

Nome: Lucília dos Passos Rosa

Vaga: Analista de Sustentação

Data: 02/03/2022

A empresa trabalha com campanhas de fidelidade, conectamos campanhas de incentivo com parceiro, onde temos os clientes que arrecadam pontos nas campanhas e com esses pontos fazem resgate de produtos nas lojas. Usando como referência as tabelas no arquivo Excel encaminhado junto, encontre o melhor caminho através de scripts SQL para realizar as tarefas sugeridas.

1) Perdemos todos os registros da nossa tabela de loja (TB_LOJA), precisamos realizar uma nova carga de informações, seguindo as regras de negócio a seguir:

A – Só devemos vincular campanhas que estejam ativas.

B – Clientes não podem ter vinculado parceiros que sejam do mesmo segmento que ele, porém, concorrentes (Ex. Cliente Samsung não pode vincular parceiro Panasonic, ambos são do segmento de eletrônicos, porém concorrentes).

C – Campanhas do segmento de construção, não podem vincular parceiros dos segmentos de Alimentos e nem Bebidas.

D – Todas as lojas devem ser vinculadas como ativas e visíveis, exceto as lojas do segmento de Bebidas, que devem ser vinculadas ativas e não visíveis.

E – Na tabela de loja o campo ID_LOJA é um sequencial auto incremental.

F – Os campos ID_PARCEIRO e ID_CAMPANHA são chaves estrangeiras.

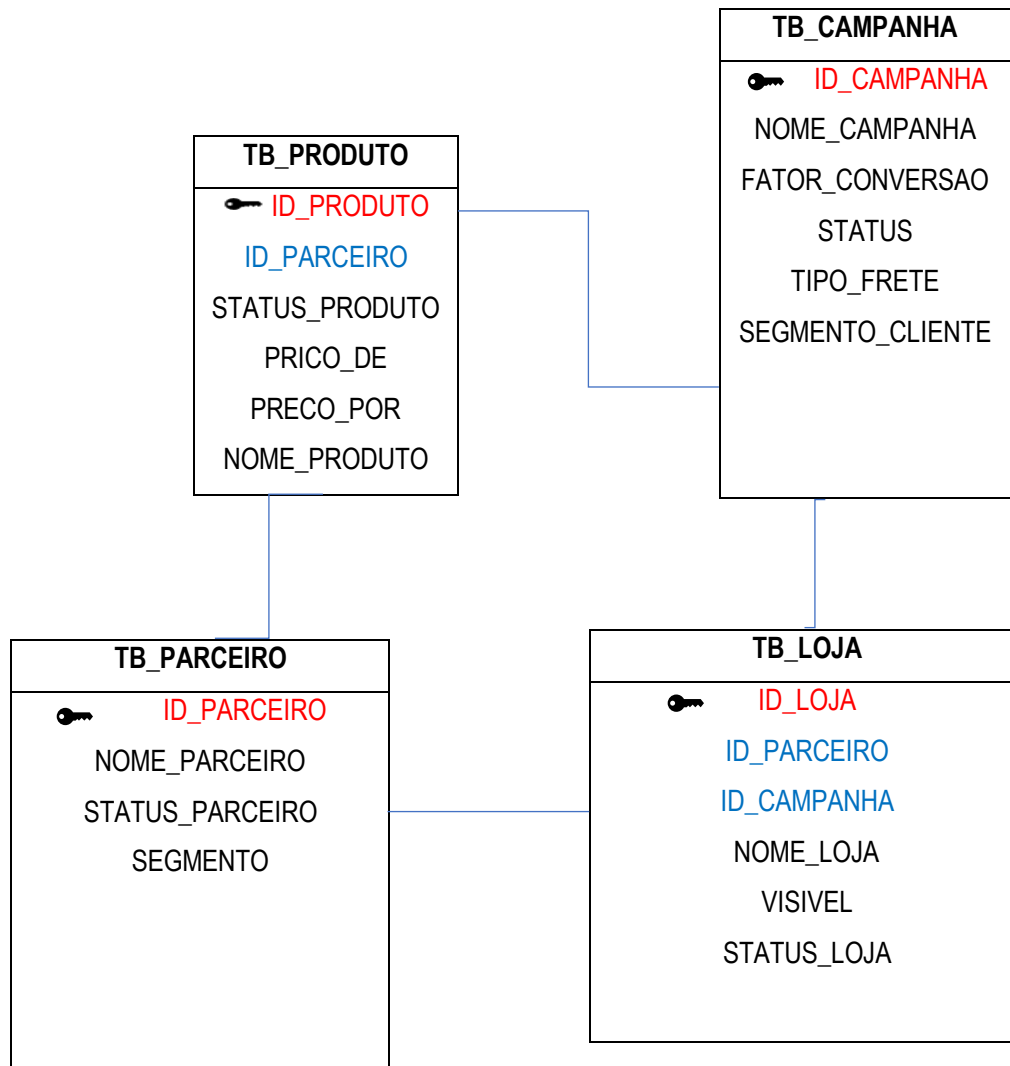
G – O nome da loja por padrão leva sempre o nome do parceiro.

H – O campo visível é booleano.

I – O status da loja segue o mesmo padrão de status das outras tabelas.

```
SELECT TOP (1000) [ID_LOJA]
, [ID_PARCEIRO]
, [ID_CAMPANHA]
, [NOME_LOJA]
, [VIVISEL]
, [STATUS_LOJA]
FROM [Vertem].[dbo].[TB_LOJA]

ALTER TABLE [TB_LOJA] (
    VIVISEL BIT,
    NOME_LOJA NVARCHAR(30),
    STATUS_LOJA NVARCHAR(30),
    ID_LOJA INT IDENTITY (1,1) PRIMARY KEY,
    ID_PARCEIRO INT FOREIGN KEY REFERENCES TB_PARCEIRO (ID_PARCEIRO),
    ID_CAMPANHA INT FOREIGN KEY REFERENCES TB_CAMPANHA (ID_CAMPANHA), );
```



Vermelho = Chave Primária

Azul = Chave estrangeira

2) Após termos todas as lojas vinculadas, precisamos de uma consulta que nos retorne todos os produtos disponíveis para as campanhas dos segmentos de construção, beleza e variedades, e que seu preço em pontos fique entre 100 pontos e 10000 pontos. Obs: o Preço em pontos é calculado pelo preço em reais na tabela de produtos multiplicado pelo fator conversão da tabela de campanha.

3) Gerar uma consulta que tragam todos os produtos que tiveram uma queda de preço, independente de parceiro ou campanha, ordenado de forma decrescente pela diferença entre os preços.

```
SELECT *
FROM [Vertem].[dbo].[TB_PRODUTO]
WHERE PRICO_DE > PRECO_POR
ORDER BY PRECO_POR DESC;
```

4) Sem ter as tabelas, apenas analisando a query abaixo, me descreva o que é esperado do retorno dessa consulta:

```
SELECT COUNT(DISTINCT A.ID_PRODUTO),  
       A.ID_PARCEIRO  
FROM TB_PRODUTO           A WITH(NOLOCK)  
INNER JOIN TB_PARCEIRO    B WITH(NOLOCK) ON A.ID_PARCEIRO =  
B.ID_PARCEIRO  
LEFT JOIN TB_CATEGORIA_PRODUTO C WITH(NOLOCK) ON A.ID_PRODUTO =  
C.ID_PRODUTO  
WHERE A.STATUS_PRODUTO = 'ATIVO'  
      AND A.PRECO_DE <> 0  
      AND C.ID_CATEGORIA_PRODUTO is null  
GROUP BY A.ID_PRODUTO
```

R = A consulta acima irá trazer como resultado a quantidade (em que os códigos sejam diferentes) de códigos dos parceiros que sejam comuns entre as tabelas, tabela TB_PRODUTO e TB_PARCEIRO.

Irá trazer todos os códigos dos parceiros (da tabela TB_PARCEIROS) e os códigos em comum com a outra tabela (TB_CATEGORIA_PRODUTO).

Onde o status dos produtos estejam como ativos e que o preço anterior (PRECO_DE) seja diferente de 0 e que o código da categoria do produto não seja nulo.

5) Conforme item anterior, descrever o retorno esperado dessa consulta:

```
SELECT C.ID_TRANSACAO  
FROM TB_LOTE_CREDITO           A WITH(NOLOCK)  
INNER JOIN TB_LOTE_CREDITO_ITEM B WITH(NOLOCK) ON  
A.ID_LOTE_CREDITO = B.ID_LOTE_CREDITO  
INNER JOIN TB_LOTE_CREDITO_TRANSACAO C WITH(NOLOCK) ON  
B.ID_LOTE_CREDITO_ITEM = C.ID_LOTE_CREDITO_ITEM  
WHERE B.TIPO_TRANSACAO = 'DEBITO'  
      AND A.LOTE_CREDITO_STATUS = 'PROCESSADO'  
      AND A.PONTOS > 0  
      AND CONVERT(DATE,A.FIM_PROCESSAMENTO,103) BETWEEN GETDATE() - 2 AND  
GETDATE() - 1
```

R = A consulta acima irá trazer todos os códigos de transação que sejam comuns entre duas tabelas (TB_LOTE_CREDITO e TB_LOTE_CREDITO_ITEM) onde o tipo de transação precisa ser debito e o status do lote credito precisa estar em processado e os pontos maiores que 0 nos dois dias antes da consulta.