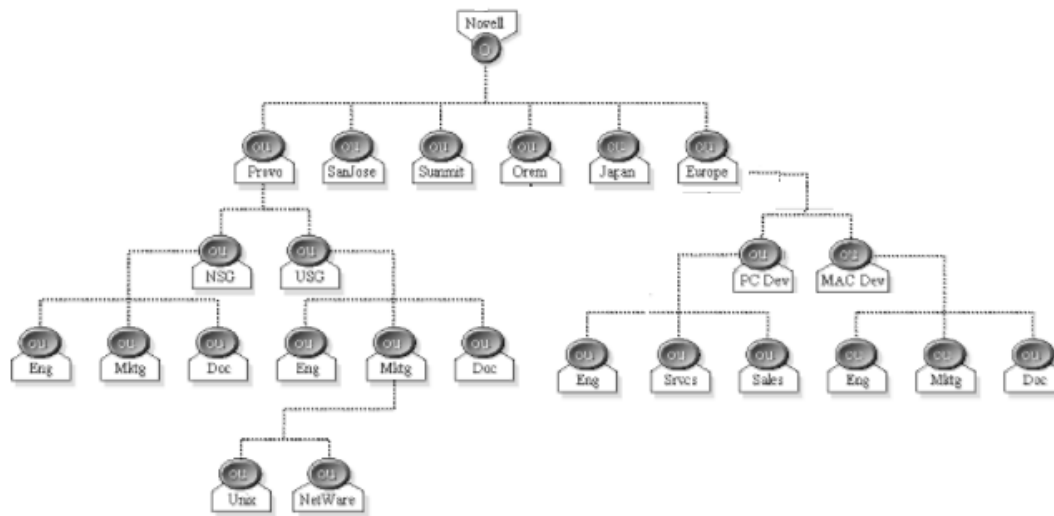


**21053 - Fundamentos de Bases de Dados
2017-2018
e-fólio B**

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

- 1) O e-fólio é constituído por 5 perguntas. A cotação global é de 5 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 10% a 100%.
- 3) Não são aceites e-fólios manuscritos, i.e. tem penalização de 100%.
- 4) O nome do ficheiro deve seguir a normal “eFolioB” + <nº estudante> + <nome estudante com o máximo de 3 palavras>
- 5) Na primeira página do e-fólio deve constar o nome completo do estudante bem como o seu número. Penalização de 10% a 100%.
- 6) 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.
- 7) A interpretação das perguntas também faz parte da sua resolução, se encontrar alguma ambiguidade deve indicar claramente como foi resolvida.
- 8) A legibilidade, a objectividade e a clareza nas respostas serão valorizadas, pelo que, a falta destas qualidades serão penalizadas.

1) Considere a seguinte figura relativa com a estrutura hierárquica com 5 níveis de uma empresa:



a) (1/2 valor) crie uma base de dados em SQL que suporte a estrutura hierárquica

Most users at one time or another have dealt with hierarchical data in a SQL database and no doubt learned that the management of hierarchical data is not what a relational database is intended for. The tables of a relational database are not hierarchical (like XML), but are simply a flat list. Hierarchical data has a parent-child relationship that is not naturally represented in a relational database table.

For our purposes, hierarchical data is a collection of data where each item has a single parent and zero or more children.

CREATE TABLE empresa(

```

    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(20) NOT NULL,
    pai INT DEFAULT NULL
  
```

);

INSERT INTO empresa

```

VALUES(1,'Novell',NULL),(2,'Prevo',1),(3,'NSG',2),(4,'Eng',3),(5,'Mktg',3),(6,'Doc',
3),
  
```

```

(7,'USG',2),(8,'Eng',7),
  
```

```

(9,'Mktg',7),(10,'Uriot',9),(11,'Netware',9),
  
```

```

(12,'Doc',8),
  
```

```
(13,'SanJose',1),(14,'Summit',1),(15,'Orem',1),(16,'Japan',1),(17,'Europe',1),
```

```
(18,'PC Dev',17),(19,'Eng',18),(20,'Srvcs',18),(21,'Sales',18),
```

```
(22,'MAC Dev',17),(23,'Eng',22),(24,'Mktg',22),(25,'Doc',22);
```

```
SELECT * FROM empresa ORDER BY id;
```

In the adjacency list model, each item in the table contains a pointer to its parent. The topmost element, in this case *electronics*, has a NULL value for its parent. The adjacency list model has the advantage of being quite simple, it is easy to see that *FLASH* is a child of *mp3* players, which is a child of *portable electronics*, which is a child of *electronics*. While the adjacency list model can be dealt with fairly easily in client-side code, working with the model can be more problematic in pure SQL.

b) (1/2 valor) crie uma consulta em SQL que devolva o número de níveis na empresa

```
SELECT COUNT(*) AS Num_Niveis FROM (SELECT distinct t1.pai FROM
empresa T1, empresa T2
where T1.id_categoria = T2.pai) AS SUBQUERY
```

2) Considere 3 utilizadores B1, B2 e B3 e suponha que o utilizador B1 cria a tabela FACTURA numa base de dados, de seguida diga:

2.a) (1/3 valor) Como B1 pode transmitir o privilégio de acesso a FACTURA a B2 e permite-lhe que propague esse privilégio a outros.

A transferência de privilégios é realizada por um utilizador com autorização para realizar operações sobre tabelas ou views e as transfere a outros utilizadores, com o comando Grant (conceder, permitir).

```
GRANT ALL ON OR SELECT ON? (access)
```

```
GRANT SELECT ON "FACTURA" TO "B2" WITH GRANT OPTION
```

2.b) (1/3 valor) Como B2 transmite o privilégio de acesso a FACTURA a B3. **Nota: nesse caso não vai ter o privilégio de propagação.**

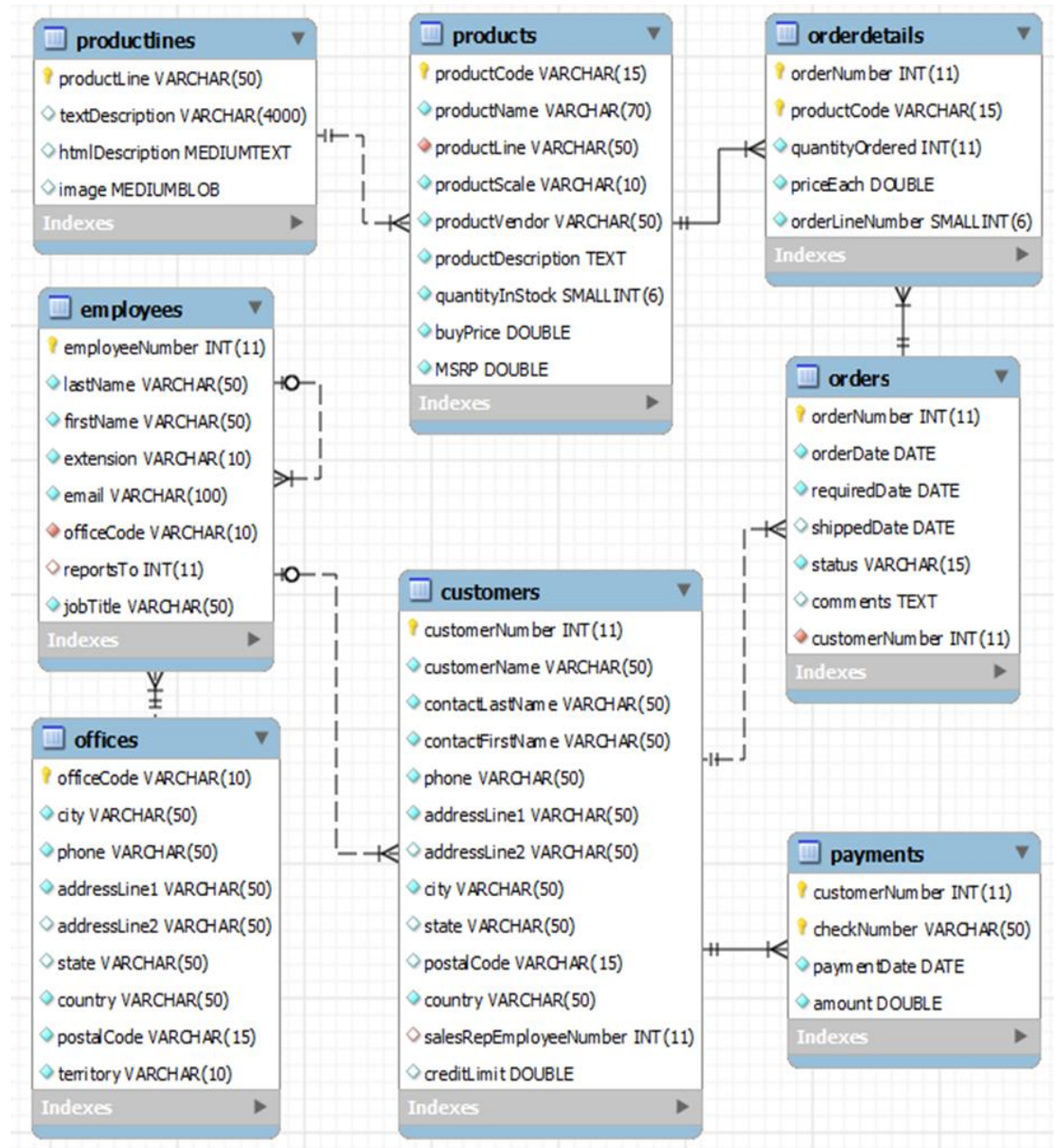
```
GRANT SELECT ON "FACTURA" TO "B3"
```

2.c) (1/3 valor) Como B1 cancela todos os privilégios de acesso a FACTURA.

Na maioria dos sistemas de banco de dados, a propagação de revogação é o comportamento padrão; assim a palavra-chave **cascade** poderá ser omitida, como se fez a seguir:

```
REVOKE SELECT ON FACTURA FROM B2
```

3) Considere a seguinte base de dados e escreva em Álgebra Relacional, considerando os operadores de σ , Π , \bowtie e G para as funções agregadoras de G_{sum} , G_{count} , $G_{average}$, etc.



a) (1/2 valor) Qual o valor monetário das encomendas por estado?
 Select orders.status, sum(orderdetails.quantityOrdered*orderdetails.priceEach) from
 orders, orderdetails
 where orders.orderNumber=orderdetails.orderNumber
 group by orders.status

orders.statusGsum(orderdetails.quantityOrdered*orderdetails.priceEach) (orders \bowtie orderdetails)

b) (1/2 valor) Qual o número do cliente com a maior soma de pagamentos no ano 2017?

```
Select payments.customerNumber, sum(payments.amount) AS SOMA
from payments
where SOMA=(SELECT MAX(SOMA)
from payments
where YEAR(paymentDate)='2017'
GROUP BY payments.customerNumber)
group by payments.customerNumber
```

```
temp1 ← payments.customerNumber  $\mathbf{Gsum}$  payments.amount
```

```
temp2 ←  $\sigma_{YEAR(payments.paymentDate)='2017'}$  (temp1)
```

```
Result ← payments.customerNumber  $\mathbf{Gsum}$  payments.amount (payments.customerNumber  $\mathbf{Gmax}$  (temp2))
```

4) (1/2 valor) Implemente em SQL o operador FACTOR para o exemplo *roster* (*passenger, class, smoking*) pp. 38 da bibliografia de David Maier do Capítulo 3.

The relations roster1 and roster2 are the result of FACTOR(roster; CLASS, SMOKING; LINK)

```
-- Criação da tabela roster2
```

```
CREATE TABLE roster2
```

```
AS (SELECT DISTINCT CLASS, SMOKING
```

```
FROM roster ORDER BY CLASS DESC, SMOKING DESC);
```

```
-- Inserção da coluna LINK na primeira posição
```

```
ALTER TABLE roster2
```

```
ADD LINK INT NOT NULL AUTO_INCREMENT PRIMARY KEY FIRST;
```

```
-- Criação da tabela roster1
```

```
CREATE TABLE roster1
```

```
AS (SELECT roster.PASSENGER, roster2.LINK
```

```
FROM roster, roster2
```

```
WHERE roster.CLASS=roster2.CLASS
```

```
AND roster.SMOKING=roster2.SMOKING)
```

5) Projeto de Bases de Dados

Numa empresa multinacional, dispersa por vários continentes, vários países e várias cidades existem escritórios com empregados. Os dados dos empregados incluem o nome, contactos (e-mails e telemóvel), função atual, salário atual e escritório atual. Cada empregado tem um histórico. Os empregados podem mudar de escritório, de função e de salário durante a sua vida na empresa. Podem ainda sair e voltar a entrar na empresa passado algum tempo. Para cada função, existe um título e um salário máximo e mínimo. Pretende-se manter a informação atual e histórica de todos os empregados da empresa multinacional.

5.a) (1/2 valor) Comece por identificar as diferentes entidades do Modelo Entidade-Relação. De seguida, identifique os relacionamentos entre as entidades do Modelo Entidade-Relação. Dê um nome ao relacionamento. Classifique cada relacionamento quanto à cardinalidade (1:1, 1:N, N:N). De seguida, desenhe o diagrama do Modelo Entidade-Relação. Identifique os relacionamentos e os atributos das entidades.

Entidades:

escritorio(id_escritorio, nome_escritorio, continente, pais, cidade)

empregados(id_empregado, nome_empregado, email, telemovel, funcao_atual, salario_atual, id_escritorio)

historico(data_evento, id_empregado, id_escritorio, id_funcao, data_saida, data_entrada)

funcao(id_funcao, titulo, salario_maximo, salario_minimo)

Relações:

escritorio-empregados 1:N têm

empregados-historico 1:N têm

empregados-funcao 1:N têm

Relação ternária com nome empregados(id_empregado, nome_empregado, email, telemovel, funcao_atual, salario_atual, id_escritorio).

5.b) (1/2 valor) Desenhe a base de dados relacional correspondente ao modelo anterior, em que 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. A base de dados não deve exceder as 7 tabelas. Se utilizar ligações que denotem transitividade, deve justificá-la convenientemente.

4 tabelas.

5.c) (1/2 valor) Identifique dois fatores críticos de sucesso e escreva as respetivas consultas em SQL.

- Quantos empregados têm a empresa? (total de empregados no mundo)
- Qual é a massa salarial? (soma dos salários pagos aos empregados actualmente)