

De uma Palavra-Chave a uma Aplicação Multilíngue

Desmistificando a Localização de Dados ([i18n](#)) no SAP Cloud Application Programming Model (CAP)

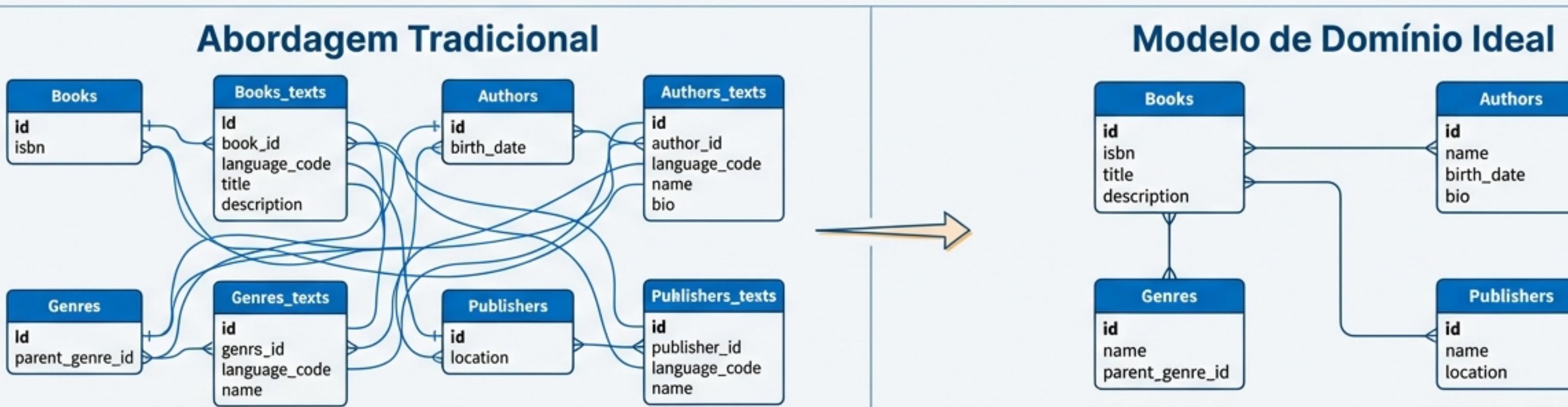
Uma análise aprofundada de como o CAP abstrai a complexidade do gerenciamento de dados em múltiplos idiomas, permitindo que desenvolvedores foquem na lógica de negócios.



O Desafio Universal: Dados em Múltiplos Idiomas

Gerenciar conteúdo multilíngue em aplicações é inherentemente complexo. A abordagem tradicional geralmente envolve:

- **Tabelas de Texto Separadas:** Criar uma tabela de ‘textos’ para cada entidade que precisa de tradução (ex: `Books` e `Books_texts`).
- **Chaves Estrangeiras e Joins:** Manter a consistência entre a **tabela principal** e a de textos, exigindo joins complexos em cada consulta.
- **Lógica de Fallback:** Implementar código para retornar um idioma padrão (ex: inglês) quando a tradução para o idioma do usuário não existe.
- **Poluição do Modelo de Dados:** O modelo de domínio fica sobrecarregado com entidades e associações técnicas, ofuscando a intenção de negócio.



A Solução do CAP: Capture a Intenção, Não a Implementação

O CAP adota a filosofia de 'O quê, não Como'. Em vez de modelar a complexidade, você simplesmente declara sua intenção de que um campo seja traduzível.

Modelo de dados "manual"

```
// Modelo de dados "manual"
entity Books {
    key ID : UUID;
    title : String; // Texto no idioma padrão
    descr : String;
    texts : Composition of many Books_texts on
        texts.book = $self;
}

entity Books_texts {
    key book : Association to Books;
    key locale : String;
    title : String;
    descr : String;
}
```

Modelo detalhado,
complexo e
propenso a erros.

Modelo de dados com CAP

```
// Modelo de dados com CAP
entity Books {
    key ID : UUID;
    title : localized String; // Apenas isso.
    descr : localized String;
    author : Association to Authors;
}
```

Modelo limpo, conciso e focado no
domínio de negócios. A palavra-chave
`localized` instrui o CAP a lidar com
toda a complexidade.



Nos Bastidores (1/3): Geração Automática da Entidade .texts

Ao compilar o modelo, o CAP identifica a palavra-chave **localized** e gera automaticamente uma nova entidade para armazenar as traduções. Esta entidade é nomeada [NomeDaEntidade].texts.

Características da Entidade Gerada:

- Herda do aspecto sap.common.TextsAspect, que inclui o campo locale.
- A chave primária é uma combinação de locale e as chaves da entidade original.
- Contém apenas os campos que foram marcados como **localized**.
- Uma associação de composição ('texts') é adicionada à entidade original para ligar a ela.

Código gerado pelo compilador CAP para Books.texts

```
// Código gerado pelo compilador CAP para Books.texts
entity Books.texts {localized
    key locale : sap.common.Locale; // Chave do idioma
    key ID      : UUID;        // Chave da entidade original (Books)
    title      : String;
    descr      : String; }
// Extensão adicionada à entidade Books original
extend entity Books with {
    texts : Composition of many Books.texts on texts.ID = ID;
}
```



Nos Bastidores (2/3): A Associação `localized` e `'\$user.locale'

Além da entidade `texts`, o CAP adiciona uma associação especial chamada `localized` à entidade original. Esta associação é o elo fundamental que conecta os dados aos textos no idioma do usuário atual.

Como Funciona:

- A associação `localized` faz um join com a entidade `texts`.
- A condição do join utiliza a pseudo-variável **\$user.locale**, que representa o idioma do usuário que fez a requisição.
- Isso permite que o sistema filtre dinamicamente as traduções corretas em tempo de execução, de forma transparente para o desenvolvedor.

```
// Extensão completa adicionada à entidade Books
extend entity Books with {
    // A composição para acessar todos os textos (para UIs de tradução)
    texts : Composition of many Books.texts on texts.ID = ID;

    // A associação inteligente para o idioma do usuário atual
    localized : Association to Books.texts
        on localized.ID = ID
        and localized.locale = $user.locale; // O ponto-chave!
}
```

O ponto-chave!

Nos Bastidores (3/3): A View `localized.` com Lógica de Fallback

O passo final é a criação de uma `view` de banco de dados chamada `localized.[NomeDaEntidade]`. Esta `view` é a que sua aplicação realmente consultará ao buscar dados localizados. Ela combina os dados da entidade original com os textos traduzidos e implementa a lógica de fallback.

A 'Mágica' do `coalesce`:

- A `view` utiliza a função SQL `coalesce(valor1, valor2)`.
- `coalesce` retorna o primeiro valor não nulo da lista.
- A consulta `coalesce(localized.title, title)` significa: 'Tente obter o título do texto traduzido (`localized.title`). Se não existir (for nulo), use o título da entidade original (`title`) como fallback.'



```
// Definição da view gerada pelo CAP
entity localized.Books as select from Books {
    *, // Todos os campos de Books
    // Lógica de fallback para cada campo localizado
    coalesce(localized.title, title) as title,
    coalesce(localized.descr, descr) as descr
};
```

Esta abordagem é extremamente performática, pois a lógica de fallback é executada diretamente no banco de dados.

Lendo Dados Localizados: A Ferramenta Certa para Cada Cenário

O CAP oferece diferentes formas de acessar os dados, dependendo do que você precisa realizar. Cada cenário utiliza uma parte diferente da estrutura que o `localized` gerou.



1. Para o Usuário Final

- Objetivo:** Mostrar os dados no idioma do usuário, com fallback para o idioma padrão.
- Como:** Consultar a [view localized.Books](#). O CAP faz isso automaticamente quando uma requisição chega com um locale de usuário.
- Resultado:** Transparente e automático.



2. Para UIs de Tradução

- Objetivo:** Ler e escrever textos em *todos* os idiomas, independentemente do usuário atual.
- Como:** Utilizar a composição [texts](#) para acessar diretamente a entidade [Books.texts](#).
- Resultado:** Controle total sobre todas as traduções.



3. Para Código Agnóstico (Processamento em Lote, etc.)

- Objetivo:** Acessar os dados no idioma padrão, sem a sobrecarga da lógica de tradução.
- Como:** Consultar a entidade base [Books](#) diretamente.
- Resultado:** Rápido e direto, ignorando a localização.

Escrevendo Dados Localizados: Operações de Criação, Atualização e Deleção

Como a entidade `texts` está ligada via composição, você pode manipular as traduções usando inserções/atualizações profundas (deep inserts/upserts) ou navegando diretamente para a coleção `texts`.

Criar Entidade com Traduções (Deep Insert)

```
POST /service/Books  
Content-Type: application/json  
  
{  
  "ID": "...",  
  "title": "Wuthering Heights",  
  "texts": [  
    { "locale": "de", "title": "Sturmhöhe" },  
    { "locale": "fr", "title": "Les Hauts de Hurlevent" }  
  ]  
}
```

Atualizar uma Tradução (PATCH)

```
PATCH /service/Books(ID=...)/texts(ID=..., locale='de')  
Content-Type: application/json  
  
{ "title": "Ein ganz neuer Titel" }
```

Adicionar/Atualizar uma Tradução Específica

```
POST /service/Books(ID=...)/texts  
Content-Type: application/json  
  
{ "locale": "de", "title": "Neuer Titel" }
```

Deletar uma Tradução

```
DELETE /service/Books(ID=...)/texts(ID=..., locale='de')
```

Carga Inicial de Dados com Arquivos .csv

Para popular suas entidades com dados iniciais, você precisará de dois arquivos .csv: um para os dados na língua padrão e outro para as traduções. O CAP identifica os arquivos pela convenção de nomenclatura.

1. Arquivo de Dados Padrão: [NomeDaEntidade].csv

Contém os dados principais, incluindo os textos no idioma de fallback.

```
// db/data/my.bookshop-Books.csv
ID;title;descr;author_ID;stock
201;Wuthering Heights;"Wuthering Heights, Emily
Brontë's only novel...";101;12
207;Jane Eyre;"Jane Eyre is a novel by English
writer...";107;11
```

2. Arquivo de Traduções: [NomeDaEntidade]_texts.csv

O sufixo **_texts** é fundamental. Deve conter a coluna **locale** e as chaves da entidade original.

```
// db/data/my.bookshop-Books_texts.csv
ID;locale;title;descr
201;de;Sturmhöhe;"Sturmhöhe (Originaltitel:
Wuthering Heights) ist der einzige...";
201;fr;Les Hauts de Hurlevent;"Les Hauts de
Hurlevent (titre original : Wuthering...)";
207;de;Jane Eyre;"Jane Eyre. Eine
Autobiographie (Originaltitel: Jane Eyre...)";
```



Lidando com Dados Localizados Aninhados

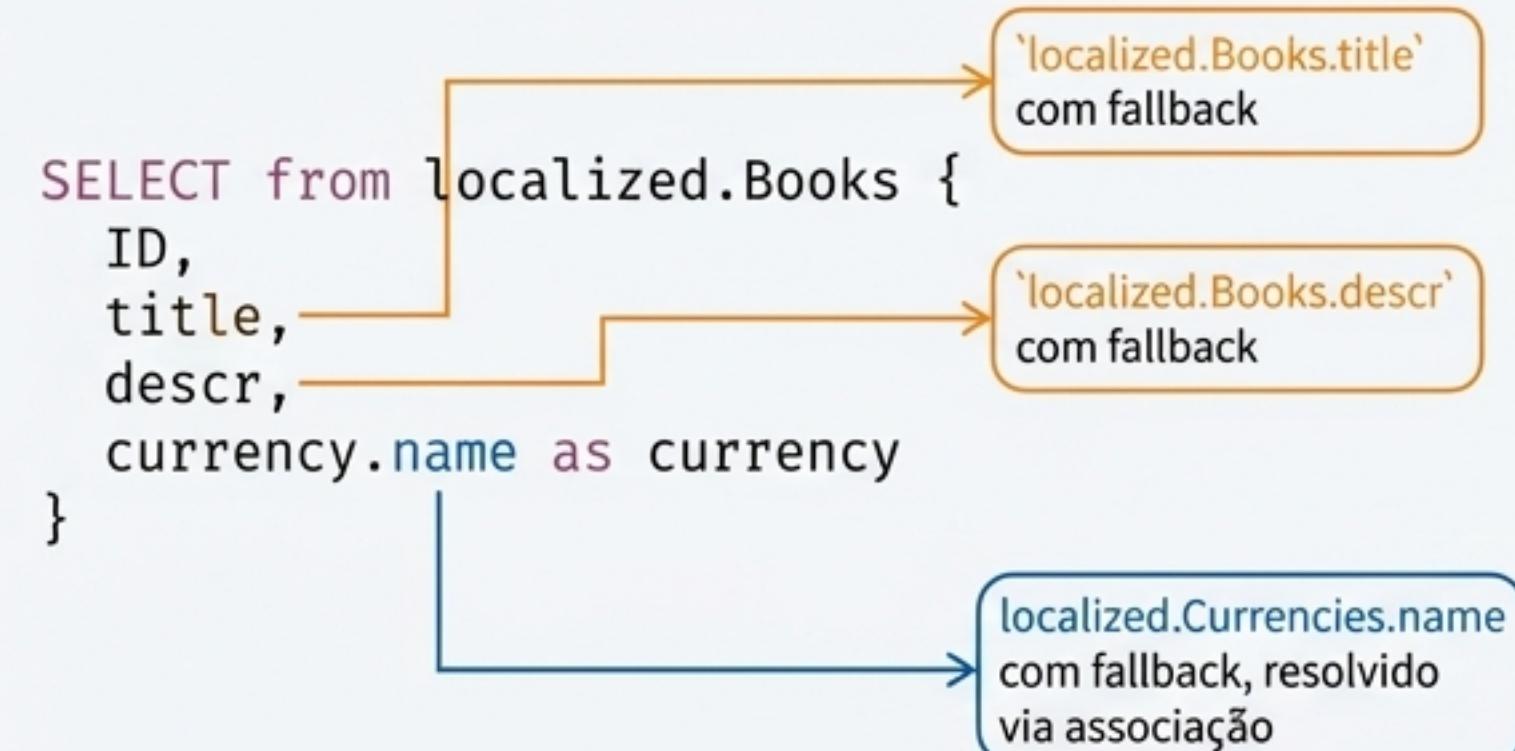
A abordagem do CAP para localização se estende de forma transparente a associações. Se uma entidade como `Books` tem uma associação para outra entidade com campos localizados (por exemplo, `Currencies`), o CAP resolve isso automaticamente.

****Como Funciona**:**

- Na `view` `localized.Books`, a associação para `Currencies` é automaticamente redirecionada para a view `localized.Currencies`.
- Isso permite que você faça consultas que expandem associações e obtenha dados localizados de múltiplas entidades com uma lógica de fallback independente para cada uma.

****Exemplo de Consulta**:**

Esta única consulta busca o título e a descrição do livro, bem como o nome da moeda, todos devidamente localizados para o idioma do usuário.



Considerações para SAP Fiori: Suporte a Rascunhos (Drafts)

Consideração Especial:

Ao habilitar o modo de rascunho (draft) do Fiori em uma entidade com dados localizados, uma consideração especial é necessária. Por padrão, a entidade `.texts` gerada tem uma chave composta (ID, locale), o que não é ideal para o mecanismo de draft do Fiori.

A Solução:

Use a anotação `@fiori.draft.enabled` na sua entidade de *base* no modelo de persistência (ex: db/schema.cds).

O compilador do CAP irá então:

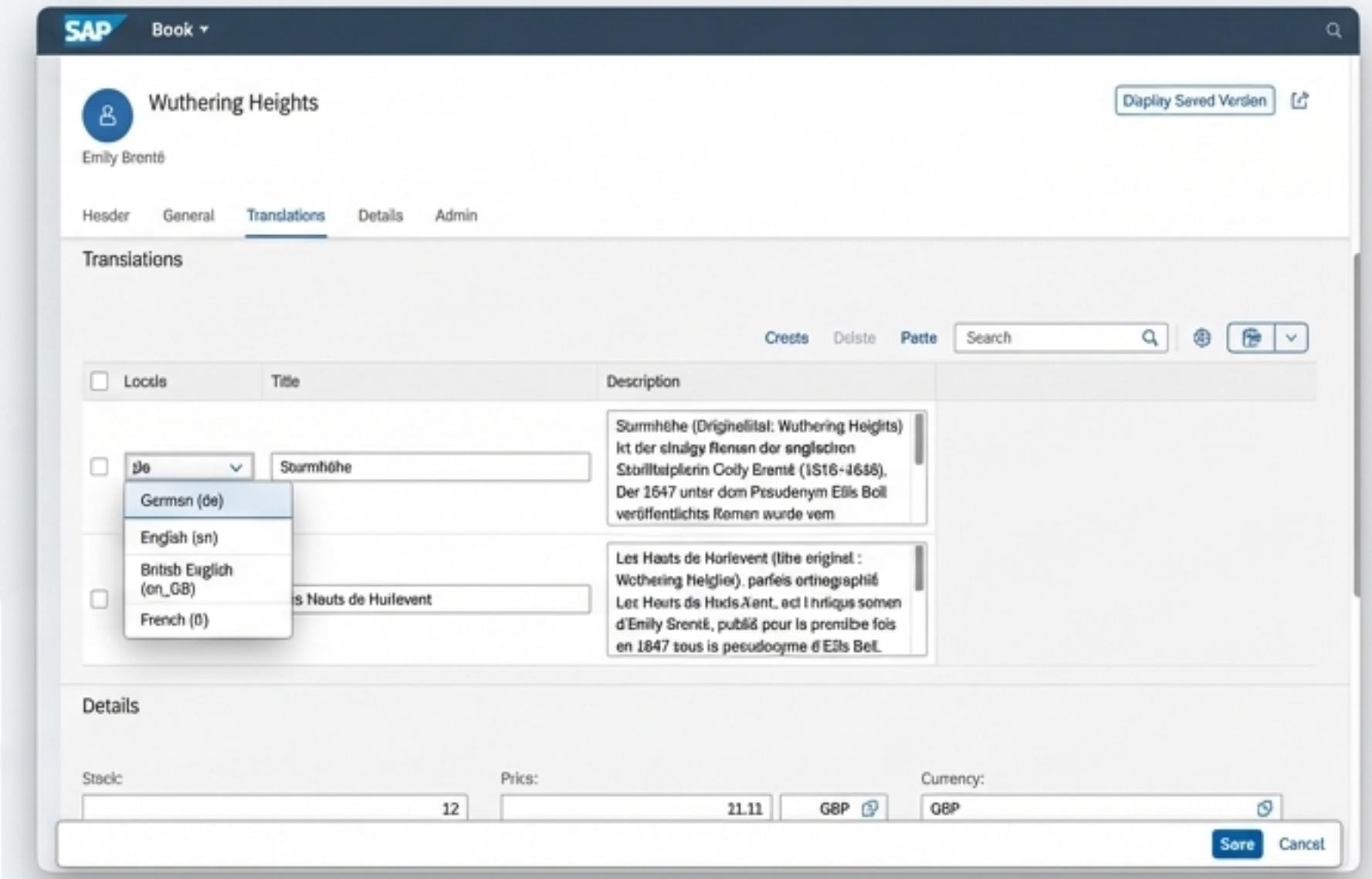
1. Adicionar uma chave primária técnica do tipo `UUID` (ex: `ID_texts`) à entidade `.texts`.
2. Remover `locale` da chave primária.

Isso garante a compatibilidade total com o fluxo de edição de rascunhos do Fiori, permitindo a edição de traduções em uma UI como a da imagem.

Código de Exemplo:

```
// Em db/schema.cds
using {cuid, managed} from '@sap/cds/common';

@fiori.draft.enabled // Habilita o draft para a entidade base e seus textos
entity Books : cuid, managed {
    title : localized String;
    descr : localized String;
    // ...
}
```



O Poder da Abstração: As Vantagens da Abordagem do CAP

A palavra-chave `localized` é mais do que um atalho. Ela representa a filosofia do CAP de focar na intenção de negócio e automatizar padrões técnicos complexos.



Modelo de Domínio Limpo: Seu `schema.cds` permanece focado na lógica de negócio, sem poluição de entidades de texto e associações técnicas.



Complexidade Abstraída: A geração de tabelas, `views` com `coalesce` e associações é totalmente automática.

‘localized’

Lógica de Fallback Automática: Garante que sempre haverá um dado a ser exibido, sem necessidade de código customizado.



Alto Desempenho: A resolução de traduções e o fallback acontecem no nível do banco de dados, a camada mais eficiente para isso.



Consistência e Padronização: Garante que a localização seja implementada da mesma forma em toda a aplicação.