

Questão de Prova Final - Banco de Dados

Curso: Sistemas de Informação

Disciplina: Banco de Dados

Professor: [Nome do Professor]

Data: 11 de Junho de 2025

Cenário

Uma empresa de e-commerce, a "IESGO Tech", está expandindo suas operações e precisa otimizar seu banco de dados. Atualmente, eles utilizam uma única tabela desnormalizada para registrar todos os pedidos, clientes e produtos. Sua tarefa é ajudar a IESGO Tech a reestruturar seu banco de dados, aplicando conceitos de normalização e utilizando recursos avançados do MySQL para melhorar a eficiência e a integridade dos dados.

Você receberá os scripts `create_denormalized_table.sql` e `insert_denormalized_data.sql` para iniciar o ambiente. Execute-os na ordem para criar o banco de dados `ecommerce_iesgo` e popular a tabela `Pedidos_Desnormalizados`. A partir daí, você deverá desenvolver os scripts SQL conforme as seções abaixo.

Instruções

Parte 1: Normalização (Para o Aluno)

Analise a tabela `Pedidos_Desnormalizados` fornecida. Identifique as dependências funcionais e as anomalias de inserção, atualização e exclusão. Com base nessa análise, normalize o esquema do banco de dados até a 3ª Forma Normal (3FN). Crie um script SQL chamado `normalize_tables.sql` que contenha as instruções `CREATE TABLE` para as novas tabelas normalizadas e as instruções `INSERT INTO` para migrar os dados da tabela `Pedidos_Desnormalizados` para as novas tabelas. Certifique-se de que as chaves primárias e estrangeiras estejam corretamente definidas.

Tabelas esperadas após a normalização (exemplo):

- * `Clientes` (`cliente_id`, `nome_cliente`, `email_cliente`, `telefone_cliente`, `endereco_cliente`, `cidade_cliente`, `estado_cliente`, `cep_cliente`)
- * `Produtos` (`produto_id`, `nome_produto`, `descricao_produto`, `preco_unitario`, `categoria_produto`)

- * Pedidos (pedido_id, cliente_id, data_pedido, status_pedido, metodo_pagamento, data_pagamento)
- * Itens_Pedido (item_id, pedido_id, produto_id, quantidade, preco_unitario_item, total_item)

Parte 2: Joins (joins_normalized.sql)

Utilizando as tabelas normalizadas que você criou, desenvolva um script SQL que demonstre o uso de diferentes tipos de JOINS (INNER, LEFT, RIGHT, CROSS, SELF JOIN) para responder às seguintes perguntas:

1. Liste todos os pedidos com os detalhes do cliente que o realizou.
2. Liste todos os itens de pedido com os detalhes do produto correspondente.
3. Liste todos os clientes e seus pedidos, incluindo clientes que não fizeram pedidos (LEFT JOIN).
4. Liste todos os produtos e os itens de pedido em que aparecem, incluindo produtos que nunca foram pedidos (RIGHT JOIN).
5. Liste clientes que fizeram mais de um pedido (SELF JOIN).
6. Liste todos os pedidos e seus itens, incluindo os detalhes do produto e do cliente.

Parte 3: Índices (indices_normalized.sql)

Crie um script SQL que adicione índices às suas tabelas normalizadas para otimizar as consultas mais frequentes. Considere os seguintes pontos:

1. Crie um índice na coluna email_cliente da tabela Clientes .
2. Crie um índice na coluna nome_produto da tabela Produtos .
3. Adicione índices em chaves estrangeiras (FOREIGN KEY) onde for apropriado para melhorar o desempenho de JOINS.
4. Crie um índice na coluna data_pedido da tabela Pedidos .

Parte 4: Views (views_normalized.sql)

Crie um script SQL que defina as seguintes views nas suas tabelas normalizadas:

1. vw_detalhes_pedidos : Uma view que combine informações completas de pedidos, clientes, produtos e itens de pedido.
2. vw_vendas_por_categoria : Uma view que mostre o total de vendas por categoria de produto (apenas pedidos concluídos).
3. vw_clientes_gastos : Uma view que liste os clientes e o total gasto por cada um em pedidos concluídos.

Parte 5: Funções Internas (`funcoes_internas_normalized.sql`)

Crie um script SQL que demonstre o uso de pelo menos 5 funções internas do MySQL, aplicando-as às suas tabelas normalizadas. Exemplos:

- `COUNT()` , `AVG()` , `SUM()` , `MAX()` , `MIN()`
- `UPPER()` , `LOWER()` , `CONCAT()` , `SUBSTRING()`
- `YEAR()` , `MONTH()` , `DAY()` , `DATEDIFF()` , `DATE_FORMAT()`

Parte 6: Criação de Funções Personalizadas (`funcoes_personalizadas_normalized.sql`)

Crie um script SQL que defina pelo menos duas funções personalizadas (UDFs - User-Defined Functions) no MySQL, utilizando suas tabelas normalizadas. Exemplos:

1. Uma função que calcule o valor total de um pedido, somando todos os itens associados a ele.
2. Uma função que retorne uma descrição mais amigável para a `categoria_produto` (ex: 'Eletrônicos' -> 'Eletrônicos em Geral').

Parte 7: Procedures (`procedures_normalized.sql`)

Crie um script SQL que defina pelo menos três stored procedures no MySQL, utilizando suas tabelas normalizadas. Exemplos:

1. Uma procedure para inserir um novo cliente na tabela `Clientes` .
2. Uma procedure para registrar um novo pedido, incluindo a inserção de múltiplos itens de pedido em uma única chamada (pode usar JSON para os itens).
3. Uma procedure para obter o histórico completo de pedidos de um cliente específico.
4. Uma procedure para atualizar o status de um pedido.

Entrega

Você deverá entregar os seguintes arquivos:

- `create_denormalized_table.sql`
- `insert_denormalized_data.sql`
- `normalize_tables.sql` (Este será o script que você criará na Parte 1)
- `joins_normalized.sql`
- `indices_normalized.sql`
- `views_normalized.sql`

- `funcoes_internas_normalized.sql`
- `funcoes_personalizadas_normalized.sql`
- `procedures_normalized.sql`
- Este documento (`questao_prova_ecommerce.md`)

Certifique-se de que todos os scripts estejam funcionando corretamente e que os resultados das consultas e execuções das procedures sejam claros e demonstrem o entendimento dos conceitos abordados.