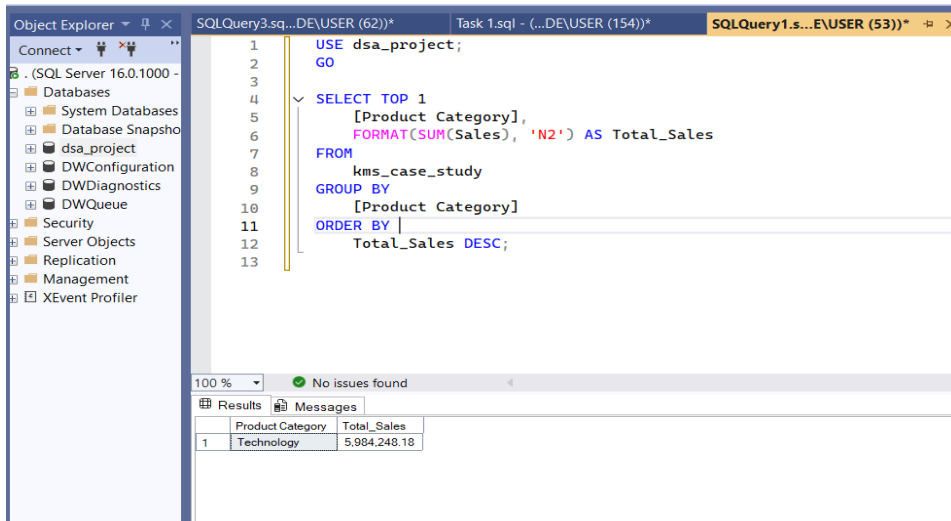


Kultra Mega Stores Inventory

Case Scenario 1

Task 1: Which product category had the highest sales?

```
SELECT TOP 1
    [Product Category],
    FORMAT(SUM(Sales), 'N2') AS Total_Sales
FROM
    kms_case_study
GROUP BY
    [Product Category]
ORDER BY
    Total_Sales DESC;
```



The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left shows the database structure. The SQL window displays the following query:

```
USE dsa_project;
GO

SELECT TOP 1
    [Product Category],
    FORMAT(SUM(Sales), 'N2') AS Total_Sales
FROM
    kms_case_study
GROUP BY
    [Product Category]
ORDER BY
    Total_Sales DESC;
```

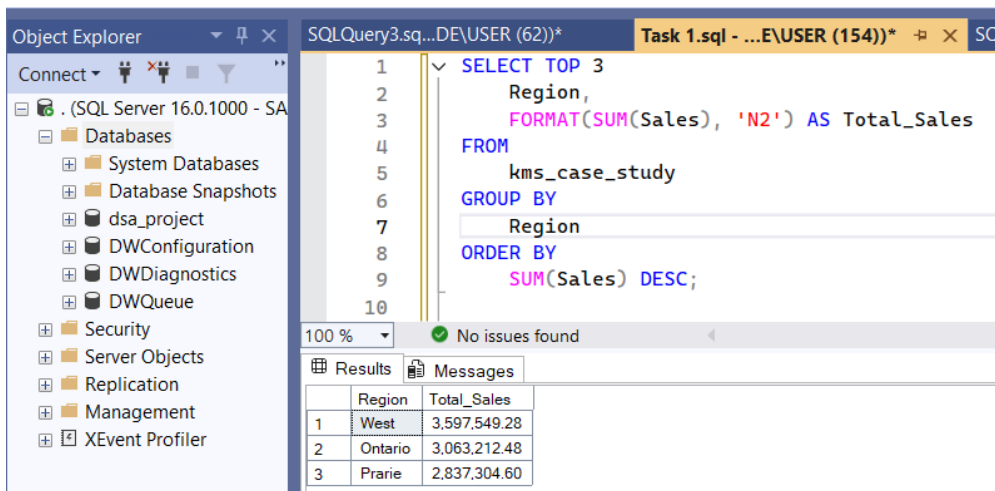
The Results pane shows the following data:

| Product Category | Total_Sales |
|------------------|--------------|
| 1 Technology | 5,984,248.18 |

Task 2: What are the Top 3 and Bottom 3 regions in terms of sales?

===== Top 3 =====

```
SELECT TOP 3
    Region,
    FORMAT(SUM(Sales), 'N2') AS Total_Sales
FROM
    kms_case_study
GROUP BY
    Region
ORDER BY
    SUM(Sales) DESC;
```



The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left shows the database structure. The SQL window displays the following query:

```
SELECT TOP 3
    Region,
    FORMAT(SUM(Sales), 'N2') AS Total_Sales
FROM
    kms_case_study
GROUP BY
    Region
ORDER BY
    SUM(Sales) DESC;
```

The Results pane shows the following data:

| Region | Total_Sales |
|-----------|--------------|
| 1 West | 3,597,549.28 |
| 2 Ontario | 3,063,212.48 |
| 3 Prairie | 2,837,304.60 |

===== Bottom 3 =====

```
SELECT TOP 3
    Region,
    FORMAT(SUM(Sales), 'N2') AS Total_Sales
FROM
    kms_case_study
GROUP BY
    Region
ORDER BY
    SUM(Sales) ASC;
```

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'SQL Server 16.0.1000 - SA'. The main window shows a query in the 'SQLQuery3.sql' file. The query is a SELECT statement with a TOP 3 clause, selecting Region and Total_Sales (formatted as N2) from the kms_case_study database, grouped by Region and ordered by SUM(Sales) ASC. The Results pane at the bottom shows the output of the query, which is a table with three rows: Nunavut (116,376.48), Northwest Territories (800,847.33), and Yukon (975,867.37).

| | Region | Total_Sales |
|---|-----------------------|-------------|
| 1 | Nunavut | 116,376.48 |
| 2 | Northwest Territories | 800,847.33 |
| 3 | Yukon | 975,867.37 |

Task 3: What were the total sales of appliances in Ontario?

```
SELECT
    FORMAT(SUM(Sales), 'N') AS Appliance_Sales_Ontario
FROM
    kms_case_study
WHERE
    [Product Sub-Category] = 'Appliances'
    AND Province = 'Ontario';
```

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'SQL Server 16.0.1000 - SA'. The main window shows a query in the 'SQLQuery3.sql' file. The query is a SELECT statement selecting Appliance_Sales_Ontario (formatted as N) from the kms_case_study database, filtered by [Product Sub-Category] = 'Appliances' and Province = 'Ontario'. The Results pane at the bottom shows the output of the query, which is a table with one row: Appliance_Sales_Ontario (202,346.84).

| | Appliance_Sales_Ontario |
|---|-------------------------|
| 1 | 202,346.84 |

Task 4: Advise the management of KMS on what to do to increase the revenue from the bottom 10 customers

Step 1: Identify bottom 10 customers by total sales

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure. The main window shows a SQL query in the 'SQLQuery4.s... \USER (161))*' tab. The query is as follows:

```
SELECT TOP 10
  [Customer Name],
  FORMAT(SUM(Sales), 'N') AS Total_Sales
FROM
  kms_case_study
GROUP BY
  [Customer Name]
ORDER BY
  Total_Sales ASC;
```

The query results are displayed in the 'Results' tab, showing the top 10 customers by total sales:

| | Customer Name | Total_Sales |
|----|--------------------|-------------|
| 1 | Anthony Jacobs | 1,029.47 |
| 2 | Toby Braunhardt | 1,060.34 |
| 3 | Stewart Carmichael | 1,134.88 |
| 4 | Bruce Money | 1,167.03 |
| 5 | Maria Etezadi | 1,219.25 |
| 6 | Bart Pistole | 1,220.14 |
| 7 | Alyssa Tate | 1,235.31 |
| 8 | Sam Zeldin | 1,290.00 |
| 9 | Anna Chung | 1,310.39 |
| 10 | Henry MacAllister | 1,348.46 |

Step 2: Recommendation

1. Assign a dedicated business developer or team to rebuild the relationship by understanding their needs and what competitors offer them.
2. Offer product price and shipping cost discounts or loyalty rewards targeted at these customers.
3. Design a bespoke product combo for this set of customers. This offers beneficial discounts for the customer and higher quantity purchases for the KMS.
4. Assure faster shipping times for the few customers impacted by shipping delays.
5. Follow up with the customers through surveys, emails, and phone calls to address their needs and gather customer feedback.

Task 5: KMS incurred the most shipping cost using which shipping method?

SELECT TOP 1

[Ship Mode],

SUM([Shipping Cost]) AS Total_Shipping_Cost

FROM

kms_case_study

GROUP BY

[Ship Mode]

ORDER BY

Total_Shipping_Cost DESC;

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure. The main window shows a SQL query in the 'SQLQuery6.sq... DE\USER (56))*' tab. The query is as follows:

```
SELECT TOP 1
  [Ship Mode],
  SUM([Shipping Cost]) AS Total_Shipping_Cost
FROM
  kms_case_study
GROUP BY
  [Ship Mode]
ORDER BY
  Total_Shipping_Cost DESC;
```

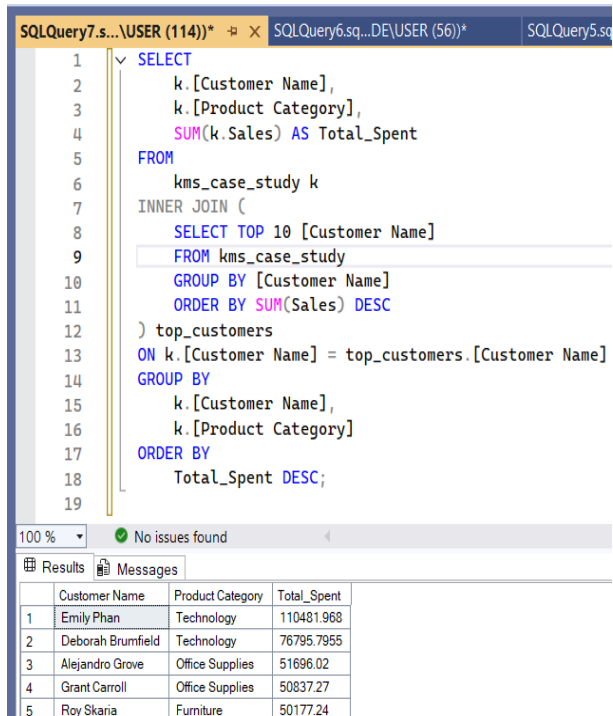
The query results are displayed in the 'Results' tab, showing the shipping method with the highest total shipping cost:

| | Ship Mode | Total_Shipping_Cost |
|---|----------------|---------------------|
| 1 | Delivery Truck | 51971.9399999998 |

Case Scenario II

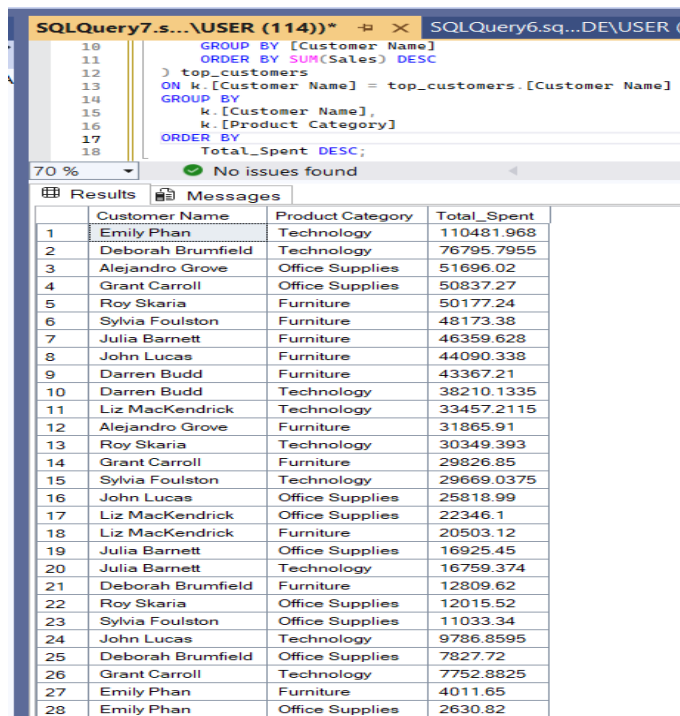
Task 6: Who are the most valuable customers, and what products or services do they typically purchase?

```
SELECT
    k.[Customer Name],
    k.[Product Category],
    SUM(k.Sales) AS Total_Spent
FROM
    kms_case_study k
INNER JOIN (
    SELECT TOP 10 [Customer Name]
    FROM kms_case_study
    GROUP BY [Customer Name]
    ORDER BY SUM(Sales) DESC
) top_customers
ON k.[Customer Name] = top_customers.[Customer Name]
GROUP BY
    k.[Customer Name],
    k.[Product Category]
ORDER BY
    Total_Spent DESC;
```



```
1 SELECT
2     k.[Customer Name],
3     k.[Product Category],
4     SUM(k.Sales) AS Total_Spent
5 FROM
6     kms_case_study k
7 INNER JOIN (
8     SELECT TOP 10 [Customer Name]
9     FROM kms_case_study
10    GROUP BY [Customer Name]
11    ORDER BY SUM(Sales) DESC
12 ) top_customers
13 ON k.[Customer Name] = top_customers.[Customer Name]
14 GROUP BY
15     k.[Customer Name],
16     k.[Product Category]
17 ORDER BY
18     Total_Spent DESC;
```

| | Customer Name | Product Category | Total_Spent |
|---|-------------------|------------------|-------------|
| 1 | Emily Phan | Technology | 110481.968 |
| 2 | Deborah Brumfield | Technology | 76795.7955 |
| 3 | Alejandro Grove | Office Supplies | 51696.02 |
| 4 | Grant Carroll | Office Supplies | 50837.27 |
| 5 | Roy Skaria | Furniture | 50177.24 |



```
10 GROUP BY [Customer Name]
11 ORDER BY SUM(Sales) DESC
12 ) top_customers
13 ON k.[Customer Name] = top_customers.[Customer Name]
14 GROUP BY
15     k.[Customer Name],
16     k.[Product Category]
17 ORDER BY
18     Total_Spent DESC;
```

| | Customer Name | Product Category | Total_Spent |
|----|-------------------|------------------|-------------|
| 1 | Emily Phan | Technology | 110481.968 |
| 2 | Deborah Brumfield | Technology | 76795.7955 |
| 3 | Alejandro Grove | Office Supplies | 51696.02 |
| 4 | Grant Carroll | Office Supplies | 50837.27 |
| 5 | Roy Skaria | Furniture | 50177.24 |
| 6 | Sylvia Foulston | Furniture | 48173.38 |
| 7 | Julia Barnett | Furniture | 46359.628 |
| 8 | John Lucas | Furniture | 44090.338 |
| 9 | Darren Budd | Furniture | 43367.21 |
| 10 | Darren Budd | Technology | 38210.1335 |
| 11 | Liz MacKendrick | Technology | 33457.2115 |
| 12 | Alejandro Grove | Furniture | 31865.91 |
| 13 | Roy Skaria | Technology | 30349.393 |
| 14 | Grant Carroll | Furniture | 29826.85 |
| 15 | Sylvia Foulston | Technology | 29669.0375 |
| 16 | John Lucas | Office Supplies | 25818.99 |
| 17 | Liz MacKendrick | Office Supplies | 22346.1 |
| 18 | Liz MacKendrick | Furniture | 20503.12 |
| 19 | Julia Barnett | Office Supplies | 16925.45 |
| 20 | Julia Barnett | Technology | 16759.374 |
| 21 | Deborah Brumfield | Furniture | 12809.62 |
| 22 | Roy Skaria | Office Supplies | 12015.52 |
| 23 | Sylvia Foulston | Office Supplies | 11033.34 |
| 24 | John Lucas | Technology | 9786.8595 |
| 25 | Deborah Brumfield | Office Supplies | 7827.72 |
| 26 | Grant Carroll | Technology | 7752.8825 |
| 27 | Emily Phan | Furniture | 4011.65 |
| 28 | Emily Phan | Office Supplies | 2630.82 |

Task 7: Which small business customer had the highest sales?

```
SELECT TOP 1
    [Customer Name],
    SUM(Sales) AS Total_Sales
FROM
    kms_case_study
WHERE
    [Customer Segment] = 'Small Business'
GROUP BY
    [Customer Name]
ORDER BY
    Total_Sales DESC;
```

The screenshot shows the SQL Server Enterprise Manager interface. On the left is the Object Explorer showing the database structure. The main window displays a SQL query in a query editor. The query is as follows:

```

1 SELECT TOP 1
2     [Customer Name],
3     SUM(Sales) AS Total_Sales
4 FROM
5     kms_case_study
6 WHERE
7     [Customer Segment] = 'Small Business'
8 GROUP BY
9     [Customer Name]
10 ORDER BY
11     Total_Sales DESC;
12

```

Below the query editor, the status bar indicates "100 %", "No issues found", and tabs for "Results" and "Messages". The "Results" tab is active, showing a table with the following data:

| | Customer Name | Total_Sales |
|---|---------------|-------------|
| 1 | Dennis Kane | 75967.5905 |

Task 8: Which Corporate Customer placed the most number of orders in 2009–2012?

WITH RankedOrders AS (

SELECT

[Customer Name],

COUNT(DISTINCT [Order ID]) AS Num_Orders,

RANK() OVER (ORDER BY COUNT(DISTINCT [Order ID]) DESC) AS OrderRank

FROM

kms_case_study

WHERE

[Customer Segment] = 'Corporate' AND

YEAR([Order Date]) BETWEEN 2009 AND 2012

GROUP BY

[Customer Name]

)

SELECT

[Customer Name],

Num_Orders

FROM

RankedOrders

WHERE

OrderRank = 1;

The screenshot shows the SQL Server Enterprise Manager interface. On the left is the Object Explorer showing the database structure. The main window displays a SQL query in a query editor. The query is as follows:

```

1 WITH RankedOrders AS (
2     SELECT
3         [Customer Name],
4         COUNT(DISTINCT [Order ID]) AS Num_Orders,
5         RANK() OVER (ORDER BY COUNT(DISTINCT [Order ID]) DESC) AS OrderRank
6     FROM
7         kms_case_study
8     WHERE
9         [Customer Segment] = 'Corporate' AND
10        YEAR([Order Date]) BETWEEN 2009 AND 2012
11    GROUP BY
12        [Customer Name]
13 )
14 SELECT
15     [Customer Name],
16     Num_Orders
17 FROM
18     RankedOrders
19 WHERE
20     OrderRank = 1;

```

Below the query editor, the status bar indicates "100 %", "7" errors, and "0" warnings. The "Results" tab is active, showing a table with the following data:

| | Customer Name | Num_Orders |
|---|---------------|------------|
| 1 | Adam Hart | 18 |
| 2 | Roy Skaria | 18 |

Task 9: Which consumer customer was the most profitable one?

SELECT TOP 1

[Customer Name],
SUM(Profit) AS Total_Profit

FROM

kms_case_study

WHERE

[Customer Segment] = 'Consumer'

GROUP BY

[Customer Name]

ORDER BY

Total_Profit DESC;

The screenshot shows the SQL Server Enterprise Manager interface. On the left is the Object Explorer showing the database structure. The main window displays a query window titled 'SQLQuery10....E\USER (63))' with the following SQL code:

```
SELECT TOP 1
[Customer Name],
SUM(Profit) AS Total_Profit
FROM
kms_case_study
WHERE
[Customer Segment] = 'Consumer'
GROUP BY
[Customer Name]
ORDER BY
Total_Profit DESC;
```

Below the query window, the 'Results' tab shows the output of the query:

| Customer Name | Total_Profit |
|---------------|--------------|
| Emily Phan | 34005.44 |

Task 10: Which customers returned items, and what segment do they belong to?

SELECT DISTINCT

k.[Customer Name],
k.[Customer Segment]

FROM

kms_case_study k

INNER JOIN

order_status o ON k.[Order ID] = o.[Order ID]

WHERE

o.Status = 'Returned';

The screenshot shows the SQL Server Enterprise Manager interface. On the left is the Object Explorer showing the database structure. The main window displays a query window titled 'SQLQuery11....\USER (158))' with the following SQL code:

```
SELECT DISTINCT
k.[Customer Name],
k.[Customer Segment]
FROM
kms_case_study k
INNER JOIN
order_status o ON k.[Order ID] = o.[Order ID]
WHERE
o.Status = 'Returned';
```

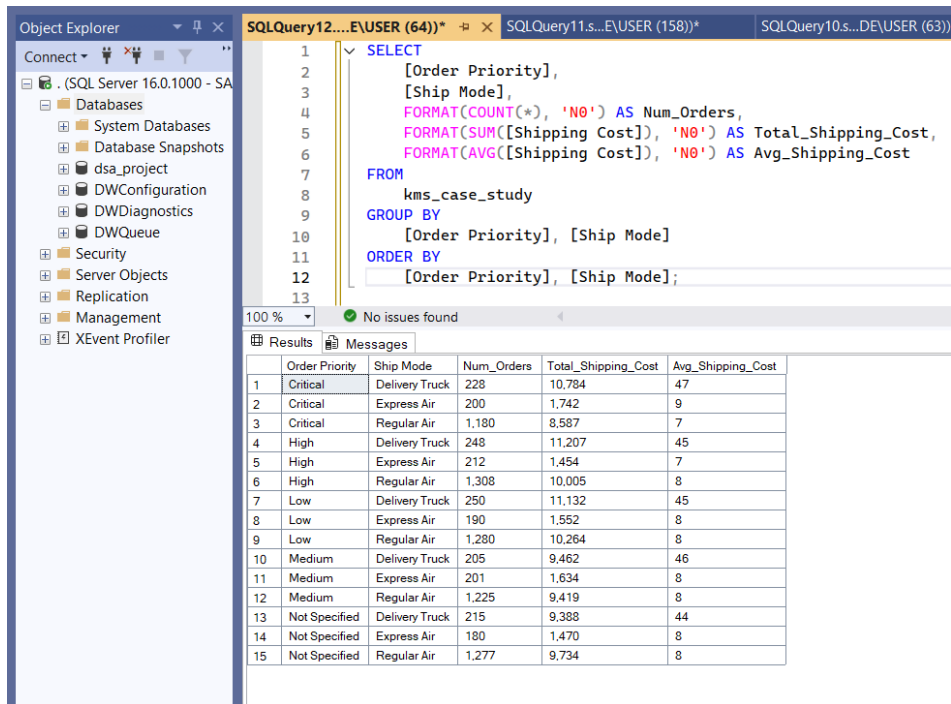
Below the query window, the 'Results' tab shows the output of the query:

| Customer Name | Customer Segment |
|---------------------|------------------|
| Aaron Bergman | Corporate |
| Aaron Hawkins | Home Office |
| Adam Bellavance | Small Business |
| Adrian Barton | Small Business |
| Alan Hwang | Corporate |
| Alan Hwang | Small Business |
| Alejandro Grove | Consumer |
| Alejandro Grove | Corporate |
| Alejandro Savely | Corporate |
| Aleksandra Gannaway | Corporate |
| Alex Russell | Home Office |
| Alice McCarthy | Corporate |
| Alyssa Tate | Small Business |
| Amy Cox | Home Office |
| Amy Cox | Small Business |
| Amy Hunt | Home Office |
| Amy Hunt | Small Business |
| Anna Chung | Home Office |
| Anna Gayman | Corporate |
| Anna Haberlin | Corporate |
| Anne Pryor | Consumer |
| Annie Thurman | Consumer |
| Anthony Ganerick | Small Business |
| Anthony O'Donnell | Consumer |
| Art Ferguson | Corporate |

At the bottom of the window, a status bar indicates 'Query executed successfully.'

Task 11: Was shipping cost aligned with order priority?

```
SELECT
    [Order Priority],
    [Ship Mode],
    FORMAT(COUNT(*), 'N0') AS Num_Orders,
    FORMAT(SUM([Shipping Cost]), 'N0') AS Total_Shipping_Cost,
    FORMAT(AVG([Shipping Cost]), 'N0') AS Avg_Shipping_Cost
FROM
    kms_case_study
GROUP BY
    [Order Priority], [Ship Mode]
ORDER BY
    [Order Priority], [Ship Mode];
```



The screenshot shows a SQL Server Enterprise Manager interface. On the left is the Object Explorer showing the database structure. The main window displays a SQL query in a query window titled 'SQLQuery12...E\USER (64)'. The query is the same as the one in the previous block. Below the query window, the 'Results' tab is active, showing a grid with 15 rows and 6 columns: Order Priority, Ship Mode, Num_Orders, Total_Shipping_Cost, and Avg_Shipping_Cost. The data is sorted by Order Priority and then Ship Mode.

| | Order Priority | Ship Mode | Num_Orders | Total_Shipping_Cost | Avg_Shipping_Cost |
|----|----------------|----------------|------------|---------------------|-------------------|
| 1 | Critical | Delivery Truck | 228 | 10,784 | 47 |
| 2 | Critical | Express Air | 200 | 1,742 | 9 |
| 3 | Critical | Regular Air | 1,180 | 8,587 | 7 |
| 4 | High | Delivery Truck | 248 | 11,207 | 45 |
| 5 | High | Express Air | 212 | 1,454 | 7 |
| 6 | High | Regular Air | 1,308 | 10,005 | 8 |
| 7 | Low | Delivery Truck | 250 | 11,132 | 45 |
| 8 | Low | Express Air | 190 | 1,552 | 8 |
| 9 | Low | Regular Air | 1,280 | 10,264 | 8 |
| 10 | Medium | Delivery Truck | 205 | 9,462 | 46 |
| 11 | Medium | Express Air | 201 | 1,634 | 8 |
| 12 | Medium | Regular Air | 1,225 | 9,419 | 8 |
| 13 | Not Specified | Delivery Truck | 215 | 9,388 | 44 |
| 14 | Not Specified | Express Air | 180 | 1,470 | 8 |
| 15 | Not Specified | Regular Air | 1,277 | 9,734 | 8 |

The average cost per order for each shipping method;

- Delivery Truck: Approximately £45.35 per order (£51,972 / 1,146 orders).
- Express Air: Approximately £7.99 per order (£7,851 / 983 orders).
- Regular Air: Approximately £7.66 per order (£48,008 / 6,270 orders).

Upon analysis of the shipping data, a contradiction emerges regarding the cost-effectiveness of methods. The Delivery Truck, initially stated as the most economical, is actually the most expensive. Express Air, presumed the most costly, proves economical. Regular Air costs approximately £7.66 per order.

This misalignment impacts the assessment of appropriate spending. Regular Air dominates shipments across all priorities. However, the consistent use of the expensive Delivery Truck for all priorities, including lower ones, suggests inefficiency. Conversely, Express Air, a fast and economical option, is underutilized for critical and high-priority orders. Therefore, current shipping practices appear misaligned with an optimal balance of cost-efficiency and priority-driven speed. Re-evaluation of shipping method selection is recommended for improved expenditure management.