Model car database analysis using MySQL workbench

Project Description: -

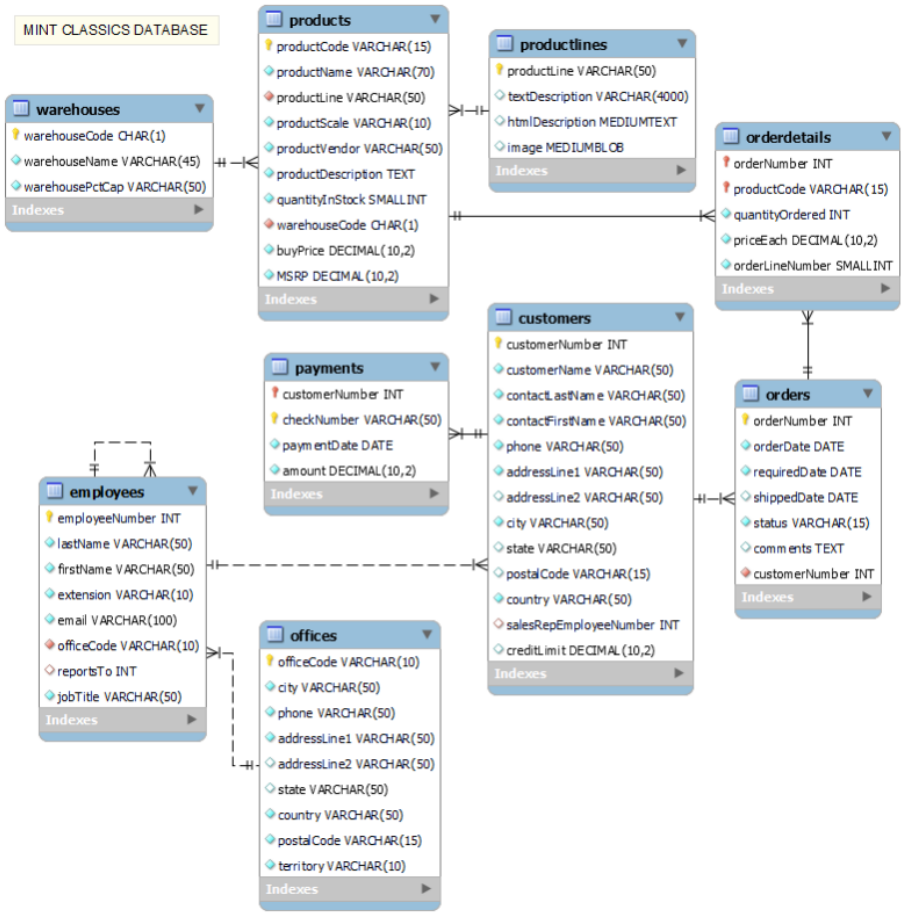
This project is dedicated to performing data analysis within the framework of the imaginary organization, Mint Classics. In this scenario, I will be taking on the role of a beginner data analyst, supporting the company in resolving challenges related to inventory management and storage facilities. The primary concern involves evaluating whether to shut down one of the current storage facilities.

Tools Used: - MySQL Workbench 8.0 CE

Tasks: -

1. Importing the classic car Model Database.
2. Understanding the Database and ERD (Entity Relationship Diagram).
3. Understanding the business issue.
4. Preprocessing the tables for analysis.
5. Formulating the queries for in-depth analysis.
6. Crafting conclusion and recommendations.

Entity Relationship Diagram of Mint classic Database: -



Approach: -

Initiated the analysis by examining fundamental metrics related to the warehouses, followed by an assessment of each warehouse's performance metrics, including total sales, average sales amounts, and the percentage of successful shipment rates. The objective was to identify warehouses that may be approaching closure. Subsequently, delved into evaluating the influence of products on warehouse performance and investigated whether employee performance and customers has an impact on warehouse efficiency.

Examining fundamental metrics of the warehouses: -

- What is the total quantity of product in each inventory?
  Query used: -

```
SELECT w.warehouseCode, w.warehouseName, sum(p.quantityInStock) as Total_Inventory
from warehouses w join products p on w.warehouseCode = p.warehouseCode
group by w.warehouseCode, w.warehouseName
order by Total_Inventory desc;
```

  This query allows you to observe the overall inventory for each warehouse.

  Result: - The East warehouse boasts the highest product count, while the South warehouse has the lowest.

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

- What is product category count and their product line percentage?
  Query used: -

```
select productLine, warehouseCode, count(productLine) as ProductCatCount,
cast(count(productLine)*100/ (select count(productLine) from products)as decimal(10,2)) as CategoryPercentage
from products
group by productLine, warehouseCode
order by CategoryPercentage desc;
```

  This query provides information on the product count within various product lines and the respective percentage distribution of products in each line.

  Result: - Classic Cars product line constitutes 34.55% of the total products, whereas the Trains product line comprises only 2.73% of the overall product distribution.

productLine	warehouseCode	ProductCatCount	CategoryPercentage
Classic Cars	b	38	34.55
Vintage Cars	c	24	21.82
Motorcycles	a	13	11.82
Planes	a	12	10.91
Trucks and Buses	d	11	10.00
Ships	d	9	8.18
Trains	d	3	2.73

Conducting an evaluation of warehouse performance: -

- What is the overall quantity of products shipped from each warehouse?
  Query used: -

```
Select warehouseName, warehouseCode, TotalquantityShipped
from (Select w.warehouseName, w.warehouseCode, sum(od.quantityOrdered) as TotalquantityShipped
      from orderdetails od
      join products p on od.productCode = p.productCode
      join warehouses w on p.warehouseCode = w.warehouseCode
Group by warehouseName, warehouseCode) as TotalquantityOrdered
order by TotalquantityShipped desc;
```

  This query provides information about the number of products shipped from each warehouse.

  Result: - Most of the products were shipped from East warehouse.

warehouseName	warehouseCode	TotalquantityShipped
East	b	35582
North	a	24650
West	c	22933
South	d	22351

- What's the total sales amount and average sales amount of each warehouse?
  Query used: -

```
Select w.warehouseName, w.warehouseCode ,
       sum(CASE WHEN o.status IN ("Shipped","Resolved") THEN od.quantityOrdered * od.priceEach ELSE 0 END) as TotalSales,
       Cast(sum(CASE WHEN o.status IN ("Shipped","Resolved") THEN od.quantityOrdered * od.priceEach ELSE 0 END)
       / sum(CASE WHEN o.status IN ("Shipped","Resolved") THEN od.quantityOrdered  ELSE 0 END)as decimal(10,2))  as AverageSales
       from warehouses w
       join products p on p.warehouseCode = w.warehouseCode
       join orderdetails od on p.productCode = od.productCode
       join orders o on od.orderNumber = o.orderNumber
Group by warehouseName, warehouseCode
order by AverageSales desc;
```

  This query provides insights into the total and average sales for each warehouse.
  Note: - This query considers only those orders which were successfully shipped.

  Result: - The East warehouse stands out with the highest overall and average sales, whereas the West warehouse exhibits the lowest figures.

warehouseName	warehouseCode	TotalSales	AverageSales
East	b	3648921.72	108.46
North	a	1951643.26	84.11
South	d	1729651.46	83.87
West	c	1669114.08	78.24

- Are warehouses making enough profit through sales?
  Query used: -

```
Select warehouseName, warehouseCode,
cast(((TotalSalesAmount - TotalBuyAmount)/TotalSalesAmount)*100 as decimal(10,2)) as profit_margin
from (
Select w.warehouseName, w.warehouseCode ,
       sum( od.quantityOrdered * od.priceEach) as TotalSalesAmount,
       sum(od.quantityOrdered * p.buyPrice) as TotalBuyAmount
       from warehouses w
       join products p on p.warehouseCode = w.warehouseCode
       join orderdetails od on p.productCode = od.productCode
       join orders o on od.orderNumber = o.orderNumber
       where o.status = "Shipped" or o.status = "Resolved"
Group by warehouseName, warehouseCode) profit
order by profit_margin desc;
```

  This query offers insights into the profit margin generated by each warehouse through product sales.

  Result: - The West warehouse boasts the highest profit margin, while the South warehouse has the lowest.

warehouseName	warehouseCode	profit_margin
West	c	41.02
North	a	40.38
East	b	39.65
South	d	38.83

- Are the products effectively shipped from the warehouse?
  Query used: -

```
select w.warehouseName,w.warehouseCode,
CAST((Sum( CASE WHEN o.status = "Shipped" or o.status = "Resolved" THEN 1 ElSE 0 END)
/COUNT(o.orderNumber))  * 100 as decimal(10,2)) as SuccessfulShipmentPer
FROM warehouses w
join products p on w.warehouseCode = p.warehouseCode
join orderdetails od on p.productCode = od.productCode
join orders o on od.orderNumber = o.orderNumber
group by w.warehouseCode, w.warehouseName
order by SuccessfulShipmentPer desc;
```

This query provides insights into the success rate of product shipment for each warehouse.
Note: - This query considers only those orders which were successfully shipped or the order ran into a problem and was later resolved.

Result: - The East warehouse stands out with the highest successful shipment percentage, whereas the South warehouse exhibits the lowest percentage.

warehouseName	warehouseCode	SuccessfulShipmentPer
East	b	95.05
North	a	94.68
West	c	93.46
South	d	92.43

Result: - Upon analysis, it is found that warehouses West and South exhibit the least performance in terms of sales, shipments, etc.

Examining the Impact of products on Warehouse performance: -

- Was there any drastic change in products demand across the years?
  Query used: -

```
SELECT p.productCode, p.productName,
    SUM(case when YEAR(o.orderDate) = 2003 then od.quantityOrdered else 0 end) as quantityOrderedIn_2003,
    cast(case when COUNT(case when YEAR(o.orderDate) = 2003 then od.priceEach else null end) > 0
         then AVG(case when YEAR(o.orderDate) = 2003 then od.priceEach end) else 0 end as decimal(10,2)) as AveragePrice_2003,

    SUM(case when YEAR(o.orderDate) = 2004 then od.quantityOrdered else 0 end) as quantityOrderedIn_2004,
    cast(case when COUNT(case when YEAR(o.orderDate) = 2004 then od.priceEach else null end) > 0
         then AVG(case when YEAR(o.orderDate) = 2004 then od.priceEach end) else 0 end as decimal(10,2)) as AveragePrice_2004,

    SUM(case when YEAR(o.orderDate) = 2005 then od.quantityOrdered else 0 end) as quantityOrderedIn_2005,
    cast(case when COUNT(case when YEAR(o.orderDate) = 2005 then od.priceEach else null end) > 0
         then AVG(case when YEAR(o.orderDate) = 2005 then od.priceEach end) else 0 end as decimal(10,2)) as AveragePrice_2005
FROM products p
INNER JOIN
    orderdetails od ON p.productCode = od.productCode
INNER JOIN
    orders o ON od.orderNumber = o.orderNumber
GROUP BY p.productCode, p.productName
ORDER BY p.productCode;
```

This query provides insights of the products demand across the years.

Result: - The demand for the overall product experienced a substantial shift in the year 2005 compared to other years.

productCode	productName	quantityOrderedIn_2003	AveragePrice_2003	quantityOrderedIn_2004	AveragePrice_2004	quantityOrderedIn_2005	AveragePrice_2005
S10_1678	1969 Harley Davidson Ultimate Chopper	334	85.07	530	87.157143	193	79.814000
S10_1949	1952 Alpine Renault 1300	342	199.30	445	196.330769	174	195.870000
S10_2016	1996 Moto Guzzi 1100i	300	108.10	469	109.848571	230	113.944000
S10_4698	2003 Harley-Davidson Eagle Drag Bike	314	171.50	462	176.083077	209	165.253333
S10_4757	1972 Alfa Romeo GTA	349	123.31	463	124.828571	218	124.304000

- Are there any product lines showing exceptional performance or underperforming?
  Query used: -

```
Select  p.warehouseCode, pl.productLine,
      sum(CASE WHEN o.status IN ("Shipped","Resolved") THEN od.quantityOrdered * od.priceEach ELSE 0 END) as TotalSales,
      Cast(sum(CASE WHEN o.status IN ("Shipped","Resolved")  THEN od.quantityOrdered * od.priceEach ELSE 0 END)
      / sum(CASE WHEN o.status IN ("Shipped","Resolved")  THEN od.quantityOrdered ELSE 0 END) as decimal(10,2))  as AverageSales,
      cast(sum(CASE WHEN o.status IN ("Shipped","Resolved") THEN od.quantityOrdered  ELSE 0 END)as decimal(10,2)) as totalQuantityOrdered
      from productlines pl
      join products p on pl.productLine = p.productLine
      join orderdetails od on p.productCode = od.productCode
      join orders o on od.orderNumber = o.orderNumber
Group by p.warehouseCode, pl.productLine;
```

This query gives insights about the total sales and average sales of each product line.

Result: - The Classic Cars product line leads in both total and average sales, whereas the Trains product line appears to underperform with the lowest overall and average sales.

warehouseCode	productLine	TotalSales	AverageSales	totalQuantityOrdered
b	Classic Cars	3648921.72	108.46	33643.00
a	Motorcycles	1084927.13	87.32	12425.00
a	Planes	866716.13	80.41	10779.00
d	Ships	586053.82	77.78	7535.00
d	Trains	175030.77	66.02	2651.00
d	Trucks and Buses	968566.87	92.81	10436.00
c	Vintage Cars	1669114.08	78.24	21332.00

- What product line yields the highest profit for the company?
  Query used: -

```
select warehouseCode, productLine, cast(((TotalSales - TotalSales1)/TotalSales)*100 as decimal(10,2)) as profitMargin
from(
Select  p.warehouseCode, pl.productLine,
      sum( od.quantityOrdered * od.priceEach ) as TotalSales,
      sum(od.quantityOrdered * p.buyPrice) as TotalSales1
      from productlines pl
      join products p on pl.productLine = p.productLine
      join orderdetails od on p.productCode = od.productCode
      join orders o on od.orderNumber = o.orderNumber
      WHERE o.status = 'Shipped' or o.status = 'Resolved'
Group by p.warehouseCode, pl.productLine) as profit
order by profitMargin desc;
```

  This query offers insights into the profit margin produced by each product line.

  Result: -

warehouseCode	productLine	profitMargin
a	Motorcycles	41.93
c	Vintage Cars	41.02
b	Classic Cars	39.65
d	Ships	39.37
d	Trucks and Buses	39.19
a	Planes	38.45
d	Trains	35.07

- Was there any product that was in demand but was out of stock?
  Query used: -

```
select productCode, productName,  productLine, warehouseCode, quantityInStock, quantityOrdered,
(quantityInStock - quantityOrdered) as InventoryShortage
from (
Select p.productName, p.quantityInStock,od.productCode,p.warehouseCode,
sum(od.quantityOrdered) as quantityOrdered, pl.productLine
from productlines pl
      join products p on pl.productLine = p.productLine
      join orderdetails od on p.productCode = od.productCode
group by p.productName, p.quantityInStock, od.productCode , pl.productLine) AS InventoryShortage
where (quantityInStock - quantityOrdered) < 0
order by quantityOrdered asc ;
```

  This query fetches details about products that experienced demand but were unavailable in stock.

  Result: - The majority of products that faced high demand but were out of stock were from the motorcycle product line.

productCode	productName	productLine	warehouseCode	quantityInStock	quantityOrdered	InventoryShortage
S12_1099	1968 Ford Mustang	Classic Cars	b	68	933	-865
S24_2000	1960 BSA Gold Star DBD34	Motorcycles	a	15	1015	-1000
S32_1374	1997 BMW F650 ST	Motorcycles	a	178	1014	-836
S50_4713	2002 Yamaha YZR M1	Motorcycles	a	600	992	-392
S700_3167	F/A 18 Hornet 1/72	Planes	a	551	1047	-496
S72_3212	Pont Yacht	Ships	d	414	958	-544
S700_1938	The Mayflower	Ships	d	737	898	-161
S32_3522	1996 Peterbilt 379 Stake Bed with Outrigger	Trucks and Buses	d	814	988	-174
S32_4289	1928 Ford Phaeton Deluxe	Vintage Cars	c	136	972	-836
S18_2795	1928 Mercedes-Benz SSK	Vintage Cars	c	548	880	-332
S18_2248	1911 Ford Town Car	Vintage Cars	c	540	832	-292

- Are there products with high inventory but low sales?

  Query used: -

```
select warehouseCode,warehouseName,productName, productLine,quantityInStock, quantityOrdered,
cast((quantityOrdered / (quantityInStock))*100 as decimal(10,2)) as SellThroughRatePer
from
( Select w.warehouseName,  W.warehouseCode, p.productName,p.productLine,
  p.quantityInStock, p.productCode,  sum(od.quantityOrdered) as quantityOrdered
  from
  warehouses w
  join products p on w.warehouseCode = p.warehouseCode
  join orderdetails od on p.productCode = od.productCode
  group by p.productCode ) AS sells
where quantityInStock > quantityOrdered
order by SellThroughRatePer desc;
```

  The sell-through rate is a metric used to measure the percentage of products that were sold compared to the total available inventory over a specific period.

$$\text{Sell-Through Rate } (\%) = \left(\frac{\text{Number of Units Sold}}{\text{Initial Inventory}}\right) \times 100$$

  This query provides insights into product performance by computing their sell-through rate.

  Result: - Trucks and buses products have the highest sells through rate, followed by planes and classic cars.

warehouseCode	warehouseName	productName	productLine	quantityInStock	quantityOrdered	SellThroughRatePer
d	South	Diamond T620 Semi-Skirted Tanker	Trucks and Buses	1016	979	96.36
a	North	P-51-D Mustang	Planes	992	917	92.44
b	East	1969 Ford Falcon	Classic Cars	1049	965	91.99
b	East	1957 Corvette Convertible	Classic Cars	1249	1013	81.10
b	East	1970 Chevy Chevelle SS 454	Classic Cars	1005	803	79.90
d	South	1958 Setra Bus	Trucks and Buses	1579	972	61.56
b	East	1952 Citroen-15CV	Classic Cars	1452	873	60.12
d	South	1962 City of Detroit Streetcar	Trains	1645	966	58.72
d	South	18th century schooner	Ships	1898	1011	53.27
b	East	1969 Dodge Super Bee	Classic Cars	1917	974	50.81
d	South	1926 Ford Fire Engine	Trucks and Buses	2018	998	49.45
d	South	The Schooner Bluenose	Ships	1897	934	49.24
d	South	The Titanic	Ships	1956	952	48.67

  To assess the product performance from each warehouse, the same query can be employed with a simple modification in the WHERE clause.

  Here is an example to see the top 5 products of warehouse “a”

  Query: -

```
select warehouseCode,warehouseName,productName, productLine,quantityInStock, quantityOrdered,
cast((quantityOrdered / (quantityInStock))*100 as decimal(10,2)) as SellThroughRatePer
from
( Select w.warehouseName,  W.warehouseCode, p.productName,p.productLine,
  p.quantityInStock, p.productCode,  sum(od.quantityOrdered) as quantityOrdered
  from
  warehouses w
  join products p on w.warehouseCode = p.warehouseCode
  join orderdetails od on p.productCode = od.productCode
  group by p.productCode ) AS sells
where warehouseCode = "a" AND quantityInStock > quantityOrdered
order by SellThroughRatePer desc
limit 5;
```

  Result: -

warehouseCode	warehouseName	productName	productLine	quantityInStock	quantityOrdered	SellThroughRatePer
a	North	P-51-D Mustang	Planes	992	917	92.44
a	North	1900s Vintage Tri-Plane	Planes	2756	1009	36.61
a	North	1974 Ducati 350 Mk3 Desmo	Motorcycles	3341	898	26.88
a	North	1928 British Royal Navy Airplane	Planes	3627	972	26.80
a	North	1936 Harley Davidson El Knucklehead	Motorcycles	4357	945	21.69

Analysing customers: -

- Who are the customers making the most significant contributions to sales?
  Query used: -

```
select c.customerName, count(distinct o.orderNumber) as NumberOfOrders, sum(od.quantityOrdered) as TotalQuantityBought,
sum(od.quantityOrdered*od.priceEach) as TotalPurchaseAmount
from customers c
join orders o on c.customerNumber = o.customerNumber
join orderdetails od on o.orderNumber = od.orderNumber
group by c.customerName
order by TotalPurchaseAmount desc;
```

  This query gives insights about the customers overall purchase amount.

  Result: -

customerName	NumberOfOrders	TotalQuantityBought	TotalPurchaseAmount
Euro+ Shopping Channel	26	9327	820689.54
Mini Gifts Distributors Ltd.	17	6366	591827.34
Australian Collectors, Co.	5	1926	180585.07
Muscle Machine Inc	4	1775	177913.95
La Rochelle Gifts	4	1832	158573.12
Dragon Souveniers, Ltd.	5	1524	156251.03
Down Under Souveniers, Inc	5	1691	154622.08
Land of Toys Inc.	4	1631	149085.15
AV Stores, Co.	3	1778	148410.09
The Sharp Gifts Warehouse	4	1656	143536.27

- What is the purchasing trend among customers? Are there customers who did not place any orders within the specified timeframe?
  Query used: -

```
select c.customerNumber, c.customerName,
sum(CASE WHEN YEAR(o.orderDate) = 2003 THEN 1 ELSE 0 END) as quantityOrderedIn_2003,
sum(CASE WHEN YEAR(o.orderDate) = 2004 THEN 1 ELSE 0 END) as quantityOrderedIn_2004,
sum(CASE WHEN YEAR(o.orderDate) = 2005 THEN 1 ELSE 0 END) as quantityOrderedIn_2005
from
customers c
join orders o on c.customerNumber = o.customerNumber
group by c.customerNumber, c.customerName
order by c.customerNumber ;
```

  This query gives information about the no. of order customers places each year.

  Result: - Among 122 customers, only 98 have made purchases within the designated timeframe, while 24 customers did not place any orders.

customerNumber	customerName	quantityOrderedIn_2003	quantityOrderedIn_2004	quantityOrderedIn_2005
103	Atelier graphique	1	2	0
112	Signal Gift Stores	1	2	0
114	Australian Collectors, Co.	2	3	0
119	La Rochelle Gifts	0	2	2
121	Baane Mini Imports	2	2	0
124	Mini Gifts Distributors Ltd.	4	6	7
128	Blauer See Auto, Co.	2	2	0
129	Mini Wheels Co.	2	1	0
131	Land of Toys Inc.	1	3	0
141	Euro+ Shopping Channel	8	9	9
144	Volvo Model Replicas, Co	1	3	0

```
select c.customerNumber, c.customerName,
sum(CASE WHEN YEAR(o.orderDate) = 2003 THEN 1 ELSE 0 END) as quantityOrderedIn_2003,
sum(CASE WHEN YEAR(o.orderDate) = 2004 THEN 1 ELSE 0 END) as quantityOrderedIn_2004,
sum(CASE WHEN YEAR(o.orderDate) = 2005 THEN 1 ELSE 0 END) as quantityOrderedIn_2005
from
customers c
join orders o on c.customerNumber = o.customerNumber
group by c.customerNumber, c.customerName
having quantityOrderedIn_2005 = 0
order by c.customerNumber ;
```

This query furnishes details about customers who were active purchasers until 2004 but did not place any orders further.

Result: - Among those 98 customers, 54 became inactive as they refrained from placing any orders in 2005.

customerNumber	customerName	quantityOrderedIn_2003	quantityOrderedIn_2004	quantityOrderedIn_2005
103	Atelier graphique	1	2	0
112	Signal Gift Stores	1	2	0
114	Australian Collectors, Co.	2	3	0
121	Baane Mini Imports	2	2	0
128	Blauer See Auto, Co.	2	2	0
129	Mini Wheels Co.	2	1	0

- Do customers have outstanding balances?

Query used: -

```
select c.customerNumber,c.customerName, c.creditLimit,
sum(od.quantityOrdered*od.priceEach) as TotalPurchaseAmount,  totalPayments.totalAmount as TotalAmountPaid,
sum(od.quantityOrdered*od.priceEach) - totalPayments.totalAmount  as DueAmount
from customers c
join orders o on c.customerNumber = o.customerNumber
join orderdetails od on o.orderNumber = od.orderNumber
join (
    select p.customerNumber, SUM(p.amount) AS totalAmount
    from payments p
    group by p.customerNumber
) totalPayments ON c.customerNumber = totalPayments.customerNumber
group by c.customerNumber, c.customerName, c.creditLimit
having DueAmount > 0
order by  DueAmount desc;
```

This query offers information regarding customers with outstanding balance.

Result: - Among those 98 customers, 21 have an outstanding balance.

customerNumber	customerName	creditLimit	TotalPurchaseAmount	TotalAmountPaid	DueAmount
141	Euro+ Shopping Channel	227600.00	820689.54	715738.98	104950.56
450	The Sharp Gifts Warehouse	77600.00	143536.27	59551.38	83984.89
382	Salzburg Collectables	71700.00	137480.07	85060.00	52420.07
362	Gifts4AllAges.com	41900.00	84340.32	33533.47	50806.85
201	UK Collectables, Ltd.	92700.00	106610.72	61167.18	45443.54

- Are there customers with credit issues that need to be addressed?

Query used: -

```
select c.customerNumber,c.customerName, c.creditLimit,
sum(od.quantityOrdered*od.priceEach) as TotalPurchaseAmount,  totalPayments.totalAmount as TotalAmountPaid,
c.creditLimit - (sum(od.quantityOrdered*od.priceEach) - totalPayments.totalAmount)  as CreditLimitCrossed
from customers c
join orders o on c.customerNumber = o.customerNumber
join orderdetails od on o.orderNumber = od.orderNumber
join (
    select p.customerNumber, SUM(p.amount) AS totalAmount
    from payments p
    group by p.customerNumber
) totalPayments ON c.customerNumber = totalPayments.customerNumber
group by c.customerNumber, c.customerName, c.creditLimit
having c.creditLimit - (sum(od.quantityOrdered*od.priceEach) - totalPayments.totalAmount) < 0
order by  CreditLimitCrossed desc;
```

This query provides insights into customers with outstanding balances who have surpassed their credit limits.

Result: - Among those 21 customers, 3 customers with outstanding balances have surpassed their credit limits.

customerNumber	customerName	creditLimit	TotalPurchaseAmount	TotalAmountPaid	CreditLimitCrossed
328	Tekni Collectables Inc.	43000.00	81806.55	38281.51	-525.04
450	The Sharp Gifts Warehouse	77600.00	143536.27	59551.38	-6384.89
362	Gifts4AllAges.com	41900.00	84340.32	33533.47	-8906.85

Analysing employee's performance: -

- How are employees performing based on their overall sales?
  Query used: -

```
Select employeeNumber, fullName, jobTitle,
      coalesce(Sum( CASE WHEN status = "Shipped" or status = "Resolved" THEN priceEach * quantityOrdered ELSE 0 END),0) as TotalSalesAmount
from (Select e.employeeNumber, concat(e.firstName, ' ' , e.lastName) as fullName,
      e.jobTitle,od.priceEach, od.quantityOrdered,o.status
      from employees e
      left join customers c on e.employeeNumber = c.salesRepEmployeeNumber
      left join orders o on c.customerNumber = o.customerNumber
      left join orderdetails od on o.orderNumber = od.orderNumber
      left join products p on  od.productCode = p.productCode
      left join warehouses w on  p.warehouseCode = w.warehouseCode ) as totalSales
      group by employeeNumber
      order by jobTitle, TotalSalesAmount desc;
```

  This query helps to obtain the list of employees and their overall sales.

  Result: - Gerard Hernandez, the sales representative, secured the highest sales amount. Meanwhile, two other sales representatives did not make any sales.

employeeNumber	fullName	jobTitle	TotalSalesAmount
1002	Diane Murphy	President	0.00
1102	Gerard Bondur	Sale Manager (EM...	0.00
1088	William Patterson	Sales Manager (A...	0.00
1143	Anthony Bow	Sales Manager (NA)	0.00
1370	Gerard Hernandez	Sales Rep	1112003.81
1165	Leslie Jennings	Sales Rep	1021661.89
1401	Pamela Castillo	Sales Rep	837984.20
1501	Larry Bott	Sales Rep	686653.25
1504	Barry Jones	Sales Rep	637672.65
1323	George Vanauf	Sales Rep	584406.80
1337	Loui Bondur	Sales Rep	569485.75
1612	Peter Marsh	Sales Rep	523860.78
1611	Andy Fixter	Sales Rep	509385.82
1286	Foon Yue Tseng	Sales Rep	488212.67

- Are employees effectively able secure deals with clients?
  Query used: -

```
Select employeeNumber, fullName, jobTitle, NumberOfDeals, NumberofSuccessfulDeals,
       CAST((NumberofSuccessfulDeals / NumberOfDeals)  * 100 as decimal(10,2)) as DealEfficiency
from
    (Select e.employeeNumber, concat(e.firstName, ' ' , e.lastName) as fullName, e.jobTitle, COUNT(*) as NumberOfDeals,
     sum( CASE WHEN o.status = "Shipped" OR o.status = "Resolved" THEN 1 ElSE 0 END) as NumberofSuccessfulDeals
       from orders o
       left join customers c on o.customerNumber = c.customerNumber
       left join employees e on c.salesRepEmployeeNumber = e.employeeNumber
       group by employeeNumber) as totalSales
       order by DealEfficiency desc;
```

  This query offers a glimpse into the success rate of employee deals, considering only orders that were either successfully shipped or encountered an issue that was subsequently resolved.

  Result: - Nine employees successfully secured deals with clients, but their orders encountered issues, impacting their deal efficiency.

employeeNumber	fullName	jobTitle	NumberOfDeals	NumberofSuccessfulDeals	DealEfficiency
1286	Foon Yue Tseng	Sales Rep	17	17	100.00
1621	Mami Nishi	Sales Rep	16	16	100.00
1337	Loui Bondur	Sales Rep	20	20	100.00
1702	Martin Gerard	Sales Rep	12	12	100.00
1166	Leslie Thompson	Sales Rep	14	14	100.00
1188	Julie Firrelli	Sales Rep	14	14	100.00
1501	Larry Bott	Sales Rep	22	21	95.45
1165	Leslie Jennings	Sales Rep	34	32	94.12

Should Mint Classics consider the closure of one of their storage facilities?

I strongly **oppose the closure** of any warehouses. Upon thorough examination of the Mint Classic database, it's evident that the East warehouse outperforms others, particularly in sales, shipment, and efficiency metrics. Notably, products from the East warehouse command an average sales price of $108.46, surpassing the average prices of products from other warehouses, which are below $100. Moreover, the demand for Classic Cars products, housed in the East inventory, is notably high compared to other lines, except for Vintage Cars. However, the profit margins generated by each warehouse and product line, except for Trains, are nearly identical which is approximately 40%.

This analysis underscores the significant influence of various factors on warehouse performance, with notable contributors being product cost and demand. Product demand is primarily driven by customer behavior, whereas product pricing is controlled by the company. Therefore, it can be inferred that optimizing warehouse performance necessitates a holistic consideration of these factors.

Given these insights, closing any storage facility could potentially disrupt operations and hinder overall performance. Therefore, it is advisable for Mint Classics to maintain its existing storage infrastructure, leveraging the strengths of warehouses to drive continued success and profitability.

Conclusions: -

- There are warehouses with low inventory, especially the West warehouse as it is utilized only 50%.
- The demand for products remained consistent in 2003 and 2004; however, there was a notable decrease in the number of orders placed in 2005. Surprisingly, this change occurred despite the average price of products remaining relatively constant throughout the specified time frame. Further investigation into this phenomenon is warranted, as it influences the sales performance of the company.
- The profit margin of all product lines is approximately 40%, except for Trains, which stands at 35.07%. This variance requires further investigation into pricing, as it influences the sales performance of the warehouse.
- 11 products that experienced high demand but are currently out of stock. Urgent attention is required for products currently out of stock, as their unavailability significantly influences the company's overall sales performance.
- Several products exhibit a sales-through rate below 50%. It is advisable to replace items with a high inventory but low sales with those boasting a high sales-through rate.
- Out of the 122 customers in total, only 98 made purchases. Among these 98 customers, 54 did not place any orders in the year 2005. This issue needs to be addressed as it signifies a loss of customers for the company.
- Among the 122 customers, 21 have outstanding balances, with 3 of them exceeding their credit limits. This situation poses a potential risk of loss for the company and requires immediate attention.
- Among the 17 sales representatives, 9 successfully secured deals with clients, but encountered issues that affected their performance. Additionally, two sales representatives with employee numbers 1619 and 1625 did not contribute to any sales. Investigation is needed to determine why the shipments were not delivered to customers if the orders were not cancelled by the customers, and why these two employees did not contribute to sales.

Recommendations: -

- It is advisable to utilize at least 80% of the warehouse capacity by stocking products that are in high demand.
- Despite the average price of products remaining relatively constant throughout the specified time frame, there was a notable decrease in the number of orders placed in 2005 as 54 customers didn’t place any order. The company should focus more on strategies to attract customers.
- Reducing the profit margin to 20%-30% on products with low Sales-through rate could facilitate their clearance and attract more customers, potentially boosting sales.
- Inventory management requires enhancement, as 11 products in high demand were not shipped due to insufficient inventory levels.
- The company should review its credit policy and enhance its credit risk management. Currently, there are 21 customers with outstanding balances, and among them, 3 customers have exceeded their credit limits. Failure to address this issue could result in financial losses for the company.
- The company should also prioritize evaluating employee performance, as it significantly impacts the company's financial performance.