Cábula do modelo EA · Entidade-Associação

Bases · Associações · Atributos e exemplo · Ajuda

Holger Sebastião · www.sebastiao.org
Versão 1.0 · 2026-11-04

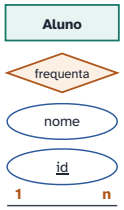
De que se trata?

O modelo EA é a planta de uma base de dados **antes** de haver tabelas: Que coisas existem, que propriedades têm, como se relacionam?

TERMO	SIGNIFICADO · EXEMPLO	PASSA A
Entidade	uma coisa concreta: a aluna Lea	linha
Tipo de entidade	todas as coisas do mesmo tipo: Aluno	tabela
Atributo	propriedade: nome	coluna
Chave	torna cada entidade única: id	chave primária
Associação	ligação: Aluno *frequenta* Turma	chave estrangeira
Cardinalidade	quantos com quantos: 1:n	página 2

Símbolos

notação de Chen



Tipo de entidade
retângulo, nome no singular

Associação
losango, geralmente um verbo

Atributo
elipse, uma propriedade

Atributo-chave
sublinhado: único

Linha com cardinalidade
1, n ou m nas pontas

Do texto ao modelo EA

1. Assinalar os **substantivos**: candidatos a tipos de entidade (aluno, turma, clube).

2. Propriedades como nome, data, preço: atributos.

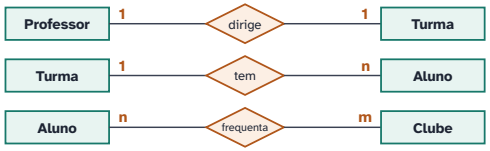
3. Verbos entre duas coisas: associações (frequenta, inscreve-se, requisita).

4. Definir as **chaves**. Os nomes repetem-se: na dúvida, um `id` artificial.

5. Perguntar a **cardinalidade** de cada associação nos dois sentidos.

> Só o que está no texto ou o que é preciso. Um modelo EA nunca está "perfeitamente certo", mas tem de servir a tarefa.

Cardinalidades



1:1	Um professor dirige **uma** turma, uma turma tem **um** diretor de turma.
1:n	Uma turma tem **muitos** alunos, um aluno está em **uma** turma.
n:m	Um aluno frequenta **muitos** clubes, um clube tem **muitos** alunos.

Determinar a cardinalidade

Começar sempre a pergunta com **uma** coisa e fazê-la nos **dois** sentidos:

Em quantas turmas está **um** aluno?	**1**
Quantos alunos tem **uma** turma?	**n**
Resultado: Turma 1 : n Aluno	**1:n**

> O número fica do lado **oposto**: o **n** junto a Aluno quer dizer "muitos alunos por turma".

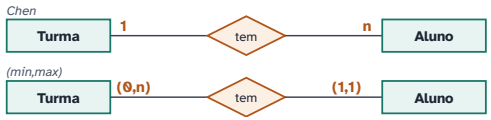
> ! A pergunta vale para o **caso normal** durante toda a vida da base de dados, não só para hoje: se um aluno pode mudar de turma e isso deve ficar guardado, passa a n:m.

Exemplos de leitura

Cliente – faz – Encomenda	1:n
Livro – é requisitado por – Leitor	n:m
Pessoa – tem – Cartão de Cidadão	1:1
Filme – passa em – Cinema	n:m
Departamento – emprega – Funcionário	1:n

Mais preciso: (min,max)

participação



(min,max) fica junto ao tipo de entidade e diz: quantas vezes participa **uma** coisa na associação?

`(0,1)`	no máximo uma vez, pode faltar
`(1,1)`	exatamente uma vez: cada aluno tem uma turma
`(0,n)`	quantas vezes quiser, até nenhuma: uma turma pode estar vazia
`(1,n)`	pelo menos uma vez

> ! Os lados estão **trocados** em relação a Chen: em Chen o 1 está junto a Turma, em (min,max) o (1,1) está junto a Aluno.

Notação pé de galinha

ferramentas

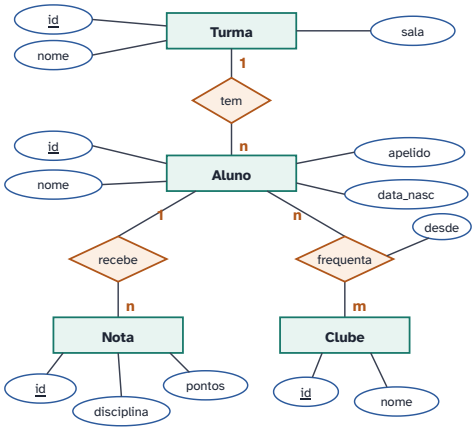


Sem losango: as pontas da linha mostram a cardinalidade. É assim que desenham o draw.io, o MySQL Workbench e muitas outras ferramentas.

Tipos de atributos

simples	`nome`: uma coluna
composto	`morada`: dividir em `rua`, `cod_postal`, `localidade`
multivalor	vários números de telefone: tipo de entidade próprio *Telefone* (1:n)
derivado	`idade`: não guardar, calcular a partir de `data_nasc`
da associação	`desde` em "frequenta": pertence ao aluno **e** ao clube

Exemplo: a nossa escola



Um aluno recebe muitas notas, uma nota pertence a um aluno. `desde` está ligado a "frequenta": diz desde quando **este** aluno está **neste** clube.

> No modelo EA **não** há chaves estrangeiras: `turma_id` só aparece ao passar para tabelas (página 2).

Erros típicos

- Um atributo como retângulo: *Nome* não é um tipo de entidade.
- Um valor como retângulo: *11IT1* é uma entidade do tipo Turma.
- Associação sem cardinalidade ou perguntada só num sentido.
- Chaves estrangeiras já desenhadas no modelo EA.
- Linha direta de retângulo a retângulo, sem losango.
- Valores derivados como *idade* ou *média* guardados.

Cábula do modelo EA · Entidade-Associação

Modelo relacional · Normalização · Ajuda

Holger Sebastião · www.sebastiao.org
Versão 1.0 · 2026-11-04

Modelo relacional — notação

Cada tabela numa linha: o nome e depois as colunas entre parênteses.

```
turma(id, nome, sala)
aluno(id, nome, apelido,
  data_nasc, turma_id)
```

sublinhado	chave primária
itálico	chave estrangeira, no BeaverSQL `turma_id#`, também `↑turma_id`

A partir daqui, nomes como em SQL: minúsculas, sem acentos nem espaços, no singular.

Regra 1: tipo de entidade → tabela

Cada tipo de entidade passa a tabela, cada atributo a coluna, a chave a chave primária.

`Aluno → aluno(id, nome, …)`

Dividir atributos compostos, deixar de fora os derivados, pôr os multivalor numa tabela própria.

Regra 2: 1:n → chave estrangeira

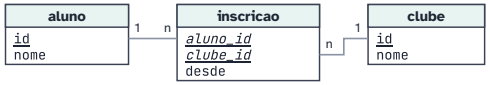


A chave primária do **lado 1** entra como chave estrangeira na tabela do **lado n**.

Não ao contrário: `turma` não recebe uma coluna `aluno_id`, senão só cabe um aluno em cada turma.

```
turma_id INT,
FOREIGN KEY (turma_id) REFERENCES turma(id)
```

Regra 3: n:m → tabela intermédia



A associação passa a uma **tabela própria**, com as chaves primárias dos dois lados como chaves estrangeiras. Juntas formam a chave primária. Os atributos da associação (`desde`) também entram.

`inscricao(aluno_id, clube_id, desde)`

Uma n:m dá **duas** 1:n. O nome é muitas vezes um substantivo do verbo: inscreve-se → inscricao.

Regra 4: 1:1

Chave estrangeira **num** dos lados, com `UNIQUE`. De preferência do lado que tem sempre par, assim nunca fica vazia:

`turma(id, nome, sala, professor_id)`

Se os dois lados andam sempre juntos, também se podem juntar numa só tabela.

A nossa escola em tabelas

```
turma(id, nome, sala)
aluno(id, nome, apelido,
  data_nasc, turma_id)
nota(id, disciplina, pontos, aluno_id)
clube(id, nome)
inscricao(aluno_id, clube_id, desde)
```

Quatro tipos de entidade, uma n:m: cinco tabelas. Os comandos estão na cábula de SQL, página 1.

Porquê normalizar?

Uma tabela com tudo lá dentro guarda muita coisa **em duplicado**:

ID	NOME	TURMA	SALA
1	Ali Yilmaz	11IT1	**B204**
2	Lea Schmitt	12IT1	B206
3	Tom Becker	11IT1	**B204**

Isto leva a três **anomalias**:

Inserção	A nova turma 13IT1 na sala B208 só entra quando tiver um aluno.
Atualização	Se a 11IT1 muda de sala, é preciso alterar `sala` em muitas linhas. Esquece-se uma e os dados contradizem-se.
Eliminação	Se a Lea sai da 12IT1, perde-se também a sala da turma.

Normalizar é dividir as tabelas de modo a que cada informação fique guardada só **uma vez**. Quem modela com cuidado como na página 1 chega quase sempre à 3FN.

1.ª forma normal — atómico

Cada campo tem **um** só valor: sem listas, sem dados compostos.

NOME	TELEFONE
Ali Yilmaz	211 123 456, 912 345 678

Solução: dividir `nome` em `nome` e `apelido`, pôr os telefones numa tabela própria:

`telefone(id, numero, aluno_id)`

2.ª forma normal — chave inteira

1FN, e cada coluna depende da chave primária **inteira**, não só de uma parte.

```
inscricao(aluno_id, clube_id, desde,
  nome_aluno, sala_clube)
```

`nome_aluno` só depende de `aluno_id`, `sala_clube` só de `clube_id`. Passar ambos para fora:

```
aluno(id, nome)
clube(id, sala)
inscricao(aluno_id, clube_id, desde)
```

A 2FN só pode falhar com uma chave **composta**. Com um único `id` está cumprida automaticamente.

3.ª forma normal — só a chave

2FN, e nenhuma coluna depende de outra coluna **que não é chave**.

`aluno(id, nome, turma, sala)`

`sala` depende de `turma`, não do aluno (id → turma → sala). Passar para fora:

```
turma(id, nome, sala)
aluno(id, nome, turma_id)
```

Para memorizar

Cada coluna depende **da chave** (1FN), **da chave inteira** (2FN) e de **nada mais que a chave** (3FN).

Lista de verificação

- ☐ Cada tabela tem chave primária.
- ☐ Cada 1:n tem chave estrangeira do lado n.
- ☐ Cada n:m tem tabela intermédia.
- ☐ Nenhum campo contém uma lista.
- ☐ Nenhuma informação está guardada duas vezes.
- ☐ Chave estrangeira e chave primária têm o mesmo tipo de dados.