

Prova de SQL

Conceitos avançados

A avaliação consistirá em 10 exercícios sobre JOINS, 10 sobre Stored Procedures, 10 sobre Trigger, 5 sobre VIEWS e 5 sobre UDFs (User Defined Functions).

Começando:

Considere as seguintes tabelas:

Tabela "Clientes":

Colunas: ID_cliente (chave primária), Nome, Endereço

Tabela "Pedidos":

Colunas: ID_pedido (chave primária), ID_cliente (chave estrangeira referenciando a tabela "Clientes"), Data, Valor

Tabela "ItensPedido":

Colunas: ID_item (chave primária), ID_pedido (chave estrangeira referenciando a tabela "Pedidos"), Produto, Quantidade

Faça os exercícios:

1. Liste todos os clientes e seus pedidos correspondentes.
2. Mostre os pedidos e seus respectivos itens.
3. Retorne os clientes que não fizeram nenhum pedido.
4. Liste todos os pedidos feitos em uma determinada data.
5. Mostre os pedidos e os itens correspondentes apenas para os clientes que fizeram pelo menos um pedido.

6. Retorne os clientes e seus pedidos apenas se o valor do pedido for superior a R\$ 100.
7. Liste os clientes que fizeram pedidos de um determinado produto.
8. Mostre os pedidos feitos por clientes que residem em uma determinada cidade.
9. Retorne os clientes que fizeram mais de um pedido.
10. Liste os clientes que fizeram um pedido de um produto específico mais de uma vez.

Exercícios Procedures:

1. Crie uma stored procedure que liste todos os clientes da tabela "Clientes".
2. Crie uma stored procedure que receba o ID de um cliente como parâmetro e retorne todos os pedidos feitos por esse cliente.
3. Crie uma stored procedure que receba o ID de um pedido como parâmetro e retorne todos os itens desse pedido.
4. Crie uma stored procedure que retorne a quantidade total de pedidos na tabela "Pedidos".
5. Crie uma stored procedure que receba o nome de um produto como parâmetro e retorne todos os pedidos que contêm esse produto.
6. Crie uma stored procedure que calcule e retorne o valor total de todos os pedidos feitos por um cliente, recebendo o ID do cliente como parâmetro.
7. Crie uma stored procedure que insira um novo cliente na tabela "Clientes", recebendo os valores dos campos como parâmetros.
8. Crie uma stored procedure que atualize o endereço de um cliente na tabela "Clientes", recebendo o ID do cliente e o novo endereço como parâmetros.
9. Crie uma stored procedure que delete um cliente da tabela "Clientes", recebendo o ID do cliente como parâmetro.
10. Crie uma stored procedure que receba o ID de um cliente como parâmetro e retorne a quantidade de pedidos feitos por esse cliente.

Exercícios Triggers:

1. Crie uma trigger que, ao inserir um novo registro na tabela "Pedidos", atualize a coluna "Data_Último_Pedido" na tabela "Clientes" com a data atual.
2. Crie uma trigger que, ao excluir um registro da tabela "Clientes", exclua automaticamente todos os pedidos desse cliente na tabela "Pedidos".
3. Crie uma trigger que, ao atualizar a coluna "Quantidade" na tabela "ItensPedido", atualize o campo "Valor_Total" na tabela "Pedidos" multiplicando a quantidade pelo preço do item.
4. Crie uma trigger que, ao inserir um novo registro na tabela "ItensPedido", atualize automaticamente o campo "Quantidade_Total" na tabela "Pedidos" somando a quantidade do novo item.
5. Crie uma trigger que, ao atualizar a coluna "Status" na tabela "Pedidos", insira um novo registro na tabela "Log_Pedidos" com o ID do pedido, a nova data e o novo status.
6. Crie uma trigger que, ao inserir um novo registro na tabela "Clientes", verifique se o cliente já possui mais de 5 pedidos na tabela "Pedidos" e, se sim, insira o cliente na tabela "Clientes_VIP".
7. Crie uma trigger que, ao excluir um registro da tabela "Pedidos", atualize automaticamente o campo "Data_Último_Pedido" na tabela "Clientes" com a data do novo último pedido.
8. Crie uma trigger que, ao atualizar a coluna "Valor" na tabela "Pedidos", verifique se o novo valor é maior que o valor anterior e, se sim, insira um registro na tabela "Log_Aumento_Valor" com o ID do pedido e a diferença de valores.
9. Crie uma trigger que, ao inserir um novo registro na tabela "ItensPedido", verifique se a quantidade total dos itens desse pedido ultrapassa 100 e, se sim, atualize a coluna "Status" na tabela "Pedidos" para "Em Análise".
10. Crie uma trigger que, ao excluir um registro da tabela "Clientes", verifique se o cliente possui pedidos pendentes na tabela "Pedidos" e, se sim, cancele esses pedidos atualizando o campo "Status" para "Cancelado".

Exercícios VIEWS:

1. Crie uma view chamada "Pedidos_Completos" que retorne todos os pedidos juntamente com as informações do cliente relacionado a cada pedido.
2. Crie uma view chamada "ItensMaisVendidos" que retorne os itens de pedido mais vendidos, juntamente com a quantidade total vendida de cada item.
3. Crie uma view chamada "ClientesVIP" que retorne os clientes que fizeram mais de 10 pedidos e possuem um valor total de pedidos acima de R\$ 1000.
4. Crie uma view chamada "PedidosRecentes" que retorne os pedidos feitos nos últimos 30 dias.
5. Crie uma view chamada "TotalPedidosPorCliente" que retorne o número total de pedidos feitos por cada cliente.

Exercícios UDFs:

1. Crie uma UDF chamada "CalcularFrete" que receba a distância em quilômetros como parâmetro e retorne o valor do frete com base em uma determinada fórmula de cálculo.
2. Crie uma UDF chamada "ConverterParaMaiúsculas" que receba uma string como parâmetro e retorne a mesma string convertida para maiúsculas.
3. Crie uma UDF chamada "CalcularIdade" que receba uma data de nascimento como parâmetro e retorne a idade atual com base na data atual.
4. Crie uma UDF chamada "CalcularDesconto" que receba o valor total da compra e o código do cupom como parâmetros e retorne o valor do desconto aplicado.
5. Crie uma UDF chamada "CalcularTotalVendas" que receba um intervalo de datas como parâmetro e retorne o valor total de vendas no período especificado.

Boa Sorte !!!