

## *Tarefa Orientada 13*

### *Agrupamento e sumário de dados*

#### Objectivos:

- Funções de agregação
- Agrupamento e sumário de dados

#### **Funções de agregação**

Nesta tarefa orientada iremos formular consultas que sumarizam os dados armazenados na base de dados. Por exemplo, pode utilizar consultas de sumário para obter as vendas totais por vendedor ou localidade, ou para obter o número de facturas processadas em cada dia de um determinado mês. Iremos também usar funções de agregação que permitem calcular, por exemplo, médias ou valores mínimos e máximos para um determinado campo de uma tabela.

Na tarefa orientada 10 foram introduzidas algumas funções escalares que permitem realizar operações sobre um valor e devolvem um valor unitário. Nesta tarefa orientada, vamos utilizar funções de agregação que permitem efectuar cálculos sobre os valores de um conjunto de registos e devolvem um valor sumário. Os valores a serem usados nos cálculos são especificados através de expressões indicadas nos argumentos da função de agregação. Na maior parte dos casos, a expressão é constituída apenas pelo nome de um campo.

Uma instrução de *SELECT* que inclua funções de agregação pode ser chamada de consulta sumário.

A seguir, apresenta-se a sintaxe para algumas funções de agregação.

Função	Resultado
AVG ([ <u>ALL</u>   DISTINCT] expressão)	Média dos valores não nulos da expressão usada no argumento da função
SUM ([ <u>ALL</u>   DISTINCT] expressão)	Total dos valores não nulos da expressão usada no argumento da função
MIN ([ <u>ALL</u>   DISTINCT] expressão)	Mínimo dos valores não nulos da expressão usada no argumento da função
MAX ([ <u>ALL</u>   DISTINCT] expressão)	Máximo dos valores não nulos da expressão usada no argumento da função
COUNT ([ <u>ALL</u>   DISTINCT] expressão)	Número de valores não nulos da expressão usada no argumento da função
COUNT (*)	Número de registos seleccionados pela consulta

A expressão especificada para as funções *AVG* e *SUM* devem resultar num valor numérico. A expressão especificada para as funções *MIN*, *MAX* e *COUNT* podem resultar num valor numérico, data ou cadeia de caracteres (*string*).

De modo predefinido, são incluídos nos cálculos todos (*ALL*) os valores, independentemente de serem, ou não, duplicados. No entanto, se pretender

omitir os valores duplicados, utilize a palavra-chave *DISTINCT*. Normalmente, esta palavra-chave é apenas utilizada com a função *COUNT*, pois não produz nenhum efeito usá-la com as funções *MIN* e *MAX*, nem faz, usualmente, sentido utilizá-la com as funções *AVG* e *SUM*. Todas as funções de agregação, excepto a função *COUNT(\*)*, ignoram os valores nulos.

As funções de agregação são comumente utilizadas com a cláusula *GROUP BY* que é usada para agrupar registos. Mais à frente voltaremos a este assunto.

Se utilizar funções de agregação na cláusula *SELECT* de uma consulta, essa cláusula não pode incluir outras colunas sem funções de agregação, excepto se a coluna contém um valor previamente definido e se a consulta incluir uma cláusula *GROUP BY*. Neste último caso, a cláusula *SELECT* pode incluir quaisquer das colunas que são especificadas na cláusula *GROUP BY*.

## EXEMPLOS

- 1 Formule, analise e execute as oito consultas a seguir apresentadas.
  - 1.1 Consulta que devolve o número de facturas que ainda não foram pagas e o montante total em dívida.

```
SELECT COUNT(*) AS [Número de facturas não pagas],  
       SUM (TotalFactura - Pagamento - Crédito) AS [Total em dívida]  
FROM Facturas  
WHERE TotalFactura - Pagamento - Crédito > 0
```

Note que a expressão usada como argumento da função *SUM* calcula o montante em dívida de uma factura. Consequentemente, o resultado da execução da função *SUM* é um valor que representa a quantia total em dívida para todas as facturas seleccionadas pela consulta. Após observar a cláusula *WHERE* da consulta, pode concluir que apenas vão ser incluídas as facturas que ainda não foram pagas, isto é, as facturas que têm um montante em dívida superior a zero.

A utilização da função de agregação *COUNT(\*)* permite obter o número de registos da tabela *Facturas* que satisfazem a condição de pesquisa da consulta, isto é, permite obter o número de facturas não pagas.

A seguir, apresenta-se o resultado obtido.

	Número de facturas não pagas	Total em dívida
1	5	6219,56

Note que o resultado desta consulta indica que existem cinco facturas ainda por pagar.

1.2 Consulta que devolve o número de facturas com data posterior a 1/1/2006. A consulta devolve ainda a quantia média e o total do montante dessas facturas.

```
SELECT 'Após 1/1/2006' AS [Data inicial de pesquisa],
COUNT(*) AS [Número de facturas],
AVG (TotalFactura) AS [Montante médio das facturas],
SUM (TotalFactura) AS [Total do montante das facturas]
FROM Facturas
WHERE DataFactura > '2006-01-01'
```

Analisando detalhadamente esta consulta, podemos verificar que é criada uma coluna com a designação *Data inicial de pesquisa* e com o "valor" *Após 1/1/2006*. É utilizada a função de agregação *COUNT(\*)* para contar os registos que satisfazem a condição de pesquisa, isto é, as facturas cuja data é posterior a 1/1/2006. Utiliza-se a função de agregação *AVG* para calcular a média do montante das facturas que verificam a condição de pesquisa. Usa-se também a função de agregação *SUM* para calcular a soma do montante das facturas cuja data é posterior a 1/1/2006.

A seguir, apresenta-se o resultado obtido.

	Data inicial de pesquisa	Número de facturas	Montante médio das facturas	Total do montante das facturas
1	Após 1/1/2006	17	4380,9305	74475,82

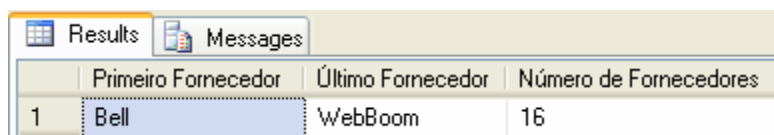
Embora as funções *MIN*, *MAX* e *COUNT* sejam normalmente usadas sobre colunas que contêm dados numéricos, também podem ser utilizadas em colunas que contêm dados alfanuméricos e no formato data.

1.3 Consulta que devolve o nome do primeiro e do último fornecedor por ordem alfabética, bem como o número de fornecedores registados na base de dados.

```
SELECT MIN (Nome) AS [Primeiro Fornecedor],  
       MAX (Nome) AS [Último Fornecedor],  
       COUNT (Nome) AS [Número de Fornecedores]  
FROM Fornecedores
```

Dado que o campo *Nome* da tabela *Fornecedores* não pode conter valores nulos, a função *COUNT(\*)* teria devolvido exactamente o mesmo resultado.

A seguir, apresenta-se o resultado obtido.



	Primeiro Fornecedor	Último Fornecedor	Número de Fornecedores
1	Bell	WebBoom	16

## Agrupamento e sumário de dados

Agora vamos formular consultas que agrupam dados e usam funções de agregação para sumariar os dados de cada grupo. A seguir apresenta-se a sintaxe de uma instrução *SELECT* com cláusulas *GROUP BY* e *HAVING*.

```
SELECT lista_de_colunas
FROM tabela_fonte
[WHERE condições_de_pesquisa]
[GROUP BY lista_de_colunas]
[HAVING condições_de_pesquisa]
[ORDER BY lista_de_colunas]
```

A cláusula *GROUP BY* determina o modo como os registos seleccionados são agrupados. A cláusula *HAVING* especifica uma condição de pesquisa para os grupos de registos. Estas cláusulas aparecem após a cláusula *WHERE* e antes da cláusula *ORDER BY*. Isto faz sentido, uma vez que a condição de pesquisa especificada na cláusula *WHERE* é aplicada antes dos registos serem agrupados e a ordenação dos registos incluídos no resultado final é efectuada depois dos registos serem agrupados.

Na cláusula *GROUP BY* especificam-se uma ou mais colunas ou expressões, separadas por vírgulas. Depois, os registos que satisfazem as condições de pesquisa definidas na cláusula *WHERE* são agrupados por essas colunas ou expressões em sequência ascendente. Consequentemente, é devolvido um único registo por cada conjunto distinto de valores das colunas especificadas na cláusula *GROUP BY*.

1.4 Consulta que calcula, para cada fornecedor, o montante médio das facturas. Apenas são devolvidos os grupos de registos relativos aos fornecedores cujas facturas apresentam um montante médio superior a 2000 euros.

```
SELECT Fornecedor, AVG (TotalFactura) AS [Montante médio das facturas]
FROM Facturas
GROUP BY Fornecedor
HAVING AVG (TotalFactura) > 2000
ORDER BY [Montante médio das facturas] DESC
```

Esta instrução *SELECT* calcula o montante médio das facturas de cada fornecedor cujas facturas apresentam um montante médio superior a 2000 euros. Para tal, agrupa as facturas por vendedor. Depois, a função de agregação *AVG* calcula a média do campo *TotalFactura*. Dado que esta consulta inclui uma cláusula *GROUP BY*, a função *AVG* vai calcular a média do campo *TotalFactura* para cada vendedor, em vez de efectuar o cálculo relativamente ao resultado global (como nos exemplos anteriores). Por esse motivo, a função de agregação pode ser designada por função de agregação vectorial, enquanto que as funções de agregação que abordámos nos exemplos anteriores podem ser designadas por funções de agregação escalar, pois devolviam um resultado unitário para todos os registos do resultado.

Esta consulta também inclui uma cláusula *HAVING*. A condição de pesquisa especificada nesta cláusula indica que apenas os fornecedores cujas facturas apresentam um montante médio superior a 2000 euros devem ser incluídos no resultado. Note que esta condição é aplicada depois dos registos estarem agrupados e a média ter sido calculada para cada grupo de registos. Resumindo, esta consulta começa por agrupar os registos por fornecedor, através da cláusula *GROUP BY*. Depois, calcula a média do campo *TotalFactura* para cada fornecedor, através da função de agregação *AVG* utilizada na cláusula *SELECT* e, finalmente, inclui no resultado apenas os registos que verificam a condição de pesquisa da cláusula *HAVING*.

A seguir, apresenta-se o resultado obtido.

	Fornecedor	Montante médio das facturas
1	110	23716,29
2	122	2640,7962

Portanto, a cláusula *GROUP BY* agrupa os registos do resultado baseando-se numa ou mais colunas ou em expressões. Esta cláusula é normalmente utilizada em consultas que incluem funções de agregação.

Quando uma consulta inclui uma cláusula *GROUP BY*, a cláusula *SELECT* pode incluir funções de agregação, as colunas usadas pela cláusula *GROUP BY* e expressões que resultem em valores constantes.

Se forem incluídas funções de agregação na cláusula *SELECT* de uma consulta que contém uma cláusula *GROUP BY*, a função de agregação efectua o cálculo para cada grupo de registos.

Se forem incluídas duas ou mais colunas ou expressões na cláusula *GROUP BY*, essas colunas formam uma hierarquia onde cada coluna ou expressão é subordinada à anterior.

A cláusula *HAVING* especifica uma condição de pesquisa para um grupo de registos. Esta condição é aplicada depois dos registos que satisfazem a condição de pesquisa especificada numa cláusula *WHERE* serem agrupados.

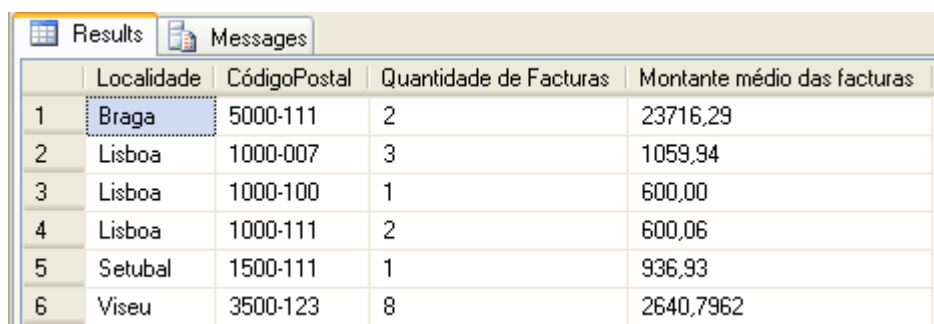
Pode utilizar uma junção numa consulta sumário para agrupar e sumarizar os dados de duas ou mais tabelas.

1.5 Consulta que devolve o número e o montante médio das facturas para os fornecedores de cada código postal de cada localidade.

```
SELECT Localidade, CódigoPostal,
       COUNT(*) AS [Quantidade de Facturas],
       AVG (TotalFactura) AS [Montante médio das facturas]
FROM Facturas INNER JOIN Fornecedores
ON Facturas.Fornecedor = Fornecedores.IDFornecedor
GROUP BY Localidade, CódigoPostal
ORDER BY Localidade, CódigoPostal
```

Esta consulta ilustra como se pode agrupar registos por mais do que uma coluna. Foi utilizada uma junção interna para combinar os valores dos campos *Localidade* e *CódigoPostal* da tabela *Fornecedores* com a contagem (*COUNT*) das facturas e com a média (*AVG*) do campo *TotalFacturas* tabela *Facturas*. Dado que os registos são agrupados por localidade e por código postal, é devolvido um registo para cada combinação destes dois campos. Depois, a cláusula *ORDER BY* ordena os registos também por localidade e por código postal. Note que, sem esta cláusula os registos seriam devolvidos sem qualquer sequência particular.

A seguir, apresenta-se o resultado obtido.



	Localidade	CódigoPostal	Quantidade de Facturas	Montante médio das facturas
1	Braga	5000-111	2	23716,29
2	Lisboa	1000-007	3	1059,94
3	Lisboa	1000-100	1	600,00
4	Lisboa	1000-111	2	600,06
5	Setubal	1500-111	1	936,93
6	Viseu	3500-123	8	2640,7962

Quando se inclui uma cláusula *WHERE* numa consulta que contém uma cláusula *GROUP BY* e funções de agregação, a condição de pesquisa é aplicada antes dos registos serem agrupados e os agregados serem calculados. Deste modo, apenas os registos que satisfazem as condições de pesquisa especificadas na cláusula *WHERE* da consulta são agrupados e sumariados.

Quando se inclui uma cláusula *HAVING* numa consulta que contém uma cláusula *GROUP BY* e funções de agregação, a condição de pesquisa é aplicada depois dos registos serem agrupados e os agregados serem calculados. Deste modo, apenas os grupos de registos que satisfazem as condições de pesquisa especificadas na cláusula *HAVING* da consulta são incluídos no resultado final.

Note que na cláusula *HAVING* apenas podem ser especificadas colunas que estejam incluídas na cláusula *SELECT* da consulta, pois filtra o resultado sumariado que está definido pelas cláusulas *SELECT*, *FROM*, *WHERE* e *GROUP BY*. Na cláusula *WHERE* podem ser especificadas quaisquer colunas pertencentes às tabelas fontes da consulta.

Por outro lado, as funções de agregação podem ser apenas especificadas na cláusula *HAVING*, ao passo que a cláusula *WHERE* de uma consulta não pode conter funções de agregação, pois a sua condição de pesquisa é aplicada antes dos registos serem agrupados.

1.6 Consulta que devolve o nome do fornecedor, a quantidade e o montante médio das suas facturas. Apenas são devolvidos os grupos de registos relativos aos fornecedores cujas facturas apresentam um montante médio superior a 500 euros.

```
SELECT Nome, COUNT(*) AS [Quantidade de Facturas],
        AVG (TotalFactura) AS [Montante médio das facturas]
FROM Fornecedores INNER JOIN Facturas
    ON Fornecedores.IDFornecedor = Facturas.Fornecedor
GROUP BY Nome
HAVING AVG (TotalFactura) > 500
ORDER BY [Quantidade de Facturas] DESC
```

Nesta consulta as facturas são agrupadas pelo nome de fornecedor. Depois, são contadas as facturas e é calculado o montante médio das facturas para cada grupo de facturas. Finalmente, são devolvidos apenas os grupos de registos que têm um montante médio das facturas superior a quinhentos euros.

A seguir, apresenta-se o resultado obtido.

	Nome	Quantidade de Facturas	Montante médio das facturas
1	Patinter	8	2640,7962
2	Mc Graw Hill	3	1059,94
3	Cartografia do Minho	2	23716,29
4	IBM	2	600,06
5	Bell	1	936,93
6	FCA	1	600,00

1.7 Consulta que devolve o nome do fornecedor, a quantidade e o montante médio das suas facturas com quantias inferiores a quinhentos euros.

```
SELECT Nome, COUNT(*) AS [Quantidade de Facturas],
        AVG (TotalFactura) AS [Montante médio das facturas]
FROM Fornecedores INNER JOIN Facturas
    ON Fornecedores.IDFornecedor = FActuras.Fornecedor
WHERE TotalFactura > 500
GROUP BY Nome
ORDER BY [Quantidade de Facturas] DESC
```

Esta consulta utiliza uma condição de pesquisa na cláusula *WHERE*, de modo a incluir nos grupos apenas as facturas com montantes superiores a quinhentos euros. Por outras palavras, a condição de pesquisa neste exemplo é aplicada a todos os registos. No exemplo anterior, a condição de pesquisa foi aplicada, na cláusula *HAVING*, a cada grupo de registos.

A seguir, apresenta-se o resultado obtido.

	Nome	Quantidade de Facturas	Montante médio das facturas
1	Patinter	8	2640,7962
2	Cartografia do Minho	2	23716,29
3	Mc Graw Hill	2	1433,135
4	Bell	1	936,93
5	FCA	1	600,00
6	IBM	1	1083,58

1.8 Consulta que agrupa, por data, as facturas cuja data se encontra entre 01-01-2006 e 31-05-2006 e calcula a quantidade e a soma do montante das facturas para cada data. Apenas devem ser incluídos no resultado os registos em cujas datas existem mais do que uma factura e cuja soma das facturas seja superior a cem euros.

```
SELECT DataFactura, COUNT(*) AS [Quantidade de Facturas],
        SUM(TotalFactura) AS [Montante total das facturas]
FROM Facturas
WHERE DataFactura BETWEEN '2006-01-01' AND '2006-05-31'
GROUP BY DataFactura
HAVING COUNT(*) > 1 AND SUM(TotalFactura) > 100
ORDER BY DataFactura DESC
```

Note que a condição de pesquisa especificada na cláusula *WHERE*, também poderia ser definida na cláusula *HAVING* da consulta. No entanto as outras duas condições de pesquisa da consulta têm que ser especificadas na cláusula *HAVING*, pois incluem funções de agregação. Se incluísse as três condições de pesquisa desta consulta na cláusula *HAVING*, o resultado final seria o mesmo.

## QUESTÕES

2 Crie as seguintes quatro consultas.

2.1 Crie uma consulta que devolva o número e o montante das facturas mais e menos dispendiosas, cuja data é posterior a 1/1/2006.

Resultado

	Data inicial de pesquisa	Número de facturas	Factura de maior montante	Factura de menor montante
1	Após 1/1/2006	17	26881,40	116,54

2.2 Crie uma consulta que devolva, para as facturas com data posterior a 1/1/2006, o número total de facturas, o número de fornecedores distintos, o montante médio e o total da quantia.

Resultado

	Número de Facturas	Número de Fornecedores	Montante médio das facturas	Total do montante das facturas
1	17	6	4380,9305	74475,82

Note que se pretender contar os registos com valores únicos de uma coluna, pode utilizar a função de agregação *COUNT* com a palavra-chave *DISTINCT* seguida pelo nome da coluna.

2.3 Crie uma consulta que devolva o número de facturas por fornecedor.

Resultado

	Fornecedor	Quantidade de Facturas
1	34	2
2	81	1
3	82	1
4	110	2
5	121	3
6	122	8

2.4 Crie uma consulta que devolva a quantidade e o montante médio das facturas para os fornecedores de cada código postal de cada localidade. O resultado deve ser ordenado por localidade e por código postal e deve incluir apenas os grupos de fornecedores por localidade e código postal que têm duas ou mais facturas.

Resultado

	Localidade	CódigoPostal	Quantidade de Facturas	Montante médio das facturas
1	Braga	5000-111	2	23716,29
2	Lisboa	1000-007	3	1059,94
3	Lisboa	1000-111	2	600,06
4	Viseu	3500-123	8	2640,7962