

Analyze Data in a Model car relational Database with MySQL workbench

By Catherin Sharmila

Project Scenario:

This project aim is to conduct a data analysis in relational database of fictional Mint classic company. I stepped as an entry level data analyst to assist the company with a goal of supporting inventory-related business decisions for closure of one of their storage facility or reduction.

Skills Demonstrated:

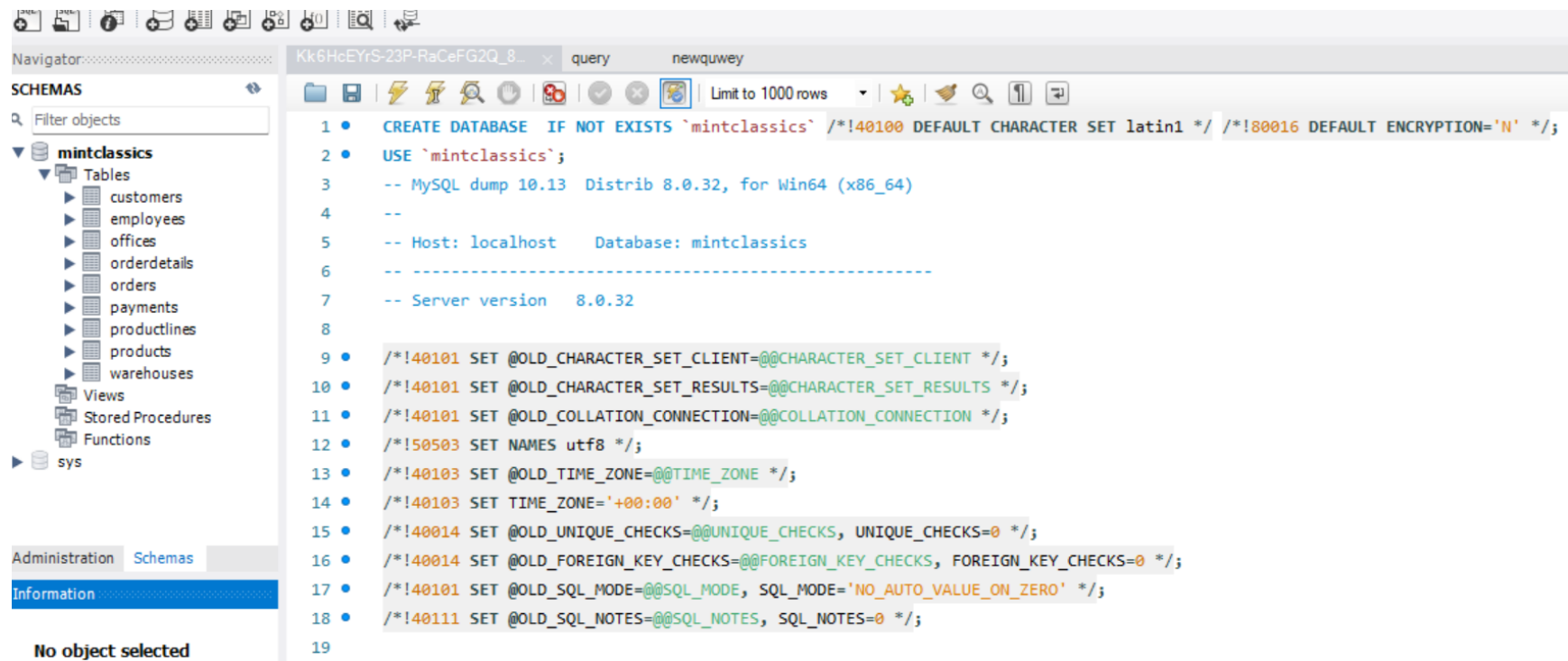
- Using SQL queries to extract and analyze data in mint classics relational database.
- Identifying and isolating the parts of data which is useful in deciding how to reducing or closing of one of the warehouses.

Tools used:

- MySQL workbench- this is utilized for create, run and analyze data using SQL queries.
- Github- This is used for uploading project and reports as part of the portfolio.

Step 1: Installing tool and Importing Model car Database

In this process I installed MySQL Workbench for creating and running sql queries, schemas. Then I created the new connections in the name of “local server”. After I imported “mint classics dataset”, runned the script successfully. Finally I ensured the database is imported correctly.



Step 2: Analyze Mint classics database tables:

I analyzed all tables how they interrelated, what fields are influenced the business problem. In this ER diagram played vital role for knowing inter relationship among tables.

Step 3: Creating SQL queries for business issues

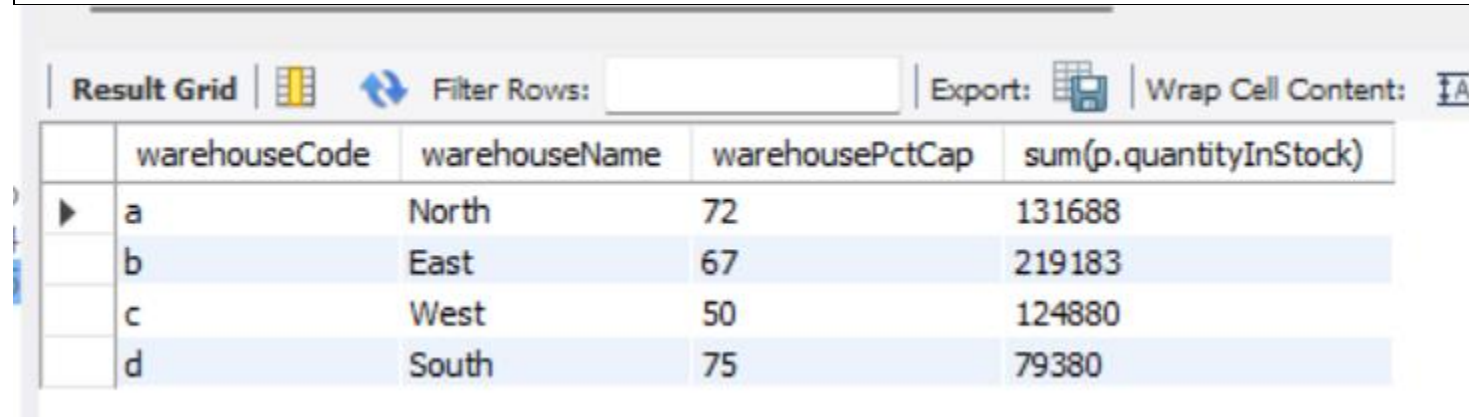
The main issue of the company is to plan to close the one of their storage facility or reducing products. Based on that I created and runned SQL queries to get the data for their problem.

Step3: Understanding Business issues and finding solution using SQL queries:

Q1: Where are items stored and if they were rearranged, could a warehouse be eliminated?

Here I chose warehousecode, warehousename, warehouse percentage capacity from warehouses table and quantityinstock from products table. From this we can know how much quantity of stocks are stored in each warehouses and their percentage.

```
select w.warehouseCode, w.warehouseName, w.warehousePctCap, sum(p.quantityInStock) from  
warehouses w, products p where w.warehouseCode=p.warehouseCode group by  
warehouseCode;
```







The screenshot shows a database query result grid with the following data:

	warehouseCode	warehouseName	warehousePctCap	sum(p.quantityInStock)
▶	a	North	72	131688
	b	East	67	219183
	c	West	50	124880
	d	South	75	79380



Then I calculated 100 percentage capacity for each warehouse. From that I concluded warehouse code “d” named “South” having less quantity of stock and less capacity to store when moving products from other warehouse.

```
select w.warehouseCode, w.warehouseName, w.warehousePctCap, sum(p.quantityInStock),  
((100*sum(p.quantityInStock))/w.warehousePctCap) as total_capacity from warehouses w,  
products p where w.warehouseCode=p.warehouseCode group by warehouseCode;
```

Result Grid   Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 					
	warehouseCode	warehouseName	warehousePctCap	sum(p.quantityInStock)	total_capacity_to_store
▶	a	North	72	131688	182900
	b	East	67	219183	327138.80597014923
	c	West	50	124880	249760
	d	South	75	79380	105840

Q2: what products category stored in each warehouse and their total quantity of stock?

```
select w.warehouseName, w.warehouseCode, sum(p.quantityInStock) as
total_quantity_in_stock , p.productLine from warehouses w left outer join products p on
w.warehouseCode=p.warehouseCode group by warehouseCode, p.productLine;
```

Result Grid				
		Filter Rows:		Export:  Wrap Cell Content: 
	warehouseName	warehouseCode	sum(p.quantityInStock)	quantity_in_stock
▶	North	a	69401	Motorcycles
	North	a	62287	Planes
	East	b	219183	Classic Cars
	West	c	124880	Vintage Cars
	South	d	35851	Trucks and Buses
	South	d	26833	Ships
	South	d	16696	Trains

From above Q1 and Q2 I concluded, we can move products from warehouse “d” to other warehouses. If we want to move first we should analyze the productCode, sum(quantityInStock), sum(quantityOrdered), percentage of quantityOrdered, avg, min and max ordered for the productLine “Trucks and Buses”, Ships”, “Trains” which are in warehouse “d”. so I created query based on that.

```
select p.productCode, p.productLine, p.quantityInStock, sum(o.quantityOrdered),
avg(o.quantityOrdered), max(o.quantityOrdered), min(o.quantityOrdered) from products p join
orderdetails o on p.productCode=o.productCode where p.productLine="Trains" or
p.productLine="Ships" or p.productLine="Trucks and Buses" group by productLine, p.productCode,
p.quantityInStock order by p.productLine;
```

Result Grid								
Filter Rows:			Export:		Wrap Cell Content:			
	productCode	productLine	quantityInStock	sum(o.quantityOrdered)	%_quantity_ordered	avg(o.quantityOrdered)	max(o.quantityOrdered)	min(o.quantityOrdered)
▶	S18_3029	Ships	4259	966	22.6814	34.5000	49	21
	S24_2011	Ships	1898	1011	53.2666	37.4444	50	21
	S700_1138	Ships	1897	934	49.2356	34.5926	50	21
	S700_1938	Ships	737	898	121.8453	33.2593	50	20
	S700_2047	Ships	3501	897	25.6213	32.0357	55	20
	S700_2610	Ships	7083	1020	14.4007	37.7778	50	21
	S700_3505	Ships	1956	952	48.6708	35.2593	50	21
	S700_3962	Ships	5088	896	17.6101	33.1852	50	20
	S72_3212	Ships	414	958	231.4010	35.4815	50	20
	S18_3259	Trains	6450	918	14.2326	34.0000	50	21
	S32_3207	Trains	8601	934	10.8592	34.5926	49	20
	S50_1514	Trains	1645	966	58.7234	35.7778	51	20
	S12_1666	Trucks and...	1579	972	61.5579	34.7143	49	20
	S12_4473	Trucks and...	6125	1056	17.2408	37.7143	54	21
	S18_1097	Trucks and...	2613	999	38.2319	35.6786	54	21
	S18_2319	Trucks and...	8258	1053	12.7513	37.6071	56	20
	S18_2432	Trucks and...	2018	998	49.4549	35.6429	50	19
	S18_4600	Trucks and...	3128	1061	33.9194	37.8929	50	21
	S24_2300	Trucks and...	2327	1029	44.2200	36.7500	70	20

From this query I got, 9 products have been ordered below 25% and average (30-38%).

productCode	productLine	quantityInStock	sum(o.quantityOrdered)	%_quantity_ordered	avg(o.quantityOrdered)
S18_3029	Ships	4259	966	22.6814	34.5000
S700_2047	Ships	3501	897	25.6213	32.0357
S700_2610	Ships	7083	1020	14.4007	37.7778
S700_3962	Ships	5088	896	17.6101	33.1852
S18_3259	Trains	6450	918	14.2326	34.0000
S32_3207	Trains	8601	934	10.8592	34.5926

S18_2319	Trucks and...	8258	1053	12.7513	37.6071
S12_4473	Trucks and...	6125	1056	17.2408	37.7143
S32_1268	Trucks and...	5099	911	17.8662	32.5357

So we may reduce these products if need. And I also analyzed their status.

```
select p.productCode, p.productLine, p.quantityInStock, sum(o.quantityOrdered),
sum(o.quantityOrdered)*100/p.quantityInStock as "%_quantity_ordered", oo.status from
products p join orderdetails o on p.productCode=o.productCode join orders oo on
o.orderNumber=oo.orderNumber where p.productLine="Trains" or p.productLine="Ships"
or p.productLine="Trucks and Buses" and oo.status="cancelled" group by productLine,
p.productCode, p.quantityInStock, oo.status order by p.productLine, oo.status
```

	productCode	productLine	quantityInStock	sum(o.quantityOrdered)	%_quantity_ordered	status
►	S18_3029	Ships	4259	99	2.3245	Cancelled
	S700_2047	Ships	3501	73	2.0851	Cancelled
	S24_2011	Ships	1898	81	4.2677	Cancelled
	S700_1138	Ships	1897	79	4.1645	Cancelled
	S700_1938	Ships	737	73	9.9050	Cancelled
	S700_2610	Ships	7083	78	1.1012	Cancelled
	S700_3505	Ships	1956	54	2.7607	Cancelled
	S700_3962	Ships	5088	63	1.2382	Cancelled
	S72_3212	Ships	414	61	14.7343	Cancelled

productCode	productLine	quantityInStock	sum(o.quantityOrdered)	%_quantity_ordered	status
S700_1138	Ships	1897	818	43.1207	Shipped
S700_1938	Ships	737	771	104.6133	Shipped
S18_3259	Trains	6450	71	1.1008	Cancelled
S32_3207	Trains	8601	27	0.3139	Cancelled
S50_1514	Trains	1645	21	1.2766	Cancelled

From this query we can conclude those 9 products that we got in previous query, status is “Cancelled”. Still they had not been moved from warehouses due to cancelled status even though they ordered. Mostly Ships product line had been cancelled.

Q3: Does the price seem to affect sales?





No. highest priced products does not affect sales. This query helps us obtain a list of products with the highest purchase prices, accompanied by the total quantity of products ordered for each of these products.

```
select p.productCode, p.productName, p.buyPrice, sum(o.quantityOrdered),p.buyPrice*
sum(o.quantityOrdered) as total_price from products p left join orderdetails o on
p.productCode=o.productCode group by p.productCode, p.productName,p.buyPrice order by
p.buyPrice desc;
```


	productCode	productName	buyPrice	sum(o.quantityOrdered)	total_price
►	S10_4962	1962 LanciaA Delta 16V	103.42	932	96387.44
	S18_2238	1998 Chrysler Plymouth Prowler	101.51	986	100088.86
	S10_1949	1952 Alpine Renault 1300	98.58	961	94735.38
	S24_3856	1956 Porsche 356A Coupe	98.30	1052	103411.60
	S12_1108	2001 Ferrari Enzo	95.59	1019	97406.21
	S12_1099	1968 Ford Mustang	95.34	933	88952.22
	S18_1984	1995 Honda Civic	93.89	917	86097.13
	S18_4027	1970 Triumph Spitfire	91.92	945	86864.40
	S10_4698	2003 Harley-Davidson Eagle Drag Bike	91.02	985	89654.70
	S12_3148	1969 Corvair Monza	89.14	963	85841.82
	S18_1749	1917 Grand Touring Sedan	86.70	918	79590.60
	S10_4757	1972 Alfa Romeo GTA	85.68	1030	88250.40
	S18_4600	1940s Ford truck	84.76	1061	89930.36
	S18_1129	1993 Mazda RX-7	83.51	947	79083.97

Q4: who are the customers ordered the products most and the total amount they purchased?

```
select c.customerNumber, c.customerName, count(o.orderNumber) as total_orders from
customers c, orders o where c.customerNumber=o.customerNumber group by
c.customerNumber, c.customerName order by total_orders desc;
```

Result Grid   Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 			
	customerNumber	customerName	total_orders
▶	141	Euro + Shopping Channel	26
	124	Mini Gifts Distributors Ltd.	17
	114	Australian Collectors, Co.	5
	353	Reims Collectables	5
	145	Danish Wholesale Imports	5
	148	Dragon Souvenirs, Ltd.	5
	323	Down Under Souvenirs, Inc	5
	381	Royale Belge	4
	276	Anna's Decorations, Ltd	4
	119	La Rochelle Gifts	4
	121	Baane Mini Imports	4
	128	Blauer See Auto, Co.	4
	131	Land of Toys Inc.	4
	144	Volvo Model Replicas, Co	4
	496	Kelly's Gift Shop	4

```
select customerNumber, customerName, sum(quantityOrdered*priceEach) as total_amount
from customers natural join orders natural join orderdetails group by customerNumber,
customerName order by total_amount desc;
```

Result Grid Filter Rows: Export: Wrap Cell Content:			
	customerNumber	customerName	total_amount
▶	141	Euro + Shopping Channel	820689.54
	124	Mini Gifts Distributors Ltd.	591827.34
	114	Australian Collectors, Co.	180585.07
	151	Muscle Machine Inc	177913.95
	119	La Rochelle Gifts	158573.12
	148	Dragon Souvenirs, Ltd.	156251.03
	323	Down Under Souvenirs, Inc	154622.08
	131	Land of Toys Inc.	149085.15
	187	AV Stores, Co.	148410.09
	450	The Sharp Gifts Warehouse	143536.27
	382	Salzburg Collectables	137480.07
	496	Kelly's Gift Shop	137460.79
	276	Anna's Decorations, Ltd	137034.22
	282	Souvenirs And Things Co.	133907.12
	321	Corporate Gift Ideas Co.	132340.78

Result 80 ×

	customerNumber	customerName	total_amount
	141	Euro + Shopping Channel	820689.54
▶	219	Boards & Toys Co.	7918.60

maxorderedvalue
minorderedvalue

Q5: how much profit are company earned for products?

Here I decided to get the productName, productLine, product code for total_profit above 50000.

```
select productName, productCode, productLine, sum((priceEach-  
buyPrice)*quantityOrdered) as total_profit from products natural join orderdetails group  
by productCode, productName having total_profit>50000 order by total_profit desc;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
productName	productCode	productLine	total_profit
1992 Ferrari 360 Spider red	S18_3232	Classic Cars	135996.78
1952 Alpine Renault 1300	S10_1949	Classic Cars	95282.58
2001 Ferrari Enzo	S12_1108	Classic Cars	93349.65
2003 Harley-Davidson Eagle Drag Bike	S10_4698	Motorcycles	81031.30
1968 Ford Mustang	S12_1099	Classic Cars	72579.26
1969 Ford Falcon	S12_3891	Classic Cars	72399.77
1928 Mercedes-Benz SSK	S18_2795	Vintage Cars	68423.18
2002 Suzuki XREO	S12_2823	Motorcycles	67641.47
1980s Black Hawk Helicopter	S18_1662	Planes	64599.11
1948 Porsche Type 356 Roadster	S18_3685	Classic Cars	62725.78
1917 Grand Touring Sedan	S18_1749	Vintage Cars	60945.00
1957 Corvette Convertible	S18_4721	Classic Cars	59910.22
1962 Volkswagen Microbus	S24_2300	Trucks and ...	55655.47
1976 Ford Gran Torino	S18_3482	Classic Cars	54647.25

Q6: How can analyze the performance of employees on sales?

This query helps to predict which employee contribute the most to the sales.

```
select e.employeeNumber, e.firstName, e.jobTitle, sum(o.quantityOrdered*o.priceEach) as  
total_sale_amount from employees e left join customers c on  
e.employeeNumber=c.salesRepEmployeeNumber left join orders d on  
d.customerNumber=c.customerNumber left join orderdetails o on  
o.orderNumber=d.orderNumber group by e.employeeNumber, e.firstName, e.jobTitle order  
by total_sale_amount desc;
```


Result Grid   Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 				
	employeeNumber	firstName	jobTitle	total_sale_amount
▶	1370	Gerard	Sales Rep	1258577.81
	1165	Leslie	Sales Rep	1081530.54
	1401	Pamela	Sales Rep	868220.55
	1501	Larry	Sales Rep	732096.79
	1504	Barry	Sales Rep	704853.91
	1323	George	Sales Rep	669377.05
	1612	Peter	Sales Rep	584593.76
	1337	Loui	Sales Rep	569485.75
	1611	Andy	Sales Rep	562582.59
	1216	Steve	Sales Rep	505875.42
	1286	Foon Yue	Sales Rep	488212.67
	1621	Mami	Sales Rep	457110.07
	1702	Martin	Sales Rep	387477.47
	1188	Julie	Sales Rep	386663.20
	1166	Leslie	Sales Rep	347533.03

The following query helps to understand employees who do not have any customers.

```

SELECT lastName, firstName, employeeNumber FROM Employees
EXCEPT
SELECT DISTINCT lastName, firstName, employeeNumber
FROM Employees E INNER JOIN Customers C
ON E.employeeNumber = C.salesrepemployeenumber
ORDER BY lastname, firstname;

```

	lastName	firstName	employeeNumber
▶	Bondur	Gerard	1102
	Bow	Anthony	1143
	Firrelli	Jeff	1076
	Kato	Yoshimi	1625
	King	Tom	1619
	Murphy	Diane	1002
	Patterson	Mary	1056
	Patterson	William	1088

Q7: How can performance of the product Line to be analyzed?

select p.productLine, p.quantityInStock, sum(o.quantityOrdered) from products p,
 orderdetails o where p.productCode=o.productCode group by
 p.productLine,p.quantityInStock order by sum(o.quantityOrdered) desc;

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content: <input type="checkbox"/>			
	productLine	quantityInStock	sum(o.quantityOrdered)
▶	Classic Cars	8347	1808
	Vintage Cars	8693	1111
	Planes	8820	1085
	Vintage Cars	2378	1076
	Vintage Cars	7062	1074
	Trucks and Buses	3128	1061
	Motorcycles	7933	1057
	Trucks and Buses	6125	1056
	Trucks and Buses	8258	1053

select sum(quantityOrdered) from orderdetails;

Result Grid Filter Rows: <input type="text"/>	
	sum(quantityOrdered)
▶	105516




select sum(o.quantityOrdered) as shipped_products_count from orderdetails o, orders d
where o.orderNumber=d.orderNumber and d.status="Shipped";

Result Grid Filter Rows: <input type="text"/>	
	shipped_products_count
▶	97141



select p.productLine, sum(o.quantityOrdered) as QuantityOrdered, d.status from orderdetails
o, orders d, products p where o.orderNumber=d.orderNumber and
p.productCode=o.productCode and d.status="Shipped" group by p.productLine;

	productLine	QuantityOrdered	status
▶	Vintage Cars	21015	Shipped
	Classic Cars	33349	Shipped
	Trucks and Buses	10233	Shipped
	Trains	2651	Shipped
	Ships	7131	Shipped
	Planes	10337	Shipped
	Motorcycles	12425	Shipped

select p.productLine, sum(o.quantityOrdered) from products p, orderdetails o where p.productCode=o.productCode group by productLine;

Result Grid   Filter Rows: <input type="text"/> Export:  W		
	productLine	sum(o.quantityOrdered)
▶	Classic Cars	35582
	Motorcycles	12778
	Planes	11872
	Ships	8532
	Trains	2818
	Trucks and Buses	11001
	Vintage Cars	22933

select sum(quantityOrdered*priceEach) as total_revenue from orderdetails;

Result Grid				Filter
	total_revenue			
▶	9604190.61			

Solution:

Here I have conducted data analysis and recommended some solution for the business problem. The following summary based on the questions that I used for analysis.

1. Warehouse Review:

With conducted of analysis, I recommend to close or move items from warehouse “d”. Because there is less quantity of stock under three product line named “Trucks and Buses”, “Trains”, “Ships”. Mostly under product line Trains and ships, first the products had been ordered and then had been cancelled. So we can reduce those products from warehouses and move remaining products to other warehouses. Through we can reduce the rental cost of warehouse “d”.

2. Product Price Review:

After analyzing, there is no effects on sales even though their price values are high. But still we have to consider the price of low sales. So I recommend, reducing the price of low sales may attract the customers to order further.

3. Customer Review:

From this we can note who the valuable customers to sales are. So I suggest, provide some special offers for the products they purchase in future. This will make them to feel more valuable customer and encourage them confidently to purchase products frequently.

4. Performance of Employees Review:

By analyzing of employees performance can improve the company’s growth and profit. So I recommend from the analysis, provide incentives for the employees who have high sales amount targets and provide some extra support and training for the employees who do not have any customers.