, Cargo: Atendente/Gerente). Um cliente pode realizar várias locações. Cada locação envolve 1 veículo e 1 atendente. A locação registra data de retirada, data de devolução prevista e data de devolução real. Cada locação pode gerar uma ou mais multas (tipo, valor, data). Modele: (1) DER completo; (2) DTR; (3) Script SQL no MySQL.

GLOSSÁRIO

Termo	Definição
Atributo	Propriedade ou característica de uma entidade ou relacionamento.
Cardinalidade	Número máximo ou mínimo de ocorrências que participam de um relacionamento.
Chave Candidata	Superchave mínima — possível candidata a chave primária.
Chave Estrangeira (FK)	Atributo que referencia a PK de outra tabela.
Chave Primária (PK)	Atributo(s) que identificam unicamente cada tupla de uma relação.
DDL	Data Definition Language — comandos SQL para definir estruturas (CREATE, ALTER, DROP).
DER	Diagrama Entidade-Relacionamento — representação gráfica do modelo conceitual.
DTR	Diagrama de Tabelas Relacionais — representação do modelo lógico relacional.
Domínio	Conjunto de valores válidos para um atributo.
Entidade	Objeto ou conceito do mundo real com existência independente.
Entidade Associativa	Relacionamento elevado à categoria de entidade.
Entidade Fraca	Entidade que depende de outra para existir.
Especialização	Processo de definir subentidades a partir de uma entidade mais genérica.
FK	Veja Chave Estrangeira.
Generalização	Processo de agrupar entidades similares em uma superentidade.
Herança	Propriedade pela qual subentidades herdam atributos da superentidade.
Integridade Referencial	Regra que garante que FKs referenciem PKs existentes.
MER	Modelo Entidade-Relacionamento.
Modelo Conceitual	Representação de alto nível independente de SGBD.
Modelo Físico	Implementação específica para um SGBD.
Modelo Lógico	Adaptação do modelo conceitual ao tipo de banco.
Modelo Relacional	Modelo baseado em relações (tabelas) proposto por Codd.
NULL	Valor especial que representa ausência de informação.

PK	Veja Chave Primária.
Relação	Tabela no Modelo Relacional.
SGBD	Sistema Gerenciador de Banco de Dados.
Superchave	Conjunto de atributos que identifica unicamente uma tupla.
Tupla	Linha/registro de uma relação.