

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

**Ficha de Trabalho 1**

**Ano letivo 21/22**

**SQL:** *Structured Query Language* que não é mais do que uma linguagem padrão de comunicação com base de dados. É uma linguagem normalizada que nos permite trabalhar com qualquer tipo de linguagem de programação (VisualBasic, ASP, PHP, Java, etc.) em combinação com qualquer tipo de base de dados (MS Access, Oracle, SQLServer, MySQL, etc.).

**DML (Data Manipulation Language):** É o conjunto de comandos SQL responsáveis pela manipulação dos dados, como: inserir, consultar, atualizar e excluir;

**DDL (Data Definition Language):** É o conjunto de comandos SQL responsável pela definição dos dados, ou seja, pela criação de base de dados, esquemas, tabelas, campos, tipos de dados, restrições etc..

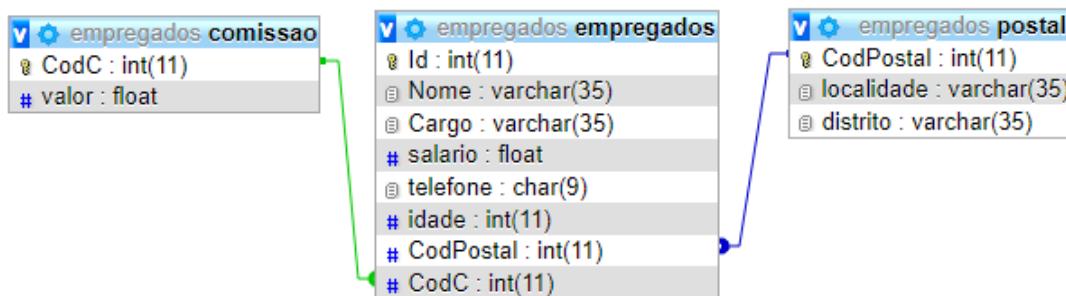
## CONSULTAR UMA BASE DE DADOS

**Projeção** - operação de consulta em que se restringe o nº de colunas a apresentar.

**Restrição** - operação de consulta em que se restringe o nº de linhas a apresentar (cláusula *where*).

A interrogação de uma base de dados relacional é feita utilizando o comando *select*, que é o comando mais utilizado e importante da linguagem SQL.

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



# 1- Comando Select

## Sintaxe:

```
SELECT [DISTINCT] {<campo1> [AS <Nomealternativo>], <campo2>, ..., <campoN>[AS  
<Nomealternativo>] | *}  
FROM <Tabela1>, ...<TabelaM>  
[WHERE condição]  
[GROUP BY ...]  
HAVING...]  
[ORDER BY <campo1> [ASC | DESC], ... <campoX> [ASC | DESC] [LIMIT] <numderegistos>
```

- Os elementos entre parêntesis retos são opcionais. O símbolo \*, quando usado, representa todos os campos da tabela.
- A cláusula **DISTINCT** permite eliminar repetições de linhas no resultado de uma seleção.  
**Exemplo:** SELECT DISTINCT salario FROM Empregados  
WHERE CodPostal='6200-571'
- A Cláusula **AS** é um alias, isto é, um Nome alternativo para o campo e com o qual será apresentado.
- A cláusula **ORDER BY**, permite ordenar os registos  
**Exemplo:** SELECT \* FROM Empregados ORDER BY salario desc
- A Cláusula **LIMIT** permite limitar o número de registos para consulta

**Exemplo:** SELECT \* FROM Empregados LIMIT 3 (apresenta os três 1.ºs primeiros registos da tabela Empregados).

## EXEMPLOS:

1-Selecionar todas as colunas da tabela Empregados.

```
SELECT * FROM Empregados
```

2- Selecionar na tabela Empregados os campos Id e Nome.

```
SELECT Id, Nome FROM Empregados
```

3- Selecionar na tabela Empregados o Nome e o salario de todas as pessoas com mais de 30 anos de idade.

```
SELECT Nome, salario FROM Empregados  
WHERE idade>30
```

## Operadores relacionais

Operador	Descrição
>	maior que
<	menor que
=	Igual a
< > OU !=	Diferente de
>=	Maior ou igual
<=	Menor ou igual

## Operadores lógicos

Operador	Descrição
AND ou &&	conjunção
OR ou	disjunção
NOT ou !	negação
XOR	Disjunção exclusiva (resulta true se e somente se apenas um dos operandos possui valor true)

Uso do XOR: <https://pt.khanacademy.org/computing/computer-science/cryptography/ciphers/a/xor-and-the-one-time-pad>

## EXEMPLOS DE UTILIZAÇÃO:

1- Selecionar os registos de todas as pessoas com idade entre 30 e 40.

```
SELECT * FROM Empregados  
WHERE idade > 30 AND idade <40
```

2- Selecionar os registos de todas as pessoas que não fazem parte da seleção anterior

```
SELECT * FROM Empregados  
WHERE NOT (idade > 30 AND idade <40)
```

## Operadores aritméticos

Operador	Descrição
DIV	Divisão de números inteiros
/	Divisão real
-	subtração
% ou MOD	Resto da divisão
+	Adição
*	Multiplificação

## Outros operadores

O conjunto seguinte de operadores tem como objetivo ser utilizado com a instrução WHERE, definindo as condições de extração de dados quando são utilizados os comandos SELECT, UPDATE e DELETE (os dois últimos referentes à linguagem e definição de dados)

Operador	Descrição
BETWEEN	Verifica se estamos num conjunto de valores específicos
IN	Verifica se existe um dado valor na tabela
IS NULL	Verifica de o campo é nulo
LIKE	Seleção de um valor específico usando um <i>wildcard</i> <sup>1</sup> (compara o dado fornecido com um elemento específico e exato)

<sup>1</sup> o wildcard é um carácter que irá substituir um ou vários caracteres e varia conforme o tipo de SGBD

Em Access/MySQL é o símbolo \* / % para qualquer conjunto de 0 ou mais caracteres, e ? / \_ (undersore) para um carácter qualquer.

**BETWEEN** – permite especificar intervalos de valores

**Sintaxe:**

```
SELECT ...  
FROM ...  
WHERE campo [not] BETWEEN valor1 AND valor2
```

As condições seguintes são equivalentes:

campo >= valor1 AND campo <=valor2

campo BETWEEN valor1 AND valor2

**Exemplo:** Selecionar os registos de todas as pessoas que não têm idade entre 30 e 40.

```
SELECT * FROM Empregados  
WHERE idade NOT BETWEEN 31 AND 39
```

**IN** – Permite verificar se algo faz parte de um conjunto de valores

**Sintaxe:**

```
SELECT ...  
FROM ...  
WHERE campo [NOT] IN (valor1, valor2, ...valorN)
```

As condições seguintes são equivalentes:

campo=1 or valor =2 or campo =3

campo IN (1, 2, 3)

**Exemplo 1:** Selecionar a lista completa com os Nomes dos Empregados com o Cargo “Chefe de secção” e “Tesoureiro”.

```
SELECT Nome  
FROM Empregados  
WHERE Cargo IN ('Chefe de secção' , 'Tesoureiro')
```

**Exemplo 2:** Selecionar os Nomes dos empregados que pertencem ao distrito da Guarda.

```
SELECT Nome  
FROM Empregados  
WHERE CodPostal IN (SELECT Codpostal from postal where Distrito='Guarda')
```

**IS** – usado para tratamento de valores nulos (associado ao valor NULL)

**Sintaxe:**

```
SELECT ...  
FROM ...  
WHERE campo IS [NOT] NULL
```

**Exemplo:**

1- Selecionar o Id e Nome da tabela Empregados das pessoas que não têm telefone (supondo que Telefone é um dos campos da tabela).

```
SELECT Id, Nome  
FROM Empregados  
WHERE Telefone IS NULL
```

1- Selecionar o Id e Nome da tabela empregados das pessoas que têm telefone (supondo que telefone é um dos campos da tabela).

```
SELECT Id, Nome  
FROM empregados  
WHERE telefone IS NOT NULL
```

**LIKE** – permite comparação de strings. A Utilização do carácter (\* ou %) substitui qualquer conjunto de caracteres e o carácter (?) substitui um carácter qualquer.

**Exemplos:**

1- Selecionar todos os Nomes da tabela Empregados que comecem por P.

```
SELECT Nome  
FROM empregados  
WHERE Nome LIKE "P*"; em MySQL seria ... LIKE 'P%'
```

(Nota: a linguagem sql não é case sensitive)

2- Selecionar todos os Nomes que NÃO contenham a string 'da'.

```
SELECT Nome  
FROM empregados  
WHERE Nome NOT LIKE "*de*"; em MySQL seria ...NOT LIKE '%de%'
```

**Bibliografia**

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

## EXERCÍCIOS (CONSOLIDAÇÃO):

1.

**NOTA:** Tem em conta o seguinte esquema de relação entre as tabelas Comissão, Empregados e Postal.



Escreve as instruções SQL para:

- Selecionar todas as colunas da tabela postal.
- Selecionar na tabela empregados os campos Nome e cargo.
- Selecionar na tabela empregados o Nome e o salario de todas as pessoas com menos de 50 anos de idade.
- Selecionar os registos de todas as pessoas com idade entre 40 e 60.
- Selecionar os registos de todas as pessoas que não fazem parte da seleção anterior.
- Selecionar os registos de todas as pessoas que não têm idade entre 25 e 45.
- Selecionar a lista completa com os Nomes e salario dos empregados com o cargo “CEO” e “Tesoureiro” (usando o operador IN e operadores relacionais).
- Selecionar os Nomes dos empregados que pertencem ao distrito da Guarda (usando o operador IN).
- Selecionar o Id e Nome da tabela empregados das pessoas que têm telefone.
- Selecionar todos os Nomes da tabela empregados que não comecem por L.
- Selecionar todos os Nomes que NÃO contenham a string ‘mar’.

2. Considera os seguintes comandos SQL, e explica o que é feito pelo SGDB, em cada alínea:

2.1.

```
SELECT Nome, CodPostal, idade  
FROM empregados  
ORDER BY CodPostal, idade
```

2.2.

```
SELECT Nome, idade as Idade_agora, Idade+1 as Idade_mais_1  
FROM empregados  
ORDER BY Nome
```

2.3.

```
SELECT CodC, valor as Liquido, valor*0.20 as Imposto,  
valor+valor*0.20 as Bruto  
FROM comissao  
WHERE CodC IN (14,25)  
ORDER BY CodC
```