Abbreviations and custom metrics

metrics	Full form	Formula	Increased indicates	Decreased indicates
QP	Profit per Quantity	Profit/Sells	Profitable product	Less Profitable Product
SO	Orders Per Stock	Orders/stock	Low stock compared to demand	High stock compared to Demand

Recommendation

Can we afford to close one warehouse?

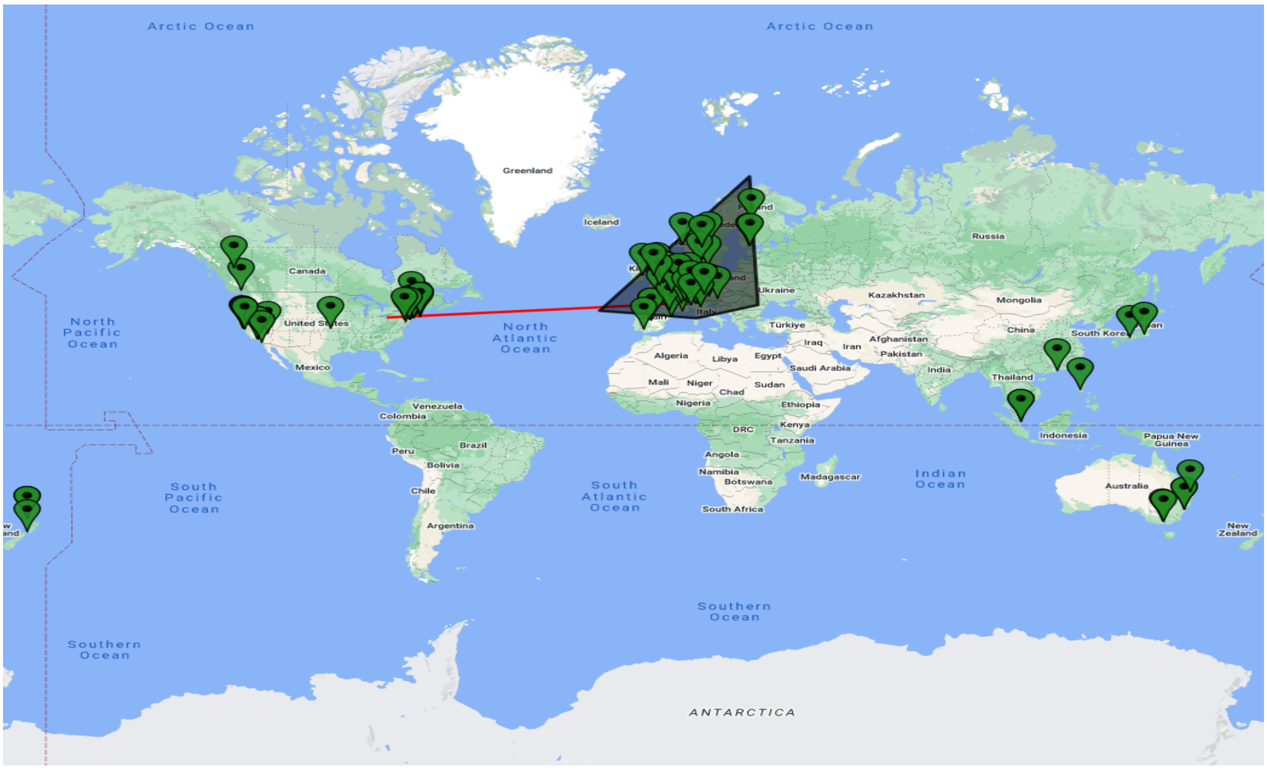
- ***General summary***

Yes , you can , i have done exploratory data analysis on given dataset and here are my main finding:

1. We are heavily overstocked Overstock Products
2. We have 1 non moving asset (Toyota supra , 7.6k units , 0 sold)
3. We also have lot of slow moving assets
   TOP 20 least profitable products

Which warehouse to close?

Based on my analysis we can close “a” and “d” warehouse. Given that B and C has most storage closing them would be harmful in heavy demand. Also dataset has no information regarding warehouse location other than their name east , west, noth , south. I tried to locate warehouse based on customer orders and delivery time but given that dataset does not contain information regarding mode of transportation, i was not able to locate them. So here is few information regarding warehouses which will help you in making decision.



- Warehouse Near France & Germany should not be close, if warehouse is in USA east then it is best location given that we can delivery product fast to high demand country
- Warehouse near Ocean (which i suspect is d) can be closed if other warehouse is also located near Ocean, due to inventory of D is filled with planes , ships & trains
- Delivery time to Japan will always be 4-7 days so trying to reduce that amount is unnecessary as it will impact other parts of bussiness

Here is list of ProductLine of each warehouse

Warehouse	ProductLine	Count
a	motorcycles	13
	Planes	12
b	Classic Cars	38
c	Vintage Cars	24
d	Trucks and buses	11
	Ships	9
	Trains	3

```
USE mnt2;
SELECT  p.productLine, COUNT(p.productCode) AS counts
FROM products AS p
WHERE p.warehouseCode = "d"
GROUP BY p.productLine
```

Products in Each Warehouse: **Mint Classic**

How to close warehouse?
I have adjusted Products in Stock , increased understocked products and decreased overstocked products setting SO near 0.8 for understock and 0.4 for overstock products. Here are steps

Introduction

As a data analyst, you have been asked to use MySQL Workbench to familiarize yourself with the general business by examining the current data. You will be provided with a data model and sample data tables to review. You will then need to isolate and identify those parts of the data that could be useful in deciding how to reduce inventory. You will write queries to answer questions like these:

1. Where are items stored and if they were rearranged, could a warehouse be eliminated?
2. How are inventory numbers related to sales figures? Do the inventory counts seem appropriate for each item?
3. Are we storing items that are not moving? Are any items candidates for being dropped from the product line?

The answers to questions like those should help you to formulate suggestions and recommendations for reducing inventory with the goal of closing one of the storage facilities.

Project Objectives

1. Explore products currently in inventory.
2. Determine important factors that may influence inventory reorganization/reduction.
3. Provide analytic insights and data-driven recommendations.

Your Challenge

Your challenge will be to conduct an exploratory data analysis to investigate if there are any patterns or themes that may influence the reduction or reorganization of inventory in the Mint Classics storage facilities. To do this, you will import the database and then analyze data. You will also pose questions, and seek to answer them meaningfully using SQL queries to retrieve data from the database provided.

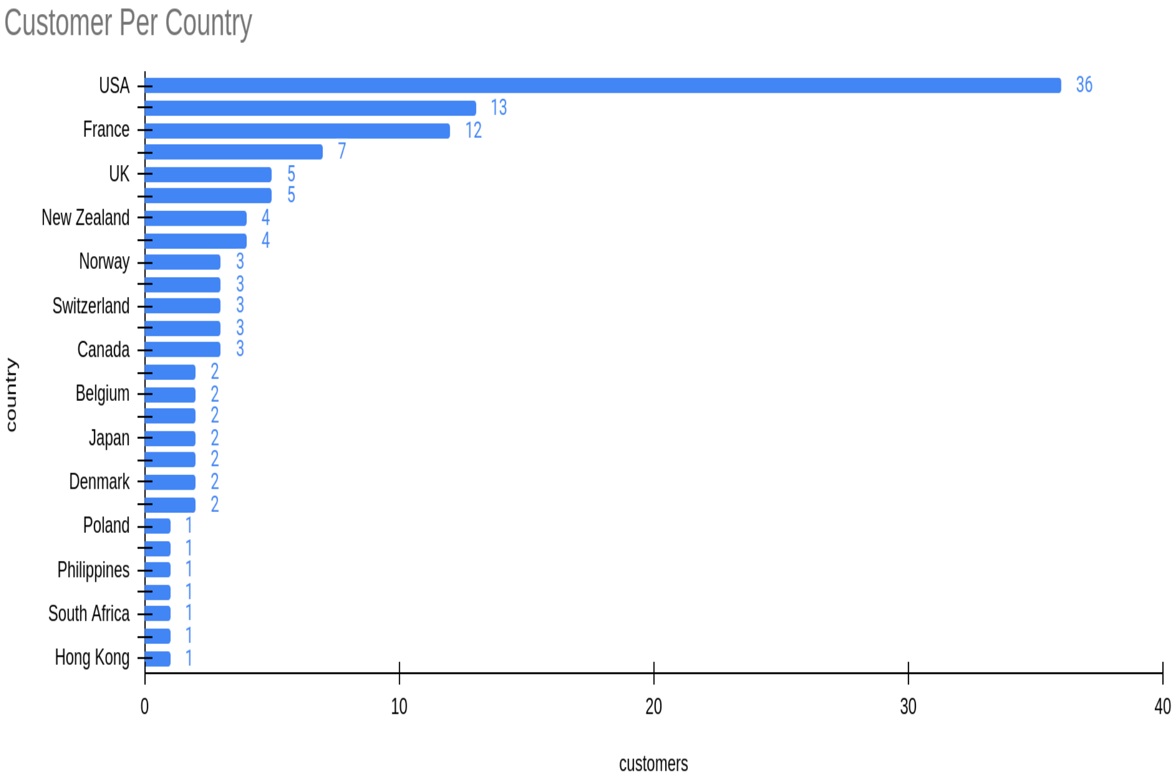
In this project, we'll use the fictional Mint Classics relational database and a relational data model. Both will be provided.

After you perform your analysis, you will share your findings

Customer Data Analysis

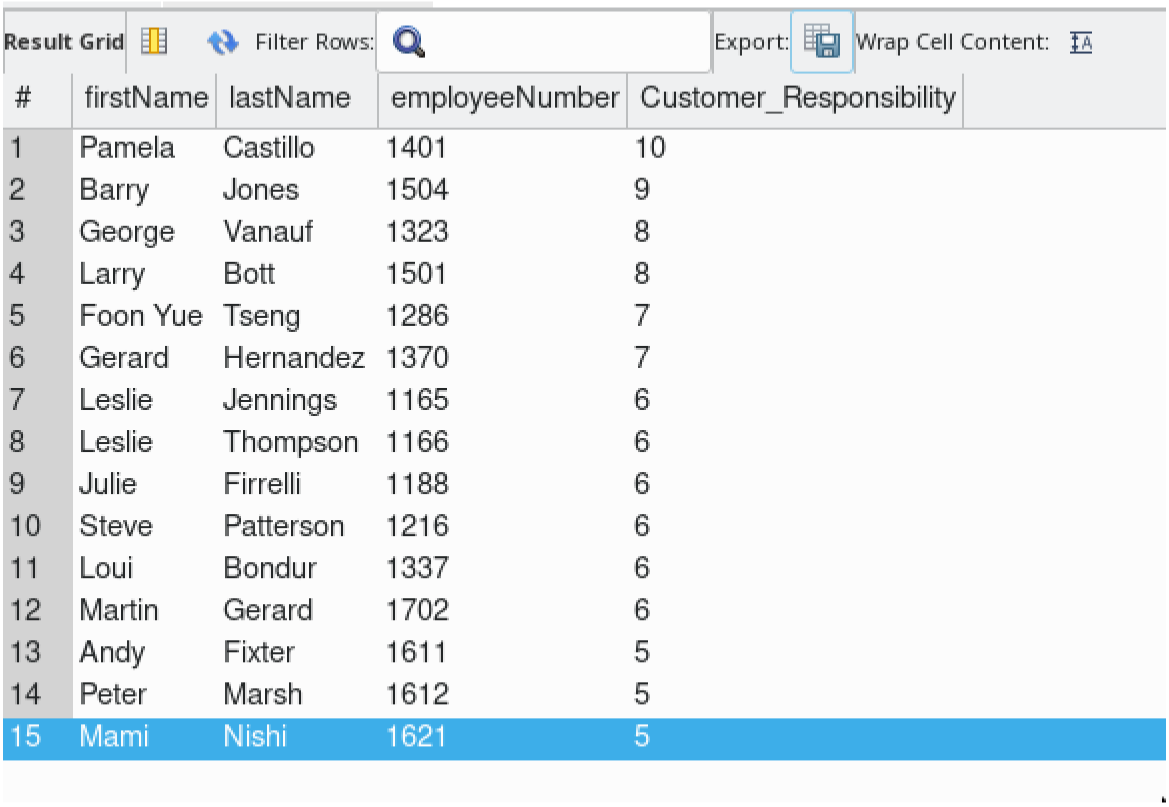
1. Customer by country

```sql
SELECT country, COUNT(customerNumber) AS customers
FROM mintclassics.customers
GROUP BY country
ORDER BY COUNT(customerName) DESC
```



2. Checking Sales Representatives

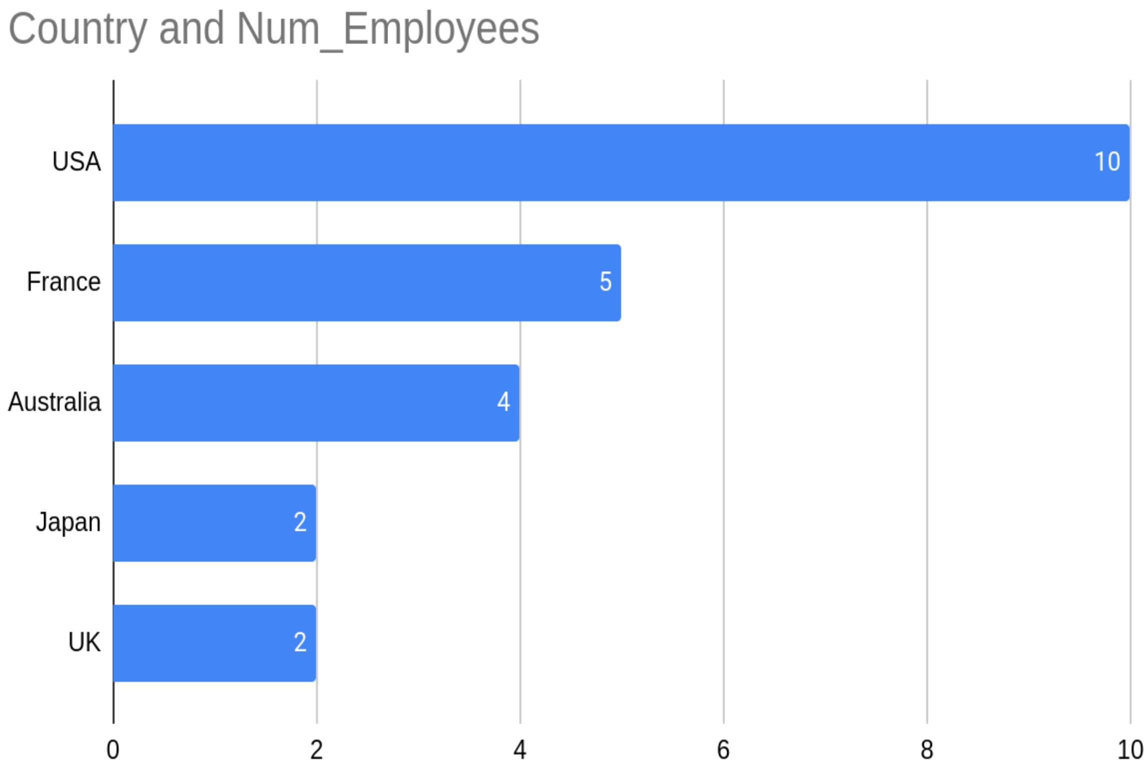
```
USE mintclassics;

SELECT firstName, lastName , employeeNumber ,
COUNT(C.salesRepEmployeeNumber) AS Customer_Responsibility
FROM employees
JOIN customers as C on employees.employeeNumber =
C.salesRepEmployeeNumber
GROUP BY employeeNumber
ORDER BY COUNT(C.salesRepEmployeeNumber) DESC;
```



#	firstName	lastName	employeeNumber	Customer_Responsibility
1	Pamela	Castillo	1401	10
2	Barry	Jones	1504	9
3	George	Vanauf	1323	8
4	Larry	Bott	1501	8
5	Foon Yue	Tseng	1286	7
6	Gerard	Hernandez	1370	7
7	Leslie	Jennings	1165	6
8	Leslie	Thompson	1166	6
9	Julie	Firrelli	1188	6
10	Steve	Patterson	1216	6
11	Loui	Bondur	1337	6
12	Martin	Gerard	1702	6
13	Andy	Fixter	1611	5
14	Peter	Marsh	1612	5
15	Mami	Nishi	1621	5

3. Employees Across Globe (country)

```
USE mintclassics;

SELECT ofc.country ,city, ofc.officeCode , COUNT(E.employeeNumber)
AS Num_Employees
FROM offices as ofc
JOIN employees as E on ofc.officeCode = E.officeCode
GROUP BY ofc.officeCode
ORDER BY COUNT(E.employeeNumber) DESC , country
```



Product Summary

Abbreviations

QP = Quantity to Profit Ration = Profit/Quantity , more QP more profitable product / Quantity

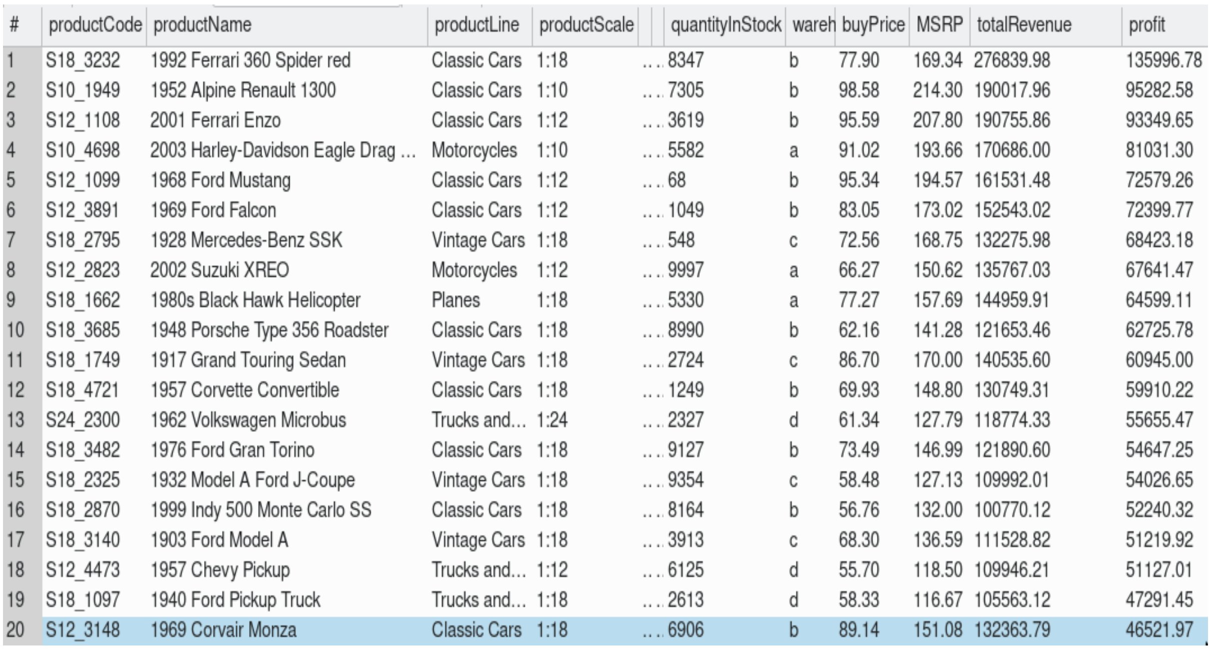
Summary Tables

Attribute	Value
Max Profit	5554.56
Average Profit	1276.996078
Min Profit	79.6
Average QP	36.19336115

Attribute	Product	QStock	QP	SO	Profit%
Most Profitable	1992 Ferrari 360 Spider red	8347	75.21	0.2166	3.55
Most Profitable (QP)	1952 Alpine Renault 1300	7305	99.15	0.1316	2.5
Most Sold	1992 Ferrari 360 Spider red	8347	75.21	0.2166	3.55
Least Profitable	1985 Toyota Supra	7733	0	0	0
Least Profitable (QP)	1985 Toyota Supra	7733	0	0	0
Least Sold	1985 Toyota Supra	7733	0	0	0

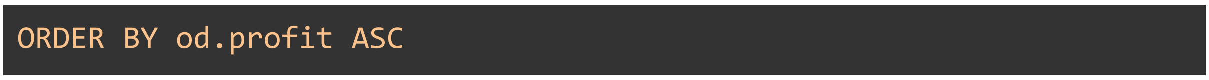
Top 20 most profitable products

```
USE mintclassics;
SELECT
    p.*,
    COALESCE(od.totalRevenue, 0) AS totalRevenue,
    COALESCE(od.profit, 0) AS profit
FROM
    products p
LEFT JOIN (
    SELECT
        odd.productCode,
        SUM(odd.priceEach * odd.quantityOrdered) AS totalRevenue,
        SUM((odd.priceEach - pp.buyPrice) * odd.quantityOrdered)
AS profit
    FROM
        orderdetails odd
    JOIN
        products pp ON odd.productCode = pp.productCode
    GROUP BY
        odd.productCode
) od ON p.productCode = od.productCode
ORDER BY
    od.profit DESC;
```

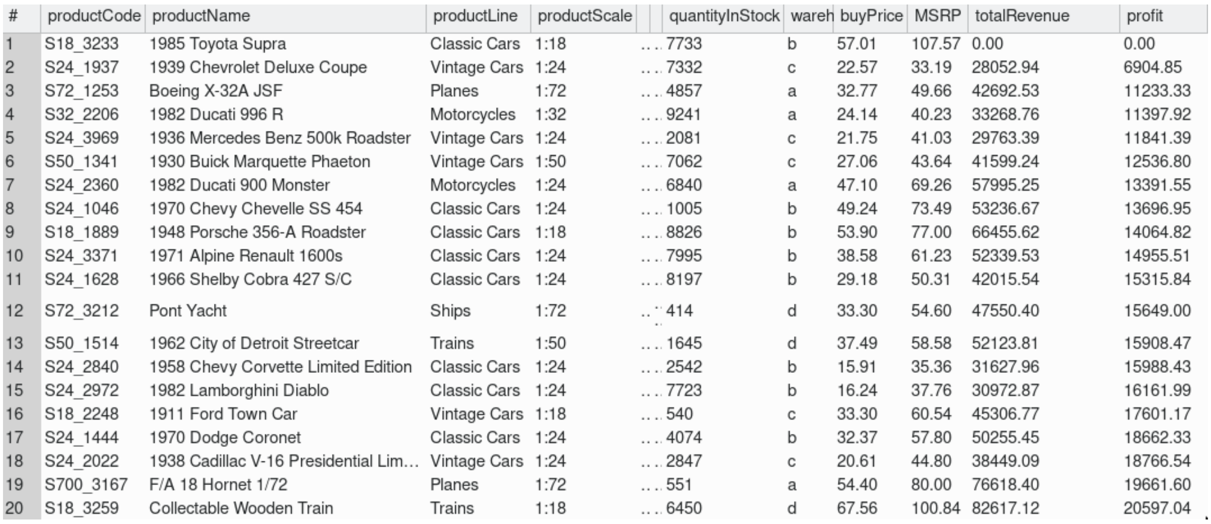


#	productCode	productName	productLine	productScale		quantityInStock	wareh	buyPrice	MSRP	totalRevenue	profit
1	S18_3232	1992 Ferrari 360 Spider red	Classic Cars	1:18	...	8347	b	77.90	169.34	276839.98	135996.78
2	S10_1949	1952 Alpine Renault 1300	Classic Cars	1:10	...	7305	b	98.58	214.30	190017.96	95282.58
3	S12_1108	2001 Ferrari Enzo	Classic Cars	1:12	...	3619	b	95.59	207.80	190755.86	93349.65
4	S10_4698	2003 Harley-Davidson Eagle Drag ...	Motorcycles	1:10	...	5582	a	91.02	193.66	170686.00	81031.30
5	S12_1099	1968 Ford Mustang	Classic Cars	1:12	...	68	b	95.34	194.57	161531.48	72579.26
6	S12_3891	1969 Ford Falcon	Classic Cars	1:12	...	1049	b	83.05	173.02	152543.02	72399.77
7	S18_2795	1928 Mercedes-Benz SSK	Vintage Cars	1:18	...	548	c	72.56	168.75	132275.98	68423.18
8	S12_2823	2002 Suzuki XREO	Motorcycles	1:12	...	9997	a	66.27	150.62	135767.03	67641.47
9	S18_1662	1980s Black Hawk Helicopter	Planes	1:18	...	5330	a	77.27	157.69	144959.91	64599.11
10	S18_3685	1948 Porsche Type 356 Roadster	Classic Cars	1:18	...	8990	b	62.16	141.28	121653.46	62725.78
11	S18_1749	1917 Grand Touring Sedan	Vintage Cars	1:18	...	2724	c	86.70	170.00	140535.60	60945.00
12	S18_4721	1957 Corvette Convertible	Classic Cars	1:18	...	1249	b	69.93	148.80	130749.31	59910.22
13	S24_2300	1962 Volkswagen Microbus	Trucks and...	1:24	...	2327	d	61.34	127.79	118774.33	55655.47
14	S18_3482	1976 Ford Gran Torino	Classic Cars	1:18	...	9127	b	73.49	146.99	121890.60	54647.25
15	S18_2325	1932 Model A Ford J-Coupe	Vintage Cars	1:18	...	9354	c	58.48	127.13	109992.01	54026.65
16	S18_2870	1999 Indy 500 Monte Carlo SS	Classic Cars	1:18	...	8164	b	56.76	132.00	100770.12	52240.32
17	S18_3140	1903 Ford Model A	Vintage Cars	1:18	...	3913	c	68.30	136.59	111528.82	51219.92
18	S12_4473	1957 Chevy Pickup	Trucks and...	1:12	...	6125	d	55.70	118.50	109946.21	51127.01
19	S18_1097	1940 Ford Pickup Truck	Trucks and...	1:18	...	2613	d	58.33	116.67	105563.12	47291.45
20	S12_3148	1969 Corvair Monza	Classic Cars	1:18	...	6906	b	89.14	151.08	132363.79	46521.97

TOP 20 least profitable products



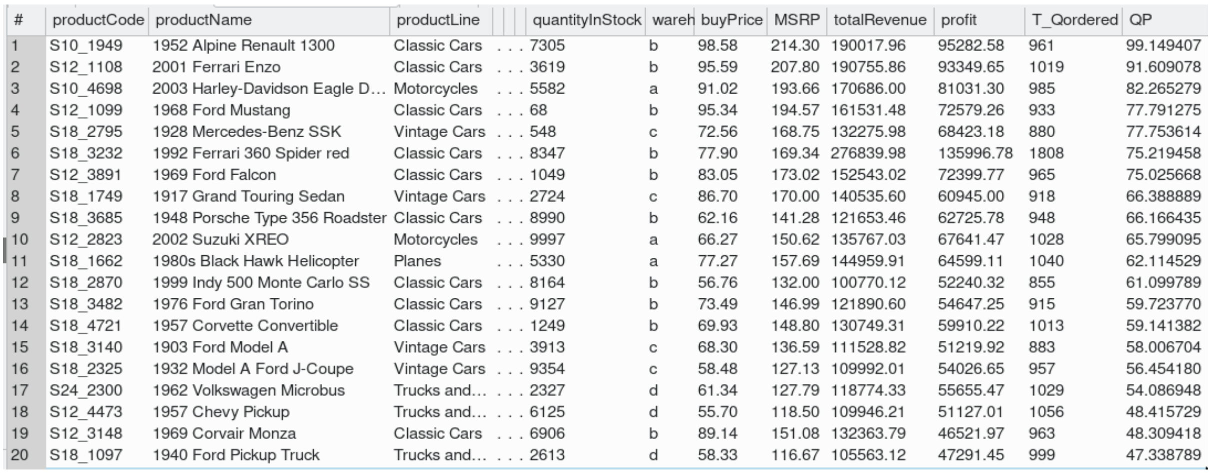
```
ORDER BY od.profit ASC
```



#	productCode	productName	productLine	productScale			quantityInStock	wareh	buyPrice	MSRP	totalRevenue	profit
1	S18_3233	1985 Toyota Supra	Classic Cars	1:18	…	…	7733	b	57.01	107.57	0.00	0.00
2	S24_1937	1939 Chevrolet Deluxe Coupe	Vintage Cars	1:24	…	…	7332	c	22.57	33.19	28052.94	6904.85
3	S72_1253	Boeing X-32A JSF	Planes	1:72	…	…	4857	a	32.77	49.66	42692.53	11233.33
4	S32_2206	1982 Ducati 996 R	Motorcycles	1:32	…	…	9241	a	24.14	40.23	33268.76	11397.92
5	S24_3969	1936 Mercedes Benz 500k Roadster	Vintage Cars	1:24	…	…	2081	c	21.75	41.03	29763.39	11841.39
6	S50_1341	1930 Buick Marquette Phaeton	Vintage Cars	1:50	…	…	7062	c	27.06	43.64	41599.24	12536.80
7	S24_2360	1982 Ducati 900 Monster	Motorcycles	1:24	…	…	6840	a	47.10	69.26	57995.25	13391.55
8	S24_1046	1970 Chevy Chevelle SS 454	Classic Cars	1:24	…	…	1005	b	49.24	73.49	53236.67	13696.95
9	S18_1889	1948 Porsche 356-A Roadster	Classic Cars	1:18	…	…	8826	b	53.90	77.00	66455.62	14064.82
10	S24_3371	1971 Alpine Renault 1600s	Classic Cars	1:24	…	…	7995	b	38.58	61.23	52339.53	14955.51
11	S24_1628	1966 Shelby Cobra 427 S/C	Classic Cars	1:24	…	…	8197	b	29.18	50.31	42015.54	15315.84
12	S72_3212	Pont Yacht	Ships	1:72	…	…	414	d	33.30	54.60	47550.40	15649.00
13	S50_1514	1962 City of Detroit Streetcar	Trains	1:50	…	…	1645	d	37.49	58.58	52123.81	15908.47
14	S24_2840	1958 Chevy Corvette Limited Edition	Classic Cars	1:24	…	…	2542	b	15.91	35.36	31627.96	15988.43
15	S24_2972	1982 Lamborghini Diablo	Classic Cars	1:24	…	…	7723	b	16.24	37.76	30972.87	16161.99
16	S18_2248	1911 Ford Town Car	Vintage Cars	1:18	…	…	540	c	33.30	60.54	45306.77	17601.17
17	S24_1444	1970 Dodge Coronet	Classic Cars	1:24	…	…	4074	b	32.37	57.80	50255.45	18662.33
18	S24_2022	1938 Cadillac V-16 Presidential Lim…	Vintage Cars	1:24	…	…	2847	c	20.61	44.80	38449.09	18766.54
19	S700_3167	F/A 18 Hornet 1/72	Planes	1:72	…	…	551	a	54.40	80.00	76618.40	19661.60
20	S18_3259	Collectable Wooden Train	Trains	1:18	…	…	6450	d	67.56	100.84	82617.12	20597.04

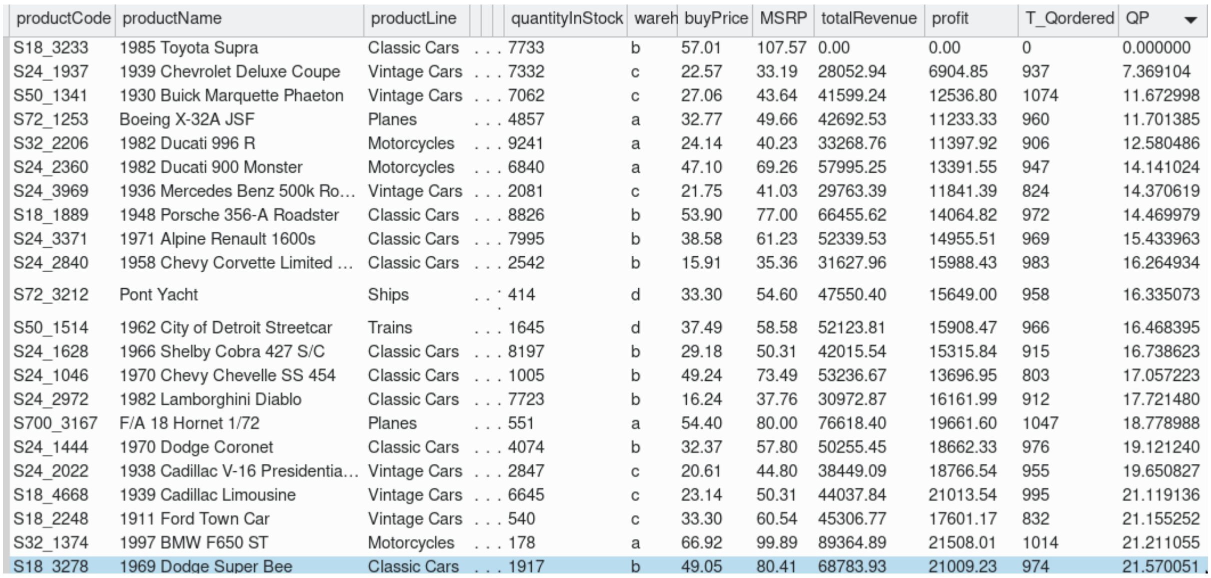
Top 20 Products Based on Quantity to profit ration(Profit/ Quantity)

```
SELECT
    p.*,
    COALESCE(od.totalRevenue, 0) AS totalRevenue,
    COALESCE(od.profit, 0) AS profit,
    COALESCE(od.totalQuantityOrdered, 0) AS totalQuantityOrdered,
    CASE
        WHEN od.profit > 0 THEN  od.profit /
od.totalQuantityOrdered
        ELSE 0
    END AS quantityToProfitRatio
FROM
    products p
LEFT JOIN (
    SELECT
        odd.productCode,
        SUM(odd.priceEach * odd.quantityOrdered) AS totalRevenue,
        SUM((odd.priceEach - pp.buyPrice) * odd.quantityOrdered)
AS profit,
        SUM(odd.quantityOrdered) AS totalQuantityOrdered
    FROM
        orderdetails odd
    JOIN
        products pp ON odd.productCode = pp.productCode
    GROUP BY
        odd.productCode
) od ON p.productCode = od.productCode
ORDER BY
    quantityToProfitRatio DESC;
```



#	productCode	productName	productLine		quantityInStock	wareh	buyPrice	MSRP	totalRevenue	profit	T_Qordered	QP
1	S10_1949	1952 Alpine Renault 1300	Classic Cars	. . .	7305	b	98.58	214.30	190017.96	95282.58	961	99.149407
2	S12_1108	2001 Ferrari Enzo	Classic Cars	. . .	3619	b	95.59	207.80	190755.86	93349.65	1019	91.609078
3	S10_4698	2003 Harley-Davidson Eagle D…	Motorcycles	. . .	5582	a	91.02	193.66	170686.00	81031.30	985	82.265279
4	S12_1099	1968 Ford Mustang	Classic Cars	. . .	68	b	95.34	194.57	161531.48	72579.26	933	77.791275
5	S18_2795	1928 Mercedes-Benz SSK	Vintage Cars	. . .	548	c	72.56	168.75	132275.98	68423.18	880	77.753614
6	S18_3232	1992 Ferrari 360 Spider red	Classic Cars	. . .	8347	b	77.90	169.34	276839.98	135996.78	1808	75.219458
7	S12_3891	1969 Ford Falcon	Classic Cars	. . .	1049	b	83.05	173.02	152543.02	72399.77	965	75.025668
8	S18_1749	1917 Grand Touring Sedan	Vintage Cars	. . .	2724	c	86.70	170.00	140535.60	60945.00	918	66.388889
9	S18_3685	1948 Porsche Type 356 Roadster	Classic Cars	. . .	8990	b	62.16	141.28	121653.46	62725.78	948	66.166435
10	S12_2823	2002 Suzuki XREO	Motorcycles	. . .	9997	a	66.27	150.62	135767.03	67641.47	1028	65.799095
11	S18_1662	1980s Black Hawk Helicopter	Planes	. . .	5330	a	77.27	157.69	144959.91	64599.11	1040	62.114529
12	S18_2870	1999 Indy 500 Monte Carlo SS	Classic Cars	. . .	8164	b	56.76	132.00	100770.12	52240.32	855	61.099789
13	S18_3482	1976 Ford Gran Torino	Classic Cars	. . .	9127	b	73.49	146.99	121890.60	54647.25	915	59.723770
14	S18_4721	1957 Corvette Convertible	Classic Cars	. . .	1249	b	69.93	148.80	130749.31	59910.22	1013	59.141382
15	S18_3140	1903 Ford Model A	Vintage Cars	. . .	3913	c	68.30	136.59	111528.82	51219.92	883	58.006704
16	S18_2325	1932 Model A Ford J-Coupe	Vintage Cars	. . .	9354	c	58.48	127.13	109992.01	54026.65	957	56.454180
17	S24_2300	1962 Volkswagen Microbus	Trucks and…	. . .	2327	d	61.34	127.79	118774.33	55655.47	1029	54.086948
18	S12_4473	1957 Chevy Pickup	Trucks and…	. . .	6125	d	55.70	118.50	109946.21	51127.01	1056	48.415729
19	S12_3148	1969 Corvair Monza	Classic Cars	. . .	6906	b	89.14	151.08	132363.79	46521.97	963	48.309418
20	S18_1097	1940 Ford Pickup Truck	Trucks and…	. . .	2613	d	58.33	116.67	105563.12	47291.45	999	47.338789

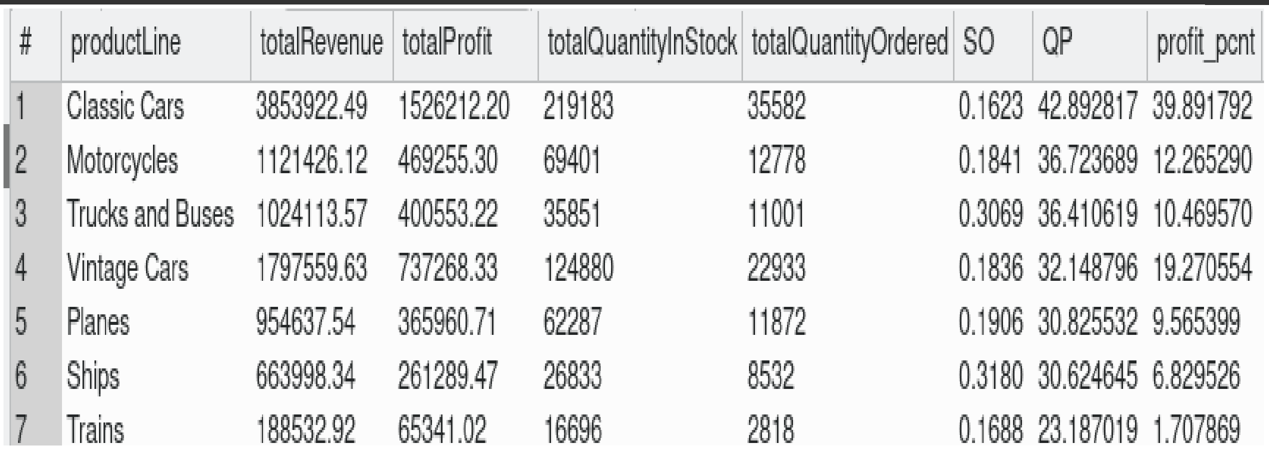
Top 20 Least Profitable Products Based on QP



productCode	productName	productLine				quantityInStock	wareh	buyPrice	MSRP	totalRevenue	profit	T_Qordered	QP
S18_3233	1985 Toyota Supra	Classic Cars	.	.	.	7733	b	57.01	107.57	0.00	0.00	0	0.000000
S24_1937	1939 Chevrolet Deluxe Coupe	Vintage Cars	.	.	.	7332	c	22.57	33.19	28052.94	6904.85	937	7.369104
S50_1341	1930 Buick Marquette Phaeton	Vintage Cars	.	.	.	7062	c	27.06	43.64	41599.24	12536.80	1074	11.672998
S72_1253	Boeing X-32A JSF	Planes	.	.	.	4857	a	32.77	49.66	42692.53	11233.33	960	11.701385
S32_2206	1982 Ducati 996 R	Motorcycles	.	.	.	9241	a	24.14	40.23	33268.76	11397.92	906	12.580486
S24_2360	1982 Ducati 900 Monster	Motorcycles	.	.	.	6840	a	47.10	69.26	57995.25	13391.55	947	14.141024
S24_3969	1936 Mercedes Benz 500k Ro…	Vintage Cars	.	.	.	2081	c	21.75	41.03	29763.39	11841.39	824	14.370619
S18_1889	1948 Porsche 356-A Roadster	Classic Cars	.	.	.	8826	b	53.90	77.00	66455.62	14064.82	972	14.469979
S24_3371	1971 Alpine Renault 1600s	Classic Cars	.	.	.	7995	b	38.58	61.23	52339.53	14955.51	969	15.433963
S24_2840	1958 Chevy Corvette Limited …	Classic Cars	.	.	.	2542	b	15.91	35.36	31627.96	15988.43	983	16.264934
S72_3212	Pont Yacht	Ships	.	.	:	414	d	33.30	54.60	47550.40	15649.00	958	16.335073
S50_1514	1962 City of Detroit Streetcar	Trains	.	.	.	1645	d	37.49	58.58	52123.81	15908.47	966	16.468395
S24_1628	1966 Shelby Cobra 427 S/C	Classic Cars	.	.	.	8197	b	29.18	50.31	42015.54	15315.84	915	16.738623
S24_1046	1970 Chevy Chevelle SS 454	Classic Cars	.	.	.	1005	b	49.24	73.49	53236.67	13696.95	803	17.057223
S24_2972	1982 Lamborghini Diablo	Classic Cars	.	.	.	7723	b	16.24	37.76	30972.87	16161.99	912	17.721480
S700_3167	F/A 18 Hornet 1/72	Planes	.	.	.	551	a	54.40	80.00	76618.40	19661.60	1047	18.778988
S24_1444	1970 Dodge Coronet	Classic Cars	.	.	.	4074	b	32.37	57.80	50255.45	18662.33	976	19.121240
S24_2022	1938 Cadillac V-16 Presidentia…	Vintage Cars	.	.	.	2847	c	20.61	44.80	38449.09	18766.54	955	19.650827
S18_4668	1939 Cadillac Limousine	Vintage Cars	.	.	.	6645	c	23.14	50.31	44037.84	21013.54	995	21.119136
S18_2248	1911 Ford Town Car	Vintage Cars	.	.	.	540	c	33.30	60.54	45306.77	17601.17	832	21.155252
S32_1374	1997 BMW F650 ST	Motorcycles	.	.	.	178	a	66.92	99.89	89364.89	21508.01	1014	21.211055
S18_3278	1969 Dodge Super Bee	Classic Cars	.	.	.	1917	b	49.05	80.41	68783.93	21009.23	974	21.570051

Product Orders And profit Based on Product Line

```
SELECT
    p.productLine, SUM(od.totalRevenue) AS totalRevenue,
    SUM(od.profit) AS totalProfit,
    SUM(p.quantityInStock) AS totalQuantityInStock,
    SUM(od.totalQuantityOrdered) AS totalQuantityOrdered,
    SUM(od.totalQuantityOrdered) / SUM(p.quantityInStock) AS SO,
    CASE
        WHEN SUM(od.profit) > 0 THEN SUM(od.profit) / SUM(od.totalQuantityOrdered)
        ELSE 0
    END AS QP,
    SUM(od.profit) * 100 / totalTotalProfit.totalTotalProfit AS profit_pcnt
FROM products p
LEFT JOIN (
    SELECT
        odd.productCode,
        SUM(odd.priceEach * odd.quantityOrdered) AS totalRevenue,
        SUM((odd.priceEach - pp.buyPrice) * odd.quantityOrdered) AS profit,
        SUM(odd.quantityOrdered) AS totalQuantityOrdered
    FROM orderdetails odd
    JOIN
        products pp ON odd.productCode = pp.productCode
    GROUP BY odd.productCode
) od ON p.productCode = od.productCode
JOIN (
    SELECT
        SUM((odd.priceEach - pp.buyPrice) * odd.quantityOrdered) AS totalTotalProfit
    FROM orderdetails odd
    JOIN
        products pp ON odd.productCode = pp.productCode
) AS totalTotalProfit
GROUP BY p.productLine, totalTotalProfit.totalTotalProfit
ORDER BY QP DESC;
```



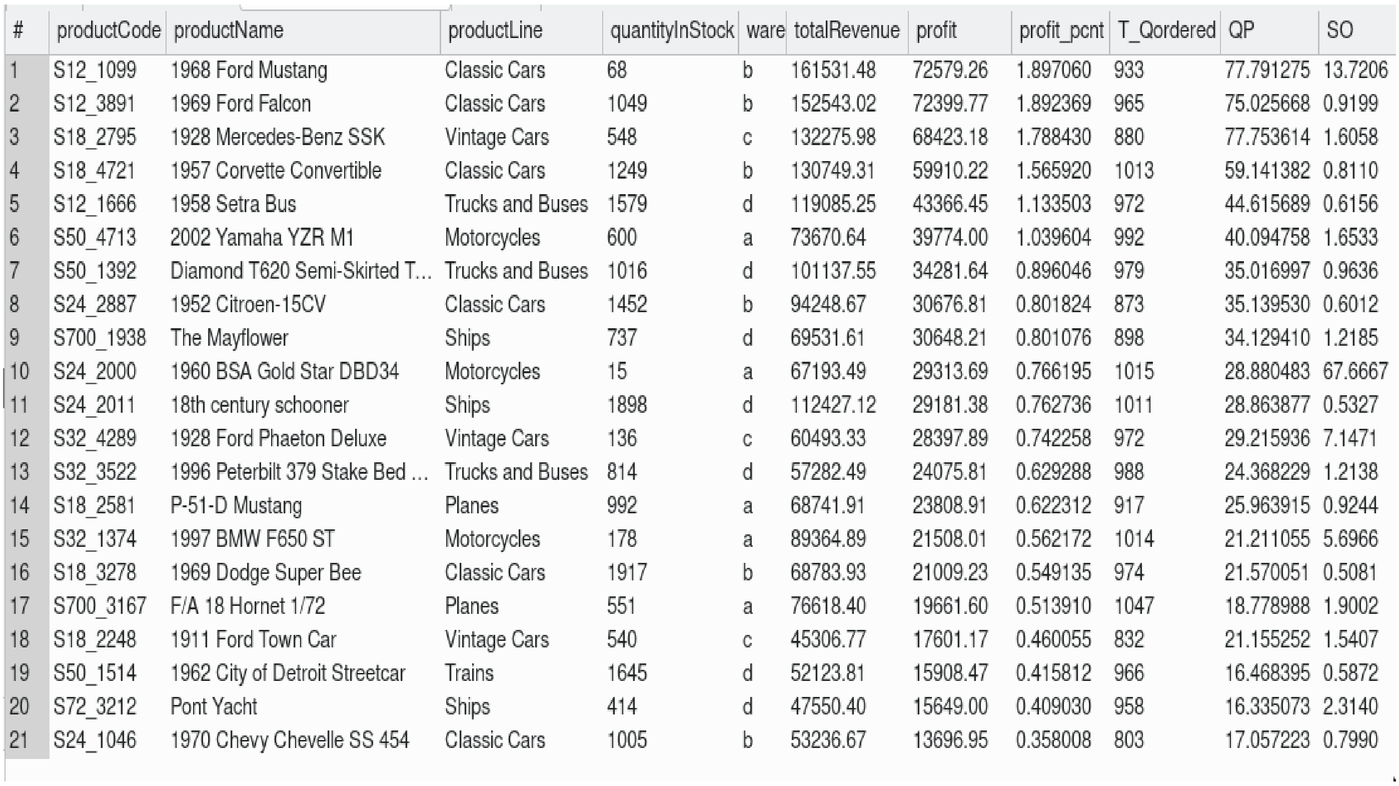
#	productLine	totalRevenue	totalProfit	totalQuantityInStock	totalQuantityOrdered	SO	QP	profit_pcnt
1	Classic Cars	3853922.49	1526212.20	219183	35582	0.1623	42.892817	39.891792
2	Motorcycles	1121426.12	469255.30	69401	12778	0.1841	36.723689	12.265290
3	Trucks and Buses	1024113.57	400553.22	35851	11001	0.3069	36.410619	10.469570
4	Vintage Cars	1797559.63	737268.33	124880	22933	0.1836	32.148796	19.270554
5	Planes	954637.54	365960.71	62287	11872	0.1906	30.825532	9.565399
6	Ships	663998.34	261289.47	26833	8532	0.3180	30.624645	6.829526
7	Trains	188532.92	65341.02	16696	2818	0.1688	23.187019	1.707869

SO = Stock to Order ratio = Order / stock = more value means orders are more and less mean stock is more

Understock Products

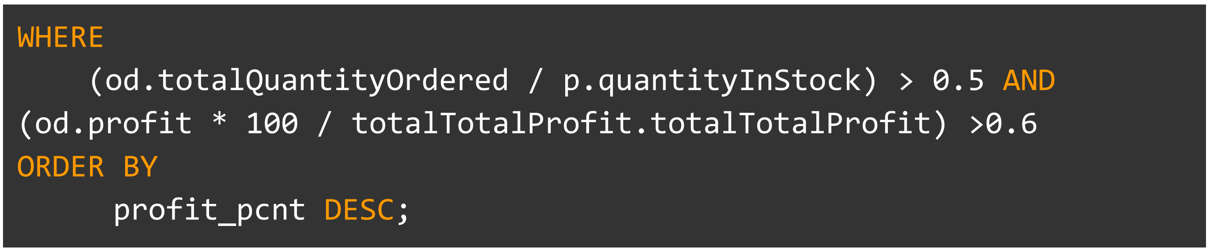
```
SELECT
    p.productCode, p.productName, p.productLine, p.quantityInStock,
p.warehouseCode,
    COALESCE(od.totalRevenue, 0) AS totalRevenue,
    COALESCE(od.profit, 0) AS profit,
    CASE
        WHEN od.profit > 0 THEN od.profit * 100 /
totalTotalProfit.totalTotalProfit
        ELSE 0
    END AS profit_pcnt,
    COALESCE(od.totalQuantityOrdered, 0) AS T_Qordered,
    CASE
        WHEN od.profit > 0 THEN od.profit / od.totalQuantityOrdered
        ELSE 0
    END AS QP,
    CASE
        WHEN od.totalQuantityOrdered > 0 THEN od.totalQuantityOrdered /
p.quantityInStock
        ELSE 0
    END AS SO
FROM products p
LEFT JOIN (
    SELECT
        odd.productCode,
        SUM(odd.priceEach * odd.quantityOrdered) AS totalRevenue,
        SUM((odd.priceEach - pp.buyPrice) * odd.quantityOrdered) AS profit,
        SUM(odd.quantityOrdered) AS totalQuantityOrdered
    FROM orderdetails odd
    JOIN products pp ON odd.productCode = pp.productCode
    GROUP BY odd.productCode
) od ON p.productCode = od.productCode
CROSS JOIN (
    SELECT
        SUM((odd.priceEach - pp.buyPrice) * odd.quantityOrdered) AS
totalTotalProfit
    FROM orderdetails odd
    JOIN products pp ON odd.productCode = pp.productCode
) AS totalTotalProfit
WHERE (od.totalQuantityOrdered / p.quantityInStock) > 0.5
ORDER BY profit_pcnt DESC;
```

Ordered By Profit

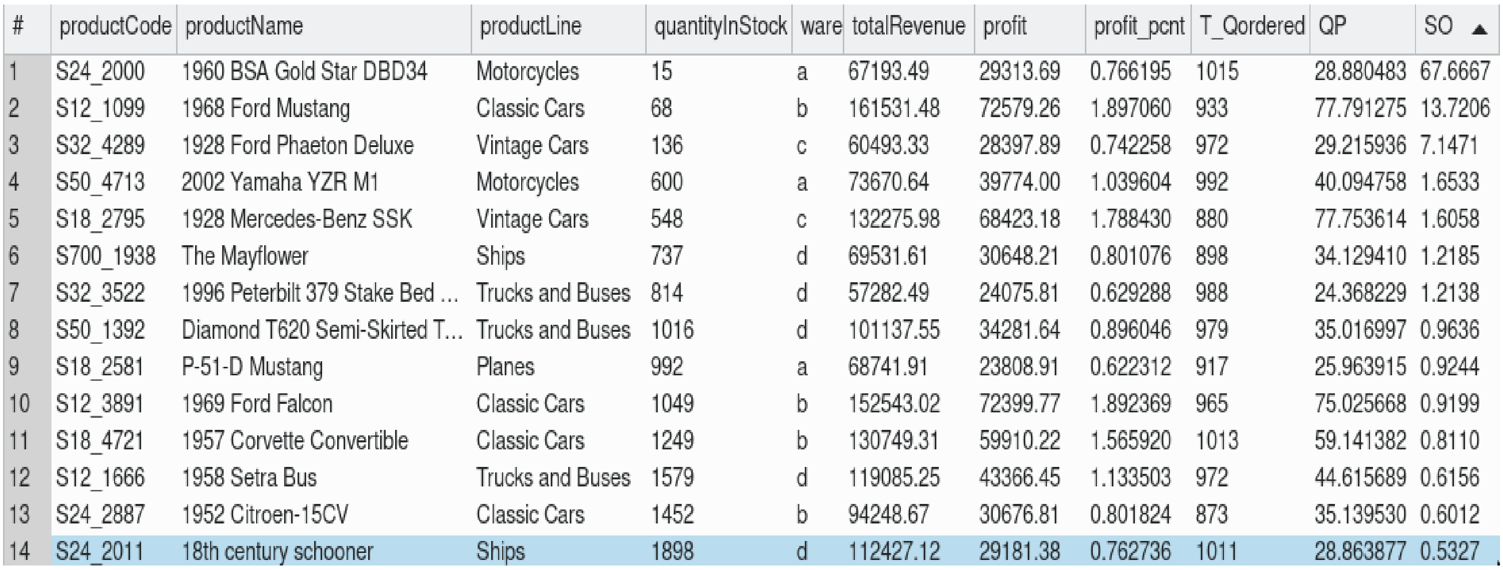


#	productCode	productName	productLine	quantityInStock	ware	totalRevenue	profit	profit_pcnt	T_Qordered	QP	SO
1	S12_1099	1968 Ford Mustang	Classic Cars	68	b	161531.48	72579.26	1.897060	933	77.791275	13.7206
2	S12_3891	1969 Ford Falcon	Classic Cars	1049	b	152543.02	72399.77	1.892369	965	75.025668	0.9199
3	S18_2795	1928 Mercedes-Benz SSK	Vintage Cars	548	c	132275.98	68423.18	1.788430	880	77.753614	1.6058
4	S18_4721	1957 Corvette Convertible	Classic Cars	1249	b	130749.31	59910.22	1.565920	1013	59.141382	0.8110
5	S12_1666	1958 Setra Bus	Trucks and Buses	1579	d	119085.25	43366.45	1.133503	972	44.615689	0.6156
6	S50_4713	2002 Yamaha YZR M1	Motorcycles	600	a	73670.64	39774.00	1.039604	992	40.094758	1.6533
7	S50_1392	Diamond T620 Semi-Skirted T...	Trucks and Buses	1016	d	101137.55	34281.64	0.896046	979	35.016997	0.9636
8	S24_2887	1952 Citroen-15CV	Classic Cars	1452	b	94248.67	30676.81	0.801824	873	35.139530	0.6012
9	S700_1938	The Mayflower	Ships	737	d	69531.61	30648.21	0.801076	898	34.129410	1.2185
10	S24_2000	1960 BSA Gold Star DBD34	Motorcycles	15	a	67193.49	29313.69	0.766195	1015	28.880483	67.6667
11	S24_2011	18th century schooner	Ships	1898	d	112427.12	29181.38	0.762736	1011	28.863877	0.5327
12	S32_4289	1928 Ford Phaeton Deluxe	Vintage Cars	136	c	60493.33	28397.89	0.742258	972	29.215936	7.1471
13	S32_3522	1996 Peterbilt 379 Stake Bed ...	Trucks and Buses	814	d	57282.49	24075.81	0.629288	988	24.368229	1.2138
14	S18_2581	P-51-D Mustang	Planes	992	a	68741.91	23808.91	0.622312	917	25.963915	0.9244
15	S32_1374	1997 BMW F650 ST	Motorcycles	178	a	89364.89	21508.01	0.562172	1014	21.211055	5.6966
16	S18_3278	1969 Dodge Super Bee	Classic Cars	1917	b	68783.93	21009.23	0.549135	974	21.570051	0.5081
17	S700_3167	F/A 18 Hornet 1/72	Planes	551	a	76618.40	19661.60	0.513910	1047	18.778988	1.9002
18	S18_2248	1911 Ford Town Car	Vintage Cars	540	c	45306.77	17601.17	0.460055	832	21.155252	1.5407
19	S50_1514	1962 City of Detroit Streetcar	Trains	1645	d	52123.81	15908.47	0.415812	966	16.468395	0.5872
20	S72_3212	Pont Yacht	Ships	414	d	47550.40	15649.00	0.409030	958	16.335073	2.3140
21	S24_1046	1970 Chevy Chevelle SS 454	Classic Cars	1005	b	53236.67	13696.95	0.358008	803	17.057223	0.7990

Products where profit percentage is > 0.6, absolutely necessary to stock



```
WHERE
    (od.totalQuantityOrdered / p.quantityInStock) > 0.5 AND
(od.profit * 100 / totalTotalProfit.totalTotalProfit) >0.6
ORDER BY
     profit_pcnt DESC;
```

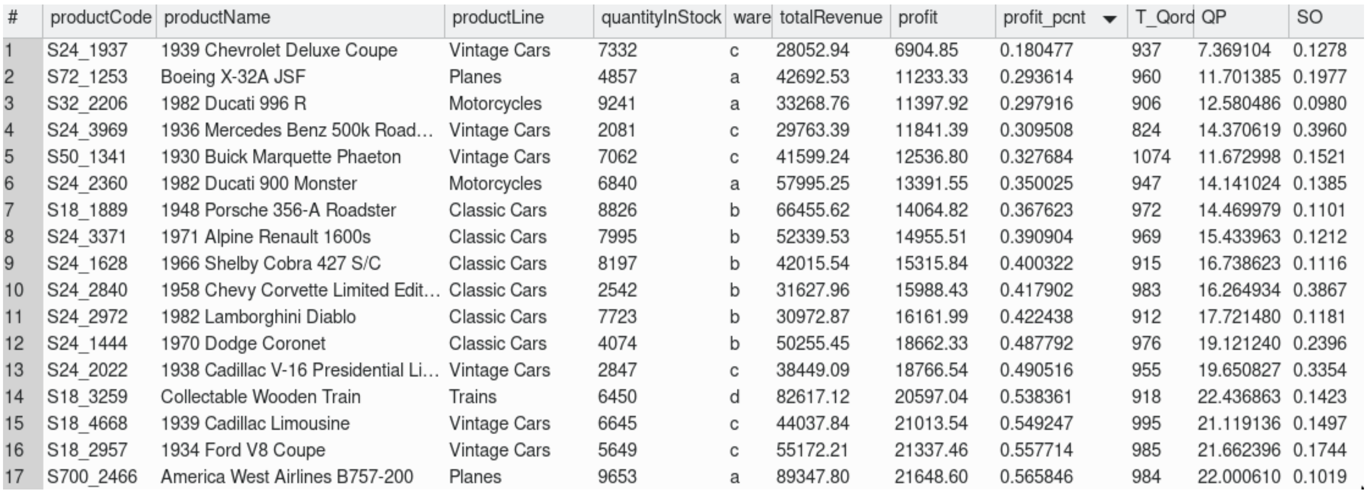


#	productCode	productName	productLine	quantityInStock	ware	totalRevenue	profit	profit_pcnt	T_Qordered	QP	SO ▲
1	S24_2000	1960 BSA Gold Star DBD34	Motorcycles	15	a	67193.49	29313.69	0.766195	1015	28.880483	67.6667
2	S12_1099	1968 Ford Mustang	Classic Cars	68	b	161531.48	72579.26	1.897060	933	77.791275	13.7206
3	S32_4289	1928 Ford Phaeton Deluxe	Vintage Cars	136	c	60493.33	28397.89	0.742258	972	29.215936	7.1471
4	S50_4713	2002 Yamaha YZR M1	Motorcycles	600	a	73670.64	39774.00	1.039604	992	40.094758	1.6533
5	S18_2795	1928 Mercedes-Benz SSK	Vintage Cars	548	c	132275.98	68423.18	1.788430	880	77.753614	1.6058
6	S700_1938	The Mayflower	Ships	737	d	69531.61	30648.21	0.801076	898	34.129410	1.2185
7	S32_3522	1996 Peterbilt 379 Stake Bed ...	Trucks and Buses	814	d	57282.49	24075.81	0.629288	988	24.368229	1.2138
8	S50_1392	Diamond T620 Semi-Skirted T...	Trucks and Buses	1016	d	101137.55	34281.64	0.896046	979	35.016997	0.9636
9	S18_2581	P-51-D Mustang	Planes	992	a	68741.91	23808.91	0.622312	917	25.963915	0.9244
10	S12_3891	1969 Ford Falcon	Classic Cars	1049	b	152543.02	72399.77	1.892369	965	75.025668	0.9199
11	S18_4721	1957 Corvette Convertible	Classic Cars	1249	b	130749.31	59910.22	1.565920	1013	59.141382	0.8110
12	S12_1666	1958 Setra Bus	Trucks and Buses	1579	d	119085.25	43366.45	1.133503	972	44.615689	0.6156
13	S24_2887	1952 Citroen-15CV	Classic Cars	1452	b	94248.67	30676.81	0.801824	873	35.139530	0.6012
14	S24_2011	18th century schooner	Ships	1898	d	112427.12	29181.38	0.762736	1011	28.863877	0.5327

Overstock Products

```
SELECT
    p.productCode, p.productName, p.productLine, p.quantityInStock,
p.warehouseCode,
    COALESCE(od.totalRevenue, 0) AS totalRevenue,
    COALESCE(od.profit, 0) AS profit,
    CASE
        WHEN od.profit > 0 THEN od.profit * 100 /
totalTotalProfit.totalTotalProfit
        ELSE 0
    END AS profit_pcnt,
    COALESCE(od.totalQuantityOrdered, 0) AS T_Qordered,
    CASE
        WHEN od.profit > 0 THEN od.profit / od.totalQuantityOrdered
        ELSE 0
    END AS QP,
    CASE
        WHEN od.totalQuantityOrdered > 0 THEN od.totalQuantityOrdered /
p.quantityInStock
        ELSE 0
    END AS SO
FROM products p
LEFT JOIN (
    SELECT
        odd.productCode,
        SUM(odd.priceEach * odd.quantityOrdered) AS totalRevenue,
        SUM((odd.priceEach - pp.buyPrice) * odd.quantityOrdered) AS profit,
        SUM(odd.quantityOrdered) AS totalQuantityOrdered
    FROM orderdetails odd
    JOIN products pp ON odd.productCode = pp.productCode
    GROUP BY odd.productCode
) od ON p.productCode = od.productCode
CROSS JOIN (
    SELECT
        SUM((odd.priceEach - pp.buyPrice) * odd.quantityOrdered) AS
totalTotalProfit
    FROM orderdetails odd
    JOIN products pp ON odd.productCode = pp.productCode
) AS totalTotalProfit
WHERE (od.totalQuantityOrdered / p.quantityInStock) < 0.5
ORDER BY profit_pcnt DESC;
```

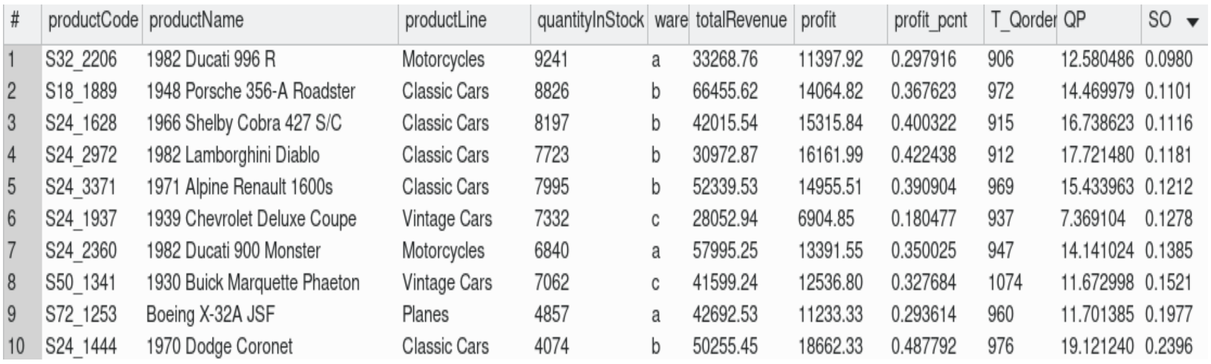
These are Sevearly overstocked and also profit is low



#	productCode	productName	productLine	quantityInStock	ware	totalRevenue	profit	profit_pcnt	T_Qord	QP	SO
1	S24_1937	1939 Chevrolet Deluxe Coupe	Vintage Cars	7332	c	28052.94	6904.85	0.180477	937	7.369104	0.1278
2	S72_1253	Boeing X-32A JSF	Planes	4857	a	42692.53	11233.33	0.293614	960	11.701385	0.1977
3	S32_2206	1982 Ducati 996 R	Motorcycles	9241	a	33268.76	11397.92	0.297916	906	12.580486	0.0980
4	S24_3969	1936 Mercedes Benz 500k Road…	Vintage Cars	2081	c	29763.39	11841.39	0.309508	824	14.370619	0.3960
5	S50_1341	1930 Buick Marquette Phaeton	Vintage Cars	7062	c	41599.24	12536.80	0.327684	1074	11.672998	0.1521
6	S24_2360	1982 Ducati 900 Monster	Motorcycles	6840	a	57995.25	13391.55	0.350025	947	14.141024	0.1385
7	S18_1889	1948 Porsche 356-A Roadster	Classic Cars	8826	b	66455.62	14064.82	0.367623	972	14.469979	0.1101
8	S24_3371	1971 Alpine Renault 1600s	Classic Cars	7995	b	52339.53	14955.51	0.390904	969	15.433963	0.1212
9	S24_1628	1966 Shelby Cobra 427 S/C	Classic Cars	8197	b	42015.54	15315.84	0.400322	915	16.738623	0.1116
10	S24_2840	1958 Chevy Corvette Limited Edit…	Classic Cars	2542	b	31627.96	15988.43	0.417902	983	16.264934	0.3867
11	S24_2972	1982 Lamborghini Diablo	Classic Cars	7723	b	30972.87	16161.99	0.422438	912	17.721480	0.1181
12	S24_1444	1970 Dodge Coronet	Classic Cars	4074	b	50255.45	18662.33	0.487792	976	19.121240	0.2396
13	S24_2022	1938 Cadillac V-16 Presidential Li…	Vintage Cars	2847	c	38449.09	18766.54	0.490516	955	19.650827	0.3354
14	S18_3259	Collectable Wooden Train	Trains	6450	d	82617.12	20597.04	0.538361	918	22.436863	0.1423
15	S18_4668	1939 Cadillac Limousine	Vintage Cars	6645	c	44037.84	21013.54	0.549247	995	21.119136	0.1497
16	S18_2957	1934 Ford V8 Coupe	Vintage Cars	5649	c	55172.21	21337.46	0.557714	985	21.662396	0.1744
17	S700_2466	America West Airlines B757-200	Planes	9653	a	89347.80	21648.60	0.565846	984	22.000610	0.1019

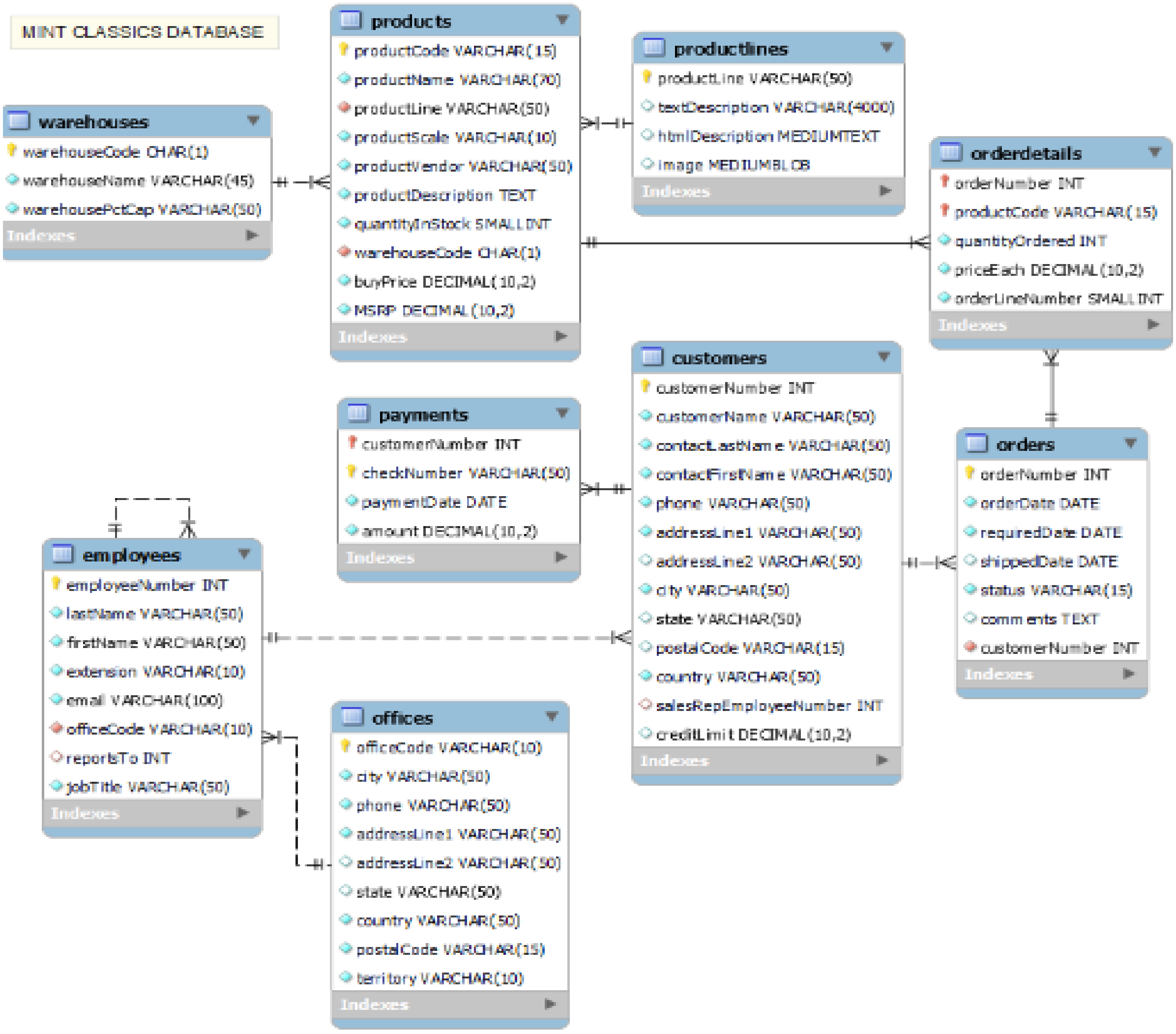
Information is important thus i have created this sheet Mint Classic

Absulutely overstocked , Need to remove them or shift them to warehouse , make one warehouse as storage box



#	productCode	productName	productLine	quantityInStock	ware	totalRevenue	profit	profit_pcnt	T_Qorder	QP	SO
1	S32_2206	1982 Ducati 996 R	Motorcycles	9241	a	33268.76	11397.92	0.297916	906	12.580486	0.0980
2	S18_1889	1948 Porsche 356-A Roadster	Classic Cars	8826	b	66455.62	14064.82	0.367623	972	14.469979	0.1101
3	S24_1628	1966 Shelby Cobra 427 S/C	Classic Cars	8197	b	42015.54	15315.84	0.400322	915	16.738623	0.1116
4	S24_2972	1982 Lamborghini Diablo	Classic Cars	7723	b	30972.87	16161.99	0.422438	912	17.721480	0.1181
5	S24_3371	1971 Alpine Renault 1600s	Classic Cars	7995	b	52339.53	14955.51	0.390904	969	15.433963	0.1212
6	S24_1937	1939 Chevrolet Deluxe Coupe	Vintage Cars	7332	c	28052.94	6904.85	0.180477	937	7.369104	0.1278
7	S24_2360	1982 Ducati 900 Monster	Motorcycles	6840	a	57995.25	13391.55	0.350025	947	14.141024	0.1385
8	S50_1341	1930 Buick Marquette Phaeton	Vintage Cars	7062	c	41599.24	12536.80	0.327684	1074	11.672998	0.1521
9	S72_1253	Boeing X-32A JSF	Planes	4857	a	42692.53	11233.33	0.293614	960	11.701385	0.1977
10	S24_1444	1970 Dodge Coronet	Classic Cars	4074	b	50255.45	18662.33	0.487792	976	19.121240	0.2396

Now Warehousing



My current goal is to identify warehouse location and connect to customers , we can create map then it will be easy to visulize which warehouse we can shutdown also with based on over and understock

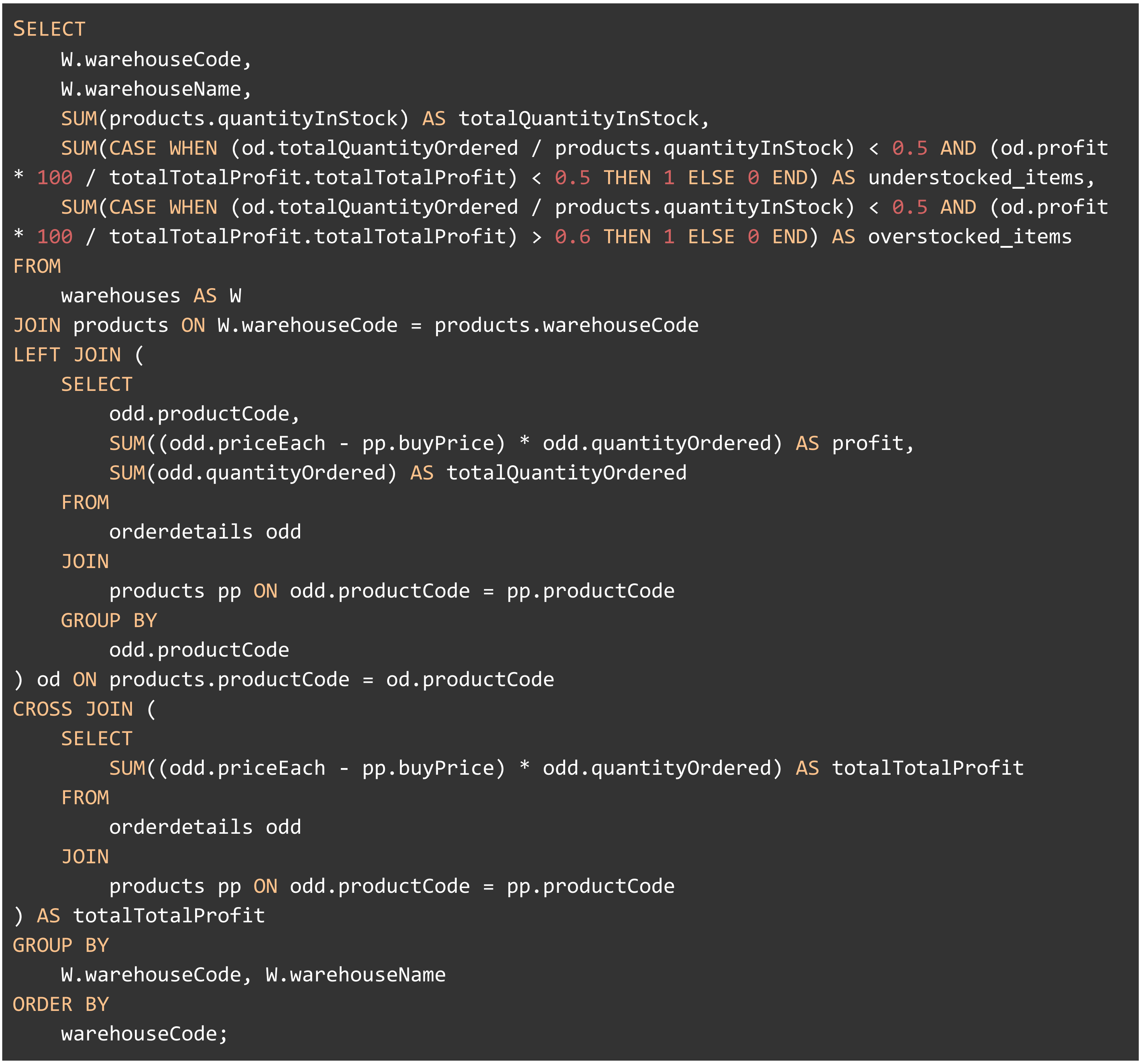
See this took entire 5 min to just connect:

I found that customer , and orders have customercode as key, now i can *connect warehousecode -> products.productcode->ordersdetails.productcode->ordedetails.ordernumber->orders.ordernumber->orders.customernumber->customer.customernumber*

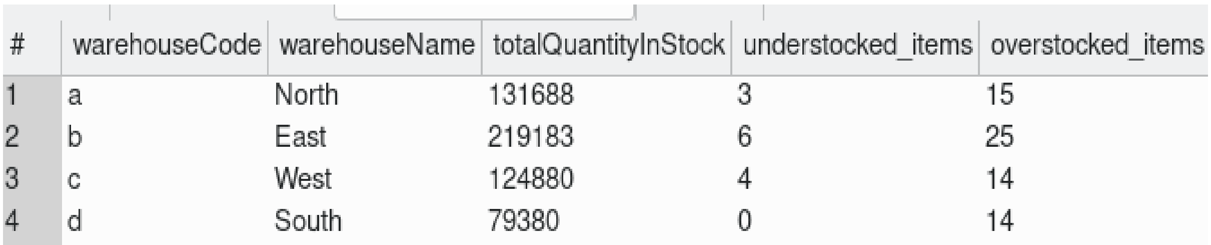
```sql
SELECT
    o.orderNumber,
    c.customerName,
    c.contactLastName,
    c.contactFirstName,
    c.addressLine1 AS customerAddressLine1,
    c.postalCode AS customerPostalCode,
    w.warehouseCode,
    p.productName,
    od.quantityOrdered,
    od.priceEach,
    emp.employeeNumber,
    emp.lastName AS employeeLastName,
    emp.firstName AS employeeFirstName,
    emp.email AS employeeEmail,
    emp.jobTitle AS employeeJobTitle,
    offs.officeCode,
    offs.city AS officeCity,
    offs.phone AS officePhone,
    offs.addressLine1 AS officeAddressLine1,
    offs.state AS officeState,
    offs.country AS officeCountry,
    offs.postalCode AS officePostalCode
FROM
    orders o
JOIN
    customers c ON o.customerNumber = c.customerNumber
JOIN
    orderdetails od ON o.orderNumber = od.orderNumber
JOIN
    products p ON od.productCode = p.productCode
JOIN
    warehouses w ON p.warehouseCode = w.warehouseCode
JOIN
    employees emp ON c.salesRepEmployeeNumber = emp.employeeNumber
JOIN
    offices offs ON emp.officeCode = offs.officeCode
ORDER BY
    o.orderNumber, p.productName;
```

Well forget this approach for now

Overstock and understock items per warehouse



```sql
SELECT
    W.warehouseCode,
    W.warehouseName,
    SUM(products.quantityInStock) AS totalQuantityInStock,
    SUM(CASE WHEN (od.totalQuantityOrdered / products.quantityInStock) < 0.5 AND (od.profit
* 100 / totalTotalProfit.totalTotalProfit) < 0.5 THEN 1 ELSE 0 END) AS understocked_items,
    SUM(CASE WHEN (od.totalQuantityOrdered / products.quantityInStock) < 0.5 AND (od.profit
* 100 / totalTotalProfit.totalTotalProfit) > 0.6 THEN 1 ELSE 0 END) AS overstocked_items
FROM
    warehouses AS W
JOIN products ON W.warehouseCode = products.warehouseCode
LEFT JOIN (
    SELECT
        odd.productCode,
        SUM((odd.priceEach - pp.buyPrice) * odd.quantityOrdered) AS profit,
        SUM(odd.quantityOrdered) AS totalQuantityOrdered
    FROM
        orderdetails odd
    JOIN
        products pp ON odd.productCode = pp.productCode
    GROUP BY
        odd.productCode
) od ON products.productCode = od.productCode
CROSS JOIN (
    SELECT
        SUM((odd.priceEach - pp.buyPrice) * odd.quantityOrdered) AS totalTotalProfit
    FROM
        orderdetails odd
    JOIN
        products pp ON odd.productCode = pp.productCode
) AS totalTotalProfit
GROUP BY
    W.warehouseCode, W.warehouseName
ORDER BY
    warehouseCode;
```



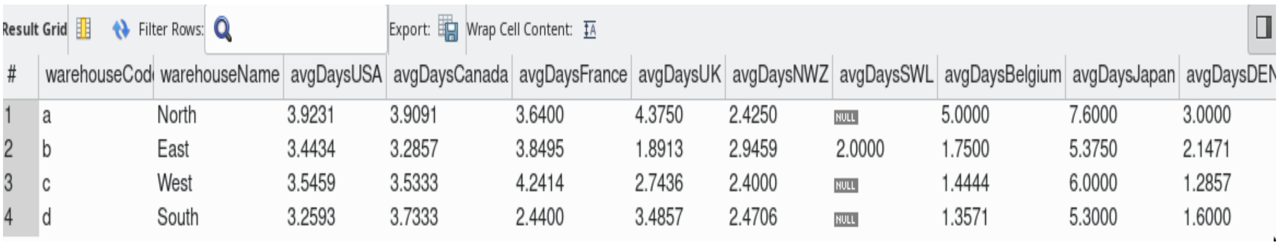
#	warehouseCode	warehouseName	totalQuantityInStock	understocked_items	overstocked_items
1	a	North	131688	3	15
2	b	East	219183	6	25
3	c	West	124880	4	14
4	d	South	79380	0	14

WAREHOUSE ANALYSIS

LOCATING WAREHOUSES - MISSION HALF FAILED

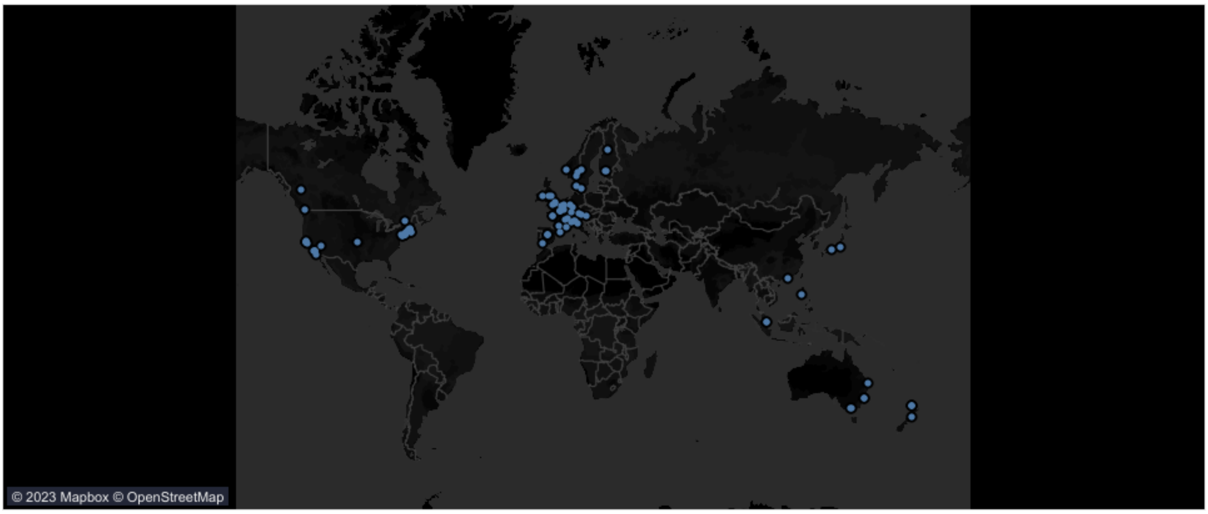
classic cars dataset   Mint Classic

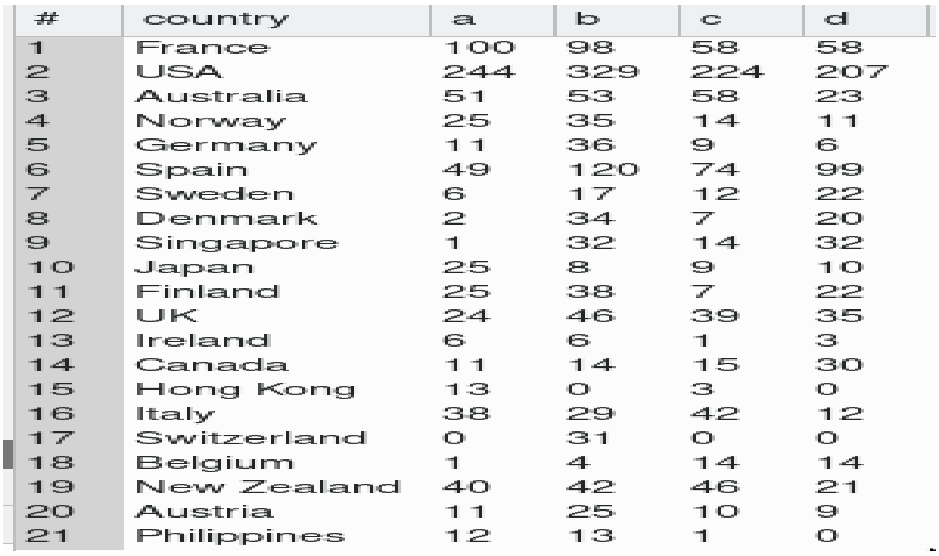
```
SELECT
    p.warehouseCode,
    W.warehouseName,
    AVG(CASE WHEN c.country = 'USA' THEN DATEDIFF(o.shippedDate,
o.orderDate) END) AS avgDaysUSA,
    AVG(CASE WHEN c.country = 'Canada' THEN
DATEDIFF(o.shippedDate, o.orderDate) END) AS avgDaysCanada,
    AVG(CASE WHEN c.country = 'France' THEN
DATEDIFF(o.shippedDate, o.orderDate) END) AS avgDaysFrance,
    AVG(CASE WHEN c.country = 'UK' THEN DATEDIFF(o.shippedDate,
o.orderDate) END) AS avgDaysUK,
    AVG(CASE WHEN c.country = 'New Zealand' THEN
DATEDIFF(o.shippedDate, o.orderDate) END) AS avgDaysNWZ,
    AVG(CASE WHEN c.country = 'Switzerland' THEN
DATEDIFF(o.shippedDate, o.orderDate) END) AS avgDaysSWL,
    AVG(CASE WHEN c.country = 'Belgium' THEN
DATEDIFF(o.shippedDate, o.orderDate) END) AS avgDaysBelgium,
    AVG(CASE WHEN c.country = 'Japan' THEN DATEDIFF(o.shippedDate,
o.orderDate) END) AS avgDaysJapan,
    AVG(CASE WHEN c.country = 'Denmark' THEN
DATEDIFF(o.shippedDate, o.orderDate) END) AS avgDaysDEN

FROM
    products p
JOIN orderdetails od ON p.productCode = od.productCode
JOIN orders o ON od.orderNumber = o.orderNumber
JOIN customers c ON o.customerNumber = c.customerNumber
JOIN warehouses as W ON p.warehouseCode = W.warehouseCode
GROUP BY
    p.warehouseCode
ORDER BY
    p.warehouseCode;
```



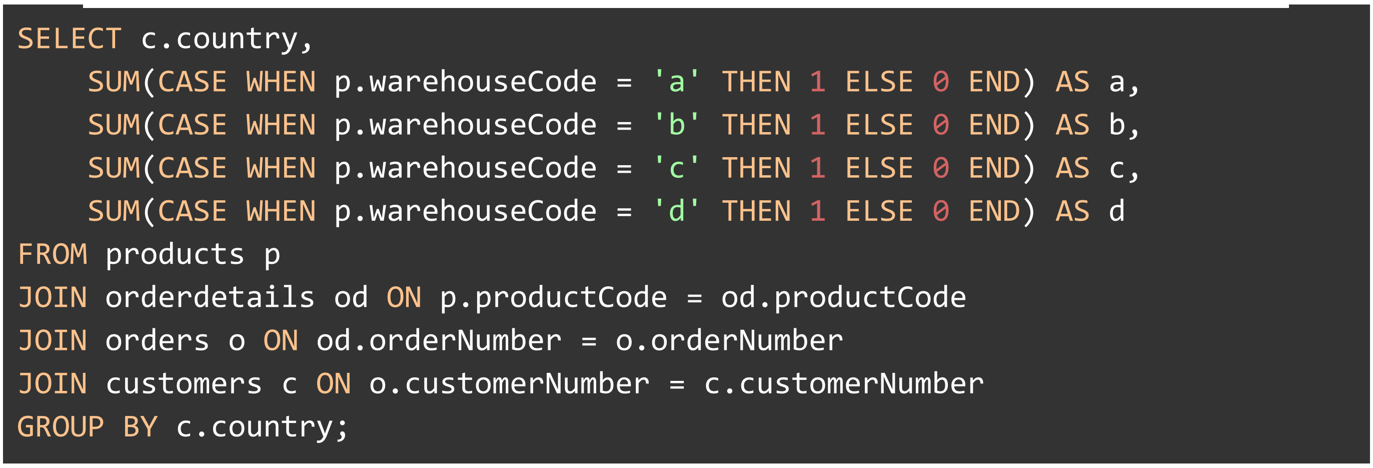
#	warehouseCod	warehouseName	avgDaysUSA	avgDaysCanada	avgDaysFrance	avgDaysUK	avgDaysNWZ	avgDaysSWL	avgDaysBelgium	avgDaysJapan	avgDaysDEN
1	a	North	3.9231	3.9091	3.6400	4.3750	2.4250	NULL	5.0000	7.6000	3.0000
2	b	East	3.4434	3.2857	3.8495	1.8913	2.9459	2.0000	1.7500	5.3750	2.1471
3	c	West	3.5459	3.5333	4.2414	2.7436	2.4000	NULL	1.4444	6.0000	1.2857
4	d	South	3.2593	3.7333	2.4400	3.4857	2.4706	NULL	1.3571	5.3000	1.6000

Sheet 1


© 2023 Mapbox © OpenStreetMap

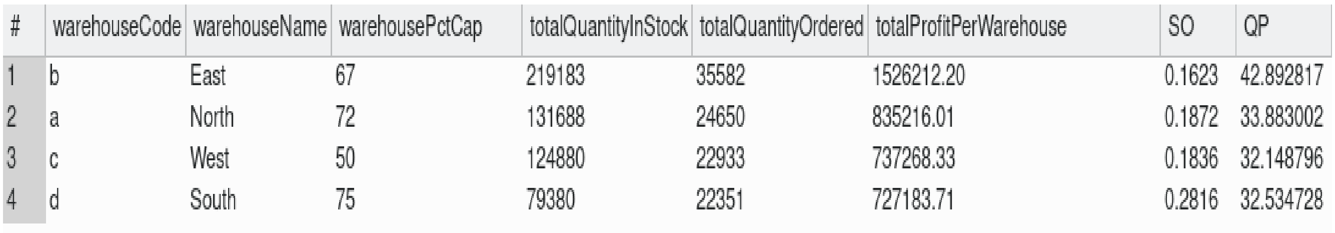


#	country	a	b	c	d
1	France	100	98	58	58
2	USA	244	329	224	207
3	Australia	51	53	58	23
4	Norway	25	35	14	11
5	Germany	11	36	9	6
6	Spain	49	120	74	99
7	Sweden	6	17	12	22
8	Denmark	2	34	7	20
9	Singapore	1	32	14	32
10	Japan	25	8	9	10
11	Finland	25	38	7	22
12	UK	24	46	39	35
13	Ireland	6	6	1	3
14	Canada	11	14	15	30
15	Hong Kong	13	0	3	0
16	Italy	38	29	42	12
17	Switzerland	0	31	0	0
18	Belgium	1	4	14	14
19	New Zealand	40	42	46	21
20	Austria	11	25	10	9
21	Philippines	12	13	1	0



```
SELECT c.country,
    SUM(CASE WHEN p.warehouseCode = 'a' THEN 1 ELSE 0 END) AS a,
    SUM(CASE WHEN p.warehouseCode = 'b' THEN 1 ELSE 0 END) AS b,
    SUM(CASE WHEN p.warehouseCode = 'c' THEN 1 ELSE 0 END) AS c,
    SUM(CASE WHEN p.warehouseCode = 'd' THEN 1 ELSE 0 END) AS d
FROM products p
JOIN orderdetails od ON p.productCode = od.productCode
JOIN orders o ON od.orderNumber = o.orderNumber
JOIN customers c ON o.customerNumber = c.customerNumber
GROUP BY c.country;
```

Warehouse Overview



#	warehouseCode	warehouseName	warehousePctCap	totalQuantityInStock	totalQuantityOrdered	totalProfitPerWarehouse	SO	QP
1	b	East	67	219183	35582	1526212.20	0.1623	42.892817
2	a	North	72	131688	24650	835216.01	0.1872	33.883002
3	c	West	50	124880	22933	737268.33	0.1836	32.148796
4	d	South	75	79380	22351	727183.71	0.2816	32.534728

```
SELECT
    W.warehouseCode,
    W.warehouseName,
    W.warehousePctCap,
    SUM(products.quantityInStock) AS totalQuantityInStock,
    SUM(od.totalQuantityOrdered) AS totalQuantityOrdered,
    SUM(od.profit) AS totalProfitPerWarehouse,
    SUM(od.totalQuantityOrdered) / SUM(products.quantityInStock) AS SO,
    CASE
        WHEN SUM(od.profit) > 0 THEN SUM(od.profit) /
SUM(od.totalQuantityOrdered)
        ELSE 0
    END AS QP
FROM
    warehouses AS W
JOIN products ON W.warehouseCode = products.warehouseCode
LEFT JOIN (
    SELECT
        odd.productCode,
        SUM((odd.priceEach - pp.buyPrice) * odd.quantityOrdered) AS profit,
        SUM(odd.quantityOrdered) AS totalQuantityOrdered
    FROM
        orderdetails odd
    JOIN
        products pp ON odd.productCode = pp.productCode
    GROUP BY
        odd.productCode
) od ON products.productCode = od.productCode
GROUP BY
    W.warehouseCode
ORDER BY
    totalQuantityInStock DESC;
```

Reducing Stock & Rearranging warehouse

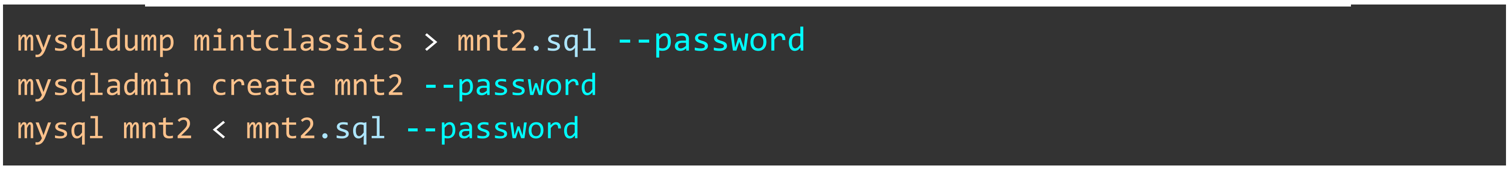
Now for reducing stock i have decided to do smart approach
We have 80 products which are heavily overstocked and out of those 80 we have 13 products which are relatively not moving and also have extreme stocks.
So plan:

[here overstock = more stock + not profitable product , understock means = less stock + product is actually very profitable]

1. Reduce toyota supra stock to 100 , [currently 7700 and zero sell]
2. Reduce those 13 products stock based on adjusting SO , = 0.8
    a. Qinstock = Orders/SO , where so will be 0.8
3. Reduce left overstock products
4. Fill heavily understocked items via 30-50 % (will be costly for sure)
5. Rearrange Products to Warehouse
    a. Based on Location of warehouse and demand of product at different countries and type of product [not possible as warehouse can't be located]
    b. Based on storage capacity
    c. Based on already stored products
    d. Based on Product line

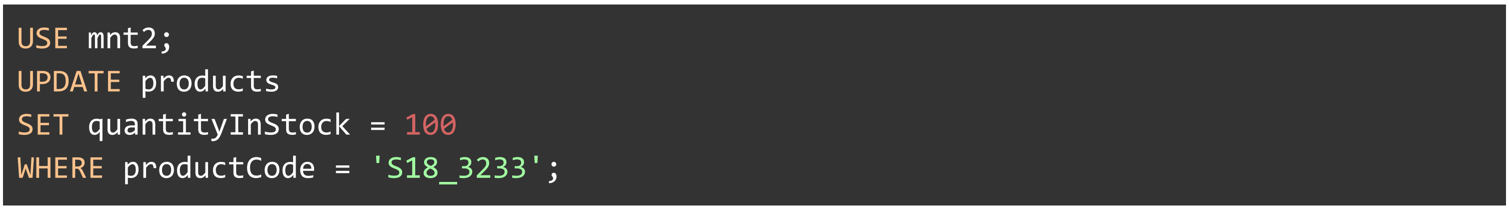
Now i am planning to duplicate the dataset and then modify stuff so i don't mess up
Because holding everything in temp tables will be extremely complicated

1. Creating duplicate



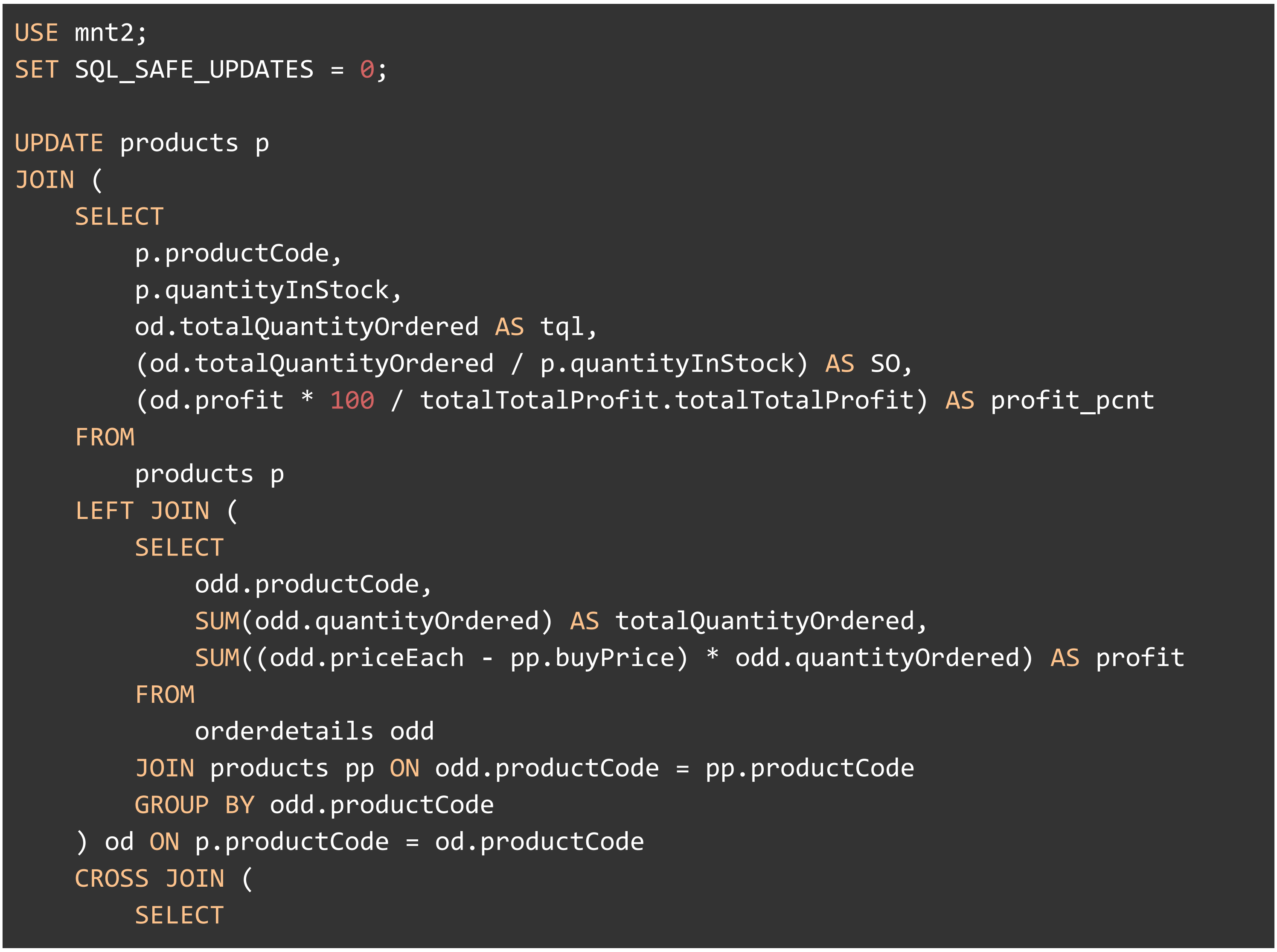
```
mysqldump mintclassics > mnt2.sql --password
mysqladmin create mnt2 --password
mysql mnt2 < mnt2.sql --password
```

2. Removing Toyota supra stocks



```
USE mnt2;
UPDATE products
SET quantityInStock = 100
WHERE productCode = 'S18_3233';
```

3. Reducing Overstock Products Based on SO = 0.8
    a. Q.instock = ordered/0.8

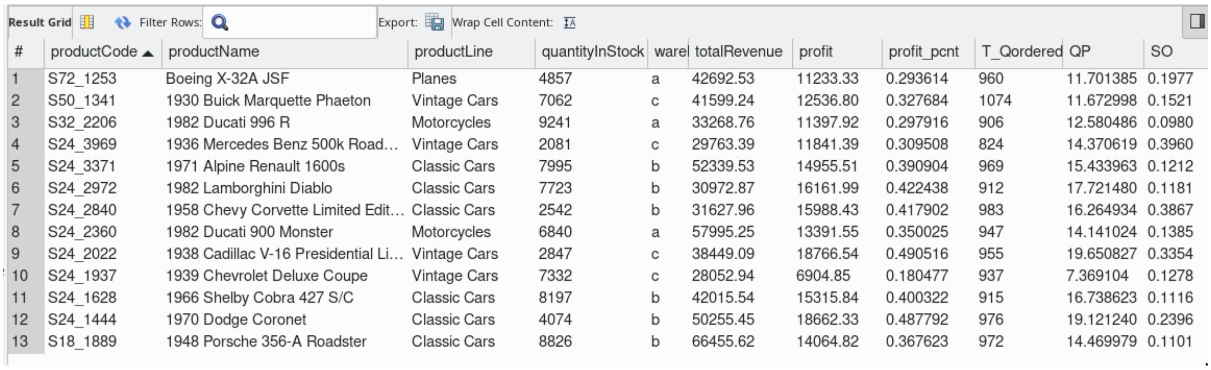


```
USE mnt2;
SET SQL_SAFE_UPDATES = 0;

UPDATE products p
JOIN (
    SELECT
        p.productCode,
        p.quantityInStock,
        od.totalQuantityOrdered AS tql,
        (od.totalQuantityOrdered / p.quantityInStock) AS SO,
        (od.profit * 100 / totalTotalProfit.totalTotalProfit) AS profit_pcnt
    FROM
        products p
    LEFT JOIN (
        SELECT
            odd.productCode,
            SUM(odd.quantityOrdered) AS totalQuantityOrdered,
            SUM((odd.priceEach - pp.buyPrice) * odd.quantityOrdered) AS profit
        FROM
            orderdetails odd
        JOIN products pp ON odd.productCode = pp.productCode
        GROUP BY odd.productCode
    ) od ON p.productCode = od.productCode
    CROSS JOIN (
        SELECT
```

```
            SUM((odd.priceEach - pp.buyPrice) * odd.quantityOrdered) AS
totalTotalProfit
        FROM
            orderdetails odd
        JOIN products pp ON odd.productCode = pp.productCode
    ) AS totalTotalProfit
) AS subquery ON p.productCode = subquery.productCode
SET p.quantityInStock = tql / 0.8
WHERE subquery.SO < 0.5 AND subquery.profit_pcnt < 0.5;

SET SQL_SAFE_UPDATES = 1;
```

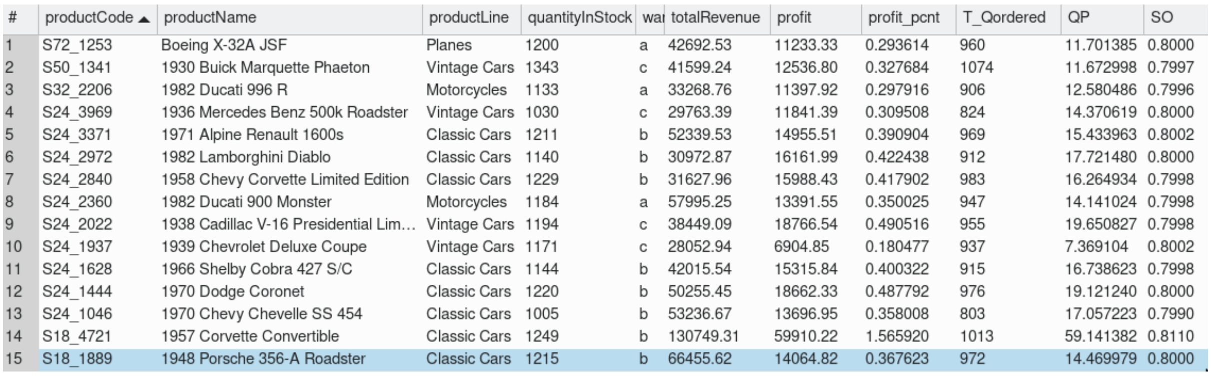
Checking the changes

BEFORE:



#	productCode ▲	productName	productLine	quantityInStock	warel	totalRevenue	profit	profit_pcnt	T_Qordered	QP	SO
1	S72_1253	Boeing X-32A JSF	Planes	4857	a	42692.53	11233.33	0.293614	960	11.701385	0.1977
2	S50_1341	1930 Buick Marquette Phaeton	Vintage Cars	7062	c	41599.24	12536.80	0.327684	1074	11.672998	0.1521
3	S32_2206	1982 Ducati 996 R	Motorcycles	9241	a	33268.76	11397.92	0.297916	906	12.580486	0.0980
4	S24_3969	1936 Mercedes Benz 500k Road…	Vintage Cars	2081	c	29763.39	11841.39	0.309508	824	14.370619	0.3960
5	S24_3371	1971 Alpine Renault 1600s	Classic Cars	7995	b	52339.53	14955.51	0.390904	969	15.433963	0.1212
6	S24_2972	1982 Lamborghini Diablo	Classic Cars	7723	b	30972.87	16161.99	0.422438	912	17.721480	0.1181
7	S24_2840	1958 Chevy Corvette Limited Edit…	Classic Cars	2542	b	31627.96	15988.43	0.417902	983	16.264934	0.3867
8	S24_2360	1982 Ducati 900 Monster	Motorcycles	6840	a	57995.25	13391.55	0.350025	947	14.141024	0.1385
9	S24_2022	1938 Cadillac V-16 Presidential Li…	Vintage Cars	2847	c	38449.09	18766.54	0.490516	955	19.650827	0.3354
10	S24_1937	1939 Chevrolet Deluxe Coupe	Vintage Cars	7332	c	28052.94	6904.85	0.180477	937	7.369104	0.1278
11	S24_1628	1966 Shelby Cobra 427 S/C	Classic Cars	8197	b	42015.54	15315.84	0.400322	915	16.738623	0.1116
12	S24_1444	1970 Dodge Coronet	Classic Cars	4074	b	50255.45	18662.33	0.487792	976	19.121240	0.2396
13	S18_1889	1948 Porsche 356-A Roadster	Classic Cars	8826	b	66455.62	14064.82	0.367623	972	14.469979	0.1101

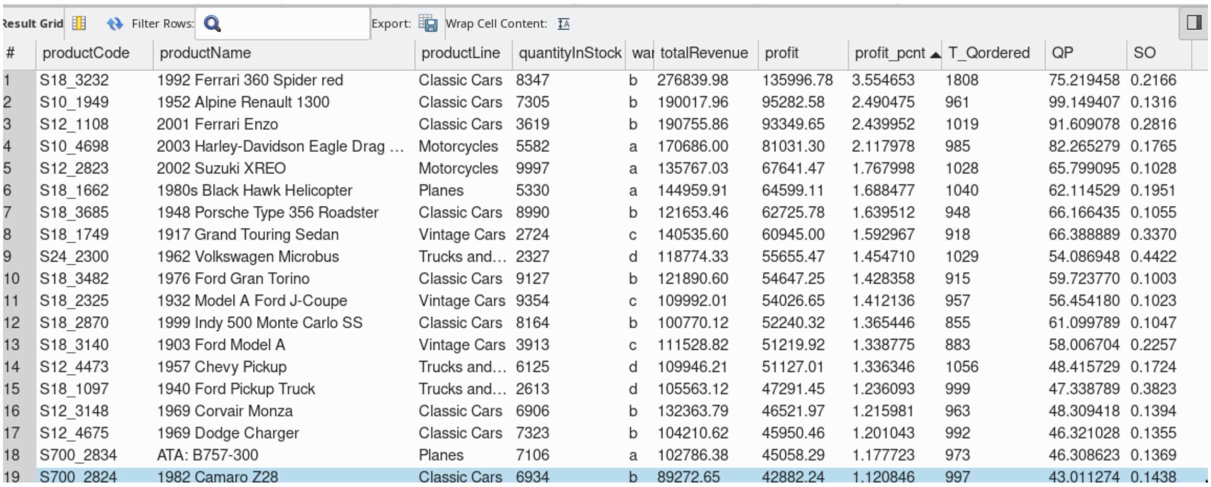
AFTER:



#	productCode ▲	productName	productLine	quantityInStock	wai	totalRevenue	profit	profit_pcnt	T_Qordered	QP	SO
1	S72_1253	Boeing X-32A JSF	Planes	1200	a	42692.53	11233.33	0.293614	960	11.701385	0.8000
2	S50_1341	1930 Buick Marquette Phaeton	Vintage Cars	1343	c	41599.24	12536.80	0.327684	1074	11.672998	0.7997
3	S32_2206	1982 Ducati 996 R	Motorcycles	1133	a	33268.76	11397.92	0.297916	906	12.580486	0.7996
4	S24_3969	1936 Mercedes Benz 500k Roadster	Vintage Cars	1030	c	29763.39	11841.39	0.309508	824	14.370619	0.8000
5	S24_3371	1971 Alpine Renault 1600s	Classic Cars	1211	b	52339.53	14955.51	0.390904	969	15.433963	0.8002
6	S24_2972	1982 Lamborghini Diablo	Classic Cars	1140	b	30972.87	16161.99	0.422438	912	17.721480	0.8000
7	S24_2840	1958 Chevy Corvette Limited Edition	Classic Cars	1229	b	31627.96	15988.43	0.417902	983	16.264934	0.7998
8	S24_2360	1982 Ducati 900 Monster	Motorcycles	1184	a	57995.25	13391.55	0.350025	947	14.141024	0.7998
9	S24_2022	1938 Cadillac V-16 Presidential Lim…	Vintage Cars	1194	c	38449.09	18766.54	0.490516	955	19.650827	0.7998
10	S24_1937	1939 Chevrolet Deluxe Coupe	Vintage Cars	1171	c	28052.94	6904.85	0.180477	937	7.369104	0.8002
11	S24_1628	1966 Shelby Cobra 427 S/C	Classic Cars	1144	b	42015.54	15315.84	0.400322	915	16.738623	0.7998
12	S24_1444	1970 Dodge Coronet	Classic Cars	1220	b	50255.45	18662.33	0.487792	976	19.121240	0.8000
13	S24_1046	1970 Chevy Chevelle SS 454	Classic Cars	1005	b	53236.67	13696.95	0.358008	803	17.057223	0.7990
14	S18_4721	1957 Corvette Convertible	Classic Cars	1249	b	130749.31	59910.22	1.565920	1013	59.141382	0.8110
15	S18_1889	1948 Porsche 356-A Roadster	Classic Cars	1215	b	66455.62	14064.82	0.367623	972	14.469979	0.8000

Pretty near hah? Yup it is pretty neat SO is optimal , stock is optimal of course both based on profit & QP

4. Reducing leftover stocks



#	productCode	productName	productLine	quantityInStock	wai	totalRevenue	profit	profit_pcnt ▲	T_Qordered	QP	SO
1	S18_3232	1992 Ferrari 360 Spider red	Classic Cars	8347	b	276839.98	135996.78	3.554653	1808	75.219458	0.2166
2	S10_1949	1952 Alpine Renault 1300	Classic Cars	7305	b	190017.96	95282.58	2.490475	961	99.149407	0.1316
3	S12_1108	2001 Ferrari Enzo	Classic Cars	3619	b	190755.86	93349.65	2.439952	1019	91.609078	0.2816
4	S10_4698	2003 Harley-Davidson Eagle Drag …	Motorcycles	5582	a	170686.00	81031.30	2.117978	985	82.265279	0.1765
5	S12_2823	2002 Suzuki XREO	Motorcycles	9997	a	135767.03	67641.47	1.767998	1028	65.799095	0.1028
6	S18_1662	1980s Black Hawk Helicopter	Planes	5330	a	144959.91	64599.11	1.688477	1040	62.114529	0.1951
7	S18_3685	1948 Porsche Type 356 Roadster	Classic Cars	8990	b	121653.46	62725.78	1.639512	948	66.166435	0.1055
8	S18_1749	1917 Grand Touring Sedan	Vintage Cars	2724	c	140535.60	60945.00	1.592967	918	66.388889	0.3370
9	S24_2300	1962 Volkswagen Microbus	Trucks and…	2327	d	118774.33	55655.47	1.454710	1029	54.086948	0.4422
10	S18_3482	1976 Ford Gran Torino	Classic Cars	9127	b	121890.60	54647.25	1.428358	915	59.723770	0.1003
11	S18_2325	1932 Model A Ford J-Coupe	Vintage Cars	9354	c	109992.01	54026.65	1.412136	957	56.454180	0.1023
12	S18_2870	1999 Indy 500 Monte Carlo SS	Classic Cars	8164	b	100770.12	52240.32	1.365446	855	61.099789	0.1047
13	S18_3140	1903 Ford Model A	Vintage Cars	3913	c	111528.82	51219.92	1.338775	883	58.006704	0.2257
14	S12_4473	1957 Chevy Pickup	Trucks and…	6125	d	109946.21	51127.01	1.336346	1056	48.415729	0.1724
15	S18_1097	1940 Ford Pickup Truck	Trucks and…	2613	d	105563.12	47291.45	1.236093	999	47.338789	0.3823
16	S12_3148	1969 Corvair Monza	Classic Cars	6906	b	132363.79	46521.97	1.215981	963	48.309418	0.1394
17	S12_4675	1969 Dodge Charger	Classic Cars	7323	b	104210.62	45950.46	1.201043	992	46.321028	0.1355
18	S700_2834	ATA: B757-300	Planes	7106	a	102786.38	45058.29	1.177723	973	46.308623	0.1369
19	S700_2824	1982 Camaro Z28	Classic Cars	6934	b	89272.65	42882.24	1.120846	997	43.011274	0.1438

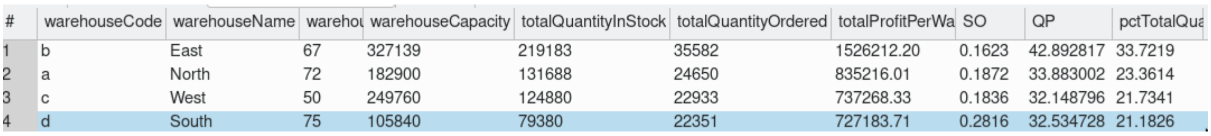
These are other 75 products which are overstocked , notice here we have high profit products also such as ferrari 360 and alpine renault , i have decided for profit>1.5% products we can use same SO method and setup all at 0.4 SO which will be enough overstock to manage if demand increases for theses products.

Now let’s decide how we reduce products where profit_prcnt < 1.5 and SO <0.3, these products can be classified as moderate profiting and at same time moderate overstocked. It will be better if we reduce their stock only via 20% because this are the products which are potentially profitable in future.
But i have decided to not change anything in high profit products

Reducing stocks of moderately profitable products [55 products]

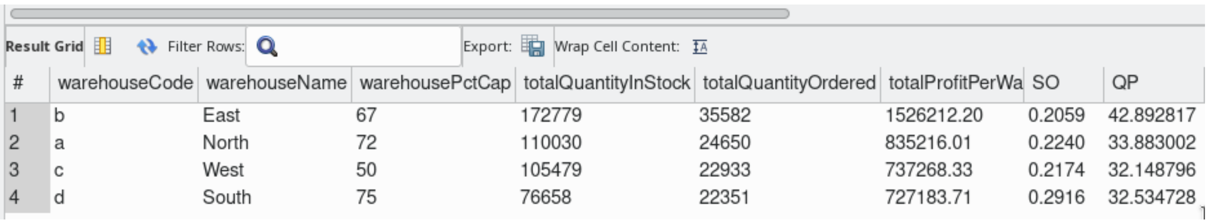
```sql
USE mnt2;
SET SQL_SAFE_UPDATES = 0;
UPDATE products p
JOIN (
    SELECT
        p.productCode,
        p.quantityInStock,
        od.totalQuantityOrdered AS tql,
        (od.totalQuantityOrdered / p.quantityInStock) AS SO,
        (od.profit * 100 / totalTotalProfit.totalTotalProfit) AS
profit_pcnt
    FROM
        products p
    LEFT JOIN (
        SELECT
            odd.productCode,
            SUM(odd.quantityOrdered) AS totalQuantityOrdered,
            SUM((odd.priceEach - pp.buyPrice) *
odd.quantityOrdered) AS profit
        FROM
            orderdetails odd
        JOIN products pp ON odd.productCode = pp.productCode
        GROUP BY odd.productCode
    ) od ON p.productCode = od.productCode
    CROSS JOIN (
        SELECT
            SUM((odd.priceEach - pp.buyPrice) *
odd.quantityOrdered) AS totalTotalProfit
        FROM
            orderdetails odd
        JOIN products pp ON odd.productCode = pp.productCode
    ) AS totalTotalProfit
) AS subquery ON p.productCode = subquery.productCode
SET p.quantityInStock = ROUND(p.quantityInStock * 0.8) -- Reduce
by 20%
WHERE subquery.SO < 0.3 AND subquery.profit_pcnt < 1.5;


SET SQL_SAFE_UPDATES = 1;
```

Before:



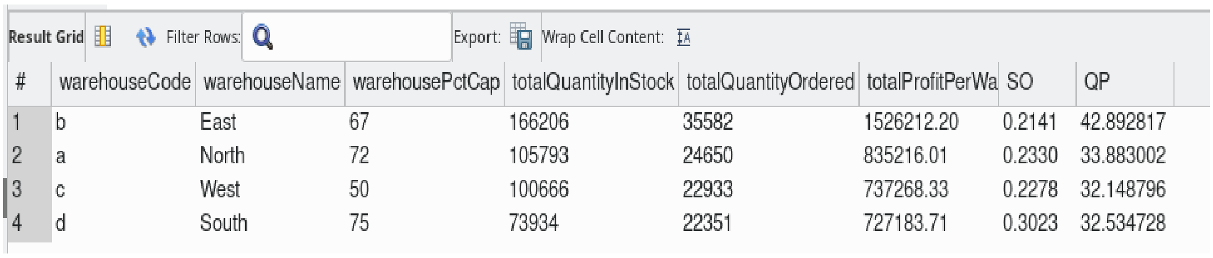
#	warehouseCode	warehouseName	warehou	warehouseCapacity	totalQuantityInStock	totalQuantityOrdered	totalProfitPerWa	SO	QP	pctTotalQua
1	b	East	67	327139	219183	35582	1526212.20	0.1623	42.892817	33.7219
2	a	North	72	182900	131688	24650	835216.01	0.1872	33.883002	23.3614
3	c	West	50	249760	124880	22933	737268.33	0.1836	32.148796	21.7341
4	d	South	75	105840	79380	22351	727183.71	0.2816	32.534728	21.1826

After removing just 5 % from moderate profit and moderate in demand products



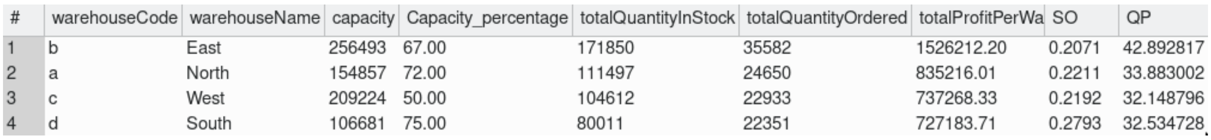
#	warehouseCode	warehouseName	warehousePctCap	totalQuantityInStock	totalQuantityOrdered	totalProfitPerWa	SO	QP
1	b	East	67	172779	35582	1526212.20	0.2059	42.892817
2	a	North	72	110030	24650	835216.01	0.2240	33.883002
3	c	West	50	105479	22933	737268.33	0.2174	32.148796
4	d	South	75	76658	22351	727183.71	0.2916	32.534728

After 10%:



#	warehouseCode	warehouseName	warehousePctCap	totalQuantityInStock	totalQuantityOrdered	totalProfitPerWa	SO	QP
1	b	East	67	166206	35582	1526212.20	0.2141	42.892817
2	a	North	72	105793	24650	835216.01	0.2330	33.883002
3	c	West	50	100666	22933	737268.33	0.2278	32.148796
4	d	South	75	73934	22351	727183.71	0.3023	32.534728

Filling up understocked items



#	warehouseCode	warehouseName	capacity	Capacity_percentage	totalQuantityInStock	totalQuantityOrdered	totalProfitPerWa	SO	QP
1	b	East	256493	67.00	171850	35582	1526212.20	0.2071	42.892817
2	a	North	154857	72.00	111497	24650	835216.01	0.2211	33.883002
3	c	West	209224	50.00	104612	22933	737268.33	0.2192	32.148796
4	d	South	106681	75.00	80011	22351	727183.71	0.2793	32.534728

```sql
-- FILLING UP UNDERSTOCKED ITEMS
USE mnt2;
SET SQL_SAFE_UPDATES = 0;

UPDATE products p
JOIN (
    SELECT
        p.productCode,
        p.quantityInStock,
        od.totalQuantityOrdered AS tql,
        (od.totalQuantityOrdered / p.quantityInStock) AS SO,
        (od.profit * 100 / totalTotalProfit.totalTotalProfit) AS profit_pcnt
    FROM
        products p
    LEFT JOIN (
        SELECT
```

```
            odd.productCode,
            SUM(odd.quantityOrdered) AS totalQuantityOrdered,
            SUM((odd.priceEach - pp.buyPrice) * odd.quantityOrdered) AS profit
        FROM
            orderdetails odd
        JOIN products pp ON odd.productCode = pp.productCode
        GROUP BY odd.productCode
    ) od ON p.productCode = od.productCode
    CROSS JOIN (
        SELECT
            SUM((odd.priceEach - pp.buyPrice) * odd.quantityOrdered) AS
totalTotalProfit
        FROM
            orderdetails odd
        JOIN products pp ON odd.productCode = pp.productCode
    ) AS totalTotalProfit
) AS subquery ON p.productCode = subquery.productCode
SET p.quantityInStock = tql / 0.4     -- increasing stocks
WHERE subquery.SO > 0.5 AND subquery.profit_pcnt > 0.6;

SET SQL_SAFE_UPDATES = 1;
```