

## Sistemas de Apoio à Decisão

SAD20152016-RT-N.05-GRUPO-N.08-

JOSEGOUVEIA.e.RICARDOPEREIRA.e.RODRIGOVELOSA

Entrega	# Participantes	José Gouveia	Rodrigo Velosa	Ricardo Pereira	Descrição
RT06	3	X	X	X	Resolução de problemas antigos no cubo de dados
RT06	3	X	X	X	Resolução de problemas antigos no cubo de dados
RT06	3	X	X	X	Resolução de problemas antigos no cubo de dados
RT06	3	X	X	X	Resolução de problemas antigos no cubo de dados
RT06	3	X	X	X	Resolução de problemas antigos no cubo de dados
RT06	3	X	X	X	Resolução de problemas antigos no cubo de dados
RT06	3	X	X	X	Resolução de problemas antigos no cubo de dados
RT06	3	X	X	X	Resolução de problemas antigos no cubo de dados
RT06	3	X	X	X	Resolução de erros relacionado com a criação correcta do ficheiro .csv
RT06	3	X	X	X	Resolução de erros relacionado com a criação correcta do ficheiro .csv
RT06	3	X	X	X	Resolução de erros relacionado com a criação correcta do ficheiro .csv
RT06	3	X	X	X	Resolução de erros relacionado com a criação correcta do ficheiro .csv
RT06	3	X	X	X	Resolução de erros relacionado com a criação correcta do ficheiro .csv
RT07	3	X	X	X	Procura na net das bibliotecas e exemplos em java
RT07	3	X	X	X	Procura na net das bibliotecas e exemplos em java
RT07	3	X	X	X	Procura na net das bibliotecas e exemplos em java
RT07	3	X	X	X	Criação da classe dinamica em JAVA de criação de data basket
RT07	3	X	X	X	Criação da classe dinamica em JAVA de criação de data basket
RT07	3	X	X	X	Criação da classe dinamica em JAVA de criação de data basket
RT07	3	X	X	X	Criação da classe dinamica em JAVA de criação de data basket
RT07	3	X	X	X	Criação da classe dinamica em JAVA de criação de data basket
RT07	3	X	X	X	Elaboração das query's SQL para o TASKDATA1
RT07	3	X	X	X	Elaboração das query's SQL para o TASKDATA1
RT07	3	X	X	X	Elaboração das query's SQL para o TASKDATA1
RT07	3	X	X	X	Elaboração das query's SQL para o TASKDATA1
RT07	3	X	X	X	Elaboração das query's SQL para o TASKDATA1
RT08	3	X	X	X	Brainstorm sobre exemplos de valor para a empresa para os proximos TASKDATA's
RT08	3	X	X	X	Brainstorm sobre exemplos de valor para a empresa para os proximos TASKDATA's
RT08	3	X	X	X	Elaboração das query's SQL para o TASKDATA2
RT08	3	X	X	X	Elaboração das query's SQL para o TASKDATA2
RT08	3	X	X	X	Elaboração das query's SQL para o TASKDATA2

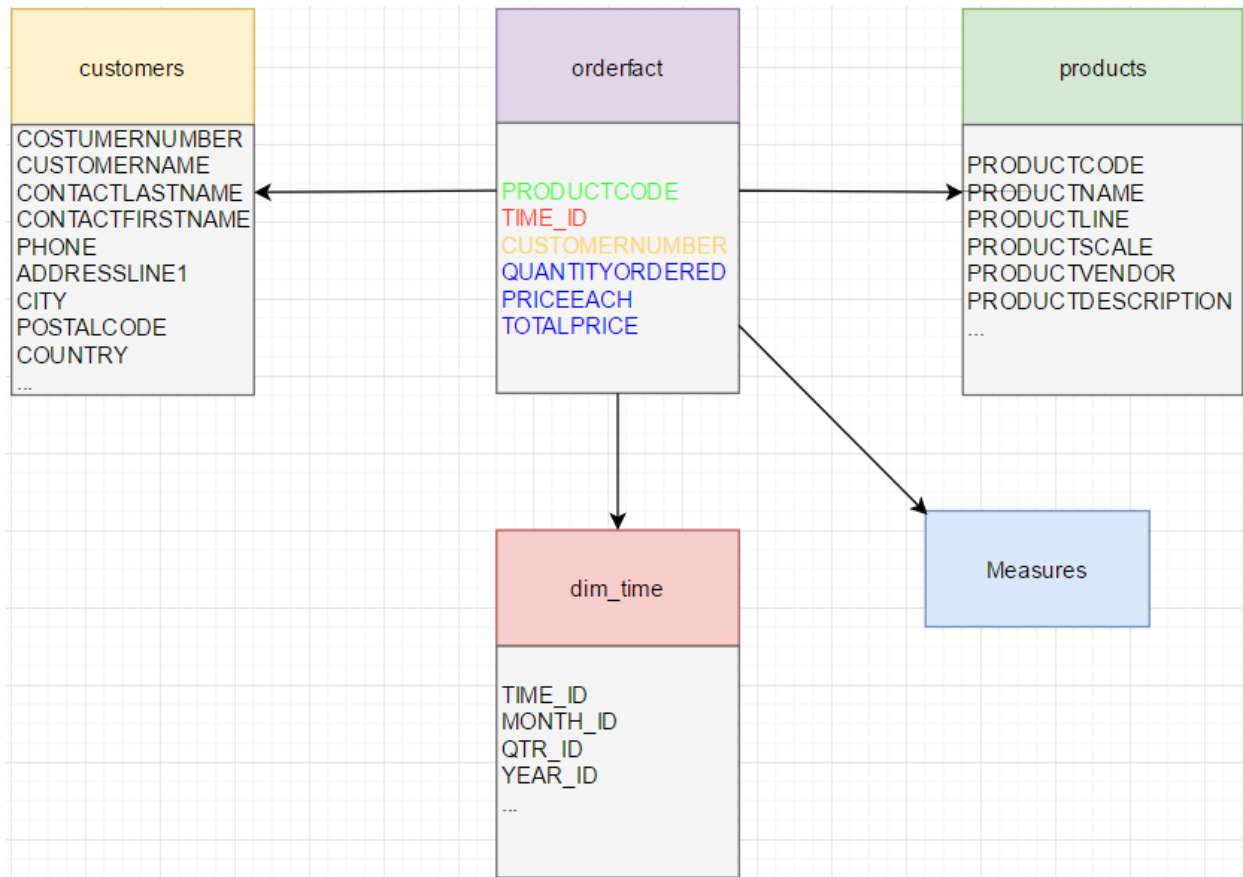
**Sistemas de Apoio à Decisão**

SAD20152016-RT-N.05-GRUPO-N.08-

JOSEGOUEIA.e.RICARDOPEREIRA.e.RODRIGOVELOSA

RT08	3	X	X	X	Elaboração das query's SQL para o TASKDATA3
RT08	3	X	X	X	Elaboração das query's SQL para o TASKDATA3
RT08	3	X	X	X	Elaboração das query's SQL para o TASKDATA3
RT08	3	X	X	X	Elaboração do código JAVA para correr as transformações automaticamente
RT08	3	X	X	X	Elaboração do código JAVA para correr as transformações automaticamente
RT08	3	X	X	X	Elaboração do código JAVA para correr as transformações automaticamente
RT08	3	X	X	X	Elaboração do código JAVA para correr as transformações automaticamente
RT08	3	X	X	X	Elaboração do código JAVA para correr as transformações automaticamente
RT08	3	X	X	X	Elaboração do código JAVA para correr as transformações automaticamente
RT08	3	X	X	X	Correcção de erros de transformação para mdb
RT08	3	X	X	X	Correcção de erros de transformação para mdb
RT08	3	X	X	X	Correcção de erros de transformação para mdb
RT08	3	X	X	X	Prints para o relatorio
RT08	3	X	X	X	Prints para o relatorio
RT08	3	X	X	X	Finalização do Relatorio
RT08	3	X	X	X	Finalização do Relatorio
RT08	3	X	X	X	Finalização do Relatorio
RT08	3	X	X	X	Finalização do Relatorio

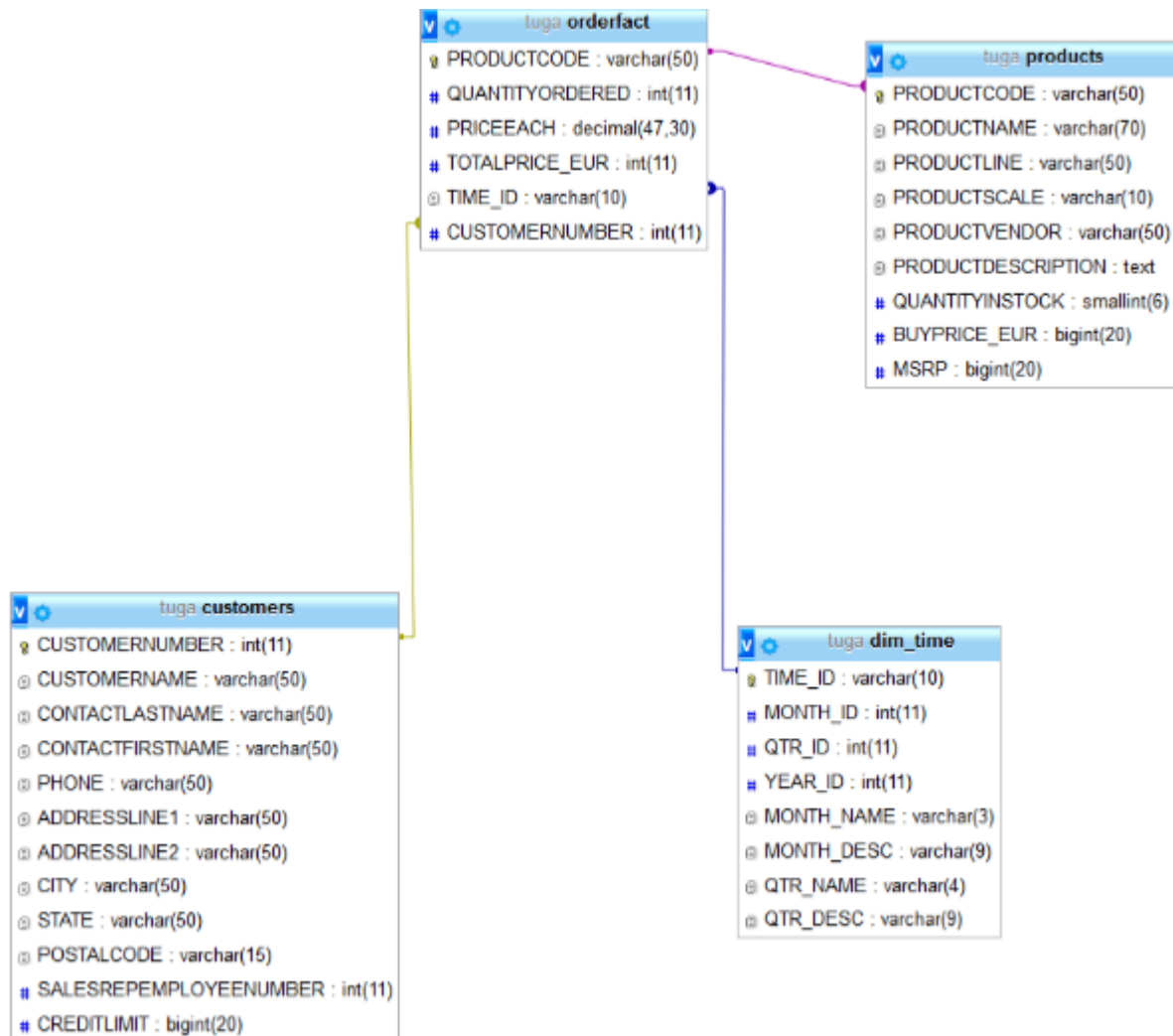
## (2) Esquema em estrela



### (3) Modelos E-R

Para a realização destes três modelos decidimos escolher três países: França, Estados Unidos e Japão. A diferença entre os três modelos vai estar no atributo TOTALPRICE visto que será utilizado Euro, Dollars e Ienes nestes três países respetivamente.

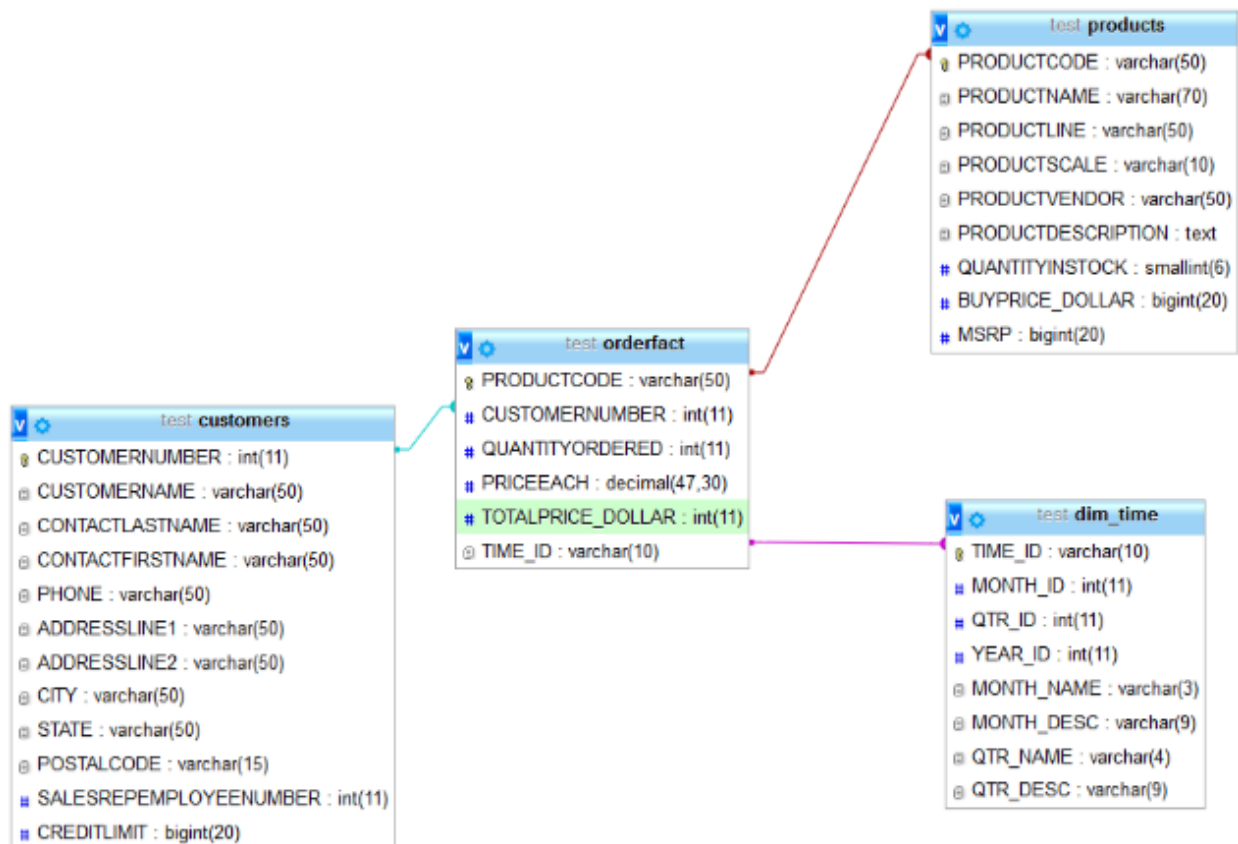
#### (3.1) França



Diferenças:

**OrderFact** – remoção dos atributos: ORDERDATE, REQUIREDDATE, SHIPPEDDATE, STATUS, COMMENTS, CUSTOMERNUMBER, QTR\_ID, MONTH\_ID e YEAR\_ID.

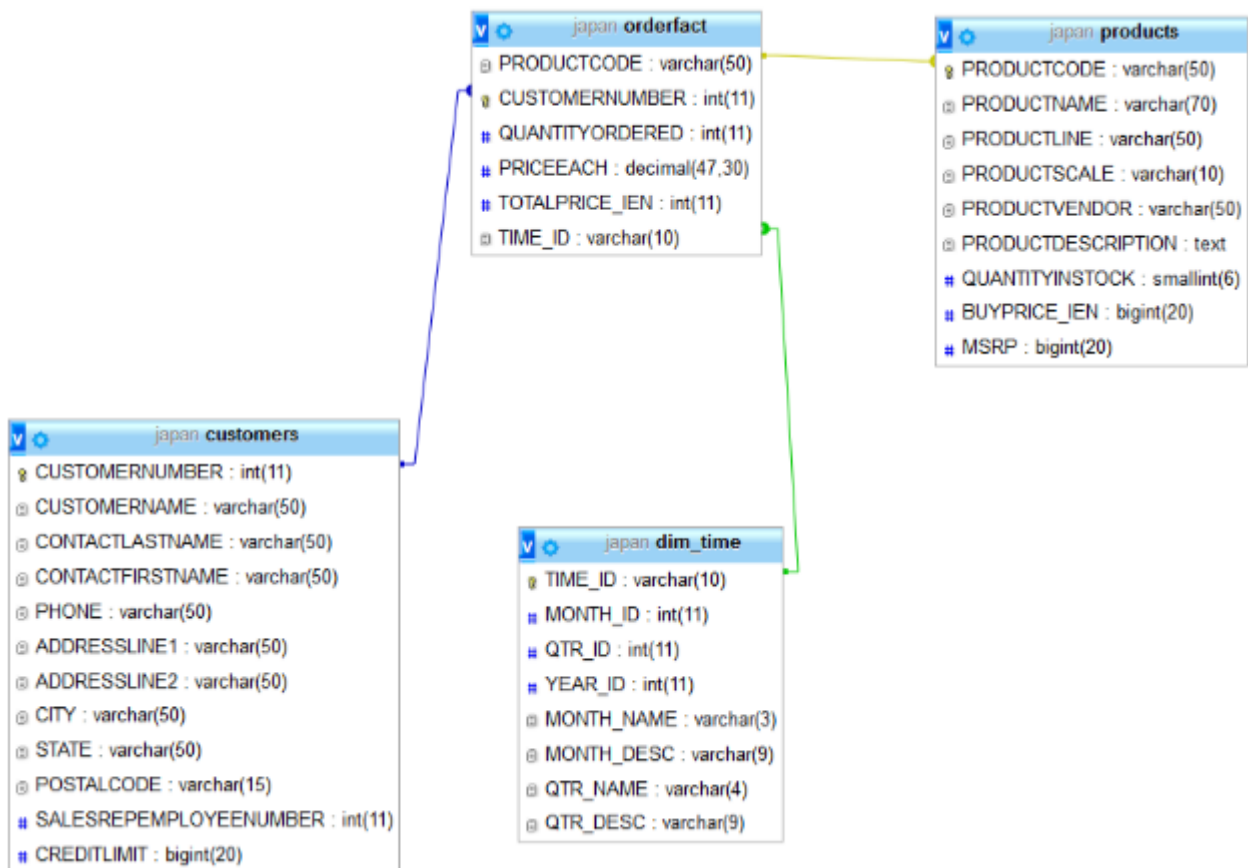
### (3.2) Estados Unidos



Diferenças:

**OrderFact** – remoção dos atributos: ORDERDATE, REQUIREDDATE, SHIPPEDDATE, STATUS, COMMENTS, CUSTOMERNUMBER, QTR\_ID, MONTH\_ID e YEAR\_ID.

### (3.3) Japão



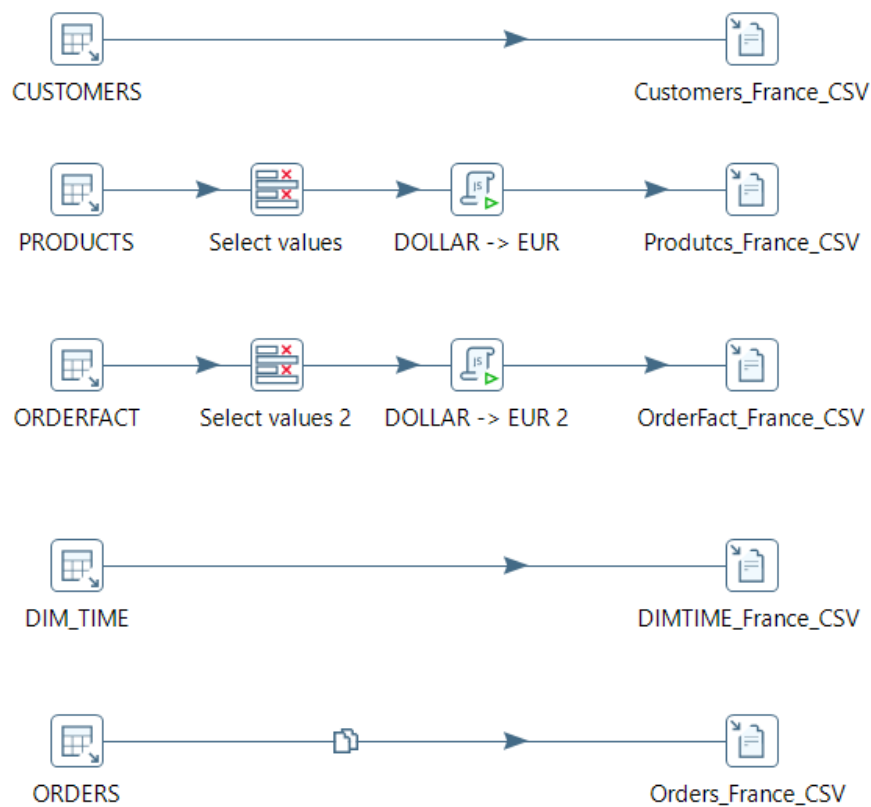
Diferenças:

**OrderFact** – remoção dos atributos: ORDERDATE, REQUIREDDATE, SHIPPEDDATE, STATUS, COMMENTS, CUSTOMERNUMBER, QTR\_ID, MONTH\_ID e YEAR\_ID.

#### (4) Fontes de dados

Criamos as fontes de dados em Excel, CSV e Access através do Spoon do Data Integration e para tal fizemos um esquema de transformação para cada tipo de fonte.

##### (4.1) Fonte de dados CSV (França)



##### (4.1.1) Query Customers

```
SELECT
  CUSTOMERNUMBER
  , CUSTOMERNAME
  , CONTACTLASTNAME
  , CONTACTFIRSTNAME
  , PHONE
  , ADDRESSLINE1
  , ADDRESSLINE2
  , CITY
  , POSTALCODE
  , SALESREPEMPLYEENUMBER
  , CREDITLIMIT
FROM CUSTOMERS
WHERE COUNTRY = 'France'
```

#### (4.1.2) Query Products

```
SELECT
  PRODUCTCODE
  , PRODUCTNAME
  , PRODUCTLINE
  , PRODUCTSCALE
  , PRODUCTVENDOR
  , PRODUCTDESCRIPTION
  , QUANTITYINSTOCK
  , BUYPRICE
  , MSRP
FROM PRODUCTS WHERE PRODUCTCODE IN
  (SELECT PRODUCTCODE FROM ORDERFACT WHERE ORDERNUMBER IN
    (SELECT ORDERNUMBER FROM ORDERS WHERE CUSTOMERNUMBER IN
      (SELECT CUSTOMERNUMBER FROM CUSTOMERS WHERE COUNTRY = 'France'))))
```

#### (4.1.3) Query Orderfact

```
SELECT
  ORDERNUMBER
  , PRODUCTCODE
  , QUANTITYORDERED
  , PRICEEACH
  , TOTALPRICE
  , TIME_ID
  , CUSTOMERNUMBER
FROM (orderfact INNER JOIN orders on orderfact.ordernumber = orders.ordernumber)
INNER JOIN customers on orders.customernumber = customers.customernumber WHERE country = 'France'
```

#### (4.1.4) Query Dim\_Time

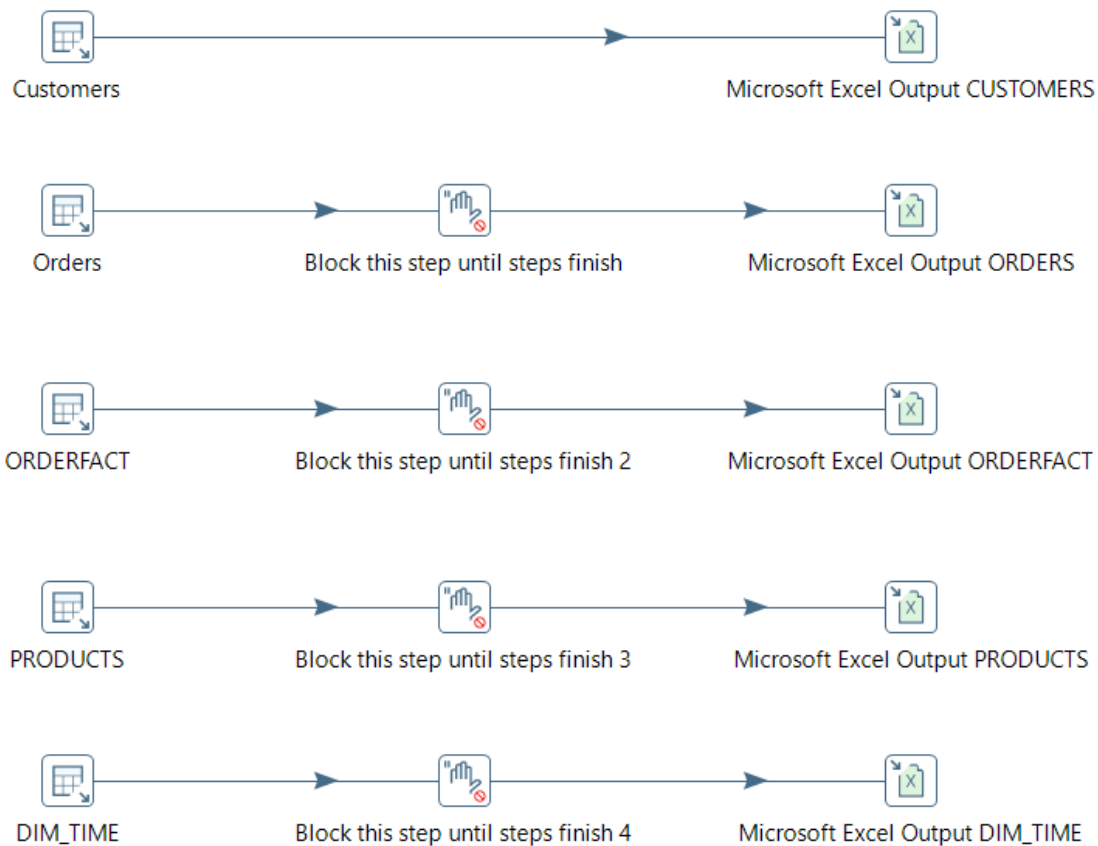
```
SELECT
  TIME_ID
  , MONTH_ID
  , QTR_ID
  , YEAR_ID
  , MONTH_NAME
  , MONTH_DESC
  , QTR_NAME
  , QTR_DESC
FROM DIM_TIME WHERE TIME_ID IN
  (SELECT TIME_ID FROM ORDERFACT WHERE ORDERNUMBER IN
    (SELECT ORDERNUMBER FROM ORDERS WHERE CUSTOMERNUMBER IN
      (SELECT CUSTOMERNUMBER FROM CUSTOMERS WHERE COUNTRY = 'France'))))
```

#### (4.1.5) Query Orders

```
SELECT
  ORDERNUMBER
  , ORDERDATE
  , REQUIREDDATE
  , SHIPPEDDATE
  , STATUS
  , COMMENTS
  , CUSTOMERNUMBER
FROM ORDERS
WHERE CUSTOMERNUMBER IN
  (SELECT CUSTOMERNUMBER FROM CUSTOMERS WHERE COUNTRY = 'France')
```

#### (4.2) Fonte de dados Excel (USA)





### (4.2.1) Query Customers

```
SELECT
  CUSTOMERNUMBER
,  CUSTOMERNAME
,  CONTACTLASTNAME
,  CONTACTFIRSTNAME
,  PHONE
,  ADDRESSLINE1
,  ADDRESSLINE2
,  CITY
,  STATE
,  POSTALCODE
,  SALESREPEMPLYEENUMBER
,  CREDITLIMIT
FROM CUSTOMERS WHERE COUNTRY = 'USA'
```

### (4.2.2) Query Orders

```
SELECT
  ORDERNUMBER
,  ORDERDATE
,  REQUIREDDATE
,  SHIPPEDDATE
,  STATUS
,  COMMENTS
,  CUSTOMERNUMBER
FROM ORDERS
WHERE CUSTOMERNUMBER IN
  (SELECT CUSTOMERNUMBER FROM CUSTOMERS WHERE COUNTRY = 'USA')
```

### (4.2.3) Query Orderfact

```
SELECT
  ORDERNUMBER
, PRODUCTCODE
, QUANTITYORDERED
, PRICEEACH
, TOTALPRICE
, TIME_ID
, CUSTOMERNUMBER
FROM (orderfact INNER JOIN orders on orderfact.ordernumber = orders.ordernumber)
INNER JOIN customers on orders.customernumber = customers.customernumber WHERE country = 'USA'
```

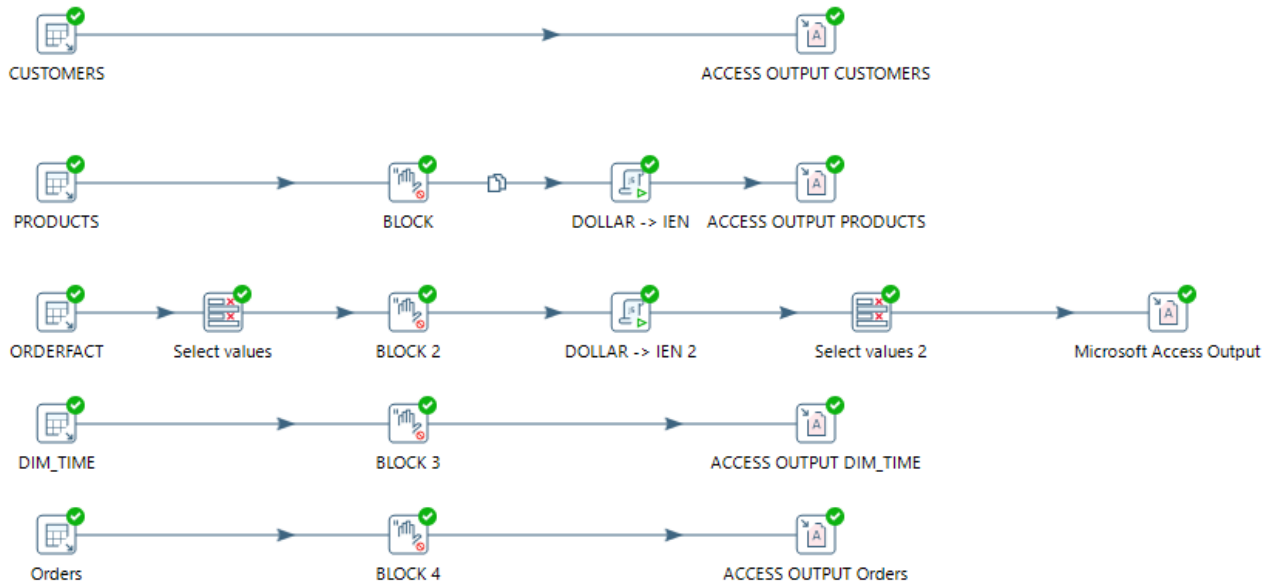
#### (4.2.4) Query Products

```
SELECT
  PRODUCTCODE
, PRODUCTNAME
, PRODUCTLINE
, PRODUCTSCALE
, PRODUCTVENDOR
, PRODUCTDESCRIPTION
, QUANTITYINSTOCK
, BUYPRICE
, MSRP
FROM PRODUCTS WHERE PRODUCTCODE IN
(SELECT PRODUCTCODE FROM ORDERFACT WHERE ORDERNUMBER IN
(SELECT ORDERNUMBER FROM ORDERS WHERE CUSTOMERNUMBER IN
(SELECT CUSTOMERNUMBER FROM CUSTOMERS WHERE COUNTRY = 'USA')))
```

#### (4.2.5) Query Dim\_Time

```
SELECT
  TIME_ID
, MONTH_ID
, QTR_ID
, YEAR_ID
, MONTH_NAME
, MONTH_DESC
, QTR_NAME
, QTR_DESC
FROM DIM_TIME WHERE TIME_ID IN
(SELECT TIME_ID FROM ORDERFACT WHERE ORDERNUMBER IN
(SELECT ORDERNUMBER FROM ORDERS WHERE CUSTOMERNUMBER IN
(SELECT CUSTOMERNUMBER FROM CUSTOMERS WHERE COUNTRY = 'USA')))
```

#### (4.3) Fonte de dados Access (Japão)



## (4.3.1) Query Customers

```

SELECT
  CUSTOMERNUMBER
  / CUSTOMERNAME
  / CONTACTLASTNAME
  / CONTACTFIRSTNAME
  / PHONE
  / ADDRESSLINE1
  / ADDRESSLINE2
  / CITY
  / STATE
  / POSTALCODE
  / SALESREPEMPOYEEENNUMBER
  / CREDITLIMIT
FROM CUSTOMERS
WHERE COUNTRY = 'Japan'
    
```

## (4.3.2) Query Products

```

SELECT
  PRODUCTCODE
  / PRODUCTNAME
  / PRODUCTLINE
  / PRODUCTSCALE
  / PRODUCTVENDOR
  / PRODUCTDESCRIPTION
  / QUANTITYINSTOCK
  / BUYPRICE
  / MSRP
FROM PRODUCTS WHERE PRODUCTCODE IN
(SELECT PRODUCTCODE FROM ORDERFACT WHERE ORDERNUMBER IN
(SELECT ORDERNUMBER FROM ORDERS WHERE CUSTOMERNUMBER IN
(SELECT CUSTOMERNUMBER FROM CUSTOMERS WHERE COUNTRY = 'Japan'))))
    
```

## (4.3.3) Query Orderfact

```
SELECT
  ORDERNUMBER
  , PRODUCTCODE
  , QUANTITYORDERED
  , PRICEEACH
  , TOTALPRICE
  , TIME_ID
  , CUSTOMERNUMBER
FROM orderfact INNER JOIN
customers on customers.customernumber = orderfact.customernumber WHERE country = 'Japan'
```

#### (4.3.4) Query Dim\_Time

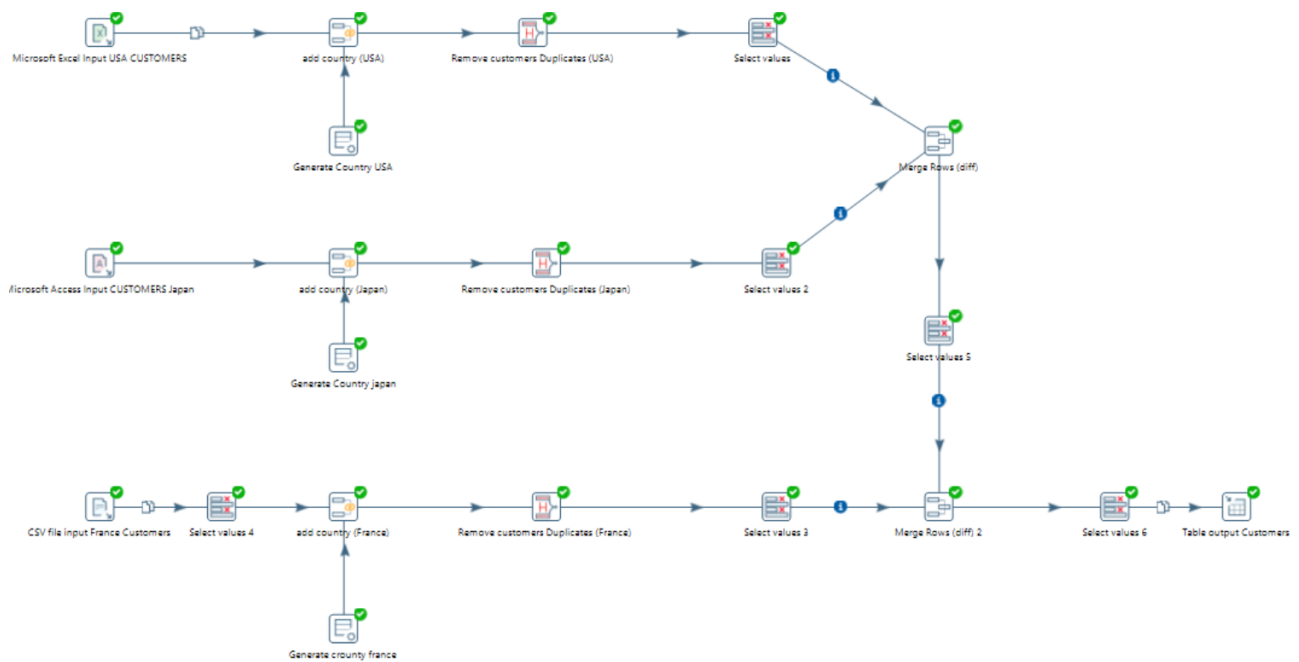
```
SELECT
  TIME_ID
  , MONTH_ID
  , QTR_ID
  , YEAR_ID
  , MONTH_NAME
  , MONTH_DESC
  , QTR_NAME
  , QTR_DESC
FROM DIM_TIME WHERE TIME_ID IN
(SELECT TIME_ID FROM ORDERFACT WHERE ORDERNUMBER IN
(SELECT ORDERNUMBER FROM ORDERS WHERE CUSTOMERNUMBER IN
(SELECT CUSTOMERNUMBER FROM CUSTOMERS WHERE COUNTRY = 'Japan')))
```

#### (4.3.5) Query Orders

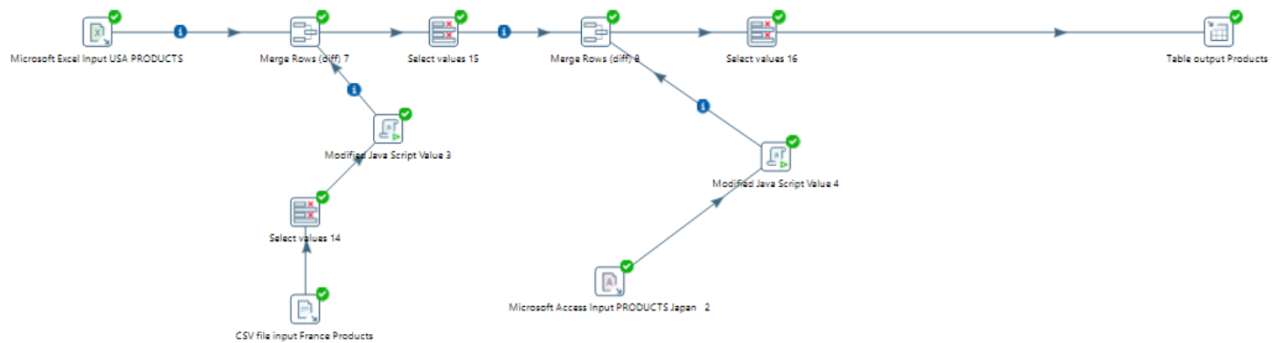
```
SELECT
  ORDERNUMBER
  , ORDERDATE
  , REQUIREDDATE
  , SHIPPEDDATE
  , STATUS
  , COMMENTS
  , CUSTOMERNUMBER
FROM ORDERS
WHERE CUSTOMERNUMBER IN
(SELECT CUSTOMERNUMBER FROM CUSTOMERS WHERE COUNTRY = 'Japan')
```

## (5) Transformação Datawarehouse

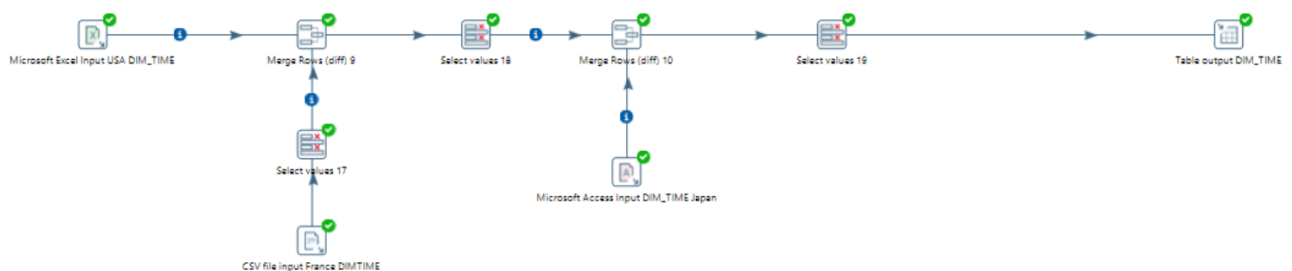
### (5.1) Transformação customers



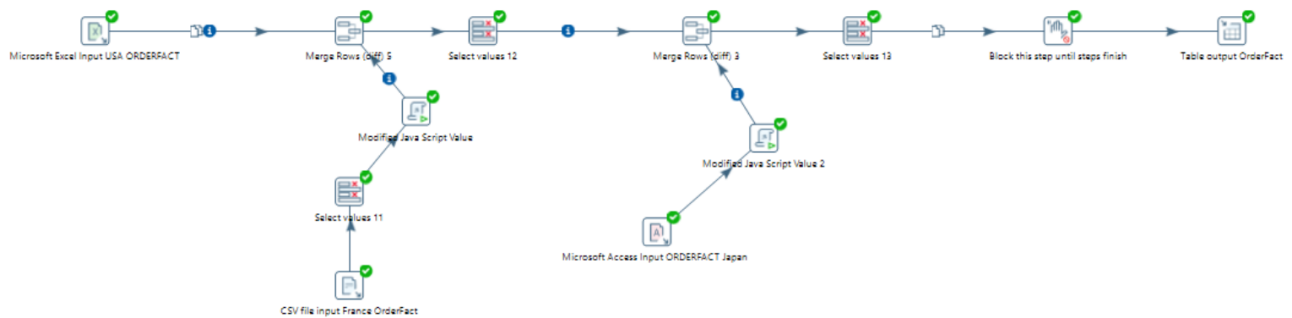
### (5.2) Transformação products



### (5.3) Transformação Dim\_Time



## (5.4) Transformação Orderfact



## 6) Repositório de dados phpmyadmin

Todos os dados foram exportados com sucesso para o phpmyadmin.

### (6.1) Tabela Products

PRODUCTCODE ▲ 1	PRODUCTNAME	PRODUCTLINE	PRODUCTSCALE	PRODUCTVENDOR	PRODUCTDESCRIPTION	QUANTITYINST
S12_3148	1969 Corvair Monza	Classic Cars	1:18	Welly Diecast Productions	1:18 scale die-cast about 10 long doors open, hood...	
S12_4473	1957 Chevy Pickup	Trucks and Buses	1:12	Exoto Designs	1:12 scale die-cast about 20 long Hood opens, Rubb...	
S18_1342	1937 Lincoln Berline	Vintage Cars	1:18	Motor City Art Classics	Features opening engine cover, doors, trunk, and f...	
S18_2319	1964 Mercedec Tour Bus	Trucks and Buses	1:18	Unimax Art Galleries	Exact replica. 100+ parts. working steering system...	
S18_2581	P-51-D Mustang	Planes	1:72	Gearbox Collectibles	Has retractable wheels and comes with a stand	
S18_2870	1999 Indy 500 Monte Carlo SS	Classic Cars	1:18	Red Start Diecast	Features include opening and closing doors. Color:...	
S18_3140	1903 Ford Model A	Vintage Cars	1:18	Unimax Art Galleries	Features opening trunk, working steering system	
S18_3782	1957 Vespa GS150	Motorcycles	1:18	Studio M Art Models	Features rotating wheels , working kick stand. Com...	
S18_4027	1970 Triumph Spitfire	Classic Cars	1:18	Min Lin Diecast	Features include opening and closing doors. Color:...	
S18_4522	1904 Buick Runabout	Vintage Cars	1:18	Exoto Designs	Features opening trunk, working steering system	
S24_1444	1970 Dodge Coronet	Classic Cars	1:24	Highway 66 Mini Classics	1:24 scale die-cast about 18 long doors open, hood...	
S24_1578	1997 BMW R 1100 S	Motorcycles	1:24	Autoart Studio Design	Detailed scale replica with working suspension and...	

**(6.2) Tabela Dim\_Time**

TIME_ID	MONTH_ID	QTR_ID	YEAR_ID	MONTH_NAME	MONTH_DESC	QTR_NAME	QTR_DESC
2003-01-06	1	1	2003	Jan	January	QTR1	Quarter 1
2003-01-10	1	1	2003	Jan	January	QTR1	Quarter 1
2003-02-24	2	1	2003	Feb	February	QTR1	Quarter 1
2003-03-10	3	1	2003	Mar	March	QTR1	Quarter 1
2003-03-25	3	1	2003	Mar	March	QTR1	Quarter 1
2003-03-26	3	1	2003	Mar	March	QTR1	Quarter 1
2003-04-01	4	2	2003	Apr	April	QTR2	Quarter 2
2003-04-04	4	2	2003	Apr	April	QTR2	Quarter 2
2003-05-07	5	2	2003	May	May	QTR2	Quarter 2
2003-05-08	5	2	2003	May	May	QTR2	Quarter 2
2003-05-20	5	2	2003	May	May	QTR2	Quarter 2
2003-05-21	5	2	2003	May	May	QTR2	Quarter 2
2003-06-03	6	2	2003	Jun	June	QTR2	Quarter 2
2003-06-16	6	2	2003	Jun	June	QTR2	Quarter 2
2003-07-01	7	3	2003	Jul	July	QTR3	Quarter 3
2003-07-02	7	3	2003	Jul	July	QTR3	Quarter 3
2003-07-04	7	3	2003	Jul	July	QTR3	Quarter 3
2003-07-10	7	3	2003	Jul	July	QTR3	Quarter 3
2003-07-24	7	3	2003	Jul	July	QTR3	Quarter 3



**(6.3) Tabela Customers**

CUSTOMERNUMBER	CUSTOMERNAME	CONTACTLASTNAME	CONTACTFIRSTNAME	PHONE	ADDRESSLINE1	CITY	POS
146	Saveley & Henriot, Co.	Saveley	Mary	78.32.5555	2, rue du Commerce	Lyon	6901
151	Muscle Machine Inc.	Young	Jeff	2125557413	4092 Furth Circle	NYC	1002
157	Diecast Classics Inc.	Yu	Kyung	2155551555	7586 Pompton St.	Allentown	7021
161	Technics Stores Inc.	Hirano	Juri	6505556809	9408 Furth Circle	Burlingame	9421
168	American Souvenirs Inc.	Franco	Sue	2035557845	149 Spinnaker Dr.	New Haven	9782
171	Daedalus Designs Imports	Ranc\u00e9	Martine	20.16.1555	184, chauss\u00e9e de Tournai	Lille	5901
172	La Come D'abondance, Co.	Bertrand	Marie	(1) 42.34.2555	265, boulevard Charonne	Paris	7501
181	Vitachrome Inc.	Frick	Michael	2125551500	2678 Kingston Rd.	NYC	1002
198	Auto-Moto Classics Inc.	Taylor	Leslie	6175558428	16780 Pompton St.	Brickhaven	5831
204	Online Mini Collectables	Barajas	Miguel	6175557555	7635 Spinnaker Dr.	Brickhaven	5831
205	Toys4GrownUps.com	Young	Julie	6265557265	78934 Hillside Dr.	Pasadena	9001
209	Mini Caravy	Citeaux	Frlu00e9d\u00e9rique	88.60.1555	24, place Kl\u00e9ber	Strasbourg	6701
219	Boards & Toys Co.	Young	Leslie	3105552373	4097 Douglas Av.	Glendale	9251
242	Alpha Cognac	Roulet	Annette	61.77.6555	1 rue Alsace-Lorraine	Toulouse	3101
250	Lvon Souveniers	Da Cunha	Daniel	+33 1 46 62	27 rue du Colonel	Paris	7551

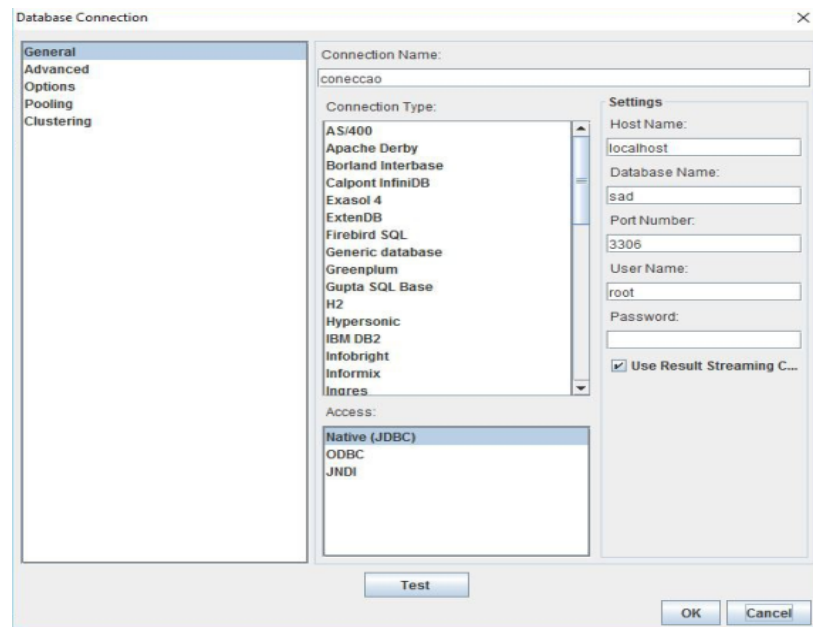
**(6.4) Tabela Orderfacts**

PRODUCTCODE	QUANTITYORDERED	PRICEEACH	TOTALPRICE	TIME_ID	CUSTOMERNUMBER
S12_3148	46	100	7366.44	2003-06-03	151
S12_4473	46	100	5723.78	2003-04-04	424
S18_1342	39	100	4808.31	2003-01-10	181
S18_2319	39	98.44000000000001	4099.455048000001	2003-04-01	172
S18_2581	35	88.979916	3114.2970600000003	2004-07-23	119
S18_2870	46	98.44000000000001	5080.685280000001	2003-05-20	103
S18_3140	24	98.44000000000001	2581.5693120000005	2004-01-02	146
S18_3782	35	48.96405600000001	1713.7419600000003	2003-05-08	350
S18_4027	25	100	3447	2003-06-03	151
S18_4522	45	86.400788	3888.0354600000005	2004-01-02	146
S24_1444	47	69.36	3259.92	2003-04-04	424
S24_1578	35	92.08077600000001	3222.8271600000003	2003-07-01	250
S24_1785	25	93.71488000000001	2342.872	2004-07-23	119
S24_2000	43	81.73473200000001	3514.593476	2003-07-01	250
S24_3949	30	71.201652	2136.0495600000004	2003-11-08	242
S24_4258	42	91.55	3845.1	2003-10-20	424
S32_1374	23	80.63220400000002	1854.5406920000003	2004-07-23	119
S32_2206	31	43.963304	1362.8624240000001	2003-05-08	350
S50_1341	38	44.681916	1697.912808	2004-07-23	119

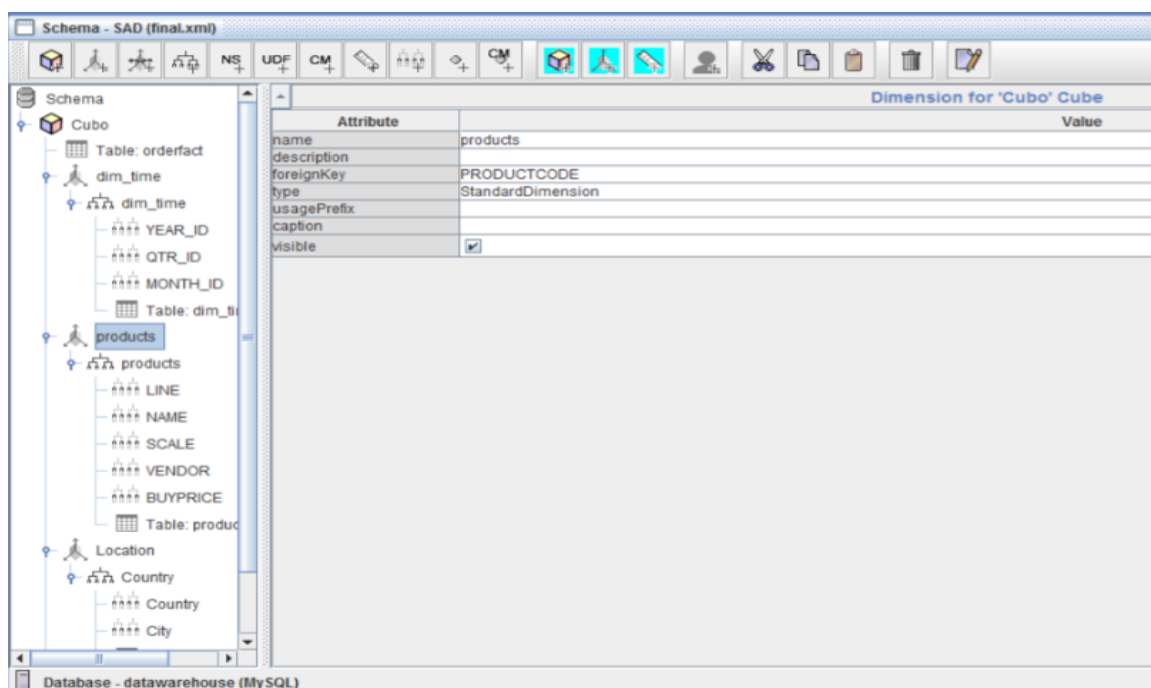
## 7. Cubo de Dados

### 7.1 Criação do Schema

Com os dados de uma amostra inseridos na base de dados, realizamos a criação do schema do cubo de dados. Primeiro fazemos a conexão a base de dados, representado na figura seguinte.

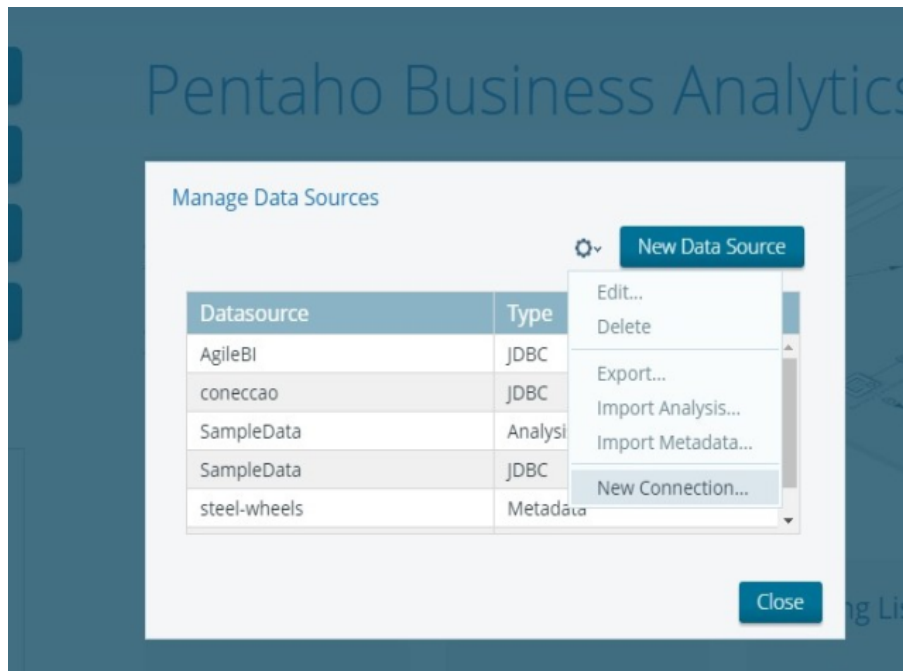


Depois, passamos ao schema do cubo de dados, onde definimos a tabela de factos com as suas três tabelas de dimensão e as métricas (como representado na seguinte figura).

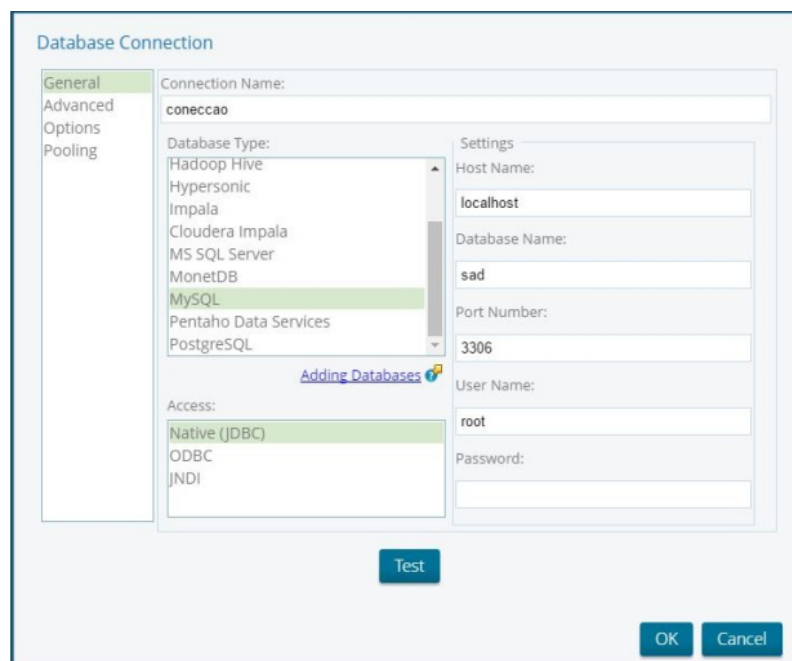


## 7.2 Criação do Cubo de Dados

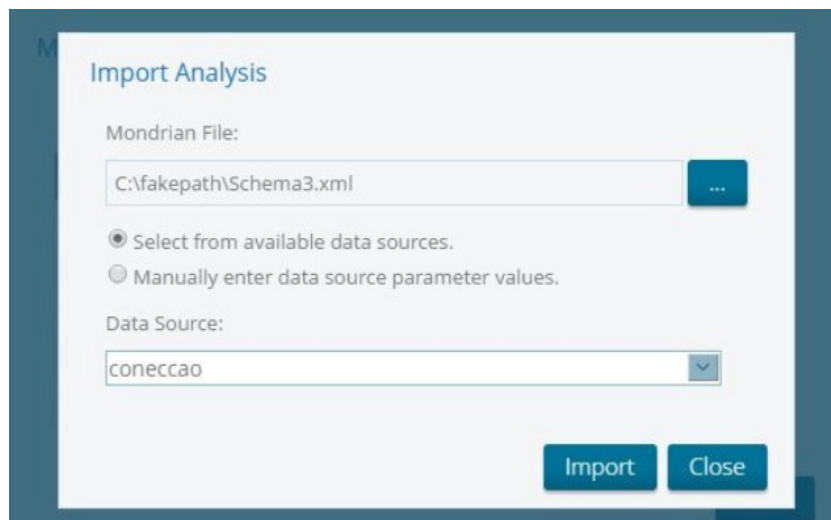
Com o schema do cubo de dados completo, passamos para a criação do cubo de dados usando o pentaho. No início começamos por aceder a página do pentaho através do link <http://localhost:8080/pentaho/Home>. A primeira configuração efetuada foi a conexão da base de dados local, criando uma nova conexão. Como na figura seguinte.



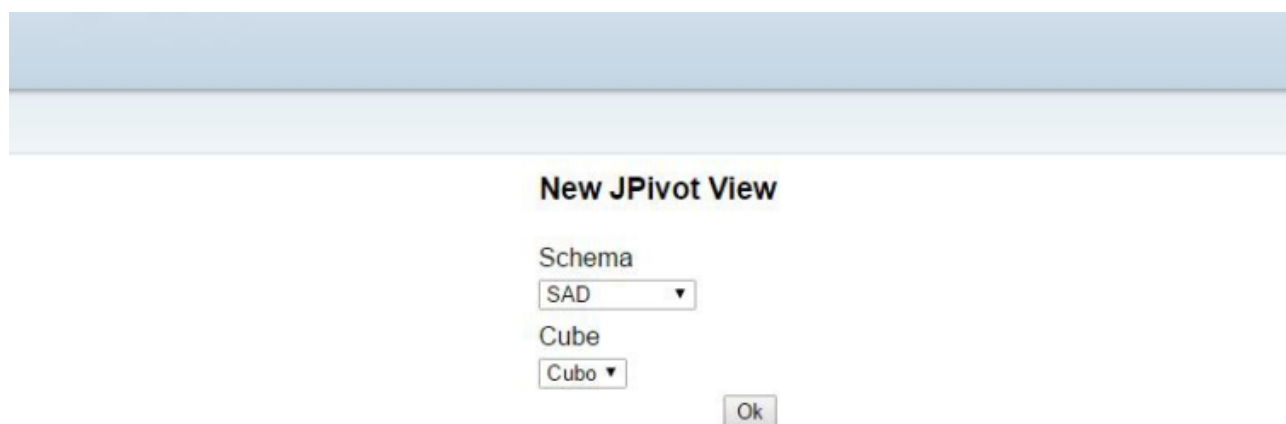
Na nova conexão foi necessário indicar os dados para fazer a ligação com a BD local (ilustrado na seguinte figura).



Depois de fazer a conexão do pentaho com a BD local, passamos por importar o schema criado anteriormente e associamos a nova conexão (como na figura seguinte).



Por ultimo a visualização do cubo de dados, criando uma nova vista Jpivot indicando o schema e o cubo de dados criado anteriormente (como na figura seguinte).



Na seguinte figura podemos ver o cubo criado corretamente.

			Measures
dim_time	products	Country	TOTALPRICE
All dim_times	All productss	All Location.Countrys	92.744,195

## 8. Operações OLAP sobre o Cubo de Dados

### 8.1. Drill Down All locations → Country → França; All products → Motorcycles

		Toulouse	9.679,999
		USA	28.450,55
	Classic Cars	All Location.Countrys	23.837,998
		France	9.764,638
		Nantes	5.080,685
		Paris	4.683,952
		USA	14.073,36
	Motorcycles	All Location.Countrys	14.910,589
		France	14.910,589
		Marseille	3.076,604
		Nantes	1.854,541
		Paris	6.737,421
		Reims	3.242,023
	1957 Vespa GS150	All Location.Countrys	1.713,742
		France	1.713,742
		Marseille	1.713,742
	1960 BSA Gold Star DBD34 1960	All Location.Countrys	3.514,593
		France	3.514,593
		Paris	3.514,593
	1982 Ducati 996 R	All Location.Countrys	1.362,862
		France	1.362,862
		Marseille	1.362,862
	1997 BMW F650 ST	All Location.Countrys	1.854,541
		France	1.854,541
		Nantes	1.854,541
	1997 BMW R 1100 S	All Location.Countrys	3.222,827
		France	3.222,827
		Paris	3.222,827
	2002 Yamaha YZR M1	All Location.Countrys	3.242,023
		France	3.242,023
		Reims	3.242,023
		All Location.Countrys	27.351,446

### 8.2. Roll Up em produtos do exemplo anterior

dim_time	products	Country	Measures
			• TOTALPRICE
All dim_times	All productss	All Location.Countrys	92.744,195
		France	64.293,645
		Lyon	6.469,605
		Marseille	3.076,604
		Nantes	26.304,586
		Paris	15.520,828
		Reims	3.242,023
		Toulouse	9.679,999
		USA	28.450,55

### 8.3. Drill Down All Dim\_times → Year → Quarters

		France	64.293,645
		Lyon	6.469,605
		Marseille	3.076,604
		Nantes	26.304,586
		Paris	15.520,828
		Reims	3.242,023
		Toulouse	9.679,999
		USA	28.450,55
2003	All products	All Location. Countrys	65.050,69
		France	36.600,14
		Marseille	3.076,604
		Nantes	5.080,685
		Paris	15.520,828
		Reims	3.242,023
		Toulouse	9.679,999
		USA	28.450,55
1	All products	All Location. Countrys	4.808,31
		USA	4.808,31
2	All products	All Location. Countrys	39.979,86
		France	20.182,72
		Marseille	3.076,604
		Nantes	5.080,685
		Paris	8.783,407
		Reims	3.242,023
		USA	19.797,14
3	All products	All Location. Countrys	6.737,421
		France	6.737,421
		Paris	6.737,421
4	All products	All Location. Countrys	13.525,099
		France	9.679,999
		Toulouse	9.679,999
		USA	3.845,1
2004	All products	All Location. Countrys	27.693,505

## 9. Criação dos ficheiros CSV

Usando o kettle do pentaho, começamos por criar três transformações com três conjunto de dados relevantes para as tarefas de data mining.

### 9.1 TASKDATA1

Relativamente ao TASKDATA1, foi retirado as colunas ordernumber, customernumber e productline das tabelas orderfact, products e customers, fazendo um inner join entre orderfact e products juntamente com um inner join com customers. Este TASKDATA é um bastante geral e permite obter regras de associação usando toda a informação colecionada em relação à todos os países, todos os anos e todos os produtos.

```
select distinct ordernumber,customernumber,productline
| from ((orderfact inner join products on products.productcode = orderfact.productcode) inner join customers on orderfact.customernumber = customers.customernumber);
```



**Group By**

Step name:

Include all rows? ☐

Temporary files directory:

TMP-file prefix:

Add line number, restart in each group ☐

Line number field name:

Always give back a result row ☐

The fields that make up the group:

#	Group field
1	ORDERNUMBER

Aggregates:

#	Name	Subject	Type
1	PRODUCTLINE	PRODUCTLINE	Concatenate strings separated by ,



## 9.2 TASKDATA2

Esta TASKDATA refere-se à todas as compras realizadas por dois clientes muito importantes para a empresa, Mini Wheels Co. e Mini Gifts Distributors Ltd. Não é colocado o Group By nem a transformação como a que mostrámos no exemplo anterior porque é igual. A única coisa que foi mudada neste exemplo foi a query.

```
select distinct ordernumber, customernumber, productline
from ((orderfact inner join products on products.productcode = orderfact.productcode) inner join customers on orderfact.customernumber = customers.customernumber)
where customername = 'Mini Wheels Co.' or customername = 'Mini Gifts Distributors Ltd.';
```

## 9.3 TASKDATA3

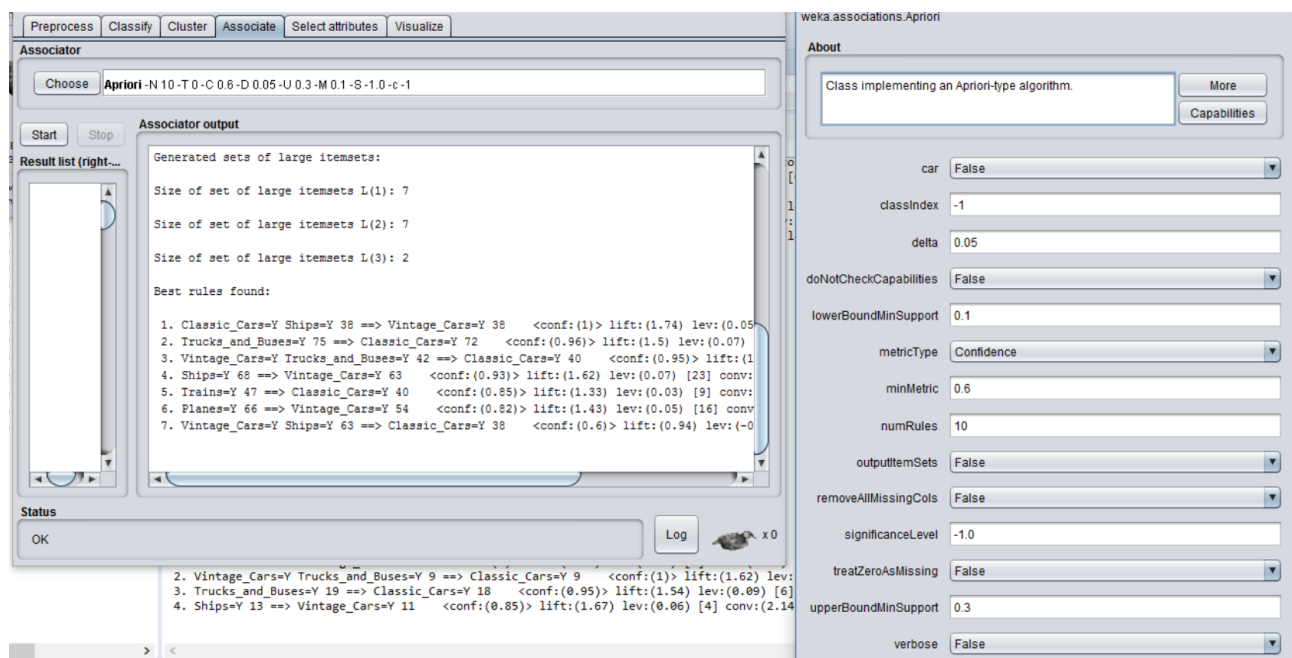
Por fim, esta TASKDATA refere-se a 3 países de forma a saber como está correndo as vendas naquela região do mundo. Para tal foram escolhidos três países vizinhos França, Espanha e Portugal. Tal como no exemplo anterior não é necessário mostrar o Group By nem a transformação.

```
select distinct ordernumber, customernumber, productline
from ((orderfact inner join products on products.productcode = orderfact.productcode) inner join customers on orderfact.customernumber = customers.customernumber)
where country = 'France' or country = 'Spain' or country = 'Portugal';
```

## 10. Regras de associação

Através dos dados analisados pelo WEKA, obtemos as seguintes regras de associação.

### 10.1 TASKDATA1



The screenshot displays the WEKA software interface with the 'Associate' tab selected. The 'Apriori' algorithm is chosen, and the 'Associator output' window shows the results of the association rule mining process.

**Associator output:**

```
Generated sets of large itemsets:
Size of set of large itemsets L(1): 7
Size of set of large itemsets L(2): 7
Size of set of large itemsets L(3): 2
Best rules found:
1. Classic_Cars=Y Ships=Y 38 ==> Vintage_Cars=Y 38 <conf:(1)> lift:(1.74) lev:(0.05)
2. Trucks_and_Buses=Y 75 ==> Classic_Cars=Y 72 <conf:(0.96)> lift:(1.5) lev:(0.07)
3. Vintage_Cars=Y Trucks_and_Buses=Y 42 ==> Classic_Cars=Y 40 <conf:(0.95)> lift:(1.62) lev:(0.07)
4. Ships=Y 68 ==> Vintage_Cars=Y 63 <conf:(0.93)> lift:(1.62) lev:(0.07) [23] conv:
5. Trains=Y 47 ==> Classic_Cars=Y 40 <conf:(0.85)> lift:(1.33) lev:(0.03) [9] conv:
6. Planes=Y 66 ==> Vintage_Cars=Y 54 <conf:(0.82)> lift:(1.43) lev:(0.05) [16] conv:
7. Vintage_Cars=Y Ships=Y 63 ==> Classic_Cars=Y 38 <conf:(0.6)> lift:(0.94) lev:(-0.05)
```

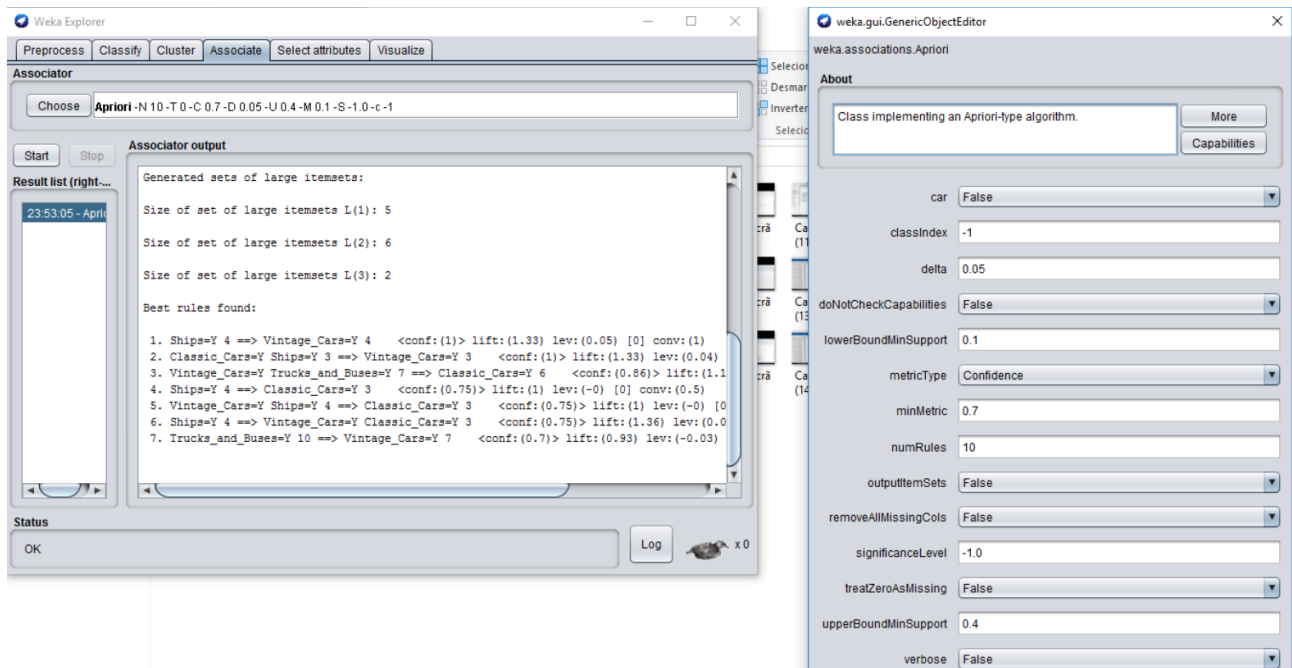
**weka.associations.Apriori About:**

Class implementing an Apriori-type algorithm.

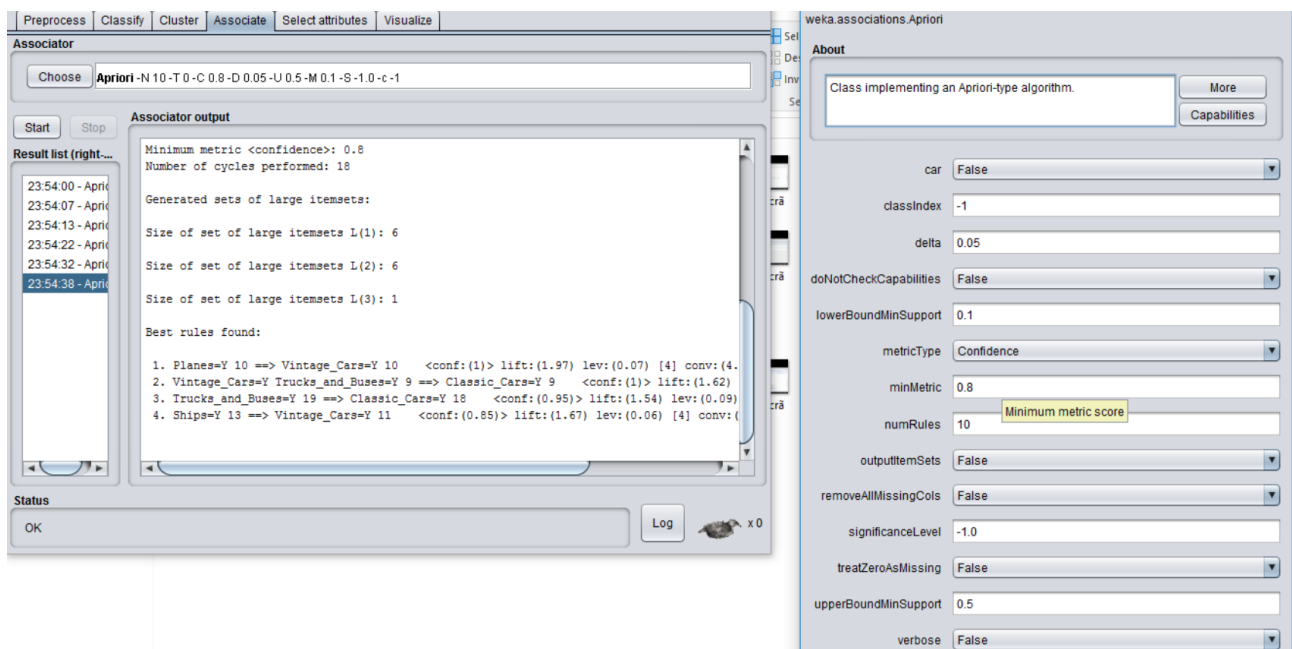
Parameters:

- car: False
- classIndex: -1
- delta: 0.05
- doNotCheckCapabilities: False
- lowerBoundMinSupport: 0.1
- metricType: Confidence
- minMetric: 0.6
- numRules: 10
- outputItemSets: False
- removeAllMissingCols: False
- significanceLevel: -1.0
- treatZeroAsMissing: False
- upperBoundMinSupport: 0.3
- verbose: False

## 10.2 TASKDATA2



## 10.3 TASKDATA3



Para a realização das regras de associação e a criação do ficheiro ARFF foi desenvolvido o código Java mostrado nos Anexos.

## **12. Problemas e Observações**

Desde logo tivemos problemas com a instalação do pentaho BI server. A resolução desse problema passou por mudar as variáveis de ambiente relativas ao JAVA, nomeadamente as do jdk e jre, dando o caminho de instalação de cada um. Numa fase futura os problemas continuaram e mais alguns e então decidimos instalar a versão mais recente do software.

Posteriormente, tivemos um erro estranho na transformação dos ficheiros MDB. Muito tempo foi perdido por causa de um erro que tinha a haver com a precisão. O erro foi resolvido ao adicionar os atributos um a um. Em relação ao cubo não foi possível a sua conclusão na primeira fase porque devido a estrutura das tabelas da nossa base de dados não era possível criar uma dimensão na localização (Country/City). Para resolver este problema, decidimos eliminar a tabela intermédia Orders. Esta solução causou repercussões nas transformações anteriores e por causa disso foi perdido muito tempo em alterar essas transformações.

Na segunda fase, tivemos um problema porque os jars que estavam no moodle para integrar não eram correctos de acordo com a nossa versão do pentaho porque a estrutura das pastas e os próprios jars, mudaram. Por isso foi necessário uma pesquisa exaustiva de forma a resolver o problema. Devido ao problema acima, não foi possível correr corretamente as transformações em JAVA.

De seguida, já com os ficheiros CSV com os dados relevantes para o Market Basket foi desenvolvida pelo grupo uma classe em JAVA, não seguindo exemplos sugeridos.

Um último problema que tivemos foi em relação aos valores de confiança e de suporte. Tivemos problemas em escolher o valor ideal tendo em conta que os dados estão sempre a variar e diferentes dados requerem diferentes valores de confiança e de suporte.

## 13. Anexos

### 13.1. Anexo 1 – Código Java Main

```
import java.io.FileReader;
import java.util.ArrayList;
import org.pentaho.di.core.KettleEnvironment;
import org.pentaho.di.trans.Trans;
import org.pentaho.di.trans.TransMeta;
import weka.associations.Apriori;
import weka.core.converters.ConverterUtils.DataSource;
import weka.core.Instances;
public class Main {
    public static void main (String[] args) throws Exception
    {
        //reader();
        //gerar os ficheiros arff
        ArrayList<String> listFiles = new ArrayList<>();
        for (int i = 1; i <= 3; i++)
        {
            String file = "C:\\Users\\topog\\Desktop\\pdi-ce-7.0.0.0-25\\TASKDATA" + i;
            transform(file + ".ktr");
            listFiles.add(file + ".arff");
            FileReader fr = new FileReader(file + ".csv");
            Data cc = new Data (fr,i);
            cc.readLines();
            cc.buildAttributes();
            cc.builData();
            cc.write();
        }
        for (String path : listFiles)
        {
            System.out.println(path + "\n");
            DataSource source = new DataSource(path);
            Instances data = source.getDataSet();
            Apriori model = new Apriori();
            model.setDelta(0.05);
            model.setLowerBoundMinSupport(0.1);
            model.setMinMetric(0.6);
            model.setUpperBoundMinSupport(0.4);
            model.buildAssociations(data);
            System.out.println(model);
        }
    }
    public static void transform (String pathName) throws Exception
    {
        KettleEnvironment.init();
        TransMeta transMeta = new TransMeta(pathName);
        Trans trans = new Trans(transMeta);

        trans.execute(null);
        trans.waitUntilFinished();
        if (trans.getErrors() > 0)
        {
            throw new RuntimeException("there were some errors during the transformation execution");
        }
    }
}
```

### 13.2. Anexo 2 – Código Java Data Class

```
import java.io.File;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.Writer;
import java.lang.reflect.Array;
import java.util.ArrayList;

import com.opencsv.CSVReader;

public class Data {
    private CSVReader file;
    private ArrayList<String> attributes;
    private ArrayList<String> dataValues;
    private ArrayList<String[]> lines;
    private int numberFile;
    public Data(FileReader csv, int numberFile)
    {
        this.numberFile = numberFile;
        this.file = new CSVReader(csv);
        attributes = new ArrayList<>();
        dataValues = new ArrayList<>();
        lines = new ArrayList<>();
    }
    //metodo para ler do ficheiro e formatar os valores
    public void readLines() throws IOException
    {
        String [] line = file.readNext();
        //le dos ficheiros
        while(line != null)
        {
            line = file.readNext();
            lines.add(line);
        }
        //formata os valores
        for (String [] aa : lines)
        {
            try
            {
                int a = lines.indexOf(aa);
                if (aa[0].contains(","))
                {
                    String [] bb;
                    bb = aa[0].split(", ");
                    lines.set(a, bb);
                }
            } catch (Exception e) {
                break;
            }
        }
    }

    public void buildAttributes()
    {
        //listagens de atributos existentes na amostra de dados
        for (String [] aa : lines)
```

```
        {
            try
            {
                for (String a : aa)
                {
                    if (!attributes.contains(a))
                    {
                        attributes.add(a);
                    }
                }
            } catch (Exception e) {
                // TODO: handle exception
            }
        }

        //novo codigo

        ArrayList<String> remove = new ArrayList<>();
        for (String values : attributes)
        {
            int i = attributes.indexOf(values);
            if (Character.isWhitespace(values.charAt(0)))
            {
                remove.add(attributes.get(i));
            }
        }
        for (String values : remove)
        {
            attributes.remove(values);
        }
    }

    public void buildData()
    {
        try
        {
            ArrayList<String> aux = new ArrayList<>();
            for (String a : attributes)
            {
                aux.add(a);
                aux.add(" ");
            }

            for (String [] aa : lines)
            {
                String res = "";
                int i = 1;
                for (String jessus : attributes)
                {
                    if(i++ == attributes.size() )
                    {
                        res += "?";
                    } else
                    {
                        res += "?,";
                    }
                }
                //res = "?,,?,?,?,?";
                StringBuilder gg = new StringBuilder(res);
            }
        }
    }
}
```

```
        for (String a : aa)
        {
            if (Character.isWhitespace(a.charAt(0)))
            {
                a = a.substring(1);
            }

            if (aux.contains(a))
            {
                int index = aux.indexOf(a);

                gg.setCharAt(index,'Y');
            }
        }
        res = gg.toString();
        dataValues.add(res);
    }
} catch (Exception e) {
    // TODO: handle exception
}

for (String values : attributes)
{
    int index = attributes.indexOf(values);
    values = values.replace(" ", "_");
    attributes.set(index, values);
}

}

public ArrayList<String> getDataValues()
{
    return dataValues;
}

public ArrayList<String> getAttributes()
{
    return attributes;
}

public void write ()
{
    String path = "C:\\Users\\topog\\Desktop\\pdi-ce-7.0.0.0-25\\TASKDATA" + numberFile + ".arff";
    File file = new File(path);
    try
    {
        Writer arff = new OutputStreamWriter(new FileOutputStream(file),"UTF8");
        arff.append("@Relation TASKDATA" + numberFile + "\n");
        for (String values : attributes)
        {
            arff.append("@Attribute ").append(values).append(" {'?',Y}\n");
        }
        arff.append("\n@data\n");
        for (String values : dataValues)
        {
            arff.append(values).append("\n");
        }

        arff.close();
    }
}
```

```
        } catch (Exception e) {  
            // TODO: handle exception  
        }  
    }  
}
```