FAP 2024 – UFRN – Introdução a bancos de dados (1) Prof. José Alfredo Costa

Exemplo Inicial: O Problema da Livraria

Imagine que temos uma livraria e queremos armazenar informações sobre nossos livros em uma única tabela:

| **Código** | **Título** | **Autor** | **Editora** | **Ano** | **ISBN** |
| --- | --- | --- | --- | --- | --- |
| 1 | O Senhor dos Anéis | J.R.R. Tolkien | Martins Fontes | 2001 | 978-8533613379 |
| 2 | O Hobbit | J.R.R. Tolkien | Martins Fontes | 2019 | 978-8578277727 |
| 3 | 1984 | George Orwell | Companhia das Letras | 2009 | 978-8535914849 |

Neste cenário, notamos que há informações redundantes. Por exemplo, o autor "J.R.R. Tolkien" e a editora "Martins Fontes" aparecem repetidamente. Isso pode levar a problemas como:

1. Desperdício de espaço de armazenamento
2. Dificuldade na atualização de dados (se o nome da editora mudar, precisamos atualizar em vários lugares)
3. Possibilidade de inconsistências (se o nome do autor for escrito de forma diferente em diferentes entradas)

Agora, vamos explicar os conceitos do material, mostrando como a organização em um banco de dados pode resolver esses problemas.

Slide 1-3: Introdução e Agenda

Explicação: Nesta aula, vamos explorar os fundamentos de bancos de dados, entendendo por que são essenciais na computação moderna e como eles nos ajudam a organizar e gerenciar informações de forma eficiente.

Slides 4-6: Por que usar banco de dados?

Explicação: Com o aumento exponencial do volume de informações no mundo digital, gerenciar dados tornou-se um desafio significativo. Bancos de dados surgem como uma solução para esse problema, oferecendo várias vantagens:

1. Redução do espaço de armazenamento: Ao eliminar redundâncias, economizamos espaço.
2. Facilidade de acesso e utilização: Dados bem organizados são mais fáceis de encontrar e usar.
3. Aumento da velocidade de pesquisa: Estruturas otimizadas permitem buscas rápidas.
4. Eliminação de redundância: Informações são armazenadas uma única vez, reduzindo inconsistências.

No nosso exemplo da livraria, poderíamos criar tabelas separadas para autores, editoras e livros, conectando-as através de relacionamentos. Isso resolveria os problemas de redundância e inconsistência.

Slides 7-8: O que é um banco de dados

Explicação: Um banco de dados é mais do que apenas uma coleção de dados. É uma estrutura organizada e inter-relacionada de informações que tem um propósito específico. No caso da nossa livraria, o propósito seria gerenciar o catálogo de livros de forma eficiente.

Slides 9-10: Dado x Informação

Explicação: É importante distinguir entre dados e informações:

* Dados são elementos brutos, como o número "2001" na nossa tabela de livros.
* Informação é o significado que atribuímos aos dados quando os contextualizamos, como "O livro 'O Senhor dos Anéis' foi publicado em 2001 pela editora Martins Fontes".

Slides 11-12: Exemplos de Banco de Dados

Explicação: Bancos de dados são onipresentes em nossa vida cotidiana. Alguns exemplos incluem:

1. Sistema bancário: Gerencia contas, transações e informações dos clientes.
2. Base de dados de biblioteca: Cataloga livros, gerencia empréstimos e membros.
3. Sistema de reserva de passagens aéreas: Administra voos, reservas e informações dos passageiros.
4. Sistema de e-commerce: Gerencia produtos, pedidos e informações dos clientes.

Slides 13-18: Modelagem Conceitual, Lógica e Física

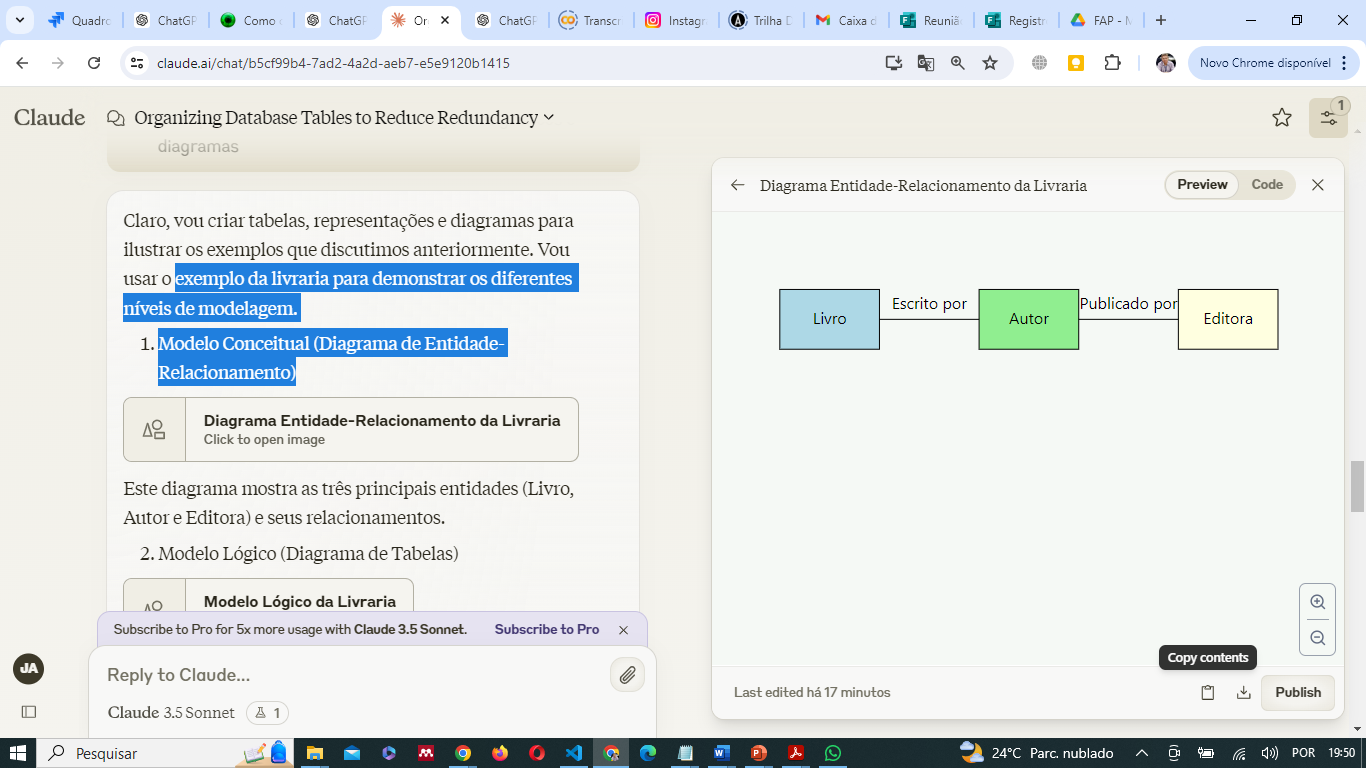
Explicação: A criação de um banco de dados eficiente envolve três níveis de modelagem:

1. **Modelagem Conceitual**: É o nível mais abstrato, próximo à compreensão humana. Para nossa livraria, identificaríamos entidades como Livro, Autor e Editora.
2. **Modelagem Lógica**: Aqui, definimos como os dados serão estruturados. Por exemplo:
   * Tabela Autor (ID, Nome)
   * Tabela Editora (ID, Nome)
   * Tabela Livro (ID, Título, Ano, ISBN, AutorID, EditoraID)
3. **Modelagem Física**: Este é o nível de implementação, onde escolhemos o Sistema Gerenciador de Banco de Dados (SGBD) e escrevemos o código SQL para criar as tabelas.

Conclusão: Ao organizar nossos dados em um banco de dados bem estruturado, resolvemos os problemas de redundância e inconsistência, facilitamos a manutenção e melhoramos a eficiência do armazenamento e recuperação de informações. Isso é crucial para sistemas de informação modernos, desde pequenas livrarias até grandes corporações.

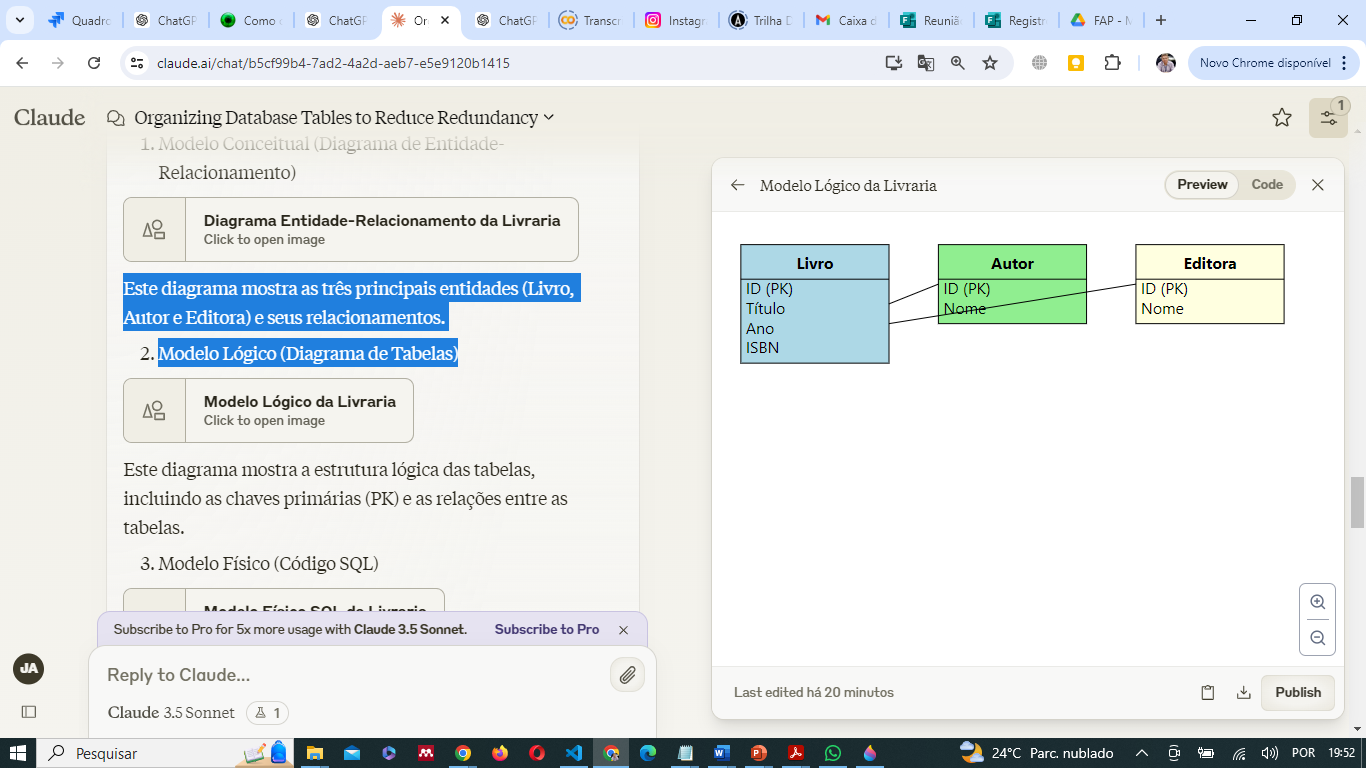
Exemplo da livraria para demonstrar os diferentes níveis de modelagem.

1. Modelo Conceitual (Diagrama de Entidade-Relacionamento)



Este diagrama mostra as três principais entidades (Livro, Autor e Editora) e seus relacionamentos.

1. Modelo Lógico (Diagrama de Tabelas)



Este diagrama mostra a estrutura lógica das tabelas, incluindo as chaves primárias (PK) e as relações entre as tabelas.

1. Modelo Físico (Código SQL)

CREATE TABLE Autor (

ID INT PRIMARY KEY,

Nome VARCHAR(100) NOT NULL

);

CREATE TABLE Editora (

ID INT PRIMARY KEY,

Nome VARCHAR(100) NOT NULL

);

CREATE TABLE Livro (

ID INT PRIMARY KEY,

Titulo VARCHAR(200) NOT NULL,

Ano INT,

ISBN VARCHAR(13),

AutorID INT,

EditoraID INT,

FOREIGN KEY (AutorID) REFERENCES Autor(ID),

FOREIGN KEY (EditoraID) REFERENCES Editora(ID)

);

Este código SQL representa a implementação física das tabelas em um sistema de gerenciamento de banco de dados relacional.

1. Exemplo de Dados nas Tabelas

Agora, vou mostrar como os dados ficariam organizados nestas tabelas:

*-- Tabela Autor*

INSERT INTO Autor (ID, Nome) VALUES

(1, 'J.R.R. Tolkien'),

(2, 'George Orwell');

*-- Tabela Editora*

INSERT INTO Editora (ID, Nome) VALUES

(1, 'Martins Fontes'),

(2, 'Companhia das Letras');

*-- Tabela Livro*

INSERT INTO Livro (ID, Titulo, Ano, ISBN, AutorID, EditoraID) VALUES

(1, 'O Senhor dos Anéis', 2001, '978-8533613379', 1, 1),

(2, 'O Hobbit', 2019, '978-8578277727', 1, 1),

(3, '1984', 2009, '978-8535914849', 2, 2);

*-- Consulta para mostrar informações completas dos livros*

SELECT L.Titulo, A.Nome AS Autor, E.Nome AS Editora, L.Ano, L.ISBN

FROM Livro L

JOIN Autor A ON L.AutorID = A.ID

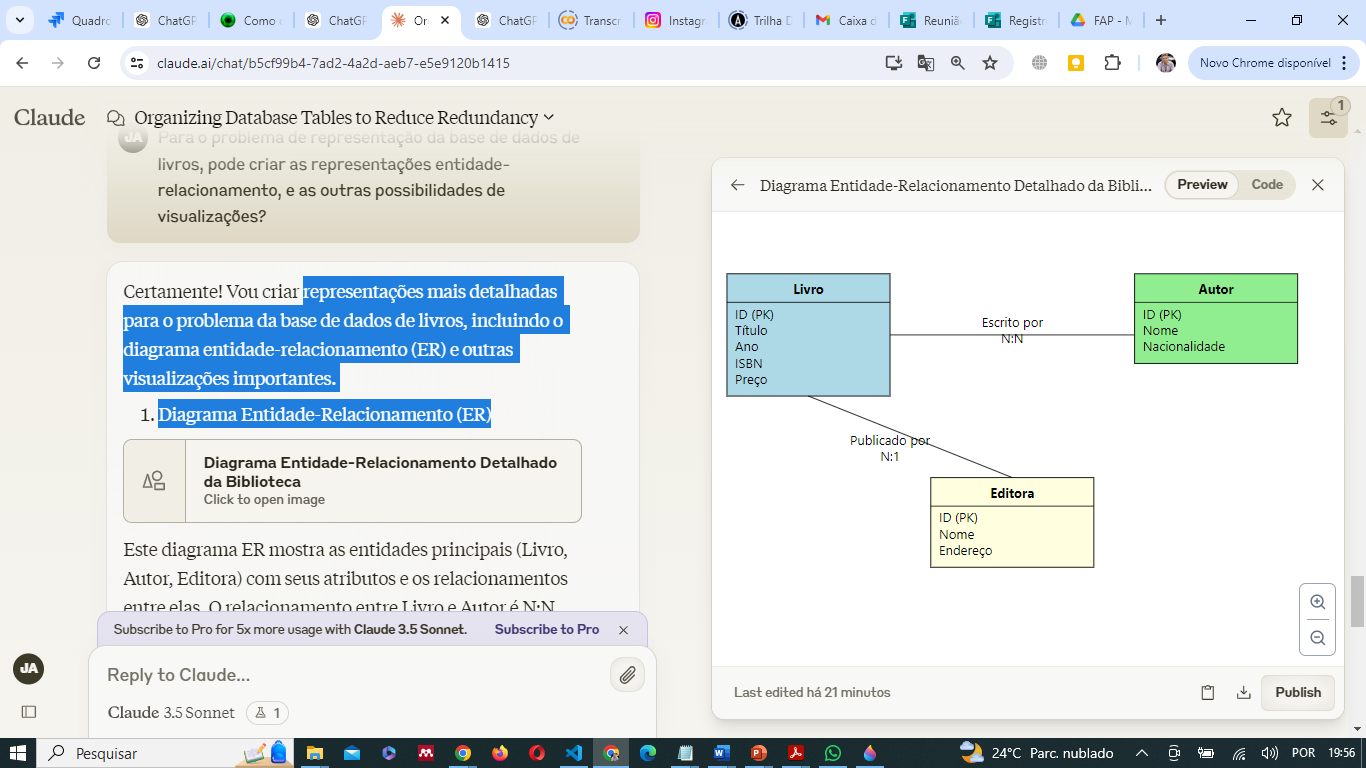
JOIN Editora E ON L.EditoraID = E.ID;

Esta representação mostra como os dados são inseridos nas tabelas e como podemos realizar uma consulta para obter informações completas sobre os livros, incluindo detalhes do autor e da editora.

Estes modelos e representações demonstram como podemos organizar eficientemente os dados da livraria, eliminando redundâncias e facilitando a manutenção e consulta das informações. A modelagem em diferentes níveis (conceitual, lógico e físico) nos permite entender e implementar o banco de dados de forma estruturada e eficaz.

Representações mais detalhadas para o problema da base de dados de livros, incluindo o diagrama entidade-relacionamento (ER) e outras visualizações importantes.

1. Diagrama Entidade-Relacionamento (ER)



Este diagrama ER mostra as entidades principais (Livro, Autor, Editora) com seus atributos e os relacionamentos entre elas. O relacionamento entre Livro e Autor é N:N (muitos para muitos), enquanto entre Livro e Editora é N:1 (muitos para um).

1. Modelo Relacional

Agora, vamos representar o modelo lógico usando o modelo relacional:

Livro (ID, Título, Ano, ISBN, Preço, EditoraID)

EditoraID referencia Editora(ID)

Autor (ID, Nome, Nacionalidade)

Editora (ID, Nome, Endereço)

Livro\_Autor (LivroID, AutorID)

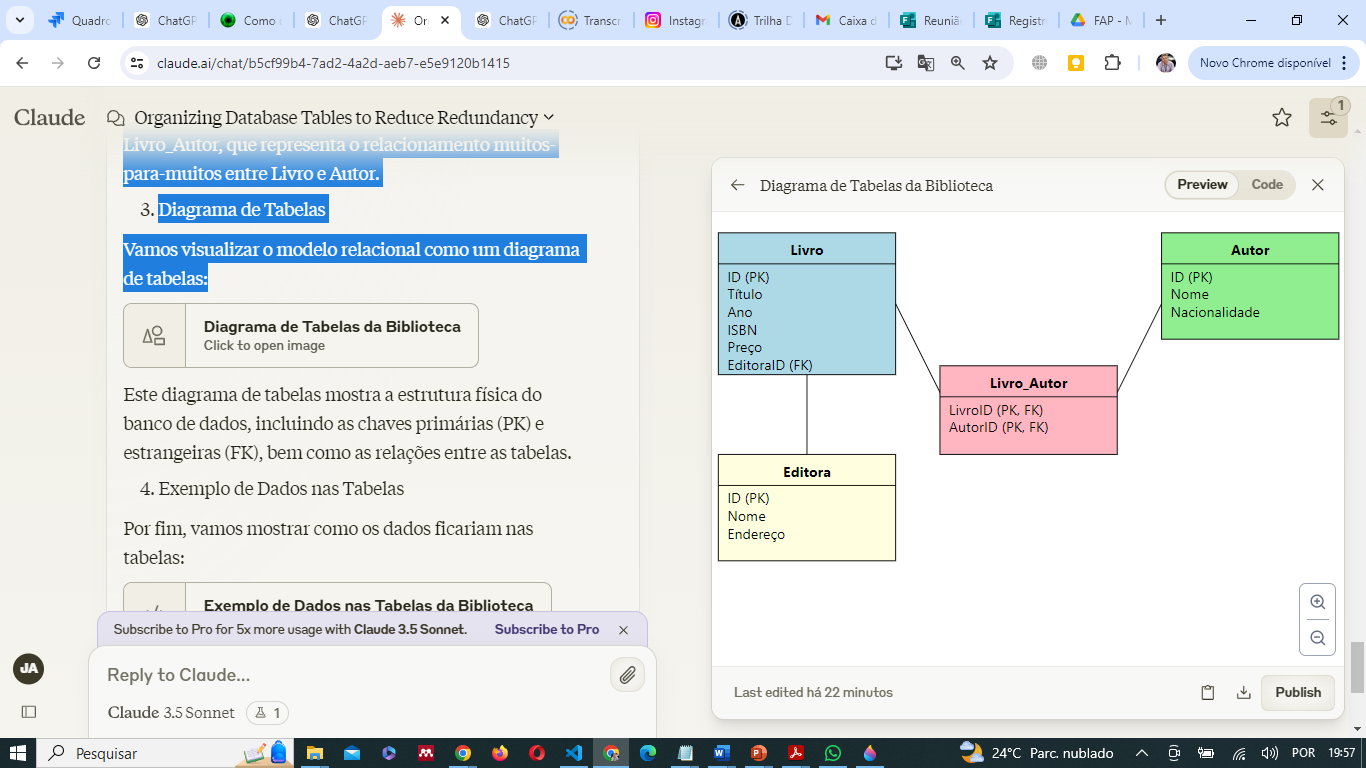
LivroID referencia Livro(ID)

AutorID referencia Autor(ID)

Este modelo relacional mostra como as entidades são transformadas em tabelas. Note a tabela adicional Livro\_Autor, que representa o relacionamento muitos-para-muitos entre Livro e Autor.

1. Diagrama de Tabelas

Vamos visualizar o modelo relacional como um diagrama de tabelas:



Este diagrama de tabelas mostra a estrutura física do banco de dados, incluindo as chaves primárias (PK) e estrangeiras (FK), bem como as relações entre as tabelas.

1. Exemplo de Dados nas Tabelas

Por fim, vamos mostrar como os dados ficariam nas tabelas:

*-- Tabela Editora*

INSERT INTO Editora (ID, Nome, Endereço) VALUES

(1, 'Martins Fontes', 'Rua Conselheiro Ramalho, 330, São Paulo'),

(2, 'Companhia das Letras', 'Rua Bandeira Paulista, 702, São Paulo');

*-- Tabela Autor*

INSERT INTO Autor (ID, Nome, Nacionalidade) VALUES

(1, 'J.R.R. Tolkien', 'Britânica'),

(2, 'George Orwell', 'Britânica');

*-- Tabela Livro*

INSERT INTO Livro (ID, Título, Ano, ISBN, Preço, EditoraID) VALUES

(1, 'O Senhor dos Anéis', 2001, '978-8533613379', 89.90, 1),

(2, 'O Hobbit', 2019, '978-8578277727', 49.90, 1),

(3, '1984', 2009, '978-8535914849', 39.90, 2);

*-- Tabela Livro\_Autor*

INSERT INTO Livro\_Autor (LivroID, AutorID) VALUES

(1, 1),

(2, 1),

(3, 2);

*-- Consulta para mostrar informações completas dos livros*

SELECT L.Título, A.Nome AS Autor, E.Nome AS Editora, L.Ano, L.ISBN, L.Preço

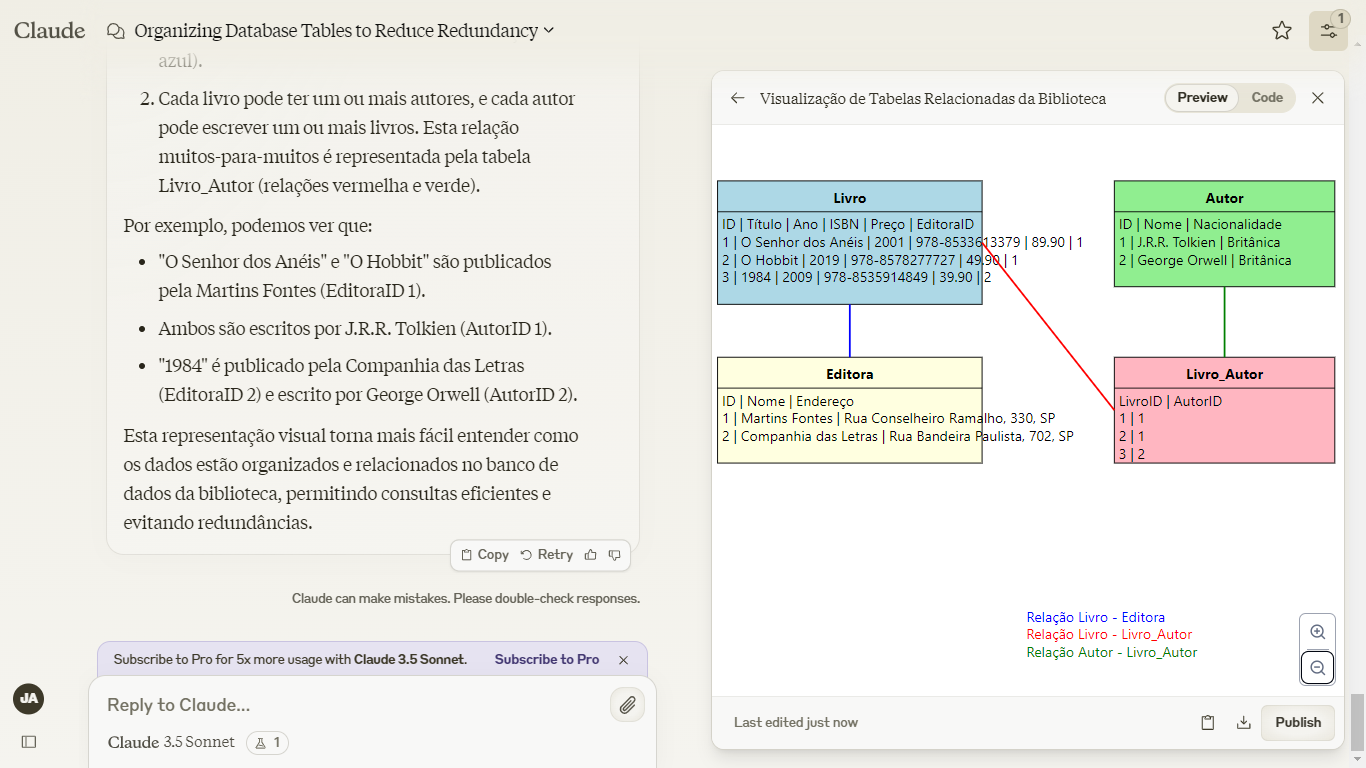
FROM Livro L

JOIN Livro\_Autor LA ON L.ID = LA.LivroID

JOIN Autor A ON LA.AutorID = A.ID

JOIN Editora E ON L.EditoraID = E.ID;

Estas representações fornecem uma visão abrangente da estrutura do banco de dados da biblioteca, desde o nível conceitual (diagrama ER) até o nível físico (tabelas com dados). Elas demonstram como a informação é organizada, eliminando redundâncias e permitindo consultas eficientes.



**Tabelas do Banco de Dados da Biblioteca**

**Tabela Livro**

| **ID** | **Título** | **Ano** | **ISBN** | **Preço** | **EditoraID** |
| --- | --- | --- | --- | --- | --- |
| 1 | O Senhor dos Anéis | 2001 | 978-8533613379 | R$ 89,90 | 1 |
| 2 | O Hobbit | 2019 | 978-8578277727 | R$ 49,90 | 1 |
| 3 | 1984 | 2009 | 978-8535914849 | R$ 39,90 | 2 |

**Tabela Autor**

| **ID** | **Nome** | **Nacionalidade** |
| --- | --- | --- |
| 1 | J.R.R. Tolkien | Britânica |
| 2 | George Orwell | Britânica |

**Tabela Editora**

| **ID** | **Nome** | **Endereço** |
| --- | --- | --- |
| 1 | Martins Fontes | Rua Conselheiro Ramalho, 330, SP |
| 2 | Companhia das Letras | Rua Bandeira Paulista, 702, SP |

**Tabela Livro\_Autor**

| **LivroID** | **AutorID** |
| --- | --- |
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |

Explicação da visualização:

1. Tabela Livro (azul claro):
   * Contém informações sobre os livros, incluindo ID, título, ano, ISBN, preço e EditoraID.
   * O EditoraID é uma chave estrangeira que se relaciona com a tabela Editora.
2. Tabela Autor (verde claro):
   * Armazena dados dos autores, como ID, nome e nacionalidade.
3. Tabela Editora (amarelo claro):
   * Contém informações sobre as editoras, incluindo ID, nome e endereço.
4. Tabela Livro\_Autor (rosa claro):
   * Esta é uma tabela de junção que representa o relacionamento muitos-para-muitos entre Livro e Autor.
   * Contém LivroID e AutorID, que são chaves estrangeiras referenciando as tabelas Livro e Autor, respectivamente.

Relações (representadas por linhas coloridas):

* Linha Azul: Conecta a tabela Livro à tabela Editora através do EditoraID.
* Linha Vermelha: Conecta a tabela Livro à tabela Livro\_Autor através do ID do livro.
* Linha Verde: Conecta a tabela Autor à tabela Livro\_Autor através do ID do autor.

Esta visualização ajuda a entender como os dados estão interconectados:

1. Cada livro está associado a uma editora (relação azul).
2. Cada livro pode ter um ou mais autores, e cada autor pode escrever um ou mais livros. Esta relação muitos-para-muitos é representada pela tabela Livro\_Autor (relações vermelha e verde).

Por exemplo, podemos ver que:

* "O Senhor dos Anéis" e "O Hobbit" são publicados pela Martins Fontes (EditoraID 1).
* Ambos são escritos por J.R.R. Tolkien (AutorID 1).
* "1984" é publicado pela Companhia das Letras (EditoraID 2) e escrito por George Orwell (AutorID 2).

Esta representação visual torna mais fácil entender como os dados estão organizados e relacionados no banco de dados da biblioteca, permitindo consultas eficientes e evitando redundâncias.