Avaliacao

# QUERY PARA AVALIAR NOTAS DOS QUARTOS

Query para identificar a avaliação dada tanto para o quarto quanto para a estadia

SELECT

DISTINCT (qu.id\_quarto + 100) quarto,

ROUND(AVG(av.conforto), 2) conforto,

ROUND(AVG(av.custo\_beneficio), 2) custo\_beneficio,

ROUND(AVG(av.funcionarios), 2) 'funcionários',

ROUND(AVG(av.localizacao), 2) 'localização',

ROUND(AVG(av.limpeza), 2) limpeza,

ROUND((AVG(av.conforto) + AVG(av.custo\_beneficio) + AVG(av.localizacao) + AVG(av.limpeza) + AVG(av.funcionarios))/5, 2) total

FROM [scComodo].[Avaliacao] av

INNER JOIN [scReserva].[Reserva] rs

ON rs.id\_reserva = av.id\_reserva

INNER JOIN [scComodo].[Quarto] qu

ON qu.id\_quarto = rs.id\_quarto

GROUP BY qu.id\_quarto

ORDER BY quarto

GO

# QUERY PARA ANALISAR OS PRINCIPAIS TIPOS DE AVALIAÇÃO

Identifica os principais meios de contato usados pelos usuários ao dar uma avaliação

SELECT

ct.descricao tipo\_contato,

COUNT(ct.descricao) total\_avaliacao

FROM [scComodo].[Avaliacao] av

INNER JOIN [scReserva].[Tipo\_Contato] ct

ON ct.id\_tp\_contato = av.tipo\_contato

GROUP BY ct.descricao

ORDER BY total\_avaliacao DESC

## ÍNDICE: Primary Key (tipo\_contato, id\_reserva)

### 1º SELECT

Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.

Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.

Table 'Reserva'. Scan count 1, logical reads 9, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.

Table 'Avaliacao'. Scan count 1, logical reads 11, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.

Table 'Quarto'. Scan count 1, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.

Diagrama

Descrição gerada automaticamente

Principais custos: Sort e Inner Join

Tempo total: 0.016s

### 2º SELECT

Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.

Table 'Tipo\_Contato'. Scan count 0, logical reads 12, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.

Table 'Avaliacao'. Scan count 1, logical reads 10, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Principais custos: Sort e seek no índice

Tempo total: menos de 0.001s

### Resultado

Resultado: foi decidido manter o formato atual de índice, pois atua perfeitamente tanto para o primeiro select quanto para o segundo, alterando apenas o fillfactor para 90

Na 1ª query, após a implementação do fillfactor e rebuild do índice, foi verificado a diminuição de logical reads na tabela “avaliação” de 11 para 10 e um custo de 0.001s a menos no inner join

Foi constado de que não há uma necessidade na adoção de um índice não clustered, o índice atual já supre todas as necessidades.

Quarto

# QUERY PARA ANALISAR O USO DO QUARTO

Seu objetivo é ver detalhes básicos do quarto e a necessidade de adição de novas camas

SELECT

DISTINCT (qu.id\_quarto + 100) quarto,

tq.descricao categoria,

qu.valor\_diaria diaria,

COUNT(rv.id\_quarto) solicitacoes,

CONCAT(qu.num\_camas, '-', AVG(rs.id\_servico)) 'disponibilidade x solicitação'

FROM [scComodo].[Quarto] qu

INNER JOIN [scComodo].[Tipo\_Quarto] tq

ON tq.id\_tp\_quarto = qu.tipo\_quarto

INNER JOIN [scReserva].[Reserva] rv

ON rv.id\_quarto = qu.id\_quarto

INNER JOIN [scReserva].[Reserva\_Servico] rs

ON rs.id\_reserva = rv.id\_reserva

INNER JOIN [scComodo].[Avaliacao] av

ON av.id\_reserva = rv.id\_reserva

WHERE

rs.id\_servico = 5

GROUP BY qu.id\_quarto, qu.valor\_diaria, tq.descricao, qu.num\_camas

ORDER BY quarto

ÍNDICE: Primary Key (id\_quarto)

Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.

Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.

Table 'Reserva'. Scan count 1, logical reads 9, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.

Table 'Avaliacao'. Scan count 1, logical reads 10, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.

Table 'Quarto'. Scan count 1, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead reads 0, lob page server read-ahead reads 0.

Gráfico

Descrição gerada automaticamente com confiança média

Principais custos: Inner joins

Tempo total: 0.056s

Resultado

Resultado: foi decidido manter o formato atual de índice, pois atua perfeitamente para o select desejado, alterando apenas o fillfactor para 90

Após a implementação do fillfactor e rebuild do índice, foi verificado a diminuição de logical reads e scan Count e um custo de 0.045s a menos na realização da query

Foi constado de que não há uma necessidade na adoção de um índice não clustered, o índice atual já supre todas as necessidades.

Reserva

QUERY PARA ANALISAR OS PRINCIPAIS SERVIÇOS DE CADA TIPO DE HÓSPEDE

QUERY PARA ANALISAR O TIPO DE HÓSPEDE E SEUS PRINCIPAIS SERVIÇOS SOLICITADOS

QUERY PARA DETERMINAR: TEMPO MÉDIO DE ESTADIA NÃO CANCELADA

QUERY PARA DETERMINAR MOVIMENTO E LUCRO DE CADA ÉPOCA

A query analisa as reservas e verifica a média de dias, a média e o total de hóspedes e a média e o total de lucro, agrupando-os em estações do ano

WITH reservaCTE AS (

SELECT

id\_reserva,

num\_hospede,

valor\_total,

DAY(data\_checkou - data\_checkin) Dias,

CASE

WHEN MONTH(data\_checkin) <= 3 THEN 'Verão'

WHEN MONTH(data\_checkin) <= 6 THEN 'Primavera'

WHEN MONTH(data\_checkin) <= 9 THEN 'Inverno'

WHEN MONTH(data\_checkin) <= 12 THEN 'Outono'

END Periodo

FROM [scReserva].[Reserva])

SELECT

Periodo,

COUNT(id\_reserva) Reservas,

AVG(Dias) Media\_Dias,

AVG(num\_hospede) Media\_Hospedes,

SUM(num\_hospede) Total\_Hospedes,

CONCAT('R$ ', FORMAT(AVG(valor\_total), 'N2')) Media\_Lucro,

CONCAT('R$ ', FORMAT(SUM(valor\_total), 'N2')) Lucro\_Total

FROM reservaCTE

GROUP BY Periodo

ORDER BY Reservas DESC

TABELAS DESCRITIVAS

Nas tabelas Tipo\_Hospede, Funcao\_Funcionario, Tipo\_Quarto, Tipo\_Pagamento, Tipo\_Cancelamento, Tipo\_Contato e Servico não foi visto uma necessidade de alteração e/ou manipulação. As tabelas citadas são descritivas, contendo apenas uma Primary Key (Identity e Índice Clustered) e uma descrição (com exceção de Servico, que conta com a coluna ‘valor’). Seus índices clustereds atuais já suprem as necessidades.