



Exercício Livraria

1. Iniciar o Oracle LiveSQL.
2. **Executar** os **scripts** disponibilizados para criar uma base de dados (BD) sobre uma **livraria**. A BD é implementada de acordo com o modelo relacional da Figura 1.

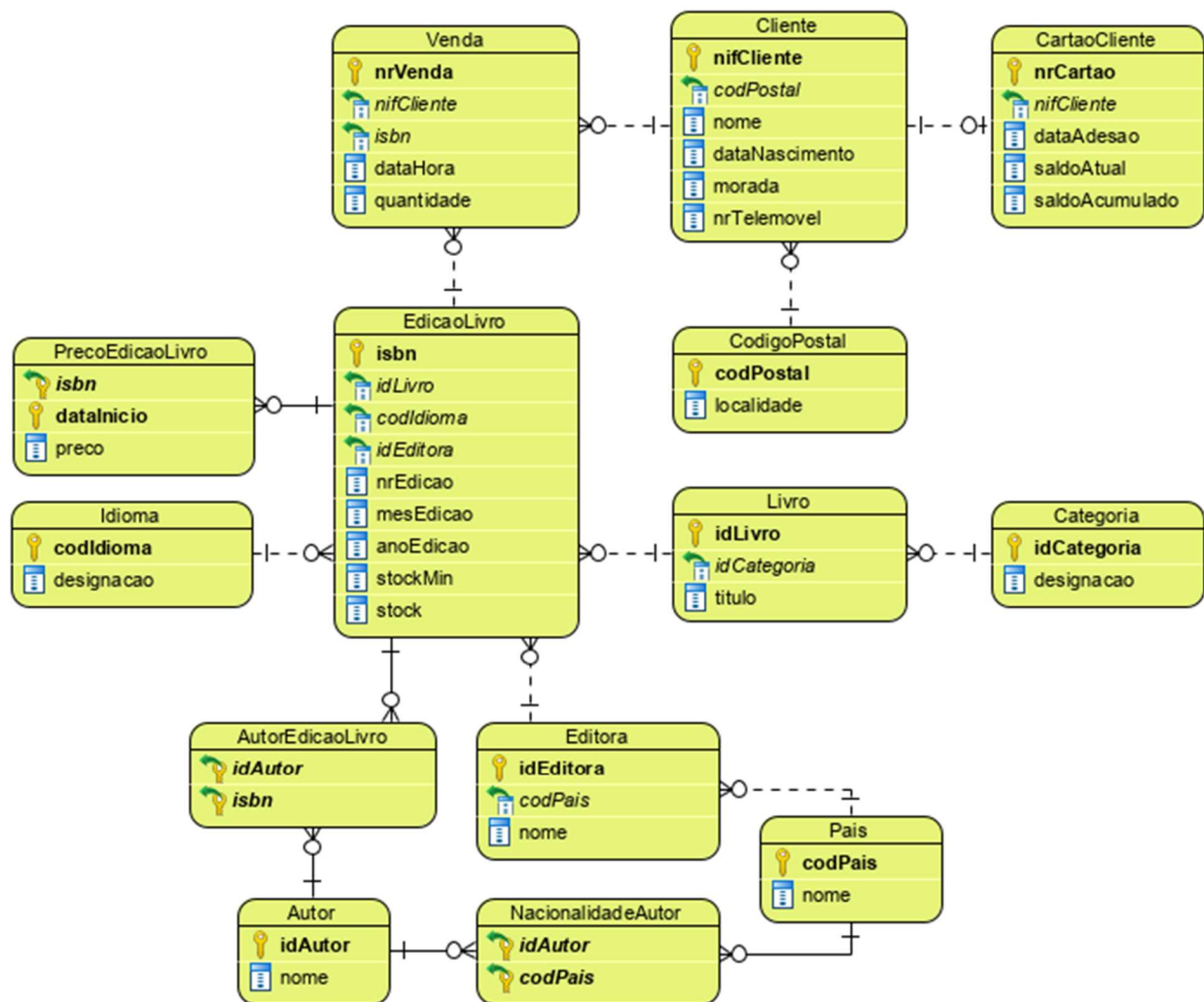


Figura 1 - Modelo Relacional

3. **Criar** um **script** com código PL/SQL para **implementar** uma **função**, designada **fncTotalClientes**, para retornar o número total de clientes. Caso não existam clientes registados, a função deve retornar o valor NULL. **Testar** adequadamente a função implementada, através de **blocos anónimos**. Para visualizar o resultado, executar o comando SET SERVEROUTPUT ON.



4. Criar um **novo script** PL/SQL para **implementar** uma **função**, designada **fncTemLivrosEditora**, para verificar se existem livros de uma dada editora em *stock*. A função deve receber, por parâmetro, o identificador da editora e tem de retornar um valor booleano, *true* ou *false*. Se o parâmetro fornecido for inválido, a função deve retornar o valor NULL, usando o mecanismo de exceções. **Testar** adequadamente a função implementada, através de **blocos anónimos**.
5. Criar um **novo script** PL/SQL para **implementar** uma **função**, designada **fncClienteInfo**, para retornar a informação pessoal de um dado cliente, recebido por parâmetro. Se o parâmetro fornecido for inválido, a função deve retornar o valor NULL, usando o mecanismo de exceções. **Testar** adequadamente a função implementada, através de **blocos anónimos**.
6. Criar um **novo script** PL/SQL para **implementar** uma **função**, designada **fncStockAnoEditora**, para retornar o *stock* dos livros editados por uma dada editora num dado ano. A função deve receber, por parâmetro, o identificador da editora e o ano. Este último parâmetro deve ser opcional na invocação da função e o seu valor por omissão deve ser o ano atual. Se qualquer parâmetro fornecido for inválido, a função deve retornar o valor NULL, usando o mecanismo de exceções. **Testar** adequadamente a função implementada, através de **blocos anónimos**.
7. Criar um **novo script** PL/SQL para **implementar** uma **função**, designada **fncPrecoVenda**, para retornar o preço aplicado a uma dada venda, recebida por parâmetro. Se o parâmetro fornecido for inválido, a função deve retornar o valor NULL, usando o mecanismo de exceções. **Testar** adequadamente a função implementada, através de **blocos anónimos**.
8. Criar um **novo script** PL/SQL para **implementar** uma **função**, designada **fncRegistrarNovoCliente**, para registar um novo cliente, em que os dados são recebido como parâmetros. Em caso de sucesso, a função deve retornar o identificador do novo cliente. Se os parâmetros fornecidos forem inválidos, a função deve retornar o valor NULL, usando o mecanismo de exceções. **Testar** adequadamente a função implementada, através de **blocos anónimos**.
9. Criar um **novo script** PL/SQL para **implementar** uma **função**, designada **fncLivrosEditora**, que deve retornar um cursor com livros de uma dada editora em *stock*. A função deve receber, por parâmetro, o identificador da editora. Se o parâmetro fornecido for inválido, a função deve retornar um cursor vazio. Use o mecanismo de exceções. **Testar** adequadamente a função implementada, através de **blocos anónimos**.
10. Criar um **novo script** PL/SQL para **implementar** uma **função**, designada **fncClientesAtivos**, que deve retornar um cursor com os clientes com vendas num dado período, incluindo o respetivo valor total. A função deve receber, por parâmetro, o período em análise. Se o período fornecido for inválido ou não houve vendas nesse período, a função deve retornar um cursor vazio. **Testar** adequadamente a função implementada, através de **blocos anónimos**.