

## *Tarefa Orientada 10*

### *Obter informação a partir de uma tabela*

#### Objectivos:

- Consultar dados de uma tabela
- Utilizar operadores aritméticos, relacionais, lógicos, de concatenação de cadeias de caracteres, LIKE e BETWEEN

### Consultar dados de uma tabela

A seguir, apresenta-se a sintaxe simplificada para consultar uma tabela.

*SELECT lista\_colunas*

*FROM tabela\_a\_consultar*

*[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 a tabela que vai ser consultada.

A cláusula *WHERE* é opcional e especifica as condições que têm que se verificar para um registo ser incluído no resultado da consulta. Se a combinação das condições de pesquisa for verdadeira, o registo que está a ser testado é incluído no resultado da consulta. Caso contrário, é omitido. Se omitir a cláusula *WHERE*, serão devolvidos todos os registos da tabela a consultar.

A cláusula *ORDER BY* é opcional e especifica o modo de ordenamento dos registos que constituem o resultado da consulta. Se for incluída uma cláusula *ORDER BY*, os registos que compõem o resultado são ordenados numa sequência especificada. Caso contrário, os registos são devolvidos pela ordem em que estão armazenados na tabela.

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*.

## EXEMPLOS

1 Formule, analise e execute as nove instruções *SELECT* a seguir apresentadas.

1.1 Instrução *SELECT* que devolve todos os registos da tabela Facturas.

```
SELECT *
FROM Facturas
```

O carácter \* pode ser utilizado para indicar que todas as colunas da tabela a consultar devem ser incluídas no resultado da consulta.

A seguir, apresenta-se o resultado obtido.

	IDFactura	Fornecedor	NúmeroFactura	DataFactura	TotalFactura	Pagamento	Crédito	CondiçãoPagamento	DataVencimentoFactura	DataPagamento
1	1	34	QP58872	2006-02-25 00:00:00	116,54	116,54	0,00	4	2006-04-22 00:00:00	2006-04-11 00:00:00
2	2	34	Q545443	2006-03-14 00:00:00	1083,58	1083,58	0,00	4	2006-05-23 00:00:00	2006-05-14 00:00:00
3	3	110	P-0608	2006-04-11 00:00:00	20551,18	19351,18	1400,00	5	2006-06-30 00:00:00	2006-08-01 00:00:00
4	4	110	P-0259	2006-04-16 00:00:00	26881,40	26881,40	0,00	3	2006-05-16 00:00:00	2006-05-12 00:00:00
5	5	81	MAB01489	2006-04-16 00:00:00	936,93	936,93	0,00	3	2006-05-16 00:00:00	2006-05-13 00:00:00
6	6	122	989319-497	2006-04-17 00:00:00	2312,20	0,00	200,00	4	2006-06-26 00:00:00	NULL
7	7	82	C73-24	2006-04-17 00:00:00	600,00	600,00	0,00	2	2006-05-10 00:00:00	2006-05-05 00:00:00
8	8	122	989319-487	2006-04-18 00:00:00	1927,54	0,00	200,00	4	2006-06-19 00:00:00	NULL
9	9	122	989319-477	2006-04-19 00:00:00	2184,11	2184,11	0,00	4	2006-06-12 00:00:00	2006-06-07 00:00:00
10	10	122	989319-467	2006-04-24 00:00:00	2318,03	2318,03	0,00	4	2006-06-05 00:00:00	2006-05-29 00:00:00
11	11	122	989319-457	2006-04-24 00:00:00	3813,33	3813,33	0,00	3	2006-05-29 00:00:00	2006-05-20 00:00:00
12	12	122	989319-447	2006-04-24 00:00:00	3689,99	3689,99	0,00	3	2006-05-22 00:00:00	2006-05-12 00:00:00
13	13	122	989319-437	2006-04-24 00:00:00	2765,36	2765,36	0,00	2	2006-05-15 00:00:00	2006-05-03 00:00:00
14	14	122	989319-427	2006-04-25 00:00:00	2115,81	2115,81	0,00	1	2006-05-08 00:00:00	2006-05-01 00:00:00
15	15	121	97/5538	2006-04-26 00:00:00	313,55	0,00	200,00	4	2006-07-09 00:00:00	NULL
16	18	121	97/553	2006-04-27 00:00:00	904,14	0,00	200,00	4	2006-07-09 00:00:00	NULL
17	19	121	97/522	2006-04-30 00:00:00	1962,13	0,00	400,00	4	2006-07-10 00:00:00	NULL

1.2 Instrução *SELECT* que devolve os valores das colunas *NúmeroFactura*, *DataFactura* e *TotalFactura* da tabela *Facturas*. Os registos do resultado vão ser ordenados por ordem crescente do total da factura.

```
SELECT NúmeroFactura, DataFactura, TotalFactura
FROM Facturas
ORDER BY TotalFactura
```

A seguir, apresenta-se o resultado obtido.

	NúmeroFactura	DataFactura	TotalFactura
1	QP58872	2006-02-25 00:00:00	116,54
2	97/553B	2006-04-26 00:00:00	313,55
3	C73-24	2006-04-17 00:00:00	600,00
4	97/553	2006-04-27 00:00:00	904,14
5	MAB01489	2006-04-16 00:00:00	936,93
6	Q545443	2006-03-14 00:00:00	1083,58
7	989319-487	2006-04-18 00:00:00	1927,54
8	97/522	2006-04-30 00:00:00	1962,13
9	989319-427	2006-04-25 00:00:00	2115,81
10	989319-477	2006-04-19 00:00:00	2184,11
11	989319-497	2006-04-17 00:00:00	2312,20
12	989319-467	2006-04-24 00:00:00	2318,03
13	989319-437	2006-04-24 00:00:00	2765,36
14	989319-447	2006-04-24 00:00:00	3689,99
15	989319-457	2006-04-24 00:00:00	3813,33
16	P-0608	2006-04-11 00:00:00	20551,18
17	P-0259	2006-04-16 00:00:00	26881,40

1.3 Instrução *SELECT* que devolve os valores das colunas *IDFactura*, *TotalFactura* e um campo calculado *CreditosTotais* do registo com o valor 5 no campo *IDFactura*.

```
SELECT IDFactura, TotalFactura, Crédito + Pagamento AS CreditosTotais
FROM Facturas
WHERE IDFactura = 5
```

Note que é utilizada a palavra-chave *AS* para definir um *alias* ou nome substituto para o campo calculado. Quando não se utiliza a palavra-chave *AS*, é dado, a cada coluna do resultado, o mesmo nome da coluna correspondente na tabela a consultar. Se pretender utilizar espaços e/ou caracteres especiais na definição de um *alias*, deve usar `[]`.

A seguir, apresenta-se o resultado obtido.

	IDFactura	TotalFactura	CreditosTotais
1	5	936,93	936,93

Se não definir um *alias* para o campo calculado da consulta anterior, esse campo irá aparecer no resultado sem nome.

	IDFactura	TotalFactura	(No column name)
1	5	936,93	936,93

A seguir apresentam-se, por ordem de precedência, os operadores aritméticos que pode utilizar para construir expressões aritméticas em campos calculados.

\* → Multiplicação

/ → Divisão

% → Resto da divisão inteira

+ → Adição

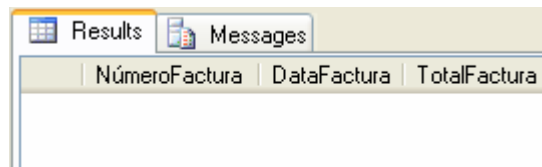
- → Subtração

Pode ainda utilizar parêntesis para modificar a ordem pela qual as operações são realizadas.

1.4 Instrução *SELECT* que devolve os valores das colunas *NúmeroFactura*, *DataFactura* e *TotalFactura*. O resultado desta consulta apenas vai incluir os registos da tabela *Facturas* que tenham um valor superior a 50000 no campo *TotalFactura*.

```
SELECT NúmeroFactura, DataFactura, TotalFactura
FROM Facturas
WHERE TotalFactura > 50000
```

Neste caso, o resultado vai ser um conjunto vazio, significando que não existem, na tabela *Facturas*, registos que verifiquem as condições de filtragem impostas na cláusula *WHERE* da consulta.




NúmeroFactura	DataFactura	TotalFactura
---------------	-------------	--------------

1.5 Instrução *SELECT* que concatena os valores dos campos *PNomeContacto* e *UNomeContacto*, da tabela *Fornecedores*, e atribui a designação *Nome* à coluna a apresentar no resultado. Os registos devolvidos pela consulta vão ser ordenados por ordem crescente da coluna *Nome*.

```
SELECT PNoneContacto + ' ' + ÚNomeContacto AS Nome
FROM Fornecedores
ORDER BY Nome
```

Pode utilizar o operador de concatenação + para combinar várias cadeias de caracteres. A seguir, apresenta-se o resultado obtido.



	Nome
1	Alexandre Farinha
2	Artur Silva
3	Bento Farinha
4	Fabio Ventura
5	João Castanheira
6	Joaquim Silva
7	Manuel Machado
8	Paulo Antunes
9	Paulo Bastos
10	Paulo Santos
11	Pedro Costa
12	Pedro Pais
13	Pedro Pinto
14	Sancho Pança
15	Sancho Pança
16	Sergio Larocas

1.6 Instrução *SELECT* que devolve os valores dos campos *NúmeroFactura*, *DataFactura* e *TotalFactura*. O resultado desta consulta apenas vai incluir os registos da tabela *Facturas* que tenham um valor inferior à data actual no campo *DataFactura*.

```
SELECT NúmeroFactura, DataFactura, TotalFactura
FROM Facturas
WHERE DataFactura < GETDATE()
```

A seguir, apresenta-se o resultado obtido.

	NúmeroFactura	DataFactura	TotalFactura
1	QP58872	2006-02-25 00:00:00	116,54
2	Q545443	2006-03-14 00:00:00	1083,58
3	P-0608	2006-04-11 00:00:00	20551,18
4	P-0259	2006-04-16 00:00:00	26881,40
5	MAB01489	2006-04-16 00:00:00	936,93
6	989319-497	2006-04-17 00:00:00	2312,20
7	C73-24	2006-04-17 00:00:00	600,00
8	989319-487	2006-04-18 00:00:00	1927,54
9	989319-477	2006-04-19 00:00:00	2184,11
10	989319-467	2006-04-24 00:00:00	2318,03
11	989319-457	2006-04-24 00:00:00	3813,33
12	989319-447	2006-04-24 00:00:00	3689,99
13	989319-437	2006-04-24 00:00:00	2765,36
14	989319-427	2006-04-25 00:00:00	2115,81
15	97/553B	2006-04-26 00:00:00	313,55
16	97/553	2006-04-27 00:00:00	904,14
17	97/522	2006-04-30 00:00:00	1962,13

Para formular condições mais complexas na cláusula *WHERE* pode utilizar os operadores relacionais, juntamente com os operadores lógicos.

A seguir apresentam-se os operadores relacionais disponíveis no *SQL SERVER*.

= → igual

> → maior que

< → menor que

>= → maior ou igual

<= → menor ou igual

<> → diferente

A seguir apresentam-se, por ordem de precedência, os operadores lógicos que pode utilizar para construir expressões compostas.

NOT → negação

AND → conjunção

OR → disjunção

A sintaxe da cláusula *WHERE* com operadores lógicos é a seguinte:

*WHERE [NOT] condição\_pesquisa1 {AND|OR} [NOT] condição\_pesquisa1 ...*

Utilize o operador *LIKE*, juntamente com caracteres especiais, na cláusula *WHERE* de uma consulta para obter os registos que verificam um determinado padrão. A seguir, apresentam-se alguns desses caracteres especiais.

% → qualquer conjunto de caracteres com zero ou mais elementos

- → qualquer carácter

[] → qualquer carácter incluído nos parêntesis rectos

[ - ] → qualquer carácter dentro de um dado intervalo

[ ^ ] → qualquer carácter não apresentado depois do sinal ^

Exemplos:

*WHERE Localidade LIKE 'Lis%' → Lisboa*

*WHERE Localidade LIKE 'L[A-F]%' → Leiria*

*WHERE CódigoPostal LIKE '[^1 - 3]%' → 2000-100; 5000-111*

Note que a utilização do operador LIKE degrada a *performance* da consulta quando comparada com outros tipos de pesquisa.

Pode utilizar a palavra-chave *DISTINCT* a seguir à palavra-chave *SELECT* para impedir a duplicação de registos no resultado da consulta. Essa inclusão também provoca que o resultado da consulta seja ordenado pela primeira coluna especificada na cláusula *SELECT*.

A utilização da palavra-chave *ALL* a seguir à palavra-chave *SELECT* especifica que todos os registos que verificam as condições de pesquisa sejam incluídos no resultado, independentemente de estarem em duplicado. Contudo, dado que esta é a situação predefinida, é uma prática comum omitir a palavra-chave *ALL*.

1.7 Por exemplo, a consulta seguinte devolve, sem registos duplicados, todas as localidades e os respectivos códigos postais registados na tabela *Fornecedores*.

```
SELECT DISTINCT Localidade, CódigoPostal  
FROM Fornecedores  
ORDER BY Localidade
```

Neste caso o resultado da execução da consulta contém 16 registos. Se omitir a palavra-chave *DISTINCT* irá obter 15 registos.

Além de poder eliminar os registos duplicados, também pode limitar o número de registos devolvidos por uma consulta. Para tal, utilize a palavra-chave *TOP* na cláusula *SELECT*. Quando usa esta palavra-chave, são devolvidos os primeiros *n* registos que verificam as condições de pesquisa da consulta.

1.8 Por exemplo, a consulta seguinte devolve o identificador do fornecedor e o total das 5 facturas com maior montante.

```
SELECT TOP 5 Fornecedor, TotalFactura  
FROM Facturas  
ORDER BY TotalFactura DESC
```

A seguir, apresenta-se o resultado obtido.

	Fornecedor	TotalFactura
1	110	26881,40
2	110	20551,18
3	122	3813,33
4	122	3689,99
5	122	2765,36

Se incluir a palavra-chave *PERCENT* a seguir à palavra-chave *TOP*, são devolvidos os primeiros *n%* dos registos que verificam as condições de pesquisa da consulta.

Por defeito, a palavra-chave *TOP* especifica que seja devolvido o número exacto de registos especificado. Todavia, se existirem mais registos com os mesmos valores do último registo devolvido, eles podem ser também incluídos no resultado da consulta se utilizar a palavra-chave *WITH TIES* juntamente com a palavra-chave *TOP*.

1.9 Por exemplo, a consulta seguinte devolve o identificador do fornecedor e a data da factura das cinco últimas facturas registadas na base de dados.

```
SELECT TOP 5 WITH TIES Fornecedor, DataFactura
FROM Facturas
ORDER BY DataFactura DESC
```

Todavia, dado que se incluiu a palavra-chave *WITH TIES*, vão ser devolvidos 8 registos.

	Fornecedor	DataFactura
1	121	2006-04-30 00:00:00
2	121	2006-04-27 00:00:00
3	121	2006-04-26 00:00:00
4	122	2006-04-25 00:00:00
5	122	2006-04-24 00:00:00
6	122	2006-04-24 00:00:00
7	122	2006-04-24 00:00:00
8	122	2006-04-24 00:00:00

Porém, se não incluir a palavra-chave *WITH TIES*, o resultado da consulta será apenas constituído pelos 5 primeiros registos.

Note que deve incluir a cláusula *ORDER BY* sempre utilizar a palavra-chave *TOP*. Caso contrário, os registos do resultado não iriam estar em nenhuma sequência particular.

## QUESTÕES

2 Crie as seguintes nove consultas.

2.1 Construa uma consulta que devolva o número da factura, a data da factura e o total da factura das facturas cuja data está entre 2006-04-01 e 2006-05-31. Pode utilizar o operador *BETWEEN* para testar se uma data está dentro de um intervalo de valores. A sintaxe deste operador é a seguinte:

*expressão\_de\_teste [NOT] BETWEEN expressão\_inicial AND expressão\_final*

Resultado

	NúmeroFactura	DataFactura	TotalFactura
1	P-0608	2006-04-11 00:00:00	20551,18
2	P-0259	2006-04-16 00:00:00	26881,40
3	MAB01489	2006-04-16 00:00:00	936,93
4	989319-497	2006-04-17 00:00:00	2312,20
5	C73-24	2006-04-17 00:00:00	600,00
6	989319-487	2006-04-18 00:00:00	1927,54
7	989319-477	2006-04-19 00:00:00	2184,11
8	989319-467	2006-04-24 00:00:00	2318,03
9	989319-457	2006-04-24 00:00:00	3813,33
10	989319-447	2006-04-24 00:00:00	3689,99
11	989319-437	2006-04-24 00:00:00	2765,36
12	989319-427	2006-04-25 00:00:00	2115,81
13	97/553B	2006-04-26 00:00:00	313,55
14	97/553	2006-04-27 00:00:00	904,14
15	97/522	2006-04-30 00:00:00	1962,13

2.2 Construa uma consulta que devolva uma coluna com nome e outra coluna com o endereço completo dos fornecedores registados na base de dados *Pagamentos*. Como auxílio, considere o seguinte exemplo de endereço completo: *Rua da Pomba, n.º 3 1000-001 Lisboa*. A coluna que vai conter o endereço completo deve ter a designação *Endereço Completo*.

### Resultado

	Nome	Endereço Completo
1	Portugal Telecom	Rua da Pomba, nº 3 1000-001 Lisboa
2	ONI Telecom	Rua da Jiboia, nº 1 1000-002 Lisboa
3	CTT	Rua das Laranjeiras, nº 1 1000-003 Lisboa
4	Bertrand	Rua do Petróleo, nº 1 1000-100 Lisboa
5	Porto Editora	Rua dos livros, nº 1 2000-100 Porto
6	WebBoom	Rua do Jardim da Celeste, nº 1 1000-100 Lisboa
7	Portugal Mail	Rua do Jardim da Joana, nº 1 3000-100 Coimbra
8	Vispapel	Rua do Jardim das Mães, nº 1 3500-100 Viseu
9	Visabeira	Rua do Jardim dasCarochas, nº 1 3500-010 Viseu
10	Construtora do Liz	Rua do Jardim das Margaridas, nº 1 3500-100 Leria
11	IBM	Rua do Jardim das Papoilas, nº 1 1000-111 Lisboa
12	Bell	Rua do Jardim das Patas, nº 1 1500-111 Setubal
13	FCA	Rua dos livros, nº 1 3500-003 Lisboa
14	Cartografia do Minho	Rua do Jardim das Lagartichas, nº 1 5000-111 Braga
15	Mc Graw Hill	Rua dos Piratas, nº 1 1000-007 Lisboa
16	Patinter	Rua do Sabão, nº 1 3500-123 Viseu

2.3 Construa uma consulta que devolva o primeiro e o último nome de contacto de cada um dos fornecedores registados na base de dados *Pagamentos*. A consulta deve ainda devolver, numa terceira coluna, as iniciais de cada fornecedor. Para formular esta consulta pode utilizar a função LEFT que tem a seguinte sintaxe:

*LEFT(parâmetro1, parâmetro2)*

*parâmetro1* - especifica uma cadeia de caracteres

*parâmetro2* - especifica o número de caracteres a devolver da cadeia de

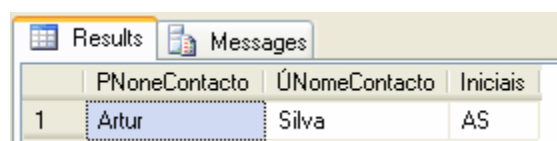
Resultado



	PNoneContacto	ÚNomeContacto	Iniciais
1	Manuel	Machado	MM
2	Pedro	Pinto	PP
3	João	Castanheira	JC
4	Sancho	Pança	SP
5	Pedro	Pais	PP
6	Sancho	Pança	SP
7	Sergio	Larocas	SL
8	Paulo	Antunes	PA
9	Paulo	Santos	PS
10	Joaquim	Silva	JS
11	Artur	Silva	AS
12	Alexandre	Farinha	AF
13	Pedro	Costa	PC
14	Bento	Farinha	BF
15	Paulo	Bastos	PB
16	Fabio	Ventura	FV

2.4 Altere a consulta anterior de modo a que sejam devolvidos apenas os registos dos fornecedores cujas iniciais são constituídas pelo carácter *A* seguido de uma qualquer carácter entre *J* e *S*.

Resultado



	PNoneContacto	ÚNomeContacto	Iniciais
1	Artur	Silva	AS

2.5 Construa uma consulta que devolva, para cada registo armazenado na tabela *Facturas*, uma coluna com a data das facturas, uma coluna com a data actual e uma coluna com a diferença entre as duas datas das outras duas colunas. Para formular esta consulta pode utilizar as funções *GETDATE* e *DATEDIFF*. Como sabe, a função *GETDATE* devolve a data actual. A função *DATEDIFF* permite calcular a diferença entre duas datas e tem a seguinte sintaxe:

*DATEDIFF* (*parâmetro1*, *parâmetro2*, *parâmetro3*)

*parâmetro1* - especifica as unidades em que o resultado da função vai ser apresentado

*parâmetro2* - especifica a data inicial

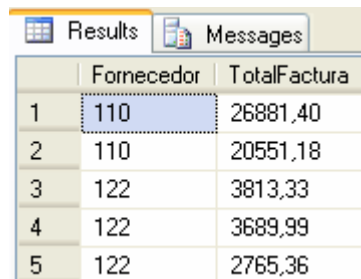
*parâmetro3* - especifica a data final

### Resultado

	DataFactura	Data actual	Idade, em dias, da factura
1	2006-02-25 00:00:00	2007-03-20 12:35:43.813	388
2	2006-03-14 00:00:00	2007-03-20 12:35:43.813	371
3	2006-04-11 00:00:00	2007-03-20 12:35:43.813	343
4	2006-04-16 00:00:00	2007-03-20 12:35:43.813	338
5	2006-04-16 00:00:00	2007-03-20 12:35:43.813	338
6	2006-04-17 00:00:00	2007-03-20 12:35:43.813	337
7	2006-04-17 00:00:00	2007-03-20 12:35:43.813	337
8	2006-04-18 00:00:00	2007-03-20 12:35:43.813	336
9	2006-04-19 00:00:00	2007-03-20 12:35:43.813	335
10	2006-04-24 00:00:00	2007-03-20 12:35:43.813	330
11	2006-04-24 00:00:00	2007-03-20 12:35:43.813	330
12	2006-04-24 00:00:00	2007-03-20 12:35:43.813	330
13	2006-04-24 00:00:00	2007-03-20 12:35:43.813	330
14	2006-04-25 00:00:00	2007-03-20 12:35:43.813	329
15	2006-04-26 00:00:00	2007-03-20 12:35:43.813	328
16	2006-04-27 00:00:00	2007-03-20 12:35:43.813	327
17	2006-04-30 00:00:00	2007-03-20 12:35:43.813	324

2.6 Construa uma consulta que devolva o identificador do fornecedor e o total da factura de exactamente 25% das facturas que totalizaram os maiores montantes.

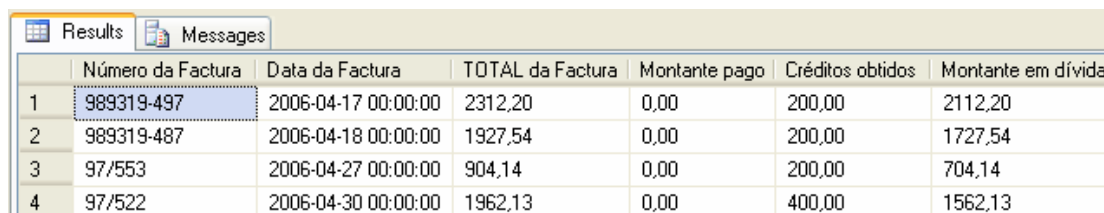
#### Resultado



	Fornecedor	TotalFactura
1	110	26881,40
2	110	20551,18
3	122	3813,33
4	122	3689,99
5	122	2765,36

2.7 Construa uma consulta que devolva o número da factura, o total da factura, o montante já pago da factura, o crédito da factura e o montante em dívida relativo à factura. Devem ser consideradas apenas as facturas com data inferior a 05-01-2006, cujo total é superior a 500 e cujo montante em dívida é superior a zero.

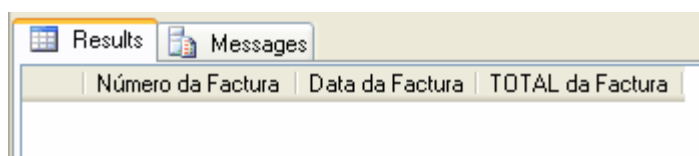
#### Resultado



	Número da Factura	Data da Factura	TOTAL da Factura	Montante pago	Créditos obtidos	Montante em dívida
1	989319-497	2006-04-17 00:00:00	2312,20	0,00	200,00	2112,20
2	989319-487	2006-04-18 00:00:00	1927,54	0,00	200,00	1727,54
3	97/553	2006-04-27 00:00:00	904,14	0,00	200,00	704,14
4	97/522	2006-04-30 00:00:00	1962,13	0,00	400,00	1562,13

2.8 Construa uma consulta que devolva o número da factura, a data da factura e o total da factura das facturas cuja data de vencimento se refere aos últimos 30 dias.

#### Resultado



	Número da Factura	Data da Factura	TOTAL da Factura
--	-------------------	-----------------	------------------

2.9 Construa uma consulta que devolva todos os dados das facturas cuja data de pagamento está definida. Para tal, pode utilizar a cláusula *IS NULL*. Esta cláusula permite verificar se uma determinada coluna

### Resultado

	IDFactura	Fornecedor	NúmeroFactura	DataFactura	TotalFactura	Pagamento	Crédito	CondiçãoPagamento	DataVencimentoFactura	DataPagamento
1	1	34	QP58872	2006-02-25 00:00:00	116,54	116,54	0,00	4	2006-04-22 00:00:00	2006-04-11 00:00:00
2	2	34	Q545443	2006-03-14 00:00:00	1083,58	1083,58	0,00	4	2006-05-23 00:00:00	2006-05-14 00:00:00
3	3	110	P-0608	2006-04-11 00:00:00	20551,18	19351,18	1400,00	5	2006-06-30 00:00:00	2006-08-01 00:00:00
4	4	110	P-0259	2006-04-16 00:00:00	26881,40	26881,40	0,00	3	2006-05-16 00:00:00	2006-05-12 00:00:00
5	5	81	MAB01489	2006-04-16 00:00:00	936,93	936,93	0,00	3	2006-05-16 00:00:00	2006-05-13 00:00:00
6	7	82	C73-24	2006-04-17 00:00:00	600,00	600,00	0,00	2	2006-05-10 00:00:00	2006-05-05 00:00:00
7	9	122	989319-477	2006-04-19 00:00:00	2184,11	2184,11	0,00	4	2006-06-12 00:00:00	2006-06-07 00:00:00
8	10	122	989319-467	2006-04-24 00:00:00	2318,03	2318,03	0,00	4	2006-06-05 00:00:00	2006-05-29 00:00:00
9	11	122	989319-457	2006-04-24 00:00:00	3813,33	3813,33	0,00	3	2006-05-29 00:00:00	2006-05-20 00:00:00
10	12	122	989319-447	2006-04-24 00:00:00	3689,99	3689,99	0,00	3	2006-05-22 00:00:00	2006-05-12 00:00:00
11	13	122	989319-437	2006-04-24 00:00:00	2765,36	2765,36	0,00	2	2006-05-15 00:00:00	2006-05-03 00:00:00
12	14	122	989319-427	2006-04-25 00:00:00	2115,81	2115,81	0,00	1	2006-05-08 00:00:00	2006-05-01 00:00:00

Neste momento, já podemos considerar uma sintaxe expandida para consultar uma tabela.

```

SELECT [ALL|DISTINCT] [TOP n [PERCENT] [WITH TIES]] coluna [[AS] alias]
        [,coluna [[AS] alias]] ...
FROM tabela_a_consultar
[WHERE condições_de_pesquisa]
[ORDER BY expressão [ASC|DESC] [, expressão [ASC|DESC]] ...]

```

Contudo, note que a sintaxe agora apresentada não inclui todas as cláusulas da instrução *SELECT*. Mais tarde voltaremos a este assunto.