

Relatório do projeto final – Gabriel Araújo

Introdução

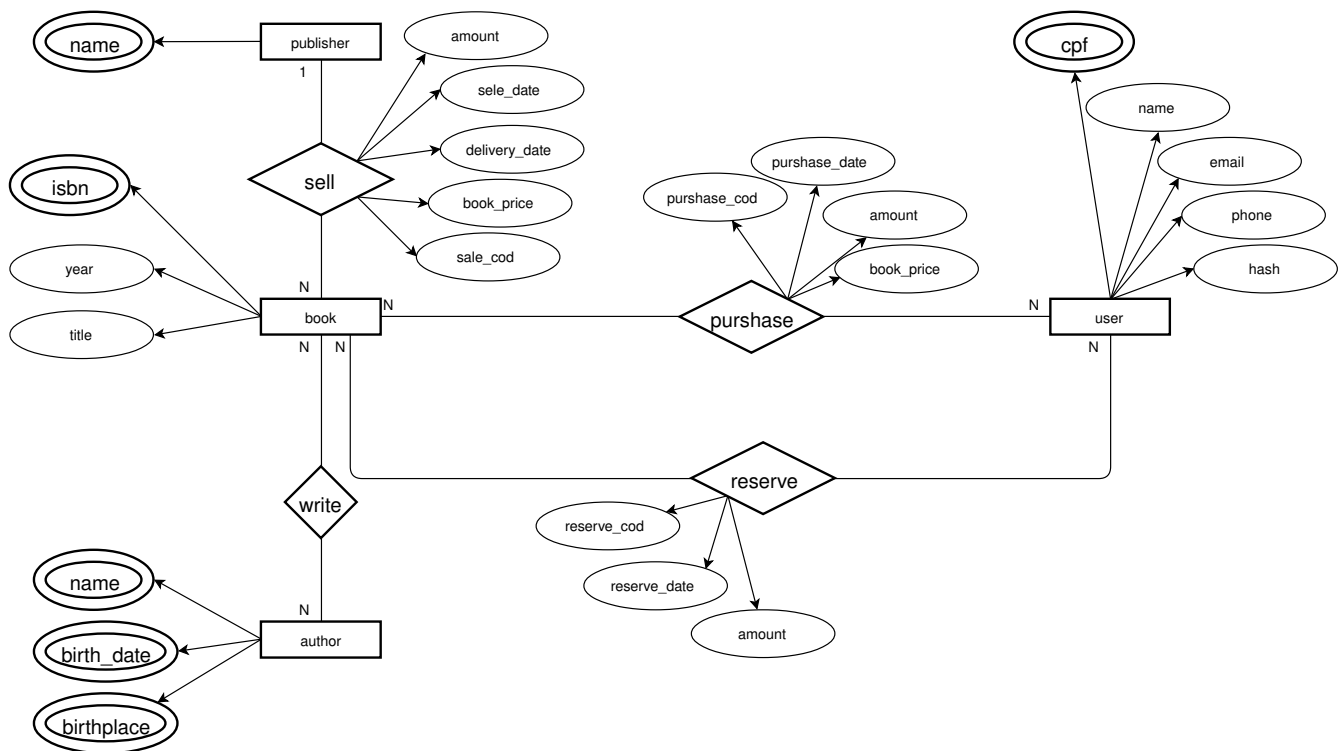
Para aplicar todas as funcionalidade aprendidas na disciplina fiz um banco de dados para uma livraria. O banco de dados é responsável por registrar os livros, autores, editoras e usuários, bem como a relação entre eles.

Objetivos

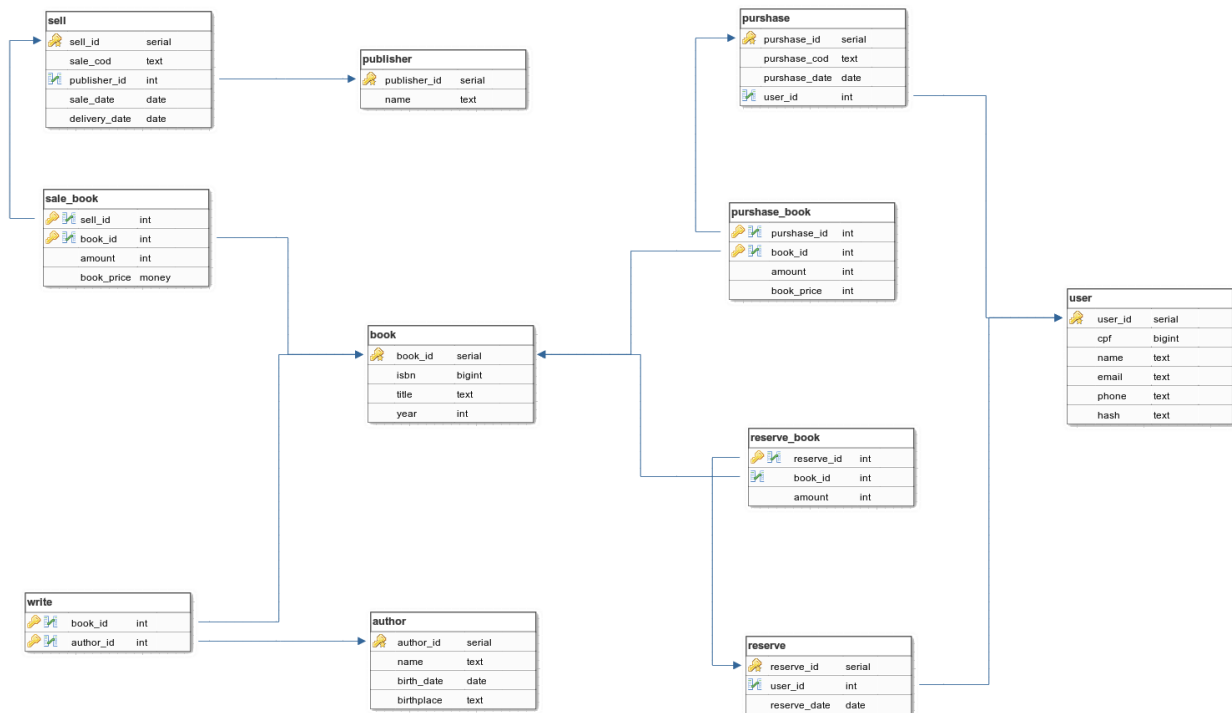
Os objetivos do banco de dados deste projeto são:

1. Gerenciar o cadastro de livros, autores, editoras e clientes.
2. Gerenciar a entrada e saída de livros no estoque da livraria.
3. Permitir vendas de livros que estejam em estoque.
4. Permitir reservas de livros por parte dos usuários.
5. Lançar relatórios diários do balanço financeiro.
6. Lançar relatórios sobre a quantidade de cada livro no estoque.

Modelo conceitual



Modelo Lógico



Usei chaves primária artificiais em todas as tabelas para facilitar a construção de chaves estrangeiras e para tornar as consultas mais ágeis.

As relações sell, write, purchase, e reserve foram traduzidas em tabelas com chaves compostas.

Além da relação entre as tabelas, para manter a integridade do banco de dados é necessário garantir que uma venda só seja feita caso haja livros no estoque. E, por motivo de segurança, é necessário que o password do usuário seja salvo em forma de hash.

Modelo Físico

O modelo físico do banco de dados está nos arquivos [scheme.sql](#), [populate.sql](#) e [query.sql](#). O primeiro arquivo contém as instruções para a criação do banco de dados físico. O segundo contém instruções para povoá-lo. O terceiro gera relatórios de estoque e de balanço.

No arquivo scheme todas as tabelas são criadas. Além disso são criados três gatilhos: um que aplica uma função hash na senha do usuário antes de salvá-la, e outros dois, que executam a mesma função, porém em tabelas diferentes, para verificar a disponibilidade de livros para compra ou para reserva.

Constraints para verificar a validade do cpf, do e-mail e do telefone do usuário também foram adicionados no modelo físico.

Três views foram criadas no modelo físico:

1. 'book_stock': para agrupar a quantidade de livros no estoque por livro;
2. 'book_availability': para agrupar a quantidade de livros no estoque menos os que foram reservados;

3. 'balance': para agrupar o balanço financeiro por dia.

As duas primeiras views são úteis nos gatilhos que garantem que nenhum livro seja vendido quando não estiver presente no estoque. A terceira é útil para realizar balanços financeiros.

Conclusão

Um banco de dados bem planejado e com todas as restrições do plano de negócios implementadas por constraints e gatilhos permite que dados sejam adicionados ou removidos sem gerar inconsistências do banco de dados.