

PARA A RESOLUÇÃO DO E-FÓLIO, ACONSELHA-SE QUE LEIA ATENTAMENTE O SEGUINTE:

- 1) O e-fólio é constituído por 3 perguntas. A cotação global é de 2 valores.
- 2) O e-fólio deve ser entregue num único ficheiro PDF, não zipado, com fundo branco, com perguntas numeradas e sem necessidade de rodar o texto para o ler. Penalização de 1 a 3 valores.
- 3) Não são aceites e-fólios manuscritos, i.e. tem penalização de 100%.
- 4) O nome do ficheiro deve seguir a normal "eFolioA" + <nº estudante> + <nome estudante com o máximo de 3 palavras>
- 5) Durante a realização do e-fólio, os estudantes devem concentrar-se na resolução do seu trabalho individual, não sendo permitida a colocação de perguntas ao professor ou entre colegas.
- 6) A interpretação das perguntas também faz parte da sua resolução, se encontrar alguma ambiguidade deve indicar claramente como foi resolvida.
- 7) A legibilidade, a objetividade e a clareza nas respostas serão valorizadas, pelo que, a falta destas qualidades serão penalizadas

Cotação de cada questão: **0,2 pontos**

Considere uma base de dados para gestão dos funcionários de uma cadeia de lojas. O registo de funcionários inclui o nome, a data de início de contrato, a identificação do departamento a que pertence, o cargo, o salário e a comissão que recebe (os balconistas). No caso de o funcionário ter outros funcionários debaixo de sua responsabilidade, o número total destes é também registado. Como existe um mecanismo de registo presença, a BD contempla também essa situação. Os funcionários podem ser classificados como “gestor”, “balconista” ou “faxineiro”. As seguintes tabelas foram criadas para esse propósito:

Funcionarios (idFunc->nome, dataInicio, idDepto, cargo, salario, comissao, nGere)

Departamentos (idDepto->nome, idLocalizacao)

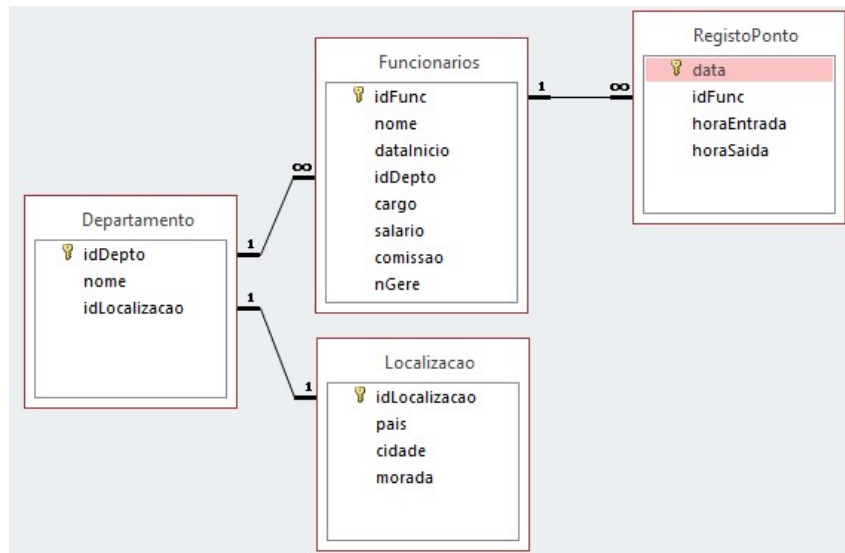
Localizacao(idLocalizacao-> pais, cidade, morada)

RegistoPonto(data, idFunc->horaEntrada, horaSaida)

- 1) O que é o modelo Entidade-Relacionamento? Para que serve? Explique de forma resumida e objetiva.
- 2) Represente graficamente a base de dados relacional, com as respetivas tabelas e ligações de chaves estrangeiras. Siga a seguinte regra para a representar: nas ligações de 1:N a tabela com uma única linha é desenhada em cima e da tabela com várias linhas é desenhada por baixo.

Solução:

- Ele inicia por identificar as entidades e seus relacionamentos, sendo posteriormente adicionados outros detalhes (atributos e restrições).
- O conceito básico no modelo E-R é a entidade-tipo.
- Os objetos podem ser reais ou apenas conceptuais (vendedor e venda).
- Cada objeto único identificável de uma entidade-tipo é designado de ocorrência da entidade.
- Um tipo de relacionamento é um conjunto de associações entre uma ou mais entidades-tipo.
- A cada relacionamento é atribuído um nome que descreve a sua função.
- Cada tipo de relacionamento é representado como uma linha conectando as entidades-tipo, rotulado com o nome do relacionamento:
 - É utilizado um verbo ou uma frase curta (supervises, LeasedBy);
 - A 1ª letra é em maiúscula;
 - Deve ser único sempre que possível;
- É definida uma direção para o relacionamento com uma seta quando binária (normalmente omitida nos demais casos).
- O grau do relacionamento é dado pelo número de entidades-tipo participantes no mesmo.



Exprima em SQL as seguintes consultas, tendo como base esta BD:

3) Mostrar a data de contrato, nome e número de departamento para todos os balconistas.

```

SELECT dataInicio, nome, idDepto
FROM Funcionarios
WHERE cargo = 'Balconista';
  
```

4) Mostrar os nomes e datas de contratação de todos os empregados contratados entre 1981 e 1982.

```

SELECT nome, dataInicio
FROM Funcionarios
WHERE dataInicio LIKE '%1981' OR dataInicio LIKE '%1982';
  
```

5) Mostrar o nome 'do departamento', o idDepto e a 'comissao ganha \$' para cada balconista em ordem decrescente (listar, montando uma frase com esse conteúdo).

```

SELECT nome || ' do departamento ' || idDepto || ' comissao ganha $' || comissao
FROM Funcionarios
WHERE cargo = 'Balconista'
ORDER BY salario DESC;
  
```

6) Mostrar o número do departamento e o total do salário dos empregados em cada departamento que tenha mais de 4 empregados.

```

SELECT idDepto, sum(salario)
FROM Funcionarios
GROUP BY idDepto
HAVING count(idFunc) >= 4;
  
```

7) Exibir uma lista de todos os departamentos com os funcionários em cada departamento.

```

SELECT nome, nome
FROM Departamentos
  
```

```
LEFT OUTER JOIN Funcionarios ON Departamentos.idDepto = Funcionarios.idDepto;
```

8) Exibir todos os departamentos com um gestor de departamento.

```
SELECT nome, nome
FROM Funcionarios
INNER JOIN Departamentos ON Funcionarios.idDepto = Departamentos.idDepto
WHERE cargo = 'Gestor';
```

9) Mostrar os nomes e cargos de todos os funcionários que tenham um cargo similar a alguém com nome Maria.

```
SELECT nome, cargo
FROM Funcionarios
WHERE cargo = (SELECT cargo
                FROM Funcionarios
                WHERE nome = Maria');
```

10) Exibir os nomes de todos os empregados, exceto os mais mal pagos.

```
SELECT nome
FROM Funcionarios
WHERE salario > ( SELECT min(salario)
                  FROM Funcionarios);
```

Comentário final e critérios de correção geral:

- O SQL não precisa ser comentado.
- As palavras-chave do SQL devem ser escritas em maiúsculas e o restante código em minúsculas.
- O SQL bem indentado deve ter as cláusulas na encostada à esquerda e as sub-consultas destacadas à direita, como foi apresentado nas atividades formativas e como se segue:
SELECT..... FROM
WHERE..... AND.....(SELECT..... FROM.....
WHERE.....) GROUP BY HAVING
- A falta de indentação é penalizada por cada questão, com 1 a 2 décimas.
- Erro comum: WHERE trabalha sobre as linhas (tuplos)
HAVING trabalha sobre os valores agregados dos grupos
- Nas próximas avaliações de evitar as cláusulas: WITH, TOP e LIMIT.