

**Linguagem SQL – Visões**  
**Lista de Exercícios 11**

**Considere o esquema relacional a seguir, do banco de dados da CooperAgri.**

A CooperAgri é uma cooperativa que auxilia pequenos agricultores a distribuir sua produção para restaurantes na região de São Paulo. Todo cooperado da CooperAgri pode distribuir produtos variados para mais de um restaurante. E cada restaurante pode receber produtos de diferentes cooperados. A CooperAgri mantém um banco de dados (BD) simples, com informações sobre os seus agricultores cooperados, os produtos distribuídos e os restaurantes clientes. Além disso, na tabela *Entrega* do BD da CooperAgri é mantido um registro de cada entrega de produto realizada por um de seus cooperados a um restaurante cliente. Cada registro nessa tabela se refere à entrega de uma certa quantidade de um produto numa determinada data.

```
Cidade(CodC: integer, NomeC: string)
Agricultor(CodA: integer, NomeA: string, CodC: integer)
Produto(CodP: integer, NomeP: string, PrecoQuilo: real)
Restaurante(CodR: integer, NomeR: string, CodC: integer)
Entrega(CodA: integer, CodP: integer, CodR: integer,
        DataEntrega: date, QtdeQuilos: integer)
```

O *script* SQL para a criação das tabelas e dos dados para teste encontra-se no e-Discliplinas.

**Exercício 1: Escreva comandos SQL para construir as visões descritas abaixo.**

- a) Uma visão chamada *AgricultorCidade*, com o código e o nome do agricultor e o nome da cidade onde ele está localizado.
- b) Uma visão chamada *ProdutosBaratos*, com todos os dados de produtos cujo preço por quilo é inferior à média dos preços.
- c) Uma visão chamada *ProdutosRU*, com o código, nome e preço dos produtos que já foram entregues pelo menos uma vez para o restaurante RU-USP.
- d) Uma visão chamada *RestauranteApoiadorAgriLocal* com os dados dos restaurantes que são abastecidos apenas por agricultores da sua cidade.
- e) Uma visão chamada *ResumoConsumo* que apresenta, para cada restaurante e cada produto, o nome do restaurante, o nome do produto, o total em quilos recebidos do produto pelo restaurante e o preço total correspondente.

**Exercício 2: Para responder as questões a seguir, considere as visões que você criou no Exercício 1 e as tabelas + restrições do BD criadas por meio do *script* “criacao\_esquema\_exercicio11.sql”.**

- a) Quais das visões do Exercício 1 são não atualizáveis? Justifique sua resposta.

Obs.: Uma visão é não atualizável quando não se pode executar comandos de inserção, alteração e

remoção diretamente sobre ela.

b) Para cada visão que você indicou na resposta do item (a), quando isso for possível, torne a visão atualizável reescrevendo a definição da visão.

c) Em quais das visões atualizáveis que você criou no Exercício 1 e no item (b) deste exercício podem ocorrer modificações com anomalias? Para cada visão apontada, justifique a sua resposta com pelo menos um exemplo de comando que cause uma anomalia.

Obs.: uma modificação com anomalia é um comando INSERT ou UPDATE sobre a visão que é aceito como válido pelo SGBD, mas que causa uma modificação que não é refletida na visão em questão ou que causa o desaparecimento de tuplas na visão.

**Exercício 3: A visão *AgricultorCidade* do item (a) do Exercício 1 é não atualizável. Crie *triggers* que possibilitem a inserção, alteração e remoção em *AgricultorCidade*.**