

Exercício Desafio: Sistema de Gestão Agrícola Integrado

Objetivo: O aluno deverá gerar todo o código SQL para criar e manipular um banco de dados voltado para o controle de uma propriedade agrícola, abrangendo culturas, fornecedores de insumos, equipamentos, funcionários e movimentações de estoque. Este documento contém apenas instruções — não há código pronto. O estudante deve escrever os comandos DDL, DML e de controle conforme pedido.

Contexto: Você é o gerente de operações de uma fazenda e precisa modelar o sistema para gerenciar as culturas plantadas, os insumos utilizados, os fornecedores, os equipamentos disponíveis, a equipe de trabalho e o estoque de insumos e produtos colhidos.

Parte 1 — Preparação

- Crie o banco de dados **fazenda_integrada** e comece a usá-lo.
 - Planeje pelo menos seis tabelas: **fornecedores_insumos**, **insumos**, **culturas**, **equipamentos**, **funcionarios**, **lotes_plantio**, **estoque_geral** e **movimentacao_estoque**. Defina chaves primárias, tipos de dados adequados e relacionamentos entre elas.
-

Parte 2 — Tarefas principais (o aluno deve escrever os comandos SQL)

Tarefa A — Modelagem e criação de tabelas

1. **fornecedores_insumos:**
 - **id** (INT AUTO_INCREMENT PK)
 - **nome** (VARCHAR(100) NOT NULL)
 - **contato** (VARCHAR(100))
 - **telefone** (VARCHAR(20))
2. **insumos:**
 - **id** (INT AUTO_INCREMENT PK)
 - **nome** (VARCHAR(100) NOT NULL)
 - **tipo** (ENUM('fertilizante', 'defensivo', 'semente', 'outros') NOT NULL)
 - **unidade_medida** (VARCHAR(10))
 - **custo_unitario** DECIMAL(10,2) NOT NULL
3. **culturas:**
 - **id** (INT AUTO_INCREMENT PK)
 - **nome_cultura** (VARCHAR(50) NOT NULL UNIQUE)
 - **tempo_colheita_dias** INT

- **condicoes_ideais** (VARCHAR(255))

4. **equipamentos:**

- **id** (INT AUTO_INCREMENT PK)
- **nome** (VARCHAR(100) NOT NULL)
- **tipo** (VARCHAR(50))
- **data_aquisicao** DATE
- **status** (ENUM('ativo', 'manutencao', 'inativo') DEFAULT 'ativo')

5. **funcionarios:**

- **id** (INT AUTO_INCREMENT PK)
- **nome** (VARCHAR(100) NOT NULL)
- **funcao** (VARCHAR(50))
- **salario** DECIMAL(10,2) NOT NULL CHECK(salario > 0)
- **data_contratacao** DATE

6. **lotes_plantio:**

- **id** (INT AUTO_INCREMENT PK)
- **fk_cultura** INT NOT NULL (FK para **culturas**)
- **area_plantada_m2** DECIMAL(10,2) NOT NULL CHECK(area_plantada_m2 > 0)
- **data_plantio** DATE NOT NULL
- **data_colheita_prevista** DATE
- **observacoes** (VARCHAR(255))
- Adicione FK para **culturas**.

7. **estoque_geral:**

- **id** (INT AUTO_INCREMENT PK)
- **fk_item_tipo** ENUM('insumo','produto_colhido') NOT NULL
- **fk_item_id** INT NOT NULL (Explique como modelar estoque genérico para insumos e produtos colhidos. Não crie uma tabela **produtos_colhidos** explicitamente; use a tabela **culturas** como referência para o **fk_item_id** quando **fk_item_tipo** for 'produto_colhido')
- **quantidade** DECIMAL(10,3) NOT NULL DEFAULT 0
- **data_atualizacao** DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP

8. **movimentacao_estoque:**

- **id** (INT AUTO_INCREMENT PK)
- **fk_estoque** INT NOT NULL (FK para **estoque_geral**)
- **tipo_mov** ENUM('entrada','saida','ajuste') NOT NULL
- **quantidade** DECIMAL(10,3) NOT NULL
- **data_mov** DATETIME NOT NULL
- **referencia** (VARCHAR(100)) (Ex: "Compra fornecedor X", "Colheita lote Y", "Aplicação cultura Z")
- Adicione FK para **estoque_geral**.

Tarefa B — Relações e Composições (Receitas de Aplicação)

1. Crie uma tabela **receitas_aplicacao** que descreve quais insumos são usados por cultura: **receitas_aplicacao(id, fk_cultura, fk_insumo, quantidade_por_m2 DECIMAL(10,3), frequencia_dias INT)**.
2. Explique como usar a tabela **receitas_aplicacao** para calcular a necessidade de insumos para um **lotes_plantio** específico.

Tarefa C — Inserções (em ordem correta)

1. Insira pelo menos dois fornecedores de insumos.
2. Insira pelo menos três insumos (ex: semente de milho, fertilizante NPK, defensivo agrícola).
3. Insira pelo menos duas culturas (ex: Milho, Soja).
4. Insira pelo menos dois equipamentos (ex: Trator, Pulverizador).
5. Insira pelo menos dois funcionários.
6. Insira dois lotes de plantio para as culturas criadas.
7. Crie registros na tabela **receitas_aplicacao** para as culturas e insumos (ex: para a cultura "Milho", use "semente de milho" 0.05 kg/m², "fertilizante NPK" 0.01 kg/m²).
8. Atualize o estoque inicial para alguns insumos (ex: semente de milho 1000 kg, fertilizante NPK 500 kg).

Tarefa D — Consultas e relatórios operacionais

1. Consulta que mostra todos os **lotes_plantio** com as culturas correspondentes e a área plantada.
2. Consulta que calcula a necessidade total de cada insumo (nome do insumo, tipo, unidade de medida e quantidade total necessária) para todos os **lotes_plantio** ativos (assuma que 'ativo' significa **data_colheita_prevista** ainda no futuro ou nula), com base nas **receitas_aplicacao**.
3. Consulta que mostra o estoque atual de cada insumo, incluindo seu valor total (quantidade * custo_unitario).
4. Relatório (query) que lista todas as movimentações de estoque do último mês, agrupadas por **tipo_mov** e **referencia**, mostrando a quantidade total movimentada.
5. Consulta que mostra quais equipamentos estão em **manutencao**.

Tarefa E — Alterações e Constraints

1. Adicione uma coluna **observacoes_manutencao** (VARCHAR(255)) na tabela **equipamentos** para registrar detalhes quando o status for 'manutencao'.
2. Adicione **CHECK constraints** para garantir que **quantidade** em **movimentacao_estoque** seja sempre maior que zero; explique as limitações do MySQL para **CHECK** constraints em versões mais antigas e como lidar com isso.
3. Adicione uma **UNIQUE constraint** na tabela **funcionarios** para evitar que dois funcionários tenham o mesmo nome (assuma que nome completo é único para simplificar).

Parte 3 — Entregáveis (o aluno deve enviar)

- Um arquivo SQL (**fazenda_exercicio.sql**) com todos os comandos executados na ordem correta para criar o banco de dados, tabelas, inserir dados e realizar as consultas e alterações.