Desafio prático para AD.

O desafio deve ser realizado em banco de dados MySql e ter um dump disponibilizado em git de sua escolha. Na entrevista técnica você irá apresentar o seu desafio compartilhando tela, por isso, mantenha o mesmo consigo.

Para a questão 3 a justificativa pode ser realizado em arquivo .txt dentro do mesmo repositório.

Boa sorte!

Cenário: Você faz parte da equipe de desenvolvimento de uma plataforma de e-commerce que está sendo reestruturada. A equipe precisa que você avalie, modele, crie e revise componentes de dados para garantir performance, clareza e aderência a boas práticas.

# 1 - Modelagem de Dados.

Normalização de um Modelo (Modelo conceitual → lógico)

Abaixo está um modelo conceitual simplificado. Modele (em formato lógico e físico, com SQL CREATE TABLE) as entidades a seguir, aplicando normalização adequada até 3FN:

* Cliente (nome, CPF, email, endereço completo, data de nascimento)
* Pedido (data, valor total, status, itens)
* Produto (nome, preço, categoria)
* Item do pedido (quantidade, valor unitário)

Requisitos:

* Relacione corretamente as tabelas.
* Crie um padrão no nomenclatura dos campos
* Inclua chaves primárias e estrangeiras.
* Utilize tipos de dados adequados.

# Parte 2 — SQL Avançado e Procedimentos Armazenados

Crie um procedimento chamado sp\_relatorio\_vendas\_periodo com os seguintes parâmetros:

* @data\_inicio DATE,
* @data\_fim DATE,
* @categoria\_produto VARCHAR(100)

 Este procedimento deve retornar:

* Total de pedidos.
* Soma do valor total dos pedidos.
* Média por pedido.
* Lista de produtos da categoria filtrada e quantidades vendidas no período.

# Parte 3 - Otimização de uma Query

Analise a consulta abaixo e proponha uma nova versão mais performática e clara criando uma procedure com o devido resultado.

SELECT c.nome, [p.id](http://p.id), p.valor\_total

FROM pedidos p, clientes c

WHERE p.id\_cliente = [c.id](http://c.id)

AND p.status = 'concluido'

AND p.valor\_total > 1000

ORDER BY p.data\_criacao DESC;

Justifique sua resposta em poucas palavras.

# Parte 4 - Migrations

Você está atuando em um time onde, por um período, scripts de banco foram aplicados manualmente em diferentes ambientes (homologação, pré-produção, treinamento, produção, etc). Isso resultou em cenários onde um script foi executado em um ambiente e não em outro, gerando falhas em futuras execuções de migrations automatizadas.

Atualmente, buscamos garantir que os scripts sejam reaplicáveis e consistentes entre ambientes, mesmo com esse histórico.

**Enunciado:**

Considere o seguinte script que foi encontrado em um repositório e aplicado em produção, mas **não foi registrado como migration**. Agora, a equipe deseja adicioná-lo ao fluxo automatizado para que, ao ser aplicado novamente, não cause falhas ou inconsistências nos ambientes onde já foi executado.

INSERT INTO recurso\_acao\_processo (

id\_recurso\_acao\_processo,

nome,

ativo,

data\_inclusao,

id\_usuario\_inclusao

) VALUES (

532,

'Vincular boleto',

1,

NAW(),

id\_user\_homologacao

);

**Tarefa**

1. Analise o script acima e liste todos os problemas e riscos potenciais, levando em consideração múltiplos ambientes e reaplicações.
2. Reescreva o script seguindo as boas práticas para que:

* Seja seguro de ser reaplicado.
* Evite conflitos com dados já existentes.
* Garanta consistência entre os ambientes.
* Utilize parâmetros dinâmicos quando necessário.

1. Explique por que suas alterações resolvem os problemas encontrados.