Jump.rnimbuscom.b

Curso SQL07 Laboratório – Módulo 01

Setup - Preparando o ambiente

01.a . Abra o script "Lab01 – Aluno.sql" e execute as instruções de setup, entre as marcações 0.1 e 0.2. Para verificar se tudo foi criado corretamente, você não deve ter recebido nenhum erro e as consultas abaixo devem retornar seus respectivos registros.

```
SELECT * FROM DBO.Customers -- 20.000 registros
SELECT * FROM DBO.Employees -- 500 registros
SELECT * FROM DBO.Orders -- 1.000.000 registros
SELECT * FROM DBO.Shippers -- 5 registros
```

Questão 01 - Analisando planos de execução

01.a . Utilizando o banco de dados SQL07, analise o plano de execução da consulta abaixo.

```
SELECT * FROM dbo.Orders
```

- O que o SQL Server está fazendo para retornar o resultado dessa consulta?
- Quantas páginas de dados foram lidas para retornar o resultado?

01.b . Utilizando o banco de dados SQL07, analise o plano de execução das consultas abaixo.

```
SELECT * FROM dbo.Orders ORDER BY orderdate
go
SELECT * FROM dbo.Orders ORDER BY orderid
go
```

- Qual a diferença entre os planos de execução?
- Você saberia dizer o motivo pelo qual o SQL Server gerou um plano diferente para a consulta ordenada pelo campo OrderID?
 - Se você executar o comando EXEC sp_help 'dbo.Orders' verá que a chave primária é definida no campo orderid.
 - Veremos índices no próximo módulo, onde será mais simples entender a diferença.

01.c . Utilizando o banco de dados SQL07, analise o plano de execução da consulta abaixo.



```
SELECT * FROM dbo.Orders ORDER BY orderdate

SELECT TOP 100 *

FROM dbo.Orders AS O

INNER JOIN dbo.Customers AS C

ON O.custid = C.custid

ORDER BY orderdate DESC

Go
```

- O que o SQL Server está fazendo para retornar o resultado dessa consulta?
- Qual é a implementação que o SQL Server está utilizando para realizar o join entre as tabelas de pedidos (Orders) e clientes (Customers)?

Questão 02 - Cursores vs. Set-Based

02.a Com base nas tabelas criadas durante o setup do laboratório, utilize as tabelas de *Orders* (Pedidos) e *Shippers* (empresas de carregamento) para retornar os *shippers* que costumavam estar ativos, mas não possuem nenhuma atividade após 1 de Janeiro de 2001. Você não deve se preocupar com *shippers* que nunca trabalharam com um pedido.

Para ajudar na pesquisa e criar cenários para o exercício, execute as instruções entre os itens 2.1 e 2.2 do script.

O script mostra uma maneira de resolver o problema acima utilizando uma abordagem cursor-based. Tente resolver o problema acima utilizando uma abordagem set-based e compare os desempenhos. Note que existe mais de uma maneira de escrever a mesma consulta, então fique à vontade para procurar maneiras diferentes.

Questão 03 - Otimizando uma consulta

03.a Abra o script "Lab01 – Aluno.sql" e execute as instruções delimitadas pelos itens 3.1 e 3.2 (destacados nos comentários). Esse script irá criar duas tabelas, unidas por um relacionamento fraco (sem chave estrangeira) e um índice na coluna "Chave" para otimizar a consulta a seguir (item 3.3).

```
SELECT A.Chave, A.Nome, B.Descricao
FROM dbo.TabelaA AS A
INNER JOIN dbo.TabelaB AS B
ON A.Chave = B.Chave
```

Analise o plano de execução da consulta acima e entenda o que o SQL Server está fazendo para retornar o resultado. Como base na sua análise e o que vimos sobre o SQL Server neste módulo, como podemos alterar a consulta ou a modelagem para uma resposta mais eficiente?



Dica: o *merge Join* é um tipo de join utilizado pelo SQL Server que parte da premissa de que as duas entradas de dados estarão com a mesma ordenação.

Questão 04 - Cursor

Usando o banco de dados AdventureWorks2008, selecionar para cada cliente os dois produtos que mais geraram entrada de dinheiro, isto é, aqueles em que o somatório das vendas (LineTotal) foi maior. Por exemplo: para o cliente com id 29704 deverá ser retornado os produtos 793 e 973.

Esse tipo de consulta pode ajudar a empresa a determinar quais produtos irão atrair mais clientes e poderão trazer um bom retorno em um marketing direto.

Esta questão deve ser resolvida através da utilização de cursores.

Para ajudar na resolução do problema, crie uma tabela temporária com o nome #TabelaProdutos que possui dois campos inteiros não nulos, chamados IDCliente e IDProduto.

Por fim, retorne os 10 produtos com maior ocorrência, consultando a #TabelaProdutos e apresentando o resultado de acordo com o exemplo abaixo:

IDProduto	Ocorrencia
707	1341
711	1266
708	1265
921	981
870	966
878	868
930	856
929	811
873	807
931	727

Tabelas envolvidas: Sales.SalesOrderHeader e Sales.SalesOrderDetails.

Questão 05 - Set-Based

Usando o banco de dados AdventureWorks2008, reescreva a questão 04 sem utilizar cursores ou tabelas temporárias. Compare o desempenho de ambas respostas.

