

Zad 1.

Jaka jest łączna suma transakcji (*SalesOrderHeader.Subtotal*) w poszczególnych latach dla poszczególnych dni tygodnia.

100 %		Results Messages	
	Year	DayOfWeek	Total
1	2011	1	539444,0004
2	2011	2	3443079,9645
3	2011	3	1080780,8945
4	2011	4	1372341,7495
5	2011	5	1248302,0837
6	2011	6	2068459,2198
7	2011	7	2889264,3005
8	2012	1	7266760,7028
9	2012	2	4303006,4943
10	2012	3	5999143,0718
11	2012	4	4161627,6385
12	2012	5	2602535,6542

	Year	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
1	2011	3443079,9645	1080780,8945	1372341,7495	1248302,0837	2068459,2198	2889264,3005	539444,0004
2	2012	4303006,4943	5999143,0718	4161627,6385	2692535,6542	4570299,305	4530928,4594	7266760,7028
3	2013	6653314,1953	6228219,3549	9039179,7091	6046847,6099	3786578,9276	6065191,4772	5803147,7797
4	2014	4564125,5899	1176647,534	4001988,8004	4628859,4293	1114231,4756	3307937,8877	1264138,0944

```
SELECT
    YEAR(S.OrderDate) as "Year",
    DATEPART(WEEKDAY, S.OrderDate) as "DayOfWeek",
    SUM(S.SubTotal) as "Total"
FROM Sales.SalesOrderHeader S
GROUP BY YEAR(S.OrderDate), DATEPART(WEEKDAY, S.OrderDate)
ORDER BY YEAR(S.OrderDate), DATEPART(WEEKDAY, S.OrderDate);

--wersja pivot (dni tygodnia zamiast numerów to nazwy)
SELECT * FROM
(
    SELECT
        YEAR(S.OrderDate) as "Year",
        DATENAME(WEEKDAY, S.OrderDate) as "DayOfWeek",
        SUM(S.SubTotal) as "Total"
    FROM Sales.SalesOrderHeader S
    GROUP BY YEAR(S.OrderDate), DATENAME(WEEKDAY, S.OrderDate)
) AS SourceTable
PIVOT
(
    SUM(Total)
    FOR DayOfWeek IN ([Monday], [Tuesday], [Wednesday], [Thursday], [Friday], [Saturday], [Sunday])
) AS PivotTable
ORDER BY Year;
```

Wnioski:

Sprzedaż rosła w latach 2011-2013, osiągając szczyt w 2013, po czym pozornie spadła w 2014 (tylko pozornie, bo w praktyce dane na rok 2014 są niekompletne i są jedynie do czerwca). W 2011 w sprzedaży dominowały soboty i poniedziałki, od 2012 większe przychody zaczęły generować niedziele. W 2013 nastąpił wyjątkowy wzrost sprzedaży w środy (łącznie ponad 9mln), co może wskazywać na skuteczną strategię promocyjną. Jednak dane te nie wydają się pokazywać wiele przydatnych informacji przy takim zestawieniu, ponieważ sam wpływ dnia tygodnia na sprzedaż wydaje się nie być tak prosto przewidywalny.

Zad 2.

Zaproponuj podział klientów na 3 rozłączne grupy wiekowe (0-50, 51-60, 61+),

a) ilu różnych klientów dokonało zakupów w kolejnych miesiącach roku w każdej z grup?

b) ilu klientów w poszczególnych grupach dokonało zakupu dokładnie raz?

100 %

Results

Messages

	OrderMonth	0-50	51-60	61+
1	2011-05	NULL	1	42
2	2011-06	25	29	87
3	2011-07	4	55	172
4	2011-08	16	63	171
5	2011-09	39	29	89
6	2011-10	17	97	213
7	2011-11	37	89	104
8	2011-12	15	56	157
9	2012-01	20	74	242
10	2012-02	9	72	138
11	2012-03	19	112	173
12	2012-04	18	45	206

	AgeGroup	OneTimeBuyers
1	0-50	1496
2	51-60	3578
3	61+	6575

```
--a)
WITH XMLNAMESPACES ('http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey' as ns)
SELECT * FROM
(
SELECT
    FORMAT(S.OrderDate, 'yyyy-MM') AS OrderMonth,
    CASE
        WHEN DATEDIFF(YEAR, BD, GETDATE()) ≤ 50 THEN '0-50'
        WHEN DATEDIFF(YEAR, BD, GETDATE()) ≤ 60 THEN '51-60'
        ELSE '61+'
    END AS AgeGroup,
    COUNT(DISTINCT S.CustomerID) AS UniqueCustomers
FROM Sales.SalesOrderHeader S
    JOIN Sales.Customer C ON S.CustomerID = C.CustomerID
    JOIN Person.Person P ON C.PersonID = P.BusinessEntityID
CROSS APPLY (
SELECT P.Demographics.value('/ns:IndividualSurvey/ns:BirthDate)[1]', 'datetime') AS BD
) AS BirthData
GROUP BY FORMAT(S.OrderDate, 'yyyy-MM'), CASE
    WHEN DATEDIFF(YEAR, BD, GETDATE()) ≤ 50 THEN '0-50'
    WHEN DATEDIFF(YEAR, BD, GETDATE()) ≤ 60 THEN '51-60'
    ELSE '61+'
END
) AS SourceTable
PIVOT
(
SUM(UniqueCustomers)
FOR AgeGroup IN ([0-50], [51-60], [61+])
) AS PivotTable
ORDER BY OrderMonth;
```

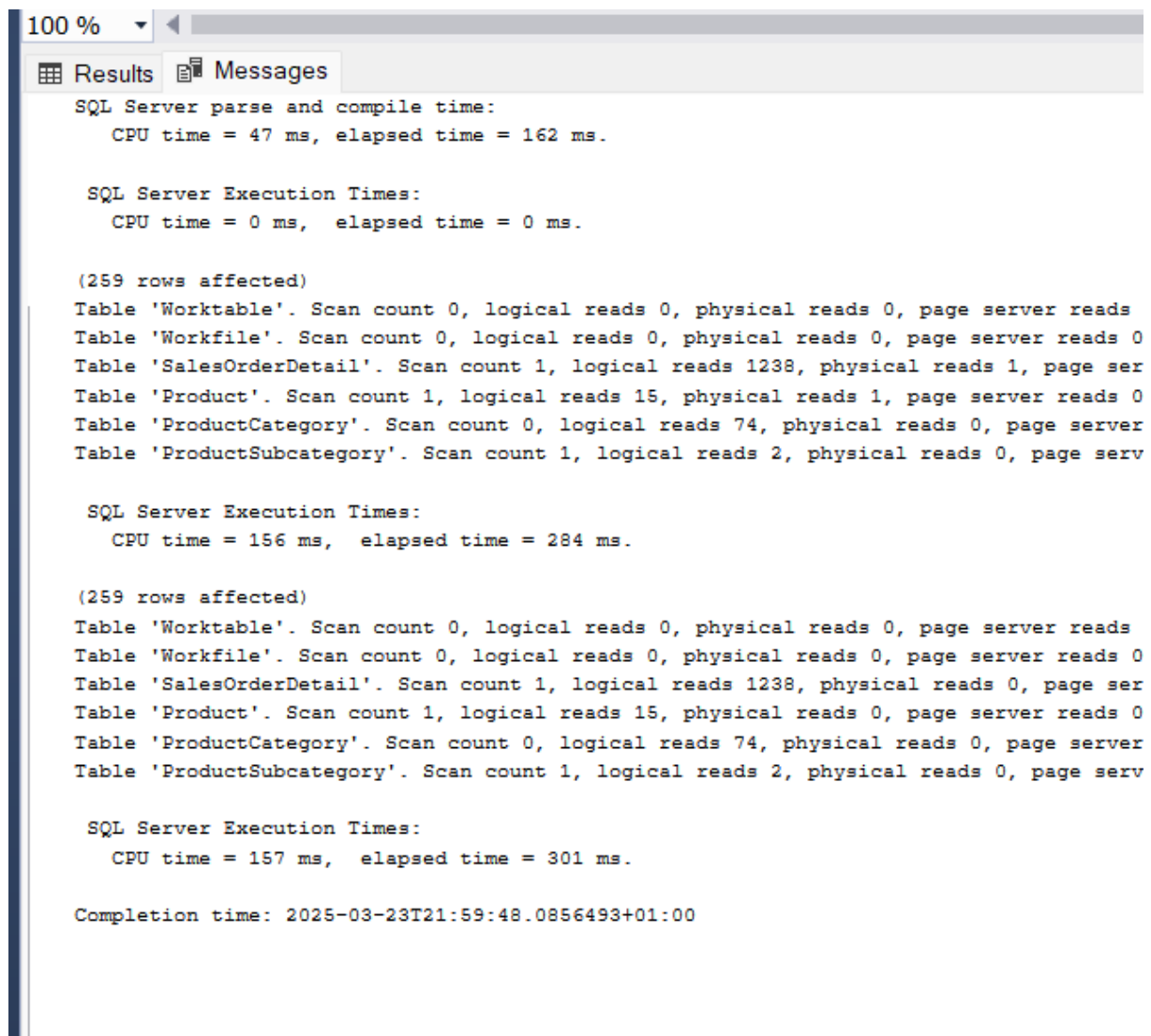
```
--b)
WITH XMLNAMESPACES ('http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/IndividualSurvey' as ns),
CustomerOrders AS (
SELECT
    S.CustomerID,
    COUNT(*) AS OrderCount,
    CASE
        WHEN DATEDIFF(YEAR, BD, GETDATE()) ≤ 50 THEN '0-50'
        WHEN DATEDIFF(YEAR, BD, GETDATE()) ≤ 60 THEN '51-60'
        ELSE '61+'
    END AS AgeGroup
FROM Sales.SalesOrderHeader S
    JOIN Sales.Customer C ON S.CustomerID = C.CustomerID
    JOIN Person.Person P ON C.PersonID = P.BusinessEntityID
CROSS APPLY (
SELECT P.Demographics.value('/ns:IndividualSurvey/ns:BirthDate)[1]', 'datetime') AS BD
) AS BirthData
GROUP BY
    S.CustomerID,
    CASE
        WHEN DATEDIFF(YEAR, BD, GETDATE()) ≤ 50 THEN '0-50'
        WHEN DATEDIFF(YEAR, BD, GETDATE()) ≤ 60 THEN '51-60'
        ELSE '61+'
    END
END
)
SELECT
    AgeGroup,
    COUNT(*) AS OneTimeBuyers
FROM CustomerOrders
WHERE OrderCount = 1
GROUP BY AgeGroup;
```

Wnioski:

Klienci tego sklepu nie należą do najmłodszych. Znaczna ich część należy do kategorii wiekowej 61+, a także jest bardzo niewielu klientów w kategorii wiekowej 0-50 (większość z nich jest bliżej górnej granicy). Widać więc że sklep ten jest skierowany raczej do starszej klienteli.

Zad 3.

Zestawienie produktów, których sprzedaje się miesięcznie co najmniej 20 sztuk. Dla każdego produktu podaj jego kategorie. Jeżeli warto użyć CTE to porównaj efektywność rozwiązania z wersją i bez CTE.



The screenshot displays the SQL Server Enterprise Manager interface. At the top, there is a progress bar at 100% and two tabs: 'Results' and 'Messages'. The 'Messages' tab is active, showing the following text:

```
SQL Server parse and compile time:
  CPU time = 47 ms, elapsed time = 162 ms.

SQL Server Execution Times:
  CPU time = 0 ms,  elapsed time = 0 ms.

(259 rows affected)
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0
Table 'SalesOrderDetail'. Scan count 1, logical reads 1238, physical reads 1, page ser
Table 'Product'. Scan count 1, logical reads 15, physical reads 1, page server reads 0
Table 'ProductCategory'. Scan count 0, logical reads 74, physical reads 0, page server
Table 'ProductSubcategory'. Scan count 1, logical reads 2, physical reads 0, page serv

SQL Server Execution Times:
  CPU time = 156 ms,  elapsed time = 284 ms.

(259 rows affected)
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0
Table 'SalesOrderDetail'. Scan count 1, logical reads 1238, physical reads 0, page ser
Table 'Product'. Scan count 1, logical reads 15, physical reads 0, page server reads 0
Table 'ProductCategory'. Scan count 0, logical reads 74, physical reads 0, page server
Table 'ProductSubcategory'. Scan count 1, logical reads 2, physical reads 0, page serv

SQL Server Execution Times:
  CPU time = 157 ms,  elapsed time = 301 ms.

Completion time: 2025-03-23T21:59:48.0856493+01:00
```

100 %

Results Messages

	ProductID	Product	Category	Sold
1	781	Mountain-200 Silver, 46	Bikes	2216
2	877	Bike Wash - Dissolver	Accessories	3319
3	802	LL Fork	Components	190
4	745	HL Mountain Frame - Black, 48	Components	131
5	902	LL Touring Frame - Yellow, 58	Components	36
6	885	HL Touring Frame - Yellow, 60	Components	387
7	978	Touring-3000 Blue, 44	Bikes	292
8	828	HL Road Rear Wheel	Components	94
9	964	Touring-3000 Yellow, 58	Bikes	293
10	929	ML Mountain Tire	Accessories	1161

	ProductID	Product	Category	Sold
1	781	Mountain-200 Silver, 46	Bikes	2216
2	877	Bike Wash - Dissolver	Accessories	3319
3	802	LL Fork	Components	190
4	745	HL Mountain Frame - Black, 48	Components	131
5	902	LL Touring Frame - Yellow, 58	Components	36
6	885	HL Touring Frame - Yellow, 60	Components	387
7	978	Touring-3000 Blue, 44	Bikes	292
8	828	HL Road Rear Wheel	Components	94
9	964	Touring-3000 Yellow, 58	Bikes	293
10	929	ML Mountain Tire	Accessories	1161
11	814	ML Mountain Frame - Black, ...	Components	514
12	864	Classic Vest, S	Clothing	4247
13	750	Road-150 Red, 44	Bikes	437
14	883	Short-Sleeve Classic Jersey, L	Clothing	2848
15	783	Mountain-200 Black, 42	Bikes	2664
16	922	Road Tire Tube	Accessories	2376
17	711	Sport-100 Helmet, Blue	Accessories	6743

```

SET STATISTICS IO, TIME ON

WITH Products AS
(
    SELECT
        PP.ProductID as "ProductID",
        PP.Name as "Product",
        PC.Name as "Category",
        SUM(SOD.OrderQty) as "Sold"
    FROM Production.Product PP
        JOIN Production.ProductSubcategory PSC ON PP.ProductSubcategoryID = PSC.ProductSubcategoryID
        JOIN Production.ProductCategory PC ON PSC.ProductCategoryID = PC.ProductCategoryID
        JOIN Sales.SalesOrderDetail SOD ON PP.ProductID = SOD.ProductID
        JOIN Sales.SalesOrderHeader SOH ON SOD.SalesOrderID = SOH.SalesOrderID
    GROUP BY PP.ProductID, PP.Name, PC.Name
    HAVING SUM(SOD.OrderQty) ≥ 20
)
SELECT * FROM Products;

SELECT * FROM
(
    SELECT
        PP.ProductID as "ProductID",
        PP.Name as "Product",
        PC.Name as "Category",
        SUM(SOD.OrderQty) as "Sold"
    FROM Production.Product PP
        JOIN Production.ProductSubcategory PSC ON PP.ProductSubcategoryID = PSC.ProductSubcategoryID
        JOIN Production.ProductCategory PC ON PSC.ProductCategoryID = PC.ProductCategoryID
        JOIN Sales.SalesOrderDetail SOD ON PP.ProductID = SOD.ProductID
        JOIN Sales.SalesOrderHeader SOH ON SOD.SalesOrderID = SOH.SalesOrderID
    GROUP BY PP.ProductID, PP.Name, PC.Name
    HAVING SUM(SOD.OrderQty) ≥ 20
) AS SourceTable;

```

Wnioski:

W tym konkretnym przypadku oba zapytania są tak samo wydajne, ponieważ silnik SQL Server zoptymalizuje je w ten sam sposób. CTE byłoby lepsze, gdybyśmy chcieli się wielokrotnie odwoływać do tymczasowej tabeli w ramach tego samego zapytania. CTE może być preferowane w złożonych zapytaniach dla lepszej czytelności kodu.

Zad 4.

Przygotuj zestawienie, w którym przeanalizujesz, ilu jest różnych klientów dla każdej płci w kolejnych miesiącach (05.2011 – 06.2014). Jak procentowo rozkłada się ich udział w całkowitej wartości sprzedaży (Sales.SalesOrderHeader.TotalDue)?

100 %

Results Messages

	OrderMonthYear	Gender	TotalSales	SalesPercentage	UniqueCustomers
1	2011-05	F	11270,967	70.454732842259511	3
2	2011-05	M	4726,492	29.545267157740488	2
3	2011-06	F	231858,2298	45.722706422552510	64
4	2011-06	M	275238,2392	54.277293577447489	77
5	2011-07	F	326154,6192	58.310429032760166	94
6	2011-07	M	233187,208	41.689570967239833	62
7	2011-08	F	257604,0412	48.047499941390803	72
8	2011-08	M	278540,4856	51.952500058609196	78
9	2011-09	F	243348,8502	43.863109427616608	71
10	2011-09	M	311442,758	56.136890572383391	86
11	2011-10	F	302340,6538	48.712919871717791	83
12	2011-10	M	318317,3864	51.287080128282208	91
13	2011-11	F	407252,994	49.950508137061811	113
14	2011-11	M	408060,0212	50.049491862938188	117
15	2011-12	F	348250,6674	52.812867789736701	98
16	2011-12	M	311151,2222	47.187132221322222	82

```

WITH CustomerGender AS (
    SELECT
        c.CustomerID,
        Demographics.value('declare default element namespace "http://schemas.microsoft.com/sqlserver/2004/07/
FROM Sales.Customer c JOIN Person.Person p ON c.PersonID = p.BusinessEntityID
WHERE c.PersonID IS NOT NULL AND Demographics.value('declare default element namespace "http://schemas.mic
),
CustomersMonthly AS (
    SELECT
        FORMAT(soh.OrderDate, 'yyyy-MM') AS OrderMonthYear,
        cg.Gender,
        COUNT(DISTINCT soh.CustomerID) AS UniqueCustomers,
        SUM(soh.TotalDue) AS TotalSales
    FROM Sales.SalesOrderHeader soh JOIN CustomerGender cg ON soh.CustomerID = cg.CustomerID
    WHERE soh.OrderDate BETWEEN '2011-05-01' AND '2014-06-30'
    GROUP BY FORMAT(soh.OrderDate, 'yyyy-MM'), cg.Gender
)
SELECT
    OrderMonthYear,
    Gender,
    TotalSales,
    (TotalSales * 100.0 / SUM(TotalSales) OVER (PARTITION BY OrderMonthYear)) AS SalesPercentage,
    UniqueCustomers
FROM CustomersMonthly
ORDER BY OrderMonthYear, Gender;

```

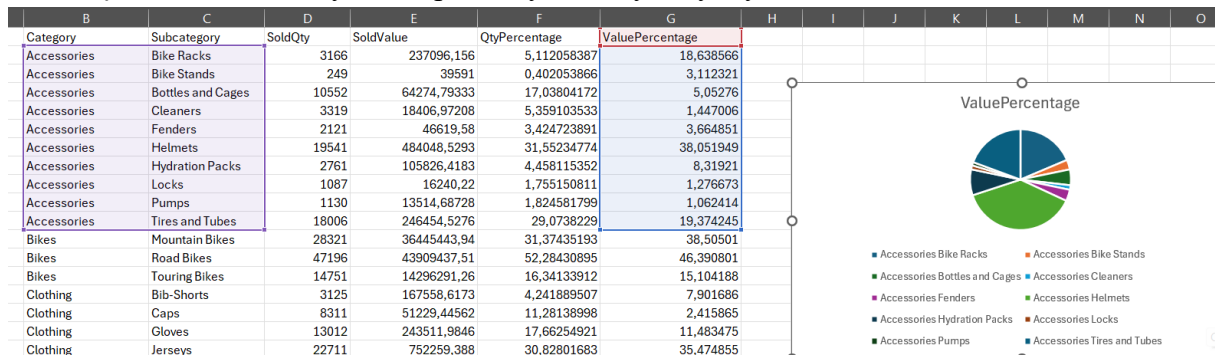
Wnioski:

Dla każdej płci jest bardzo podobna ilość klientów (około 50%) i nie widać tutaj żadnego trendu na którąś z płci. Widać standardowy nagły wzrost w wielkości liczb w roku 2013 (tak jak w poprzednich zadaniach spowodowany ten wzrost jest lepszym dla firmy okresem sprzedażowym). Warto dodać że dużo osób ma wpisaną płć jako NULL, więc dane zostały przefiltrowane, tak żeby osoby bez wpisanej płci wykluczyć z zestawienia dla zwiększenia jego czytelności.

Zad 5.

100 %						
Results Messages						
	Category	Subcategory	SoldQty	SoldValue	QtyPercentage	ValuePercentage
1	Accessories	Bike Racks	3166	237096.156000	5.112058386617	18.638566
2	Accessories	Bike Stands	249	39591.000000	0.402053865529	3.112321
3	Accessories	Bottles and Cages	10552	64274.793327	17.038041723180	5.052760
4	Accessories	Cleaners	3319	18406.972080	5.359103532907	1.447006
5	Accessories	Fenders	2121	46619.580000	3.424723890718	3.664851
6	Accessories	Helmets	19541	484048.529277	31.552347736226	38.051949
7	Accessories	Hydration Packs	2761	105826.418334	4.458115352321	8.319210
8	Accessories	Locks	1087	16240.220000	1.755150810566	1.276673
9	Accessories	Pumps	1130	13514.687276	1.824581799392	1.062414
10	Accessories	Tires and Tubes	18006	246454.527632	29.073822902538	19.374245
11	Bikes	Mountain Bikes	28321	36445443.937380	31.374351929809	38.505010
12	Bikes	Road Bikes	47196	43909437.508212	52.284308946692	46.390801
13	Bikes	Touring Bikes	14751	14296291.259139	16.341339123498	15.104188
14	Clothing	Bib-Shorts	3125	167558.617307	4.241889507262	7.901686
15	Clothing	Caps	8311	51229.445623	11.281389982353	2.415865
16	Clothing	Gloves	13012	243511.984567	17.662549205918	11.483475
17	Clothing	Jerseys	22711	752259.388000	30.828016831817	35.474855

Dane są w formacie, który łatwo przełożyć na czytelny wykres:




```

WITH ProductSales AS (
    SELECT
        PC.Name AS Category,
        PSC.Name AS Subcategory,
        SUM(SOD.OrderQty) AS SoldQty,
        SUM(SOD.LineTotal) AS SoldValue
    FROM Production.Product PP
        JOIN Production.ProductSubcategory PSC ON PP.ProductSubcategoryID = PSC.ProductSubcategoryID
        JOIN Production.ProductCategory PC ON PSC.ProductCategoryID = PC.ProductCategoryID
        JOIN Sales.SalesOrderDetail SOD ON PP.ProductID = SOD.ProductID
    GROUP BY PC.Name, PSC.Name
),
CategorySales AS (
    SELECT
        PC.Name AS Category,
        SUM(SOD.OrderQty) AS SoldQty,
        SUM(SOD.LineTotal) AS SoldValue
    FROM Production.Product PP
        JOIN Production.ProductSubcategory PSC ON PP.ProductSubcategoryID = PSC.ProductSubcategoryID
        JOIN Production.ProductCategory PC ON PSC.ProductCategoryID = PC.ProductCategoryID
        JOIN Sales.SalesOrderDetail SOD ON PP.ProductID = SOD.ProductID
    GROUP BY PC.Name
)
SELECT
    PS.Category,
    PS.Subcategory,
    PS.SoldQty,
    PS.SoldValue,
    (PS.SoldQty * 100.0 / CS.SoldQty) AS QtyPercentage,
    (PS.SoldValue * 100.0 / CS.SoldValue) AS ValuePercentage
FROM ProductSales PS
    JOIN CategorySales CS ON PS.Category = CS.Category
ORDER BY PS.Category, PS.Subcategory;

```

Wnioski:

W kategorii Accessories dominują kaski – mają one zarówno największy udział ilościowy (31,55%), jak i wartościowy (38,05%), inne produkty takie jak butelki czy bagażniki prezentują znaczne różnice między udziałem ilościowym a wartością i są daleko z tyłu za kaskami.

W kategorii Bikes dominuje segment rowerów szosowych, który odpowiada za ponad połowę sprzedaży pod względem ilości i blisko połowy wartości, z rowerami górskimi zajmującymi drugie miejsce.

W kategorii Clothing najważniejsze są koszulki (jerseys), które osiągają największe udziały zarówno ilościowe (30,83%), jak i wartość (35,47%), zaś akcesoria takie jak czapki, mimo wyższego udziału ilościowego, generują znacznie mniejszą wartość.

W kategorii Components największy udział mają ramy – zarówno górskie, jak i szosowe, które łącznie stanowią ponad 47% sprzedaży ilościowej i aż 72% wartości sprzedaży, co podkreśla ich kluczowe znaczenie w strukturze sprzedaży.

Zad 6.

100 %				
Results Messages				
	Region	USD	EUR	PLN
1	AU	11804417,2776	NULL	NULL
2	CA	18208668,1173	NULL	NULL
3	DE	2544081,0459	NULL	NULL
4	FR	754377,1051	NULL	NULL
5	GB	8569134,9366	NULL	NULL
6	US	944,2676	NULL	NULL

	Currency	Territory
1	NULL	Australia
2	NULL	Canada
3	NULL	Central
4	NULL	France
5	NULL	Germany
6	NULL	Northeast
7	NULL	Northwest
8	NULL	Southeast
9	NULL	Southwest
10	NULL	United K...
11	USD	Australia
12	USD	Canada
13	USD	France
14	USD	Germany
15	USD	Northwest
16	USD	Southwest
17	USD	United K...

```

SELECT * FROM
(
    SELECT
        st.CountryRegionCode AS Region,
        cr.FromCurrencyCode AS Currency,
        SUM(soh.TotalDue) AS TotalSales
    FROM Sales.SalesOrderHeader soh
        JOIN Sales.SalesTerritory st ON soh.TerritoryID = st.TerritoryID
        JOIN Sales.CurrencyRate cr ON soh.CurrencyRateID = cr.CurrencyRateID
    GROUP BY st.CountryRegionCode, cr.FromCurrencyCode
) AS SourceTable
PIVOT
(
    SUM(TotalSales)
    FOR Currency IN ([USD], [EUR], [PLN])
) AS PivotTable
ORDER BY Region;

SELECT
    cr.FromCurrencyCode AS Currency,
    st.Name AS Territory
FROM Sales.SalesOrderHeader soh
    LEFT JOIN Sales.SalesTerritory st ON soh.TerritoryID = st.TerritoryID
    LEFT JOIN Sales.CurrencyRate cr ON soh.CurrencyRateID = cr.CurrencyRateID
GROUP BY cr.FromCurrencyCode, st.Name
ORDER BY cr.FromCurrencyCode, st.Name;

```

Wnioski:

Jedyna waluta w jakiej występują wszystkie opłaty to USD. W każdym z Territory jest ona obowiązującą walutą.