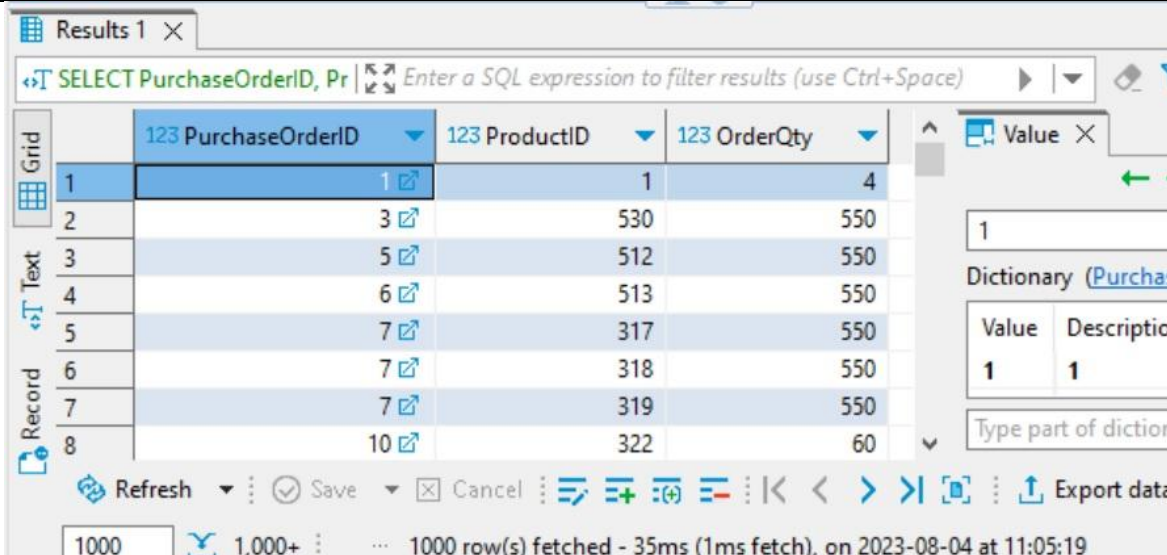


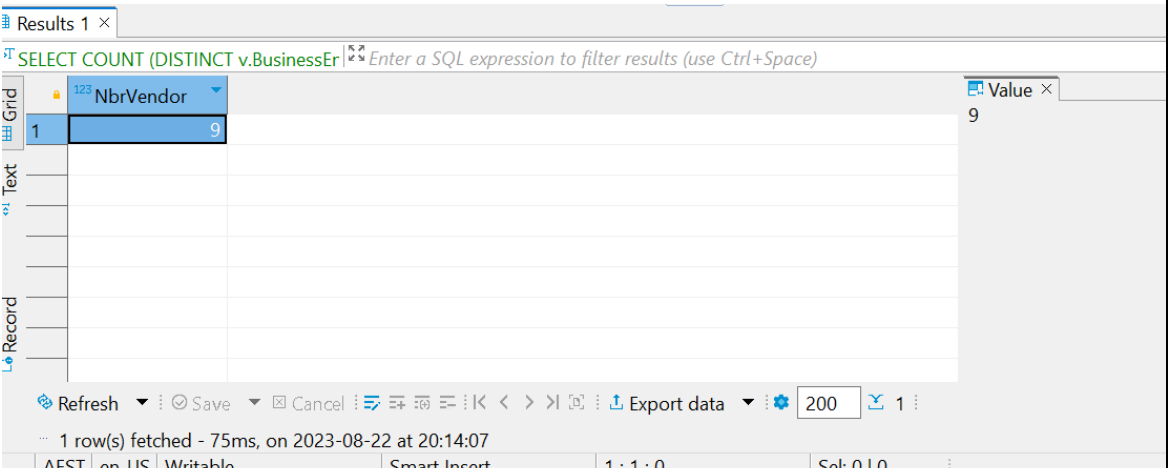
Data Wrangling and R

Using SQL for Business Insights

2.2.1. Display the orders that have an order quantity greater than 3.

Code	SELECT pod.PurchaseOrderID, pod.ProductID , pod.OrderQty FROM Purchasing.PurchaseOrderDetail pod WHERE pod.OrderQty > 3																								
Returned table	 <p>Results 1</p> <p>Enter a SQL expression to filter results (use Ctrl+Space)</p> <p>Grid</p> <p>123 PurchaseOrderID 123 ProductID 123 OrderQty</p> <table><tr><td>1</td><td>1</td><td>4</td></tr><tr><td>2</td><td>3</td><td>550</td></tr><tr><td>3</td><td>5</td><td>550</td></tr><tr><td>4</td><td>6</td><td>550</td></tr><tr><td>5</td><td>7</td><td>550</td></tr><tr><td>6</td><td>7</td><td>550</td></tr><tr><td>7</td><td>7</td><td>550</td></tr><tr><td>8</td><td>10</td><td>60</td></tr></table> <p>Value</p> <p>1</p> <p>Dictionary (Purchasing)</p> <p>Value Description</p> <p>1 1</p> <p>Type part of dictionary</p> <p>Refresh Save Cancel Export data</p> <p>1000 1,000+ 1000 row(s) fetched - 35ms (1ms fetch), on 2023-08-04 at 11:05:19</p>	1	1	4	2	3	550	3	5	550	4	6	550	5	7	550	6	7	550	7	7	550	8	10	60
1	1	4																							
2	3	550																							
3	5	550																							
4	6	550																							
5	7	550																							
6	7	550																							
7	7	550																							
8	10	60																							

2.2.2. Querying Vendor Orders: Counting Vendors with Credit Ratings Above 2

Code	SELECT COUNT (DISTINCT v.BusinessEntityID) as NbrVendor FROM Purchasing.Vendor v JOIN Purchasing.PurchaseOrderHeader poh on v.BusinessEntityID = poh.VendorID WHERE v.CreditRating > 2		
Returned table	 <p>Results 1</p> <p>Enter a SQL expression to filter results (use Ctrl+Space)</p> <p>Grid</p> <p>123 NbrVendor</p> <table><tr><td>1</td><td>9</td></tr></table> <p>Value</p> <p>9</p> <p>Refresh Save Cancel Export data</p> <p>200 1 1 row(s) fetched - 75ms, on 2023-08-22 at 20:14:07</p> <p>AEST en_US Writable Smart Insert 1 : 1 : 0 Sel: 0 0</p>	1	9
1	9		

2.2.3. Display names of vendors and their preferred status of vendors who live in Washington province.

Code	SELECT v.Name , v.PreferredVendorStatus, vvwa.StateProvinceName FROM Purchasing.Vendor v, Purchasing.vVendorWithAddresses vvwa WHERE vvwa.StateProvinceName = 'Washington'
------	---

Return
ed
table

Results 1 x

SQL

SELECT v.Name , v.PreferredVendorStat

Enter a SQL expression to filter results (use Ctrl+Space)

Grid

	abc Name	123 PreferredVendorStatus	abc StateProvinceName
1	Australia Bike Retailer	1	Washington
2	Allenson Cycles	1	Washington
3	Advanced Bicycles	1	Washington
4	Trikes, Inc.	1	Washington
5	Morgan Bike Accessories	1	Washington
6	Cycling Master	1	Washington
7	Chicago Rent-All	1	Washington
8	Greenwood Athletic Company	1	Washington
9	Complete Enterprises, Inc	1	Washington

Text

Value x

Australia Bike R

Record

Refresh

Save

Cancel

Export data

200

200+

200 row(s) fetched - 1.377s (2ms fetch), on 2023-08-22 at 20:15:30

AEST

en_US

Writable

Smart Insert

7 : 1 : 193

Sel: 0 | 0

2.2.4. Distinct 2012 Orders Shipped by 'Truck' or 'Cargo Ship': Count and Total Quantity

Code	<pre>SELECT COUNT (DISTINCT pod.PurchaseOrderID) as TotalNbrOrder, SUM(pod.OrderQty) as TheTotalQuantity FROM Purchasing.PurchaseOrderHeader poh JOIN Purchasing.ShipMethod sm on sm.ShipMethodID = poh.ShipMethodID JOIN Purchasing.PurchaseOrderDetail pod on pod.PurchaseOrderID = poh.PurchaseOrderID WHERE (sm.Name LIKE '%truck%' or sm.Name LIKE '%cargo%') AND (poh.OrderDate BETWEEN '2012-01-01' AND '2012-12-31')</pre>						
Returned table	<div>Results 1 x</div> <div>SELECT COUNT (DISTINCT pod.PurchaseOrderID) as TotalNbrOrder, SUM(pod.OrderQty) as TheTotalQuantity</div> <table><thead><tr><th></th><th>TotalNbrOrder</th><th>TheTotalQuantity</th></tr></thead><tbody><tr><td>1</td><td>135</td><td>76,794</td></tr></tbody></table> <div>Value x 135</div> <div>All columns are read-only</div> <div>Refresh Save Cancel Export data 200 1</div> <div>1 row(s) fetched - 80ms, on 2023-08-22 at 20:48:50</div> <div>AEST en_US Writable Smart Insert 11 : 1 [417] Sel: 417 7</div>		TotalNbrOrder	TheTotalQuantity	1	135	76,794
	TotalNbrOrder	TheTotalQuantity					
1	135	76,794					

2.2.5. Display vendors name, state and its latest receipt cost in descending order of both standard price and average lead times.

Code	SELECT v.Name as VendorName , vvwa.StateProvinceName , pv.LastReceiptCost FROM Purchasing.Vendor v JOIN Purchasing.vVendorWithAddresses vvwa on v.BusinessEntityID = vvwa.BusinessEntityID JOIN Purchasing.ProductVendor pv on pv.BusinessEntityID = v.BusinessEntityID ORDER BY pv.AverageLeadTime DESC, pv.StandardPrice DESC
------	---

Returned table	Results 1 x		
	<pre>SELECT v.Name as VendorName , vvwa.St</pre>		
	Grid	VendorName	StateProvinceName
	1	International Trek Center	California
	2	Fitness Association	Washington
	3	Fitness Association	Washington
	4	Fitness Association	Washington
	5	Fitness Association	Washington
	6	Fitness Association	Washington
	7	Fitness Association	Washington
	8	Integrated Sport Products	California
	9	Green Lake Bike Company	Colorado
	10	Green Lake Bike Company	Colorado
	11	Integrated Sport Products	California
	12	Integrated Sport Products	California
	13	Integrated Sport Products	California
	Record		
	Refresh	Save	Cancel
	200 row(s) fetched - 79ms (3ms fetch), on 2023-09-10 at 00:54:49		

2.2.6. Calculate the quarter sales of vendors in Washington province.

Code

```
SELECT V.Name AS VendorName , YEAR(poh.OrderDate) AS 'Year',
DATEPART(QUARTER,poh.OrderDate) AS 'Quarter',
SUM(pod.OrderQty) as TotalQuantity, SUM(pod.LineTotal) AS TotalSales
FROM Purchasing.PurchaseOrderHeader poh
JOIN Purchasing.PurchaseOrderDetail pod
ON pod.PurchaseOrderID = poh.PurchaseOrderID
JOIN Purchasing.Vendor v
ON v.BusinessEntityID = poh.VendorID
JOIN Purchasing.vVendorWithAddresses vvwa
ON vvwa.BusinessEntityID = v.BusinessEntityID
WHERE vvwa.StateProvinceName ='Washington'
GROUP BY v.Name, YEAR(poh.OrderDate), DATEPART(QUARTER,poh.Orderdate)
ORDER BY YEAR(poh.Orderdate), DATEPART (QUARTER,poh.Orderdate)
```

Returned table

Results 1 x

SELECT V.Name AS VendorName , YEAR(poh.OrderDate) AS 'Year', DATEPART(QUARTER,poh.OrderDate) AS 'Quarter', SUM(pod.OrderQty) as TotalQuantity, SUM(pod.LineTotal) AS TotalSales FROM Purchasing.PurchaseOrderHeader poh JOIN Purchasing.PurchaseOrderDetail pod ON pod.PurchaseOrderID = poh.PurchaseOrderID JOIN Purchasing.Vendor v ON v.BusinessEntityID = poh.VendorID JOIN Purchasing.vVendorWithAddresses vvwa ON vvwa.BusinessEntityID = v.BusinessEntityID WHERE vvwa.StateProvinceName ='Washington' GROUP BY v.Name, YEAR(poh.OrderDate), DATEPART(QUARTER,poh.Orderdate) ORDER BY YEAR(poh.Orderdate), DATEPART (QUARTER,poh.Orderdate)

	VendorName	Year	Quarter	TotalQuantity	TotalSales
1	Advanced Bicycles	2,011	2	6	272.1015
2	Aurora Bike Center	2,011	2	15	693.3780
3	Australia Bike Retailer	2,011	4	15	694.1655
4	Compete Enterprises, Inc	2,011	4	1,650	37312.2750
5	Advanced Bicycles	2,012	1	12	560.0070
6	Aurora Bike Center	2,012	1	18	773.0100
7	Australia Bike Retailer	2,012	1	15	656.5860
8	Compete Enterprises, Inc	2,012	1	1,100	35394.9750
9	Electronic Bike Repair & Supplies	2,012	1	1,650	48030.6750
10	Mitchell Sports	2,012	1	2,200	87756.9000
11	Mountain Works	2,012	1	90	3939.0120
12	National Bike Association	2,012	1	240	923.5800
13	Northern Bike Travel	2,012	1	2	27.0755

Refresh

Save

Cancel

Export data

200

169

169 row(s) fetched - 96ms (4ms fetch), on 2023-09-10 at 00:57:48

2.2.7. Top Two Vendors by Order Quantity

Code	<pre>SELECT TOP 2 v.Name AS VendorName, SUM(pod.OrderQty) AS TotalOrderQuantity FROM Purchasing.PurchaseOrderHeader poh JOIN Purchasing.PurchaseOrderDetail pod ON pod.PurchaseOrderID = poh.PurchaseOrderID JOIN Purchasing.Vendor v ON v.BusinessEntityID = poh.VendorID GROUP BY v.Name ORDER BY sum(pod.OrderQty) DESC</pre>
------	--

Return d table	Results 1 x	
	SELECT TOP 2 v.Name AS VendorName, SUM(poc Enter a SQL expression to filter results (use Ctrl+Space)	
	Grid	Value x
	1	SUPERSALES INC. 125,000
	2	Custom Frames, Inc. 115,500
	Record	
	Refresh	Save Cancel
	Export data	200 2
	2 row(s) fetched - 81ms, on 2023-09-04 at 23:07:41	
	AEST en_AU Writable	Smart Insert 33 : 32 : 1510 Sel: 0 0

2.2.8. Identifying the Best Vendors by Shipping Type: Criteria Proposal, Justification, and SQL Solution for Optimized Order Processing

Proposed criterion	Select the top five vendors for the five shipping methods based on highest order quantity processed in the lowest average processing time.
Justification	<p>The vendors are selected by the highest order quantity handled for each shipping type. This guarantees that the vendors with the highest sales volume are prioritized.</p> <p>Minimum average processing time is an important variable in determining order fulfillment efficiency.</p> <p>The goal is to determine the top five suppliers for every delivery method who not only handle a large volume of orders but also finish the deal in the shortest period of time, showing effectiveness in their operations. It will help to increase the company's sales and reputation.</p>
Code	<p>WITH AvgProcessTime AS (SELECT poh.VendorID, AVG(DATEDIFF(DAY, poh.OrderDate, poh.ShipDate)) AS AvgProcessTime, sm.ShipMethodID FROM Purchasing.PurchaseOrderHeader poh JOIN Purchasing.ShipMethod sm ON poh.ShipMethodID = sm.ShipMethodID JOIN Purchasing.PurchaseOrderDetail pod ON poh.PurchaseOrderID = pod.PurchaseOrderID GROUP BY poh.VendorID, sm.ShipMethodID), TotalOrderQuantity AS (SELECT poh.VendorID as VendorId, poh.ShipMethodID as ShipMethodID, sm.Name as ShipMethod, SUM(pod.OrderQty) AS TotalOrderQty FROM Purchasing.PurchaseOrderDetail pod JOIN Purchasing.PurchaseOrderHeader poh ON poh.PurchaseOrderID = pod.PurchaseOrderID JOIN Purchasing.ShipMethod sm ON sm.ShipMethodID = poh.ShipMethodID GROUP BY poh.VendorID, poh.ShipMethodID, sm.Name), RankedVendors AS (SELECT TotalOrderQuantity.ShipMethodID AS ShipMethodID, TotalOrderQuantity.ShipMethod, TotalOrderQuantity.VendorID AS VendorID, TotalOrderQuantity.TotalOrderQty AS TotalOrderQty, AvgProcessTime.AvgProcessTime AS AvgProcessTime, ROW_NUMBER() OVER (PARTITION BY TotalOrderQuantity.ShipMethodID ORDER BY TotalOrderQuantity.TotalOrderQty DESC) AS Rank FROM TotalOrderQuantity JOIN AvgProcessTime ON TotalOrderQuantity.VendorID = AvgProcessTime.VendorID AND TotalOrderQuantity.ShipMethodID = AvgProcessTime.ShipMethodID) SELECT RankedVendors.ShipMethodID, RankedVendors.ShipMethod, RankedVendors.VendorID, v.Name AS VendorName,</p>

	RankedVendors.TotalOrderQty, RankedVendors.AvgProcessTime FROM RankedVendors JOIN Purchasing.Vendor v ON RankedVendors.VendorID = v.BusinessEntityID WHERE Rank = 1 ORDER BY ShipMethodID ASC ;
Returned table	

2.2.9. Display the vendor that have sold the highest orders and lowest lead time that is the best vendor.

Code	<pre> WITH AvgProcessTime AS (SELECT poh.VendorID, AVG(DATEDIFF(DAY, poh.OrderDate, poh.ShipDate)) AS AvgProcessTime, sm.ShipMethodID FROM Purchasing.PurchaseOrderHeader poh JOIN Purchasing.ShipMethod sm ON poh.ShipMethodID = sm.ShipMethodID JOIN Purchasing.PurchaseOrderDetail pod ON poh.PurchaseOrderID = pod.PurchaseOrderID GROUP BY poh.VendorID, sm.ShipMethodID), TotalOrderQuantity AS (SELECT poh.VendorID as VendorId, poh.ShipMethodID as ShipMethodID, sm.Name as ShipMethod, SUM(pod.OrderQty) AS TotalOrderQty FROM Purchasing.PurchaseOrderDetail pod JOIN Purchasing.PurchaseOrderHeader poh ON poh.PurchaseOrderID = pod.PurchaseOrderID JOIN Purchasing.ShipMethod sm ON sm.ShipMethodID = poh.ShipMethodID GROUP BY poh.VendorID, poh.ShipMethodID, sm.Name), RankedVendors AS (SELECT TotalOrderQuantity.ShipMethodID AS ShipMethodID, TotalOrderQuantity.ShipMethod, TotalOrderQuantity.VendorID AS VendorID, TotalOrderQuantity.TotalOrderQty AS TotalOrderQty, AvgProcessTime.AvgProcessTime AS AvgProcessTime, ROW_NUMBER() OVER (PARTITION BY TotalOrderQuantity.ShipMethodID ORDER BY TotalOrderQuantity.TotalOrderQty DESC) AS Rank FROM TotalOrderQuantity JOIN AvgProcessTime ON TotalOrderQuantity.VendorID = AvgProcessTime.VendorID AND TotalOrderQuantity.ShipMethodID = AvgProcessTime.ShipMethodID) SELECT TOP 1 RankedVendors.ShipMethodID as ShipMethodID, RankedVendors.ShipMethod, RankedVendors.VendorID, v.Name AS VendorName, RankedVendors.TotalOrderQty, pv.AverageLeadTime FROM RankedVendors JOIN Purchasing.Vendor v ON RankedVendors.VendorID = v.BusinessEntityID </pre>
------	--

	JOIN Purchasing.ProductVendor pv ON RankedVendors.VendorID = pv.BusinessEntityID JOIN Purchasing.ShipMethod sm ON sm.ShipMethodID = RankedVendors.ShipMethodID ORDER BY RankedVendors.TotalOrderQty DESC , pv.AverageLeadTime ASC ;
Returned table	

2.2.10. Finding the Most Efficient and Cost-Effective Vendor: Criteria Definition and SQL Solution for Optimal Selection

Proposed criterion	Select the best vendor that has the smallest order return rate and the cheapest standard price.
Justification	<p>Low return rates tend to be indicative of great product quality, customer happiness, and efficient customer service.</p> <p>The Lower standard cost might make a vendor more appealing to clients and contribute to their market competitiveness.</p> <p>The goal is to choose a vendor who not only has a solid track record of reducing order returns but also offers affordable pricing. When it comes to quality and pricing, this vendor is likely to be the greatest choice for the organization.</p>
Code	WITH ReturnRate AS (SELECT pv.BusinessEntityID AS VendorID, AVG (pod.RejectedQty / pod.OrderQty) AS AvgOrderReturnRate FROM Purchasing.ProductVendor pv JOIN Purchasing.PurchaseOrderDetail pod ON pv.ProductID = pod.ProductID GROUP BY pv.BusinessEntityID), VendorCheapestPrice AS (SELECT pv.BusinessEntityID AS VendorID, MIN (pv.StandardPrice) AS MinStandardPrice FROM Purchasing.ProductVendor pv GROUP BY pv.BusinessEntityID) SELECT TOP 1 a.VendorID, v.Name AS VendorName, a.AvgOrderReturnRate, b.MinStandardPrice FROM ReturnRate a JOIN VendorCheapestPrice b ON a.VendorID = b.VendorID JOIN Purchasing.Vendor v ON b.VendorID = v.BusinessEntityID ORDER BY a.AvgOrderReturnRate ASC , b.MinStandardPrice ASC ;

Returned
table

The screenshot shows a SQL query window with the following query:

```
WITH ReturnRate AS ( SELECT pv.BusinessEntityID AS VendorID,
AVG(pod.RejectedQty / pod.OrderQty) AS AvgOrderReturnRate
FROM Purchasing.ProductVendor pv
JOIN Purchasing.PurchaseOrderDetail pod
ON pv.ProductID = pod.ProductID
GROUP BY pv.BusinessEntityID),
VendorCheapestPrice AS (
SELECT pv.BusinessEntityID AS VendorID,
MIN(pv.StandardPrice) AS MinStandardPrice
FROM Purchasing.ProductVendor pv
GROUP BY pv.BusinessEntityID),
MostEffectiveVendor AS (
SELECT TOP 1 a.VendorID
FROM ReturnRate a
JOIN VendorCheapestPrice b
ON a.VendorID = b.VendorID
ORDER BY a.AvgOrderReturnRate ASC, b.MinStandardPrice ASC),
OrderFulfillment AS (
SELECT AVG(po.Freight + po.TaxAmt) AS AvgOrderFulfillmentCost,
AVG(DATEDIFF(day, po.OrderDate, po.ShipDate)) AS AvgOrderFulfillmentTime
FROM Purchasing.PurchaseOrderHeader po
WHERE po.VendorID = (SELECT VendorID FROM MostEffectiveVendor))
SELECT OrderFulfillment.AvgOrderFulfillmentTime, AvgOrderFulfillmentCost
FROM OrderFulfillment;
```

The results grid shows one row:

Grid	VendorID	VendorName	AvgOrderReturnRate	MinStandardPrice
1	1546	Green Lake Bike Company	0	0.8000

The status bar indicates: 1 row(s) fetched - 44ms, on 2023-09-06 at 00:53:29.

2.2.11. Calculate the average order fulfillment cost and time

Code

```
WITH ReturnRate AS (
SELECT pv.BusinessEntityID AS VendorID, AVG(pod.RejectedQty / pod.OrderQty) AS
AvgOrderReturnRate
FROM Purchasing.ProductVendor pv
JOIN Purchasing.PurchaseOrderDetail pod
ON pv.ProductID = pod.ProductID
GROUP BY pv.BusinessEntityID),
VendorCheapestPrice AS (
SELECT pv.BusinessEntityID AS VendorID,
MIN(pv.StandardPrice) AS MinStandardPrice
FROM Purchasing.ProductVendor pv
GROUP BY pv.BusinessEntityID),
MostEffectiveVendor AS (
SELECT TOP 1 a.VendorID
FROM ReturnRate a
JOIN VendorCheapestPrice b
ON a.VendorID = b.VendorID
ORDER BY a.AvgOrderReturnRate ASC, b.MinStandardPrice ASC),
OrderFulfillment AS (
SELECT AVG(po.Freight + po.TaxAmt) AS AvgOrderFulfillmentCost,
AVG(DATEDIFF(day, po.OrderDate, po.ShipDate)) AS AvgOrderFulfillmentTime
FROM Purchasing.PurchaseOrderHeader po
WHERE po.VendorID = (SELECT VendorID FROM MostEffectiveVendor))
SELECT OrderFulfillment.AvgOrderFulfillmentTime, AvgOrderFulfillmentCost
FROM OrderFulfillment;
```

Returned
table

The screenshot shows a SQL query window with the same query as above. The results grid shows one row:

Grid	AvgOrderFulfillmentTime	AvgOrderFulfillmentCost
1	25	3470.3750

The status bar indicates: 1 row(s) fetched - 90ms, on 2023-09-06 at 01:41:40.