

U.C. 21103

Sistemas de Gestão de Bases de Dados

2020-2021

Resolução e Critérios de Correção

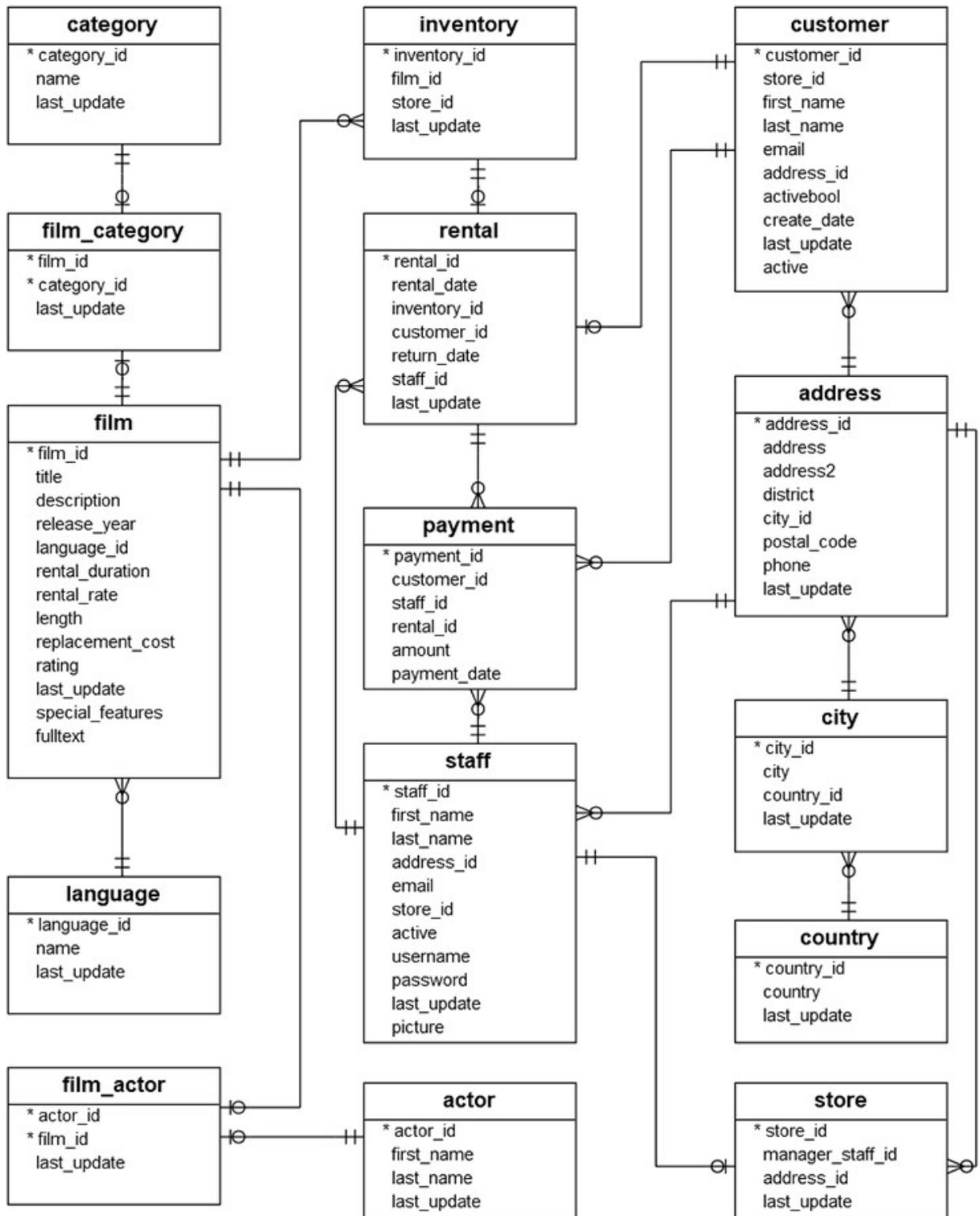
INSTRUÇÕES

- 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. Cada pergunta com uma ou mais páginas, deve ser iniciada numa nova página. 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>. Penalização de 10% a 100%.
- 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 será penalizada.
- 9) Critérios de correção gerais: todas as respostas devem ser justificadas, incluir imagens e exemplos com vista a clarificar os argumentos expostos.

Vetor Cotações

1 2 3, 4 5 pergunta
10 10 10, 10 10 décimas

Neste e-fólio B considere a seguinte base de dados de Aluguer de DVD:



1) (1 valor) Prática em MySQL de planos de execução de consultas

Em primeiro lugar instale no seu computador o SGBD MySQL. De seguida considere a base de dados de Aluguer de DVD apresentada no início do e-fólio B. Esta base de dados tem o nome Sakila dos exemplos do MySQL.

1.a) Escreva uma consulta que devolva o nome dos clientes que alugaram mais de 5 filmes.

1.b) Mostre o plano de execução disponível no MySQL e utilize o comando EXPLAIN. Analise e comente os resultados.

Resposta:

1.a) Aqui usamos o identificador em vez do nome.

```
SELECT C.customer_id, COUNT(R.rental_id) as totalRental
FROM rental R, customer C
WHERE C.customer_id = R.customer_id
GROUP BY R.customer_id
HAVING totalRental > 5
```

1.b)

A tabela Customer tem 599 linhas

A tabela Rental tem 16.044 linhas

No Nested-Loop da junção teremos:

1 Ciclo externo Customer

2 Ciclo interno Rental

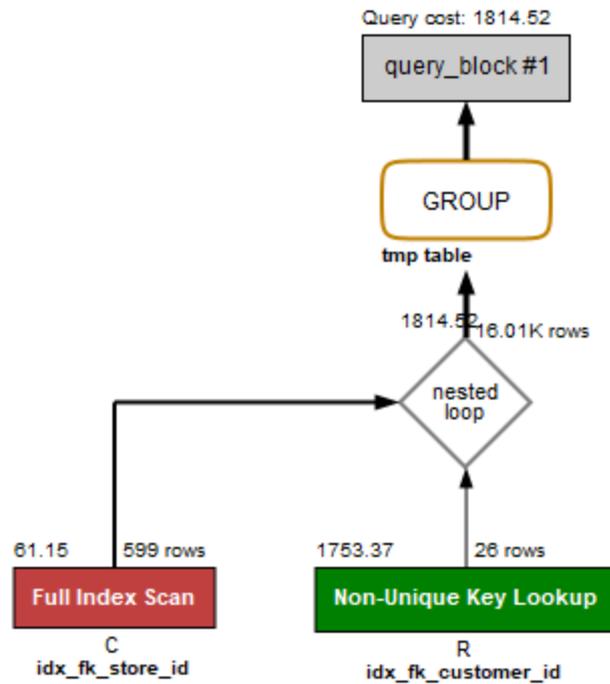
3 Junção Customer com Rental

4 fimCiclo

5 fimCiclo

Em Rental temos uma estimativa de 26 linhas por cada linha de Customer: $16.044/599=26,7$ linhas.

Tabular Explain					
id	select_type	table	type	possible_keys	key
1	SIMPLE	C	index	PRIMARY	idx_fk_store_id
1	SIMPLE	R	ref	rental_date,idx_fk_customer_id	idx_fk_customer_id
		key_len	ref	rows	Extra
		1		599	Using index; Using temporary
		2	sakila.C.customer_id	26	Using index



Com este plano de execução podemos verificar que é feita uma leitura completa à tabela Customer (alias C) resultando em 599 linhas com o custo de 61.15. O custo é encontrado através de uma heurística.

O custo total de leitura da tabela Rental (alias R) é de 1753.27.

A junção tem um custo de 1814.42 (i.e. 1753.27 + 61.15) e resultam 16.01K linhas, o total de linhas da tabela Rental.

Critérios de correção:

- 1.a) 2 décimas, consulta SQL
- 1.b) 8 décimas, plano execução e comentários
- erros, omissões ou redundâncias: -20% a -100%

2) (1 valor) Capítulo 15, Concurrency Control

2.a) Defina o protocolo 2-PL relativamente à seriabilidade, à recuperação e ao 'deadlock':

- (i) o protocolo 2-PL (não) garante a seriabilidade;
 - (ii) o protocolo 2-PL (não) evita o 'deadlock';
 - (iii) o protocolo 2-PL (não) é recuperável em caso de falha;
- Justifique as respostas.

2.b) Considere o protocolo 2-PL e acrescente os operadores X-lock(), S-lock() e Unlock(). Existe a possibilidade de 'deadlock'? Justifique a resposta.

T1:
read(A);
read(B);
if A==0 then B = B + 1;
write(B).

T2:
read(B);
read(A);
if B==0 then A = A + 1;
write(A).

Resposta:

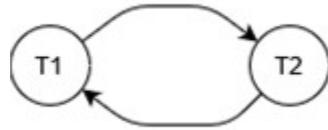
2.a)

- (i) o protocolo 2-PL garante a seriabilidade;
 - recorrendo a bloqueios para aceder aos recursos em duas fases:
Na fase de crescimento, onde podem ser pedidos bloqueios, não podendo haver desbloqueios;
 - Na fase de encolhimento, onde podem haver desbloqueios, não podem ser obtidos novos bloqueios;

- (ii) o protocolo 2-PL não evita o 'deadlock';
 - a alínea 2.b exemplifica que o protocolo 2-PL não evita o 'deadlock';

- (iii) com o protocolo 2-PL nem sempre é possível recuperar as transações em caso de falha;
 - no caso da transação já tiver realizado o 'commit' é possível a sua recuperação;

2.b) O 'deadlock' ocorre quando duas transações aguardam os recursos uma da outra. T1 aguarda recursos bloqueados por T2 e T2 aguardar recursos bloqueados por T1. Esquemáticamente teremos:



Dada a seguinte matriz de compatibilidade de 'locks':

<i>State of the lock</i>	<i>Lock request type</i>	
	<i>Shared</i>	<i>Exclusive</i>
<i>Shared</i>	Yes	No
<i>Exclusive</i>	No	No

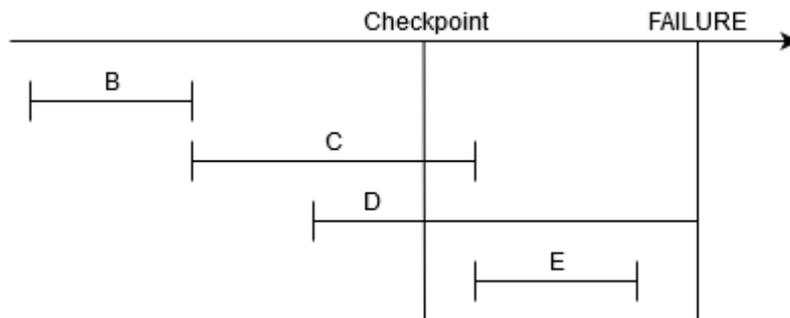
Existe a possibilidade de 'deadlock', como por exemplo:

T1	T2	
S-lock(A), read(A)		
	S-lock(B), read(B)	
	S-lock(A),read(A)	
	if B==0 then A=A+1	
	X-lock(A), write(A)	Não é possível obter o lock exclusivo. Fica a guardar
S-lock(B), read(B)		
if A==0 then B=B+1		
X-lock(B), write(B)		Não é possível obter o lock exclusivo. Fica a aguardar, gerando deadlock.
		Deadlock

Critérios de correção:

- a) 4 décimas, definições
- b) 6 décimas, exemplificar 'deadlock'
- erros, omissões ou redundâncias: -20% a -100%

Resposta: As transações definidas ficariam representadas da seguinte forma na linha do tempo:



As transações que foram confirmadas antes do último checkpoint são ignoradas (transação B) porque as operações já foram gravadas em disco.

As transações confirmadas depois do último checkpoint (e antes da falha), são refeitas (transações C e E).

As transações não confirmadas na altura da falha, são desfeitas (transação D).

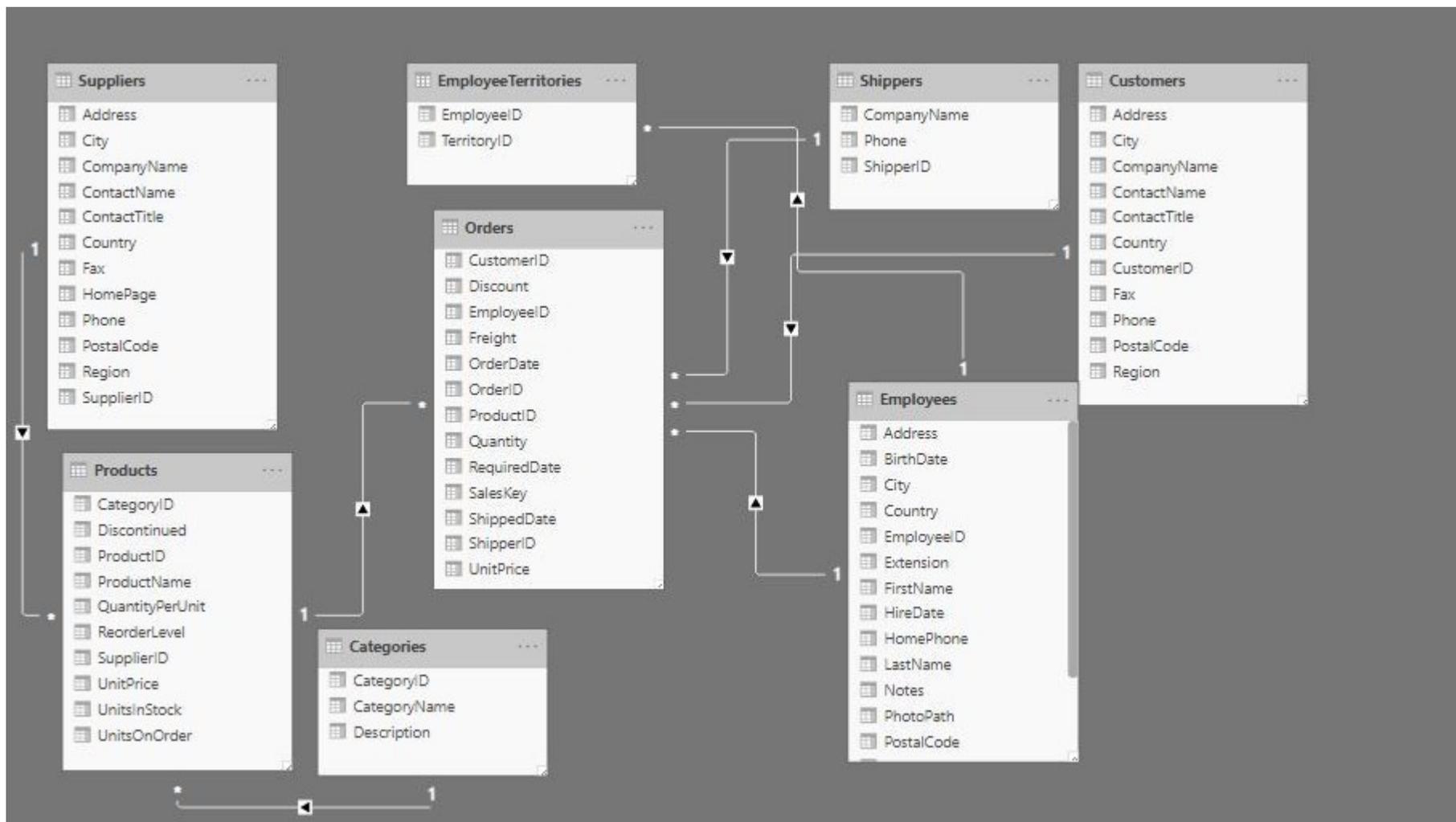
Checkpoint	Início Redo-Phase	
Start B	Undo-list = B	
Update B	Redo B	
Commit B	Undo-list = vazia	
Start C	Undo-list = C	
Update C	Redo C	
Start D	Undo-list = C, D	
Update D	Redo D	
Checkpoint	Checkpoint (C, D)	
Commit C	Undo-list= D	Fim Undo-Phase
Start E	Undo-list= D, E	Undo-list= vazia
Update E	Redo E	Undo D
Commit E	Undo-list= D	Início Undo-Phase
FAIL
Redo C		Registos acrescentados na Redo-Phase e Undo-Phase
Redo D		
Redo E		
Undo D		
Abort D		

No caso da transação D as operações realizadas antes do checkpoint, gravadas em disco, devem ser desfeitas fisicamente. Já as operações realizadas depois do checkpoint são desfeitas ao nível lógico.

Crítérios de correção:

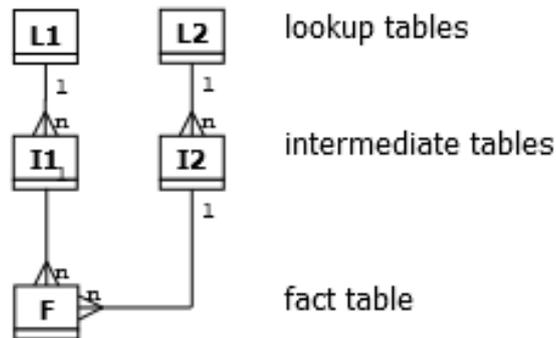
- 4 décimas, fase redo
- 4 décimas, fase undo e novos registos
- 2 décimas, representação no tempo
- erros, omissões ou redundâncias: -20% a -100%

Para as perguntas 4) e 5) **Desnormalização e Data Warehousing**, considere a seguinte base de dados:



4) (1 valor) Desnormalização

4.a) Reutilize a base de dados transacional na 3ª forma normal. Faça o carregamento de dados. Represente graficamente as ligações de 1:N, a tabela com uma única linha é desenhada em cima e a tabela com várias linhas é desenhada por baixo. Depois de representar as tabelas classifique-as segundo a tipologia indicada.



4.b). Encontre a 1FD (1ª forma desnormalizada) e a 2FD (1ª forma desnormalizada). Justifique a resposta.

Resposta:

4.a) Nova representação gráfica e tipologia das tabelas

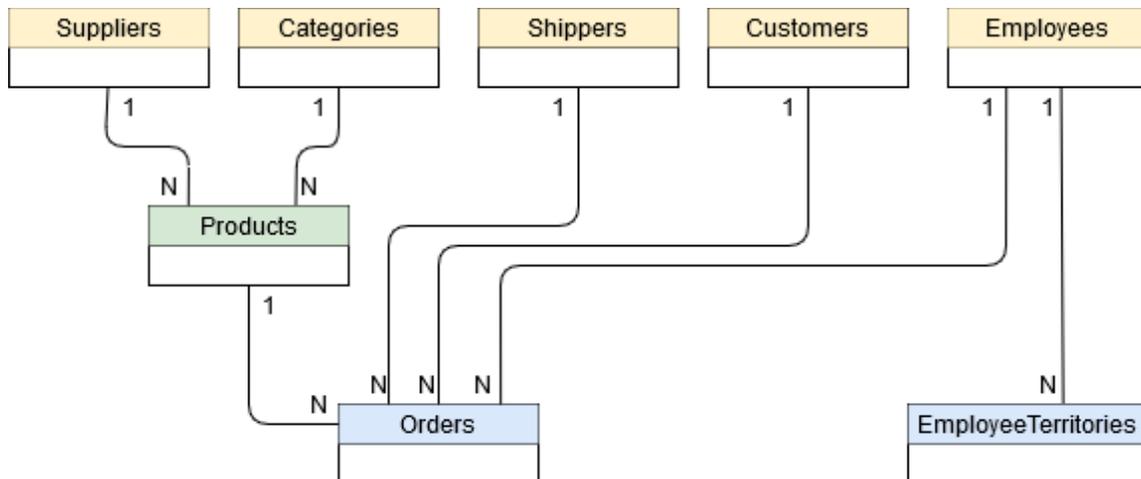
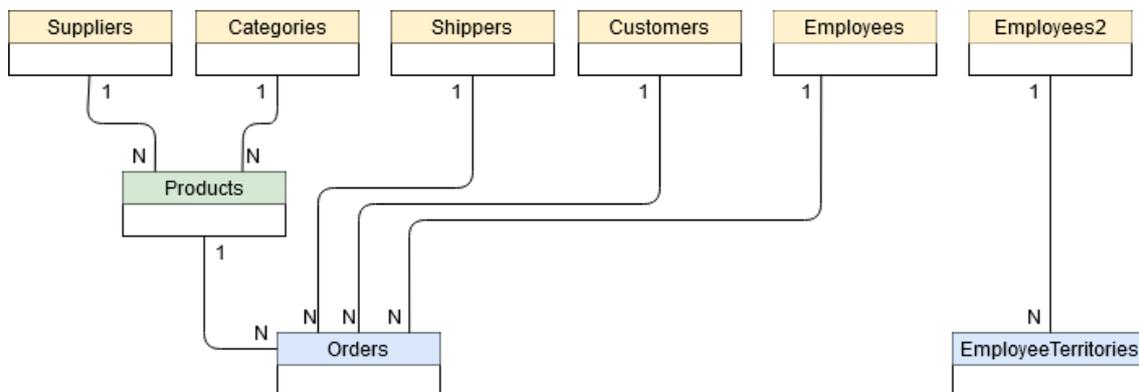


Tabela	Tipologia
Suppliers	lookup
Categories	lookup
Shippers	lookup
Customers	lookup
Employees	lookup
Products	intermediate
Orders	fact
EmployeeTerritories	fact

4.b)

Para estar na 1ª forma desnormalizada (1DF), a estrutura tem de ter a forma de uma poly-tree, em que apenas existe um caminho entre duas tabelas. No exemplo dado já se encontra na 1DF, não existindo caminhos múltiplos.

Para estar na 2ª forma desnormalizada (2DF), a estrutura tem de ter a forma de uma ou múltiplas árvores. Para separar em duas árvores, foi criado um alias para a tabela Employees (Employees2):



Critérios de correção:

- a) 5 décimas, base dados na 3FN e tipologia das tabelas
- b) 5 décimas, 1 e 2 DF
- erros, omissões, redundâncias ou apresentação desadequada: -20% a -100%

5) (1 valor) Data Warehouse

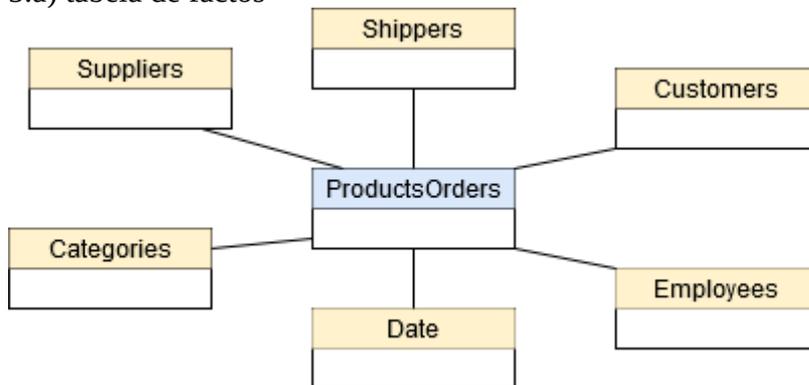
5.a) Pretendemos desenhar um “Data Warehouse” relacional em estrela ou em constelação, i.e. com duas ou mais estrelas com a maior granularidade possível. Defina a(s) tabela(s) de factos e mostre a tabela depois da desnormalização dos dados. Defina as dimensões com os níveis de agregação para o “Data Warehouse” relacional. Apresente a(s) tabela(s) de factos associada às dimensões. Quantas tabelas de factos encontrou? Preencha a 'bus matrix' (ou 'business matrix'). Justifique a resposta.

5.b) Crie a matriz oferta-procura com os seguintes requisitos:
 - quais as categorias de produtos para cada cliente;
 - quais os transportadores ('shippers') de cada cliente.

5.c) Traduza os requisitos para SQL com Pivot Tables utilizando pelo menos duas dimensões (OLAP). Justifique a resposta.

Resposta:

5.a) tabela de factos



5.b) Matriz oferta-procura

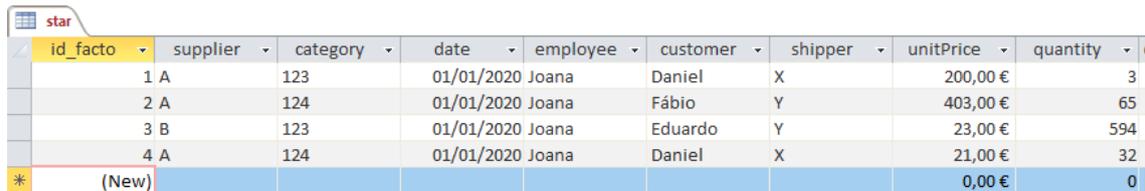
#	fact table	type	Shippers	Products	Customers	Employees	requirements	1
1	Orders	wf		x	x		Categorias produtos para cada cliente	x
			x		x		Transportadores de cada cliente	x

5.c) Perguntas com Pivot Table

A tabela de factos Orders/ProductsOrders é obtida através da junção de 7 tabelas: Suppliers, Categories, Products, Shippers, Customers, Employees e Orders.

Os dados desnormalizados estão armazenados numa única tabela de factos, onde se irão processar as consultas.

A figura seguinte mostra um excerto da tabela de factos Orders/ProductsOrders com os atributos que foram requeridos.



id_facto	supplier	category	date	employee	customer	shipper	unitPrice	quantity
1	A	123	01/01/2020	Joana	Daniel	X	200,00 €	3
2	A	124	01/01/2020	Joana	Fábio	Y	403,00 €	65
3	B	123	01/01/2020	Joana	Eduardo	Y	23,00 €	594
4	A	124	01/01/2020	Joana	Daniel	X	21,00 €	32
*	(New)						0,00 €	0

Para obter uma Pivot Table das Categorias de Produtos por Clientes, fazemos:

```
TRANSFORM Count(O.id_facto)
SELECT O.category
FROM Orders O
GROUP BY O.category
PIVOT O.customer;
```

Obtendo o seguinte resultado:

category	customer		
	Daniel	Eduardo	Fábio
123	1	1	
124	1		1

Critérios de correção:

- a) 3 décimas, tabela de factos
- b) 3 décimas, matriz oferta-procura
- c) 4 décimas, 2 perguntas OLAP (não existem junções, só agregações)
- erros, omissões, redundâncias ou apresentação desadequada: -20% a -100%