**SIMULADO 1 BIMESTRE - BASEADO EM COMO SERÁ A PROVA**

**1.** Analise as afirmações abaixo: (usada)

i - Um backup diferencial captura todas as alterações feitas desde o último backup completo.

ii - Um backup incremental captura apenas as alterações desde o último backup feito, seja ele completo, diferencial ou incremental

iii – O tamanho do backup diferencial geralmente é menor que o incremental

iv – Os backup’s incrementais e diferenciais, quando usados em conjunto, substituem o backup completo.

Está correto o que se afirma em:

1. i, ii, iv
2. i e ii
3. i e iii
4. ii e iv
5. ii e iii

**R:** A Resposta correta é a B – i, ii.

**3.** O comando **select\* into ##tab\_temp from empresa** terá o seguinte resultado.(usada)

1. Irá criar uma tabela temporária com todos os campos e dados da tabela empresa. A tabela temporária criada terá um escopo global, estará visível para qualquer sessão e ficará ativa até que a sessão que a criou permaneça ativa. A tabela temporária criada ficará armazenada no banco de dados tempdb.
2. Irá criar uma tabela temporária vazia, com todos os campos da tabela empresa. A tabela temporária criada terá um escopo local, estará visível para qualquer sessão do servidor e ficará ativa até que a sessão que a criou permaneça ativa. A tabela temporária ficara armazenada no banco de dados tempdb.
3. Irá criar uma tabela temporária vazia, com todos os campos da tabela empresa. A tabela temporária criada terá um escopo local, estará visível somente para a sessão que a criou e será excluída quando a sessão que a criou for encerrada. A tabela temporária criada ficara armazenada no mesmo banco de dados da tabela empresa.
4. Irá criar uma tabela temporária com todos os campos e dados da tabela empresa. A tabela temporária criada terá um escopo local, estará visível somente para a sessão que a criou e será excluída quando a sessão que a criou for encerrada. A tabela temporária criada ficara armazenada no mesmo banco de dados da tabela empresa.
5. Irá criar uma tabela normal com o nome ##tab\_temp no mesmo banco de dados da tabela empresa.

**R:** Alternativa Correta é a primeira (i). OBS: **#** -> Escopo Local; **##** -> Escopo Global.

**4.** Assinale a alternativa que descreve as VIEWS correta. (usada)

1. É uma tabela física, derivada de outra tabela ou consulta e uma vez criada, não tem nenhum vínculo com a tabela ou consulta que a originou. Alterações feitas nas tabelas de origem não afetam a view, e alterações feitas na view nunca afetam as tabelas que a origem
2. É uma tabela virtual, derivada de outra tabela ou consulta e após a criação mantém um vínculo com as tabelas de origem. Alterações na tabela de origem irão afetar a view e alterações feitas na view podem afetar as tabelas de origem
3. É uma tabela virtual, derivada de outra tabela ou consulta e após a criação não mantém nenhum vínculo com as tabelas de origem. Alterações na tabela de origem não irão afetar a view e alterações feitas na view podem afetar as tabelas de origem
4. É uma tabela virtual, derivada de outra tabela ou consulta e após a criação mantém um vínculo com as tabelas de origem. Alterações na tabela de origem afetam a view e alterações feitas na view não afetam as tabelas de origem
5. É uma tabela fisica, derivada de outra tabela ou consulta e após a criação mantém um vínculo com as tabelas de origem. Alterações na tabela de origem afetam a view e alterações feitas na view podem afetar as tabelas de origem

**R:** Alternativa Correta é a segunda (ii).

**5.** Qual das seguintes afirmações melhor descreve uma view em um banco de dados?

a) Uma view é uma cópia física dos dados que pode ser modificada sem afetar as tabelas originais.

b) Uma view é uma tabela virtual que não mantém qualquer vínculo com as tabelas originais após sua criação.

c) Uma view é uma tabela virtual que representa dados derivados de uma consulta ou de uma ou mais tabelas originais.

d) Uma view é um procedimento armazenado que automatiza tarefas no banco de dados.

e) Uma view é um conjunto de regras aplicadas ao banco de dados para garantir a segurança dos dados.

**R:** Alternativa Correta Letra C.

**6.** Considere que uma view pode ser criada para simplificar consultas complexas, mas também pode gerar problemas de desempenho em cenários de alto volume de dados. Explique em que situações o uso de views é recomendável e em quais casos pode se tornar um gargalo.

**R:** A diferença principal é que a cláusula WHERE filtra registros individuais antes do agrupamento, enquanto a cláusula HAVING filtra grupos após a agregação ter sido realizada. Portanto, WHERE é usada para critérios de linhas individuais e não pode conter funções de agregação, ao passo que HAVING é usada com funções de agregação (como COUNT(), SUM()) e exige a cláusula GROUP BY para filtrar os resultados agregados. Ex:

SELECT Departamento, SUM(Valor) AS TotalVendas

FROM Vendas

WHERE Produto = 'Teclado'

GROUP BY Departamento

HAVING SUM(Valor) > 1000;

**7.** Qual a diferença entre as cláusulas **WHERE** e **HAVING** em consultas que utilizam agregações? Apresente um exemplo prático onde o uso de HAVING seja indispensável, justificando sua resposta.

**R:** A diferença principal é que a cláusula WHERE filtra registros individuais antes do agrupamento, enquanto a cláusula HAVING filtra grupos após a agregação ter sido realizada. Portanto, WHERE é usada para critérios de linhas individuais e não pode conter funções de agregação, ao passo que HAVING é usada com funções de agregação (como COUNT(), SUM()) e exige a cláusula GROUP BY para filtrar os resultados agregados.

* OBS: A cláusula WHERE pode ser usada para filtrar vendas de um produto específico ("Teclado"), mas não pode determinar quais departamentos tiveram um SUM(Valor) acima de 1000, pois SUM() é uma função de agregação que opera em um grupo de linhas, não em linhas individuais.
* A cláusula HAVING é aplicada após o GROUP BY Departamento, o que permite que ela use a função de agregação SUM(Valor) para verificar o total de vendas de cada departamento e manter apenas aqueles com um total acima de 1000.

**8.** Suponha que você tenha a tabela **Funcionarios(id\_func, nome, departamento, salario)**.  
Crie uma **VIEW** chamada vw\_salario\_depto que mostre o salário médio de cada departamento.

Depois, escreva uma consulta sobre essa view que retorne apenas os departamentos com salário médio **acima de R$ 5.000**.

**R:** CREATE VIEW vw\_salario\_depto AS

SELECT

departamento,

AVG(salario) AS salario\_medio

FROM

Funcionarios

GROUP BY

departamento;

**Consulta:**

SELECT

departamento

FROM

vw\_salario\_depto

WHERE

salario\_medio > 5000;