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

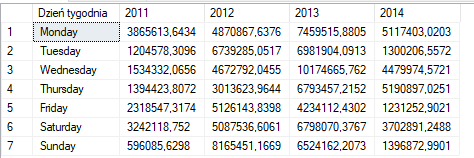
        WHEN X.[Dzień tygodnia]='Wednesday' THEN 3

        WHEN X.[Dzień tygodnia]='Thursday'THEN 4

        WHEN X.[Dzień tygodnia]='Friday' THEN 5

        WHEN X.[Dzień tygodnia]='Saturday' THEN 6

    END;



**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

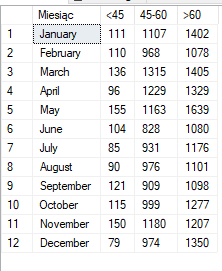
        WHEN X.Miesiąc='September' THEN 9

        WHEN X.Miesiąc='October' THEN 10

        WHEN X.Miesiąc='November' THEN 11

        WHEN X.Miesiąc='December' THEN 12

    END;

**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

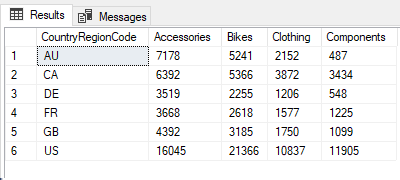
    JOIN Production.ProductSubcategory PSC ON PSC.ProductSubcategoryID=P.ProductSubcategoryID

    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;



**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;

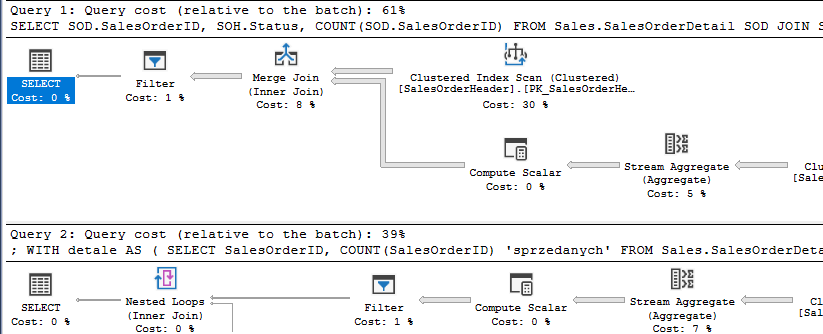
WITH detale AS (

    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.