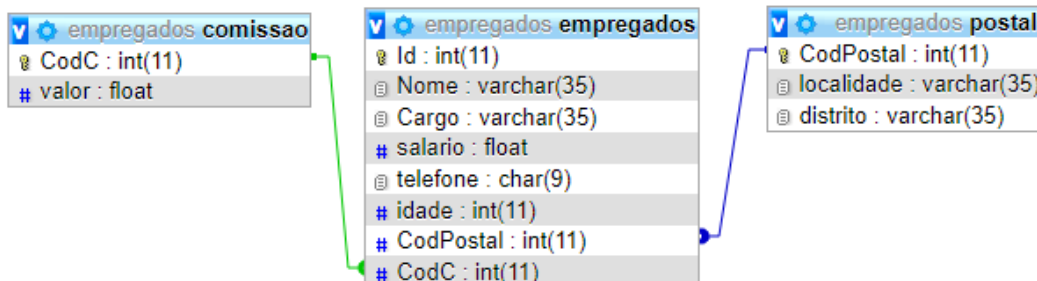


**Curso Profissional: Programador/a de Informática**  
**PSD – 10.º ano: UFCD 10778 – Fundamentos da linguagem SQL**

**Ficha de Trabalho 3**

**Ano letivo 21/22**

**NOTA:** Relações em que se baseiam os exemplos da ficha de trabalho:



## JUNÇÃO DE TABELAS

A junção mais comum, de duas ou mais tabelas, é a inner join onde são apresentados os registos em que existe ligação entre as tabelas, esta estabelece-se através da chave externa de uma e da chave primária da outra, na cláusula WHERE.

**Exemplo:** Selecionar o nome, Cod Postal e Localidade de todos os empregados.

```
SELECT Nome, CodPostal, localidade
FROM empregados, postal
WHERE empregados.CodPostal=postal.CodPostal
```

- Como os campos usados na junção na cláusula WHERE têm nomes iguais é necessário fazer referência à tabela a que eles pertencem, isto é, é necessário preceder o campo do nome da tabela, seguido de um ponto, tipo, *empregados.CodPostal*;
- Caso os nomes dos campos de ligação entre as tabelas fossem diferentes, era indiferente colocar um ou outro no comando SELECT.

## O predicado TOP

Usado para restringir o n.º de linhas a apresentar resultantes de uma consulta. Pode-se definir o n.º ou a percentagem de linhas a apresentar. Este predicado apenas é válido em SQL Access.

## O predicado LIMIT

Tem a mesma função que o predicado TOP mas a aplicar em MySQL

**Exemplo:** Selecionar o nome dos empregados com os 10 melhores ordenados da empresa.

SGBD Access	SGBD MySQL
<pre>SELECT TOP 10 Nome FROM empregados ORDER BY salario desc</pre>	<pre>SELECT Nome FROM empregados ORDER BY salario desc LIMIT 10</pre>

## FUNÇÕES DE AGREGAÇÃO

FUNÇÃO	DESCRIÇÃO
COUNT	Devolve o n.º de linhas
MAX	Devolve o maior valor da coluna
MIN	Devolve o menor valor da coluna
SUM	Devolve a soma de todos os valores da coluna
AVG	Devolve a média de todos os valores da coluna

### Função COUNT

FORMA	DESCRIÇÃO
COUNT(*)	contagem do n.º de linhas de uma tabela (nulos incluídos)
COUNT(campo)	contagem do n.º de ocorrências diferentes de null
COUNT(DISTINCT campo)	contagem do n.º de ocorrências diferentes de null, sem repetições

### Exemplos:

1- Contagem do n.º de empregados.

```
SELECT COUNT(*) as TOTAL
FROM empregados
```

2- Qual o n.º de pessoas e quantas têm telefone

```
SELECT COUNT(*) as TOTALPessoas, COUNT(telefone) AS TOTAL_telefone
FROM empregados
```

3- Quantas pessoas não têm telefone.

```
SELECT COUNT(*) as 'TOTAL n.º de Telefone'
FROM empregados
WHERE telefone IS NULL
```

As restantes funções de agregação podem ser aplicadas de modo idêntico, tendo em atenção que:

- ✚ A 1ª forma não lhes é aplicável (função(\*));
- ✚ As funções SUM e AVG só podem ser aplicadas a campos numéricos;
- ✚ Também elas ignoram os campos com valores nulos.

## CLÁUSULA GROUP BY

Esta cláusula está intimamente ligada às funções de agregação e possibilita o agrupamento de informação.

- ✚ Se um comando SELECT contiver a cláusula GROUP BY, então todos os campos selecionados (no SELECT) devem estar presentes na cláusula GROUP BY.

**Exemplos:** Supondo existir uma tabela de Comissões com os campos **CodC** (código da comissão) e **valor** (da comissão recebida).

1- Mostrar para cada CodC o valor total das comissões.

```
SELECT CodC, SUM(valor) AS TOTALComissao
FROM comissao
GROUP BY CodC
```

2- Mostrar para cada nome do empregado o valor total das comissões.

```
SELECT nome, SUM(valor) AS TOTALComissao
FROM comissao, empregados
WHERE comissao.CodC=empregados.CodC
GROUP BY Nome
```

3-Mostrar o valor total (Salario + comissões) a receber por cada empregado.

```
SELECT nome, valor + salario AS 'TOTAL a RECEBER'
FROM comissao, empregados
WHERE comissao.CodC=empregados.CodC
GROUP BY Nome, valor, salario
```

nome	TOTAL a RECEBER
Ana Marques	1400
Charles Xavier	1050
Cristina Ferreira	1050
Filomena Cautela	1450
Jorge Jesus	900
Lili Canêças	1200
Marcelo de Sousa	2850
Nilton	1050

**Nota:** É feito o agrupamento também por valor e salario porque esse campo é usado no cálculo da expressão a apresentar.

4- Selecionar na tabela empregados o nº de comissões de cada CodC.

```
SELECT CodC, COUNT(*) as 'Número Total de Comissões'
FROM empregados
GROUP BY CodC
```

CodC	Número Total de Comissões
1	2
2	2
3	3
4	1

## CLÁUSULA HAVING

Serve para fazer restrições ao nível dos grupos que estão a ser processados, isto é, atua sobre os resultados dos grupos, isto é, sobre o resultado de uma função de agregação.

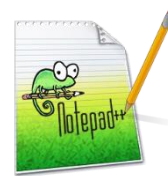
**Exemplo:** Mostrar os CodC, da tabela empregados com comissões totais superiores a 500.

```
SELECT comissao.CodC, SUM(valor) as 'Valor Total'
FROM empregados, comissao
WHERE empregados.CodC=comissao.CodC
GROUP BY comissao.CodC
HAVING SUM(valor)>500
```

CodC	Valor Total
1	700
2	600

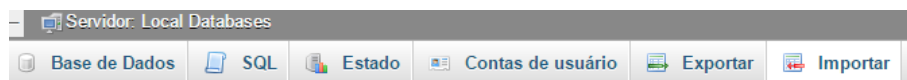
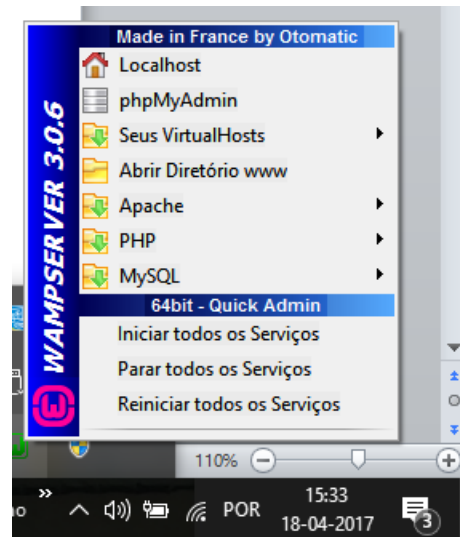
Caso não tenhas já instalado, instala o Notepad++.

<http://notepad-plus-plus.org/download>



## Exercícios

1. Retira do Classroom, da disciplina de PSD, o ficheiro **SQLVendas.SQL** e abre-o com o Notepad++.
2. Inicializa o serviço de **Wamp**, e através do PHPMYAdmin elabora as seguintes operações: **phpmyadmin**.
3. Configura o mapa de caracteres para latin1\_bin:
4. No MySQL seleciona o separador **Importar** e pressiona no botão **Escolher ficheiro** para selecionar o ficheiro **SQLVENDAS\_UTF8.SQL**, que se encontra na *Classroom, Disciplina de PSD, no tópico UFCD 10778 - Fundamentos da linguagem SQL*. Escolhe o mapa de caracteres **utf-8**:



### Fazendo importação para o servidor atual

#### File to import:

O ficheiro pode ser comprimido (gzip, bzip2, zip) ou descomprimido.

O nome de um ficheiro comprimido tem que acabar em **[format].[compression]**. Exemplo: **.sql.zip**

Procurar no seu computador:  SQLVENDAS\_UTF8.SQL (Tamanho máximo: 128MB)

You may also drag and drop a file on any page.

Configurar o Mapa de Caracteres do ficheiro:

**Nota:** Este ficheiro em SQL, cria uma base de dados de nome “SQLVENDAS” com quatro tabelas pessoa, postal, mensagem e comissão e insere os dados respetivos.

5. Seleciona o botão **Executa**.

A base de dados correspondente ao seguinte esquema:



Com as seguintes tabelas:

COMISSAO

ID	ID_Msg	VALOR
14	10	10500
14	60	2600
14	70	400
14	100	3750
25	10	2500
25	30	370
37	10	5500
37	30	14230
37	40	20
37	50	120
42	20	20
42	30	170
49	20	2300

MENSAGEM

ID_Msg	MENSAGEM
10	Comissão de Vendas
20	Frete Individuais
30	Frete Empresariais
40	Vendas Extra
50	Deslocações
60	Refeições
70	Combustíveis
80	Transportes
90	Telefonemas
100	Ofertas

PESSOA

POSTAL

CODIGO	LOCALIDADE
1000	LISBOA
1100	LISBOA
1200	LISBOA
1500	LISBOA
2000	SANTARÉM
2300	TOMAR
3000	COIMBRA
4000	PORTO
9000	FUNCHAL

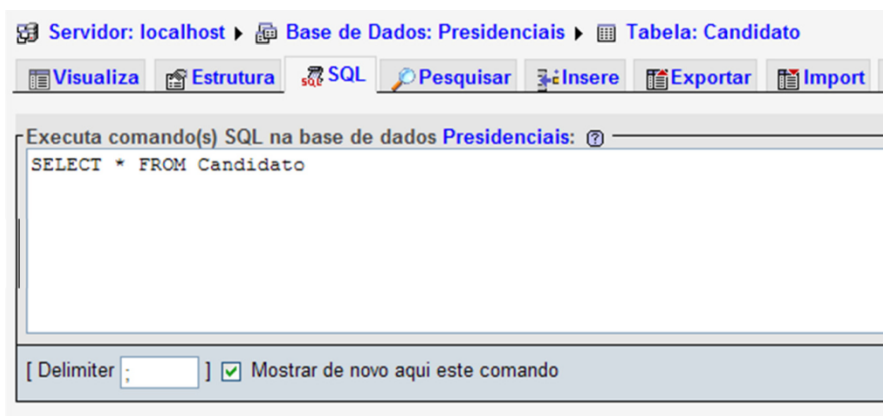
ID	NOME	IDADE	SALARIO	TELEFONE	CODPOSTAL
5	Célia Morais	26	17000	123456000	1100
14	Nascimento Augusto	35	22000	456123000	2300
25	Paulo Viegas	32	9500		1500
32	Florinda Simões	35	14700		4000
37	Isabel Espada	28	8600		1100
42	António Dias	43	7400	789654000	1500
49	José António	19	21000		1500

6. Selecciona o separador **SQL** e executa as seguintes questões à base de dados:

- Seleccionar todas as pessoas com 35 anos.
- Seleccionar o Id, Nome e Salario de todas as pessoas com idade para votar.
- Seleccionar a descrição de todas as Mensagens cujo Id\_Msg é diferente de 20.
- Seleccionar o Id, o Nome, a Idade e o Salario de todas as pessoas com Idade entre os 30 e os 40 anos.
  - Utiliza os operadores: AND e >=.
  - Utiliza os operadores: BETWEEN e AND.

- e. Selecionar o Id, o Nome, a Idade e o salario de todas as pessoas que não fazem parte da seleção anterior, isto é, cuja idade **não** está entre os 30 e os 40 anos.
  - i. Utiliza os operadores: OR, < e >.
  - ii. Utiliza os operadores: NOT, AND, >= e <=.
  - iii. Utiliza os operadores NOT, BETWEEN, e AND.
- f. Selecionar a lista completa (todos os campos) de Códigos Postais de LISBOA e TOMAR.
  - i. Utiliza os operadores OR e =.
  - ii. Utiliza o operador IN.
- g. Selecionar os Códigos Postais que não pertencem a LISBOA nem a TOMAR.
  - i. Utiliza os operadores AND e <>.
  - ii. Utiliza os operadores NOT e IN.
- h. Selecionar os Nomes das pessoas que são de LISBOA ou TOMAR.
- i. Selecionar o valor total pago em comissões, por nome.
- j. Selecionar os Nomes das pessoas sem telefone, (utilize IS NULL).
- k. Selecionar os Nomes e localidade das pessoas com salarios superiores a 10000.
- l. Selecionar todas as mensagens começadas por T (operador LIKE).
- m. Selecionar todas as mensagens que acabam em “as” (operador LIKE).
- n. Selecionar todas as mensagens que contenham a palavra “Vendas” (operador LIKE).
- o. Selecionar todos os Nomes que não contenham a string “da” (operador NOT LIKE).
- p. Selecionar todos os Nomes cujo segundo carácter é “a” (operador LIKE).
- q. Selecionar o nome e salário das pessoas com os 3 melhores salários, ordenados por ordem descendente.
- r. Selecciona o salário máximo.
- s. Selecionar o nome e idade da pessoa com o melhor salário (usa um alias).
- t. Selecionar o nº de pessoas por Localidade (usa um alias).

7. Selecciona agora base de dados de nome Presidenciais (criada na ficha anterior).
8. Selecciona o separador SQL:



9. Executa as seguintes pesquisas:
  - a. Obter as votações e nome dos candidatos:

```
SELECT Votos, NomeCand
FROM votacao,candidato
WHERE votacao.idcandidato = candidato.idcandidato
```

Votos	NomeCand
411341	Cavaco Silva
64719	Cavaco Silva
52168	Cavaco Silva
136633	Fernando Nobre
14534	Fernando Nobre
7687	Fernando Nobre
184446	Manuel Alegre
14534	Manuel Alegre
9105	Manuel Alegre
72465	Francisco Lopes
3937	Francisco Lopes
2346	Francisco Lopes
29070	José Coelho
4425	José Coelho
46247	José Coelho
12668	Defensor Moura
11745	Defensor Moura
986	Defensor Moura

- b. Votações, nomes e distrito dos candidatos:

```
SELECT Votos, NomeCand, distrito
FROM Votacao, Distrito, candidato
WHERE votacao.idcandidato=candidato.IdCandidato
AND distrito.IdDistrito=votacao.IdDistrito
```

Votos	NomeCand	Distrito
411341	Cavaco Silva	Lisboa
64719	Cavaco Silva	Viana
52168	Cavaco Silva	Madeira
136633	Fernando Nobre	Lisboa
14534	Fernando Nobre	Viana
7687	Fernando Nobre	Madeira
184446	Manuel Alegre	Lisboa
14534	Manuel Alegre	Viana
9105	Manuel Alegre	Madeira
72465	Francisco Lopes	Lisboa
3937	Francisco Lopes	Viana
2346	Francisco Lopes	Madeira
29070	José Coelho	Lisboa
4425	José Coelho	Viana
46247	José Coelho	Madeira
12668	Defensor Moura	Lisboa
11745	Defensor Moura	Viana
986	Defensor Moura	Madeira

- c. Número total de votos na Madeira.
  - d. Número de votos do candidato 1, no distrito 2.
  - e. Total de votos por distrito do candidato 1.
  - f. Total de votos por distrito de todos os candidatos.
  - g. Os candidatos ordenados por ordem alfabética.
  - h. Nomes dos distritos e candidatos em que a votação foi superior a 100000.

#### Bibliografia

Damas, L. (2005). SQL. Lisboa: FCA  
Tavares, F. (2015). MySql. Lisboa: FCA

CORREÇÃO EX9





c. Número total de votos na Madeira.

```
SELECT SUM(votos) AS Totalvotos FROM Votacao, Distrito WHERE distrito.IdDistrito=votacao.IdDistrito AND distrito.Distrito='Madeira'
```

d. Número de votos do candidato 1 no distrito 2:

```
SELECT SUM(votos) AS Totalvotos FROM Votacao WHERE IdCandidato=1 AND IdDistrito=2;
```

e. Total de votos do candidato 1 no distrito 2.

f. Total de votos por distrito do candidato 1.

```
SELECT iddistrito, SUM(votos) AS Totalvotos FROM Votacao WHERE IdCandidato=1 GROUP BY idDistrito
```

g. Total de votos por distrito de todos os candidatos.

```
SELECT iddistrito, SUM(votos) AS Totalvotos FROM Votacao GROUP BY idDistrito
```

h. Os candidatos ordenados por ordem alfabética.

```
SELECT NomeCand FROM Candidato ORDER BY NomeCand ASC
```

i. O nome do distrito com maior número de votos por candidato.

(X)

O distrito e o candidato com maior votação.

```
SELECT NomeCand, Distrito, votos FROM Votacao, Distrito, candidato WHERE distrito.IdDistrito=votacao.IdDistrito AND candidato.IdCandidato=votacao.IdCandidato AND votos=(SELECT MAX(votos) FROM votacao)
```

O distrito e o candidato com menor votação.

```
SELECT NomeCand, Distrito, votos FROM Votacao, Distrito, candidato WHERE distrito.IdDistrito=votacao.IdDistrito AND candidato.IdCandidato=votacao.IdCandidato AND votos=(SELECT MIN(votos) FROM votacao)
```

j. Nomes dos distritos e candidatos em que a votação foi superior a 100 000.

```
SELECT NomeCand, Distrito, SUM(votos) FROM Votacao, Distrito, candidato WHERE distrito.IdDistrito=votacao.IdDistrito AND candidato.IdCandidato=votacao.IdCandidato GROUP BY NomeCand, distrito HAVING SUM(votos)>100000
```

k. Nomes dos distritos e candidatos em que a votação foi inferior ou igual a 100 000.

```
SELECT NomeCand, Distrito, SUM(votos) FROM Votacao, Distrito, candidato WHERE distrito.IdDistrito=votacao.IdDistrito AND candidato.IdCandidato=votacao.IdCandidato GROUP BY NomeCand, distrito HAVING SUM(votos)<100000
```

```
CREATE view maxcand AS SELECT NomeCand, MAX(votos) FROM Votacao, Distrito, candidato WHERE distrito.IdDistrito=votacao.IdDistrito AND candidato.IdCandidato=votacao.IdCandidato GROUP BY NomeCand
```