

Data Analysis Project on the Mint Classics Company Database using CRISP-DM Methodology with SQL Workbench

Krantikumar Marathe

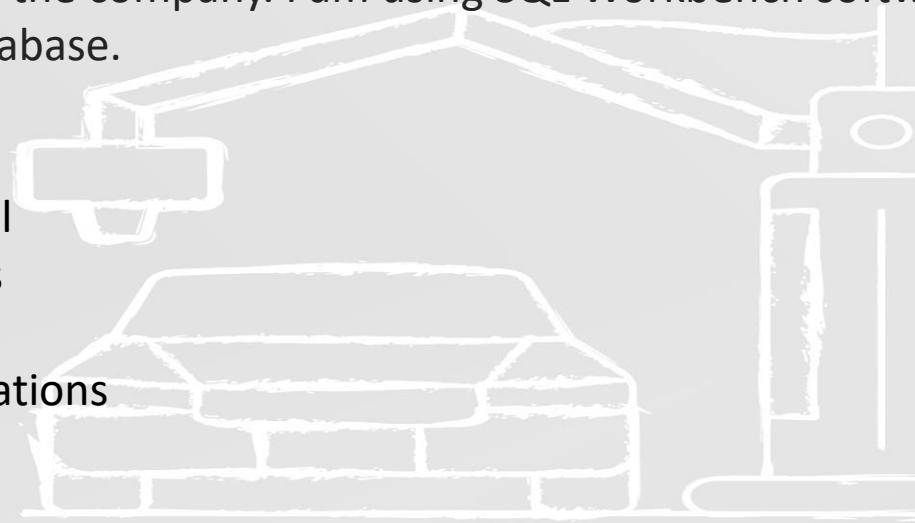


Agenda

Data plays a crucial role in decision-making within a company. With numerous datasets at hand, it's essential to process data in such a way that it becomes valuable. In this project, I will attempt to put into practice how to conduct data analysis from a company's database. The database used is a fictional one from a classic vehicle and car company known as Mint Classics Company. The project's goal is to explore the data to provide insights and address issues within the company. I am using SQL Workbench software to facilitate data analysis from the database.

•Overview:

- Business Goal
- Data Analysis
- Findings
- Recommendations
- Conclusion



Business Goal Summary for Mint Classics Company

Objective:

Mint Classics Company aims to optimize its inventory management and warehouse utilization to enhance operational efficiency and cost-effectiveness. The primary focus is on evaluating the viability of closing one of their storage facilities while maintaining timely service to customers.

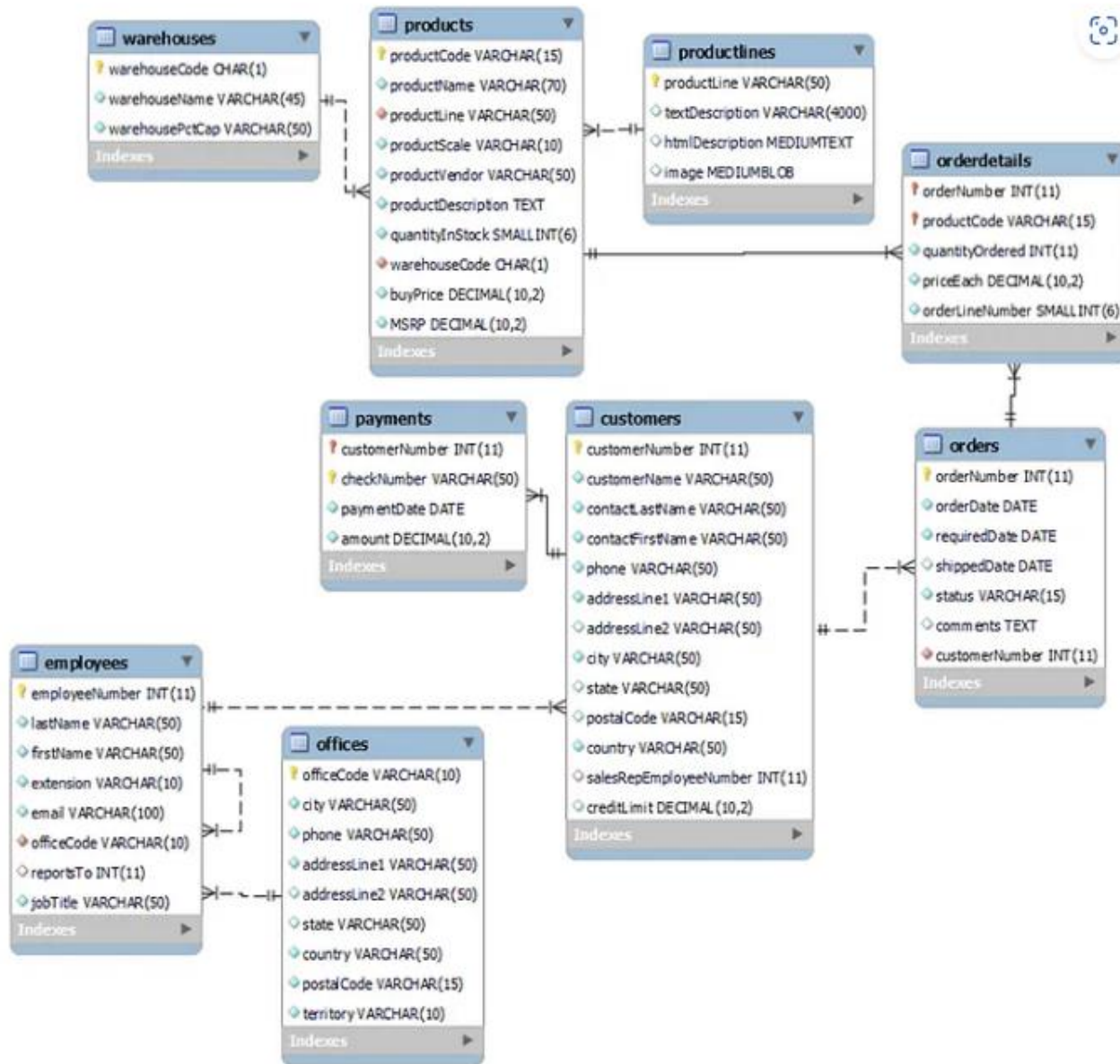
Key Goals:

1. Analyze Sales Patterns:
2. Inventory Optimization:
3. Warehouse Utilization:
4. Customer Service Impact:
5. Cost Reduction:



Final Outcome:

Provide data-driven recommendations to Mint Classics Company on whether to close a storage facility, how to reorganize or reduce inventory, and strategies to maintain or improve customer service levels while reducing operational costs.



Warehouses compile data on the company's warehouses used to store product inventory.

Products compile data about the product types, stock quantities, sales quantities, purchase prices, and more.

Product lines compile data about the descriptions of each product line they sell.

Customers compile data about the company's customer profiles, such as customer names, addresses, credit limits, and others.

Payments compile data about payments made by customers, including payment dates and amounts.

Orders compile data regarding customer orders for specific products.

Order details compile data about orders for specific products, including the quantity of products ordered and the price of each product.

Employees compile data about company employees, including names, addresses, offices, and more.

Offices compile data about company office profiles.

INVENTORY ANALYSIS

1. 15 -- Total quantity stock in each warehouse
16 • SELECT w.warehouseName, SUM(p.quantityInStock) stock FROM products AS p
17 INNER JOIN warehouses AS w ON w.warehouseCode = p.warehouseCode
18 GROUP BY w.warehouseName
19 ORDER BY stock DESC;

Output

	warehouseName	stock
►	East	219183
	North	131688
	West	124880
	South	79380

2. 29 • SELECT productCode, productName, quantityInStock
30 FROM products
31 ORDER BY quantityInStock DESC
32 LIMIT 10; -- Highest inventory

Output

	productCode	productName	quantityInStock
►	S12_2823	2002 Suzuki XREO	9997
	S18_1984	1995 Honda Civic	9772
	S700_2466	America West Airlines B757-200	9653
	S24_3432	2002 Chevy Corvette	9446
	S18_2325	1932 Model A Ford J-Coupe	9354
	S32_2206	1982 Ducati 996 R	9241
	S24_3151	1912 Ford Model T Delivery Wagon	9173
	S18_3482	1976 Ford Gran Torino	9127
	S12_3380	1968 Dodge Charger	9123
	S18_1589	1965 Aston Martin DB5	9042
•	NULL	NULL	NULL

INVENTORY ANALYSIS

1.

```
34 • SELECT productCode, productName, quantityInStock
35 FROM products
36 ORDER BY quantityInStock ASC
37 LIMIT 10; -- Lowest inventory
```

Output

	productCode	productName	quantityInStock
▶	S24_2000	1960 BSA Gold Star DBD34	15
	S12_1099	1968 Ford Mustang	68
	S32_4289	1928 Ford Phaeton Deluxe	136
	S32_1374	1997 BMW F650 ST	178
	S72_3212	Pont Yacht	414
	S18_2248	1911 Ford Town Car	540
	S18_2795	1928 Mercedes-Benz SSK	548
	S700_3167	F/A 18 Hornet 1/72	551
	S50_4713	2002 Yamaha YZR M1	600
	S700_1938	The Mayflower	737
•	NULL	NULL	NULL

3.

```
46 -- Total stock for each product
47 • SELECT productName, quantityInStock FROM products
48 ORDER BY quantityInStock DESC
49 LIMIT 10;
```

Output

	productName	quantityInStock
▶	2002 Suzuki XREO	9997
	1995 Honda Civic	9772
	America West Airlines B757-200	9653
	2002 Chevy Corvette	9446
	1932 Model A Ford J-Coupe	9354
	1982 Ducati 996 R	9241
	1912 Ford Model T Delivery Wagon	9173
	1976 Ford Gran Torino	9127
	1968 Dodge Charger	9123
	1965 Aston Martin DB5	9042

2.

```
40 -- Total stock in each productline
41 • SELECT w.warehouseName, p.productline, SUM(p.quantityInStock) stock FROM products AS p
42 INNER JOIN warehouses AS w ON w.warehouseCode = p.warehouseCode
43 GROUP BY w.warehouseName, p.productline
44 ORDER BY stock DESC;
```

Output

	warehouseName	productline	stock
▶	East	Classic Cars	219183
	West	Vintage Cars	124880
	North	Motorcycles	69401
	North	Planes	62287
	South	Trucks and Buses	35851
	South	Ships	26833
	South	Trains	16696

-PRODUCT SALE ANALYSIS

1. 53 -- Comparative data between Total Stock and Total Ordered

```
54 • SELECT p.productName, p.quantityInStock, SUM(quantityOrdered) ordered, (p.quantityInStock - SUM(quantityOrdered)) CurrentInventory FROM products p
55 LEFT JOIN orderdetails od ON od.productCode = p.productCode
56 GROUP BY p.productName, p.quantityInStock
57 ORDER BY CurrentInventory DESC;
```

Output

	productName	quantityInStock	ordered	CurrentInventory
▶	2002 Suzuki XREO	9997	1028	8969
	1995 Honda Civic	9772	917	8855
	America West Airlines B757-200	9653	984	8669
	2002 Chevy Corvette	9446	894	8552
	1932 Model A Ford J-Coupe	9354	957	8397
	1982 Ducati 996 R	9241	906	8335
	1976 Ford Gran Torino	9127	915	8212
	1968 Dodge Charger	9123	925	8198
	1912 Ford Model T Delivery Wagon	9173	991	8182
	1965 Aston Martin DB5	9042	914	8128
	1948 Porsche Type 356 Roadster	8990	948	8042
	1948 Porsche 356-A Roadster	8826	972	7854

2. 59 -- Sales trends
60 -- get total ordered by time
61 • SELECT o.orderDate, Total_ordered FROM orders o LEFT JOIN

```
62 (SELECT od.orderNumber, SUM(quantityOrdered) Total_ordered FROM orderdetails od
63 GROUP BY od.orderNumber) AS od
64 on od.orderNumber = o.orderNumber
65 ORDER BY orderDate;
```

Avg Order is 35.2190

orderDate	Total_ordered
2004-11-05	329
2004-11-09	223
2004-11-10	253
2004-11-12	495
2004-11-15	532
2004-11-16	158
2004-11-17	437
2004-11-17	621
2004-11-18	261
2004-11-19	217
2004-11-19	117
2004-11-20	447
2004-11-21	285

Output

-PRODUCT SALE ANALYSIS

1.

```
69  -- Identify the most frequently sold items
70 • SELECT productCode, SUM(quantityOrdered) AS totalSold
71 FROM orderdetails
72 GROUP BY productCode
73 ORDER BY totalSold DESC
74 LIMIT 10; -- Top 10 selling products
```

Output

	productCode	totalSold
►	S18_3232	1808
	S18_1342	1111
	S700_4002	1085
	S18_3856	1076
	S50_1341	1074
	S18_4600	1061
	S10_1678	1057
	S12_4473	1056
	S18_2319	1053
	S24_3856	1052

2.

```
77  -- how does the product price affect sales volume
78 • SELECT p.productCode, p.productName, p.buyPrice, SUM(od.quantityOrdered) AS totalSold
79 FROM products p
80 JOIN orderdetails od ON p.productCode = od.productCode
81 GROUP BY p.productCode, p.productName, p.buyPrice
82 ORDER BY p.buyPrice;
```

Output

	productCode	productName	buyPrice	totalSold
►	S24_2840	1958 Chevy Corvette Limited Edition	15.91	983
	S24_2972	1982 Lamborghini Diablo	16.24	912
	S24_2022	1938 Cadillac V-16 Presidential Limousine	20.61	955
	S24_3969	1936 Mercedes Benz 500k Roadster	21.75	824
	S24_1937	1939 Chevrolet Deluxe Coupe	22.57	937
	S18_4668	1939 Cadillac Limousine	23.14	995
	S32_2206	1982 Ducati 996 R	24.14	906
	S18_2625	1936 Harley Davidson El Knucklehead	24.23	945
	S18_1367	1936 Mercedes-Benz 500K Special Roadster	24.26	960
	S18_2432	1926 Ford Fire Engine	24.92	998
	S32_2509	1954 Greyhound Scenicruiser	25.98	955
	S24_3420	1937 Horch 930V Limousine	26.30	884
	S24_3337	1950's Chrysler Superfast Line Coupe	26.75	874

-PRODUCT SALE ANALYSIS

1.

```
84  -- are there any special pattern in thhis sales
85  •  SELECT MONTH(o.orderDate) AS orderMonth, SUM(od.quantityOrdered) AS totalSold
86  FROM orders o
87  JOIN orderdetails od ON o.orderNumber = od.orderNumber
88  GROUP BY orderMonth
89  ORDER BY orderMonth;
```

Output

	orderMonth	totalSold
▶	1	7997
	2	7959
	3	8294
	4	7906
	5	9464
	6	5433
	7	5721
	8	6538
	9	5681
	10	11214
	11	21540
	12	7769

2.

Output




COMPANY REVENUE

1. 93 -- Get The Total revenue by each warehouse

```
94 • SELECT warehouseName AS warehouse,  
95        SUM(quantityOrdered) Total_ordered,  
96        SUM(od.quantityOrdered * od.priceEach) AS total_revenue  
97 FROM warehouses w  
98 INNER JOIN products as p on p.warehouseCode = w.warehouseCode  
99 INNER JOIN orderdetails as od on od.productCode = p.productCode  
100 GROUP BY warehouse  
101 ORDER BY total_revenue DESC;
```

Output

Result Grid  Filter Rows: <input type="text"/>			
	warehouse	Total_ordered	total_revenue
▶	East	35582	3853922.49
	North	24650	2076063.66
	South	22351	1876644.83
	West	22933	1797559.63

2.

```
104 -- revenue from each product line  
105 • SELECT productLine,  
106        SUM(quantityOrdered) Total_ordered,  
107        SUM(od.quantityOrdered * od.priceEach) AS total_revenue  
108 FROM products p  
109 INNER JOIN orderdetails od ON od.productCode = p.productCode  
110 GROUP BY productLine  
111 ORDER BY total_revenue DESC;
```

Output

	productLine	Total_ordered	total_revenue
▶	Classic Cars	35582	3853922.49
	Vintage Cars	22933	1797559.63
	Motorcycles	12778	1121426.12
	Trucks and Buses	11001	1024113.57
	Planes	11872	954637.54
	Ships	8532	663998.34
	Trains	2818	188532.92

COMPANY REVENUE

```
1. 113 -- revenue from each product -- Using each price
114 • SELECT distinct productName, quantityInStock, buyPrice, priceEach,
115      SUM(quantityOrdered) Total_ordered,
116      SUM(od.quantityOrdered * od.priceEach) AS total_revenue
117 FROM products p
118 INNER JOIN orderdetails od ON p.productCode = od.productCode
119 GROUP BY productName, quantityInStock, buyPrice, priceEach
120 ORDER BY total_revenue desc;
```

Output

	productName	quantityInStock	buyPrice	priceEach	Total_ordered	total_revenue
▶	1917 Grand Touring Sedan	2724	86.70	153.00	195	29835.00
	1968 Ford Mustang	68	95.34	184.84	159	29389.56
	1969 Corvair Monza	6906	89.14	145.04	200	29008.00
	2001 Ferrari Enzo	3619	95.59	176.63	158	27907.54
	1952 Alpine Renault 1300	7305	98.58	205.73	130	26744.90
	2001 Ferrari Enzo	3619	95.59	193.25	130	25122.50
	1969 Ford Falcon	1049	83.05	169.56	146	24755.76
	1992 Ferrari 360 Spider red	8347	77.90	137.17	175	24004.75
	1957 Corvette Convertible	1249	69.93	120.53	198	23864.94
	ATA: B757-300	7106	59.33	98.48	237	23339.76
	1992 Ferrari 360 Spider red	8347	77.90	169.34	136	23030.24
	1980s Black Hawk Helicopter	5330	77.27	137.19	167	22910.73
	2003 Harley-Davidson Eagl...	5582	91.02	189.79	120	22774.80

```
2. 122 -- without each price only products and total revenue by products
123 • SELECT distinct productName,
124      SUM(quantityOrdered) Total_ordered,
125      SUM(od.quantityOrdered * od.priceEach) AS total_revenue
126 FROM products p
127 INNER JOIN orderdetails od ON p.productCode = od.productCode
128 GROUP BY productName
129 ORDER BY total_revenue desc;
```

Output

	productName	Total_ordered	total_revenue
▶	1992 Ferrari 360 Spider red	1808	276839.98
	2001 Ferrari Enzo	1019	190755.86
	1952 Alpine Renault 1300	961	190017.96
	2003 Harley-Davidson Eagle Drag Bike	985	170686.00
	1968 Ford Mustang	933	161531.48
	1969 Ford Falcon	965	152543.02
	1980s Black Hawk Helicopter	1040	144959.91
	1998 Chrysler Plymouth Prowler	986	142530.63
	1917 Grand Touring Sedan	918	140535.60
	2002 Suzuki XREO	1028	135767.03

CUSTOMER ANALYSIS

1.

```

133  -- GET Customer Profile Data including orders and payments
134  •  SELECT c.customerNumber, c.customerName, c.country, c.creditLimit, Total_order,total_payment, (total_payment- creditLimit) as creditLimitdiff
135         FROM (SELECT customerNumber, customerName, country, creditLimit FROM customers) c
136  LEFT JOIN
137      (SELECT customerNumber, SUM(amount) as total_payment
138       FROM payments
139       GROUP BY customerNumber) p
140  on c.customerNumber = p.customerNumber
141  LEFT JOIN
142      (SELECT customerNumber,Count(orderNumber) Total_order FROM orders
143       GROUP BY customerNumber) o
144  on c.customerNumber = o.customerNumber
145  GROUP BY customerName, c.customerNumber, c.country, c.creditLimit, Total_order,total_payment
146  ORDER BY total_payment DESC;

```

Output

	customerNumber	customerName	country	creditLimit	Total_order	total_payment	creditLimitdiff
▶	141	Euro+ Shopping Channel	Spain	227600.00	26	715738.98	488138.98
	124	Mini Gifts Distributors Ltd.	USA	210500.00	17	584188.24	373688.24
	114	Australian Collectors, Co.	Australia	117300.00	5	180585.07	63285.07
	151	Muscle Machine Inc	USA	138500.00	4	177913.95	39413.95
	148	Dragon Souveniers, Ltd.	Singapore	103800.00	5	156251.03	52451.03
	323	Down Under Souveniers, Inc	New Zealand	88000.00	5	154622.08	66622.08
	187	AV Stores, Co.	UK	136800.00	3	148410.09	11610.09
	276	Anna's Decorations, Ltd	Australia	107800.00	4	137034.22	29234.22
	321	Corporate Gift Ideas Co.	USA	105000.00	4	132340.78	27340.78
	146	Saveley & Henriot, Co.	France	123900.00	3	130305.35	6405.35
	278	Rovelli Gifts	Italy	119600.00	3	127529.69	7929.69
	353	Reims Collectables	France	81100.00	5	126983.19	45883.19
	119	La Rochelle Gifts	France	118200.00	4	116949.68	-1250.32
	363	Online Diecast Creations Co.	USA	114200.00	3	116449.29	2249.29
	496	Kelly's Gift Shop	New Zealand	110000.00	4	114497.19	4497.19
	458	Corrida Auto Replicas, Ltd	Spain	104600.00	3	112440.09	7840.09


CUSTOMER ANALYSIS


1.

```
150 -- Get data on the number of employees in each office
151 • SELECT COUNT(employeeNumber) cnt_of_employee, o.officeCode, o.city, o.country
152 FROM offices o
153 LEFT JOIN employees e on e.officeCode = o.officeCode
154 GROUP BY officeCode
155 ORDER BY cnt_of_employee DESC;
```

Output

Result Grid





Filter Rows:





Export:

	cnt_of_employee	officeCode	city	country
▶	6	1	San Francisco	USA
	5	4	Paris	France
	4	6	Sydney	Australia
	2	2	Boston	USA
	2	3	NYC	USA
	2	5	Tokyo	Japan
	2	7	London	UK

2.

```
157 -- Get employee performance Data
158 • SELECT employeeNumber, firstName, lastName, jobTitle,
159 COUNT(orderNumber) count_of_orders
160 FROM employees e
161 LEFT JOIN customers c on c.salesRepEmployeeNumber = e.employeeNumber
162 LEFT JOIN orders o on o.customerNumber = c.customerNumber
163 GROUP BY employeeNumber, firstName, lastName, jobTitle
164 ORDER BY count_of_orders DESC;
```

Output

Result Grid		  Filter Rows:	Export:   Wrap Cell		
	employeeNumber	firstName	lastName	jobTitle	count_of_orders
▶	1370	Gerard	Hernandez	Sales Rep	43
	1165	Leslie	Jennings	Sales Rep	34
	1401	Pamela	Castillo	Sales Rep	31
	1504	Barry	Jones	Sales Rep	25
	1323	George	Vanauf	Sales Rep	22
	1501	Larry	Bott	Sales Rep	22
	1337	Loui	Bondur	Sales Rep	20
	1611	Andy	Fixter	Sales Rep	19
	1612	Peter	Marsh	Sales Rep	19
	1216	Steve	Patterson	Sales Rep	18
	1286	Foon Yue	Tseng	Sales Rep	17
	1621	Mami	Nishi	Sales Rep	16
	1166	Leslie	Thompson	Sales Rep	14
	1188	Julie	Firrelli	Sales Rep	14
	1702	Martin	Gerard	Sales Rep	12
	1002	Diane	Murphy	President	0

CONDUCT WHAT-IF ANALYSIS

1. 133 -- What-if analysis for reducing inventory by 5%

134 • SELECT productCode, productName, quantityInStock, quantityInStock * 0.95 AS reducedStock

135 FROM products; -- To understand the impact of reducing inventory by 5%, we need to calculate the new inventory levels.

136

Output



	productCode	productName	quantityInStock	reducedStock
▶	S10_1678	1969 Harley Davidson Ultimate Chopper	7933	7536.35
	S10_1949	1952 Alpine Renault 1300	7305	6939.75
	S10_2016	1996 Moto Guzzi 1100i	6625	6293.75
	S10_4698	2003 Harley-Davidson Eagle Drag Bike	5582	5302.90
	S10_4757	1972 Alfa Romeo GTA	3252	3089.40
	S10_4962	1962 LanciaA Delta 16V	6791	6451.45
	S12_1099	1968 Ford Mustang	68	64.60
	S12_1108	2001 Ferrari Enzo	3619	3438.05
	S12_1666	1958 Setra Bus	1579	1500.05
	S12_2823	2002 Suzuki XREO	9997	9497.15
	S12_3148	1969 Corvair Monza	6906	6560.70

FORMULATE SUGGESTIONS AND RECOMMENDATIONS

1.

```
139 -- A. Scenario 1: Close the Least Utilized Warehouse
140 -- Identify the least utilized warehouse
141 • SELECT p.warehouseCode, warehouseName, SUM(quantityInStock) AS totalInventory
142 FROM products p
143 join warehouses w ON w.warehouseCode = p.warehouseCode
144 GROUP BY warehouseCode
145 ORDER BY totalInventory ASC; -- South
```

Output

Result Grid			 Filter Rows:		Export
	warehouseCode	warehouseName	totalInventory		
▶	d	South	79380		
	c	West	124880		
	a	North	131688		
	b	East	219183		

	warehouseCode	warehouseName	totalInventory
▶	d	South	79380

2.

```
147 -- B. Scenario 2: Optimize Inventory Levels
148 -- Identify slow-moving products
149 • SELECT p.productCode, productName, quantityInStock, SUM(od.quantityOrdered) AS totalSold
150 FROM products p
151 JOIN orderdetails od ON p.productCode = od.productCode
152 GROUP BY p.productCode, p.productName, p.quantityInStock
153 HAVING totalSold <
154 (SELECT AVG(quantityOrdered) FROM orderdetails)
155 ORDER BY totalSold ASC;
```

Output

productCode	productName	quantityInStock	totalSold

There is no Slow Moving Products all products Giving their best

FORMULATE SUGGESTIONS AND RECOMMENDATIONS

1.

```
165  -- What-if analysis for reducing inventory of slow-moving products by 5%
166  •  SELECT productCode, productName, quantityInStock, quantityInStock * 0.95 AS reducedStock
167      FROM products
168  WHERE productCode IN (
169      SELECT productCode
170      FROM (SELECT productCode, SUM(quantityOrdered) AS totalSold
171            FROM orderdetails
172            GROUP BY productCode
173            HAVING totalSold < (SELECT AVG(quantityOrdered) FROM orderdetails)) AS slow_moving
174  );
```

Output

Result Grid	Filter Rows:	Export:	W
productCode	productName	quantityInStock	reducedStock

Recommendations

•Closing the Least Utilized Warehouse:

- Redistribute inventory to other warehouses.
- Analyze fulfillment and delivery times before and after closure.

•Reducing Inventory of Slow-Moving Products:

- Implement a 5% reduction in inventory levels for slow-moving products.
- Present the potential reduction in storage needs.

In deployment step, the task is to present the findings. Therefore, I will attempt to recommend some policies that the company may consider, including:

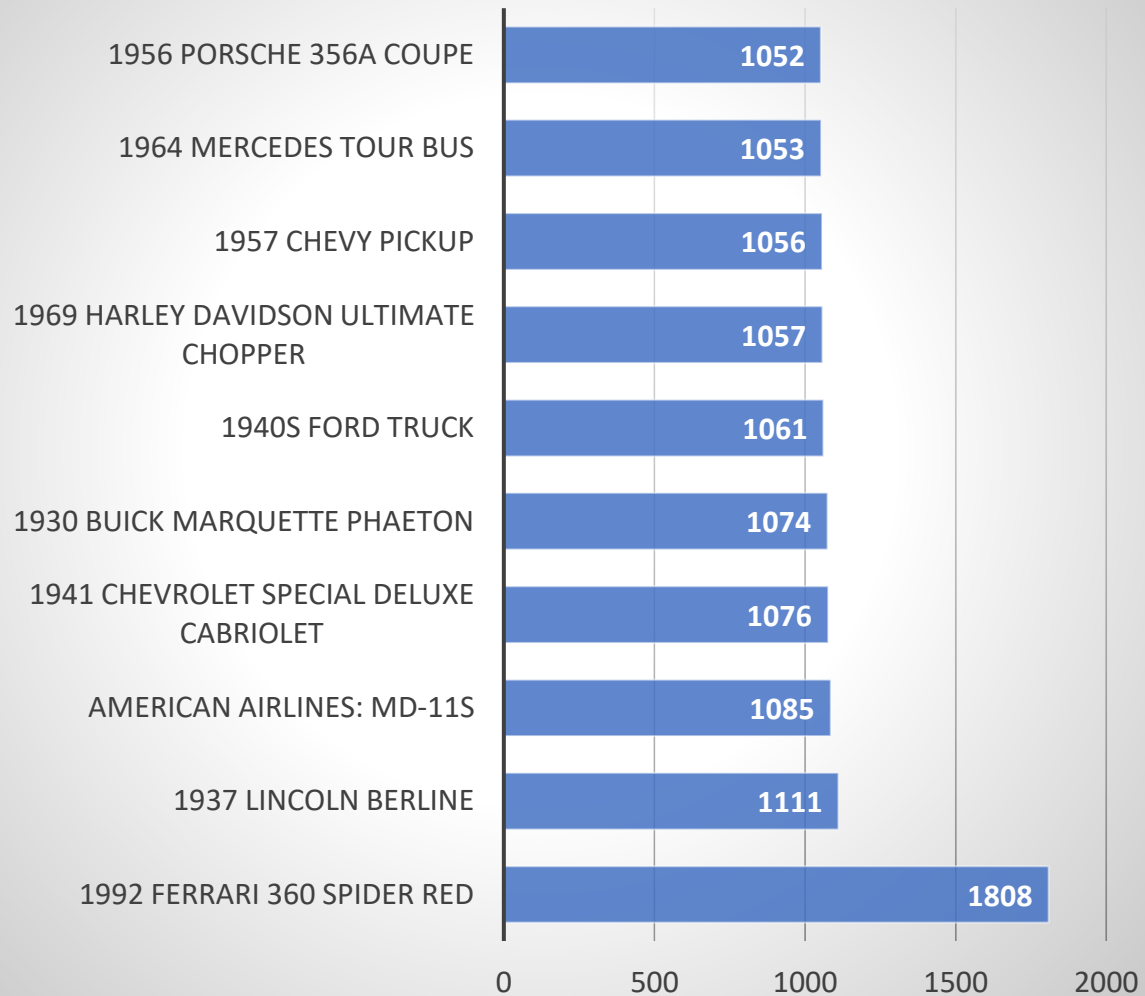
- Evaluating which products should be added and which should be reduced
- Promoting in the countries with the highest order rates
- Adjusting product prices so that the products can be sold
- Implementing policies for customers with payment issues
- Recognizing employees with good performance and providing training for those with lower performance

In evaluation step

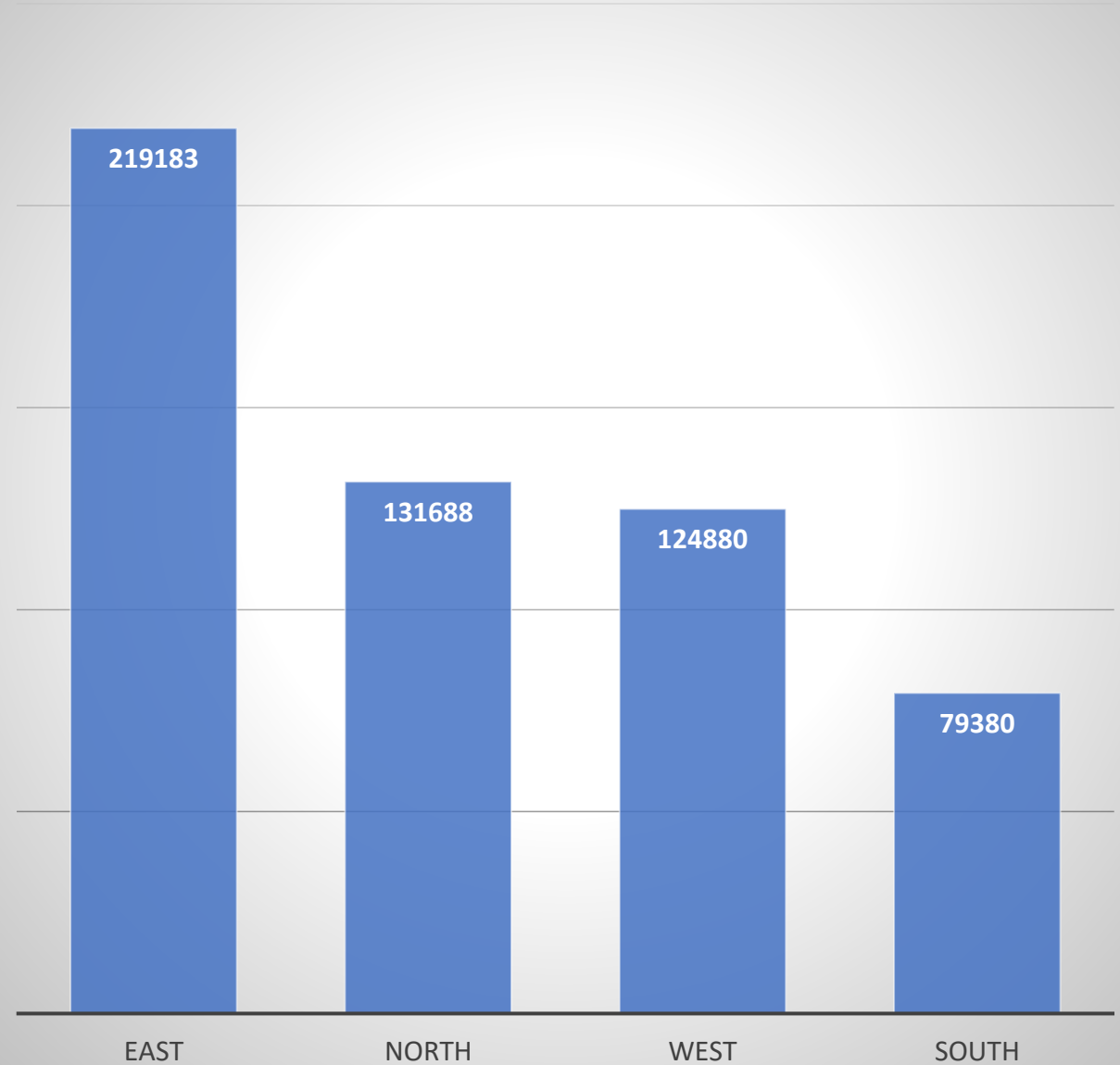
- There are warehouses with a significant product stock but generate relatively low revenue.
- Some products have a high stock quantity but low sales or vice versa.
- Certain product lines have substantial stock quantities but yield low revenue.
- There are customers with good payments, but there are also customers with payment issues.
- Some employees exhibit high sales performance, while others have lower sales performance.

VISUALIZATION

totalSold



quantityInstock



VISUALIZATION

Price V TotalSold

