

Desenvolvimento de Aplicações de Banco de Dados JDBC, JPA, Hibernate (Exercícios)

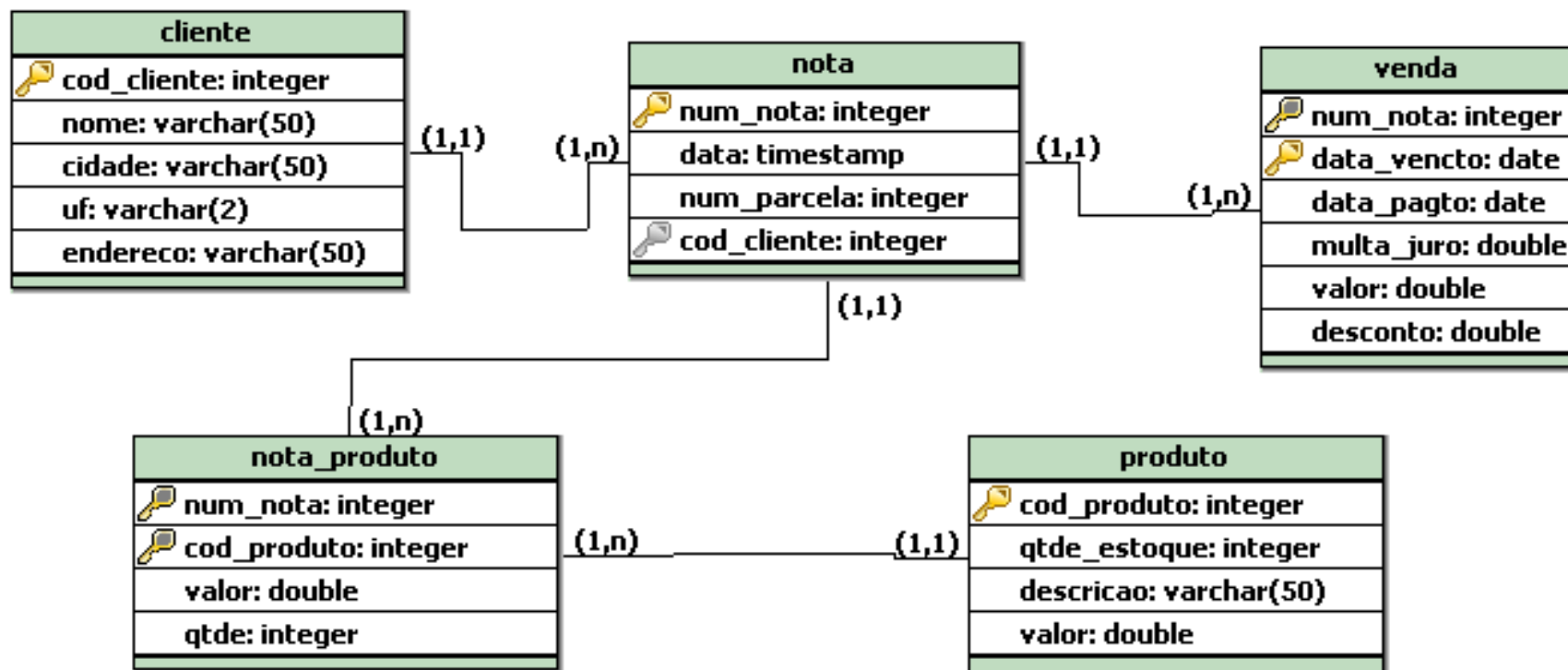
Prof. Alexandre L. Gonçalves
E-mail: a.l.goncalves@ufsc.br

■ Exercício 1

- Implemente um projeto em Java ou outra linguagem de seu interesse considerando os itens abaixo:
 - Crie um schema no banco de dados e duas tabelas com relacionamento 1:N;
 - No projeto realize inserções nestas duas tabelas (se for Java utilize JDBC);
 - Após as inserções execute uma consulta utilizando as duas tabelas e apresente o resultado na tela (se for Java utilize JDBC).

■ Exercício 2

□ Modelo proposto



■ Exercício 2

- Implemente um programa Java ou outra linguagem de seu interesse que faça inserções e atualizações nas tabelas do modelo proposto considerando:
 - A ordem de inserção com base nas restrições de chave estrangeira;
 - Antes de inserir um item de produto vinculado a nota deve-se verificar se o produto possui quantidade em estoque; se sim o estoque deve ser reduzido, caso contrário, o sistema deve informar uma mensagem de estoque insuficiente;

■ Exercício 2

- Após a inserção de todos os itens da nota deve-se gerar um ou mais itens de venda dependendo do número de parcelas;
 - Se parcela igual a 0 gere uma venda à vista ($\text{data_vencto} = \text{data_pagto}$) com o valor total da nota,
 - Se parcela maior ou igual a 1 gere n vendas dividindo o valor total da nota por n com data de pagamento nula e data de vencimento de 30 dias (1 parcela), 60 dias (2 parcelas) e assim por diante;
 - Se for uma venda à vista ou se for um pagamento adiantado poderá existir um desconto que deve ser menor que o valor da prestação (atributo valor);
 - Se for um pagamento depois do vencimento cobrar 2% de multa;

■ Exercício 3

- Depois da carga inicial do banco de dados implemente um conjunto de métodos que realize as seguintes consultas:
 - Selecione o total de vendas por cliente e por mês;
 - Selecione o total de vendas por estado;
 - Selecione o produto mais vendido em um período determinado de tempo;
 - Selecione o total recebido por cliente em um determinado período de tempo;
 - Selecione o total a receber por cliente em um determinado período de tempo;

Obs: a função `month(date)` será necessária para a primeira consulta e a função `coalesce(expr1, expr2,...)` será necessária para a quarta consulta.

```
month(data_vencimento)
coalesce(multa_juro,0)
```