

Hurtownie Danych - laboratorium

Lista 2

Podstawy SQL: PIVOT i CASE

Analiza jakości i profilowanie danych

Wstęp teoretyczny

Polecenie PIVOT umożliwia prezentowanie wyniku zapytania w postaci tabeli przestawnej. Stosuje się następującą składnię polecenia PIVOT:

```
SELECT <non-pivoted column>,  
    [first pivoted column] AS <column name>,  
    [second pivoted column] AS <column name>,  
    ...  
    [last pivoted column] AS <column name>  
FROM  
    (<SELECT query that produces the data>  
    AS <alias for the source query>  
PIVOT  
    (  
        <aggregation function>(<column being aggregated>)  
FOR  
        [<column that contains the values that will become column headers>  
        IN ( [first pivoted column], [second pivoted column],  
        ... [last pivoted column])  
    ) AS <alias for the pivot table>  
    <optional ORDER BY clause>;
```

Polecenie CASE sprawdza określony warunek w postaci wyrażenia logicznego i zwraca jeden z możliwych wyników. Wynikiem jest pierwszy przypadek, dla którego spełniony jest warunek.

```
Searched CASE expression:  
CASE  
    WHEN Boolean_expression THEN result_expression [ ...n ]  
    [ ELSE else_result_expression ]  
END
```

Możliwe jest również porównanie wyrażenia input_expression z kolejnymi warunkami when_expression w celu określenia wyniku result_expression.

```
Simple CASE expression:  
CASE input_expression  
    WHEN when_expression THEN result_expression [ ...n ]  
    [ ELSE else_result_expression ]  
END
```

Źródła:

<https://docs.microsoft.com/en-us/sql/t-sql/queries/from-using-pivot-and-unpivot?view=sql-server-ver15>
<https://docs.microsoft.com/en-us/sql/t-sql/language-elements/case-transact-sql?view=sql-server-ver15>

Zadania do wykonaniaBaza danych: **AdventureWorks****Rozwiązania:****Zad. 1. Ekstrakcja danych**

1. Utworzyć tabelę Sprzedaz zawierającą dane dotyczące liczby sprzedanych produktów przez pracowników firmy AdventureWorks w poszczególnych latach zgodnie z podanym schematem:

Sprzedaz(pracID, prodID, "Nazwa produktu", Rok, Liczba).

Wykonać to samo zapytanie przy użyciu polecenia pivot:

- a. w kolejnych kolumnach wyświetlić lata;
- b. w kolejnych kolumnach wyświetlić 5 najlepszych produktów.

```
CREATE TABLE Sprzedaz (
    pracID INT,
    prodID INT,
    Nazwa_produkту VARCHAR(255),
    Rok INT,
    Liczba INT,
    PRIMARY KEY (pracID, prodID, Rok)
);
```

```
INSERT INTO [dbo].[Sprzedaz]
SELECT
    sp.BusinessEntityID,
    prod.ProductID,
    prod.Name,
    YEAR(soh.OrderDate),
    SUM(sod.OrderQty)
FROM
    Sales.SalesPerson sp
JOIN Sales.SalesOrderHeader soh ON sp.BusinessEntityID = soh.SalesPersonID
JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID
JOIN Production.Product prod ON sod.ProductID = prod.ProductID
GROUP BY
    sp.BusinessEntityID, prod.ProductID, prod.Name, YEAR(soh.OrderDate);
```

	pracID	prodID	Nazwa produktu	Rok	Liczba
1	274	707	Sport-100 Helmet, Red	2011	1
2	274	707	Sport-100 Helmet, Red	2012	21
3	274	707	Sport-100 Helmet, Red	2013	38
4	274	707	Sport-100 Helmet, Red	2014	19
5	274	708	Sport-100 Helmet, Black	2011	1
6	274	708	Sport-100 Helmet, Black	2012	15
7	274	708	Sport-100 Helmet, Black	2013	39
8	274	708	Sport-100 Helmet, Black	2014	16
9	274	709	Mountain Bike Socks, M	2012	18
10	274	711	Sport-100 Helmet, Blue	2011	1
11	274	711	Sport-100 Helmet, Blue	2012	21
12	274	711	Sport-100 Helmet, Blue	2013	28
13	274	711	Sport-100 Helmet, Blue	2014	18
14	274	712	AWC Logo Cap	2012	27
15	274	712	AWC Logo Cap	2013	39
16	274	712	AWC Logo Cap	2014	15

a)

```

DROP TABLE Sprzedaz;
DECLARE @cols NVARCHAR (MAX);
SELECT @cols = COALESCE (@cols + ',' + CAST(rok AS VARCHAR) + ' ] INT',
                        '[' + CAST(rok AS VARCHAR) + ' ] INT')
FROM (SELECT DISTINCT YEAR(OrderDate) AS rok FROM [Sales].[SalesOrder-
Header]) PV
ORDER BY rok

DECLARE @query NVARCHAR(MAX)
SET @query = '
CREATE TABLE Sprzedaz (
    pracID INT,
    prodID INT,
    [Nazwa produktu] VARCHAR(255),
    ' + @cols + '
    PRIMARY KEY (pracID, prodId)
);'

EXEC SP_EXECUTESQL @query;

DECLARE @cols2 NVARCHAR (MAX);
SELECT @cols2 = COALESCE (@cols2 + ',' + CAST(rok AS VARCHAR) + ''',
                        '' + CAST(rok AS VARCHAR) + ''')
FROM (SELECT DISTINCT YEAR(OrderDate) AS rok FROM [Sales].[SalesOrder-
Header]) PV
ORDER BY rok

DECLARE @queryInsert NVARCHAR(MAX)
SET @queryInsert = '
INSERT INTO Sprzedaz
SELECT BusinessEntityID AS PracId, ProductID AS ProdId, Name AS "Nazwa produktu",
    COALESCE([2011], 0) AS [2011],
    COALESCE([2012], 0) AS [2012],
    COALESCE([2013], 0) AS [2013],
    COALESCE([2014], 0) AS [2014]
FROM (SELECT
    sp.BusinessEntityID,
    prod.ProductID,
    prod.Name,
    YEAR(soh.OrderDate) AS rok,
    SUM(sod.OrderQty) As Suma
FROM
    Sales.SalesPerson sp
    JOIN Sales.SalesOrderHeader soh ON sp.BusinessEntityID = soh.SalesPersonID
    JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID
    JOIN Production.Product prod ON sod.ProductID = prod.ProductID
GROUP BY
    sp.BusinessEntityID, prod.ProductID, prod.Name, YEAR(soh.OrderDate)) AS X
PIVOT (
    MIN(X.Suma)
    FOR X.rok IN (' + @cols2 + ')
)AS Y'

EXEC SP_EXECUTESQL @queryInsert;
SELECT * FROM Sprzedaz;

```

	pracID	prodID	Nazwa produktu	2011	2012	2013	2014
1	274	707	Sport-100 Helmet, Red	1	21	38	19
2	274	708	Sport-100 Helmet, Black	1	15	39	16
3	274	709	Mountain Bike Socks, M	0	18	0	0
4	274	711	Sport-100 Helmet, Blue	1	21	28	18
5	274	712	AWC Logo Cap	0	27	39	15
6	274	714	Long-Sleeve Logo Jersey, M	1	13	22	13
7	274	715	Long-Sleeve Logo Jersey, L	4	24	42	18
8	274	716	Long-Sleeve Logo Jersey, XL	0	12	5	0
9	274	717	HL Road Frame - Red, 62	0	0	3	0
10	274	718	HL Road Frame - Red, 44	0	0	4	0
11	274	722	LL Road Frame - Black, 58	0	1	5	0
12	274	725	LL Road Frame - Red, 44	0	1	3	0
13	274	726	LL Road Frame - Red, 48	0	3	2	0
14	274	729	LL Road Frame - Red, 60	0	9	1	0
15	274	730	LL Road Frame - Red, 62	0	7	2	0

b)

```

SELECT *
FROM (
SELECT *
FROM (
SELECT
    sp.BusinessEntityID as PracId,
    YEAR(soh.OrderDate) AS Rok,
    ROW_NUMBER() OVER (
        PARTITION BY sp.BusinessEntityID, YEAR(soh.OrderDate)
        ORDER BY sod.OrderQty DESC
    ) AS RowNum,
    prod.Name
FROM
    Sales.SalesPerson sp
    JOIN Sales.SalesOrderHeader soh ON sp.BusinessEntityID = soh.SalesPersonID
    JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID
    JOIN Production.Product prod ON sod.ProductID = prod.ProductID
) AS SalesData
WHERE RowNum <= 5 ) AS X
PIVOT (
    MAX(Name) FOR RowNum IN ([1], [2], [3], [4], [5])
) AS P
ORDER BY PracId, Rok;

```

	PracId	Rok	1	2	3	4	5
1	274	2011	Long-Sleeve Logo Jersey, L	Road-650 Black, 52	Road-650 Red, 60	Mountain-100 Black, 42	Mountain-100 Silver, 48
2	274	2012	Women's Tights, S	Road-650 Red, 60	AWC Logo Cap	Men's Sports Shorts, M	Women's Tights, L
3	274	2013	Short-Sleeve Classic Jersey, XL	Sport-100 Helmet, Black	AWC Logo Cap	Racing Socks, L	Road-650 Red, 44
4	274	2014	Women's Mountain Shorts, S	Classic Vest, S	Classic Vest, S	Water Bottle - 30 oz.	Sport-100 Helmet, Red
5	275	2011	Long-Sleeve Logo Jersey, L	Mountain Bike Socks, M	Road-450 Red, 58	Road-650 Red, 60	Road-650 Red, 62
6	275	2012	Full-Finger Gloves, L	Full-Finger Gloves, L	Full-Finger Gloves, M	Full-Finger Gloves, L	Full-Finger Gloves, M
7	275	2013	Full-Finger Gloves, L	Classic Vest, S	Women's Mountain Shorts, S	Women's Mountain Shorts, L	Racing Socks, L
8	275	2014	Racing Socks, L	Women's Mountain Shorts, S	Classic Vest, S	Racing Socks, L	Women's Mountain Shorts, L
9	276	2011	Mountain Bike Socks, M	Mountain-100 Silver, 38	Mountain-100 Silver, 42	Mountain-100 Black, 42	Mountain-100 Black, 38
10	276	2012	Full-Finger Gloves, L	Full-Finger Gloves, L	Long-Sleeve Logo Jersey, L	Full-Finger Gloves, M	Women's Tights, S
11	276	2013	Full-Finger Gloves, L	Women's Mountain Shorts, S	Women's Mountain Shorts, L	Women's Mountain Shorts, L	Full-Finger Gloves, L
12	276	2014	Women's Mountain Shorts, S	Road-350-W Yellow, 48	Racing Socks, L	Women's Mountain Shorts, L	Women's Mountain Shorts, L
13	277	2011	Mountain Bike Socks, M	Mountain Bike Socks, M	Mountain-100 Silver, 42	AWC Logo Cap	AWC Logo Cap
14	277	2012	Full-Finger Gloves, L	Full-Finger Gloves, L	Full-Finger Gloves, L	Full-Finger Gloves, L	Road-650 Black, 58
15	277	2013	Full-Finger Gloves, L	Women's Mountain Shorts, S	Women's Mountain Shorts, L	Women's Mountain Shorts, S	Bike Wash - Dissolver
16	277	2014	Racing Socks, L	Women's Mountain Shorts, L	Racing Socks, L	Racing Socks, L	Classic Vest, S
17	278	2011	Road-450 Red, 58	Road-650 Black, 52	Mountain-100 Black, 44	Mountain Bike Socks, M	Long-Sleeve Logo Jersey, L
18	278	2012	Full-Finger Gloves, L	Full-Finger Gloves, L	Full-Finger Gloves, M	Road-650 Red, 44	Full-Finger Gloves, M

2. Utworzyć zestawienie, które dla poszczególnych miesięcy i lat przedstawi informacje o liczbie różnych klientów. Przygotuj zapytanie z i bez użycia polecenia pivot.

--2a

```
SELECT YEAR(OrderDate) AS Rok, MONTH(OrderDate) AS Miesiąc, COUNT(DISTINCT CustomerID)
AS LiczbaKlientów
FROM Sales.SalesOrderHeader
GROUP BY YEAR(OrderDate), MONTH(OrderDate)
ORDER BY YEAR(OrderDate), MONTH(OrderDate);
```

	Rok	Miesiąc	LiczbaKlientów
1	2011	5	43
2	2011	6	141
3	2011	7	231
4	2011	8	250
5	2011	9	157
6	2011	10	327
7	2011	11	230
8	2011	12	228
9	2012	1	336
10	2012	2	219
11	2012	3	304
12	2012	4	269
13	2012	5	293
14	2012	6	390
15	2012	7	385
16	2012	8	285
17	2012	9	352
18	2012	10	321

--2b

```
SELECT
*
FROM (
    SELECT YEAR(OrderDate) AS "Rok",
           MONTH(OrderDate) AS "Miesiąc",
           COUNT(DISTINCT CustomerID) AS LiczbaKlientów
    FROM Sales.SalesOrderHeader
    GROUP BY CustomerID, YEAR(OrderDate), MONTH(OrderDate)
) AS SourceTable
PIVOT (
    COUNT(LiczbaKlientów)
    FOR Miesiąc IN ([1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12])
) AS PivotTable
ORDER BY Rok
```

	Rok	1	2	3	4	5	6	7	8	9	10	11	12
1	2011	0	0	0	0	43	141	231	250	157	327	230	228
2	2012	336	219	304	269	293	390	385	285	352	321	383	378
3	2013	400	325	441	428	426	713	1675	1727	1741	1893	2041	1970
4	2014	2073	1713	2342	2058	2350	898	0	0	0	0	0	0

3. Utworzyć zestawienie zawierające w wierszach imiona i nazwiska sprzedawców, a w kolumnach kolejne lata. Wartością będzie liczba obsłużonych transakcji.

```
SELECT *
FROM (
    SELECT CONCAT(per.FirstName, ' ', per.LastName) AS "Imie i nazwisko",
           YEAR(OrderDate) AS Rok, SalesOrderID "liczba"
    FROM Sales.SalesOrderHeader soh
    JOIN Person.Person per ON soh.SalesPersonID = per.BusinessEntityID
) AS SourceTable
PIVOT(
    COUNT(Liczba)
    FOR Rok IN ([2011],[2012],[2013],[2014])
) AS PivotTable;
```

	Imie i nazwisko	2011	2012	2013	2014
1	Amy Alberts	0	7	29	3
2	David Campbell	28	63	72	26
3	Garrett Vargas	30	80	89	35
4	Jae Pak	0	111	170	67
5	Jillian Carson	59	166	185	63
6	José Saraiva	56	86	86	43
7	Linda Mitchell	46	151	162	59
8	Lynn Tsoflias	0	0	66	43
9	Michael Blythe	65	148	175	62
10	Pamela Ansman-Wolfe	22	45	19	9
11	Rachel Valdez	0	0	86	44
12	Ranjit Varkey Chudukatil	0	42	94	39
13	Shu Ito	33	74	98	37
14	Stephen Jiang	4	22	14	8
15	Syed Abbas	0	0	12	4
16	Tete Mensa-Annan	0	24	82	34
17	Tsvi Reiter	63	153	159	54

Wyświetlić tylko tych sprzedawców, którzy pracowali przez wszystkie 4 lata.

```
SELECT *
FROM (
    SELECT CONCAT(per.FirstName, ' ', per.LastName) AS "Imie i nazwisko",
           YEAR(OrderDate) AS Rok, SalesOrderID "liczba"
    FROM Sales.SalesOrderHeader soh
    JOIN Person.Person per ON soh.SalesPersonID = per.BusinessEntityID
) AS SourceTable
PIVOT(
    COUNT(Liczba)
    FOR Rok IN ([2011],[2012],[2013],[2014])
) AS PivotTable
WHERE [2011] != 0 AND [2012] != 0 AND [2013] != 0 AND [2014] != 0;
```

	Imie i nazwisko	2011	2012	2013	2014
1	David Campbell	28	63	72	26
2	Garrett Vargas	30	80	89	35
3	Jillian Carson	59	166	185	63
4	José Saraiva	56	86	86	43
5	Linda Mitchell	46	151	162	59
6	Michael Blythe	65	148	175	62
7	Pamela Ansman-Wolfe	22	45	19	9
8	Shu Ito	33	74	98	37
9	Stephen Jiang	4	22	14	8
10	Tsvi Reiter	63	153	159	54

Zakładając, że jeżeli jeżeli pracuje to musiał cokolwiek sprzedać...

Lub:

```

SELECT *
FROM (
    SELECT CONCAT(per.FirstName, ' ', per.LastName) AS "Imie i nazwisko",
           YEAR(OrderDate) AS Rok, SalesOrderID "liczba"
    FROM Sales.SalesOrderHeader soh
    JOIN Person.Person per ON soh.SalesPersonID = per.BusinessEntityID
    WHERE per.BusinessEntityID IN (
        SELECT P.BusinessEntityID
        FROM Person.Person P
        JOIN HumanResources.Employee E ON P.BusinessEntityID =
            E.BusinessEntityID
        WHERE YEAR(E.HireDate) = 2011)
    ) AS SourceTable
PIVOT(
    COUNT(Liczba)
    FOR Rok IN ([2011],[2012],[2013],[2014])
    ) AS PivotTable;

```

	Imie i nazwisko	2011	2012	2013	2014
1	David Campbell	28	63	72	26
2	Garrett Vargas	30	80	89	35
3	Jillian Carson	59	166	185	63
4	José Saraiva	56	86	86	43
5	Linda Mitchell	46	151	162	59
6	Michael Blythe	65	148	175	62
7	Pamela Ansm...	22	45	19	9
8	Shu Ito	33	74	98	37
9	Stephen Jiang	4	22	14	8
10	Tsvi Reiter	63	153	159	54

4. Zdefiniować zapytanie wyznaczające sumę kwot sprzedaży towarów oraz liczbę różnych produktów w zamówieniach w poszczególnych latach, miesiącach, dniach.

```
SELECT YEAR(OrderDate) AS Rok,
       MONTH(OrderDate) AS Miesiac,
       DAY(OrderDate) AS Dzień,
       SUM([TotalDue]) AS Suma,
       COUNT(DISTINCT ProductID) AS 'Liczba różnych produktów'
FROM Sales.SalesOrderDetail
INNER JOIN Sales.SalesOrderHeader
ON Sales.SalesOrderDetail.SalesOrderID = Sales.SalesOrderHeader.SalesOrderID
GROUP BY YEAR(OrderDate), MONTH(OrderDate), DAY(OrderDate)
ORDER BY Rok, Miesiac, Dzień;
```

	Rok	Miesiac	Dzień	Suma	Liczba różnych produktów
1	2011	5	31	8094970.2066	47
2	2011	6	1	15394.3298	4
3	2011	6	2	16588.4572	4
4	2011	6	3	7907.9768	2
5	2011	6	4	16588.4572	4
6	2011	6	5	15815.9536	3
7	2011	6	6	8680.4804	3
8	2011	6	7	8680.4804	2
9	2011	6	8	23105.3072	5
10	2011	6	9	11664.9658	3
11	2011	6	10	15815.9536	3
12	2011	6	11	15618.9542	4
13	2011	6	12	7907.9768	2
14	2011	6	13	27677.9188	4
15	2011	6	14	12409.8444	4

5. Wykorzystując polecenie CASE przygotować podsumowania do zestawienia z poprzedniego zadania tak, aby sumowane były kwoty zamówień oraz obliczana liczba różnych produktów dla poszczególnych miesięcy i dni tygodnia.

Uwaga: Pamiętaj o wybraniu właściwego atrybutu funkcji datepart tak, aby zgadzała się nazwa dnia tygodnia.

```
case
    when Dzień = 1 then 'Poniedziałek'
    when Dzień = 2 then 'Wtorek'
    when Dzień = 3 then 'Środa'
    when Dzień = 4 then 'Czwartek'
    when Dzień = 5 then 'Piątek'
    when Dzień = 6 then 'Sobota'
    when Dzień = 7 then 'Niedziela'
end
```

Przykład wykorzystania polecenia CASE.


```

SELECT
CASE
WHEN MONTH(H.OrderDate) = 1 THEN 'Styczen'
WHEN MONTH(H.OrderDate) = 2 THEN 'Luty'
WHEN MONTH(H.OrderDate) = 3 THEN 'Marzec'
WHEN MONTH(H.OrderDate) = 4 THEN 'Kwiecien'
WHEN MONTH(H.OrderDate) = 5 THEN 'Maj'
WHEN MONTH(H.OrderDate) = 6 THEN 'Czerwiec'
WHEN MONTH(H.OrderDate) = 7 THEN 'Lipiec'
WHEN MONTH(H.OrderDate) = 8 THEN 'Sierpień'
WHEN MONTH(H.OrderDate) = 9 THEN 'Wrzesień'
WHEN MONTH(H.OrderDate) = 10 THEN 'Pazdziernik'
WHEN MONTH(H.OrderDate) = 11 THEN 'Listopad'
WHEN MONTH(H.OrderDate) = 12 THEN 'Grudzien'
END "Miesiac",
SUM(H.SubTotal) "Suma kwot",
COUNT(DISTINCT D.ProductID) "Liczba różnych produktów"
FROM Sales.SalesOrderHeader H JOIN Sales.SalesOrderDetail D ON H.SalesOrderID
=D.SalesOrderID
GROUP BY MONTH(H.OrderDate)
ORDER BY MONTH(H.OrderDate) ASC;

```

	Miesiac	Suma kwot	Liczba różnych produktów
1	Styczen	162948889.3944	211
2	Luty	91887841.9651	200
3	Marzec	339699263.2856	246
4	Kwiecien	100265759.5521	200
5	Maj	346435416.8907	263
6	Czerwiec	323879162.85	253
7	Lipiec	318807487.1861	260
8	Sierpień	183995405.5416	245
9	Wrzesień	228138296.6322	234
10	Pazdziernik	276923981.722	247
11	Listopad	84420434.5777	201
12	Grudzien	139068517.0799	211

```

SELECT
    YEAR(OrderDate) AS 'Rok',
    MIN(
        CASE
            WHEN DATEPART(M, OrderDate) = 1 THEN 'Styczeń'
            WHEN DATEPART(M, OrderDate) = 2 THEN 'Luty'
            WHEN DATEPART(M, OrderDate) = 3 THEN 'Marzec'
            WHEN DATEPART(M, OrderDate) = 4 THEN 'Kwiecień'
            WHEN DATEPART(M, OrderDate) = 5 THEN 'Maj'
            WHEN DATEPART(M, OrderDate) = 6 THEN 'Czerwiec'
            WHEN DATEPART(M, OrderDate) = 7 THEN 'Lipiec'
            WHEN DATEPART(M, OrderDate) = 8 THEN 'Sierpień'
            WHEN DATEPART(M, OrderDate) = 9 THEN 'Wrzesień'
            WHEN DATEPART(M, OrderDate) = 10 THEN 'Październik'
            WHEN DATEPART(M, OrderDate) = 11 THEN 'Listopad'
            WHEN DATEPART(M, OrderDate) = 12 THEN 'Grudzień'
        END) AS 'Miesiąc',
    MIN(CASE
        WHEN DATEPART(dw, OrderDate) = 1 THEN 'Niedziela'
        WHEN DATEPART(dw, OrderDate) = 2 THEN 'Poniedziałek'
        WHEN DATEPART(dw, OrderDate) = 3 THEN 'Wtorek'
        WHEN DATEPART(dw, OrderDate) = 4 THEN 'Środa'
        WHEN DATEPART(dw, OrderDate) = 5 THEN 'Czwartek'
        WHEN DATEPART(dw, OrderDate) = 6 THEN 'Piątek'
        WHEN DATEPART(dw, OrderDate) = 7 THEN 'Sobota'
    END) AS 'Dzień tygodnia',
    SUM(TotalDue) AS 'Suma',
    COUNT(DISTINCT ProductID) AS 'Liczba różnych produktów'
FROM Sales.SalesOrderDetail
INNER JOIN Sales.SalesOrderHeader
    ON Sales.SalesOrderDetail.SalesOrderID = Sales.SalesOrderHeader.SalesOrderID
GROUP BY
    YEAR(OrderDate),
    MONTH(OrderDate),
    DAY(OrderDate)
ORDER BY Rok, MONTH(OrderDate), DAY(OrderDate);

```

	Rok	Miesiąc	Dzień tygodnia	Suma	Liczba różnych produktów
1	2011	Maj	Wtorek	8094970.2066	47
2	2011	Czerwiec	Sroda	15394.3298	4
3	2011	Czerwiec	Czwartek	16588.4572	4
4	2011	Czerwiec	Piatek	7907.9768	2
5	2011	Czerwiec	Sobota	16588.4572	4
6	2011	Czerwiec	Niedziela	15815.9536	3
7	2011	Czerwiec	Poniedzialek	8680.4804	3
8	2011	Czerwiec	Wtorek	8680.4804	2
9	2011	Czerwiec	Sroda	23105.3072	5
10	2011	Czerwiec	Czwartek	11664.9658	3
11	2011	Czerwiec	Piatek	15815.9536	3
12	2011	Czerwiec	Sobota	15618.9542	4
13	2011	Czerwiec	Niedziela	7907.9768	2
14	2011	Czerwiec	Poniedzialek	27677.9188	4
15	2011	Czerwiec	Wtorek	12409.8444	4
16	2011	Czerwiec	Sroda	15815.9536	3

6. Przygotować zestawienie, w którym dla wybranych klientów przygotujemy kartę lojalnościową:
 - a. srebrną, jeśli klient wykonał co najmniej 5 transakcji w sklepie;
 - b. złotą, jeśli klient wykonał co najmniej 2 transakcje, której kwota przekraczała 150% średniej wartości zamówień w bazie (łącznie kwota przekraczała 300% i każda z transakcji przekraczała 150%);
 - c. platynową, jeśli klient co roku spełniał warunki otrzymania karty złotej.

Schemat wynikowej tabeli powinien wyglądać następująco:

KartyLojalnoscowe(Imie, Nazwisko, Liczba transakcji, Łączna kwota transakcji, Kolor karty)

```

SELECT
  P.FirstName "Imie",
  P.LastName "Nazwisko",
  [liczba transakcji] "Liczba transakcji",
  [łączna kwota transakcji] "Łączna kwota transakcji",
  CASE
    WHEN [liczba powyżej w roku] >= 2 AND
         [liczba lat] = 4 THEN 'Platynowa'
    WHEN [liczba powyżej średniej] >= 2 THEN 'Złota'
    WHEN [liczba transakcji] >= 5 THEN 'Srebrna'
    ELSE 'Brak'
  END AS 'Kolor karty'
FROM (SELECT
  CustomerID,
  SUM([powyżej średniej w roku]) AS 'liczba powyżej średniej',
  COUNT(CustomerID) 'liczba lat',
  MIN([powyżej średniej w roku]) AS 'liczba powyżej w roku'
FROM (SELECT
  CustomerID,
  Rok,
  SUM([powyżej średniej]) AS 'powyżej średniej w roku'
FROM (SELECT
  CustomerID,
  YEAR(OrderDate) AS Rok,
  IF(AVG(TotalDue) OVER () * 1.5 < TotalDue, 1, 0) 'powyżej średniej'
FROM Sales.SalesOrderHeader) powyżej
GROUP BY CustomerID,
         Rok) rok
GROUP BY CustomerID) calosc
INNER JOIN (SELECT
  CustomerID,
  COUNT(SalesOrderID) AS 'liczba transakcji',
  SUM(TotalDue) AS 'łączna kwota transakcji'
FROM Sales.SalesOrderHeader
GROUP BY CustomerID) transakcje
ON transakcje.CustomerID = calosc.CustomerID
JOIN Sales.Customer C
ON C.CustomerID = transakcje.CustomerID
JOIN Person.Person P
ON C.PersonID = P.BusinessEntityID
WHERE ([liczba powyżej w roku] >= 2
AND [liczba lat] = 4)
OR [liczba powyżej średniej] >= 2
OR [Liczba transakcji] >= 5;

```

	Imie	Nazwisko	Liczba transakcji	Łączna kwota transakcji	Kolory kart
4	Gustavo	Achong	7	147804.9208	Złota
5	Pilar	Ackerman	4	249804.8673	Złota
6	Carla	Adams	4	98273.5468	Złota
7	Frances	Adams	12	428350.5326	Złota
8	Jay	Adams	6	158025.1722	Złota
9	Kaitlyn	Adams	5	297.3336	Srebrna
10	Miguel	Adams	5	259.6862	Srebrna
11	Samuel	Agcaoili	9	18275.1045	Srebrna
12	Robert	Ahlering	4	107741.109	Złota
13	Kim	Akers	8	5835.052	Srebrna
14	Stanley	Alan	4	244854.0475	Złota
15	Amy	Alberts	8	264645.3146	Złota
16	Anna	Albright	7	147107.0296	Złota
17	Gregory	Alderson	7	11542.2556	Srebrna
18	Elijah	Alexander	5	366.5065	Srebrna
19	L. Phillip	Alexander	4	101006.1000	Złota

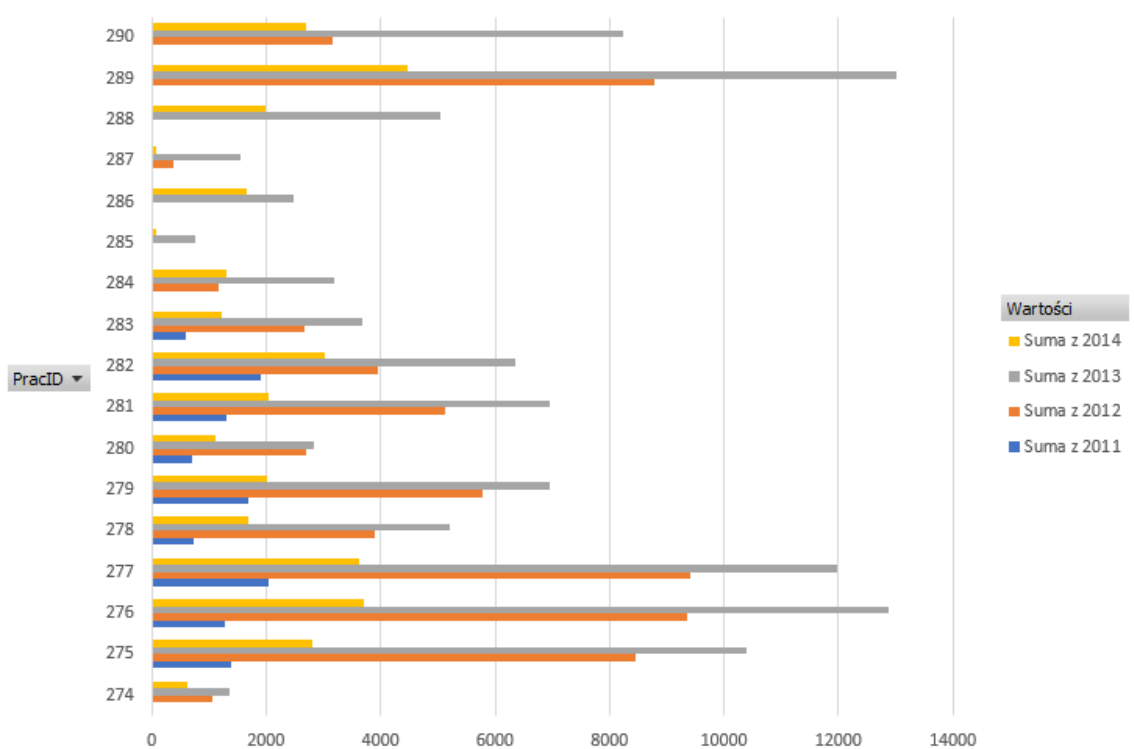
Zad. 2. Analiza danych

- Przedstaw wyniki zadania 1 w postaci tabel i wykresów przestawnych w programie MS Excel. Zinterpretuj wyniki. (pliki workbook) (ddl też podać)

1.1

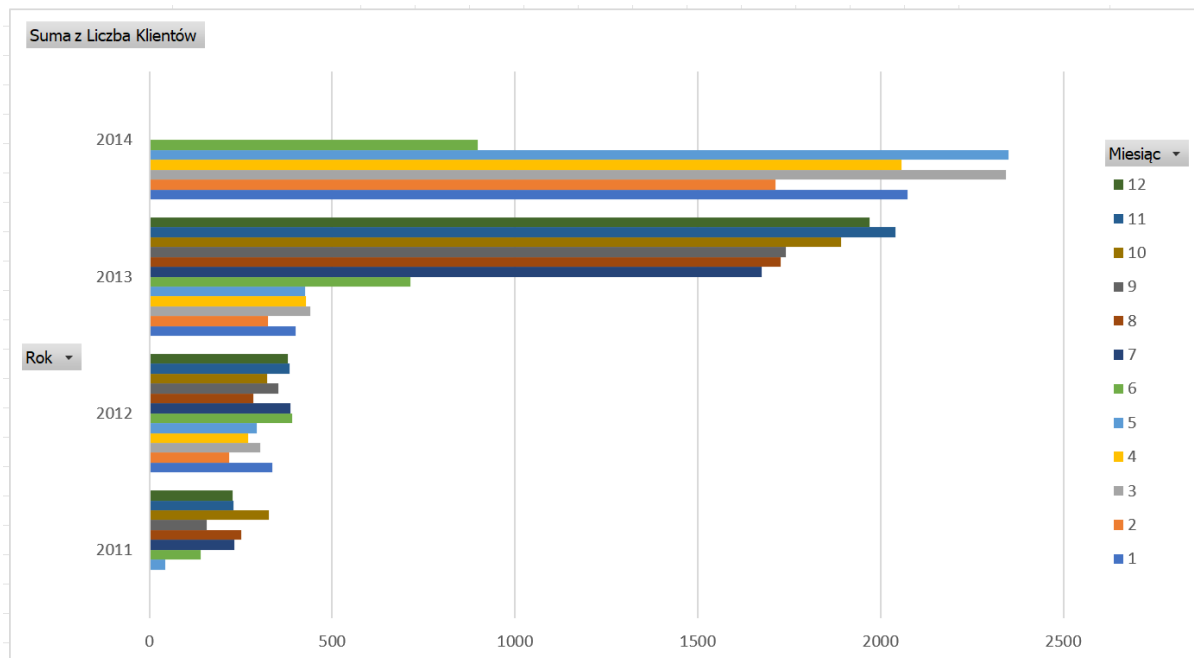
PracID	Suma z 2011	Suma z 2012	Suma z 2013	Suma z 2014
274	30	1061	1366	638
275	1394	8456	10391	2817
276	1289	9349	12880	3711
277	2053	9396	11975	3627
278	746	3897	5215	1686
279	1681	5780	6958	2012
280	709	2705	2823	1123
281	1293	5114	6940	2050
282	1896	3944	6349	3031
283	596	2662	3692	1222
284	0	1171	3185	1294
285	0	0	752	73
286	0	0	2470	1653
287	0	373	1557	82
288	0	0	5035	1998
289	0	8768	13002	4461
290	0	3160	8239	2686
Suma końcowa	11687	65836	102829	34164

Suma z 2011 Suma z 2012 Suma z 2013 Suma z 2014



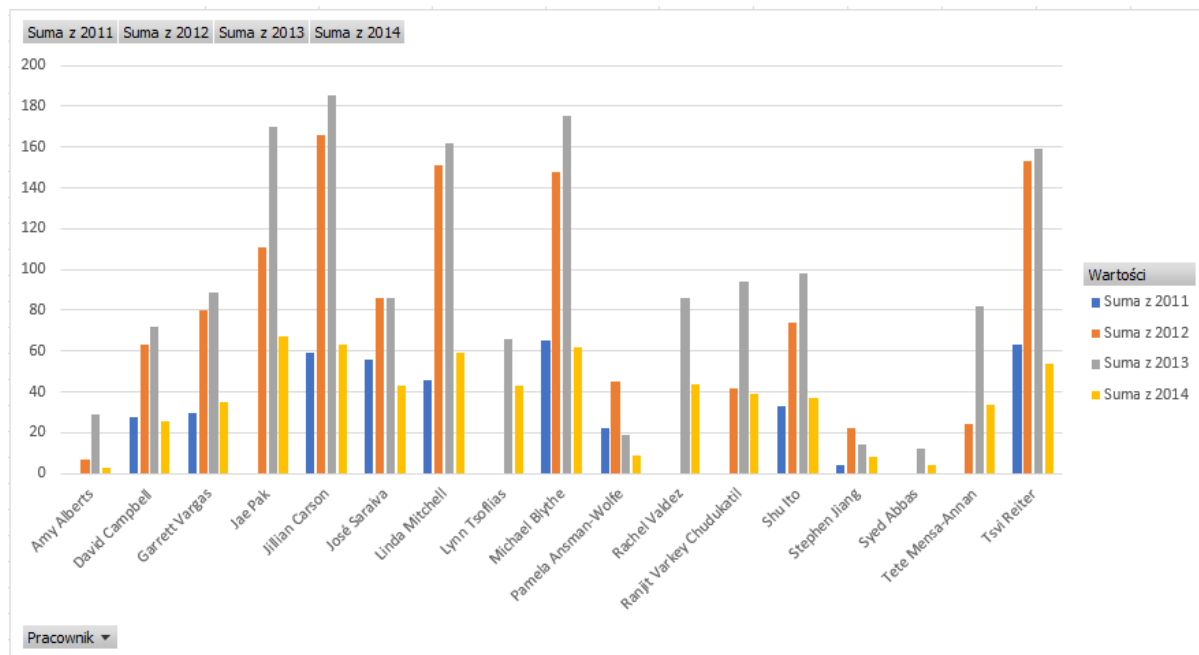
1.2

Suma z Liczba Klientów		Etykiety kolumn												Suma końcowa
Etykiety wierszy		1	2	3	4	5	6	7	8	9	10	11	12	
2011						43	141	231	250	157	327	230	228	1607
2012		336	219	304	269	293	390	385	285	352	321	383	378	3915
2013		400	325	441	428	426	713	1675	1727	1741	1893	2041	1970	13780
2014		2073	1713	2342	2058	2350	898							11434
Suma końcowa		2809	2257	3087	2755	3112	2142	2291	2262	2250	2541	2654	2576	30736



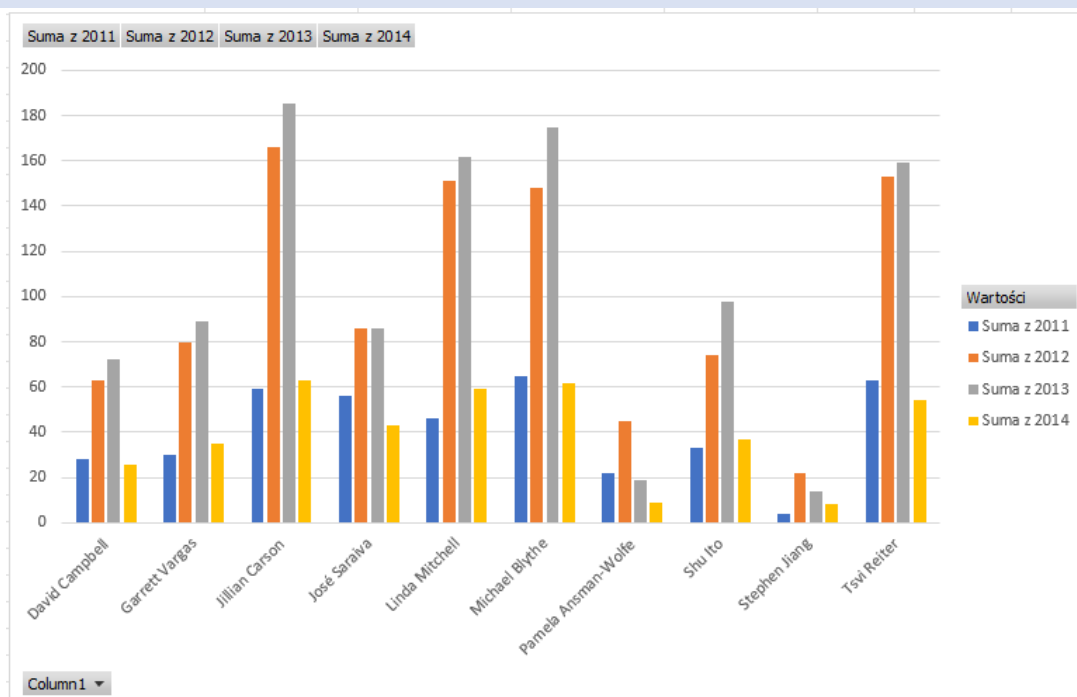
1.3a

Pracownik	Suma z 2011	Suma z 2012	Suma z 2013	Suma z 2014
Amy Alberts	0	7	29	3
David Campbell	28	63	72	26
Garrett Vargas	30	80	89	35
Jae Pak	0	111	170	67
Jillian Carson	59	166	185	63
José Saraiva	56	86	86	43
Linda Mitchell	46	151	162	59
Lynn Tsoflis	0	0	66	43
Michael Blythe	65	148	175	62
Pamela Ansman-Wolfe	22	45	19	9
Rachel Valdez	0	0	86	44
Ranjit Varkey Chudukatil	0	42	94	39
Shu Ito	33	74	98	37
Stephen Jiang	4	22	14	8
Syed Abbas	0	0	12	4
Tete Mensa-Annan	0	24	82	34
Tsvi Reiter	63	153	159	54
Suma końcowa	406	1172	1598	630

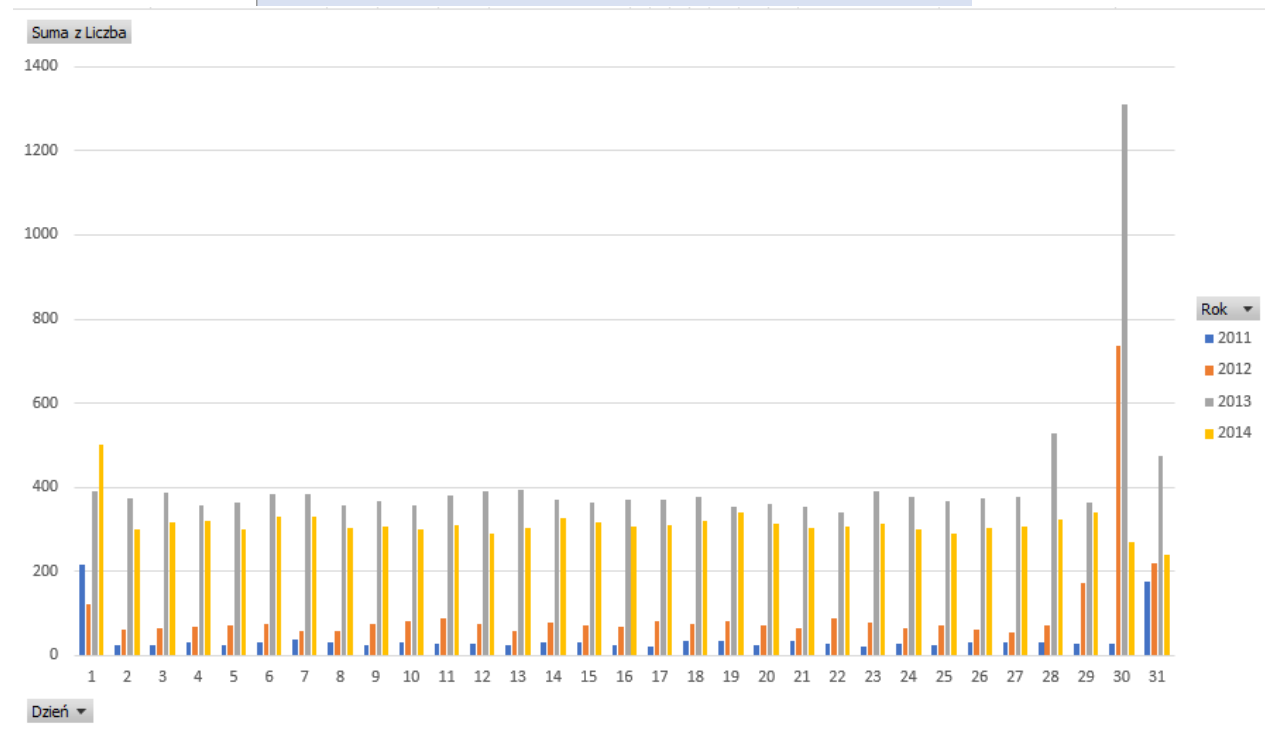


1.3b

Pracownik	Suma z 2011	Suma z 2012	Suma z 2013	Suma z 2014
David Campbell	28	63	72	26
Garrett Vargas	30	80	89	35
Jillian Carson	59	166	185	63
José Saraiva	56	86	86	43
Linda Mitchell	46	151	162	59
Michael Blythe	65	148	175	62
Pamela Anzman-Wolfe	22	45	19	9
Shu Ito	33	74	98	37
Stephen Jiang	4	22	14	8
Tsvi Reiter	63	153	159	54
Suma końcowa	406	988	1059	396

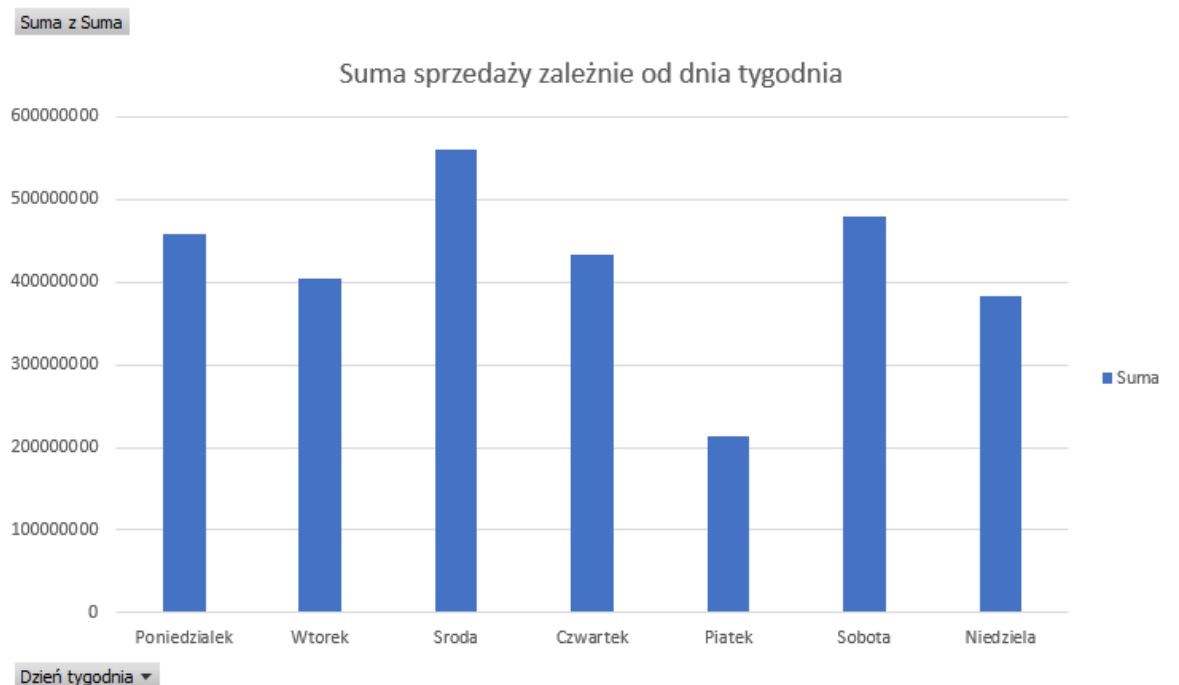


Suma z Liczba	Etykiety kolumn				
Etykiety wierszy	2011	2012	2013	2014	Suma końcowa
1	217	122	391	503	1233
2	25	61	375	300	761
3	26	65	389	318	798
4	33	70	357	319	779
5	24	71	363	300	758
6	30	76	383	330	819
7	39	60	385	329	813
8	33	60	356	305	754
9	25	76	366	306	773
10	30	81	358	300	769
11	28	89	379	311	807
12	27	76	391	289	783
13	26	59	394	305	784
14	33	77	372	326	808
15	31	72	364	318	785
16	25	67	371	307	770
17	20	83	371	311	785
18	34	74	376	321	805
19	34	83	354	340	811
20	26	71	361	314	772
21	36	66	354	305	761
22	27	89	341	308	765
23	20	78	391	312	801
24	27	66	378	300	771
25	26	73	367	291	757
26	30	63	374	304	771
27	32	55	376	308	771
28	33	72	528	325	958
29	27	173	365	339	904
30	28	735	1310	269	2342
31	176	220	473	241	1110
Suma końcowa	1228	3183	12713	9754	26878

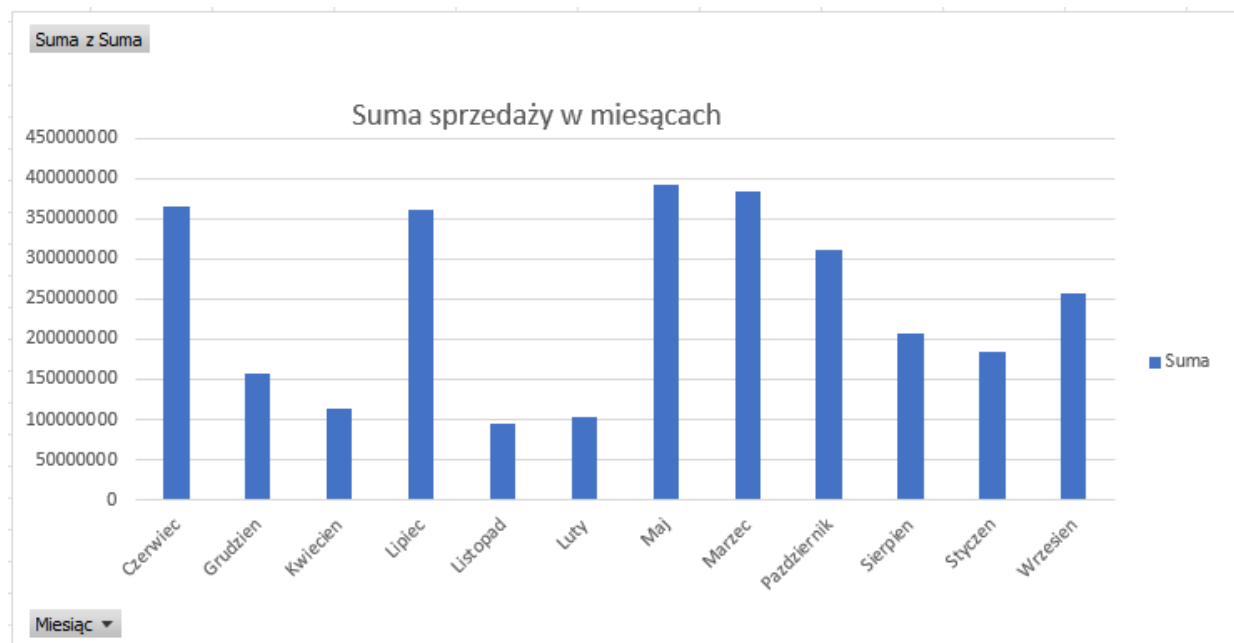


1.5

Etykiety wierszy ▾	Suma z Suma
Poniedziałek	457072471.5
Wtorek	403492724.4
Sroda	560644420.9
Czwartek	432484099.6
Piatek	212585108.7
Sobota	478917953.3
Niedziela	381773345.6
Suma końcowa	2926970124



Etykiety wierszy ▾	Suma z Suma
Czerwiec	365654294.5
Grudzien	156509185.3
Kwiecien	113514051
Lipiec	360088403.3
Listopad	94933254.06
Luty	103403614.4
Maj	390954184.8
Marzec	382618986.2
Pazdziernik	311820065.3
Sierpień	207215346
Styczeń	183378292.8
Wrzesień	256880446.3
Suma końcowa	2926970124



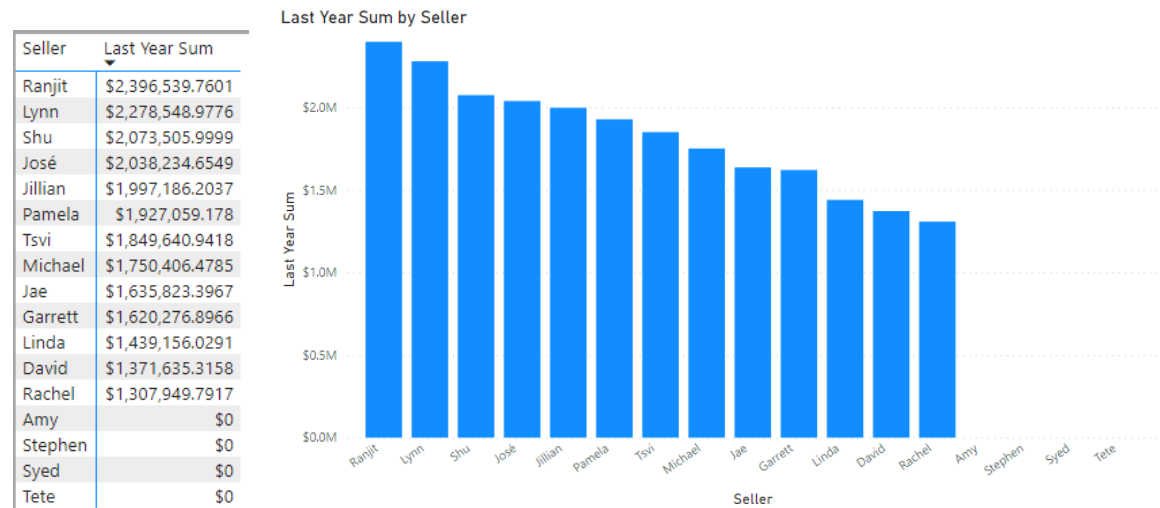
1.6

Etykiety wierszy ▾	Liczba z łączna kwota
Platynowa	13
Srebrna	190
Złota	366
Suma końcowa	569

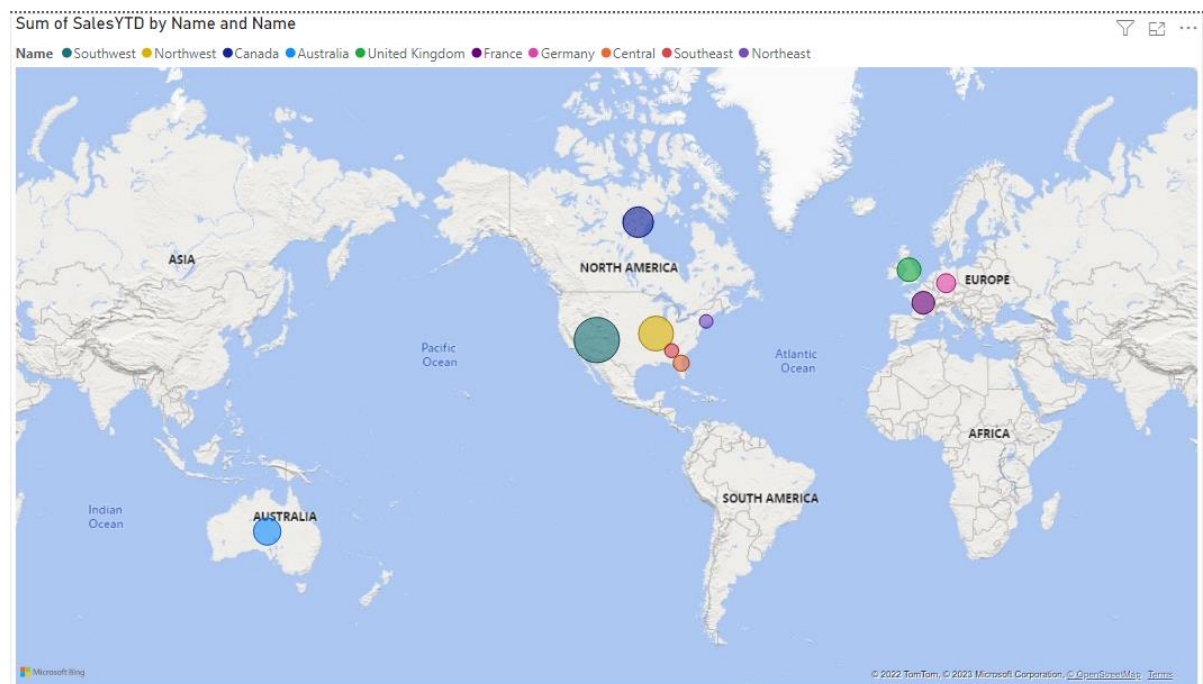


- Przygotuj 5 dodatkowych tabel/wykresów, które pokażą ciekawe zależności w bazie AdventureWorks przy użyciu narzędzia Power BI (screenshotty) lub Tableau. (taka złożoność jak zadania, ranking, grupowanie)

- Wykres przedstawiający sumę wszystkich sprzedaży pracowników za poprzedni rok z tabeli vSalesPerson

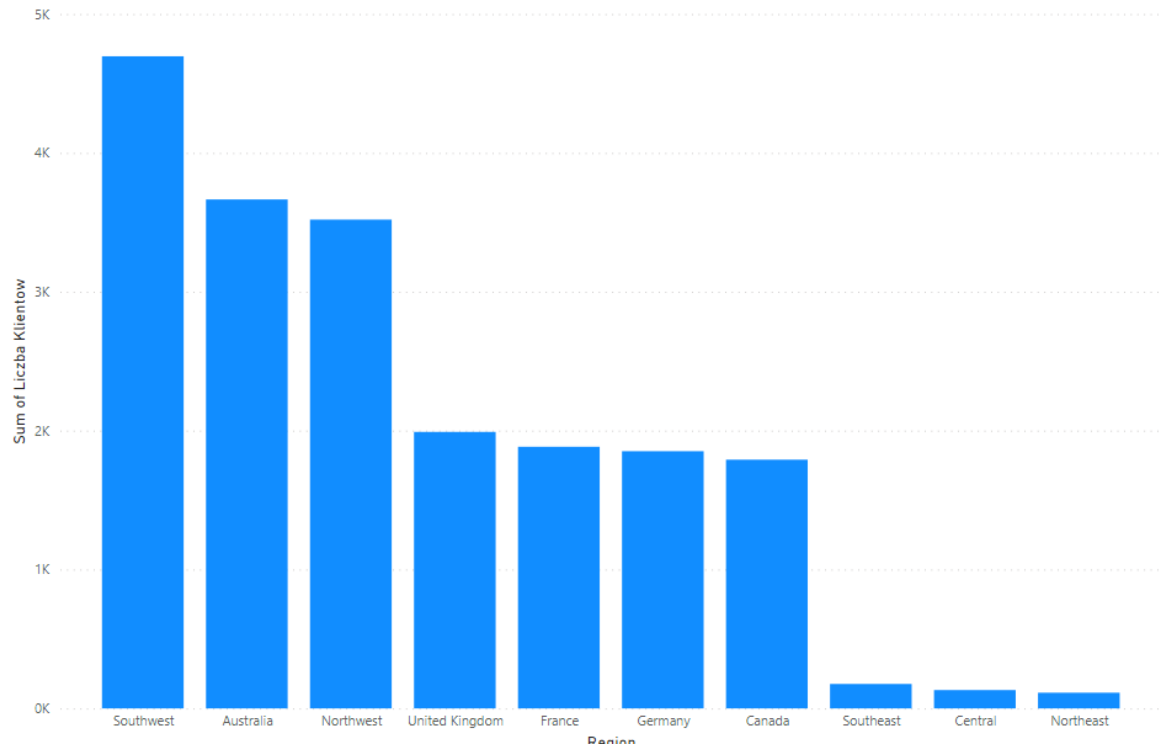


- Mapa pokazująca wszystkie przychody firmy od początku roku do bieżącego dnia włącznie:



3. Liczba klientów według regionu:

Sum of Liczba Klientow by Region



4. Najczęściej kupowany produkt w każdej kategorii w poszczególnych latach

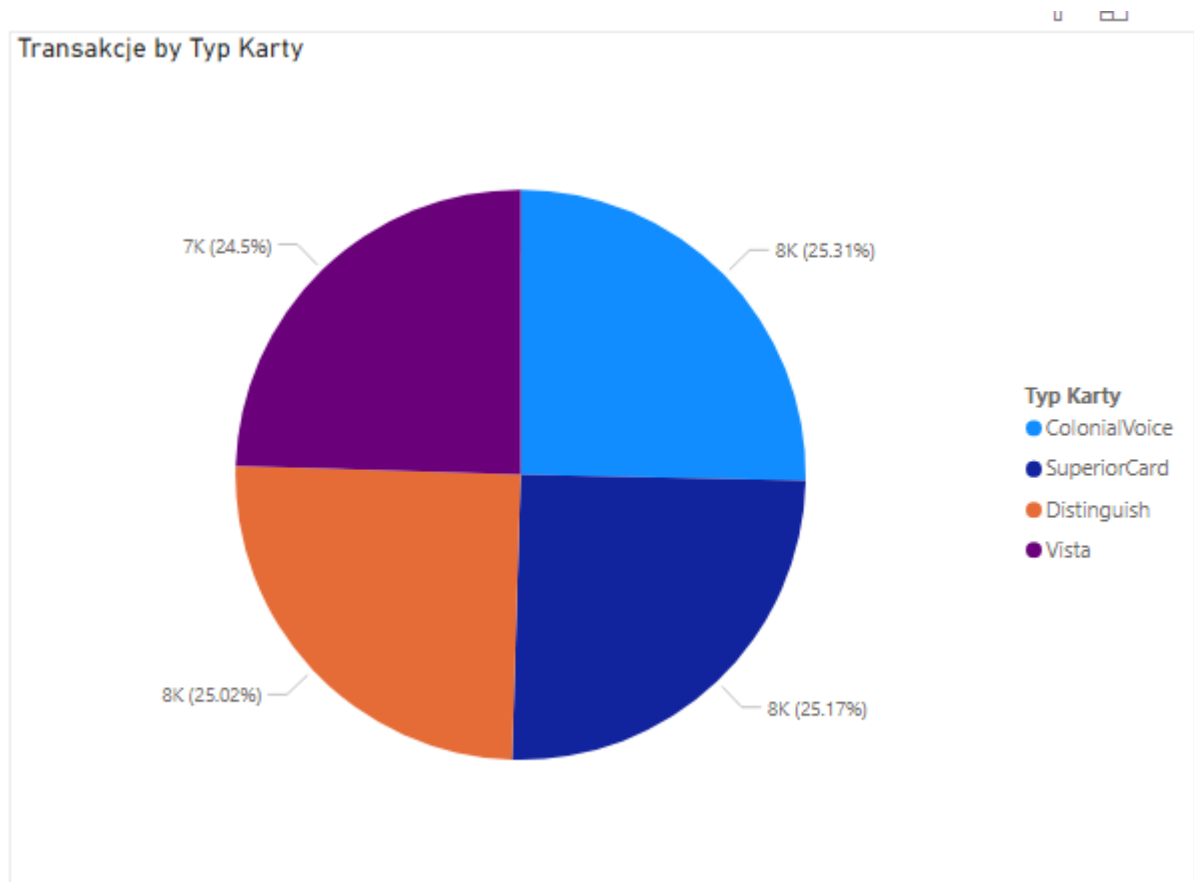
```

SELECT * FROM
(SELECT YEAR(soh.OrderDate) AS OrderYear,
      pc.Name AS CategoryName,
      MAX(p.Name) AS BestSellingItem
FROM Sales.SalesOrderHeader AS soh
INNER JOIN Sales.SalesOrderDetail AS sod
      ON soh.SalesOrderID = sod.SalesOrderID
INNER JOIN Production.Product AS p
      ON sod.ProductID = p.ProductID
INNER JOIN Production.ProductSubcategory AS psc
      ON p.ProductSubcategoryID = psc.ProductSubcategoryID
INNER JOIN Production.ProductCategory AS pc
      ON psc.ProductCategoryID = pc.ProductCategoryID
GROUP BY YEAR(soh.OrderDate), pc.Name) AS X
PIVOT
(
  MAX(BestSellingItem)
  FOR OrderYear IN ([2011], [2012], [2013], [2014])
) AS PivotTable;

```

CategoryName	2014	2013	2012	2011
Accessories	Water Bottle - 30 oz.	Water Bottle - 30 oz.	Sport-100 Helmet, Red	Sport-100 Helmet, Red
Bikes	Touring-3000 Yellow, 62	Touring-3000 Yellow, 62	Road-650 Red, 62	Road-650 Red, 62
Clothing	Women's Tights, S	Women's Tights, S	Women's Tights, S	Mountain Bike Socks, M
Components	Touring Pedal	Touring Pedal	ML Road Rear Wheel	ML Road Frame - Red, 52

5. Podział transakcji ze względu na typ karty płatniczej



Wnioski:

1. Pierwszym wnioskiem jaki nasuwa się po wykonaniu listy jest możliwość wprowadzenia danych bezpośrednio przy użyciu SELECT, mianowicie używając INTO. Nie musimy wstawiać osobno każdego wiersza, ustanawiać połączenie z bazą danych i je zamykać tylko na raz podczas jednego połączenia możemy wstawić wynik naszego zapytania.
2. Za pomocą operatora PIVOT możemy przedstawić dane w formie tabeli przestawnej prezentacja danych w ten sposób ułatwi ich odczytanie oraz analizę.
3. Jeżeli znamy ilość unikalnych wartości kolumny do rotacji to nie ma jak najmniejszego problemu by użyć PIVOT'a według tychże wartości. Aczkolwiek jeżeli nie jesteśmy pewni co do wartości kolumny do rotacji, zostaje nam użyć dynamicznego SQL'a, który został przedstawiony jako rozwiązanie zadania 1.1a. w ten sposób nie martwimy się o zawarcie wszystkich unikalnych wartości kolumny do rotacji, ponieważ mamy to w naszym łańcuchu dynamicznego SQL'a, jednakże wymaga to od nas większego nakładu pracy.
4. Przy pomocy operatora CASE, możemy warunkowo wyodrębnić informacje ukryte w danych lub niejawnie przedstawione w taki sposób aby były czytelne na pierwszy rzut oka. prostym przykładem będzie określenie płci użytkownika po przez znajomość faktu że kobiecie imiona w języku polskim kończą się na literę „a”. jak i też dobrym przykładem będzie wyznaczenie dnia tygodnia z daty. Użycie tego operatora pozwala na dodatkową i szerszą analizę danych.
5. Powyższe wnioski dotyczą wyników tabel zapytań SQL, natomiast najlepszą metodą prezentacji jest metoda wizualna. można tego użyć narzędzi takich jak Excel, który oferuje bogatą ilość różnych prezentacji danych ze względu na ich złożoność, jak i np. oprogramowania Power BI, która pomaga organizacjom gromadzić zarządzać przetwarzać i analizować dane z różnych źródeł. Sam sposób prezentacji wydaje się bardzo atrakcyjny w obu przypadkach, za różnicą że Power BI jest bardziej skoncentrowany na Business Intelligence (analitika biznesowa).

6. Ostatnim wnioskiem jest dość niezadowolający sposób łączenia się z bazą danych oraz ekstrakcją danych podczas połączenia, mianowicie nie ma żadnego prostego edytora zapytań, proces polega na połączeniu się wraz z gotowym zapytaniem, co w przypadku jeżeli nie znamy wyniku zapytania, jest mało przydatne. Było to widoczne w obu przypadkach programów, co jak na firmę Microsoft zdziwiło mnie, to znaczy brak możliwości edycji, zapytań po otrzymaniu przykładowo błędnego wyniku.

Uwaga!

- Sprawozdanie, bez wniosków podsumowujących aspekt zagadnień analizowanych na zajęciach laboratoryjnych i zawartych w sprawozdaniu, jest automatycznie oceniane negatywnie!