

Tarefa Orientada 11

Junção Interna

Objectivos:

- Junção Interna (INNER JOIN)

Junção Interna

A operação de junção interna (*INNER JOIN*) é utilizada para combinar colunas de duas ou mais tabelas. O resultado de uma consulta que contém uma operação *INNER JOIN*, inclui apenas os registos que satisfazem a(s) condição(ões) de junção.

Uma condição de junção refere uma coluna da cada uma das tabelas envolvidas na junção e indica o modo como as colunas devem ser comparadas.

Normalmente, utiliza-se o operador de igualdade para fazer a comparação dos valores das colunas usadas na condição de junção. Todavia, também se podem utilizar outros operadores relacionais.

Na maior parte dos casos, a operação de junção interna entre tabelas é baseada no relacionamento entre a chave primária de uma tabela e uma chave forasteira da outra tabela. Contudo, também pode formular condições de junção baseando-se em relacionamentos não definidos na base de dados. A este tipo de relacionamentos dá-se comumente a designação de *relacionamentos ad hoc*.

A seguir, apresenta-se a sintaxe explícita (*SQL-92*) para formular uma operação de junção interna.

```
SELECT lista_colunas  
FROM tabela1  
    [INNER] JOIN tabela2  
        ON condição_de_junção1  
    [[INNER] JOIN tabela3  
        ON condição_de_junção2]...  
[WHERE condições_de_pesquisa]  
[ORDER BY lista_de_ordenação]
```

A cláusula *SELECT* descreve as colunas que irão ser incluídas no resultado da consulta.

A cláusula *FROM* indica as tabelas que vão ser consultadas. São utilizadas as palavras-chave *INNER JOIN* e *ON* para especificar a operação de junção e a condição de junção, respectivamente. A palavra-chave *INNER* é opcional.

A cláusula *WHERE* é opcional e especifica as condições de pesquisa que têm que se verificar para um registo ser incluído no resultado da consulta. Note que ao utilizar a cláusula *WHERE* para definir as condições de pesquisa e ao usar a cláusula *FROM*, juntamente com as palavras-chave *INNER JOIN* e *ON*, para especificar as condições de junção, está a tornar mais fácil a leitura a consulta.

A cláusula *ORDER BY* é opcional e especifica o modo de ordenamento dos registos que constituem o resultado da consulta. Para definir uma ordenação decrescente, utilize a palavra-chave *DESC* a seguir ao nome da coluna pela qual quer ordenar os registos. Para definir uma ordenação crescente, pode utilizar a palavra-chave *ASC* a seguir ao nome da coluna pela qual quer

ordenar os registos. Todavia, uma vez que essa é a situação predefinida, não é necessário colocar a palavra-chave *ASC*.

Se as colunas usadas numa condição de junção tiverem o mesmo nome, tem que se identificar a tabela à qual cada coluna pertence. Para tal, utilize a sintaxe *nomeTabela.nomeColuna*.

Note que pode formular operações de junção utilizando uma notação implícita. Voltaremos, mais tarde, a este assunto.

EXEMPLOS

1 Formule, analise e execute as sete consultas a seguir apresentadas.

1.1 Consulta que devolve as facturas de cada fornecedor ordenadas pelo nome do fornecedor.

```
SELECT Nome AS [Nome do Fornecedor], NúmeroFactura AS [Número da Factura]
FROM Fornecedores INNER JOIN Facturas
ON Fornecedores.IDFornecedor = Facturas.Fornecedor
ORDER BY Nome
```

A seguir, apresenta-se o resultado obtido.

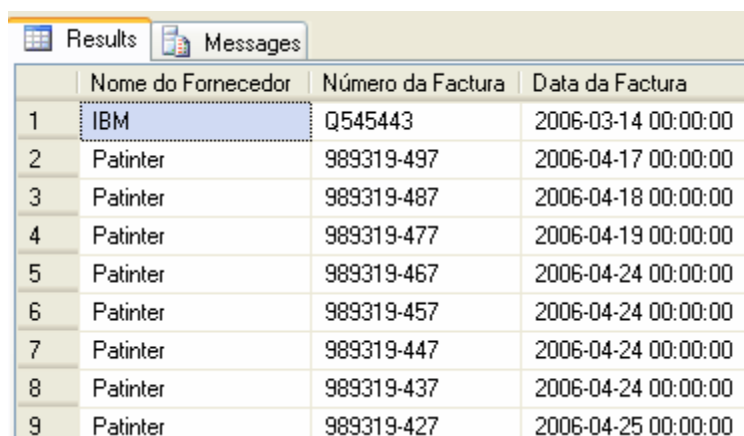
	Nome do Fornecedor	Número da Factura
1	Bell	MAB01489
2	Cartografia do Minho	P-0608
3	Cartografia do Minho	P-0259
4	FCA	C73-24
5	IBM	QP58872
6	IBM	Q545443
7	Mc Graw Hill	97/553B
8	Mc Graw Hill	97/553
9	Mc Graw Hill	97/522
10	Patinter	989319-497
11	Patinter	989319-487
12	Patinter	989319-477
13	Patinter	989319-467
14	Patinter	989319-457
15	Patinter	989319-447
16	Patinter	989319-437
17	Patinter	989319-427

Utilize a cláusula *WHERE* para acrescentar condições de pesquisa a uma consulta.

1.2 Consulta que devolve o nome dos fornecedores, o número e a data das facturas, dos meses de Março e Abril de 2006, dos fornecedores *Patinter* e *IBM*. Os registos do resultado vão ser ordenados por ordem crescente do nome do fornecedor.

```
SELECT Nome AS [Nome do Fornecedor], NúmeroFactura AS [Número da Factura],
       DataFactura AS [Data da Factura]
FROM Fornecedores INNER JOIN Facturas
     ON Fornecedores.IDFornecedor = Facturas.Fornecedor
WHERE Nome IN ('Patinter', 'IBM') AND
       DataFactura BETWEEN '2006-03-01' AND '2006-04-30'
ORDER BY Nome
```

A seguir, apresenta-se o resultado obtido.



	Nome do Fornecedor	Número da Factura	Data da Factura
1	IBM	Q545443	2006-03-14 00:00:00
2	Patinter	989319-497	2006-04-17 00:00:00
3	Patinter	989319-487	2006-04-18 00:00:00
4	Patinter	989319-477	2006-04-19 00:00:00
5	Patinter	989319-467	2006-04-24 00:00:00
6	Patinter	989319-457	2006-04-24 00:00:00
7	Patinter	989319-447	2006-04-24 00:00:00
8	Patinter	989319-437	2006-04-24 00:00:00
9	Patinter	989319-427	2006-04-25 00:00:00

Pode associar nomes temporários (*aliases*) às tabelas indicadas na cláusula *FROM*.

1.3 Consulta que devolve os detalhes das facturas dos meses de Março e Abril de 2006. O resultado vai ser ordenado por ordem crescente da data da factura.

```
SELECT NúmeroFactura, MontanteItem, DescriçãoItem AS [Descrição do Item]
FROM Facturas INNER JOIN ItensFactura As DetalhesFactura
     ON Facturas.IDFactura = DetalhesFactura.IDFactura
WHERE DataFactura BETWEEN '2006-03-01' AND '2006-04-30'
ORDER BY DataFactura
```

Note que ao definir um *alias* para uma tabela na cláusula *FROM*, deve usar sempre esse *alias* ao longo da consulta. Não pode utilizar o nome original da tabela. Apenas faz sentido utilizar um *alias* nesta situação se for para simplificar e clarificar a leitura da consulta.

A seguir, apresenta-se o resultado obtido.

	NúmeroFactura	MontanteItem	Descrição do Item
1	Q545443	1083,58	MSDN
2	P-0608	20551,18	CICS Parte 2
3	P-0259	26881,40	MVS JCL
4	MAB01489	936,93	Manutenção trimestral
5	989319-497	2312,20	Fretes
6	C73-24	600,00	Negócio publicitário
7	989319-487	1927,54	Fretes
8	989319-477	2184,11	Fretes
9	989319-467	2318,03	Fretes
10	989319-457	3813,33	Fretes
11	989319-447	3689,99	Fretes
12	989319-437	2765,36	Fretes
13	989319-427	2115,81	Fretes
14	97/522	1197,00	MC Pub
15	97/522	765,13	SCMD Flyer

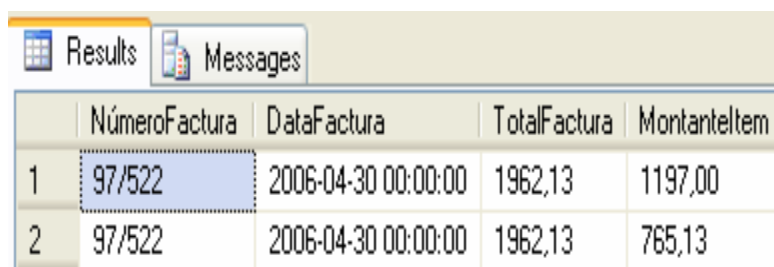
Também pode incluir duas ou mais condições de junção numa consulta, através a utilização dos operadores lógicos *AND* e *OR*. No exemplo seguinte, é formulada uma junção entre as tabelas *Facturas* e *ItensFactura*, baseada em duas condições.

1.4 Consulta que devolve as facturas que contêm mais do que dois itens.

```
SELECT NúmeroFactura, DataFactura, TotalFactura, MontanteItem
FROM Facturas INNER JOIN ItensFactura As DetalhesFactura
ON (Facturas.IDFactura = DetalhesFactura.IDFactura) AND
(Facturas.TotalFactura > DetalhesFactura.MontanteItem)
ORDER BY NúmeroFactura
```

Esta consulta utiliza duas condições de junção. A primeira compara a chave forasteira da tabela *ItensFactura* com a chave primária correspondente da tabela *Facturas*. A segunda compara os valores do campo *TotalFactura* da tabela *Facturas* com os valores do campo *MontanteItem* da tabela *ItensFactura*. Apenas vão ser considerados os casos em que o valor do campo *TotalFactura* é superior ao valor do campo *MontanteItem*. O que significa que apenas as facturas que têm dois ou mais itens irão ser incluídas no resultado.

A seguir, apresenta-se o resultado obtido.



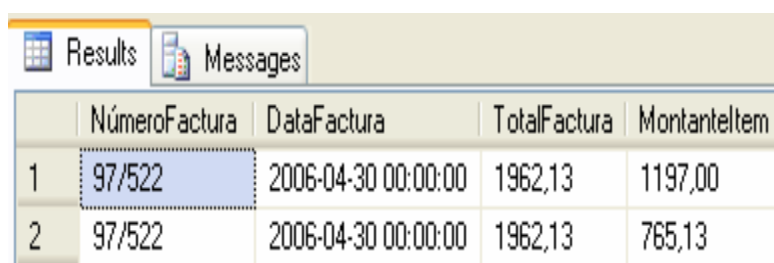
	NúmeroFactura	DataFactura	TotalFactura	MontanteItem
1	97/522	2006-04-30 00:00:00	1962,13	1197,00
2	97/522	2006-04-30 00:00:00	1962,13	765,13

Note que a consulta anterior poderia ser reformulada, de modo a incluir a comparação dos valores dos campos *TotalFactura* e *MontanteItem* numa cláusula *WHERE*.

1.5 Consulta que devolve as facturas que contêm mais do que dois itens.

```
SELECT NúmeroFactura, DataFactura, TotalFactura, MontanteItem
FROM Facturas INNER JOIN ItensFactura As DetalhesFactura
ON Facturas.IDFactura = DetalhesFactura.IDFactura
WHERE Facturas.TotalFactura > DetalhesFactura.MontanteItem
ORDER BY NúmeroFactura
```

O resultado vai ser idêntico ao da consulta 1.4.



	NúmeroFactura	DataFactura	TotalFactura	MontanteItem
1	97/522	2006-04-30 00:00:00	1962,13	1197,00
2	97/522	2006-04-30 00:00:00	1962,13	765,13

Até ao momento apenas formulámos consultas com junções entre duas tabelas. Contudo, o SQL SERVER permite-lhe formular junções entre até 256 tabelas. Note, porém, que as operações de junção consomem bastantes recursos. Consequentemente, deve limitar o número de junções a realizar, sempre que possível.

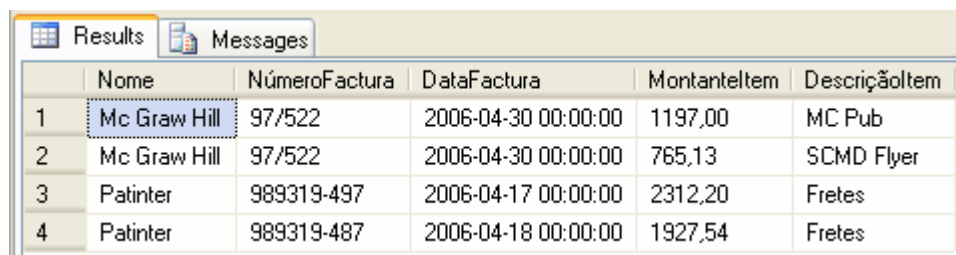
1.6 Consulta que contém operações de junção entre quatro tabelas: *Fornecedores*, *Facturas*, *ItensFactura* e *ContasBalanço*. Cada operação de junção é baseada no relacionamento entre a chave forasteira de uma tabela com uma chave primária de outra tabela. Por exemplo, o campo *ContaBalanço* é chave forasteira na tabela *ItensFactura* e o campo *NúmeroConta* é chave primária na tabela *ContasBalanço*.

```

SELECT Nome, NúmeroFactura, DataFactura, MontanteItem, DescriçãoItem
FROM Fornecedores INNER JOIN Facturas
  ON Fornecedores.IDFornecedor = Facturas.Fornecedor INNER JOIN ItensFactura
  ON Facturas.IDFactura = ItensFactura.IDFactura INNER JOIN ContasBalanço
  ON ItensFactura.ContaBalanço = ContasBalanço.NúmeroConta
WHERE TotalFactura - Pagamento - Crédito > 0
ORDER BY Nome, MontanteItem DESC

```

A seguir, apresenta-se o resultado obtido.



	Nome	NúmeroFactura	DataFactura	MontanteItem	DescriçãoItem
1	Mc Graw Hill	97/522	2006-04-30 00:00:00	1197,00	MC Pub
2	Mc Graw Hill	97/522	2006-04-30 00:00:00	765,13	SCMD Flyer
3	Patinter	989319-497	2006-04-17 00:00:00	2312,20	Fretes
4	Patinter	989319-487	2006-04-18 00:00:00	1927,54	Fretes

Para facilitar a compreensão da consulta anterior, considere que uma operação de junção entre várias tabelas pode ser vista como uma série de junções entre duas tabelas processadas da esquerda para a direita. É efectuada a junção entre as duas primeiras tabelas, resultando numa tabela temporária. Depois é realizada a operação de junção entre a tabela temporária e a próxima tabela, e assim sucessivamente. Por exemplo, no caso da consulta anterior, pode supor que é efectuada em primeiro lugar a junção entre a tabela *Fornecedores* e a tabela *Facturas*, resultando numa

tabela temporária. Depois é efectuada a junção entre a tabela *ItensFactura* e a tabela temporária resultante da primeira operação de junção. Finalmente, é efectuada a junção entre a tabela *ContasBalanço* e a tabela temporária resultante da segunda operação de junção.

Note, contudo, que as afirmações anteriores visam apenas a facilitação da compreensão da operação de junção entre várias tabelas. O SQL SERVER pode processar este tipo de junções exactamente do modo aqui explicitado.

Em vez de formular a condição de junção na cláusula FROM, também pode fazê-lo na cláusula WHERE da consulta, juntamente com as condições de pesquisa. Deste modo, apenas necessita de especificar, na cláusula FROM, separadas por vírgulas, as tabelas sobre as quais pretende efectuar a operação de junção. A esta sintaxe, que era usada anteriormente à norma SQL-92, dá-se comumente a designação de sintaxe implícita.

1.7 Por exemplo, a consulta 1.1 pode ser formulada, de forma implícita, do seguinte modo.

```
SELECT Nome AS [Nome do Fornecedor], NúmeroFactura AS [Número da Factura]
FROM Fornecedores, Facturas
WHERE Fornecedores.IDFornecedor = Facturas.Fornecedor
ORDER BY Nome
```

Naturalmente, o resultado será idêntico ao da consulta 1.1

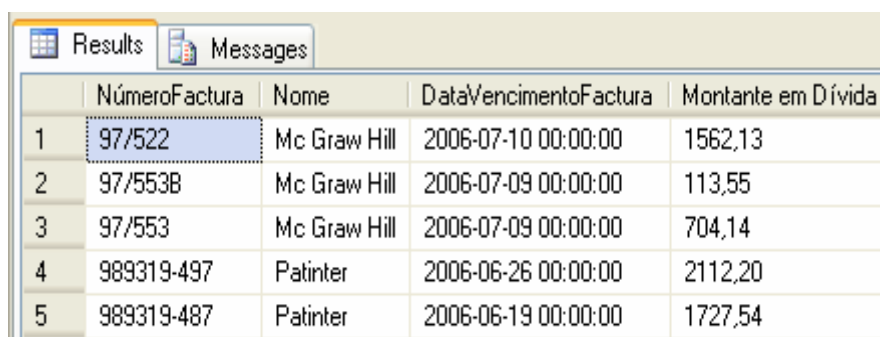
As consultas formuladas através da sintaxe explícita, para a definição de operações de junção, são, normalmente, mais fáceis de interpretar do que as consultas formuladas recorrendo à sintaxe implícita, pelo facto de a sintaxe explícita permitir a separação da definição das condições de junção das condições de pesquisa. Note que se omitir a cláusula *WHERE* na consulta anterior, irá ser processado o produto cartesiano (também conhecido por *cross join*) entre as tabelas enunciadas na cláusula *FROM* da consulta. Adiante tornaremos a este assunto.

QUESTÕES

2 Crie as seguintes consultas.

2.1 Crie uma consulta que devolva o nome do fornecedor, o número, a data de vencimento e o montante em dívida das facturas que ainda não estão completamente pagas. O resultado deve ser ordenado por ordem decrescente da data de vencimento das facturas.

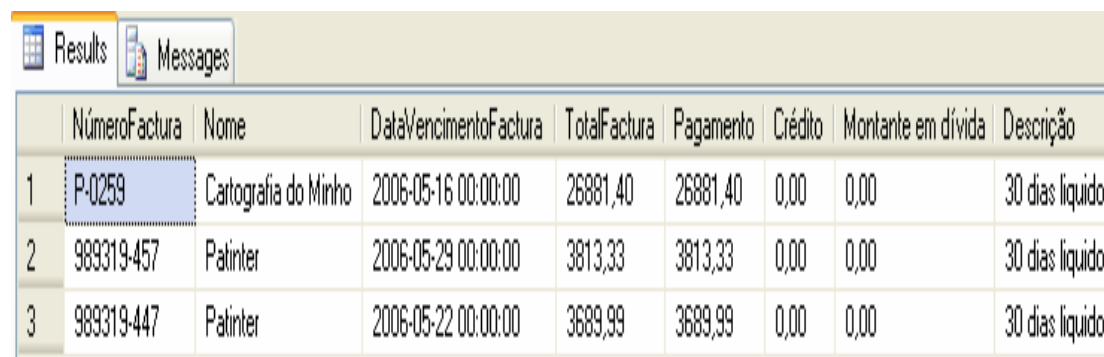
Resultado



	NúmeroFactura	Nome	DataVencimentoFactura	Montante em Dívida
1	97/522	Mc Graw Hill	2006-07-10 00:00:00	1562,13
2	97/553B	Mc Graw Hill	2006-07-09 00:00:00	113,55
3	97/553	Mc Graw Hill	2006-07-09 00:00:00	704,14
4	989319-497	Patinter	2006-06-26 00:00:00	2112,20
5	989319-487	Patinter	2006-06-19 00:00:00	1727,54

2.2 Crie uma consulta que devolva o nome do fornecedor, o número, a data de vencimento, o total, o montante já pago, o crédito, o montante em dívida e a descrição das condições de pagamento das três facturas mais dispendiosas, cujas condições de pagamento são de 30 dias líquido.

Resultado



	NúmeroFactura	Nome	DataVencimentoFactura	TotalFactura	Pagamento	Crédito	Montante em dívida	Descrição
1	P-0259	Cartografia do Minho	2006-05-16 00:00:00	26881,40	26881,40	0,00	0,00	30 dias liquido
2	989319-457	Patinter	2006-05-29 00:00:00	3813,33	3813,33	0,00	0,00	30 dias liquido
3	989319-447	Patinter	2006-05-22 00:00:00	3689,99	3689,99	0,00	0,00	30 dias liquido

2.3 Construa uma consulta semelhante à consulta anterior que aumente ainda o grau de detalhe, acrescentado a descrição dos itens que compõem cada factura. Neste caso, interessa considerar todas as facturas que têm como condições de pagamento 60 dias líquidos. O resultado deve ser ordenado pelo número das facturas.

Resultado

	NúmeroFactura	Nome	DescriçãoItem	DataVencimentoFactura	TotalFactura	Pagamento	Crédito	Montante em dívida	Descrição
1	977522	Mc Graw Hill	MC Pub	2006-07-10 00:00:00	1962,13	0,00	400,00	1562,13	60 dias liquido
2	977522	Mc Graw Hill	SCMD Flyer	2006-07-10 00:00:00	1962,13	0,00	400,00	1562,13	60 dias liquido
3	989319-467	Patinter	Fretes	2006-06-05 00:00:00	2318,03	2318,03	0,00	0,00	60 dias liquido
4	989319-477	Patinter	Fretes	2006-06-12 00:00:00	2184,11	2184,11	0,00	0,00	60 dias liquido
5	989319-487	Patinter	Fretes	2006-06-19 00:00:00	1927,54	0,00	200,00	1727,54	60 dias liquido
6	989319-497	Patinter	Fretes	2006-06-26 00:00:00	2312,20	0,00	200,00	2112,20	60 dias liquido
7	Q545443	IBM	MSDN	2006-05-23 00:00:00	1083,58	1083,58	0,00	0,00	60 dias liquido
8	QP58872	IBM	MVS Online Library	2006-04-22 00:00:00	116,54	116,54	0,00	0,00	60 dias liquido