

Kajetan Pynka 254495, wykład 2

Zad. 1

```
SELECT *
FROM (SELECT TotalDue, Year(OrderDate) 'rok',
DATENAME(WEEKDAY, OrderDate) 'Dzień tygodnia'
FROM Sales.SalesOrderHeader) S
PIVOT(SUM(TotalDue) FOR S.rok IN ([2011], [2012], [2013],
[2014])) AS X
ORDER BY CASE
WHEN X.[Dzień tygodnia]='Sunday' THEN 7
WHEN X.[Dzień tygodnia]='Monday' THEN 1
WHEN X.[Dzień tygodnia]='Tuesday' THEN 2
WHEN X.[Dzień tygodnia]='Wednesday' THEN 3
WHEN X.[Dzień tygodnia]='Thursday' THEN 4
WHEN X.[Dzień tygodnia]='Friday' THEN 5
WHEN X.[Dzień tygodnia]='Saturday' THEN 6
END;
```

	Dzień tygodnia	2011	2012	2013	2014
1	Monday	3865613,6434	4870867,6376	7459515,8805	5117403,0203
2	Tuesday	1204578,3096	6739285,0517	6981904,0913	1300206,5572
3	Wednesday	1534332,0656	4672792,0455	10174665,762	4479974,5721
4	Thursday	1394423,8072	3013623,9644	6793457,2152	5190897,0251
5	Friday	2318547,3174	5126143,8398	4234112,4302	1231252,9021
6	Saturday	3242118,752	5087536,6061	6798070,3767	3702891,2488
7	Sunday	596085,6298	8165451,1669	6524162,2073	1396872,9901

Wniosek: Suma transakcji w następujących po sobie latach nie wydaje się w jakikolwiek sposób zależeć od dnia tygodnia (ogólnie poniedziałek wydaje się dość obfity w transakcje, reszta dni na przestrzeni lat bardzo mocno się waha).

Zad. 2

```
SELECT X.Miesiąc, X.[1] '<45', X.[2] '45-60', X.[3] '>60'
      FROM (SELECT DISTINCT SOH.CustomerID, DATENAME(MONTH,
SOH.OrderDate) 'Miesiąc',
      (CASE WHEN DATEDIFF(YEAR, PD.BirthDate, SYSDATETIME()) < 45
THEN 1
      WHEN DATEDIFF(YEAR, PD.BirthDate, SYSDATETIME()) < 60 THEN
2
      ELSE 3 END) 'wiekGrupa'
      FROM Sales.SalesOrderHeader SOH
      JOIN Sales.Customer C ON C.CustomerID=SOH.CustomerID
      JOIN Person.Person P ON P.BusinessEntityID=C.PersonID
      JOIN Sales.vPersonDemographics PD ON
PD.BusinessEntityID=P.BusinessEntityID) T
      PIVOT(COUNT(CustomerID) FOR T.wiekGrupa IN([1], [2], [3])) AS
X
      ORDER BY CASE
      WHEN X.Miesiąc='January' THEN 1
      WHEN X.Miesiąc='February' THEN 2
      WHEN X.Miesiąc='March' THEN 3
      WHEN X.Miesiąc='April' THEN 4
      WHEN X.Miesiąc='May' THEN 5
      WHEN X.Miesiąc='June' THEN 6
      WHEN X.Miesiąc='July' THEN 7
      WHEN X.Miesiąc='August' THEN 8
      WHEN X.Miesiąc='September' THEN 9
      WHEN X.Miesiąc='October' THEN 10
      WHEN X.Miesiąc='November' THEN 11
      WHEN X.Miesiąc='December' THEN 12
      END;
```

	Miesiąc	<45	45-60	>60
1	January	111	1107	1402
2	February	110	968	1078
3	March	136	1315	1405
4	April	96	1229	1329
5	May	155	1163	1639
6	June	104	828	1080
7	July	85	931	1176
8	August	90	976	1101
9	September	121	909	1098
10	October	115	999	1277
11	November	150	1180	1207
12	December	79	974	1350

Wniosek: Bardzo duży odsetek klientów stanowią osoby w wieku 45+. Na przestrzeni wszystkich grup, wydaje się, że klienci kupują najwięcej w marcu i maju, najmniej w czerwcu.

Zad. 3

```
SELECT * FROM
    (SELECT PC.Name, ST.CountryRegionCode, P.ProductID FROM
Sales.SalesOrderDetail SOD
    JOIN Production.Product P ON P.ProductID=SOD.ProductID
    JOIN Production.ProductSubcategory PSC ON
PSC.ProductSubcategoryID=P.ProductSubcategoryID
    JOIN Production.ProductCategory PC ON
PC.ProductCategoryID=PSC.ProductCategoryID
    JOIN Sales.SalesOrderHeader SOH ON
SOH.SalesOrderID=SOD.SalesOrderID
    JOIN Sales.SalesTerritory ST ON
ST.TerritoryID=SOH.TerritoryID) Z
    PIVOT(COUNT(ProductID) FOR Name IN([Accessories], [Bikes],
[Clothing], [Components])) AS X;
```

	CountryRegionCode	Accessories	Bikes	Clothing	Components
1	AU	7178	5241	2152	487
2	CA	6392	5366	3872	3434
3	DE	3519	2255	1206	548
4	FR	3668	2618	1577	1225
5	GB	4392	3185	1750	1099
6	US	16045	21366	10837	11905

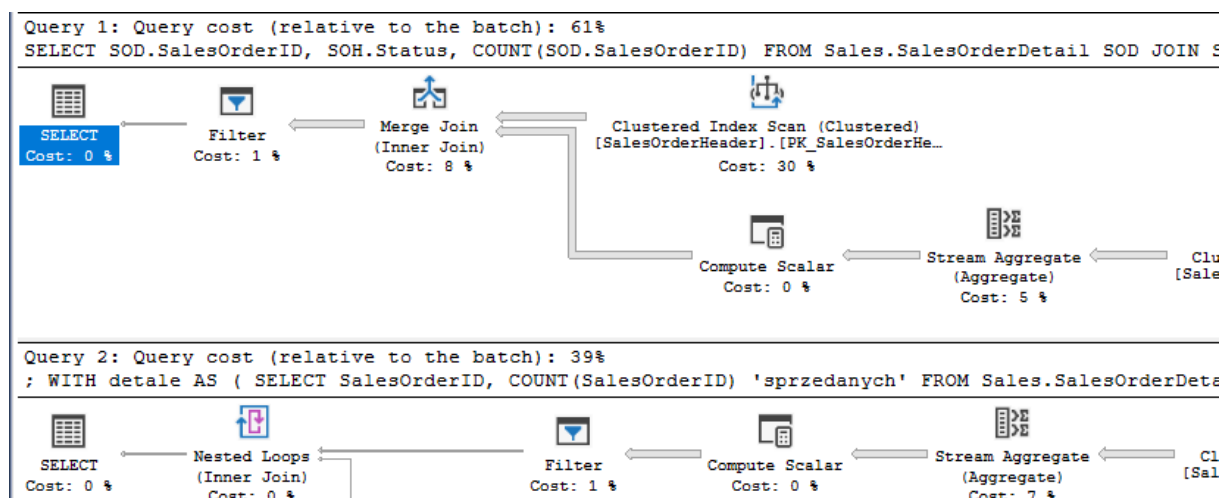
Wniosek: Ogólnie najwięcej produktów sprzedaje się w Stanach Zjednoczonych. W Stanach również najwięcej sprzedanych zostało rowerów, podczas gdy w innych regionach najczęściej kupowane były akcesoria. Części do rowerów sprzedają się lepiej od ubrań w Stanach natomiast w pozostałej części świata to ubrania sprzedają się dużo lepiej. Stany Zjednoczone a reszta świata zdecydowanie się różnią jeśli chodzi o sprzedaż produktów z poszczególnych kategorii.

CTE:

Ułożyłem swoje zapytanie do wykorzystania CTE, ponieważ PIVOT'y z poprzednich zadań stwarzały problemy do pokazania różnicy wydajnościowej.

```
SELECT SOD.SalesOrderID, SOH.Status, COUNT(SOD.SalesOrderID) FROM
Sales.SalesOrderDetail SOD
    JOIN Sales.SalesOrderHeader SOH ON
SOD.SalesOrderID=SOH.SalesOrderID
    GROUP BY SOD.SalesOrderID, SOH.Status HAVING
COUNT(SOD.SalesOrderID)>10;

WITH detale AS (
    SELECT SalesOrderID, COUNT(SalesOrderID) 'sprzedanych' FROM
Sales.SalesOrderDetail
    GROUP BY SalesOrderID HAVING COUNT(SalesOrderID)>10
) SELECT D.SalesOrderID, SOH.Status, D.sprzedanych FROM detale D
    JOIN Sales.SalesOrderHeader SOH ON
D.SalesOrderID=SOH.SalesOrderID;
```



Wniosek: Nie jest to niesamowita różnica, ale już na tym przykładzie widać oszczędność jeśli chodzi o operację złączenia. Dołączenie SalesOrderHeader po wcześniejszym odfiltrowaniu rekordów zapewnia około 20% większą wydajność zapytania.