

Rumo a certificação 70-761

Segundo encontro – Gerenciando dados com Transact-SQL

Lindos e lindas, bom dia... boa tarde... boa noite!!!

Esse não é o formato definitivo, mas como meu fone não chegou para fazer o áudio e eu não quero repetir o vexame da semana passada... Vamos começar no texto tradicional mesmo...

Vamos começar a estudar o conteúdo do livro “Querying Data with Transact-SQL” de Itzik Ben-Gan. Este livro é da Microsoft Press tem 325 páginas, e é o livro indicado para a prova 70-761.

A leitura é tranquila, ele aborda bem todos os assuntos que precisaremos para o exame. Não se assustem com o começo. Nada é engessado ou absurdamente complicado (quando for eu aviso e faremos na prática, testaremos, nos encontraremos... enfim daremos um jeito até ficar fácil ☺).

Esta semana a Amanda postou uma foto da prova dela de banco de dados. Combinei com ela de analisarmos a prova juntas no domingo à noite... Quem quiser fazer parte, é só me adicionar no Skype (DaniMonteiroDBA).

Vamos começar com o Capítulo 1! Lembrem que estamos juntas (os) na jornada!

1. Transact-SQL é uma versão (dialeto) da linguagem SQL;
2. A linguagem SQL permite gerenciar e alterar os dados em um banco de dados;
3. T-SQL é a adaptação da linguagem SQL, feita pela Microsoft para os bancos de dados SQL Server e Azure SQL Database;
4. A linguagem SQL é um padrão ISO e ANSI, tem diversas versões e é usada na maioria dos bancos de dados relacionais;
 - a. A linguagem foi evoluindo e ganhando novas versões:
 - i. SQL-86;
 - ii. SQL-89;
 - iii. SQL-92;
 - iv. SQL:1999;
 - v. SQL:2003;
 - vi. SQL:2006;
 - vii. SQL:2008;
 - viii. SQL:2011;
5. O T-SQL não é exatamente igual ao SQL;
6. Usar o padrão SQL deixa seu código mais portátil; e é uma boa prática;
 - a. Quando uso então o T-SQL? Sempre que precisamos de um recurso que não existe no padrão ;-)
7. O padrão SQL está baseado no modelo relacional, que é um modelo matemático para gerenciar e manipular os dados;

8. O modelo relacional foi desenvolvido por Edgar F. Codd em 1969, e teve colaborações de Chris Datta, Hugh Darwen e outros
 - a. Dica... os livros de Banco de Dados destes autores, costumam ser os melhores!
9. O nome Modelo Relacional não está ligado aos relacionamentos entre as tabelas! (Pasmem!) Está ligado ao conceito matemático de relação! E uma relação é representada por uma tabela!!!
10. Uma relação contém uma “cabeça” e um “corpo” na cabeça temos um conjunto de atributos que possuem nome e tipo, e no corpo temos as tuplas (que são representadas pelas linhas).
11. Observação by Dani:
 - a. Na verdade usamos partes do modelo relacional! Me arrisco a dizer que nunca usaremos 100% do modelo proposto por Codd, mas é sempre muito bom entender! Faz com que o trabalho com BD fique mais fácil!
 - b. Para ver as diferenças entre o SQL e o modelo Relacional, o livro da MS cita o livro: *SQL and Relational Theory*, 3rd Edition do C.J Date
12. T-SQL tem um lado lógico que é a interpretação conceitual de uma consulta que retorna um resultado e tem o lado físico que é o processamento da query pela engine do banco de dados;
13. O processamento da query pela engine deve retornar o resultado definido pelo processamento lógico;
14. Para alcançar este objetivo existe a otimização, que pode reorganizar os passos para execução da query, remover passos... enfim é a responsável por fazer a execução da query da melhor maneira possível para retornar o resultado esperado.
 - a. A ideia é alcançar um objetivo, usando a menor quantidade de recursos.
15. SQL é uma linguagem declarativa, ou seja, vc define O QUE vc quer, e quem define o como este objetivo será alcançado é o otimizador;
16. Quando falamos em SQL, logo pensamos no comando SELECT. SELECT lista de atributos FROM nome da tabela. Selecione algo de algum lugar! Simples assim 😊
17. Pessoal deste tópico para frente, temos alguns dos tópicos mais importantes para a certificação e para escrever boas queries!
18. Temos 6 cláusulas principais:
 - a. SELECT
 - b. FROM
 - c. WHERE
 - d. GROUP BY
 - e. HAVING
 - f. ORDER BY
19. Embora tenhamos 6 cláusulas que aparecem na ordem acima quando escrevemos a consulta, o processamento lógico da query acontece de forma diferente! (Isso cai na prova!)
 - a. FROM
 - b. WHERE
 - c. GROUP BY
 - d. HAVING
 - e. SELECT
 - f. ORDER BY
20. Um ponto que costuma cair nas provas...

- a. ALIAS é um apelido que damos para uma coluna. Ou seja temos a coluna CD_ADM_MSG e queremos que no retorno da query, o nome exibido seja “Código Administrativo Mensagem”. Podemos usar o ALIAS que nós criamos na cláusula SELECT, na cláusula WHERE? A resposta é NÃO!!! Pq na execução da query, a cláusula WHERE é executada antes da cláusula SELECT. Entenderam? (em caso de dúvidas vemos este caso no domingo!)

21. FROM

- a. É a primeira cláusula “avaliada”
- b. É uma cláusula obrigatória;
- c. Onde indicamos as tabelas que farão parte da consulta
 - i. Vou adiantar uma coisa... Na cláusula FROM indicamos as relações que fazem parte de uma consulta... E estas podem ser provenientes de outras consultas, de uma view, de uma tabela, de uma função... ;-)
- d. Imagine que a engine pega as tabelas indicadas na cláusula FROM e as separa. Cria uma tabela virtual e temporária.

22. WHERE

- a. É a cláusula que filtra as linhas que devem ser retornadas na consulta
- b. É a segunda etapa da consulta
- c. Como funciona o filtro? É dada uma condição e só as linhas que obedecem a condição serão retornadas.
- d. Lembrando que os ALIAS definidos na cláusula SELECT não podem ser usados na cláusula WHERE

23. GROUP BY

- a. Traduzo para: Agrupe as linhas por...
- b. Lembra que eu falei que a engine pega as tabelas da cláusula FROM e deixa separado? Na realidade internamente é criada uma tabela virtual, onde estão os dados das tabelas da cláusula FROM. Depois os dados desta tabela virtual são filtrados pela cláusula WHERE. E nesta etapa são agrupados, e só os grupos diferentes farão parte do retorno da query.
- c. O resultado final da execução desta cláusula deve ser um valor para cada grupo
- d. Se na cláusula SELECT vc usar a uma coluna que não faz parte do agrupamento, vc precisará usar uma função, por exemplo, MAX, MIN, AVG e afins... Isso porque só um valor deve ser retornado por grupo. (dúvidas aqui? Podemos fazer um exemplo para deixar mais claro, acho que na prática é mais fácil de entender ☺)

24. HAVING

- a. Essa é legal, já vi uns desenvolvedores velhos que não sabem usar direito... Having é um filtro que é executado depois que os dados são agrupados.
- b. Também contém em uma condição, ou seja, é aplicado um filtro no grupo e só os grupos que obedecem uma condição serão retornados;
 - i. Por exemplo, SELECT Nome Funcionário, Idade Funcionário FROM Funcionário GROUP BY Idade Funcionário, Nome Funcionário HAVING AVG(Idade Funcionário) > 35
 - ii. Ou seja, selecione o nome e a idade dos funcionários, agrupando os funcionários por idade e por nome, e filtre pelos grupos de funcionários que possuem a média de idade maior que 35 anos.
- c. A diferença entre o HAVING e o WHERE (importante!!!)

- i. WHERE = filtro executado sobre um conjunto de linhas, antes delas serem agrupadas. Ou seja, a condição é verificada para cada linha;
- ii. HAVING = filtro executado após o agrupamento das linhas, avalia uma determinada condição para cada grupo.

25. SELECT

- a. Viram quanta coisa é executada antes da cláusula SELECT ser executada? É a primeira cláusula escrita e a quinta a ser executada;
- b. É uma cláusula obrigatória
- c. É executada em duas etapas:
 - i. Verificar se as expressões usadas produzem algum valor, se as colunas existem. E se a query possui um agrupamento, as colunas listadas devem fazer parte da cláusula GROUP BY
 - ii. A segunda etapa será executada somente se vc estiver usando a cláusula DISTINCT (que faz parte da cláusula SELECT). Esta cláusula remove do retorno da query as informações duplicadas.
- d. Se criamos um ALIAS na cláusula SELECT, ele não é “visto” pelas outras colunas dentro do mesmo comando, por exemplo:
 - i. SELECT YEAR(AnoNascimento) AS Ano , Ano + 6 AS EntradaEscola
 - 1. FROM Pessoa
 - 2. Esta query retornará um erro porque estamos usando um ALIAS que acabou de ser gerado. O correto seria:
 - ii. SELECT YEAR(AnoNascimento) AS Ano , YEAR(AnoNascimento) + 6 AS EntradaEscola FROM Pessoa
- e. Perceberam que na cláusula SELECT podemos usar colunas, funções, fazer cálculos...

26. ORDER BY

- a. A última cláusula a ser executada;
- b. Significa que uma query deve ser apresentada de acordo com a ordem definida nesta cláusula.
- c. É possível usar os ALIAS definidos na Cláusula SELECT. Isso porque ela é executada após o SELECT 😊
- d. Cuidado com as ordenações, elas podem ser custosas...

Pessoal, acho que tivemos muitaaa informação, e a ideia é que vcs entendam bem o que “conversamos” até aqui. Inicialmente eu queria conversar sobre o capítulo 1 inteiro. Mas acho que não vai ser saudável! Banco de dados é uma coisa gostosa (acreditem!) e acho que correr não nos ajudará! Paramos na metade do capítulo 1 (página 17) e nossos próximos alvos são as cláusulas FROM e SELECT. Nos profundaremos bastante nelas 😊

Beijos para todos e até domingo!

Dani