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:

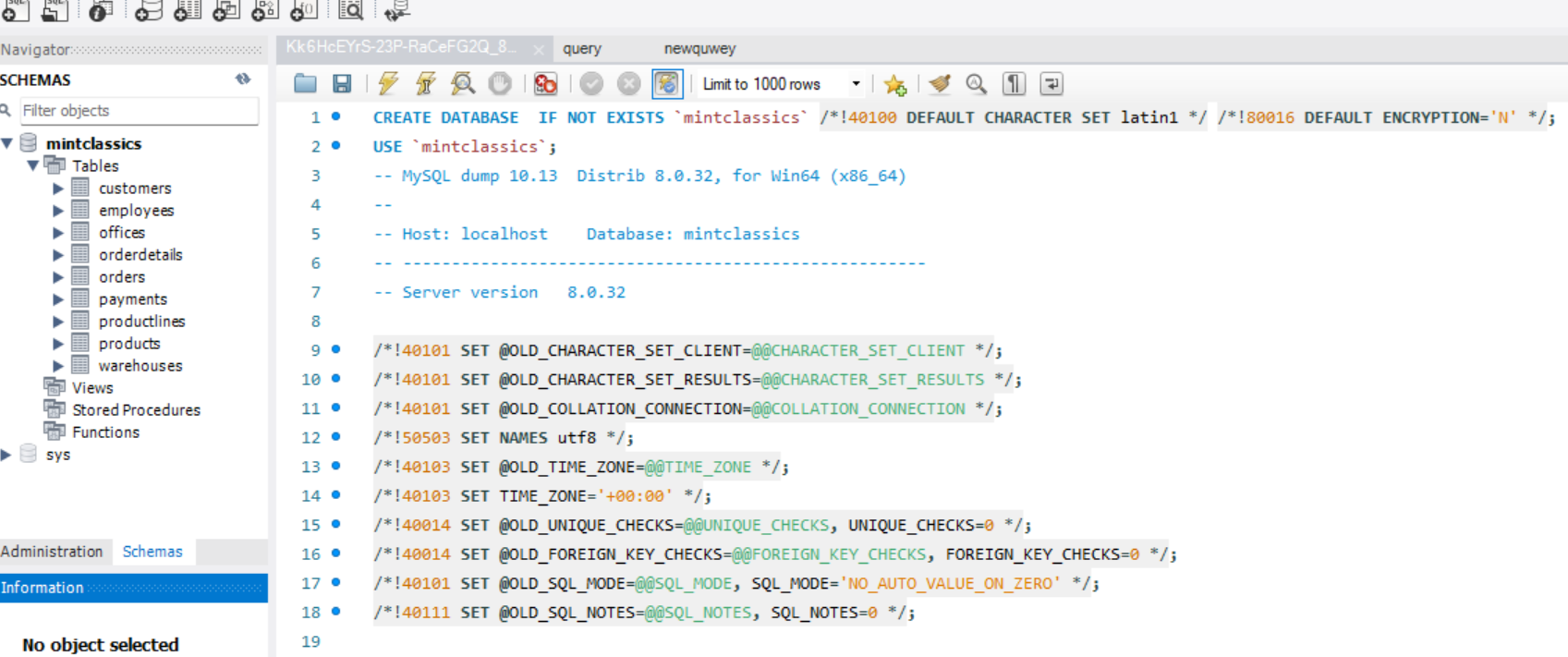
* Using SQL queries to extract and analyze data in mint classics relational database.
* Identifying and isolating the parts of data which is useful in deciding how to reducing or closing of one of the warehouses.

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**

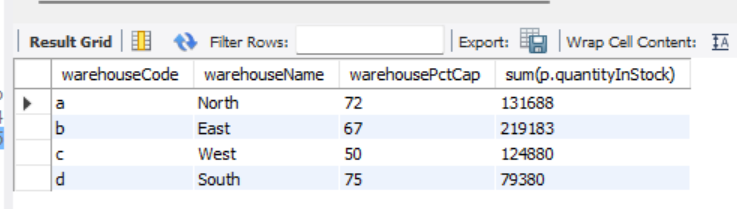
The main issue of the company is to plan to close the one of their storage facility or reducing products. Based on that I created and runned SQL queries to get the data for their problem.

**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?**

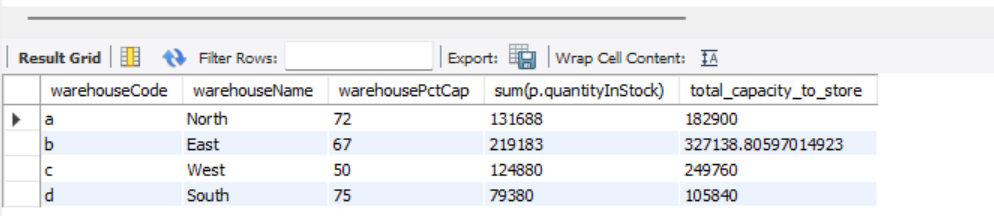
Here I chose warehousecode, warehousename, warehouse percentage capacity from warehouses table and quantityinstock from products table. From this we can know how much quantity of stocks are stored in each warehouses and their percentage.

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



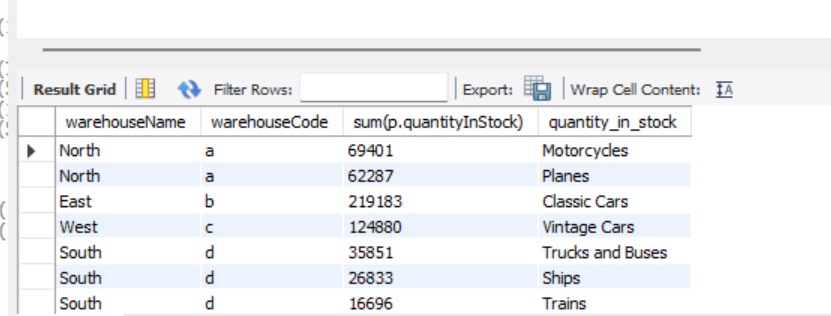
Then I calculated 100 percentage capacity for each warehouse. From that I concluded warehouse code “d” named “South” having less quantity of stock and less capacity to store when moving products from other warehouse.

|  |
| --- |
| 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; |



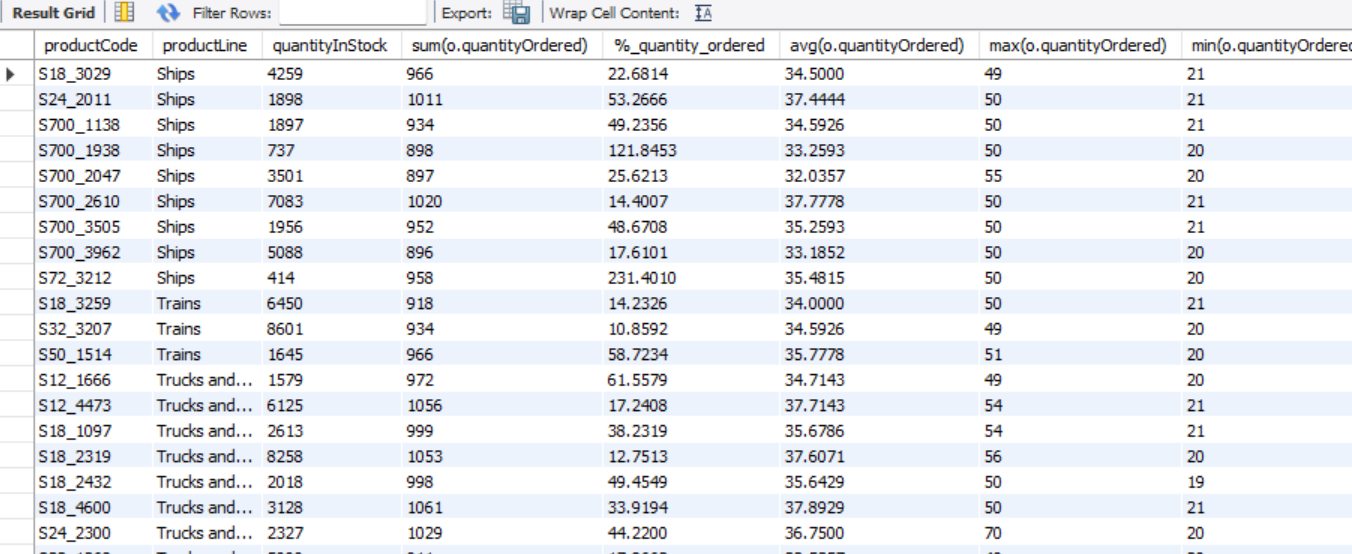
**Q2: what products category stored in each warehouse and their total quantity of stock?**

|  |
| --- |
| 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; |



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; |



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











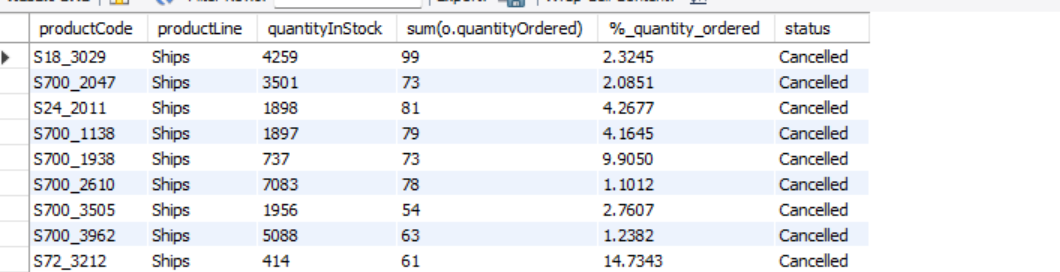


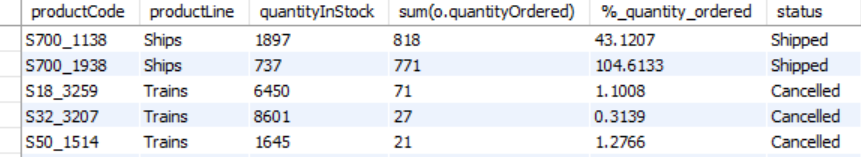




So we may reduce these products if need. And I also analyzed their status.

|  |
| --- |
| 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 |



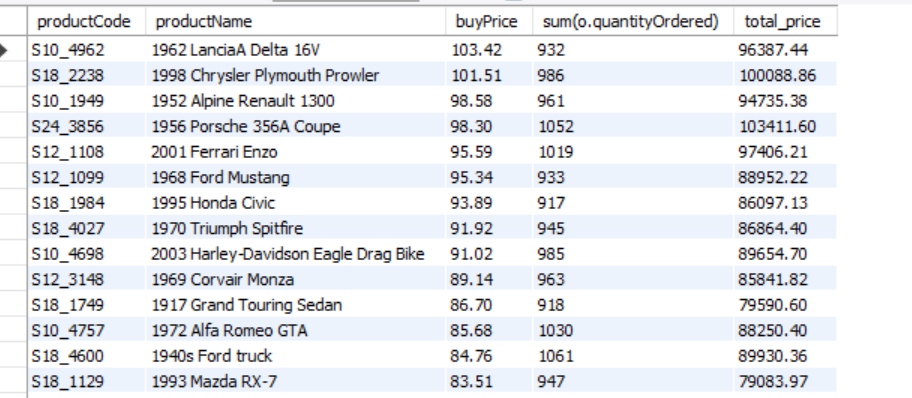


From this query we can conclude those 9 products that we got in previous query, status is “Cancelled”. Still they had not been moved from warehouses due to cancelled status even though they ordered. Mostly Ships product line had been cancelled.

**Q3: Does the price seem to affect sales?**

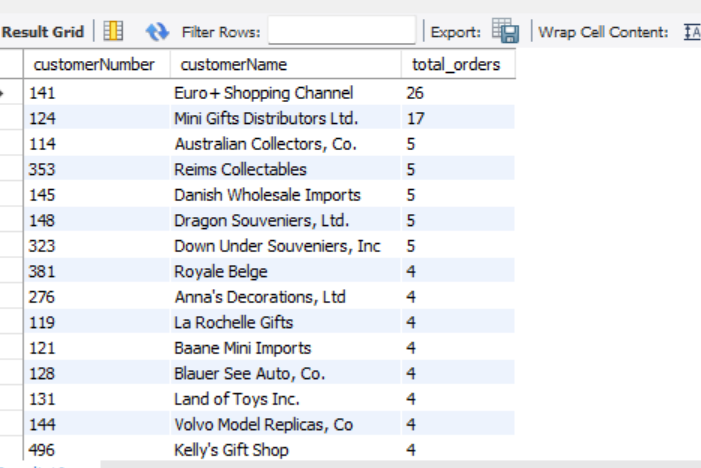
No. highest priced products does not affect sales. This query helps us obtain a list of products with the highest purchase prices, accompanied by the total quantity of products ordered for each of these products.

|  |
| --- |
| 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; |

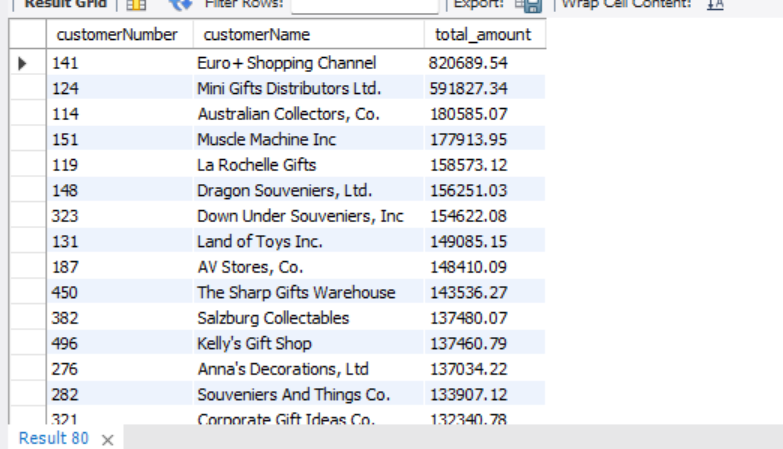


**Q4: who are the customers ordered the products most and the total amount they purchased?**

|  |
| --- |
| 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; |



|  |
| --- |
| 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; |



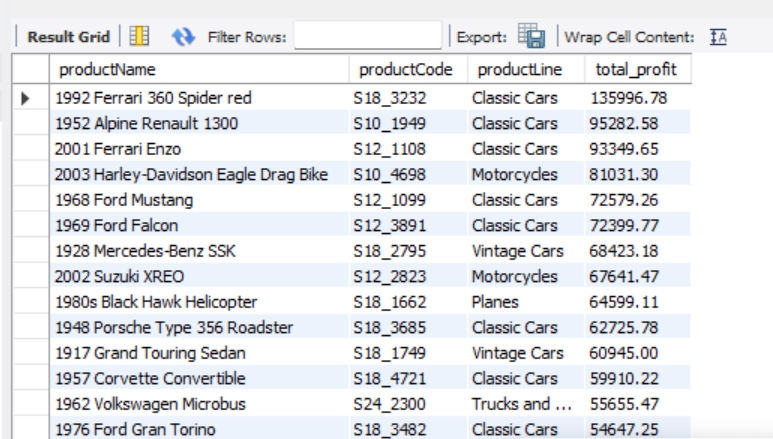
maxorderedvalue

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.

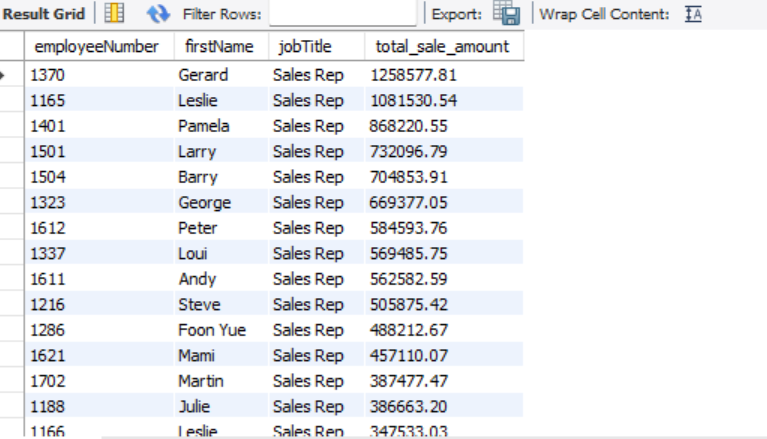
|  |
| --- |
| select productName, productCode, productLine, sum((priceEach-buyPrice)\*quantityOrdered) as total\_profit from products natural join orderdetails group by productCode, productName having total\_profit>50000 order by total\_profit desc; |



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

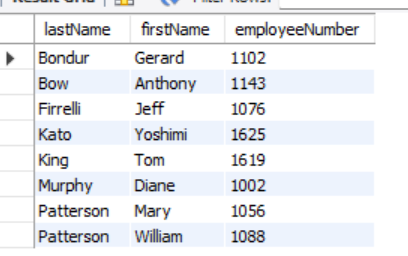
This query helps to predict which employee contribute the most to the sales.

|  |
| --- |
| 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; |



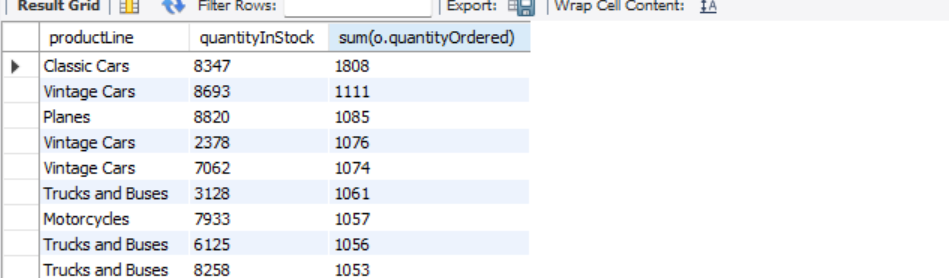
The following query helps to understand employees who do not have any customers.

|  |
| --- |
| 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; |

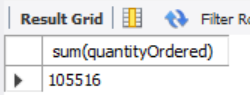


**Q7: How can performance of the product Line to be analyzed?**

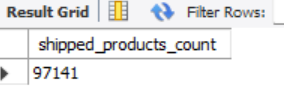
select p.productLine, p.quantityInStock, sum(o.quantityOrdered) from products p, orderdetails o where p.productCode=o.productCode group by p.productLine,p.quantityInStock order by sum(o.quantityOrdered) desc;



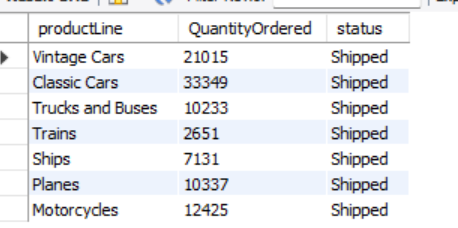
select sum(quantityOrdered) from orderdetails;



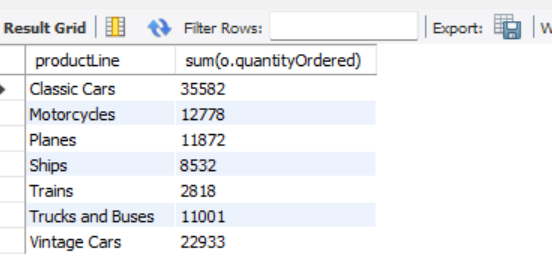
select sum(o.quantityOrdered) as shipped\_products\_count from orderdetails o, orders d where o.orderNumber=d.orderNumber and d.status="Shipped";



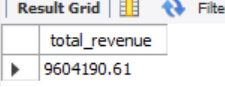
select p.productLine, sum(o.quantityOrdered) as QuantityOrdered, d.status from orderdetails o, orders d, products p where o.orderNumber=d.orderNumber and p.productCode=o.productCode and d.status="Shipped" group by p.productLine;



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



select sum(quantityOrdered\*priceEach) as total\_revenue from orderdetails;



**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”.

1. **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.

1. **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.

1. **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.