

Tarefa Orientada 12

Junção Externa, Auto-Junção e União

Objectivos:

- Junção externa (*Outer JOIN*)
- Junção externa à esquerda (*LEFT Outer JOIN*)
- Junção externa à direita (*RIGHT Outer JOIN*)
- Junção externa completa (*FULL Outer JOIN*)
- Auto-Junção (*Self-JOIN*)
- União

Junção Externa

A operação de junção externa (*Outer JOIN*) é utilizada para combinar colunas de duas ou mais tabelas. O resultado de uma consulta que contém uma operação *Outer JOIN* entre duas tabelas inclui os registos que satisfazem a(s) condição(ões) de junção, mais os registos que não verificam as condições de junção de uma (*LEFT OUTER JOIN*, *RIGHT OUTER JOIN*) ou de ambas (*FULL OUTER JOIN*) as tabelas.

Quando se utiliza uma operação de junção externa à esquerda (*LEFT OUTER JOIN*), o resultado inclui todos os registos da primeira (esquerda) tabela. Analogamente, quando se usa uma operação de junção externa à direita (*RIGHT OUTER JOIN*), o resultado inclui todos os registos da segunda (direita) tabela. Quando se utiliza uma operação de junção externa completa (*FULL OUTER JOIN*), o resultado inclui todos os registos de ambas as tabelas.

Quando é devolvido um registo em que os valores das colunas utilizadas na condição de junção não correspondem, as colunas da outra tabela que são incluídas no resultado vão conter valores nulos.

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 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 externa.

```
SELECT lista_colunas  
FROM tabela1  
    {LEFT|RIGHT|FULL} [OUTER] JOIN tabela2  
        ON condição_de_junção1  
    {LEFT|RIGHT|FULL} [OUTER] JOIN tabela3  
        ON condição_de_junção2]...  
[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 *OUTER JOIN* e *ON* para especificar a operação de junção e a condição de junção, respectivamente. A palavra-chave *OUTER* é opcional. As palavras-chave *LEFT*, *RIGHT* e *FULL* permitem especificar o tipo de junção externa a realizar.

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*.

EXEMPLOS

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

1.1 Consulta que estabelece uma operação de junção externa à esquerda entre as tabelas *Fornecedores* e *Facturas*. Esta consulta devolve o nome de todos os fornecedores, o número e o total das respectivas facturas, mesmo para os fornecedores que não têm qualquer factura associada.

```
SELECT Nome, NúmeroFactura, TotalFactura
FROM Fornecedores LEFT OUTER JOIN Facturas
ON Fornecedores.IDFornecedor = Facturas.Fornecedor
ORDER BY Nome
```

A seguir, apresenta-se o resultado obtido.

	Nome	NúmeroFactura	TotalFactura
1	Bell	MAB01489	936,93
2	Bertrand	NULL	NULL
3	Cartografia do Minho	P-0608	20551,18
4	Cartografia do Minho	P-0259	26881,40
5	Construtora do Liz	NULL	NULL
6	CTT	NULL	NULL
7	FCA	C73-24	600,00
8	IBM	QP58872	116,54
9	IBM	Q545443	1083,58
10	Mc Graw Hill	97/553B	313,55
11	Mc Graw Hill	97/553	904,14
12	Mc Graw Hill	97/522	1962,13
13	ONI Telecom	NULL	NULL
14	Patinter	989319-497	2312,20
15	Patinter	989319-487	1927,54
16	Patinter	989319-477	2184,11
17	Patinter	989319-467	2318,03
18	Patinter	989319-457	3813,33
19	Patinter	989319-447	3689,99
20	Patinter	989319-437	2765,36
21	Patinter	989319-427	2115,81
22	Porto Editora	NULL	NULL
23	Portugal Mail	NULL	NULL
24	Portugal Telecom	NULL	NULL
25	Visabeira	NULL	NULL
26	Vispapel	NULL	NULL
27	WebBoom	NULL	NULL

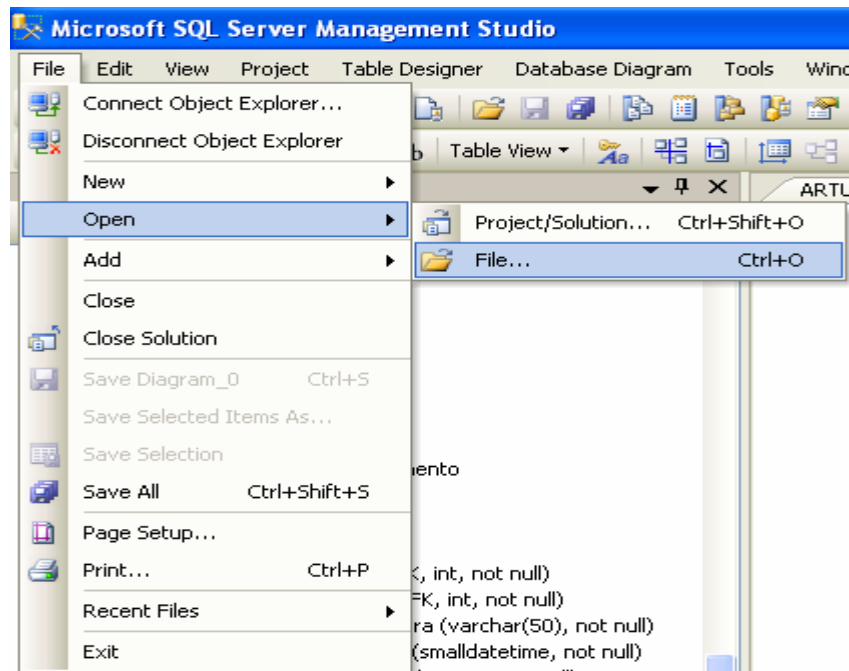
Note que o resultado desta consulta inclui todos os registos da tabela *Fornecedores*, mesmo que não tenham registos associados na tabela *Facturas*. Nesse caso, são devolvidos valores nulos para as colunas da tabela *Facturas* que são incluídas na cláusula *SELECT* da consulta.

- 1.2 Copie, para a sua área de trabalho, o *script CriarTabelasEmpregadosProjectosDepartamentos.sql* que se encontra na página da disciplina:
- 1.3 Inicie o SQL Server Management Studio, através do menu **INICIAR**.
- 1.4 Estabeleça a ligação ao servidor *diserver2*.
- 1.5 Expanda a pasta *Databases* e seleccione a base de dados que lhe está destinada.
- 1.6 Na barra de ferramentas do *SQL Server Management Studio*, pressione o botão *New Query*, para aceder ao editor de consultas.

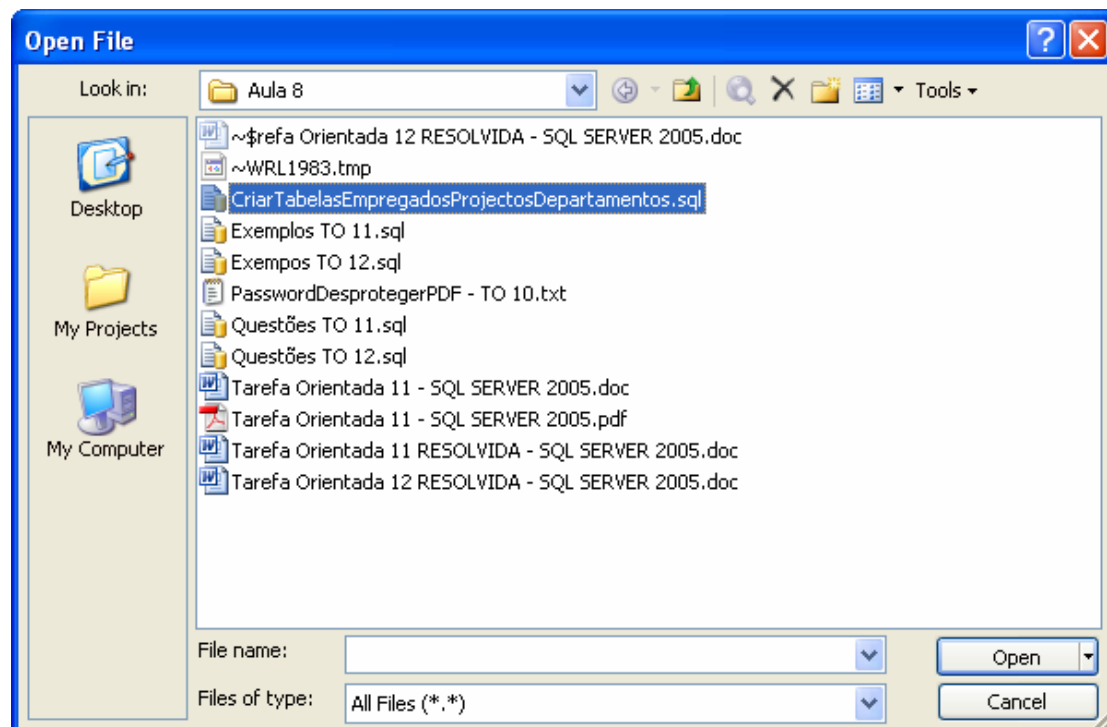


No lado direito do *SQL Server Management Studio* vai aparecer o editor de consultas

- 1.7 Execute o *script CriarTabelasEmpregadosProjectosDepartamentos.sql*. Para tal, aceda ao menu *File* e seleccione a opção *File* do comando *Open*.

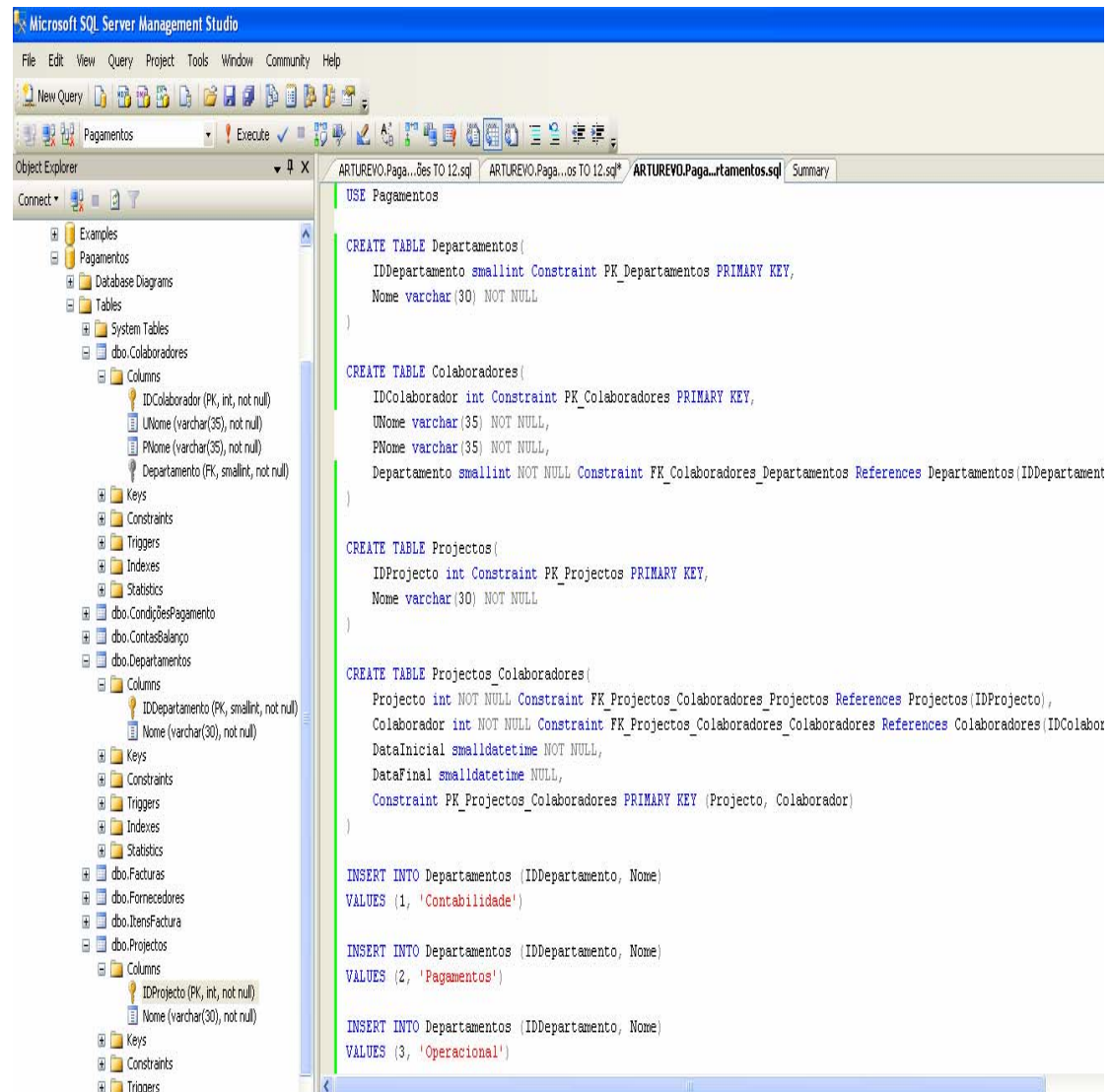


1.8 Depois, especifique o caminho para o local onde armazenou o ficheiro *CriarTabelasEmpregadosProjectosDepartamentos.sql*

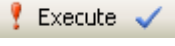
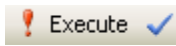


1.9 Pressione o botão *Open*.

Aparece, no lado direito do *SQL Server Management Studio*, o script que contém o código SQL (DDL) que permite criar as tabelas da base de dados *pagamentos*.



1.10 Analise o código.

1.11 Pode executar o *script* de uma só vez, colocando o cursor no final do código e pressionando o botão  da barra de ferramentas. Alternativamente, como ilustra a figura anterior, pode seleccionar o código relativo à criação de uma determinada tabela e pressionar o botão . Deste modo, pode criar uma tabela de cada vez.

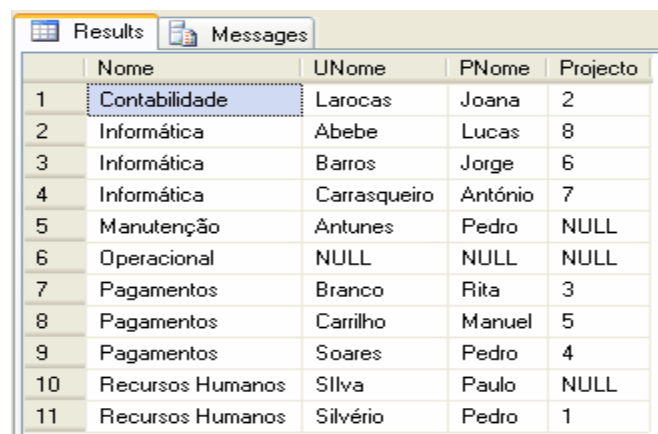
1.12 Crie um novo *Datagram* que contemple as tabelas acabadas de criar nos passos anteriores.

1.13 Consulta que estabelece operações de junção externa à esquerda entre as tabelas *Departamentos*, *Colaboradores* e *Projectos_Colaboradores*.

```
SELECT Departamentos.Nome, UNome, PNome, Projecto
FROM Departamentos LEFT OUTER JOIN Colaboradores
    ON Departamentos.IDDepartamento = Colaboradores.Departamento
    LEFT OUTER JOIN Projectos_Colaboradores
    ON Colaboradores.IDColaborador = Projectos_Colaboradores.Colaborador
ORDER BY Departamentos.Nome, UNome, Projecto
```

Analisando detalhadamente esta consulta, podemos verificar que a primeira operação de junção externa à esquerda permite obter os empregados de cada departamento e que a segunda operação de junção externa à esquerda devolve os projectos de cada empregado. Resumindo, esta consulta vai devolver os projectos em que os empregados de cada departamento estão envolvidos.

A seguir, apresenta-se o resultado obtido.



	Nome	UNome	PNome	Projecto
1	Contabilidade	Larocas	Joana	2
2	Informática	Abebe	Lucas	8
3	Informática	Barros	Jorge	6
4	Informática	Carrasqueiro	António	7
5	Manutenção	Antunes	Pedro	NULL
6	Operacional	NULL	NULL	NULL
7	Pagamentos	Branco	Rita	3
8	Pagamentos	Carrilho	Manuel	5
9	Pagamentos	Soares	Pedro	4
10	Recursos Humanos	Silva	Paulo	NULL
11	Recursos Humanos	Silvério	Pedro	1

Por exemplo, as linhas 5 e 10 do resultado indicam que os colaboradores *Pedro Antunes* e *Paulo Silva*, pertencentes, respectivamente, aos departamentos de *Contabilidade* e de *Recursos Humanos*, não estão, nem nunca estiveram, envolvidos em qualquer projecto. Do resultado desta consulta é ainda possível observar que o departamento *Operacional* não tem empregados associados.

1.14 Consulta que combina uma operação de junção interna com uma operação de junção externa à esquerda.

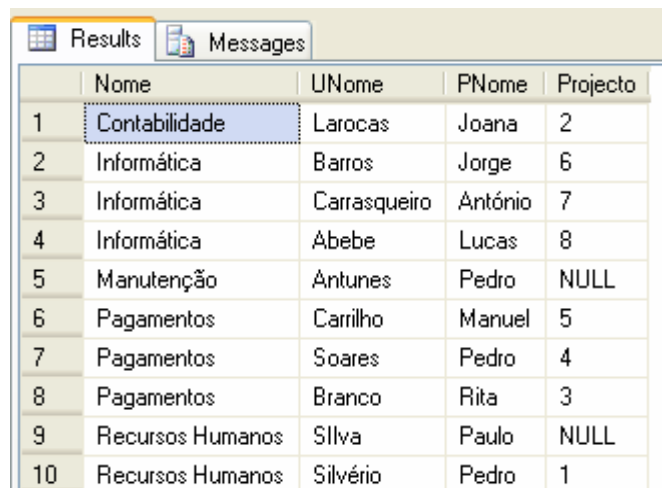
```

SELECT Departamentos.Nome, UNome, PNome, Projecto
FROM Departamentos INNER JOIN Colaboradores
  ON Departamentos.IDDepartamento = Colaboradores.Departamento
  LEFT OUTER JOIN Projectos_Colaboradores
    ON Colaboradores.IDColaborador = Projectos_Colaboradores.Colaborador
ORDER BY Departamentos.Nome

```

Analisando detalhadamente esta consulta, podemos verificar que a operação de junção interna permite obter os empregados de cada departamento, mas apenas são considerados os departamentos que têm empregados associados. A operação de junção externa à esquerda devolve os projectos de cada empregado.

A seguir, apresenta-se o resultado obtido.



	Nome	UNome	PNome	Projecto
1	Contabilidade	Larocas	Joana	2
2	Informática	Barros	Jorge	6
3	Informática	Carrasqueiro	António	7
4	Informática	Abebe	Lucas	8
5	Manutenção	Antunes	Pedro	NULL
6	Pagamentos	Carrilho	Manuel	5
7	Pagamentos	Soares	Pedro	4
8	Pagamentos	Branco	Rita	3
9	Recursos Humanos	Silva	Paulo	NULL
10	Recursos Humanos	Silvério	Pedro	1

Se comparar o resultado obtido após a execução da consulta anterior, pode verificar que este é muito semelhante ao resultado devolvido pela consulta 1.13. A principal diferença reside no facto de que na consulta 1.14 já não são incluídos os departamentos que não têm empregados associados.

Note que pode converter uma operação de junção externa à direita por uma operação de junção externa à esquerda se inverter a ordem pela qual as tabelas são indicadas na cláusula *FROM* e se utilizar a palavra-chave *LEFT*, em vez da palavra-chave *RIGHT*.

Auto-Junção

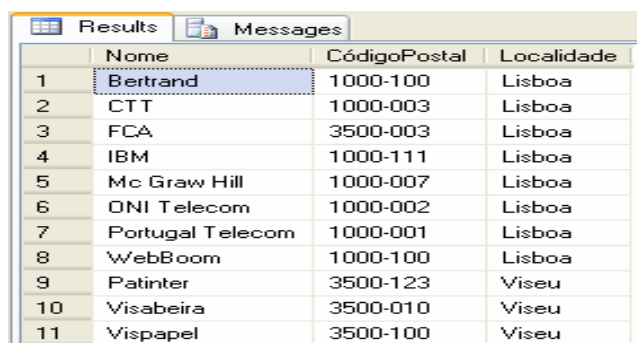
Uma auto-junção (*Self-Join*) consiste numa operação de junção de uma tabela consigo própria. Apesar das auto-junções se utilizarem raramente, existem algumas situações que podem ser resolvidas de forma expedita através do uso de operações de auto-junção.

1.15 Consulta que devolve os fornecedores de localidades comuns.

```
SELECT DISTINCT Fornecedores1.Nome, Fornecedores1.CódigoPostal,
                Fornecedores1.Localidade
FROM Fornecedores AS Fornecedores1 JOIN Fornecedores AS Fornecedores2
    ON (Fornecedores1.Localidade = Fornecedores2.Localidade) AND
        (Fornecedores1.IDFornecedor <> Fornecedores2.IDFornecedor)
ORDER BY Fornecedores1.Localidade
```

A condição de junção da consulta anterior utiliza duas comparações. A primeira compara os valores da coluna *Localidade*, de modo a que sejam devolvidos os registos dos fornecedores com localidades comuns. Dado que um fornecedor reside na mesma localidade que si próprio, é utilizada uma segunda comparação para excluir esses registos, através da utilização do operador <>.

Note que quando formula uma consulta que realiza uma operação de auto-junção, deve usar nomes de correlação para as tabelas e deve preceder a designação das colunas com esses nomes de correlação. As consultas que realizam operações de auto-junção incluem frequentemente a palavra-chave *DISTINCT*, de modo a eliminar os registos duplicados. A seguir, apresenta-se o resultado obtido.



	Nome	CódigoPostal	Localidade
1	Bertrand	1000-100	Lisboa
2	CTT	1000-003	Lisboa
3	FCA	3500-003	Lisboa
4	IBM	1000-111	Lisboa
5	Mc Graw Hill	1000-007	Lisboa
6	ONI Telecom	1000-002	Lisboa
7	Portugal Telecom	1000-001	Lisboa
8	WebBoom	1000-100	Lisboa
9	Patinter	3500-123	Viseu
10	Visabeira	3500-010	Viseu
11	Vispapel	3500-100	Viseu

União

Tal como a operação de junção, a operação de união combina dados de duas ou mais tabelas. Contudo, em vez de combinar as colunas das tabelas fonte, a operação de união combina registos de dois ou mais resultados de instruções *SELECT*. Cada resultado deve devolver o mesmo número de campos e as colunas correspondentes em cada resultado devem ter associados tipos de dados compatíveis. A seguir, apresenta-se a sintaxe para a operação de união.

```
Instrução_SELECT_1
UNION [ALL]
    Instrução_SELECT_2
[UNION [ALL]
    Instrução_SELECT_3] ...
[ORDER BY lista_colunas_ Instrução_SELECT_1]
```

De modo predefinido, numa operação de união são eliminados os registos duplicados. Todavia, se pretender contemplar registos duplicados, inclua a palavra-chave *ALL* a seguir ao operador *UNION*.

Os nomes das colunas do resultado de uma operação de união são formados a partir dos nomes das colunas estabelecidos na primeira cláusula *SELECT*. Os nomes das colunas, ou de eventuais *aliases*, da(s) outra(s) instrução(ões) de *SELECT* não produzem qualquer efeito no resultado final.

Para ordenar os registos do resultado final, utilize a cláusula *ORDER BY* depois da última instrução de *SELECT*. Note que a cláusula *ORDER BY* deve referir-se aos nomes das colunas definidos na primeira instrução de *SELECT*.

1.16 Consulta que combina dados de duas tabelas. Nesta consulta, cada instrução *SELECT* junta dados das tabelas Facturas e Fornecedores. A primeira instrução *SELECT* devolve as facturas cujo total é superior a 10000 euros e calcula o desconto de 25 % e o pagamento a efectuar. Os restantes comandos *SELECT* são semelhantes. O segundo devolve as facturas cujo total está entre 500 e 10000 euros e calcula 50% de desconto e o pagamento a efectuar. A terceira instrução *SELECT* devolve as facturas cujo total é inferior a 500 euros e não aplica nenhum desconto.

```

SELECT NúmeroFactura, Nome, 'Desconto 25%' AS [Tipo de Desconto],
       TotalFactura AS Total, TotalFactura * 0.25 AS Desconto,
       TotalFactura - (TotalFactura * 0.25) AS Pagamento
FROM Facturas JOIN Fornecedores
       ON Facturas.Fornecedor = Fornecedores.IDFornecedor
WHERE TotalFactura > 10000
UNION
SELECT NúmeroFactura, Nome, 'Desconto 50%' AS [Tipo de Desconto],
       TotalFactura AS Total, TotalFactura * 0.5 AS Desconto,
       (TotalFactura * (1 - 0.5)) AS Pagamento
FROM Facturas JOIN Fornecedores
       ON Facturas.Fornecedor = Fornecedores.IDFornecedor
WHERE TotalFactura BETWEEN 500 AND 10000
UNION
SELECT NúmeroFactura, Nome, 'Sem Desconto' AS [Tipo de Desconto],
       TotalFactura AS Total, TotalFactura * 0 AS Desconto,
       TotalFactura AS Pagamento
FROM Facturas JOIN Fornecedores
       ON Facturas.Fornecedor = Fornecedores.IDFornecedor
WHERE TotalFactura < 500
ORDER BY [Tipo de Desconto], Nome, NúmeroFactura

```

Para ordenar os registos do resultado final, utiliza-se a cláusula *ORDER BY* depois da última instrução de *SELECT*. Note que a cláusula *ORDER BY* se refere aos nomes das colunas definidos na primeira instrução de *SELECT*.

Após observar o resultado da execução da consulta, apresentado a seguir, note que os nomes das colunas do resultado de uma operação de união são formados a partir dos nomes das colunas estabelecidos na primeira cláusula *SELECT*. Os nomes das colunas, ou de eventuais *aliases*, da(s) outra(s) instrução(ões) de *SELECT* não produzem qualquer efeito no resultado final.

A seguir, apresenta-se o resultado obtido.

	NúmeroFactura	Nome	Tipo de Desconto	Total	Desconto	Pagamento
1	P-0259	Cartografia do Minho	Desconto 25%	26881,40	6720.350000	20161.050000
2	P-0608	Cartografia do Minho	Desconto 25%	20551,18	5137.795000	15413.385000
3	MAB01489	Bell	Desconto 50%	936,93	468.465000	468.465000
4	C73-24	FCA	Desconto 50%	600,00	300.000000	300.000000
5	Q545443	IBM	Desconto 50%	1083,58	541.790000	541.790000
6	97/522	Mc Graw Hill	Desconto 50%	1962,13	981.065000	981.065000
7	97/553	Mc Graw Hill	Desconto 50%	904,14	452.070000	452.070000
8	989319-427	Patinter	Desconto 50%	2115,81	1057.905000	1057.905000
9	989319-437	Patinter	Desconto 50%	2765,36	1382.680000	1382.680000
10	989319-447	Patinter	Desconto 50%	3689,99	1844.995000	1844.995000
11	989319-457	Patinter	Desconto 50%	3813,33	1906.665000	1906.665000
12	989319-467	Patinter	Desconto 50%	2318,03	1159.015000	1159.015000
13	989319-477	Patinter	Desconto 50%	2184,11	1092.055000	1092.055000
14	989319-487	Patinter	Desconto 50%	1927,54	963.770000	963.770000
15	989319-497	Patinter	Desconto 50%	2312,20	1156.100000	1156.100000
16	QP58872	IBM	Sem Desconto	116,54	0.000000	116.540000
17	97/553B	Mc Graw Hill	Sem Desconto	313,55	0.000000	313.550000

QUESTÕES

2 Crie as seguintes três consultas.

2.1 Crie uma consulta que aumente o grau de informação relativamente à consulta 1.12, de modo a incluir também no resultado a designação dos projectos em que os empregados de cada departamento estão envolvidos

Resultado



	Nome	UNome	PNome	IDProjecto	Nome
1	Contabilidade	Larocas	Joana	2	Gestão Salários 2007
2	Informática	Abebe	Lucas	8	LAN Empresa 2007
3	Informática	Barros	Jorge	6	BD Gestão Stocks
4	Informática	Carrasqueiro	António	7	BD Gestão Pagamentos
5	Manutenção	Antunes	Pedro	NULL	NULL
6	Operacional	NULL	NULL	NULL	NULL
7	Pagamentos	Branco	Rita	3	Gestão Pagamentos 2006
8	Pagamentos	Carrilho	Manuel	5	Gestão Facturas 2006
9	Pagamentos	Soares	Pedro	4	Gestão Pagamentos 2007
10	Recursos Humanos	Silva	Paulo	NULL	NULL
11	Recursos Humanos	Silvério	Pedro	1	Férias 2007

2.2 Crie uma consulta que devolva o nome do departamento, o último e o primeiro nome do colaborador, o identificador do projecto e a descrição dos projectos em que estão envolvidos os colaboradores dos departamentos de *Contabilidade* e de *Pagamentos*.

Resultado

	Nome	UNome	PNome	IDProjecto	Nome do Projecto
1	Contabilidade	Larocas	Joana	2	Gestão Salários 2007
2	Pagamentos	Branco	Rita	3	Gestão Pagamentos 2006
3	Pagamentos	Carrilho	Manuel	5	Gestão Facturas 2006
4	Pagamentos	Soares	Pedro	4	Gestão Pagamentos 2007

- 2.3 Crie uma consulta que devolva o número, a data e o total das facturas registadas na base de dados. O resultado deve ser ordenado por ordem decrescente do total da factura e deve ainda incluir uma coluna *Estado* que indique, para cada registo, se a factura já está totalmente paga ou se ainda está por pagar.

Resultado

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