

Hurtownie danych – Spr. 1.

PWr. WIZ, Informatyka, Data: 11.03.2020

Student	Email: 242493@student.pwr.edu.pl	Ocena
Indeks	<u>242493</u>	
Imię	<u>Arkadiusz</u>	
Nazwisko	<u>Rasz</u>	

Spis treści

Zadanie 1.....	2
a)	2
b)	2
Zadanie 2.....	5
a)	5
b)	6
Zadanie 3.....	7
Zadanie 4.....	8
Zadanie 5.....	9
Zadanie 6.....	12
Zadanie 7.....	15
Zadanie 8.....	17

Zadanie 1. Utworzyć tabelę Sprzedaz zawierającą dane dotyczące liczby sprzedanych produktów przez pracowników firmy w poszczególnych latach zgodnie z podanym schematem: Sprzedaz(pracID, prodID, "Nazwa produktu", Rok, Liczba).

a) W kolejnych kolumnach wyświetlić lata:

```
with Sprzedaz(pracID, prodID, "Nazwa produktu", Rok, Liczba) as
(
    select H.SalesPersonID, D.ProductID, P.Name, YEAR(H.OrderDate), D.OrderQty
    from Sales.SalesOrderHeader H
    join Sales.SalesOrderDetail D
        on H.SalesOrderID = D.SalesOrderID
    join Production.Product P
        on D.ProductID = P.ProductID
)
select pracId, prodId,
    COALESCE(CAST([2010] AS DEC(10, 0)), 0) [2010],
    COALESCE(CAST([2011] AS DEC(10, 0)), 0) [2011],
    COALESCE(CAST([2012] AS DEC(10, 0)), 0) [2012],
    COALESCE(CAST([2013] AS DEC(10, 0)), 0) [2013],
    COALESCE(CAST([2014] AS DEC(10, 0)), 0) [2014],
    COALESCE(CAST([2015] AS DEC(10, 0)), 0) [2015],
    COALESCE(CAST([2016] AS DEC(10, 0)), 0) [2016],
    COALESCE(CAST([2017] AS DEC(10, 0)), 0) [2017]
from Sprzedaz
pivot(SUM(Liczba) for Rok in ([2010], [2011], [2012], [2013], [2014], [2015], [2016],
[2017])) as X;
```

Wynik zapytania:

pracId	prodId	2010	2011	2012	2013	2014	2015	2016	2017
279	723	0	1	6	2	0	0	0	0
280	787	0	0	27	10	0	0	0	0
279	984	0	0	0	25	4	0	0	0
289	841	0	0	51	37	0	0	0	0
290	905	0	0	0	33	14	0	0	0
279	769	0	5	40	7	0	0	0	0
288	708	0	0	0	133	59	0	0	0
289	795	0	0	63	73	23	0	0	0
280	810	0	0	14	19	4	0	0	0
280	710	0	10	3	0	0	0	0	0
281	774	0	33	26	0	0	0	0	0
278	736	0	0	12	7	1	0	0	0
290	805	0	0	2	0	0	0	0	0
287	767	0	0	0	5	0	0	0	0
277	910	0	0	0	23	11	0	0	0
278	974	0	0	0	22	9	0	0	0
277	718	0	1	29	45	16	0	0	0
290	759	0	0	7	0	0	0	0	0
278	782	0	0	19	30	9	0	0	0
288	877	0	0	0	144	39	0	0	0

....

b) w kolejnych kolumnach wyświetlić 5 najlepszych sprzedawców.

Do odnalezienia 5 najlepszych sprzedawców użyto zapytania:

```
select H.SalesPersonID, SUM(D.OrderQty)
from Sales.SalesOrderHeader H
join Sales.SalesOrderDetail D
    on H.SalesOrderID = D.SalesOrderID
join Production.Product P
    on D.ProductID = P.ProductID
group by H.SalesPersonID
order by SUM(D.OrderQty) desc;
```

ID pracowników, którzy wykonali najwięcej sprzedaży: 276, 277, 289, 275, 279. Sprzedaże bez specyfikowanego sprzedawcy pominięto. Odnalezione ID sprzedawców użyto w zapytaniu:

```
with Sprzedaz(pracID, prodID, "Nazwa produktu", Rok, Liczba) as
(
    select H.SalesPersonID, D.ProductID, P.Name, YEAR(H.OrderDate), D.OrderQty
    from Sales.SalesOrderHeader H
    join Sales.SalesOrderDetail D
        on H.SalesOrderID = D.SalesOrderID
    join Production.Product P
        on D.ProductID = P.ProductID
)
select prodID, "Nazwa produktu", Rok,
    COALESCE(CAST([276] AS DEC(10, 0)), 0) [276],
    COALESCE(CAST([277] AS DEC(10, 0)), 0) [277],
    COALESCE(CAST([289] AS DEC(10, 0)), 0) [289],
    COALESCE(CAST([275] AS DEC(10, 0)), 0) [275],
    COALESCE(CAST([279] AS DEC(10, 0)), 0) [279]
from Sprzedaz
pivot(SUM(Liczba) for pracID in ([276],[277],[289],[275],[279])) as X
order by prodID, Rok;
```

Wynik zapytania:

prodID	Nazwa produktu	Rok	276	277	289	275	279
707	Sport-100 Helmet, Red	2011	46	53	0	21	35
707	Sport-100 Helmet, Red	2012	191	162	173	143	117
707	Sport-100 Helmet, Red	2013	244	243	227	137	95
707	Sport-100 Helmet, Red	2014	69	63	98	24	38
708	Sport-100 Helmet, Black	2011	38	67	0	33	27
708	Sport-100 Helmet, Black	2012	215	161	188	148	107
708	Sport-100 Helmet, Black	2013	269	254	241	118	151
708	Sport-100 Helmet, Black	2014	70	105	89	32	27
709	Mountain Bike Socks, M	2011	62	81	0	64	114
709	Mountain Bike Socks, M	2012	94	74	0	37	99
710	Mountain Bike Socks, L	2011	8	8	0	6	11
710	Mountain Bike Socks, L	2012	0	2	0	4	2
711	Sport-100 Helmet, Blue	2011	43	73	0	19	47
711	Sport-100 Helmet, Blue	2012	227	210	199	182	131
711	Sport-100 Helmet, Blue	2013	258	297	265	177	128
711	Sport-100 Helmet, Blue	2014	84	97	72	33	50
712	AWC Logo Cap	2011	43	114	0	53	71
712	AWC Logo Cap	2012	278	300	258	219	203
712	AWC Logo Cap	2013	309	322	382	215	151
712	AWC Logo Cap	2014	110	97	104	53	33
713	Long-Sleeve Logo Jersey, S	2013	0	0	0	0	0
713	Long-Sleeve Logo Jersey, S	2014	0	0	0	0	0
714	Long-Sleeve Logo Jersey, M	2011	19	47	0	22	32
714	Long-Sleeve Logo Jersey, M	2012	173	156	125	103	92
714	Long-Sleeve Logo Jersey, M	2013	187	191	172	122	99
714	Long-Sleeve Logo Jersey, M	2014	46	57	51	11	30
715	Long-Sleeve Logo Jersey, L	2011	58	123	0	49	74
715	Long-Sleeve Logo Jersey, L	2012	303	306	278	268	177

....

Zapytanie zwróciło 610 wyników. Nazwami ostatnich kolumn są ID pracowników, ale mogłyby być również reprezentowane przez np. Imię oraz nazwisko sprzedawcy podając ich wartości zamiast odpowiednich nazw kolumn [ID].

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

a) Bez użycia pivot

Zapytanie:

```
select
    Year(H.OrderDate) as "Year",
    MONTH(H.OrderDate) as "Month",
    count(distinct H.CustomerID) as "Customer
count"
from Sales.SalesOrderHeader H
group by Year(H.OrderDate), MONTH(H.OrderDate)
order by Year(H.OrderDate), MONTH(H.OrderDate) asc;
```

Jak widać, wynik zapytania nie jest przejrzysty oraz zwraca wiele wierszy. Następne zapytanie z użyciem polecenia pivot zwraca wynik łatwiejszy do dalszego przetwarzania i analizy.

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

b) z poleceniem pivot

Zapytanie:

```
with Zestawienie(_year, _month, _cust) as
(
    select distinct Year(H.OrderDate), MONTH(H.OrderDate), H.CustomerID
    from Sales.SalesOrderHeader H
)
select _year,
    COALESCE(CAST([1] AS DEC(10, 0)), 0) [Jan],
    COALESCE(CAST([2] AS DEC(10, 0)), 0) [Feb],
    COALESCE(CAST([3] AS DEC(10, 0)), 0) [Mar],
    COALESCE(CAST([4] AS DEC(10, 0)), 0) [Apr],
    COALESCE(CAST([5] AS DEC(10, 0)), 0) [May],
    COALESCE(CAST([6] AS DEC(10, 0)), 0) [Jun],
    COALESCE(CAST([7] AS DEC(10, 0)), 0) [Jul],
    COALESCE(CAST([8] AS DEC(10, 0)), 0) [Aug],
    COALESCE(CAST([9] AS DEC(10, 0)), 0) [Sep],
    COALESCE(CAST([10] AS DEC(10, 0)), 0) [Oct],
    COALESCE(CAST([11] AS DEC(10, 0)), 0) [Nov],
    COALESCE(CAST([12] AS DEC(10, 0)), 0) [Dec]
from Zestawienie
pivot(Count(_cust) for _month in ([1], [2], [3], [4], [5], [6], [7], [8], [9], [10],
[11], [12])) as X
order by _year;
```

Wynik zapytania:

_year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2011	0	0	0	0	43	141	231	250	157	327	230	228
2012	336	219	304	269	293	390	385	285	352	321	383	378
2013	400	325	441	428	426	713	1675	1727	1741	1893	2041	1970
2014	2073	1713	2342	2058	2350	898	0	0	0	0	0	0

Zadanie 3. Utworzyć tabelę przestawną zawierającą imię i nazwisko klienta w wierszach, lata 2012 i 2014 w kolumnach. Wartością będzie liczba zamówień złożonych przez tych klientów w poszczególnych latach, jednak tylko tych, którzy nie składali zamówień w latach 2011 i 2013.

Użyte zapytanie:

```
with ClientIDs(clientID) as
(
    select distinct H.CustomerID
    from Sales.SalesOrderHeader H
    where H.CustomerID not in
    (
        select distinct CustomerID
        from Sales.SalesOrderHeader
        where Year(OrderDate) = 2011 or Year(OrderDate) = 2013
    )
)
select _name,
COALESCE(CAST([2012] AS DEC(10, 0)), 0) [2012],
COALESCE(CAST([2014] AS DEC(10, 0)), 0) [2014]
from
(
    select
        C.clientID as _clientId,
        P.FirstName + ' ' + P.LastName as _name,
        H.SalesOrderID as _orderId,
        Year(H.OrderDate) as _year
    from ClientIDs C
    join Person.Person P
        on P.BusinessEntityID = C.clientID
    join Sales.SalesOrderHeader H
        on H.CustomerID = C.clientID
) X
pivot(Count(_orderId) for _year in ([2012], [2014])) as Y;
```

Wyrażenie ClientIDs zawiera korzysta z podzapytania zwracającego id klientów, którzy zrobili zakupy w latach 2011 lub 2013. Użyte są one do przefiltrowania tabeli SalesOrderHeader, i znalezienia wszystkich pozostałych klientów. Ich Id są łączone znowu z SalesOrderHeader oraz Person oraz użyte w operacji pivot dla lat zamówienia 2012 oraz 2014. Wynik zapytania:

_name	2012	2014
Aaron Adams	0	1
Adam Adams	0	1
Alex Adams	0	1
Connor Adams	0	1
Elijah Adams	0	1
Evan Adams	0	2
Gabriel Adams	0	1
Isaiah Adams	0	1
James Adams	1	2
Jason Adams	0	1
Jonathan Adams	1	2
Jose Adams	0	1
Kyle Adams	0	1
Logan Adams	0	2
Luis Adams	0	1

... (3326 wyników)

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

Zapytanie:

```
select Year(H.OrderDate) as "Year",
       Month(H.OrderDate) as "Month",
       Day(H.OrderDate) as "Day",
       Sum(H.TotalDue) as "Price sum",
       Count(Distinct D.ProductID) as "Distinct products"
from Sales.SalesOrderHeader H
join Sales.SalesOrderDetail D
    on H.SalesOrderID = D.SalesOrderID
group by rollup(Year(H.OrderDate), Month(H.OrderDate), Day(H.OrderDate))
order by Year(H.OrderDate), Month(H.OrderDate), Day(H.OrderDate);
```

W grupowaniu zostało użyte wielokrotne grupowanie ROLLUP, przez co zwracane są wiersze dla grupowania (Year), (Year, Month), (Year, Month, Day). Wynik zapytania:

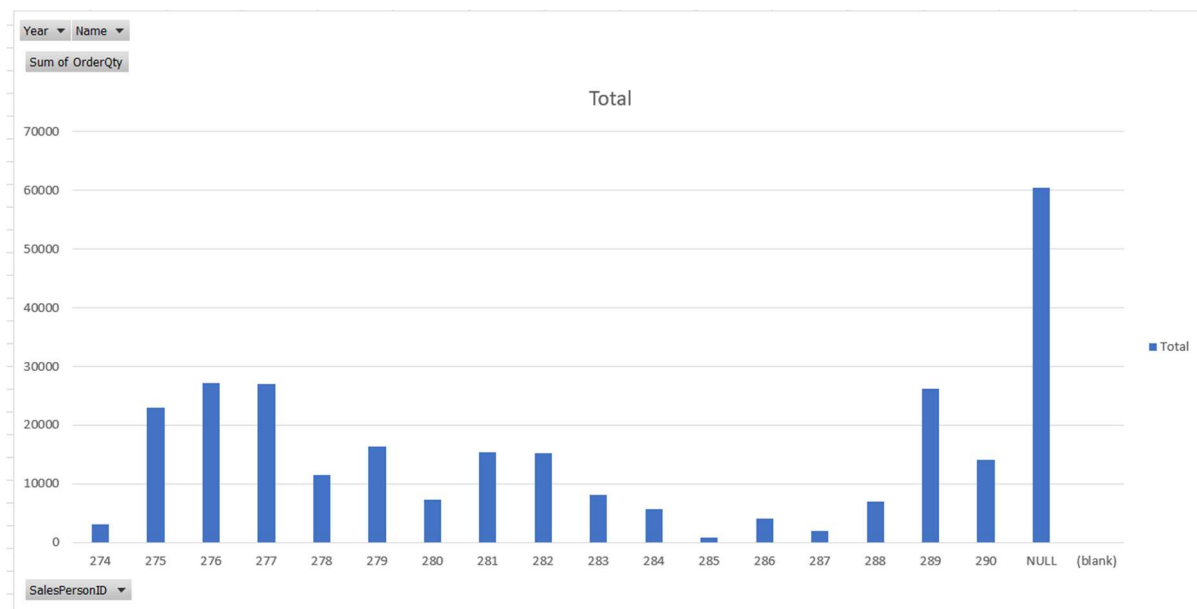
Year	Month	Day	Price sum	Distinct products
NULL	NULL	NULL	2926970124	266
2011	NULL	NULL	170783735	60
2011	5	NULL	8094970.207	47
2011	5	31	8094970.207	47
2011	6	NULL	507096.469	21
2011	6	1	15394.3298	4
2011	6	2	16588.4572	4
2011	6	3	7907.9768	2
2011	6	4	16588.4572	4
2011	6	5	15815.9536	3
2011	6	6	8680.4804	3
2011	6	7	8680.4804	2
2011	6	8	23105.3072	5
2011	6	9	11664.9658	3
2011	6	10	15815.9536	3
2011	6	11	15618.9542	4
2011	6	12	7907.9768	2
2011	6	13	27677.9188	4
2011	6	14	12409.8444	4
2011	6	15	15815.9536	2
2011	6	16	15618.9542	4
2011	6	17	7683.3524	2

.... (1167 wierszy).

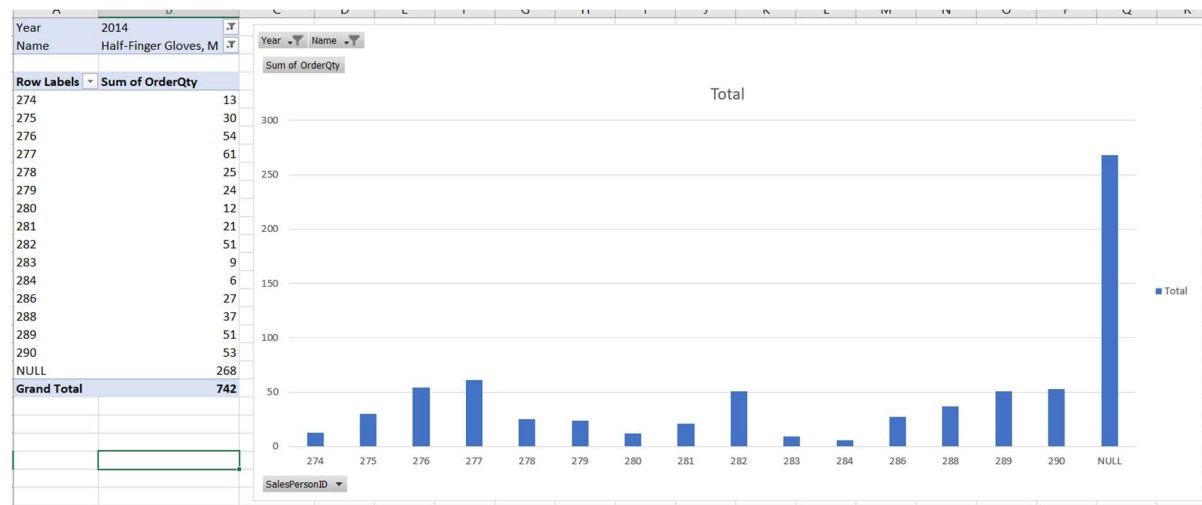
Suma kolejnych grupowań jest równa poprzednim (np. Suma dla grupy (2011, month) jest równa wartości dla wiersza (2011, NULL).

Zadanie 5. Przedstaw wyniki zadań 1-4 w postaci tabel i wykresów przestawnych w programie MS Excel.

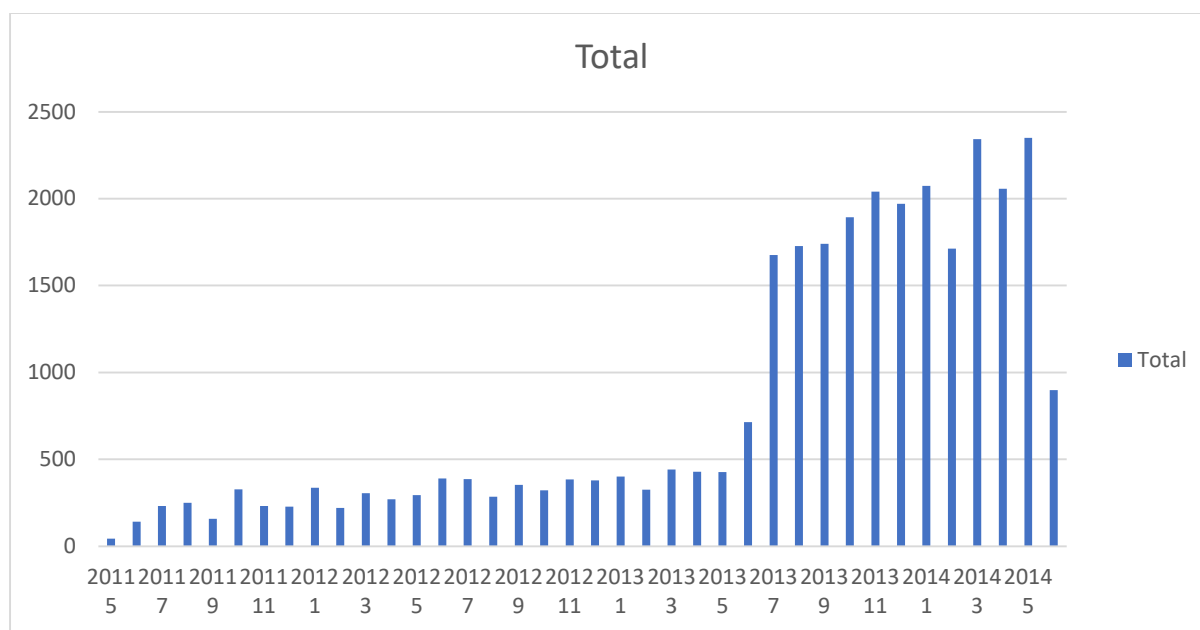
1. Liczba danych w wyniku zapytania jest zbyt duża, aby przedstawić je na wykresie. Jako przykładowy wykres przedstawiono liczbę sprzedanych produktów przez sprzedawców. Dane filtrować można po nazwie produktu oraz roku sprzedania:



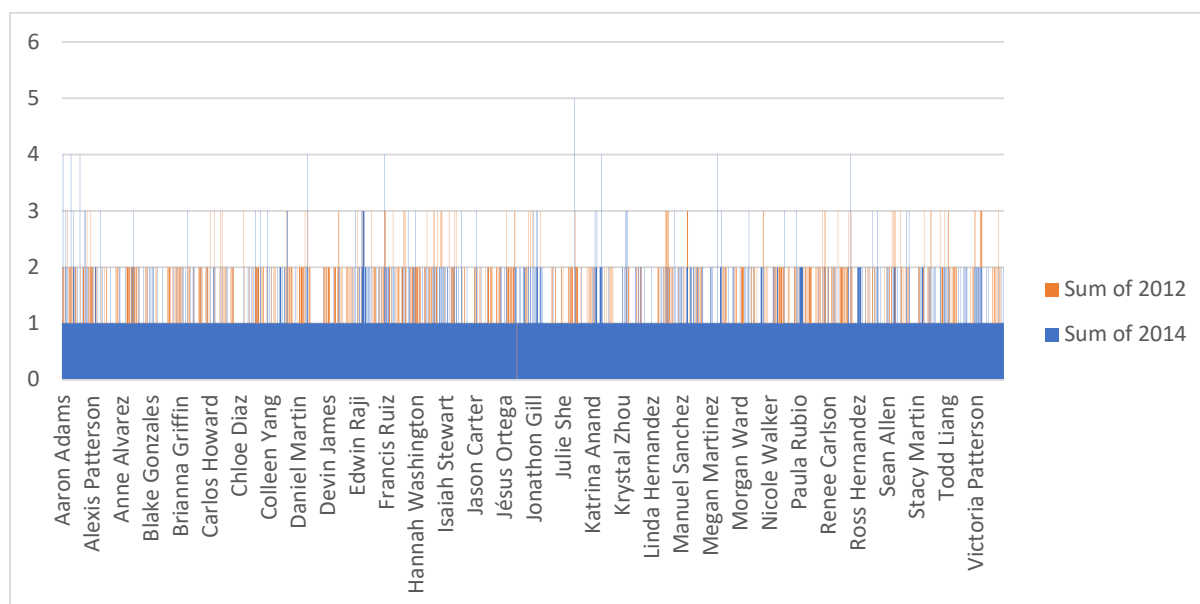
Dane przefiltrowane dla produktu Half-Finger Gloves w roku 2014:



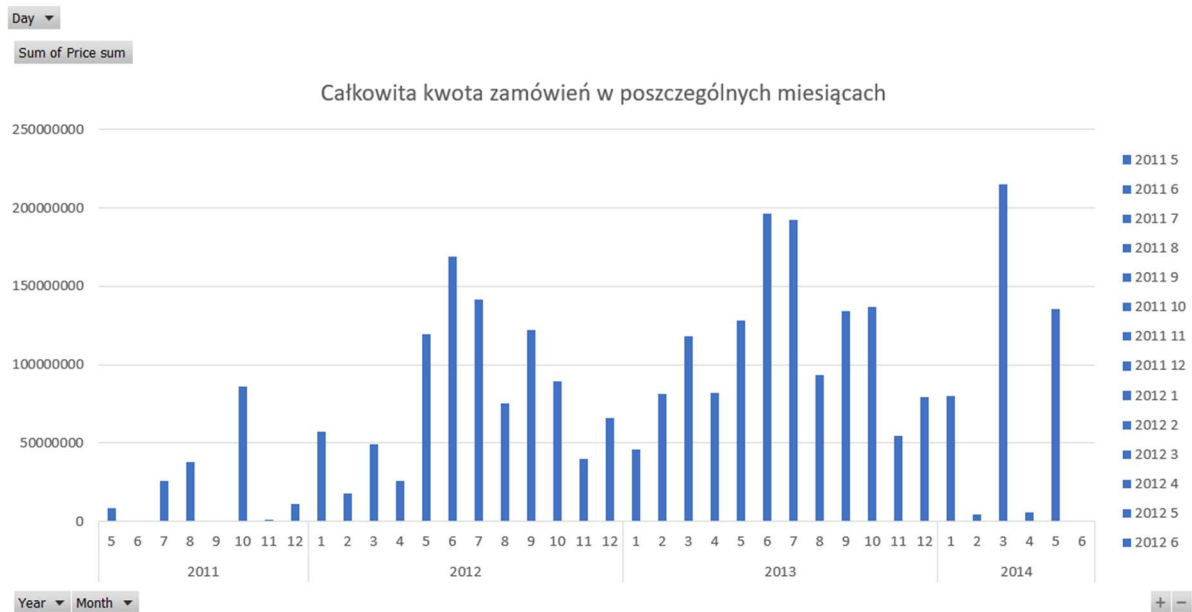
2. Wykres przedstawia liczbę klientów w każdym miesiącu w poszczególnych latach. Widoczny jest wyraźny wzrost popularności sklepu.



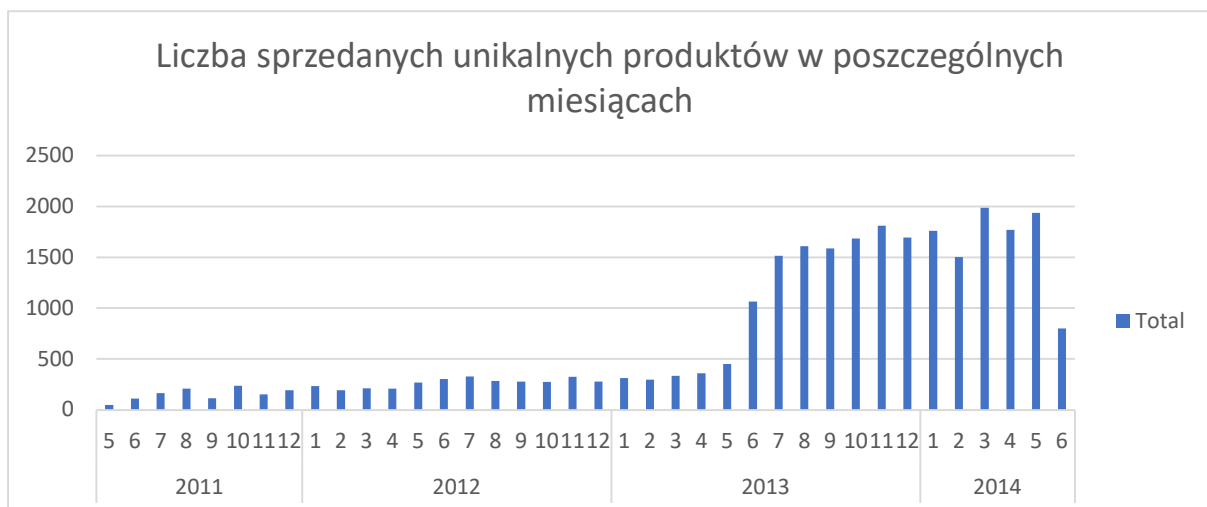
3. Ze względu na ilość danych, utworzenie przejrzystego wykresu z wyniku zapytania nie jest możliwe. Poniższy wykres przedstawia wszystkich odnalezionych klientów oraz pokazuje, że liczba zrobionych przez nich zakupów nie przekraczała 4:



4. Ponownie mamy do czynienia z wielką ilością danych. Dla danych o całkowitej kwocie w poszczególnych miesiącach użyto filtru na dzień:

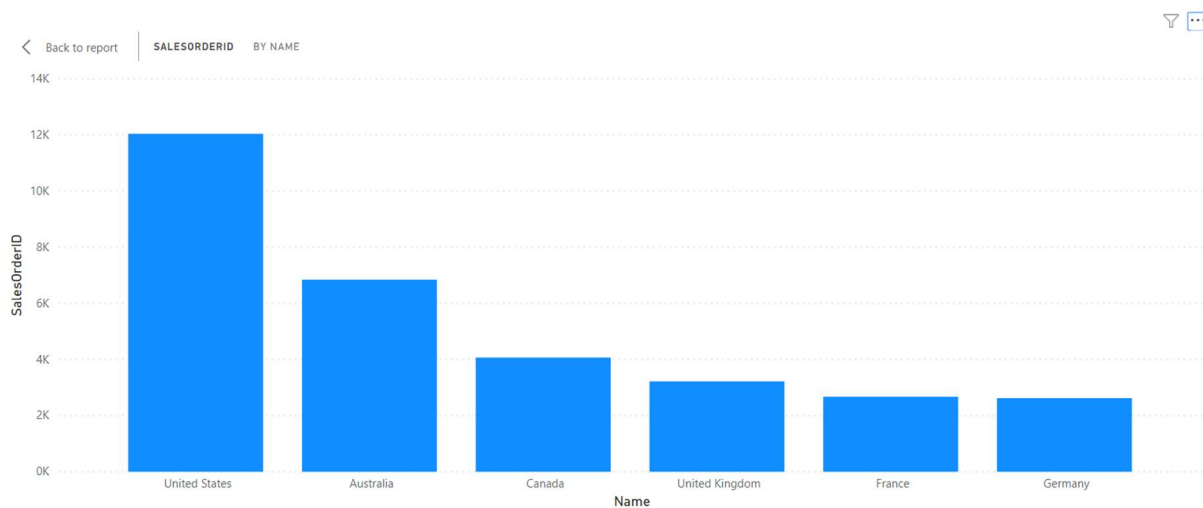


Liczba sprzedanych unikalnych produktów przedstawia rosnący trend, podobnie jak w przypadku zadania 1. :

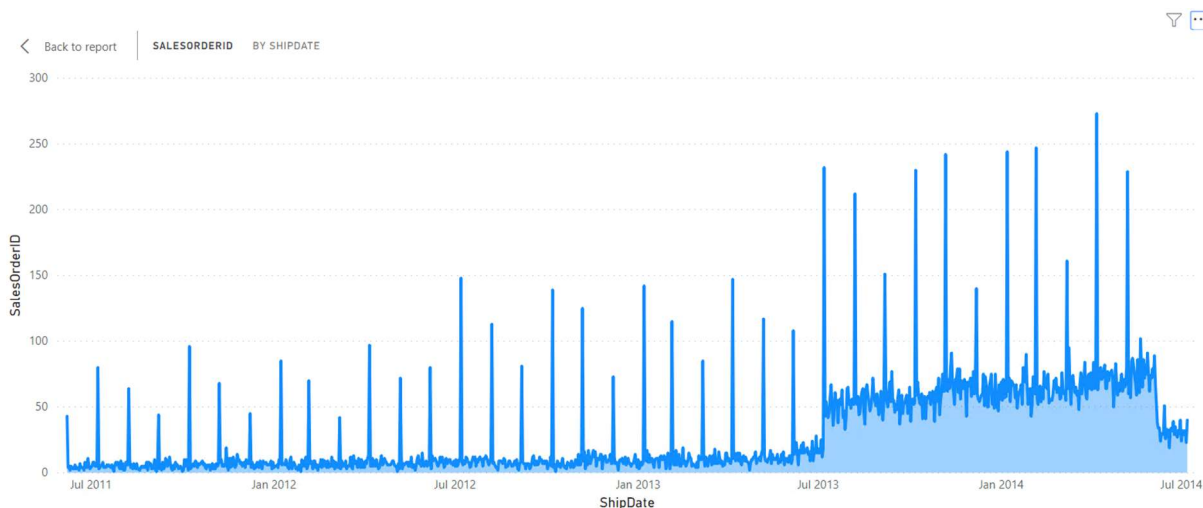


Zadanie 6. Przygotuj 5 dodatkowych tabel/wykresów, które pokażą ciekawe zależności w bazie AdventureWorks przy użyciu narzędzia Power BI lub Tableau.

Wykres przedstawiający liczbę zamówień na docelowy kraj. Przedstawione dla 6 krajów o najwyższej liczbie zamówień.



Liczba zamówień w czasie:

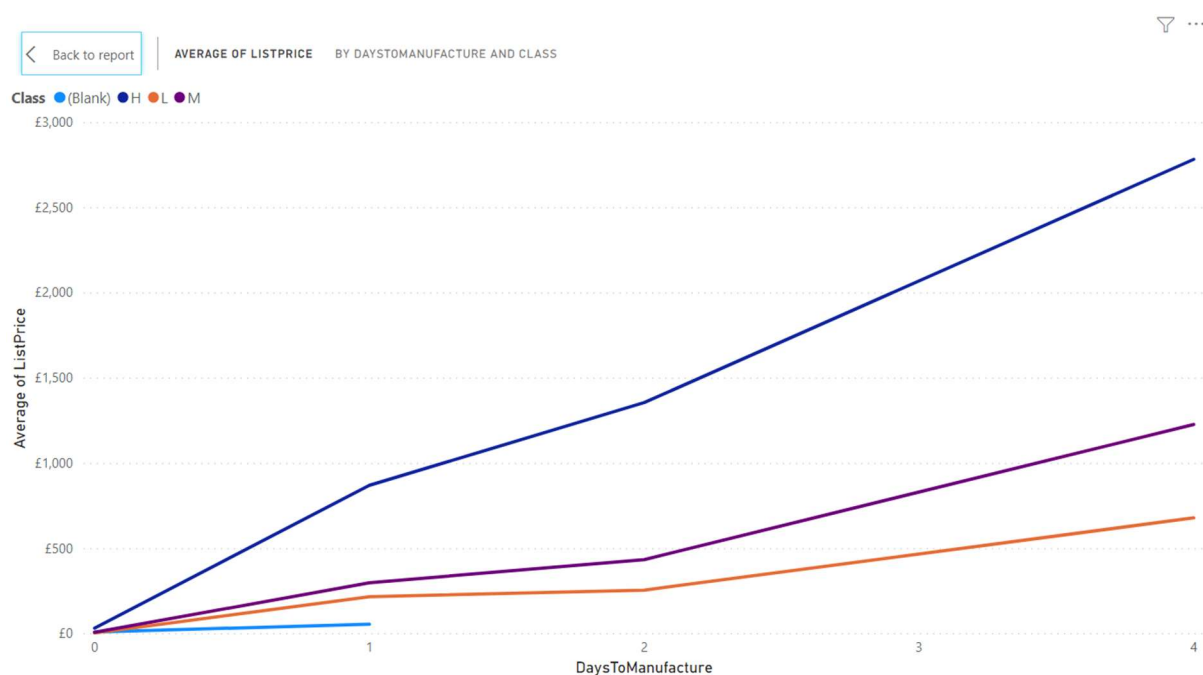


The chart displays the average delay for various order dates. The y-axis represents the 'Average of Delay' from 7.00 to 7.05. The x-axis represents the 'OrderDate' from Jul 2011 to Jan 2014. The data shows a sharp increase in average delay for the order date around Jan 2014, peaking at approximately 7.05.

OrderDate	Average of Delay
Jul 2011	7.00
Jan 2012	7.00
Jul 2012	7.00
Jan 2013	7.00
Jul 2013	7.00
Jan 2014	7.032
Feb 2014	7.052
Mar 2014	7.026
Apr 2014	7.024

[illegible]

Wykres przedstawiający rosnącą zależność ceny produktu od czasu jego wytwarzania. Dane dla każdej klasy produktowej.



Zadanie 7. Przygotować zestawienie, w którym dla wybranych klientów przygotujemy kartę lojalnościową

```
with X1(_customerId, _orderId, _orderYear, _totalDue, _isAboveAvg) as
(
    select H.CustomerID, H.SalesOrderID, Year(H.OrderDate), H.TotalDue,
           case when H.TotalDue > 1.5*AVG(H.TotalDue) over() then 1 else 0 end
    from Sales.SalesOrderHeader H
),
X2(_customerId, _orderId, _orderYear, _totalDue, _isAboveAvg, _aboveAvgCount) as
(
    select _customerId, _orderId, _orderYear, _totalDue, _isAboveAvg,
           sum(_isAboveAvg) over (partition by _customerId)
    from X1
),
X3(_customerId, _orderYear, _orderCount, _totalSum, _aboveAvgCountYear,
   _aboveAvgCount) as
(
    select _customerId, _orderYear, count(_orderId), sum(_totalDue),
           sum(_isAboveAvg), max(_aboveAvgCount)
    from X2
    group by _customerId, _orderYear
),
KartyLojalnoscione(Imie, Nazwisko, liczbaTransakcji, lacznaKwota, kolorKarty) as
(
    select max(P.FirstName), max(P.LastName), sum(_orderCount), sum(_totalSum),
    case
        when count(case when _aboveAvgCountYear > 1 then 1 end) = 4 then
        'platynowa'
        when max(_aboveAvgCount) > 1 then 'zlota'
        when sum(_orderCount) > 4 then 'srebrna' end
    from X3
    left join Person.P P
        on X3._customerId = P.BusinessEntityID
    group by _customerId
    having sum(_orderCount) > 4 or max(_aboveAvgCount) > 1
)
select * from KartyLojalnoscione
order by kolorKarty asc;
```

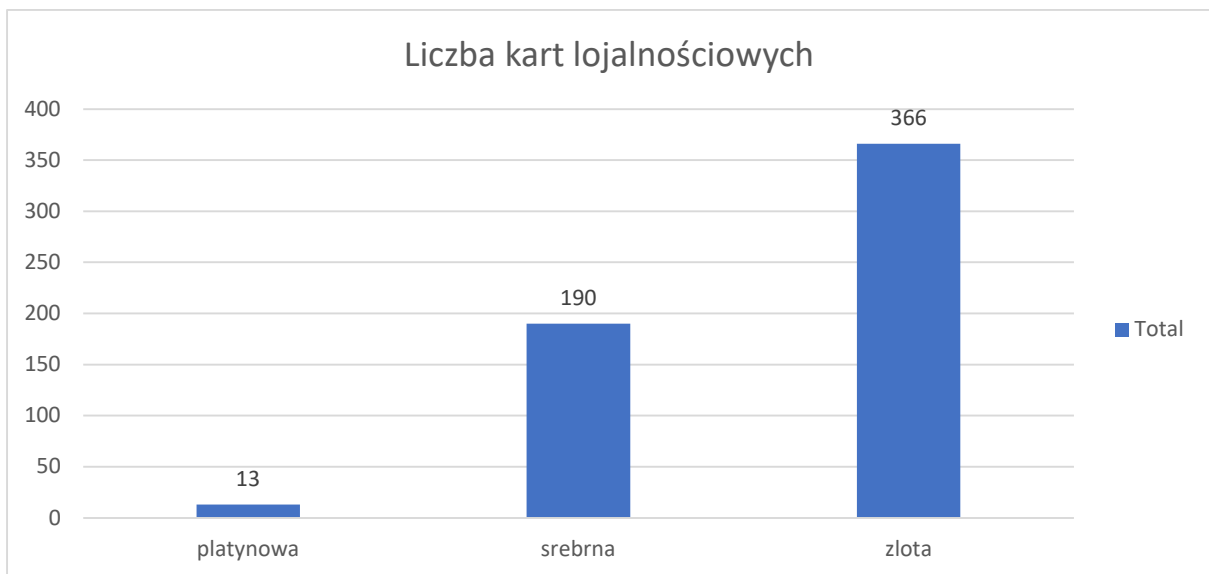
Co ciekawe, id klientów posiadających karty platynowe oraz złote nie istnieją w tabeli Person.

Wynik zapytania:

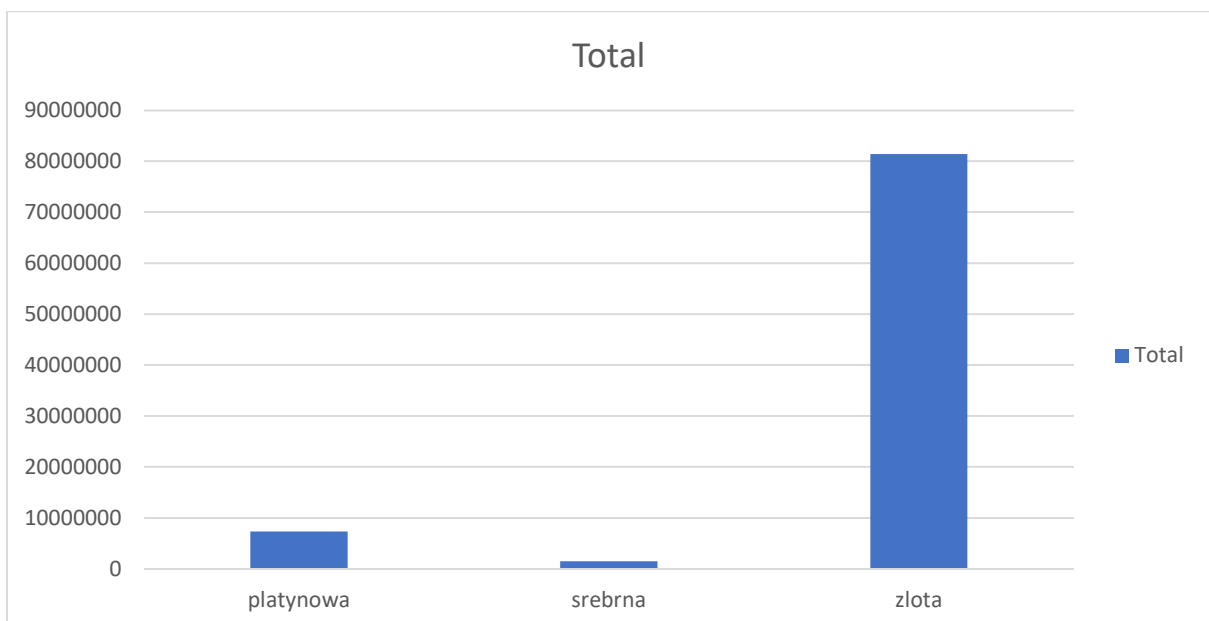
Imie	Nazwisko	liczbaTransakcji	lacznaKwota	kolorKarty
NULL	NULL	12	887090.4	platynowa
NULL	NULL	12	954021.9	platynowa
NULL	NULL	10	262408.3	platynowa
NULL	NULL	12	432800.3	platynowa
NULL	NULL	12	447403.5	platynowa
NULL	NULL	12	481357.9	platynowa
NULL	NULL	12	442256.9	platynowa
NULL	NULL	12	605358.5	platynowa
NULL	NULL	12	961675.9	platynowa
NULL	NULL	12	449023.1	platynowa
NULL	NULL	12	420767.7	platynowa
NULL	NULL	12	407563	platynowa
NULL	NULL	12	584949.1	platynowa

NULL	NULL	7	4010.949	srebrna
NULL	NULL	8	30113.53	srebrna
Barbara	Zeng	5	262.0951	srebrna
NULL	NULL	10	24082.9	srebrna
NULL	NULL	11	41979.08	srebrna
NULL	NULL	8	8393.509	srebrna
Hannah	Lee	5	14662.54	srebrna
Jenny	Shan	16	620.4912	srebrna
NULL	NULL	6	10196.2	srebrna
Morgan	Walker	17	885.9564	srebrna
NULL	NULL	6	3205.159	srebrna

Wykres przedstawiający liczbę odznak w zależności od koloru:



Pomimo najmniejszej liczby platynowych odznak, ich nosiciele pozostawili większą kwotę w sklepie, niż nosiciele kart srebrnych:



Zadanie 8 Przeanalizować, scharakteryzować i ocenić dane znajdujące się w pliku AviationData.xls profilowanie danych pakietu SSIS (projekt SQL Server Data Tools).

Ze względów technicznych, problemów z kompatybilnością wersji programów, wyjątkami object reference not set to an instance of an object w importerach i wielu innych.. nie byłem w stanie na dzień laboratoriów przygotować rozwiązania zadania.