

The Mint Classics Project

MySQL Workbench

Pedro Goncalves

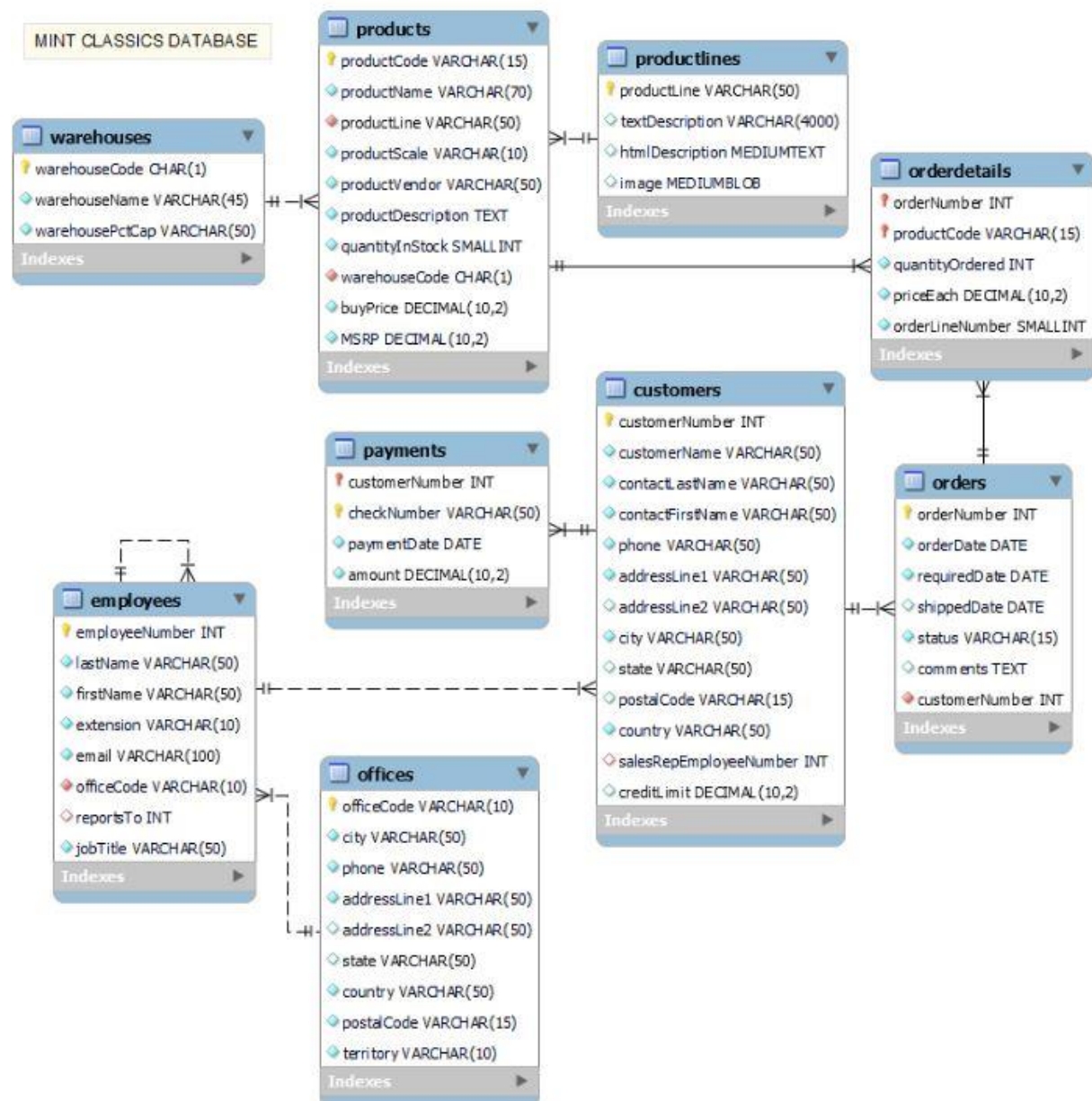
Contents

Overview	3
Database ERD.....	4
- 1. Database “mintclassics” exploration	5
- 2. Exploring distinct products and warehouses.....	6
2.1 - 110 distinct products	6
2.2 – Warehouses	6
2.3 - Products total stock	6
-3. Determine which products sell the most/least	7
3.1 - Which products have the most number of orders?	7
3.2 - Which products have the least number of orders?.....	7
3.3 - Which product lines had the highest sales in terms of both number of pieces sold and their value?.....	8
3.4 - Which product lines generated a lower total value of sales?	9
3.5 - Which products generated a lower total value of sales?	9
-4. Inventory versus Sales.....	10
4.1 - Which products have high inventory but low sales?	10
4.2 - Which products have inventory shortage but high sales?	11
-5. Review Inventory	12
5.1 – Warehouses total inventory	12
5.2 - Warehouse inventory by product	13
5.3 - Warehouses inventory by product line	14
-6. Which products/product lines can be dropped out?	15
6.1. Products Inventory to Sales Ratio.....	15
6.2 Products Lines Inventory to Sales Ratio.....	17
-7. Analysis and Recommendations	18

Overview

As a data analyst, I am tasked with analysing data in a relational database to assist in making inventory-related business decisions. Mint Classics Company, a retailer of classic model cars and other vehicles, is considering shutting down one of its storage facilities. To get a better understanding of the business, I am using MySQL Workbench to examine the current data.

Database ERD



- 1. Database “mintclassics” exploration

```

USE mintclassics;
SELECT
    orders.orderNumber,
    customers.customerName,
    orders.orderDate,
    products.productName,
    products.productCode,
    products.productLine,
    products.buyPrice,
    products.quantityInStock,
    orderdetails.quantityOrdered,
    orderdetails.priceEach,
    (orderdetails.quantityOrdered * orderdetails.priceEach) AS TotalOrder,
    payments.paymentDate,
    orders.shippedDate,
    timestampdiff(day, orders.orderDate, orders.shippedDate) AS LeadTime,
    concat(employees.firstName, " ", employees.lastName) AS "SalesRep",
    offices.country AS Country
FROM products
JOIN productlines ON products.productLine = productlines.productLine
JOIN warehouses ON products.warehouseCode = warehouses.warehouseCode
JOIN orderdetails ON products.productCode = orderdetails.productCode
JOIN orders ON orders.orderNumber = orderdetails.orderNumber
JOIN customers ON customers.customerNumber = orders.customerNumber
JOIN payments ON customers.customerNumber = payments.customerNumber
JOIN employees ON employees.employeeNumber = customers.salesRepEmployeeNumber
JOIN offices ON offices.officeCode = employees.officeCode
ORDER BY orders.orderdate ASC;

```

Table view left.

orderNumber	customerName	orderDate	productName	productCode	productLine	buyPrice
10100	Online Diecast Creations Co.	2003-01-06	1917 Grand Touring Sedan	S18_1749	Vintage Cars	86.70
10100	Online Diecast Creations Co.	2003-01-06	1911 Ford Town Car	S18_2248	Vintage Cars	33.30
10100	Online Diecast Creations Co.	2003-01-06	1932 Alfa Romeo 8C2300 Spider Sport	S18_4409	Vintage Cars	43.26
10100	Online Diecast Creations Co.	2003-01-06	1936 Mercedes Benz 500k Roadster	S24_3969	Vintage Cars	21.75
10100	Online Diecast Creations Co.	2003-01-06	1917 Grand Touring Sedan	S18_1749	Vintage Cars	86.70
10100	Online Diecast Creations Co.	2003-01-06	1911 Ford Town Car	S18_2248	Vintage Cars	33.30

Table view right.

quantityInStock	quantityOrdered	priceEach	TotalOrder	paymentDate	shippedDate	LeadTime	SalesRep	Country
2724	30	136.00	4080.00	2004-11-17	2003-01-10	4	Steve Patterson	USA
540	50	55.09	2754.50	2004-11-17	2003-01-10	4	Steve Patterson	USA
6553	22	75.46	1660.12	2004-11-17	2003-01-10	4	Steve Patterson	USA
2081	49	35.29	1729.21	2004-11-17	2003-01-10	4	Steve Patterson	USA
2724	30	136.00	4080.00	2003-01-16	2003-01-10	4	Steve Patterson	USA
540	50	55.09	2754.50	2003-01-16	2003-01-10	4	Steve Patterson	USA

- 2. Exploring distinct products and warehouses

2.1- 110 distinct products

```
SELECT COUNT(products.productCode) as products
FROM products
  JOIN productlines ON products.productLine = productlines.productLine
  JOIN warehouses ON products.warehouseCode = warehouses.warehouseCode;
```

	products
▶	110

2.2 – Warehouses

```
SELECT DISTINCT
  warehouses.warehouseCode,
  (warehouses.warehouseName) AS Warehouse
FROM products
  JOIN productlines ON products.productLine = productlines.productLine
  JOIN warehouses ON products.warehouseCode = warehouses.warehouseCode;
```

warehouseCode	Warehouse
a	North
b	East
c	West
d	South

2.3- Products total stock

```
SELECT SUM(products.quantityInStock) AS products_total_stock
FROM products
  JOIN productlines ON products.productLine = productlines.productLine
  JOIN warehouses ON products.warehouseCode = warehouses.warehouseCode;
```

	products_total_stock
▶	555131

-3. Determine which products sell the most/least

3.1- Which products have the most number of orders?

```
SELECT productsCode, productName, COUNT(orderNumber) AS total_orders
FROM products
JOIN orderdetails ON products.productCode = orderdetails.productCode
GROUP BY productName
ORDER BY total_orders DESC;
```

Example

productName	total_orders
1992 Ferrari 360 Spider red	53
1969 Harley Davidson Ultimate Chopper	28
1952 Alpine Renault 1300	28
1996 Moto Guzzi 1100i	28
2003 Harley-Davidson Eagle Drag Bike	28
1972 Alfa Romeo GTA	28

3.2- Which products have the least number of orders?

```
SELECT productName, COUNT(orderNumber) AS total_orders
FROM products
JOIN orderdetails ON products.productCode = orderdetails.productCode
GROUP BY productName
ORDER BY total_orders ASC;
```

productName	total_orders
1957 Ford Thunderbird	24
1952 Citroen-15CV	24
1965 Aston Martin DB5	25
1917 Grand Touring Sedan	25
1911 Ford Town Car	25
1999 Indy 500 Monte Carlo SS	25

3.3- Which **product lines** had the highest sales in terms of both number of pieces sold and their value?

Number of Pieces

```
SELECT products.productLine, SUM(quantityOrdered) AS total_units_sold
FROM products
    JOIN productlines ON products.productLine = productlines.productLine
    JOIN orderdetails ON products.productCode = orderdetails.productCode
    JOIN orders ON orders.orderNumber = orderdetails.orderNumber
GROUP BY products.productLine
ORDER BY total_units_sold DESC;
```

productLine	total_units_sold
Classic Cars	35582
Vintage Cars	22933
Motorcycles	12778
Planes	11872
Trucks and Buses	11001
Ships	8532
Trains	2818

Value

```
SELECT products.productLine, SUM(amount) AS total_value
FROM products
    JOIN orderdetails ON products.productCode = orderdetails.productCode
    JOIN orders ON orders.orderNumber = orderdetails.orderNumber
    JOIN customers ON orders.customerNumber = customers.customerNumber
    JOIN payments ON customers.customerNumber = payments.customerNumber
GROUP BY products.productLine
ORDER BY total_value DESC;
```

productLine	total_value
Classic Cars	195715285.48
Vintage Cars	112040838.12
Trucks and Buses	65447435.69
Motorcycles	48280743.43
Planes	44171665.16
Ships	39892604.12
Trains	19127343.02

3.4- Which product lines generated a lower total value of sales?

```
SELECT products.productLine, SUM(amount) AS total_value
FROM products
  JOIN productlines ON products.productLine = productlines.productLine
  JOIN orderdetails ON products.productCode = orderdetails.productCode
  JOIN orders ON orders.orderNumber = orderdetails.orderNumber
  JOIN customers ON customers.customerNumber = orders.customerNumber
  JOIN payments ON payments.customerNumber = customers.customerNumber
GROUP BY products.productLine
ORDER BY total_value ASC;
```

productLine	total_value
Trains	19127343.02
Ships	39892604.12
Planes	44171665.16
Motorcycles	48280743.43
Trucks and Buses	65447435.69
Vintage Cars	112040838.12
Classic Cars	195715285.48

3.5- Which products generated a lower total value of sales?

```
SELECT products.productCode, products.productName, SUM(amount) AS total_value
FROM products
  JOIN productlines ON products.productLine = productlines.productLine
  JOIN orderdetails ON products.productCode = orderdetails.productCode
  JOIN orders ON orders.orderNumber = orderdetails.orderNumber
  JOIN customers ON customers.customerNumber = orders.customerNumber
  JOIN payments ON payments.customerNumber = customers.customerNumber
GROUP BY 1, 2
ORDER BY total_value ASC;
```

productCode	productName	total_value
S24_2000	1960 BSA Gold Star DBD34	2713978.68
S10_4757	1972 Alfa Romeo GTA	2768327.28
S12_2823	2002 Suzuki XREO	3166150.70
S700_2834	ATA: B757-300	3202828.79
S32_1374	1997 BMW F650 ST	3230427.96
S18_3856	1941 Chevrolet Special Deluxe Cabriolet	3288253.67
S10_2016	1996 Moto Guzzi 1100i	3293715.22

-4. Inventory versus Sales

4.1- Which products have high inventory but low sales?

```
USE mintclassics;
SELECT
    productCode,
    productName,
    productLine,
    quantityInStock,
    totalOrdered,
    (quantityInStock - totalOrdered) AS inventory
FROM
    (SELECT
        pr.productCode,
        pr.productName,
        pr.productLine,
        pr.quantityInStock,
        SUM(ord.quantityOrdered) AS totalOrdered
    FROM mintclassics.products AS pr
    LEFT JOIN mintclassics.orderdetails AS ord ON pr.productCode = ord.productCode
    GROUP BY
        pr.productCode,
        pr.productName,
        pr.quantityInStock
    ) AS inventoryData
WHERE (quantityInStock - totalOrdered) > 0
```

productCode	productName	quantityInStock	totalOrdered	inventory
S12_2823	2002 Suzuki XREO	9997	1028	8969
S18_1984	1995 Honda Civic	9772	917	8855
S700_2466	America West Airlines B757-200	9653	984	8669
S24_3432	2002 Chevy Corvette	9446	894	8552
S18_2325	1932 Model A Ford J-Coupe	9354	957	8397
S32_2206	1982 Ducati 996 R	9241	906	8335
S18_3482	1976 Ford Gran Torino	9127	915	8212
S12_3380	1968 Dodge Charger	9123	925	8198
S24_3151	1912 Ford Model T Delivery Wagon	9173	991	8182

This query effectively determines the surplus inventory of a product by calculating the difference between the quantity available in stock and the total quantity ordered. The resulting value is labelled as "inventory." The query then selects only the products that have a surplus inventory, with the condition that the difference must be greater than 0. The results are sorted in descending order based on the magnitude of the inventory.

4.2- Which products have inventory shortage but high sales?

```
USE mintclassics;
SELECT
    productCode,
    productName,
    productLine,
    quantityInStock,
    totalOrdered,
    (quantityInStock - totalOrdered) AS inventory
FROM
    (SELECT
        pr.productCode,
        pr.productName,
        pr.productLine,
        pr.quantityInStock,
        SUM(ord.quantityOrdered) AS TotalOrdered
    FROM mintclassics.products AS pr
    LEFT JOIN orderdetails AS ord ON pr.productCode = ord.productCode
    GROUP BY
        pr.productCode,
        pr.productName,
        pr.productLine,
        pr.quantityInStock
    ) AS inventoryData
WHERE (quantityInStock - totalOrdered) < 0
ORDER BY inventory ASC;
```

This query effectively determines the inventory shortage of a product by calculating the difference between the quantity available in stock and the total quantity ordered. The resulting value is labelled as "inventory." The query then selects only the products that have a shortage in stock, with the condition that the difference must be greater than 0. The results are sorted in ascending order based on the magnitude of the inventory shortage.

productCode	productName	productLine	quantityInStock	totalOrdered	inventory
S24_2000	1960 BSA Gold Star DBD34	Motorcycles	15	1015	-1000
S12_1099	1968 Ford Mustang	Classic Cars	68	933	-865
S32_1374	1997 BMW F650 ST	Motorcycles	178	1014	-836
S32_4289	1928 Ford Phaeton Deluxe	Vintage Cars	136	972	-836
S72_3212	Pont Yacht	Ships	414	958	-544
S700_3167	F/A 18 Hornet 1/72	Planes	551	1047	-496
S50_4713	2002 Yamaha YZR M1	Motorcycles	600	992	-392
S18_2795	1928 Mercedes-Benz SSK	Vintage Cars	548	880	-332
S18_2248	1911 Ford Town Car	Vintage Cars	540	832	-292

-5. Review Inventory

5.1 – Warehouses **total** inventory

```
USE mintclassics;
SELECT
    warehouses.warehouseCode,
    warehouses.warehouseName,
    SUM(products.quantityInStock) AS totalInventory
FROM products
    JOIN productlines ON products.productLine = productlines.productLine
    JOIN warehouses ON products.warehouseCode = warehouses.warehouseCode
GROUP BY 1
ORDER BY 3 DESC;
```

The query results in a list of warehouses with the highest total inventory at the top.

warehouseCode	warehouseName	totalInventory
b	East	219183
a	North	131688
c	West	124880
d	South	79380

5.2- Warehouse inventory by product

```
USE mintclassics;
SELECT
    products.productCode,
    products.productName,
    warehouses.warehouseCode,
    warehouses.warehouseName,
    SUM(products.quantityInStock) AS stock
FROM products
    JOIN productlines ON products.productLine = productlines.productLine
    JOIN warehouses ON products.warehouseCode = warehouses.warehouseCode
GROUP BY 1, 2, 3
ORDER BY 5 ASC;
```

The query sorts the total inventory of each product in each warehouse from smallest to largest.

productCode	productName	warehouseCode	warehouseName	stock
S24_2000	1960 BSA Gold Star DBD34	a	North	15
S12_1099	1968 Ford Mustang	b	East	68
S32_4289	1928 Ford Phaeton Deluxe	c	West	136
S32_1374	1997 BMW F650 ST	a	North	178
S72_3212	Pont Yacht	d	South	414
S18_2248	1911 Ford Town Car	c	West	540
S18_2795	1928 Mercedes-Benz SSK	c	West	548
S700_3167	F/A 18 Hornet 1/72	a	North	551
S50_4713	2002 Yamaha YZR M1	a	North	600
S700_1938	The Mayflower	d	South	737
S32_3522	1996 Peterbilt 379 Stake ...	d	South	814

5.3- Warehouses inventory by product line

```
SELECT
    productlines.productLine,
    warehouses.warehouseCode,
    warehouses.warehouseName,
    SUM(products.quantityInStock) AS inventory
FROM products
    JOIN productlines ON products.productLine = productlines.productLine
    JOIN warehouses ON products.warehouseCode = warehouses.warehouseCode
GROUP BY 1, 2
ORDER BY 4 DESC;
```

This query sorts the total inventory of each product line in each warehouse from smallest to largest.

productLine	warehouseCode	warehouseName	inventory
Classic Cars	b	East	219183
Vintage Cars	c	West	124880
Motorcycles	a	North	69401
Planes	a	North	62287
Trucks and Buses	d	South	35851
Ships	d	South	26833
Trains	d	South	16696

-6. Which products/product lines can be dropped out?

6.1. Products Inventory to Sales Ratio

A good ratio of product inventory to sales is less than 1, and businesses typically aim for a ratio between 0.17 and 0.25. The ratio can be expressed as a fraction or a decimal. Generally, the closer the ratio is to 0, the better for a business, as it indicates that sales are high, and inventory is relatively low.

Inventory to Sales Ratio = (Inventory Level) / (Net Sales)

The year 2003

```
USE mintclassics;
SELECT
    products.productCode,
    products.productName,
    products.productLine,
    SUM(products.quantityInStock) AS Inventory,
    SUM(orderdetails.quantityOrdered) AS quantitySold,
    SUM(orderdetails.priceEach * orderdetails.quantityOrdered) AS netSales,
    (SUM(products.quantityInStock)/SUM(orderdetails.priceEach * orderdetails.quantityOrdered)) AS InventoryToSalesRatio
FROM products
JOIN orderdetails ON products.productCode = orderdetails.productCode
JOIN orders ON orders.orderNumber = orderdetails.orderNumber
WHERE YEAR(orders.orderDate) <> 2004 AND YEAR(orders.orderDate) <> 2005 -- YEAR 2003
GROUP BY 1, 3
ORDER BY 7 ASC;
```

The ASC command was utilized to exhibit products that have a good inventory ratio, low inventory, and strong sales. However, it could also be the outcome of inadequate inventory. Out of all the problems a company can face, ensuring that it has enough inventory to support robust sales is better than having to reduce inventory because the business is lagging.

productCode	productName	productLine	Inventory	quantitySold	netSales	InventoryToSalesRatio
S24_2000	1960 BSA Gold Star DBD34	Motorcycles	135	306	20930.66	0.0064
S12_1099	1968 Ford Mustang	Classic Cars	680	337	56462.25	0.0120
S32_1374	1997 BMW F650 ST	Motorcycles	1602	284	24739.63	0.0648
S32_4289	1928 Ford Phaeton Deluxe	Vintage Cars	1224	272	17150.96	0.0714
S18_2795	1928 Mercedes-Benz SSK	Vintage Cars	5480	308	46332.65	0.1183
S12_3891	1969 Ford Falcon	Classic Cars	9441	363	57403.47	0.1645
S700_3167	F/A 18 Hornet 1/72	Planes	4959	353	26292.80	0.1886
S50_4713	2002 Yamaha YZR M1	Motorcycles	5400	340	25501.12	0.2118
S72_3212	Pont Yacht	Ships	3726	344	16729.57	0.2227

The year 2004

```
USE mintclassics;
SELECT
    products.productCode,
    products.productName,
    products.productLine,
    SUM(products.quantityInStock) AS Inventory,
    SUM(orderdetails.quantityOrdered) AS quantitySold,
    SUM(orderdetails.priceEach * orderdetails.quantityOrdered) AS netSales,
    (SUM(products.quantityInStock)/SUM(orderdetails.priceEach * orderdetails.quantityOrdered)) AS InventoryToSalesRatio
FROM products
JOIN orderdetails ON products.productCode = orderdetails.productCode
JOIN orders ON orders.orderNumber = orderdetails.orderNumber
WHERE YEAR(orders.orderDate) <> 2003 AND YEAR(orders.orderDate) <> 2005 -- YEAR 2004
GROUP BY 1, 3
ORDER BY 7 DESC;
```

In this scenario, the DESC command was utilized to exhibit products that don't have a good ratio, with high inventory and low sales and are eligible to be discontinued. A low inventory turnover ratio may indicate weak sales or excessive inventory, which is also known as overstocking. This could imply an issue with the retail chain's merchandising strategy or insufficient marketing.

productCode	productName	productLine	Inventory	quantitySold	netSales	InventoryToSalesRatio
S32_2206	1982 Ducati 996 R	Motorcycles	120133	414	15076.58	7.9682
S24_1937	1939 Chevrolet Deluxe Coupe	Vintage Cars	95316	410	12206.66	7.8085
S24_2972	1982 Lamborghini Diablo	Classic Cars	92676	409	14058.02	6.5924
S18_1367	1936 Mercedes-Benz 500K Special Roadster	Vintage Cars	112255	363	17362.43	6.4654
S24_1628	1966 Shelby Cobra 427 S/C	Classic Cars	90167	387	17748.66	5.0802
S50_1341	1930 Buick Marquette Phaeton	Vintage Cars	98868	535	20430.48	4.8392
S32_3207	1950's Chicago Surface Lines Streetcar	Trains	111813	434	25013.54	4.4701
S24_3371	1971 Alpine Renault 1600s	Classic Cars	95940	411	22897.96	4.1899
S18_4668	1939 Cadillac Limousine	Vintage Cars	86385	462	20678.32	4.1776
S18_1889	1948 Porsche 356-A Roadster	Classic Cars	105912	404	27882.47	3.7985
S18_3782	1957 Vespa GS150	Motorcycles	99957	471	26770.90	3.7338

6.2 Products Lines Inventory to Sales Ratio

The year 2003

```
USE mintclassics;
SELECT
    products.productLine,
    SUM(products.quantityInStock) AS Inventory,
    SUM(orderdetails.quantityOrdered) AS quantitySold,
    SUM(orderdetails.priceEach * orderdetails.quantityOrdered) AS netSales,
    (SUM(products.quantityInStock)/SUM(orderdetails.priceEach * orderdetails.quantityOrdered)) AS InventoryToSalesRatio
FROM products
JOIN orderdetails ON products.productCode = orderdetails.productCode
JOIN orders ON orders.orderNumber = orderdetails.orderNumber
WHERE YEAR(orders.orderDate) <> 2004 AND YEAR(orders.orderDate) <> 2005 -- YEAR 2003
GROUP BY 1
ORDER BY 5 ASC;
```

productLine	Inventory	quantitySold	netSales	InventoryToSalesRatio
Trucks and Buses	358510	4056	376657.12	0.9518
Ships	241497	2844	222182.08	1.0869
Classic Cars	2174026	12762	1374832.22	1.5813
Motorcycles	624609	4031	348909.24	1.7902
Planes	560583	3833	309784.20	1.8096
Vintage Cars	1217602	7913	619161.48	1.9665
Trains	158865	1000	65822.05	2.4136

The year 2004

```
USE mintclassics;
SELECT
    products.productLine,
    SUM(products.quantityInStock) AS Inventory,
    SUM(orderdetails.quantityOrdered) AS quantitySold,
    SUM(orderdetails.priceEach * orderdetails.quantityOrdered) AS netSales,
    (SUM(products.quantityInStock)/SUM(orderdetails.priceEach * orderdetails.quantityOrdered)) AS InventoryToSalesRatio
FROM products
JOIN orderdetails ON products.productCode = orderdetails.productCode
JOIN orders ON orders.orderNumber = orderdetails.orderNumber
WHERE YEAR(orders.orderDate) <> 2003 AND YEAR(orders.orderDate) <> 2005 -- YEAR 2004
GROUP BY 1
ORDER BY 5 ASC;
```

productLine	Inventory	quantitySold	netSales	InventoryToSalesRatio
Trucks and Buses	466063	5024	465390.00	1.0014
Ships	375662	4309	337326.10	1.1136
Classic Cars	2659620	16085	1763136.73	1.5085
Motorcycles	916771	5976	527243.84	1.7388
Planes	872018	5820	471971.46	1.8476
Vintage Cars	1630842	10864	854551.85	1.9084
Trains	225143	1409	96285.53	2.3383

-7. Simplified Data Analysis and Recommendations

There is a high inventory but low sales of certain products, which can be resolved by taking a few steps:

- Reduce inventory of products with a high Inventory to sales ratio to lower storage costs and optimize resources allocation.
- Evaluate actual demand.
- Ensuring that it has enough inventory to support robust sales is better than having to reduce inventory due to lagging sales, therefore reviewing the company sales and marketing strategy is required to improve sales.
- Balance inventory turnover with last two years sales.
- Reduce the number of warehouses and rental costs optimizing inventory allocation.