

Título	<b>Ficha SQL 3 – Junções de tabelas</b>
Contexto	<b>Aulas práticas</b>
Capítulo	<b>Capítulo 2 – Manipulação de dados</b>
Conteúdos abordados	<ul style="list-style-type: none"> <li>▪ Equi-junções</li> <li>▪ Junções externas</li> <li>▪ Junções verticais</li> </ul>

A utilização destes exercícios pressupõe a preparação da arquitetura e cenário definidos na ficha *SQL 1 - Introdução à linguagem SQL*.

Nos exercícios seguintes, utilizar comandos da linguagem SQL para apresentar a informação solicitada, respeitando ao máximo os *outputs* apresentados.

### Equi-junções

Tipicamente, quando na mesma *query* é necessário aceder a dados presentes em mais que uma tabela, é necessário indicar ao servidor de bases de dados como as linhas de uma tabela se relacionam com as linhas da outra (geralmente, usando as suas chaves primária e estrangeira).

Supondo que, com as tabelas seguintes, pretendemos descobrir o id/nome de cada estudante e ainda o nome da cidade de cada um. Para isso, será necessário dizer ao servidor de bases de dados:

STUDENT		
<b>id</b>	<b>name</b>	<b>city_id</b>
2080918	António Freitas	1
2080919	Rita Marujo	2
2080940	Carlos da Silva	3
2080943	Carla Batista	1
2080954	Ana Oliveira	4
2080957	Ana Oliveira	3
...	...	...

CITY		
<b>id</b>	<b>name</b>	<b>region</b>
1	Leiria	Center
2	Lisboa	South
3	Coimbra	Center
4	Porto	North

**1.º** Quais as tabelas onde estão os dados necessários.

**2.º** Como se descobre o nome de uma cidade a partir dos dados do estudante.

**3.º** Quais as colunas que se pretende mostrar no resultado.

Ou seja, neste caso concreto, a consulta seria a seguinte:

The diagram shows the following SQL query with annotations:

```
SELECT student.id AS student_id,
       student.name AS student_name
       city.name AS city_name
  FROM students
 JOIN city
    ON student.city_id = city.id;
```

- 1.º quais as tabelas necessárias
- 2.º a condição lógica que define a forma de relacionar as linhas das tabelas (chamada *condição de junção*)
- 3.º as colunas a apresentar

Portanto, é necessária uma operação de junção entre as tabelas STUDENT e CITY. Como a condição de junção utiliza uma igualdade (símbolo `=`), trata-se de uma equi-junção.

O resultado da *query* acima apresentada é o seguinte:

student_id	student_name	city_name
2080918	António Freitas	Leiria
2080919	Rita Marujo	Lisboa
2080940	Carlos da Silva	Coimbra
2080943	Carla Batista	Leiria
2080954	Ana Oliveira	Porto
2080957	Ana Oliveira	Coimbra
...	...	...

1. Relativamente aos produtos com iva a 23%:

- a) Listá-los, com nome e código da categoria, ordenados por nome.

PRODUTO_NOME	PRODUTO_CAT_ID
Monitor 24" Philips 243V	6
Monitor 27" Gaming E-Line 272E1GAJ	6
PEN 32 GB	20
PEN 64 GB	20
Pilhas AAA	4

output

- b) Listá-los, novamente. Desta vez, mostre também o id e nome da categoria de cada produto.

PRODUTO_NOME	PRODUTO_CAT_ID	CATEGORIA_ID	CATEGORIA_NOME
Monitor 24" Philips 243V	6	6	PERIFÉRICOS
Monitor 27" Gaming E-Line 272E1GAJ	6	6	PERIFÉRICOS
PEN 32 GB	20	20	ARMAZENAMENTO
PEN 64 GB	20	20	ARMAZENAMENTO
Pilhas AAA	4	4	MERCEARIA

dados vindos da  
tabela PRODUTO

dados vindos da  
tabela CATEGORIA

2. Relativamente à loja *online*:

- a) Mostrar o seu identificador e nome e ainda o identificador do seu concelho.

LOJA_ID	LOJA_NOME	LOJA_CONCELHO_ID
0	online	1312

output

- b) Mostrar os mesmos dados, mas acrescentando o nome do concelho.

LOJA_ID	LOJA_NOME	LOJA_CONCELHO_ID	CONCELHO_NOME
0	online	1312	PORTO

output

- c) Mostrar os mesmos dados, mas acrescentando o nome do distrito.

LOJA_ID	LOJA_NOME	LOJA_CONCELHO_ID	CONCELHO_NOME	DISTRITO_NOME
0	online	1312	PORTO	Porto

output

**3.** Apresentar todos os produtos das categorias PERIFÉRICOS e ARMAZENAMENTO.

			<i>output</i>
◊ PRODUTO_ID	◊ PRODUTO_NOME	◊ CATEGORIA_NOME	Y
1104	Monitor 24" Philips 243V	PERIFÉRICOS	
1109	Monitor 27" Gaming E-Line 272E1GAJ	PERIFÉRICOS	
1060	PEN 32 GB	ARMAZENAMENTO	
1058	PEN 64 GB	ARMAZENAMENTO	

**4.** Mostrar o nome dos produtos vendidos no dia *1 de julho de 2021*, na loja *MODELO EIRAS*.

Sugestão de resolução:

- 1º. Obter as vendas do dia 1 de julho de 2021
- 2º. Obter, dessas vendas, só as que foram feitas na loja MODELO EIRAS
- 3º. Com a venda que resta, obter o identificador das linhas de venda associadas
- 4º. A partir das linhas de venda, obter os nomes dos produtos vendidos

				<i>output</i>
◊ ID	◊ LOJA_NOME	◊ PRODUTO_ID	◊ PRODUTO_NOME	◊ UNIDADES
1459	Modelo Eiras	1060	PEN 32 GB	1
1459	Modelo Eiras	2040	Pilhas AAA	3

## Junções externas

Nalgumas situações envolvendo duas tabelas, pode ser necessário obter as linhas que resultam da junção **E, ADICIONALMENTE**, obter ainda as linhas para as quais não há correspondência na junção. Nesses casos, pode recorrer-se às **junções externas**.

5. Considerar apenas as vendas feitas em junho de 2021.

a) Mostrar essas vendas.

VENDA_ID	MÊS
1450	2021/JUNE
1451	2021/JUNE

b) Mostrar os *ids* dos produtos vendidos nessas vendas. Por que desaparece a venda 1451?

VENDA_ID	MÊS	PRODUTO_ID
1450	2021/JUNE	1104
1450	2021/JUNE	1058
1450	2021/JUNE	1060

vendas de junho  
COM produtos.  
A venda 1451 desapareceu

c) Como pode forçar-se que surja a venda 1451 no resultado, mesmo não tendo produtos vendidos?

VENDA_ID	MÊS	PRODUTO_ID
1450	2021/JUNE	1104
1450	2021/JUNE	1058
1450	2021/JUNE	1060
1451	2021/JUNE	

vendas de junho  
COM produtos  
vendas de junho  
SEM produtos (extra)

6. Apresentar os produtos associados às categorias CEREAIS, FRUTAS E LEGUMES e FRUTOS SECOS. Estas categorias devem ser mostradas mesmo que não tenham produtos.

CATEG_NOME	PROD_NOME
CEREAIS	
FRUTAS E LEGUMES	Maçã Fuji
FRUTAS E LEGUMES	Maçã Fuji
FRUTOS SECOS	Caju a GRANEL

Categoria  
SEM produtos  
Categorias e  
seus produtos

## Junções verticais: operações com conjuntos

A linguagem SQL possui operadores para realizar as clássicas operações de conjuntos (*sets*) : **interseção** (operador *intersect*) , a **união** (operador *union*) e a **diferença** (operador *minus*).

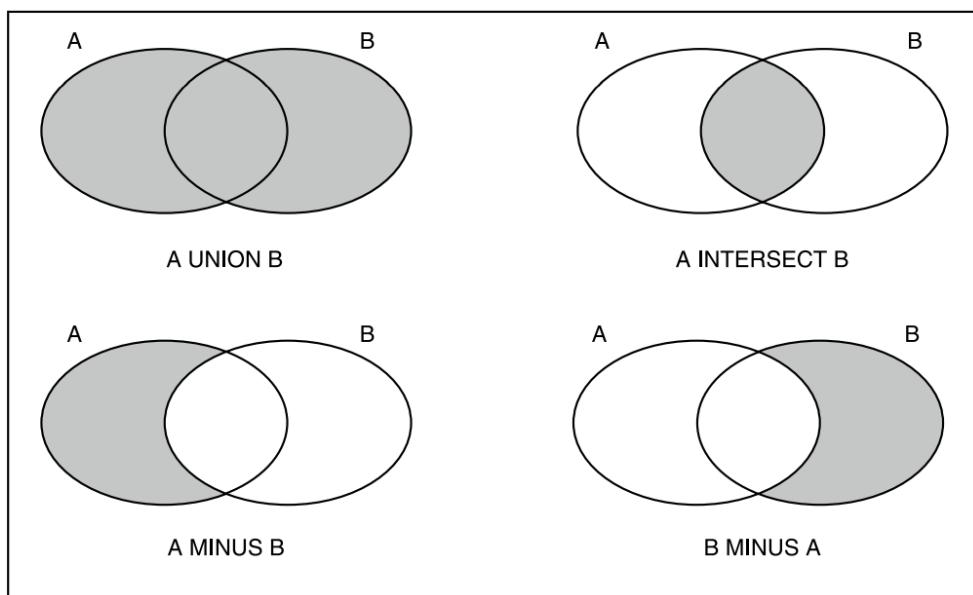


Figura 1 – Representação visual das operações clássicas de conjuntos com a linguagem SQL  
(retirado de *Beginning Oracle SQL*. Haan, L. et all, página 14)

A sintaxe utilizada para trabalhar com estes operadores é a seguinte:

```
<query A>
{UNION | UNION ALL | MINUS | INTERSECT}
<query B>;
```

Algumas notas sobre as *queries* usadas com estes operadores:

- Qualquer ordenação que se pretenda realizar, tem de ser realizada na <*query B*>
- Ambas as *queries* têm de devolver o mesmo número de colunas
- As colunas na mesma posição em ambas as *queries* têm de ter tipos de dados compatíveis
- Os títulos definitivos das colunas são definidos na <*query A*>

7. Na base de dados estão registados concelhos para os quais não há lojas.

- a) Obter o nome dos concelhos existentes (**query A**).

CONCELHO_NOME	output
BATALHA	
CONDEIXA-A-NOVA	
COIMBRA	
LEIRIA	
MARINHA GRANDE	
MIRANDA DO CORVO	
PORTO	

- b) Obter o nome dos concelhos onde há lojas (**query B**).

CONCELHO_NOME	output
COIMBRA	
COIMBRA	
LEIRIA	
LEIRIA	
PORTO	

- c) Utilizar as *queries A* e *B* para obter o nome dos concelhos onde não há lojas.

<table border="1"><thead><tr><th>CONCELHO_NOME</th></tr></thead><tbody><tr><td>BATALHA</td></tr><tr><td>CONDEIXA-A-NOVA</td></tr><tr><td>COIMBRA</td></tr><tr><td>LEIRIA</td></tr><tr><td>MARINHA GRANDE</td></tr><tr><td>MIRANDA DO CORVO</td></tr><tr><td>PORTO</td></tr></tbody></table> <p><i>output(query A)</i></p>	CONCELHO_NOME	BATALHA	CONDEIXA-A-NOVA	COIMBRA	LEIRIA	MARINHA GRANDE	MIRANDA DO CORVO	PORTO	<table border="1"><thead><tr><th>CONCELHO_NOME</th></tr></thead><tbody><tr><td>COIMBRA</td></tr><tr><td>COIMBRA</td></tr><tr><td>LEIRIA</td></tr><tr><td>LEIRIA</td></tr><tr><td>PORTO</td></tr></tbody></table> <p><i>output(query B)</i></p>	CONCELHO_NOME	COIMBRA	COIMBRA	LEIRIA	LEIRIA	PORTO	<table border="1"><thead><tr><th>CONCELHO_NOME</th></tr></thead><tbody><tr><td>BATALHA</td></tr><tr><td>CONDEIXA-A-NOVA</td></tr><tr><td>MARINHA GRANDE</td></tr><tr><td>MIRANDA DO CORVO</td></tr></tbody></table> <p><i>output</i> = <i>output(query A)</i> - <i>output(query B)</i></p>	CONCELHO_NOME	BATALHA	CONDEIXA-A-NOVA	MARINHA GRANDE	MIRANDA DO CORVO
CONCELHO_NOME																					
BATALHA																					
CONDEIXA-A-NOVA																					
COIMBRA																					
LEIRIA																					
MARINHA GRANDE																					
MIRANDA DO CORVO																					
PORTO																					
CONCELHO_NOME																					
COIMBRA																					
COIMBRA																					
LEIRIA																					
LEIRIA																					
PORTO																					
CONCELHO_NOME																					
BATALHA																					
CONDEIXA-A-NOVA																					
MARINHA GRANDE																					
MIRANDA DO CORVO																					

8. Mostrar os identificadores dos produtos vendidos tanto no dia 30 de junho de 2021 como no dia 1 de julho de 2021.

PRODUTO_ID	output
1060	

9. Mostrar, num só resultado, os nomes dos distritos (em maiúsculas) e os nomes dos concelhos (em minúsculas).

output	
[D] PORTO [D] LISBOA [D] LEIRIA [D] COIMBRA	distritos
[C]porto [C]miranda do corvo [C]marinha grande [C]leiria [C]condeixa-a-nova [C]coimbra [C]batalha	concelhos