

The background of the slide features a light blue gradient with several realistic water droplets of varying sizes scattered across the surface. A soft, circular light flare is centered in the upper half of the image, creating a subtle glow effect.

ONLINE SHOPPING DATASETS ANALYTIC

BY LINDAWATI HENDRAWAN

SOURCE DATA, DATA TRANSFER AND DATA CLEANING

- [HTTPS://WWW.KAGGLE.COM/TANYADAYANAND/ONLINE-SHOPPING](https://www.kaggle.com/tanyadayanand/online-shopping)

DATA TRANSFER AND DATA CLEANING

```
--ALTER COLUMN SHIPDATE
```

```
UPDATE SHIPMENT
```

```
SET SHIP_DATE = CONVERT(NVARCHAR(255),CONVERT(SMALLDATETIME, SHIP_DATE,105))
```

```
ALTER TABLE SHIPMENT
```

```
ALTER COLUMN SHIP_DATE SMALLDATETIME
```

```
--ALTER COLUMN PRODUCT_BASE_MARGIN
```

```
UPDATE MARKET
```

```
SET PRODUCT_BASE_MARGIN = '0' WHERE PRODUCT_BASE_MARGIN = 'NA'
```

```
ALTER TABLE MARKET
```

```
ALTER COLUMN PRODUCT_BASE_MARGIN FLOAT
```

IMPORTING DATASETS TO MS SQL

SQL Query1.sql - DESKTOP-DBQB47\MSSQLSERVER01 OnlineShopping (DESKTOP-DBQB47\linda (56)) - Microsoft SQL Server Management Studio

Object Explorer

- DESKTOP-DBQB47\MSSQL
- Databases
- System Databases
- Database Snapshots
- AdventureWorks2019
- AdventureWorksLT201
- case_study_1
- CaseStudyExercise
- ContosoUniDB
- Database Diagrams
- Tables
- Views
- External Resources
- Synonyms
- Programmability
- Service Broker
- Storage
- Security
- exercises
- tfidf
- Lab2Database
- New Database...
- New Query
- Script Database as
- Tasks
- Policies
- Facets
- Start PowerShell
- Azure Data Studio
- Reports
- Rename
- Delete
- Refresh
- Properties

Import Flat File 'OnlineShopping'

Specify Input File

Introduction

Specify Input File

Preview Data

Modify Columns

Summary

Results

Specify Input File

This operation will create a table from your input file.

Location of file to be imported

C:\Users\linda\Desktop\Data analyst\Capstone Project 2 associate data analyst\c

Browse

New table name:

customer

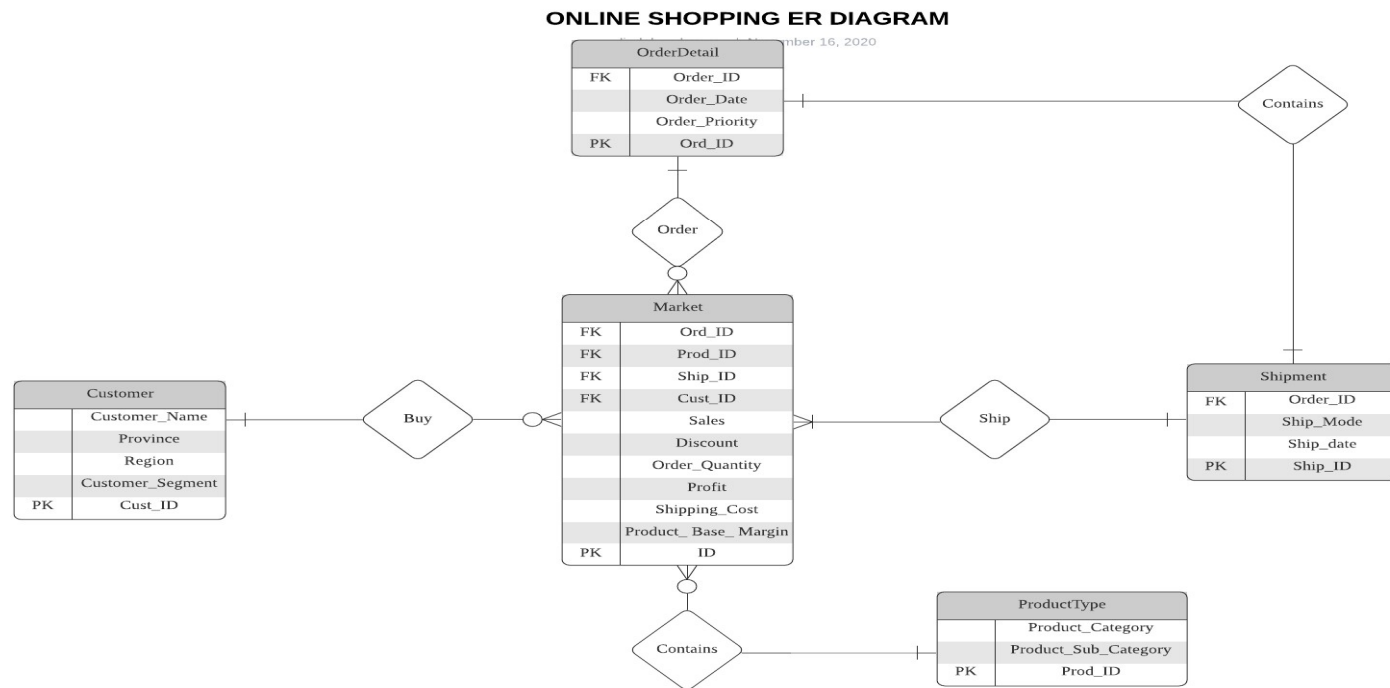
Table schema:

dbo

OnlineShopping

- Database Diagrams
- Tables
- System Tables
- FileTables
- External Tables
- Graph Tables
- dbo.customer
- dbo.market
- dbo.orderdetail
- dbo.producttype
- dbo.shipment

ER DIAGRAM ONLINE SHOPPING



DATA TRANSFER FROM MS SQL TO EXCELL

SQL Server database

Server ^①
desktop-dbqbj47\mssqlserver01

Database (optional)
onlineshopping

Advanced options

Command timeout in minutes (optional)
0

SQL statement (optional, requires database)
SELECT datepart(yy, o.order_date) as 'Year', c.Customer_Segment, count(distinct c.Cust_id) FROM customer c JOIN market m ON c.cust_ID = m.Cust_id JOIN orderdetail o ON o.Ord_id = m.Ord_id WHERE datepart(yy, o.order_date) = 2012

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

OK Cancel

desktop-dbqbj47\mssqlserver01: onlineshopping

Year	Customer_Segment	No.ofCust
2012	CORPORATE	358
2012	HOME OFFICE	222
2012	CONSUMER	190
2012	SMALL BUSINESS	195

Year	Customer_Segment	No.ofCust
2012	CORPORATE	358
2012	HOME OFFICE	222
2012	CONSUMER	190
2012	SMALL BUSINESS	195

QUERIES

```
--CUSTOMERSEGMENT
SELECT DATEPART(YY, O.ORDER_DATE) AS 'YEAR',    C.CUSTOMER_SEGMENT, COUNT(DISTINCT C.CUST_ID) AS 'NO.OFCUST',
SUM(M.SALES) AS 'T.SALES',
SUM(M.PROFIT) AS 'T.PROFIT',
ROUND(SUM(M.PROFIT)/SUM(M.SALES),2) AS 'GROSS_MARGIN',
COUNT(M.ORD_ID) AS 'NO.OFORDER'
FROM CUSTOMER C
JOIN MARKET M ON C.CUST_ID = M.CUST_ID
JOIN ORDERDETAIL O ON O.ORD_ID = M.ORD_ID
GROUP BY DATEPART(YY, ORDER_DATE), C.CUSTOMER_SEGMENT
ORDER BY DATEPART(YY, ORDER_DATE), [T.SALES] DESC
```

```
-- Topprofit by product
With Toprow AS (
SELECT ROW_NUMBER() over (partition by datepart(yy, order_date) order by round(sum(m.profit),2) desc) AS R,
datepart(yy, order_date) as 'Year', p.Product_Sub_Category,
c.region,
sum(m.Order_Quantity) AS 'Quantity', round(sum(m.sales),2) AS 'Total Sales',
round(sum(m.profit),2) AS 'TotalProfit',
round(sum(m.Profit)/sum(m.Sales),2) AS 'Gross_Margin',
round(avg(m.discount),2) AS 'Discount',
round(avg(m.Product_Base_Margin),2) AS 'ProductBaseMargin',
round(avg(m.Shipping_Cost),2) AS 'Shippingcost'
FROM orderdetail o
JOIN market m ON o.ord_id = m.Ord_id
JOIN producttype p ON m.Prod_id = p.prod_id
JOIN customer c ON c.Cust_id = m.Cust_id
GROUP by datepart(yy, order_date), p.Product_sub_Category, c.region)
Select *
From Toprow
where Toprow.R <=20
```

```
-- PROFIT BY PRODUCT
SELECT DATEPART(YEAR, ORDER_DATE) AS 'YEAR', P.PRODUCT_CATEGORY, SUM(M.ORDER_QUANTITY) AS 'QUANTITY', ROUND(SUM(M.SALES),2) AS 'TOTALSALES',
ROUND(SUM(M.PROFIT),2) AS 'TOTALPROFIT', ROUND(SUM(M.PROFIT)/SUM(M.SALES),2) AS 'GROSS MARGIN', ROUND(AVG(M.DISCOUNT),2) AS 'DISCOUNT'
FROM ORDERDETAIL O
JOIN MARKET M ON O.ORD_ID = M.ORD_ID
JOIN PRODUCTTYPE P ON M.PROD_ID = P.PROD_ID
JOIN CUSTOMER C ON C.CUST_ID = M.CUST_ID
GROUP BY DATEPART(YEAR,ORDER_DATE), P.PRODUCT_CATEGORY
ORDER BY DATEPART(YEAR,ORDER_DATE), [TOTALPROFIT] DESC, P.PRODUCT_CATEGORY
```

```
--avg shipment cost
```

```
Select Ship_Mode, p.Product_Category,
       round(avg(m.Shipping_Cost),2) AS 'Avg_Shipcost'
```

```
FROM orderdetail o
JOIN shipment s ON s.Order_ID = o.Order_ID
JOIN market m ON m.Ord_id = o.Ord_id
JOIN customer c ON c.Cust_id = m.Cust_id
join producttype p on p.Prod_id = m.Prod_id
GROUP by ship_mode, p.Product_Category
ORDER by ship_mode
```

```
-- Losses
```

```
With Toprow AS (
SELECT ROW_NUMBER() over (partition by datepart(yy, order_date) order by round(sum(m.profit),2)asc) AS R,
datepart(yy, order_date) as 'Year', p.Product_Sub_Category, sum(m.Order_Quantity) AS 'Quantity', round(sum(m.sales),2) AS 'Total Sales',
round(sum(m.profit),2) AS 'TotalProfit',
round(sum(m.Profit)/sum(m.Sales),2) AS 'Gross_Margin',
round(avg(m.discount),2) AS 'Discount',
round(avg(m.Product_Base_Margin),2) AS 'ProductBaseMargin',
round(avg(m.Shipping_Cost),2) AS 'Shippingcost',
count(distinct m.Cust_id) AS 'No.ofcust',
count(m.ord_id) AS 'No.ofOrder'
FROM orderdetail o
JOIN market m ON o.ord_id = m.Ord_id
JOIN producttype p ON m.Prod_id = p.prod_id
GROUP by datepart(yy, order_date), p.Product_sub_Category)
```

```
Select *
From Toprow
where Toprow.R <=10
```

```

--CUSTOMER LIFE CYCLE
SET DATEFORMAT DMY
WITH O_SEQUENCE AS (
SELECT O.ORDER_DATE, C.CUST_ID,
ROW_NUMBER() OVER (PARTITION BY C.CUST_ID ORDER BY
O.ORDER_DATE ASC) AS ORDER_SEQUENCE,
LAG(O.ORDER_DATE) OVER (PARTITION BY C.CUST_ID ORDER BY
O.ORDER_DATE ASC) AS PREVIOUS_ORDER_DATE
FROM ORDERDETAIL O
JOIN MARKET M ON M.ORD_ID = O.ORD_ID
JOIN CUSTOMER C ON C.CUST_ID = M.CUST_ID
GROUP BY O.ORDER_DATE, C.CUST_ID),

TIME_BETWEEN_ORDERS AS (
SELECT ORDER_DATE, CUST_ID, ORDER_SEQUENCE,
CASE WHEN PREVIOUS_ORDER_DATE IS NULL THEN ORDER_DATE
ELSE PREVIOUS_ORDER_DATE END AS PREVIOUS_ORDER_DATE,
DATEDIFF(DAY, PREVIOUS_ORDER_DATE, ORDER_DATE)
AS DAYS_BETWEEN_ORDERS
FROM O_SEQUENCE),

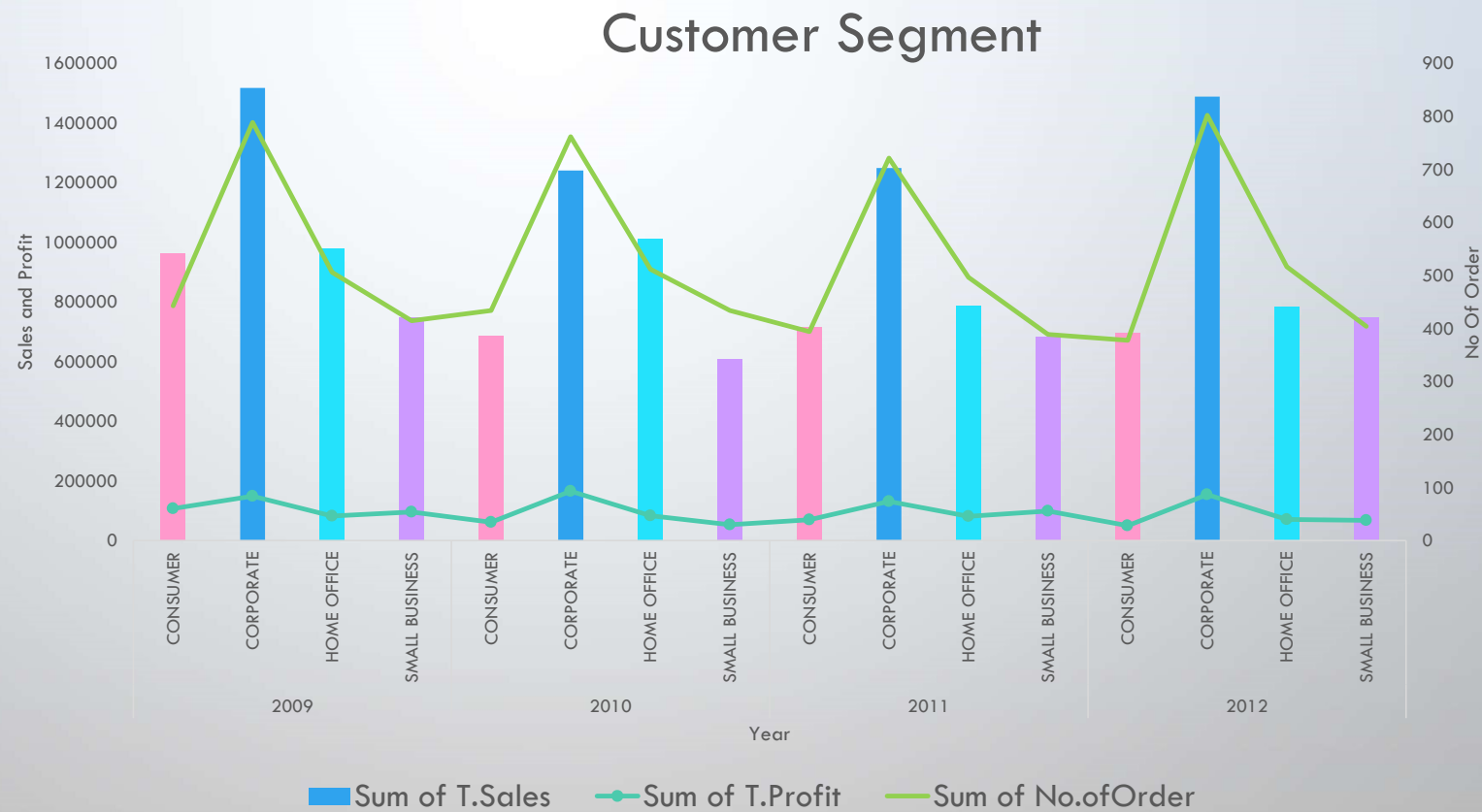
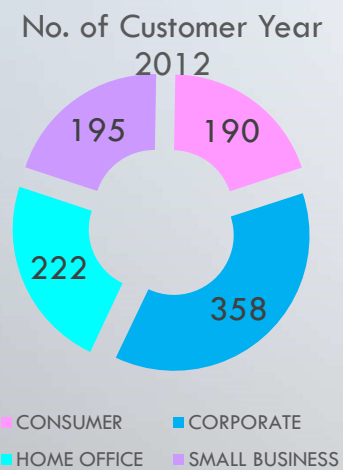
CUSTOMER_LIFE_CYCLE AS (
SELECT ORDER_DATE, CUST_ID,
CASE
WHEN ORDER_SEQUENCE = 1 THEN 'NEW CUSTOMER'
WHEN DAYS_BETWEEN_ORDERS > 0 AND DAYS_BETWEEN_ORDERS < 366
THEN 'ACTIVE CUSTOMER'
WHEN DAYS_BETWEEN_ORDERS > 365 THEN 'LAPSED CUSTOMER'
ELSE 'UNKNOWN'
END AS CUSTOMER_LIFE_CYCLE,

ORDER_SEQUENCE,
PREVIOUS_ORDER_DATE,

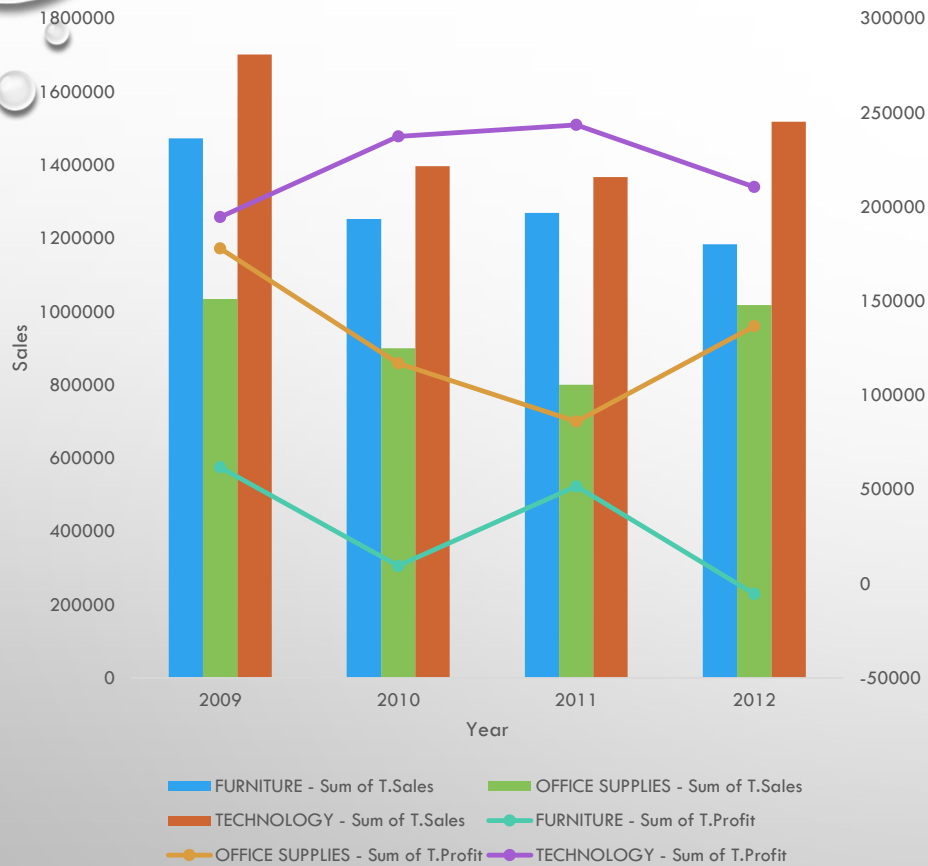
CASE
WHEN DAYS_BETWEEN_ORDERS IS NULL THEN 0
ELSE DAYS_BETWEEN_ORDERS
END AS DAYS_BETWEEN_ORDERS
FROM TIME_BETWEEN_ORDERS)

SELECT
O.ORDER_DATE,
C.CUST_ID,
T1.CUSTOMER_LIFE_CYCLE,
T1.ORDER_SEQUENCE,
T1.DAYS_BETWEEN_ORDERS,
O.ORDER_ID,
C.CUSTOMER_SEGMENT
FROM ORDERDETAIL O
JOIN MARKET M ON M.ORD_ID = O.ORD_ID
JOIN CUSTOMER C ON C.CUST_ID = M.CUST_ID
JOIN CUSTOMER_LIFE_CYCLE T1 ON
(C.CUST_ID=T1.CUST_ID
AND O.ORDER_DATE=T1.ORDER_DATE)

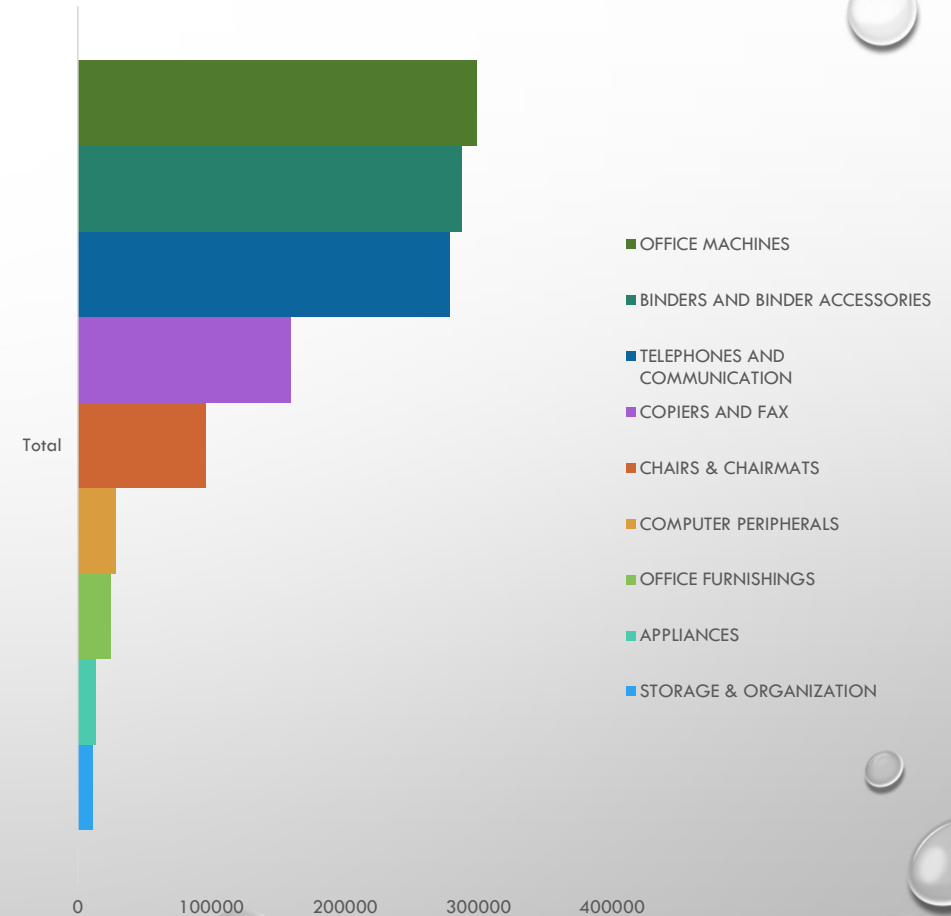
```

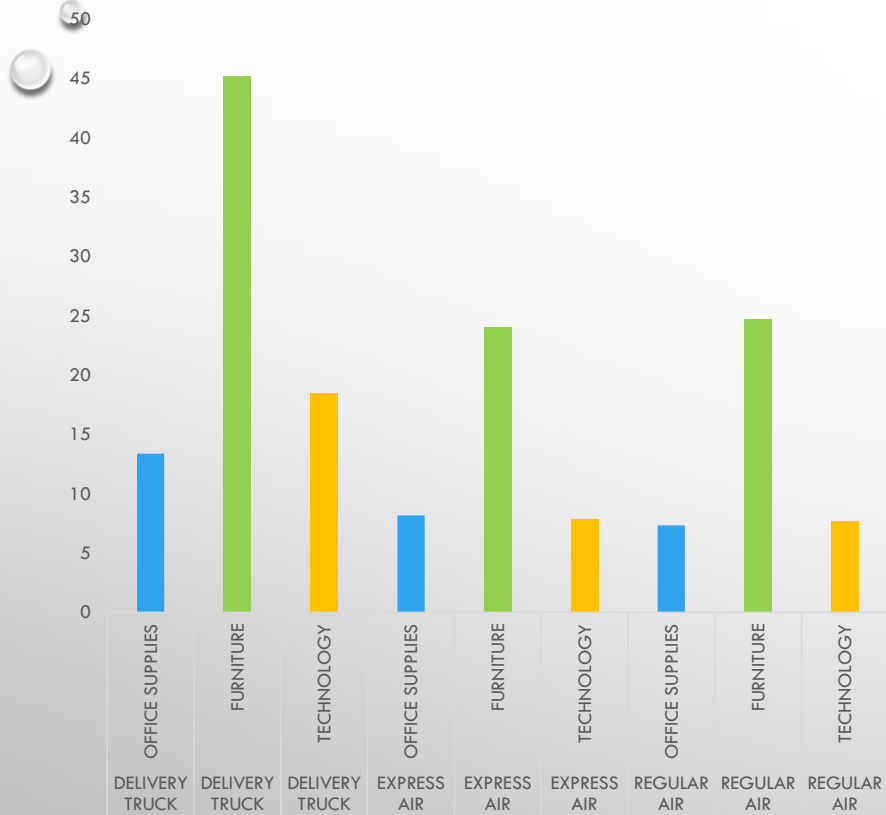
Sales vs Profit by Product



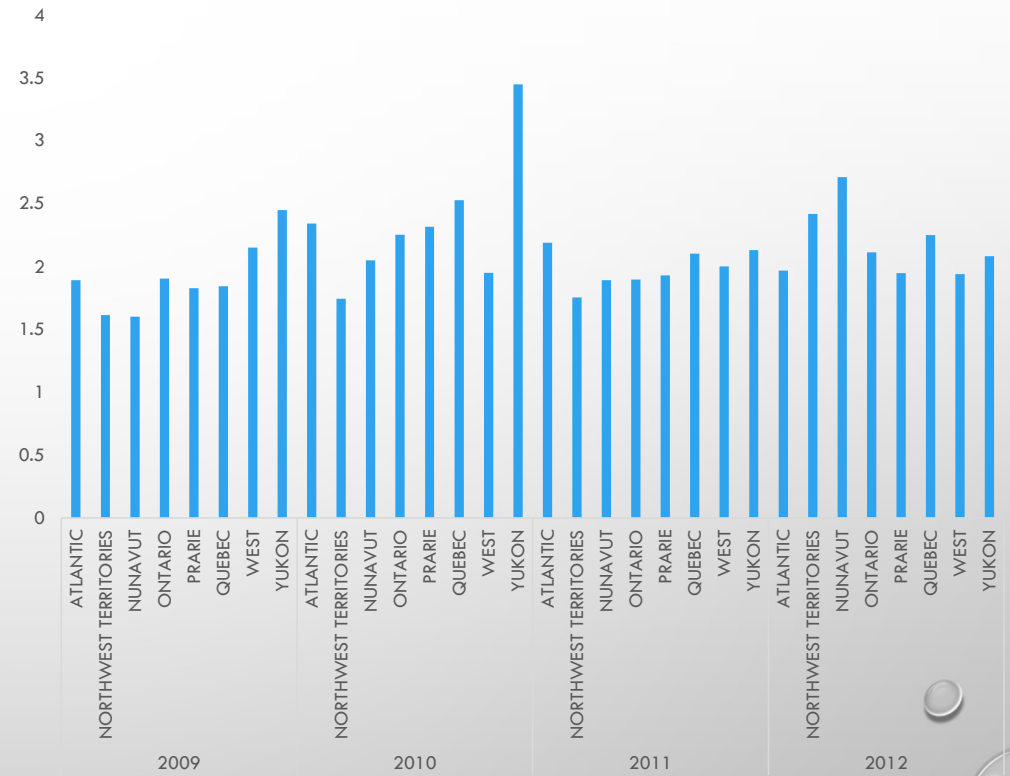
3 years Highest Profit From Product sub category



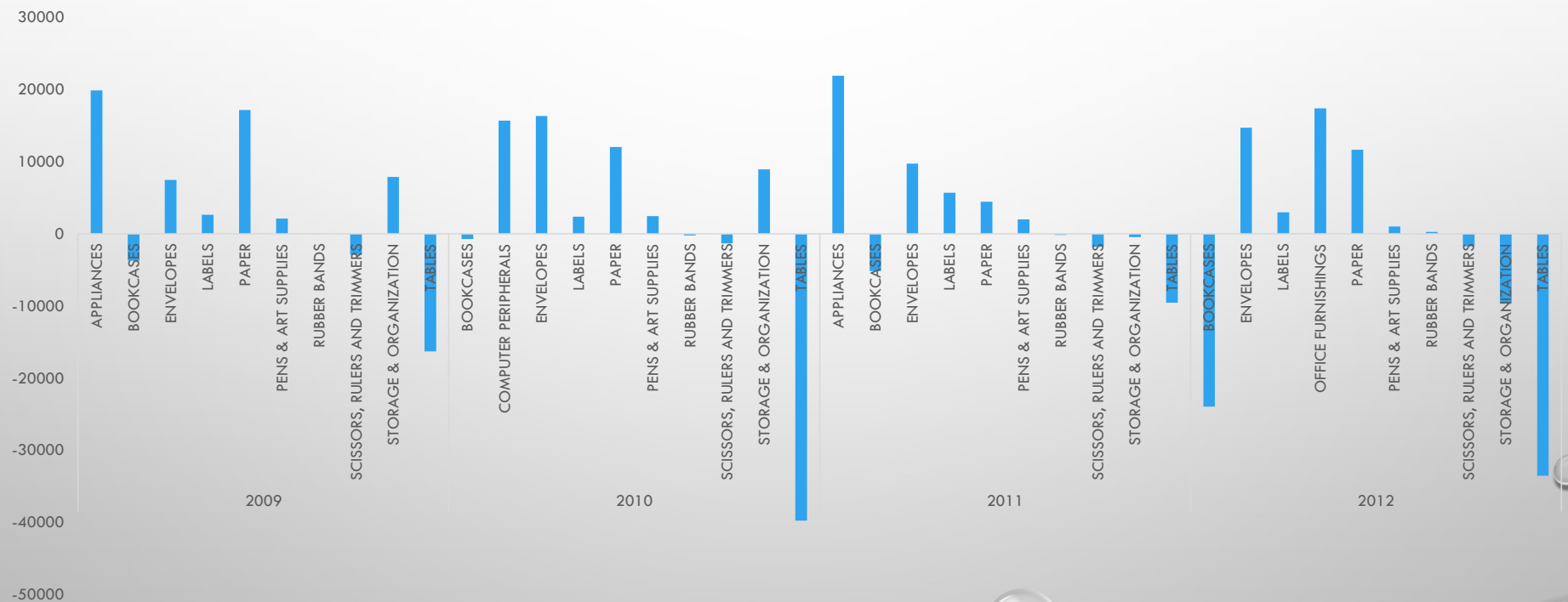
Avg Shipping Cost



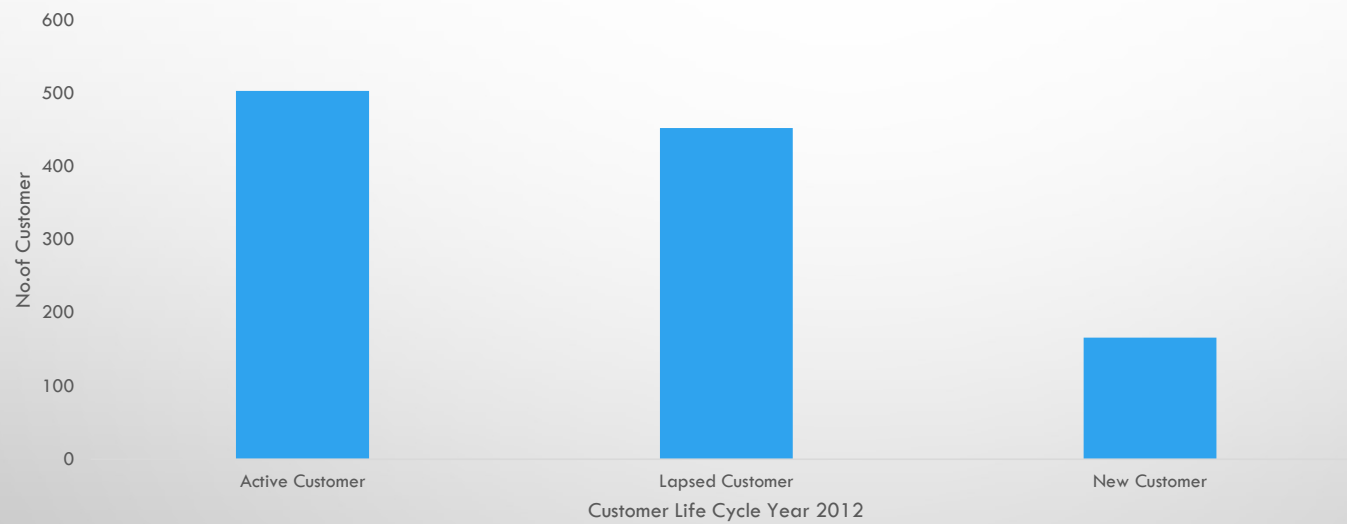
Avg Daystoship



LOSSES FROM PRODUCT SUB CATEGORY



CUSTOMER LIFE CYLCE



PROBLEMS

- LATE ORDER PROCESSING TO SHIP THE PRODUCTS TO CUSTOMER WITH SOME ORDER WERE SHIP TO CUSTOMER > 7 DAYS
- SOME PRODUCTS WERE SOLD BELOW PRODUCT MARGIN AND INCURRED LOSSES

RECOMMENDATION

- IMPROVE THE ORDER PROCESSING TO SHIP DAYS TO CUSTOMER
- IMPROVE THE SUPPLY CHAIN SYSTEMS
- OPT FOR DELIVERY MODE WITH MORE EFFICIENT SHIPPING COST
- IMPROVE CUSTOMER RETENTION BY MAINTAINING THE CUSTOMER LIFE CYCLE DATA



THANK YOU