Normalização

Aula 09 — 1FN, 2FN e 3FN

Aula 09 · 20/10/2026 · Indicadores 2, 4

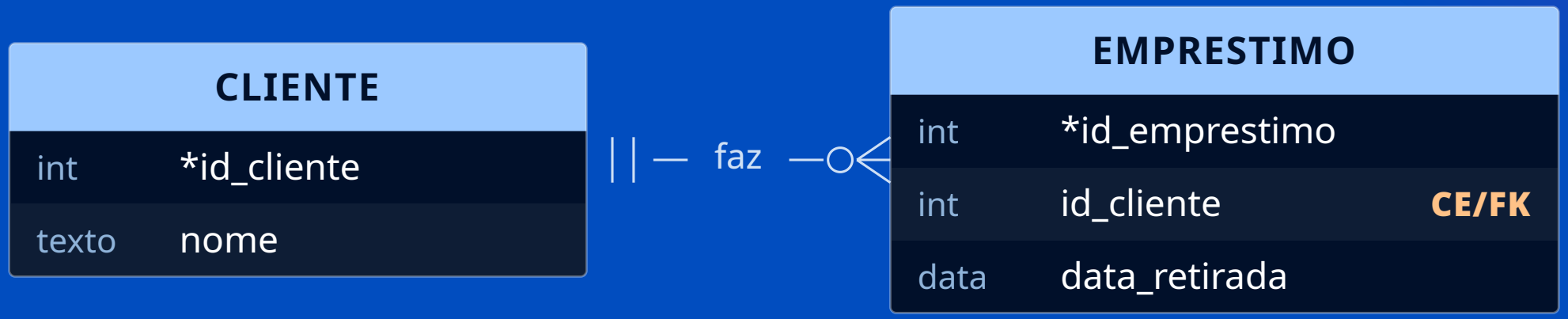
UniSenac Centro Universitário RS
Senac
Fecomércio
IFEP
Sesc
Sindicatos Empresariais

O mapa de hoje

- **Bloco 1** — **1FN**: uma célula, um valor
- **Bloco 2** — **2FN**: nada dependendo de um pedaço da chave
- **Bloco 3** — **3FN**: nada dependendo de carona · e quando **não** normalizar
- **Bloco 4** — mão na massa: normalizar o esquema do grupo até a 3FN

> No fim de hoje, o esquema de vocês está pronto para virar **SQL de verdade** na Aula 11. Esta é a última aula de papel.

Lembrete — a legenda do DER



Símbolo	É	Símbolo	É
caixa, cabeçalho escuro	entidade	`CE/FK`	chave estrangeira
linha na caixa	atributo	linha com verbo	relacionamento
`*` no nome	chave primária	marcadores nas pontas	cardinalidade

Retomando a Aula 08

- Quais são as **três anomalias**?
- O que quer dizer `A → B`?
- Qual a diferença entre repetir **referência** e repetir **informação**?

> Peguem a lista de dependências do grupo. Ela é o roteiro de trabalho de hoje.

Normalização, em uma frase

Normalizar é separar uma tabela em tabelas menores até que cada tabela fale de **um assunto só** — eliminando as três anomalias.

Três perguntas, em ordem. Cada uma resolve um problema:

TRÊS PERGUNTAS, EM ORDEM

1ª — a célula guarda mais de um valor?
se sim, separe → **Chega à 1FN**

2ª — depende de só um pedaço da chave?
se sim, separe → **Chega à 2FN**

3ª — depende de outro atributo comum?
se sim, separe → **Chega à 3FN**

Bloco 1

1ª Forma Normal — uma célula, um valor

A pergunta da 1FN

Alguma célula guarda mais de um valor?

id_cliente	nome	telefones
1	Ana Souza	9999-1111 / 9888-2222
2	Bruno Lima	9777-3333

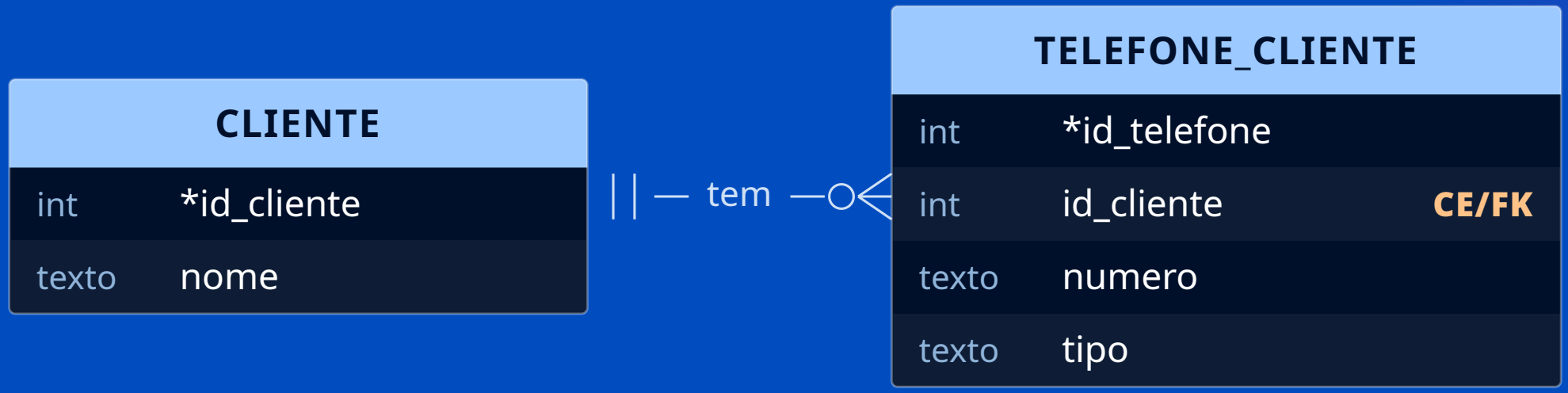
A célula `telefones` da Ana guarda **dois** valores. Isso fere a 1FN.

Por que isso quebra

- Como **buscar** por um telefone específico?
- Como saber **qual** é o celular e qual é o fixo?
- Como **contar** quantos telefones cada cliente tem?
- E se um cliente tiver **cinco**?

> Repetir colunas (`telefone1`, `telefone2`, `telefone3`) **também** fere a 1FN — e ainda impõe um limite arbitrário.

O conserto




```
cliente          (id_cliente*, nome)
telefone_cliente (id_telefone*, id_cliente CE/FK → cliente, numero, tipo)
```

Agora cada célula tem **um valor só**, e cabem quantos telefones for preciso.

⚠ Erro comum — separar o que não é lista

`endereco = "Rua das Flores, 120"` tem duas informações, mas **não** é multivalorado — é **composto** (Aula 04). Composto se resolve dividindo em colunas na mesma tabela; multivalorado se resolve criando outra tabela.

Caso	É	Conserto
`telefones = "A / B"`	multivalorado	tabela nova
`endereco = "Rua X, 120"`	composto	colunas `logradouro`, `numero`

Checagem do Bloco 1

Fere a 1FN? Se sim, qual o conserto?

- `generos = "Ação, Aventura, RPG"` na tabela `jogo`
- `nome_completo = "Ana Maria Souza"` na tabela `cliente`
- `ingredientes = "farinha; ovo; açúcar"` na tabela `bolo`
- `email` na tabela `cliente`



`forms.gle/M44Awj16aVoNPYqY9`

Intervalo

10 minutos

Bloco 2

2ª Forma Normal — nada dependendo de um pedaço da chave

A pergunta da 2FN

> Estando em 1FN: **a chave primária é composta e algum atributo depende de só um pedaço dela?**

Se a chave primária é **uma coluna só**, a 2FN já está garantida — não existe "pedaço" de uma coluna.

Um caso com chave composta

Tabela `item_emprestimo`, chave primária = `id_emprestimo + id_jogo`.
É a tabela do meio do N:N entre `emprestimo` e `jogo` — a mesma do DER de vocês:

*id_emprestimo	*id_jogo	titulo_jogo	data_retirada	preco_pago
1	10	God of War	02/10	12,00
1	11	FIFA 25	02/10	10,00
2	12	Zelda TotK	02/10	14,00

- `titulo_jogo` depende só de `id_jogo` → **parcial**
- `data_retirada` depende só de `id_emprestimo` → **parcial**
- `preco_pago` depende das **duas** → total, está certo

O conserto

```
emprestimo      (id_emprestimo*, data_retirada, id_cliente CE/FK → cliente)
jogo            (id_jogo*, titulo, genero)
item_emprestimo (id_emprestimo* CE/FK → emprestimo,
                 id_jogo* CE/FK → jogo,
                 preco_pago)
```

Cada atributo foi para a tabela do **seu** determinante.
Na tabela do meio sobra só o que depende da **chave inteira**.

Checagem do Bloco 2

Tabela `matricula`, chave = `id_aluno + id_disciplina`:

`(id_aluno*, id_disciplina*, nome_aluno, nome_disciplina, nota, semestre)`

- Quais atributos têm dependência **parcial**?
- Para onde vai cada um?
- O que sobra em `matricula`?

`forms.gle/AAWZrgdGqWaLyj3EA`

Intervalo

10 minutos

Bloco 3

3ª Forma Normal — e quando parar

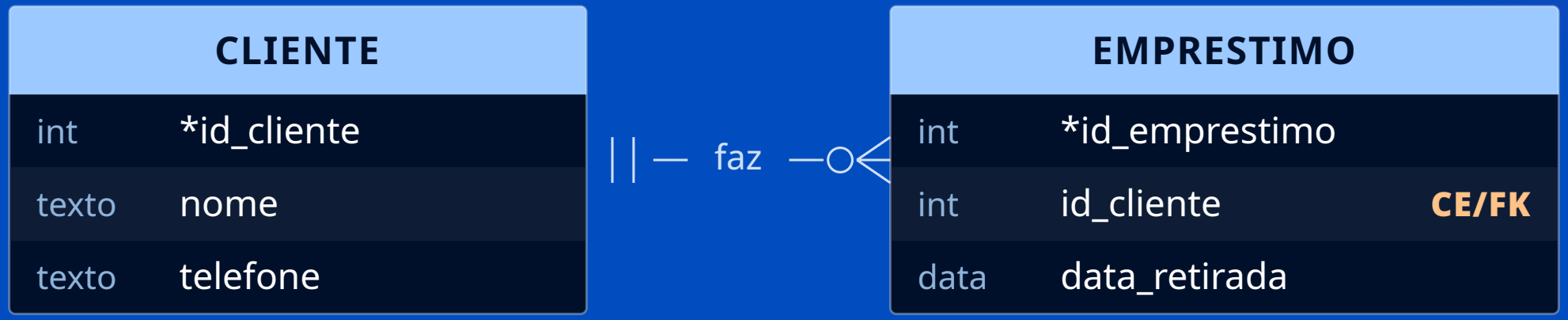
A pergunta da 3FN

Estando em 2FN: **algum atributo depende da chave "de carona", através de outro atributo?**

*id_emprestimo	data_retirada	id_cliente	nome_cliente	telefone
1	02/10	1	Ana Souza	9999-1111
2	02/10	2	Bruno Lima	9999-2222
3	09/10	1	Ana Souza	9999-1111

`id_emprestimo → id_cliente → nome_cliente`. O nome vem **de carona**.

O conserto




```
cliente    (id_cliente*, nome, telefone)
emprestimo (id_emprestimo*, data_retirada, id_cliente CE/FK → cliente)
```

Sobra em `jogo` só a **referência** — que é repetição legítima (Aula 08).

O resumo das três perguntas

1FN
alguma célula guarda mais de um valor?
separe em outra tabela

2FN
a chave é composta e algo depende de só um pedaço?
separe o pedaço

3FN
algo depende da chave de carona, via outro atributo?
separe o carona

Regra de bolso do mercado: *"cada atributo depende da chave, da chave inteira, e de nada além da chave"*. As três perguntas em uma frase.

Quando NÃO normalizar

> Normalizar tem um custo: mais tabelas significa mais `JOIN` nas consultas (Aula 14), e consulta com muitos JOINs é mais lenta e mais difícil de escrever.

Casos em que se aceita repetir de propósito:

- **Dado histórico** — o preço na nota fiscal é o preço **do dia da venda**, não o atual
- **Relatório muito pesado**, consultado o tempo todo, que compensa pré-calcular

⚠ Erro comum — normalizar até o absurdo

> Criar uma tabela `genero` com duas colunas para guardar seis gêneros de jogo raramente compensa. Cada tabela nova é mais um `JOIN`, mais uma tela de cadastro e mais uma coisa para manter.

Pergunte antes de separar:

- Esse dado **repete muito** e **muda** com alguma frequência?
- Ele tem **atributos próprios** além do nome?

Se as duas respostas forem "não", deixe como coluna.

Checagem do Bloco 3

Está em 3FN? Se não, o que separar?

- `emprestimo (id_emprestimo*, data_retirada, id_cliente, nome_cliente)`
- `endereco (id_endereco*, id_cliente, logradouro, cep, cidade)`
- `item_emprestimo (id_emprestimo*, id_jogo*, preco_pago)`

forms.gle/rk3L2yjurRqg9DZG7

Intervalo

10 minutos

MÃO NA MASSA

Normalizar o esquema do grupo até a 3FN

1. Peguem o esquema (Aula 07) e a lista de dependências (Aula 08)
2. **1FN:** alguma célula guardaria mais de um valor? Separem
3. **2FN:** alguma chave composta com dependência parcial? Separem
4. **3FN:** algum atributo de carona? Separem
5. Reescrevam o esquema relacional completo, já normalizado
6. **Atualizem o** `der.drawio.svg` para bater com o esquema novo
7. Anotem qualquer repetição que vocês decidiram **manter de propósito**, com o motivo

MÃO NA MASSA

O combinado desta entrega

GLOSSÁRIO

`grupoNN-aulaNN-artefato.<ext>` `grupo03-aula09-esquema-3fn.xlsx`

O que eu confiro: Alguma coluna ainda guarda **lista** · Alguma tabela tem mais de um **determinante** · Toda tabela nova ganhou **chave primária** própria · Toda separação deixou uma **chave estrangeira** ligando de volta · O **DER** foi atualizado junto com o esquema · As repetições mantidas de propósito estão **justificadas** por escrito

PARA CASA · OPCIONAL

Antes da próxima aula

Testar em casa: aplique as três perguntas naquela planilha que você analisou na Aula 08. Em que forma normal ela está?

Tarefa opcional: confira se o `der.drawio.svg` do grupo está **idêntico** ao esquema normalizado. Qualquer diferença agora vira erro de SQL na Aula 11.

Para ver acontecer: demonstrações da Aula 09 — procurar um telefone dentro de uma célula com dois e ver a busca errar, e aplicar as três formas normais vendo cada coluna sair — e oito perguntas com resposta na hora. No site da UC, na página da aula.

> **Logística importante:** na próxima aula vamos instalar o PostgreSQL e o pgAdmin. Quem for usar notebook próprio, traga carregado — e com espaço em disco.

Fim do bloco de Modelo Relacional

O modelo está pronto no papel.

Na próxima aula ele vira um banco de dados de verdade.

UniSenac Centro Universitário RS
Senac
Fecomércio
IFEP
Sesc
Sindicatos Empresariais