**Curitiba, 22 de Novembro de 2024**

**Thiago Ferreira Medici,**

**Trabalho de Banco de Dados**

**Criação de Tabelas (CREATE TABLE)**

**Tabela: Categorias**



O código acima trata da criação da tabela "categorias", que armazena informações sobre as categorias dos produtos que serão apresentados posteriormente. A tabela inclui uma chave primária "categoria\_id" com incremento automático(auto\_increment) e outros campos para nome(VARCHAR) e descrição(TEXT).

**Tabela: Fornecedores**



Essa tabela fornece informações sobre os fornecedores dos produtos, como o nome do fornecedor, seu contato(email), seu endereço e cidade onde ficam. Possui uma chave primária com auto\_increment e o nome é "NOT NULL", que impossibilita do campo ficar vazio.

**Tabela: Produtos**



Esse trecho do código cria a tabela dos produtos que serão utilizados, com uma primary key incrementada automaticamente e duas chaves estrangeiras que referenciam as tabelas categorias e fornecedores, respectivamente.



Essa tabela trata dos clientes que serão atendidos, aqueles que comprarão os produtos, contendo uma chave primaria e 6 variáveis de tipo varchar, contendo as informações desses clientes.

**Tabela: Funcionários**



A tabela "Funcionarios" guarda as informações dos funcionarios responsaveis pelas vendas dos produtos. Possui uma chave primária de incrementação automática e mais 6 variáveis contendo informações dos funcionarios.

**Tabela: Pedidos**



Essa tabela trata dos pedidos realizados a respeito dos produtos. Contem uma chave primária e duas estrangeiras, que referenciam a tabela clientes e a tabela funcionarios, respectivamente.

**Tabela: Itens\_Pedido**



A tabela acima armazena os produtos incluídos em cada pedido, com detalhes como quantidade e preço unitário. Possui uma chave primária e duas chaves estrangeiras referenciando as tabelas pedidos e produtos.

**Tabela: Pagamentos**



A tabela, como o nome sugere, trata dos pagamentos e seus métodos, bem como seu valor e data. Possui uma chave primaria e uma estrangeira que referencia a tabela pedidos.

**Inserção de Dados (INSERT)**

**Inserção em Categorias**



Este comando insere os nomes de cada categoria na tabela "categorias", preenchendo o campo "nome" da tabela.

**Inserção em Fornecedores**



Esse Insert preenche os campos "nome", "contato", "endereço" e "cidade" com os valores acima, para que mais tarde sejam consultados.

**Inserção em Produtos**



Esse insert preenche os campos nome, preco, estoque, categoria\_id (para relacionar com a tabela categorias) e fornecedor\_id(para relacionar com a tabela fornecedor), relacionando as tabelas.

**Inserção em Clientes**



Esse trecho do código insere os valores em laranja nos campos nome, email, telefone, cpf, endereço e cidade, preenchendo as informações da tabela e dos clientes.

**Inserção em Funcionarios**



Esse insert preenche os campos nome, cargo, salario, data\_contratação, email e telefone dos funcionarios na tabela, inserindo as informações dos funcionarios.

**Inserção em Pedidos**



Esse trecho preenche as informações dos pedidos, quanto a sua data(data\_pedido), o cliente que comprará/comprou (cliente\_id), o funcionario responsavel(funcionario\_id) e o status desse pedido.

**Inserção em Itens\_Pedido**



Nesse insert, há o registro dos produtos incluídos em diferentes pedidos, especificando as quantidades e os preços de cada produto requerido.

**Inserção em Pagamentos**



No trecho, há o preenchimento da tabela pagamentos, nos campos pedido\_id, valor, metodo\_pagamento e data\_pagamento, preenchendo qual pedido se refere, seu valor, qual o metodo de pagamento e quando o pagamento foi feito.

**Atualizações e Modificações (UPDATE)**

**Aumentando Preços de Produtos**



Este comando aplica um aumento de 10% (preço anterior \* 1.1) no preço de todos os produtos da categoria "Eletrônicos". O subquery identifica a "categoria\_id" correspondente e o campo where filtra para apenas produtos da categoria requisitada.

**Desconto em Produtos com Estoque Alto**



Este trecho do código atualiza o preço para preço\*0.95, ou seja, reduz em 5% o valor onde o estoque é menor do que 20 unidades.

**Promoção em Fornecedores Selecionados**



Esse update aplica 15% de desconto ( preço \* 0.85) filtrando por id de fornecedor e onde o nome é LifeTech.

**Atualização de Dados de Clientes**



O trecho do código altera o telefone e o endereço do cliente cujo id é 5, mudando os valores desses campos.

**Promoção de Funcionários**



Esse update atualiza o cargo do funcionario de id 3 para "Gerente" e aumenta seu salario em 20%(salario \* 1.20).

**Atualização do Status de Pedidos**



Esse trecho atualiza o status dos pedidos que tenham um pagamento associado para "Concluido", quando o pagamento relacionado foi concluido.

**Reajuste de Preços Pós-Promoção**



Aqui, os preços atualizados precisam voltar para o normal.

Primeiro, se adiciona uma nova coluna preco\_original. Depois, se define os valores originais para os produtos(os definidos lá no insert). Após isso, é redefinido o preço atual para os valores originais.

Essa nova coluna permite armazenar preços fixos e o CASE define os preços para produtos específicos.

**Atualização de Estoque após Vendas**



Nesse trecho, é preciso se reduzir a quantia em estoque após um pedido.

O join conecta as tabelas produtos, itens\_pedido e pedidos. Depois, o where filtra apenas pedidos com status 'Concluído' e é feita a atualização.

**Correção de Informações de Fornecedores**



Aqui é preciso atualizar o endereco e contato do fornecedor cujo id é 2, para isso, é usado um where para filtragem e o update/set para atualizar.

**Atualização de Quantidades em Itens de Pedidos**



Aqui, é atualizada a quantidade do pedido cujo id é 12 para 5, para isso é usado o where para filtrar.

**Consultas e Relacionamentos (JOIN)**

**Produtos por Categoria e Fornecedor**



Esse código lista produtos com estoque maior que 10, incluindo seus nomes, suas categorias, o nome do fornecedor e a quantidade no estoque.O JOIN categorias conecta produtos com categorias pela chave categoria\_id. O JOIN fornecedores conecta produtos com fornecedores pela chave fornecedor\_id. O WHERE p.estoque > 10 filtra produtos com estoque acima de 10.

**Pedidos de Clientes**



Esse trecho exibe informações sobre pedidos feitos por clientes nos últimos 30 dias, com o nome do cliente, a data do pedido, o funcionário responsável e o status do pedido. O JOIN clientes conecta pedidos com clientes pela chave cliente\_id. O JOIN funcionarios relaciona o pedido ao funcionário responsável (funcionario\_id). Nesse código, apenas pedidos dos últimos 30 dias são incluídos.

**Total de Vendas por Produto**



Esse join mostra a quantidade total vendida de cada produto que teve mais de 1 unidade vendida. O JOIN produtos relaciona itens do pedido com seus produtos correspondentes usando USING(produto\_id). O GROUP BY p.nome agrupa os dados por produto. Já o HAVING total\_vendido > 1 filtra produtos com mais de uma unidade vendida.

**Pagamentos por Pedido**



O código da print exibe informações sobre pagamentos realizados nos últimos 60 dias. O WHERE p.data\_pagamento >= CURDATE() - INTERVAL 60 DAY faz a filtragem dos pagamentos com base na data.

**Pedidos Pendentes**



Aqui, há a listagem pedidos pendentes ou sem pagamento, com o valor total acima de 100. O LEFT JOIN itens\_pedido inclui itens associados aos pedidos. O LEFT JOIN pagamentos verifica se o pedido possui um pagamento registrado. Já o WHERE faz a filtragem pedidos pendentes. O HAVING, por fim, exclui pedidos cujo valor total seja menor ou igual a 100.

**Relatório de Estoque**



Acima, lista-se os produtos com estoque inferior a 5, incluindo informações sobre fornecedores. O INNER JOIN relaciona produtos e fornecedores. O WHERE p.estoque < 5 filtra produtos com estoque abaixo de 5. O ORDER BY p.nome ordena os resultados alfabeticamente.

**Total de Vendas por Cliente**



O bloco acima lista clientes que gastaram mais de 500 em compras. O INNER JOIN conecta clientes com pedidos e itens dos pedidos. O GROUP BY agrupa dados por cliente. O HAVING filtra clientes com gastos superiores a 500.

**Funcionários com Vendas Acima da Média**



Aqui, se lista funcionários cujas vendas totais superam a média de vendas. A Subquery calcula a média total de vendas. O HAVING compara vendas de cada funcionário com a média.

**Análise de Vendas Mensais**



Essa consulta exibe o total de vendas por mês e ano, considerando apenas os meses com vendas totais superiores a 1000. O INNER JOIN itens\_pedido conecta as tabelas pedidos e itens\_pedido para acessar os itens de cada pedido. O trecho "YEAR(p.data\_pedido), MONTH(p.data\_pedido)" extrai o ano e o mês da data do pedido. O GROUP BY agrupa os pedidos por ano e mês. O HAVING filtra os resultados, incluindo apenas os meses com vendas totais superiores a 1000. Por fim, O ORDER BY ordena os resultados por ano e mês.

**Produtos com Baixa Rotatividade**



O trecho exibe os produtos que não foram vendidos nos últimos 90 dias. O INNER JOIN categorias conecta a tabela de produtos com categorias para incluir o nome da categoria de cada produto. A Subquery do trecho busca todos os produto\_id de pedidos realizados nos últimos 90 dias. O termo NOT IN exclui da lista os produtos que apareceram na subconsulta (aqueles vendidos recentemente). E o ORDER BY ordena os produtos pelo nome.

**Relatório de Funcionários por Cargo**



O código exibe o número de funcionários por cargo, mas somente para cargos com mais de 2 funcionários. O GROUP BY f.cargo agrupa os funcionários por cargo. O COUNT(f.funcionarios\_id) conta o número de funcionários em cada cargo. Já o HAVING filtra para mostrar apenas os cargos com mais de 2 funcionários.

**Resumo de Pedidos por Cliente e Funcionário**



Aqui, se exibe clientes, o total de pedidos feitos por eles e quantos desses pedidos foram atendidos por um funcionário (com status 'Concluído'). O JOIN pedidos relaciona clientes com seus pedidos. O JOIN funcionarios relaciona pedidos com os funcionários responsáveis. O COUNT(CASE WHEN ...) conta quantos pedidos foram concluídos (status 'Concluído') para cada cliente e funcionário. O HAVING filtra os resultados para mostrar apenas clientes com mais de 3 pedidos.

**Histórico de Compras de um Cliente**



Acima, exibe-se os pedidos feitos por um cliente específico (cliente\_id = 4) no último ano, com os produtos, quantidades, preços e data do pedido. O JOIN pedidos relaciona os itens do pedido com as informações do pedido. O JOIN produtos relaciona os itens com os produtos comprados. O JOIN clientes relaciona os pedidos com o cliente que fez a compra. O WHERE filtra para o cliente com cliente\_id = 4 e pedidos feitos no último ano. O ORDER BY ordena os resultados pela data do pedido, do mais recente para o mais antigo.

**Produto com Maior Receita**



Esse bloco de código exibe o produto que gerou a maior receita total a partir das vendas. O INNER JOIN itens\_pedido relaciona os produtos com os itens dos pedidos para calcular a receita. O SUM(itp.preco\_unitario \* itp.quantidade) calcula a receita total de cada produto, multiplicando o preço unitário pela quantidade vendida. O GROUP BY p.produto\_id, p.nome agrupa os dados por produto. O ORDER BY receita\_total DESC ordena os resultados pela receita total, da maior para a menor. O LIMIT 1 retorna apenas o produto com a maior receita.

**Alter Table**



**Alteração em funcionários:**

No Alter table funcionarios se adiciona a coluna endereço de tipo varchar à tabela funcionarios.

**Alteração em fornecedores:**

No Alter table fornecedores se adiciona a coluna telefone de tipo varchar à tabela fornecedores.

**Alteração em clientes:**

No Alter table clientes se adiciona a coluna idade de tipo int à tabela clientes.