

Aula 7 - Prática com exercícios

Docupedia Export

Author:Goncalves Donathan (SO/OPM-TS21-BR)

Date:17-Jul-2024 14:48

Table of Contents

1	Prática de alguns exercícios	3
2	Group By - Agrupar todos os casos de determinado registro.	4
3	3.2 Quanto tempo(Minutos) cada sala foi ou será utilizada no total?	5
4	4. Encontre a quantidade de eventos que cada pessoa participou.	6
5	5. Preciso controlar se as reuniões estão sendo realizadas nas salas com a capacidade correta.	7
5.1	Dica: Use "Case When"	7
6	6. Verificar os equipamentos que estão disponíveis para atribuição.	8
6.1	Fim do cenário proposto no começo das aulas.	8
7	IN - Seleciona registros que estejam em um grupo de informações.	9
8	LIKE - Encontra as Strings que possuem uma parte específica de caracteres.	10
9	VIEW	11
9.1	Utilizado para salvar uma visualização do Banco.	11
10	FUNÇÕES	13
10.1	Utilizado para retornar dados em formato de tabelas, evitando repetição.	13
10.2	Função que retorna valores	13
10.3	Função que retorna tabela	13
11	Crie uma função cujo o retorno deverá ser um único select e retornar uma tabela com:	15

1 Prática de alguns exercícios

2 Group By - Agrupar todos os casos de determinado registro.

ID	Nome	Idade	Cidade
1	Rodrigo	27	Curitiba
2	David	36	Araucária
3	Rebeca	23	Curitiba

SELECT Cidade, **COUNT**(*) **AS** Qtd
FROM Funcionario
GROUP BY Cidade

Cidade	Qtd
Curitiba	2
Araucária	1

ID	Nome	Setor	Salario
1	Rodrigo	TI	6.000
2	David	RH	8.000
3	Rebeca	TI	10.000

SELECT Setor, **SUM**(Salario) **AS** Total
FROM Funcionario
GROUP BY Setor
HAVING **SUM**(Salario) > 10000

Setor	Total
TI	16.000

3 3.2 Quanto tempo(Minutos) cada sala foi ou será utilizada no total?

Tempo de cada sala

Tempo total de cada sala

```
SELECT
    S.Nome AS "Nome da Sala",
    SUM(DateDiff(MINUTE, E.DtHrInicio, E.DtHrFim)) AS "Utilização (minutos)"

FROM Sala S
INNER JOIN Evento E
    ON S.IDSala = E.IDSala

GROUP BY S.Nome
```

4. Encontre a quantidade de eventos que cada pessoa participou.

- Os que mais participaram, aparecem por primeiro (Decrescente).
- Deve aparecer "0" para quem nunca participou (Vai ocorrer).

Quantidade por pessoa

```
SELECT
    P.Nome,
    COUNT(EP.IDEvento) AS Quantidade
FROM Pessoa P
LEFT JOIN EventoPessoa EP
    ON P.IDPessoa = EP.IDPessoa
GROUP BY P.Nome
ORDER BY Quantidade DESC
```

5. Preciso controlar se as reuniões estão sendo realizadas nas salas com a capacidade correta.

- Liste as **Salas**, **Eventos**, **Capacidade** das salas e **Quantidade de Participantes**.
- Crie uma **coluna** chamada "**Avaliação da Capacidade**" com dados como: "**Ultrapassou o limite**", "**Limite ok**".

5.1 Dica: Use "Case When"

Resolução

Avaliar a capacidade das salas

```
SELECT
    E.Descricao AS "Evento",
    S.Nome      AS "Sala",
    S.Capacidade,
    COUNT(EP.IDPessoa) AS "Quantidade Participantes",

    CASE WHEN S.Capacidade < COUNT(EP.IDPessoa) THEN 'Ultrapassou o Limite!'
    ELSE 'Limite OK' END AS "Avaliação da Capacidade"

FROM EventoPessoa EP
INNER JOIN Evento E
    ON EP.IDEvento = E.IDEvento
INNER JOIN Sala S
    ON S.IDSala = E.IDSala

GROUP BY E.Descricao, S.Nome, S.Capacidade, E.IDEvento
```

6. Verificar os equipamentos que estão disponíveis para atribuição.

Resultado

2 - Equipamentos disponíveis

```
SELECT
    Nome AS 'Equipamento',
    Tipo
FROM Equipamento

WHERE IDSala is NULL
```

6.1 Fim do cenário proposto no começo das aulas.

7 IN - Seleciona registros que estejam em um grupo de informações.

SUB-SELECT / IN

```
-- Selecionando eventos que utilizam equipamentos do tipo "Informática" ou "Multimídia"
SELECT * FROM Evento E
INNER JOIN EventoEquipamento EQ
    ON E.IDEvento = EQ.IDEvento

WHERE EQ.IDEquipamento IN

(SELECT IDEquipamento
FROM Equipamento
WHERE Tipo IN ('Informatica','Multimidia'))
```

8 LIKE - Encontra as Strings que possuem uma parte específica de caracteres.

Like

-- Deve terminar com "IOT" na descrição

```
SELECT * FROM Evento WHERE Descricao LIKE '%IOT'
```

-- Deve começar com "Aula" na descrição

```
SELECT * FROM Evento WHERE Descricao LIKE 'Aula%'
```

-- Palestra em qualquer lugar da descrição

```
SELECT * FROM Evento WHERE Descricao LIKE '%Industria%'
```

9

VIEW

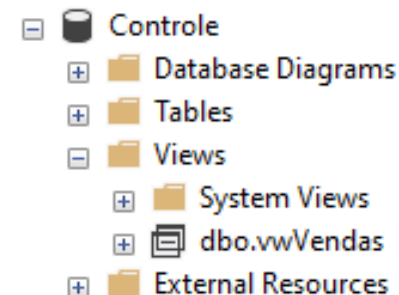
9.1

Utilizado para salvar uma visualização do Banco.

View

```
-- Criando View
CREATE VIEW vwVendas AS
SELECT * FROM Tabela

-- Utilizando View
Select * From vwVendas
ORDER BY MesAno
```



7. Refazendo exercício 5.

- Com o código feito no exercício 5, que foi classificado com "ok" ou "ultrapassado" das pessoas que confirmaram estar presentes no evento.
- Encontre com Sub-Select, os eventos que iriam ter um limite ultrapassado, mas que não tiveram por causa de algumas pessoas que faltaram no dia.
- Crie uma VIEW para salvar esse Select.

Resolução

```
CREATE VIEW vwCapacidadeEventos as
SELECT
    E.Descricao,
    S.Capacidade,
    COUNT(EP.IDPessoa) AS 'Presentes'
FROM Evento E
INNER JOIN EventoPessoa EP
    ON E.IDEvento = EP.IDEvento
INNER JOIN Sala S
    ON S.IDSala = E.IDSala
GROUP BY E.IDEvento, S.Capacidade, EP.Presenca, E.Descricao
```

HAVING

```
COUNT(EP.IDPessoa) <= S.Capacidade AND  
EP.Presenca = 1 AND  
E.IDEvento IN
```

```
(
```

SELECT

```
E.IDEvento
```

```
FROM Evento E
```

```
INNER JOIN EventoPessoa EP
```

```
ON E.IDEvento = EP.IDEvento
```

```
INNER JOIN Sala S
```

```
ON S.IDSala = E.IDSala
```

```
GROUP BY E.IDEvento, S.Capacidade
```

```
HAVING COUNT(EP.IDPessoa) > S.Capacidade
```

```
)
```

```
SELECT * FROM vwCapacidadeEventos
```

10

FUNÇÕES

10.1 Utilizado para retornar dados em formato de tabelas, evitando repetição.

10.2 Função que retorna valores

Criar função que retorna valores

```
CREATE FUNCTION fMesAno (@Data DateTime)
RETURNS VARCHAR(7) AS
BEGIN
    RETURN(SUBSTRING(CONVERT(VARCHAR(10), @Data, 103), 4, 10))
END
```

Utilizar

```
SELECT
    dbo.fMesAno(DtHrInicio) AS MesAno,
    COUNT(*) AS Eventos
FROM Evento
GROUP BY dbo.fMesAno(DtHrInicio)
```

10.3

Função que retorna tabela

Criar função que retorna tabelas

```
CREATE FUNCTION fnVendaAcimaDe(@Valor float)
RETURNS TABLE AS
RETURN(SELECT * FROM NotaFiscal WHERE Valor >= @Valor)
```

Utilizar

```
SELECT * FROM fnVendaAcimaDe(1000)
```

Toda variável deve possuir um @ no começo da palavra

```
-- Declarando Variável  
DECLARE @Variavel INT  
SELECT @Variavel = Capacidade FROM Sala
```

11 Crie uma função cujo o retorno deverá ser um único select e retornar uma tabela com:

- 1- **Quantidade de pessoas** presentes no **primeiro evento (mais antigo)** que ocorreu.
- 2- **Nome da sala** em que esse evento ocorreu.
- 3- A Função será chamada com um **parâmetro de busca**. Como **Aula, Palestra ou Reunião**.
Nome Evento, Nome Sala, Qtd Pessoas, Data Início, Hora Início.

Desafio

```
-- Criando a função
CREATE FUNCTION fEventoAntigo (@Valor Varchar(20))
RETURNS TABLE AS
RETURN(
    SELECT TOP 1
        E.Descricao AS 'Evento',
        S.Nome      AS 'Sala',
        COUNT(EP.IDPessoa) AS 'Qtd Pessoas',
        Convert(Varchar(10), E.DtHrInicio, 103) AS 'Data de Início',
        Convert(Varchar(10), E.DtHrInicio, 108) AS 'Hora Início'
    FROM Evento E
    INNER JOIN Sala S
        ON S.IDSala = E.IDSala
    INNER JOIN EventoPessoa EP
        ON EP.IDEvento = E.IDEvento
    WHERE E.Descricao LIKE @Valor+'%'
    GROUP BY E.Descricao, S.Nome, E.DtHrInicio
    ORDER BY E.DtHrInicio
)

-- Verificando os dados
SELECT * FROM fEventoAntigo('Aula')
```