

# Sales Analysis Interactive Dashboard

End-to-end process

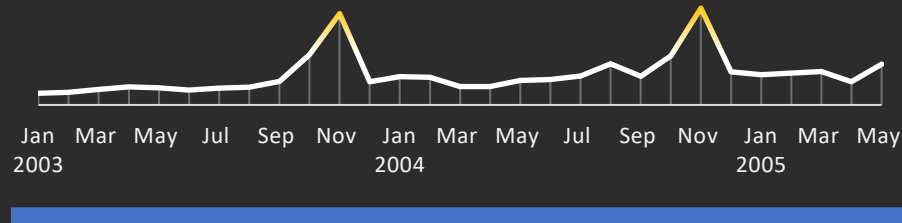
## About the data

2.5 years of sales data of transportation toys  
like trucks, planes, motorcycles, ships,  
vintage cars, ships, etc.

---

Source: [Kaggle.com](https://www.kaggle.com/datasets/robikscube/toyota)

# Objectives



1. Understand the quarterly revenue growth
2. Understand customer orders pattern thru segmentation
3. Find out which product price ranges sell more thru clustering
4. Distribution of sales by countries

# The Process

# Normalization

## Table Columns

Field Name
ORDERNUMBER
QUANTITYORDERED
PRICEEACH
ORDERLINENUMBER
SALES
ORDERDATE
STATUS
QTR_ID
MONTH_ID
YEAR_ID
PRODUCTLINE
MSRP
PRODUCTCODE
CUSTOMERNAME
PHONE
ADDRESSLINE1
ADDRESSLINE2
CITY
STATE
POSTALCODE
COUNTRY
TERRITORY
CONTACTLASTNAME
CONTACTFIRSTNAME
DEALSIZE



## Type Identification and Columns Clean-Up

Field Name	Type	Null
ORDERNUMBER	int	NOT NULL
QUANTITYORDERED	int	NOT NULL
PRICEEACH	money	NOT NULL
ORDERLINENUMBER	smallint	NOT NULL
SALES	money	NOT NULL
ORDERDATE	date	NOT NULL
STATUS	varchar(10)	NOT NULL
QTR_ID	int	
MONTH_ID	int	
YEAR_ID	int	
PRODUCTLINE	varchar(20)	NOT NULL
MSRP	money	NOT NULL
PRODUCTCODE	varchar(10)	NOT NULL
CUSTOMERNAME	varchar(50)	NOT NULL
PHONE	varchar(20)	NOT NULL
ADDRESSLINE1	varchar(50)	NOT NULL
ADDRESSLINE2	varchar(50)	
CITY	varchar(20)	NOT NULL
STATE	varchar(20)	
POSTALCODE	varchar(10)	
COUNTRY	varchar(20)	NOT NULL
TERRITORY	varchar(20))	
CONTACTLASTNAME	varchar(20))	NOT NULL
CONTACTFIRSTNAME	varchar(20))	NOT NULL
DEALSIZE	varchar(10)	NOT NULL

# First normal form (1NF)

Field Name	Type	Null
ORDERNUMBER	int	NOT NULL
QUANTITYORDERED	int	NOT NULL
PRICEEACH	money	NOT NULL
ORDERLINENUMBER	smallint	NOT NULL
SALES	money	NOT NULL
ORDERDATE	date	NOT NULL
STATUS	varchar(10)	NOT NULL
QTR_ID	int	
MONTH_ID	int	
YEAR_ID	int	
PRODUCTLINE	varchar(20)	NOT NULL
MSRP	money	NOT NULL
PRODUCTCODE	varchar(10)	NOT NULL
CUSTOMERNAME	varchar(50)	NOT NULL
PHONE	varchar(20)	NOT NULL
ADDRESSLINE1	varchar(50)	NOT NULL
ADDRESSLINE2	varchar(50)	
CITY	varchar(20)	NOT NULL
STATE	varchar(20)	
POSTALCODE	varchar(10)	
COUNTRY	varchar(20)	NOT NULL
TERRITORY	varchar(20))	
CONTACTLASTNAME	varchar(20))	NOT NULL
CONTACTFIRSTNAME	varchar(20))	NOT NULL
DEALSIZE	varchar(10)	NOT NULL

ProductLine		
Field Name	Attribute	Null
ID	smallint	
Line	varchar(20)	NOT NULL

OrderStatus		
Field Name	Attribute	Null
ID	smallint	
Status	varchar(10)	NOT NULL



Customers		
Field Name	Attribute	Null
ID	int	
BusinessName	varchar(50)	NOT NULL
Address1	varchar(50)	NOT NULL
Address2	varchar(50)	
PostalCode	varchar(10)	
ContactLName	varchar(20))	NOT NULL
ContactFName	varchar(20))	NOT NULL
Phone	varchar(20)	NOT NULL
ID	smallint	
City	varchar(20)	NOT NULL
State	varchar(20)	
Country	varchar(20)	NOT NULL
Territory	varchar(20))	

OrderLines		
Field Name	Attribute	Null
ID	int	
Orders_ID	int	NOT NULL
OrderLineNo	int	NOT NULL
Quantity	int	NOT NULL
Products_ID	varchar(10)	NOT NULL
Price	money	NOT NULL
SalesAmount	money	NOT NULL

Products		
Field Name	Attribute	Null
ID	varchar(10)	
ProductLine_ID	int	NOT NULL
MSRP	money	NOT NULL

???		
Field Name	Attribute	Null
QTR_ID	int	
MONTH_ID	int	
YEAR_ID	int	
DealSize_ID	varchar(10)	

Orders		
Field Name	Attribute	Null
ID	int	
Customers_ID	int	NOT NULL
TotalAmount	money	NOT NULL
OrderDate	date	NOT NULL
OrderStatus_ID	bit	NOT NULL

# Second normal form (2NF)

ProductLine		
Field Name	Attribute	Null
ID	smallint	
Line	varchar(20)	NOT NULL

OrderStatus		
Field Name	Attribute	Null
ID	smallint	
Status	varchar(10)	NOT NULL

Territory		
Field Name	Attribute	Null
ID	smallint	
Territory	varchar(10)	NOT NULL

Customers		
Field Name	Attribute	Null
ID	int	
BusinessName	varchar(50)	NOT NULL
Address1	varchar(50)	NOT NULL
Address2	varchar(50)	
PostalCode	varchar(10)	
ContactLName	varchar(20))	NOT NULL
ContactFName	varchar(20))	NOT NULL
Phone	varchar(20)	NOT NULL
ID	smallint	
City	varchar(20)	NOT NULL
State	varchar(20)	
Country	varchar(20)	NOT NULL
Territory	varchar(20))	

OrderLines		
Field Name	Attribute	Null
ID		
Orders_ID		NOT NULL
OrderLineNo		NOT NULL
Quantity		NOT NULL
ProductID		NOT NULL
Price		NOT NULL
Salesperson		NOT NULL

Product		
Field Name	Attribute	Null
ID	varchar(10)	
ProductLine_ID	int	NOT NULL
MSRP	money	NOT NULL

Country		
Field Name	Attribute	Null
ID	smallint	
Country	varchar(30)	NOT NULL
Territory_ID	smallint	NOT NULL

City		
Field Name	Attribute	Null
ID	smallint	
City	varchar(20)	NOT NULL
State	varchar(20)	
Country_ID	smallint	NOT NULL

???		
Field Name	Attribute	Null
QTR_ID	int	
MONTH_ID	int	
YEAR_ID	int	
DealSize_ID	varchar(10)	

Orders		
Field Name	Attribute	Null
ID	int	
Customers_ID	int	NOT NULL
TotalAmount	money	NOT NULL
OrderDate	date	NOT NULL
OrderStatus_ID	bit	NOT NULL

Customers		
Field Name	Attribute	Null
ID	int	
BusinessName	varchar(50)	NOT NULL
ContactLName	varchar(20)	NOT NULL
ContactFName	varchar(20)	NOT NULL
Phone	varchar(20)	NOT NULL
Address1	varchar(50)	NOT NULL
Address2	varchar(50)	
PostalCode	varchar(10)	
City_ID	smallint	NOT NULL

# Third normal form (3NF)

Customers		
Field Name	Attribute	Null
ID	int	
BusinessName	varchar(50)	NOT NULL
ContactLName	varchar(20)	NOT NULL
ContactFName	varchar(20)	NOT NULL
Phone	varchar(20)	NOT NULL
Address1	varchar(50)	NOT NULL
Address2	varchar(50)	
PostalCode	varchar(10)	
City_ID	smallint	NOT NULL

Territory		
Field Name	Attribute	Null
ID	smallint	
Territory	varchar(10)	NOT NULL

Country		
Field Name	Attribute	Null
ID	smallint	
Country	varchar(30)	NOT NULL
Territory_ID	smallint	NOT NULL

City		
Field Name	Attribute	Null
ID	smallint	
City	varchar(20)	NOT NULL
State	varchar(20)	
Country_ID	smallint	NOT NULL

OrderStatus		
Field Name	Attribute	Null
ID	smallint	
Status	varchar(10)	NOT NULL

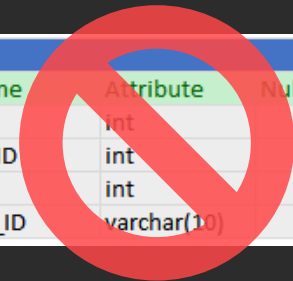
OrderLines		
Field Name	Attribute	Null
ID	int	
Orders_ID	int	NOT NULL
OrderLineNo	int	NOT NULL
Quantity	int	NOT NULL
Products_ID	varchar(10)	NOT NULL
Price	money	NOT NULL
SalesAmount	money	NOT NULL

Products		
Field Name	Attribute	Null
ID	varchar(10)	
ProductLine_ID	int	NOT NULL
MSRP	money	NOT NULL

Orders		
Field Name	Attribute	Null
ID	int	
Customers_ID	int	NOT NULL
TotalAmount	money	NOT NULL
OrderDate	date	NOT NULL
OrderStatus_ID	bit	NOT NULL

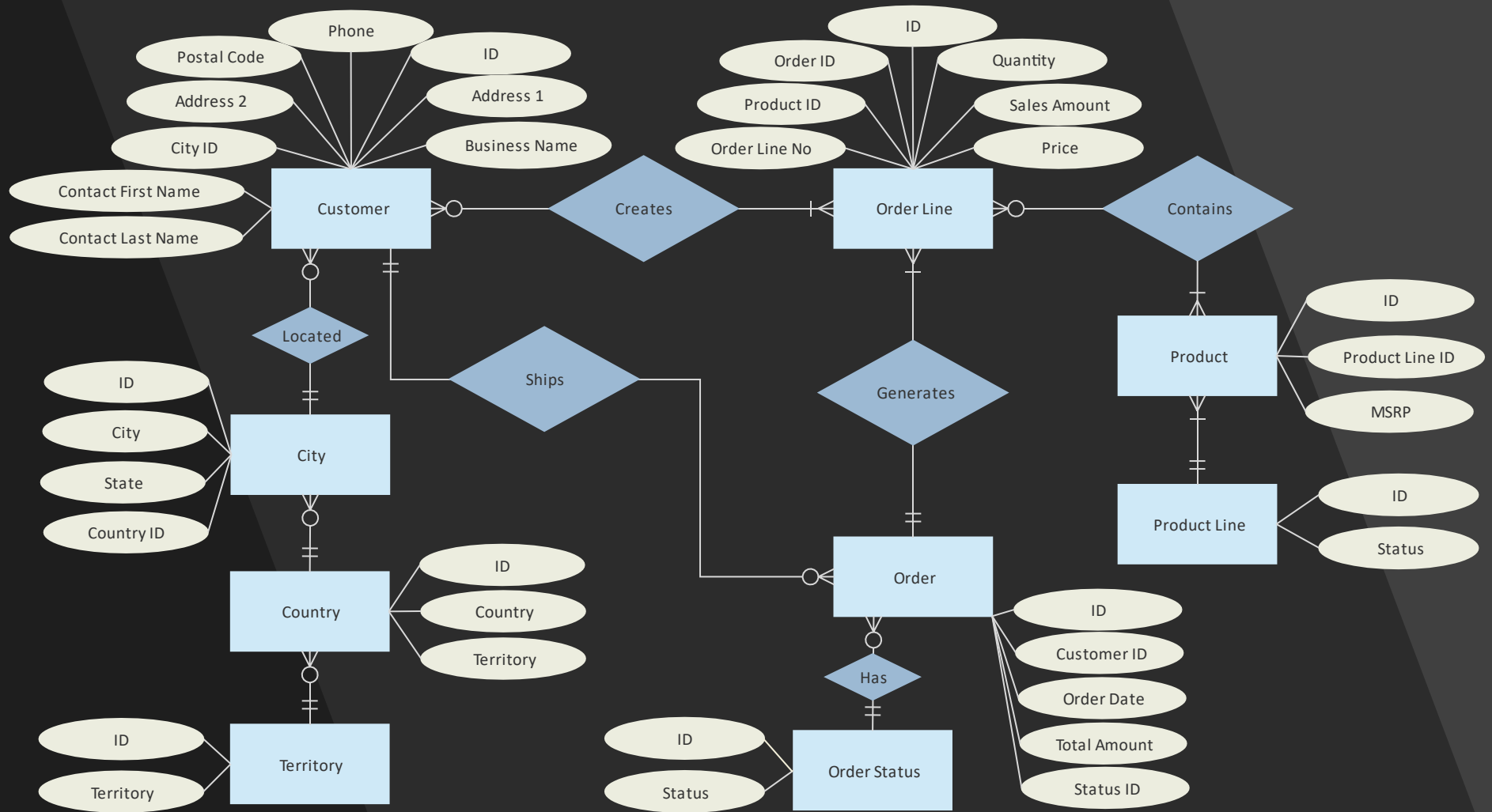
ProductLine		
Field Name	Attribute	Null
ID	smallint	
Line	varchar(20)	NOT NULL

???		
Field Name	Attribute	Null
QTR_ID	int	
MONTH_ID	int	
YEAR_ID	int	
DealSize_ID	varchar(10)	

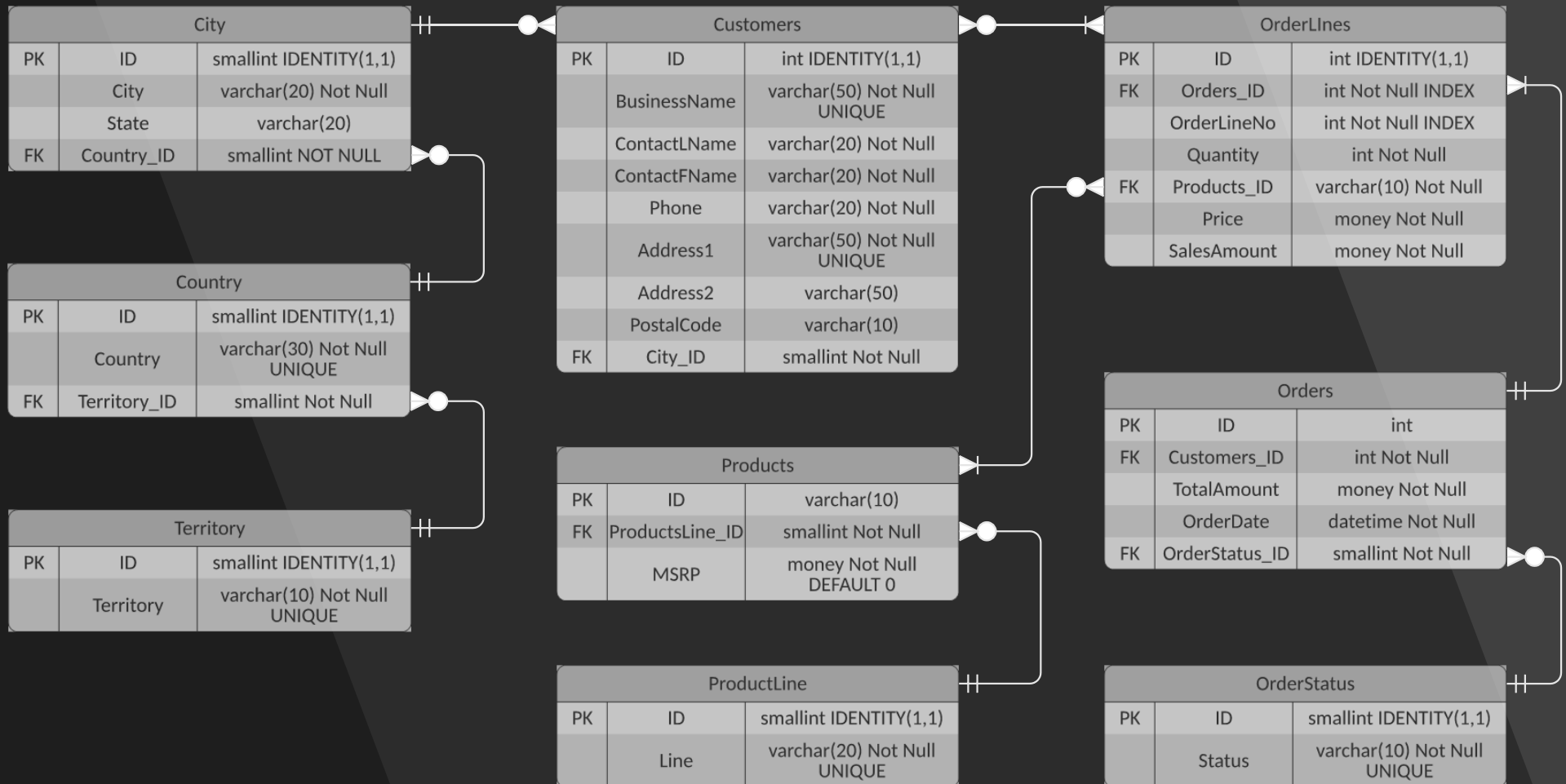




# Entity relationship diagram



# Relational schematic diagram



# Data cleanup using PowerQuery (1 of 3)

Some Price values does not match  
[QUANTITY / SALES] values

---

\$ Price	ABC 123 Price_
95.70	95.7
81.35	81.35
94.74	94.74
83.26	83.26
100.00	106.23
96.66	96.66
86.13	86.13
100.00	114.84
98.57	98.57
100.00	114.84
100.00	107.18
100.00	101.44
100.00	113.88
100.00	108.14
92.83	92.83



New Column using [QUANTITY / SALES]  
and changed Data Type to Currency

---

\$ Price
95.70
81.35
94.74
83.26
106.23
96.66
86.13
114.84
98.57
114.84
107.18
101.44
113.88
108.14
92.83

## Data cleanup using PowerQuery (2 of 3)

## Some Singapore records are tagged to Japan territory

A <sup>B</sup> <sub>C</sub> Country	A <sup>B</sup> <sub>C</sub> Territory
Singapore	Japan
Singapore	Japan
Singapore	APAC
Singapore	APAC
Singapore	Japan
Singapore	Japan
Singapore	Japan
Singapore	Japan
Singapore	Japan
Singapore	Japan
Singapore	Japan
Singapore	Japan
Singapore	Japan
Singapore	APAC
Singapore	Japan
Singapore	Japan



Replaced affected 'Japan' values with Singapore and Japan columns filtered

[illegible]

# Data cleanup using PowerQuery (3 of 3)

## Correcting date column to follow machine date format

The diagram illustrates the process of correcting a date column in PowerQuery. It shows a sequence of four tables connected by large orange arrows, representing the steps in the query editor.

**Table 1 (Initial State):**

OrderDate
2/24/2003 0:00
05/07/2003 00:00:00
07/01/2003 00:00:00
8/25/2003 0:00
10/10/2003 00:00:00
10/28/2003 0:00
11/11/2003 00:00:00
11/18/2003 0:00
12/01/2003 00:00:00
1/15/2004 0:00
2/20/2004 0:00
04/05/2004 00:00:00
5/18/2004 0:00
6/28/2004 0:00
7/23/2004 0:00

**Table 2 (Step 1):**

OrderDate.1	OrderDate.2
2/24/2003	00:00:00
05/07/2003	00:00:00
07/01/2003	00:00:00
8/25/2003	00:00:00
10/10/2003	00:00:00
10/28/2003	00:00:00
11/11/2003	00:00:00
11/18/2003	00:00:00
12/01/2003	00:00:00
1/15/2004	00:00:00
2/20/2004	00:00:00
04/05/2004	00:00:00
5/18/2004	00:00:00
6/28/2004	00:00:00
7/23/2004	00:00:00

**Table 3 (Step 2):**

OrderDate.1.2	OrderDate.1.1	OrderDate.1.3
24	2	2003
7	5	2003
1	7	2003
25	8	2003
10	10	2003
28	10	2003
11	11	2003
18	11	2003
1	12	2003
15	1	2004
20	2	2004
5	4	2004
18	5	2004
28	6	2004
23	7	2004

**Table 4 (Final State):**

OrderDate
24/02/2003
07/05/2003
01/07/2003
25/08/2003
10/10/2003
28/10/2003
11/11/2003
18/11/2003
01/12/2003
15/01/2004
20/02/2004
05/04/2004
18/05/2004
28/06/2004
23/07/2004

## Deletion of unnecessary columns and data type change

The diagram illustrates the process of deleting unnecessary columns and changing data types in a PowerQuery table. It shows a sequence of two tables connected by a large orange arrow, representing the steps in the query editor.

**Table 1 (Initial State):**

QTR_ID	MONTH_ID	YEAR_ID
1	2	2003
2	5	2003
3	7	2003
3	8	2003
4	10	2003
4	10	2003
4	11	2003
4	11	2003
4	12	2003
1	1	2004
1	2	2004
2	4	2004
2	5	2004
2	6	2004
3	7	2004

**Table 2 (Final State):**

QTR_ID	MONTH_ID	YEAR_ID
1	2	2003
2	5	2003
3	7	2003
3	8	2003
4	10	2003
4	10	2003
4	11	2003
4	11	2003
4	12	2003
1	1	2004
1	2	2004
2	4	2004
2	5	2004
2	6	2004
3	7	2004

**Query Settings Panel:**

**PROPERTIES**

Name: SalesData

**APPLIED STEPS**

- Source
- Changed Type
- Split Column by Delimiter
- Changed Type1
- Removed Columns
- Split Column by Delimiter1
- Changed Type2
- Reordered Columns
- Merged Columns
- Changed Type3
- Removed Columns1
- Changed Type4

# DDL queries



Capstone2.sql

## Database and table definitions

```
2  USE master
3
4  IF EXISTS (SELECT * FROM sys.databases WHERE name = N'Capstone2')
5  DROP DATABASE Capstone2
6  GO
7
8  CREATE DATABASE Capstone2
9  GO
10 USE Capstone2
11 GO
12
13 CREATE SCHEMA refTable
14 GO
15 CREATE SCHEMA Products
16 GO
17 CREATE SCHEMA Customers
18 GO
19 CREATE SCHEMA Orders
20 GO
```

```
69 CREATE TABLE Orders.OrderStatus
70 (
71     ID smallint IDENTITY(1,1)
72     , Status varchar(10) NOT NULL
73 )
```

```
75 CREATE TABLE Orders.Orders
76 (
77     ID int NOT NULL
78     , Customers_ID int NOT NULL
79     , TotalAmount money NOT NULL
80     , OrderDate datetime NOT NULL
81     , OrderStatus_ID smallint NOT NULL
82 )
```

```
43 CREATE TABLE Products.ProductLine
44 (
45     ID smallint IDENTITY(1,1)
46     , Line varchar(20) NOT NULL
47 )
```

```
21 CREATE TABLE refTable.Territory
22 (
23     ID smallint IDENTITY(1, 1)
24     , Territory varchar(10) NOT NULL UNIQUE
25 )
```

```
28 CREATE TABLE refTable.Country
29 (
30     ID smallint IDENTITY(1, 1)
31     , Country varchar(30) NOT NULL
32     , Territory_ID smallint
33 )
```

```
35 CREATE TABLE refTable.City
36 (
37     ID smallint IDENTITY(1,1)
38     , City varchar(20) NOT NULL
39     , State varchar(20)
40     , Country_ID smallint NOT NULL
41 )
```

```
84 CREATE TABLE Orders.OrderLines
85 (
86     ID int IDENTITY(1, 1)
87     , Orders_ID int NOT NULL
88     , OrderLineNo int NOT NULL
89     , Quantity int NOT NULL
90     , Products_ID varchar(10) NOT NULL
91     , Price money NOT NULL
92     , SalesAmount money NOT NULL
93 )
```

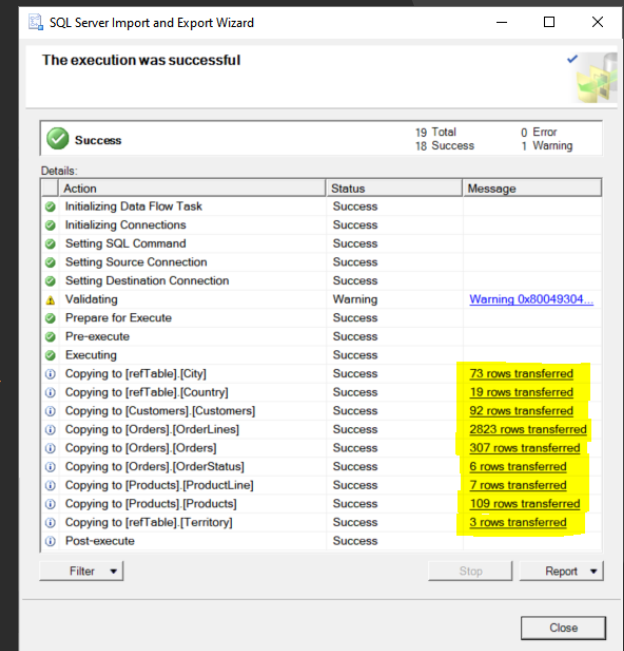
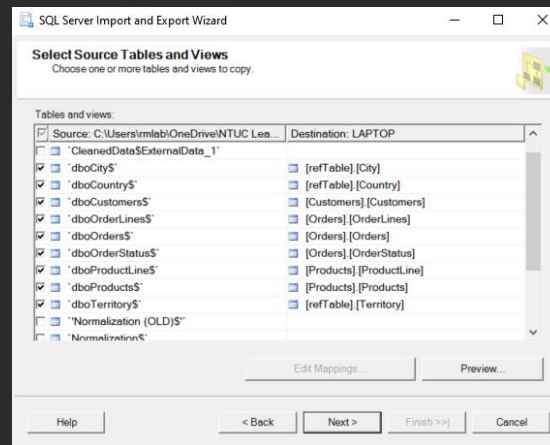
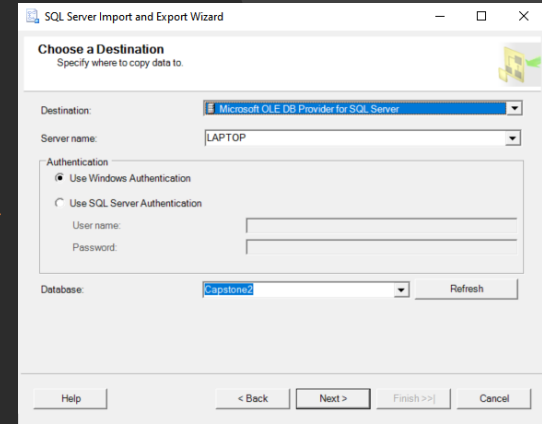
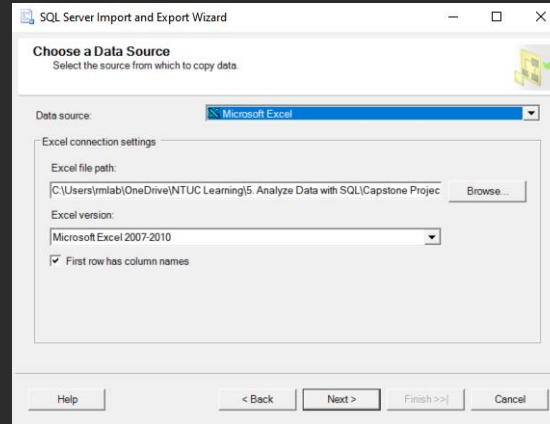
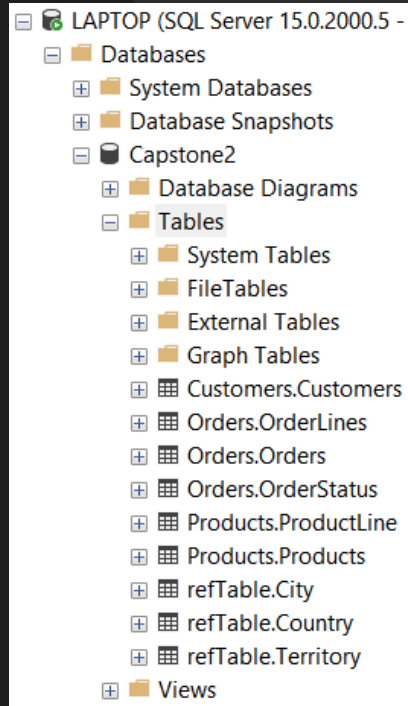
```
49 CREATE TABLE Products.Products
50 (
51     ID varchar(10) NOT NULL
52     , ProductLine_ID smallint NOT NULL
53     , MSRP money NOT NULL DEFAULT 0
54 )
```

```
56 CREATE TABLE Customers.Customers
57 (
58     ID int IDENTITY(1, 1)
59     , BusinessName varchar(50) NOT NULL
60     , ContactLName varchar(20) NOT NULL
61     , ContactFName varchar(20) NOT NULL
62     , Phone varchar(20) NOT NULL
63     , Address1 varchar(50) NOT NULL
64     , Address2 varchar(50)
65     , PostalCode varchar(10)
66     , City_ID smallint NOT NULL
67 )
```

## Constraints and indexes definitions

```
97 ALTER TABLE refTable.Territory
98     ADD CONSTRAINT [Territory_ID_pk] PRIMARY KEY ([ID])
99     , CONSTRAINT [Territory_Territory_uq] UNIQUE ([Territory])
100
101 ALTER TABLE refTable.Country
102     ADD CONSTRAINT [Country_ID_pk] PRIMARY KEY ([ID])
103     , CONSTRAINT [Country_Territory_ID_fk] FOREIGN KEY ([Territory_ID]) REFERENCES refTable.Territory ([ID])
104     , CONSTRAINT [Country_Country_uq] UNIQUE ([Country])
105
106 ALTER TABLE refTable.City
107     ADD CONSTRAINT [City_ID_pk] PRIMARY KEY ([ID])
108     , CONSTRAINT [City_Country_ID_fk] FOREIGN KEY ([Country_ID]) REFERENCES refTable.Country ([ID])
109
110 ALTER TABLE Products.ProductLine
111     ADD CONSTRAINT [ProductLine_ID_pk] PRIMARY KEY ([ID])
112     , CONSTRAINT [ProductLine_Line_uq] UNIQUE ([Line])
113
114 ALTER TABLE Products.Products
115     ADD CONSTRAINT [Products_ID_pk] PRIMARY KEY ([ID])
116     , CONSTRAINT [Products_ProductLine_ID_fk] FOREIGN KEY ([ProductLine_ID]) REFERENCES Products.ProductLine ([ID])
117
118 ALTER TABLE Customers.Customers
119     ADD CONSTRAINT [Customers_ID_pk] PRIMARY KEY ([ID])
120     , CONSTRAINT [Customers_City_ID_fk] FOREIGN KEY ([City_ID]) REFERENCES refTable.City ([ID])
121     , CONSTRAINT [Customers_BusinessName_uq] UNIQUE ([BusinessName], [Address1])
122
123 ALTER TABLE Orders.OrderStatus
124     ADD CONSTRAINT [OrderStatus_ID_pk] PRIMARY KEY ([ID])
125     , CONSTRAINT [OrderStatus_Status_uq] UNIQUE ([Status])
126
127 ALTER TABLE Orders.Orders
128     ADD CONSTRAINT [Orders_ID_pk] PRIMARY KEY ([ID])
129     , CONSTRAINT [Orders_Customers_ID_fk] FOREIGN KEY ([Customers_ID]) REFERENCES Customers.Customers ([ID])
130     , CONSTRAINT [Orders_OrderStatus_ID_fk] FOREIGN KEY ([OrderStatus_ID]) REFERENCES Orders.OrderStatus ([ID])
131
132 ALTER TABLE Orders.OrderLines
133     ADD CONSTRAINT [OrderLines_ID_pk] PRIMARY KEY ([ID])
134     , CONSTRAINT [OrderLines_Orders_ID_fk] FOREIGN KEY ([Orders_ID]) REFERENCES Orders.Orders ([ID])
135     , CONSTRAINT [OrderLines_Products_ID_fk] FOREIGN KEY ([Products_ID]) REFERENCES Products.Products ([ID])
136
137 CREATE INDEX [OrderLines_Orders_ID_OrderLineNo_idx] ON Orders.OrderLines ([Orders_ID], [OrderLineNo])
```

# Data import using Tasks > Import Data



# DML Queries – Create View



vCustomerSales.sql

View 'Sales.vCustomerSales' will be the source of the Excel dashboard

```
2 IF Object_ID('Sales.vCustomerSales') IS NOT NULL DROP VIEW Sales.vCustomerSales
3 IF EXISTS (select * from sys.schemas WHERE name='Sales') DROP SCHEMA Sales
4 GO
5 CREATE SCHEMA Sales
6 GO
7
8 CREATE VIEW Sales.vCustomerSales AS
9 SELECT o.ID AS OrderID
10      , ol.Products_ID
11      , pl.Line
12      , ol.Quantity
13      , ol.Price
14      , ol.SalesAmount
15      , FORMAT(o.OrderDate, 'dd/MM/yyyy') AS OrderDate
16      , Orders.svfDealSize(ol.SalesAmount) as 'DealSize'
17      , os.Status AS OrderStatus
18      , c.BusinessName AS CustomerName
19      , ct.City
20      , co.Country
21 FROM Orders.Orders o
22 JOIN Customers.Customers c
23     ON o.Customers_ID = c.ID
24 JOIN refTable.City ct
25     ON c.City_ID = ct.ID
26 JOIN refTable.Country co
27     ON ct.Country_ID = co.ID
28 JOIN Orders.OrderStatus os
29     ON o.OrderStatus_ID = os.ID
30 JOIN Orders.OrderLines ol
31     ON ol.Orders_ID = o.ID
32 JOIN Products.Products p
33     ON ol.Products_ID = p.ID
34 JOIN Products.ProductLine pl
35     ON p.ProductLine_ID = pl.ID
```



Views	
System Views	
Sales.vCustomerSales	
Columns	
OrderID (int, not null)	
Products_ID (varchar(10), not null)	
Line (varchar(20), not null)	
Quantity (int, not null)	
Price (money, not null)	
SalesAmount (money, not null)	
OrderDate (nvarchar(4000), null)	
DealSize (varchar(10), null)	
OrderStatus (varchar(10), not null)	
CustomerName (varchar(50), not null)	
City (varchar(20), not null)	
Country (varchar(30), not null)	
Triggers	



# DML Queries – Create Function



svfDealSize.sql

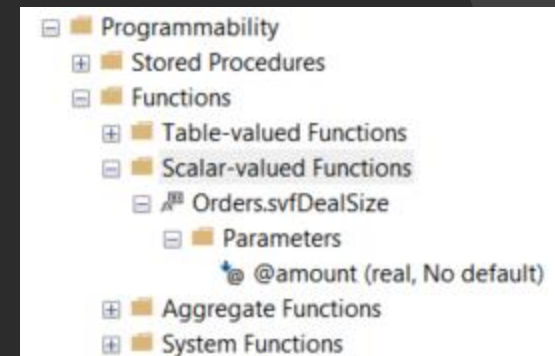
Function 'Orders.svfDealSize' to classify ranges of sales amount into:

Small : Less than \$3000

Medium : More than \$3000 and less than \$7000

Large : Above \$7000

```
3 IF Object_ID('Orders.svfDealSize') IS NOT NULL DROP FUNCTION dbo.DealSize
4 GO
5
6 CREATE FUNCTION Orders.svfDealSize(@amount real)
7 RETURNS varchar(10)
8 AS
9 BEGIN
10     DECLARE @size varchar(10)
11     IF @amount > 7000
12         SET @size = 'Large'
13     ELSE IF @amount > 3000
14         SET @size = 'Medium'
15     ELSE
16         SET @size = 'Small'
17     RETURN @size
18 END;
```



# View 'Sales.vCustomerSales' data

View 'Sales.vCustomerSales' to be referenced by Excel dashboard

```
22 Select * from Sales.vCustomerSales
```

100 %

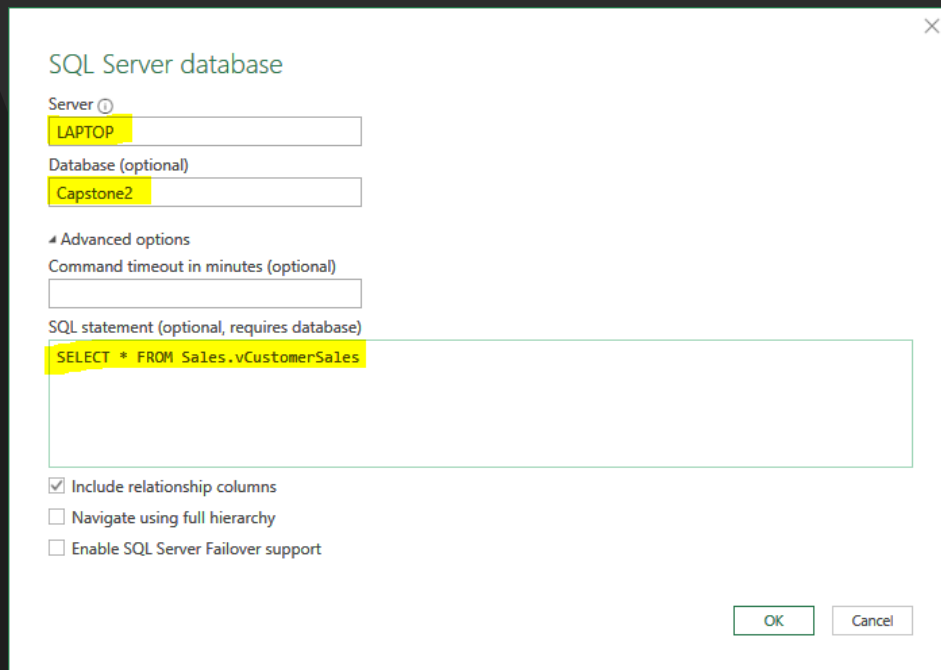
Results Messages

	OrderID	Products_ID	Line	Quantity	Price	SalesAmount	OrderDate	DealSize	OrderStatus	CustomerName	City	Country
1	10100	S24_3969	Vintage Cars	49	34.47	1689.03	06/01/2003	Small	Shipped	Online Diecast Creations Co.	Nashua	USA
2	10100	S18_2248	Vintage Cars	50	67.80	3390.00	06/01/2003	Medium	Shipped	Online Diecast Creations Co.	Nashua	USA
3	10100	S18_1749	Vintage Cars	30	171.70	5151.00	06/01/2003	Medium	Shipped	Online Diecast Creations Co.	Nashua	USA
4	10100	S18_4409	Vintage Cars	22	86.51	1903.22	06/01/2003	Small	Shipped	Online Diecast Creations Co.	Nashua	USA
5	10101	S18_2795	Vintage Cars	26	145.13	3773.38	09/01/2003	Medium	Shipped	Blauer See Auto, Co.	Frankfurt	Germany
6	10101	S24_2022	Vintage Cars	46	53.76	2472.96	09/01/2003	Small	Shipped	Blauer See Auto, Co.	Frankfurt	Germany
7	10101	S24_1937	Vintage Cars	45	31.20	1404.00	09/01/2003	Small	Shipped	Blauer See Auto, Co.	Frankfurt	Germany
8	10101	S18_2325	Vintage Cars	25	151.28	3782.00	09/01/2003	Medium	Shipped	Blauer See Auto, Co.	Frankfurt	Germany
9	10102	S18_1367	Vintage Cars	41	50.14	2055.74	10/01/2003	Small	Shipped	Vitachrome Inc.	NYC	USA
10	10102	S18_1342	Vintage Cars	39	123.29	4808.31	10/01/2003	Medium	Shipped	Vitachrome Inc.	NYC	USA
11	10103	S24_2300	Trucks and Buses	36	102.23	3680.28	29/01/2003	Medium	Shipped	Baane Mini Imports	Stavern	Norway
12	10103	S18_2432	Trucks and Buses	22	54.09	1189.98	29/01/2003	Small	Shipped	Baane Mini Imports	Stavern	Norway
13	10103	S32_1268	Trucks and Buses	31	104.01	3224.31	29/01/2003	Medium	Shipped	Baane Mini Imports	Stavern	Norway
14	10103	S10_4962	Classic Cars	42	128.53	5398.26	29/01/2003	Medium	Shipped	Baane Mini Imports	Stavern	Norway
15	10103	S18_4600	Trucks and Buses	36	117.45	4228.20	29/01/2003	Medium	Shipped	Baane Mini Imports	Stavern	Norway
16	10103	S700_2824	Classic Cars	42	106.21	4460.82	29/01/2003	Medium	Shipped	Baane Mini Imports	Stavern	Norway
17	10103	S32_3522	Trucks and Buses	45	75.63	3403.35	29/01/2003	Medium	Shipped	Baane Mini Imports	Stavern	Norway

# Data Import using Excel PowerQuery

# PowerQuery – Importing data from SQL server

## Referencing 'Sales.vCustomerSales' data



SQL Server database

Server ⓘ  
LAPTOP

Database (optional)  
Capstone2

⚙ Advanced options

Command timeout in minutes (optional)

SQL statement (optional, requires database)  
SELECT \* FROM Sales.vCustomerSales

☒ Include relationship columns  
☐ Navigate using full hierarchy  
☐ Enable SQL Server Failover support


OK Cancel

# PowerQuery – Data preview

## Loading imported data thru PowerQuery:

LAPTOP: Capstone2

OrderID	Products_ID	Line	Quantity	Price	SalesAmount	OrderDate	DealSize	OrderStatus	CustomerName
10100	S24_3969	Vintage Cars	49	34.47	1689.03	06/01/2003	Small	Shipped	Online Diecast Creations C
10100	S18_2248	Vintage Cars	50	67.8	3390	06/01/2003	Medium	Shipped	Online Diecast Creations C
10100	S18_1749	Vintage Cars	30	171.7	5151	06/01/2003	Medium	Shipped	Online Diecast Creations C
10100	S18_4409	Vintage Cars	22	86.51	1903.22	06/01/2003	Small	Shipped	Online Diecast Creations C
10101	S18_2795	Vintage Cars	26	145.13	3773.38	09/01/2003	Medium	Shipped	Blauer See Auto, Co.
10101	S24_2022	Vintage Cars	46	53.76	2472.96	09/01/2003	Small	Shipped	Blauer See Auto, Co.
10101	S24_1937	Vintage Cars	45	31.2	1404	09/01/2003	Small	Shipped	Blauer See Auto, Co.
10101	S18_2325	Vintage Cars	25	151.28	3782	09/01/2003	Medium	Shipped	Blauer See Auto, Co.
10102	S18_1367	Vintage Cars	41	50.14	2055.74	10/01/2003	Small	Shipped	Vitachrome Inc.
10102	S18_1342	Vintage Cars	39	123.29	4808.31	10/01/2003	Medium	Shipped	Vitachrome Inc.
10103	S24_2300	Trucks and Buses	36	102.23	3680.28	29/01/2003	Medium	Shipped	Baane Mini Imports
10103	S18_2432	Trucks and Buses	22	54.09	1189.98	29/01/2003	Small	Shipped	Baane Mini Imports
10103	S32_1268	Trucks and Buses	31	104.01	3224.31	29/01/2003	Medium	Shipped	Baane Mini Imports
10103	S10_4962	Classic Cars	42	128.53	5398.26	29/01/2003	Medium	Shipped	Baane Mini Imports
10103	S18_4600	Trucks and Buses	36	117.45	4228.2	29/01/2003	Medium	Shipped	Baane Mini Imports
10103	S700_2824	Classic Cars	42	106.21	4460.82	29/01/2003	Medium	Shipped	Baane Mini Imports
10103	S32_3522	Trucks and Buses	45	75.63	3403.35	29/01/2003	Medium	Shipped	Baane Mini Imports
10103	S12_1666	Trucks and Buses	27	125.74	3394.98	29/01/2003	Medium	Shipped	Baane Mini Imports
10103	S18_4668	Vintage Cars	41	47.29	1938.89	29/01/2003	Small	Shipped	Baane Mini Imports
10103	S18_1097	Trucks and Buses	35	112	3920	29/01/2003	Medium	Shipped	Baane Mini Imports

 The data in the preview has been truncated due to size limits.



Load

Transform Data

Cancel

Continued in Excel Dashboard...

So, were the objectives met?

## In summary

### Understand the quarterly revenue growth

Outliers in Q4 growth are seasonal and are driven by thanksgiving season sale, surge in 2005  
Q2 is something to be monitored if sustainable outside of the usual Q4 outliers

### Understand customer orders pattern thru segmentation

Surprisingly, most companies' orders are in the amount of \$37k-\$47k range. Study the recency of the high-value customers; proactively engage high-value customers but order less frequent

### Find out which product price ranges sell more thru clustering

Overall, products with prices ranging \$57-\$87 sell the most followed by even higher price range of \$87-\$117. Something to investigate why the cheapest range of \$27-\$57 sells poorly than products in the next price ranges

### Distribution of sales by countries

Biggest source of revenue comes from USA while more than 50 % of revenue is already covered by 5 countries (USA, Spain, France, Australia, UK). Look into advertising in smaller revenue countries or do further research in their toy markets



Q&A