

Hurtownie danych – Spr. 2.

PWr. WIZ, Informatyka, Data: 18.03.2020

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
Zadanie 2.....	5
Zadanie 3.....	8
Zadanie 4.....	9
Zadanie 5.....	10
Zadanie 6.....	11

Zadanie 1.

Przygotować zestawienie przedstawiające, ile pieniędzy wydali klienci na zamówienia na przestrzeni poszczególnych lat. Wykonaj zestawienie przy użyciu polecenia:

- ROLLUP
- CUBE
- GROUPING SETS

ROLLUP:

```
select
    P.FirstName + ' ' + P.LastName as "Klient",
    Year(H.OrderDate) as "Rok",
    SUM(H.SubTotal) as "Kwota"
from Sales.SalesOrderHeader H
join Person.Person P
    on P.BusinessEntityID = H.CustomerID
group by rollup(P.FirstName + ' ' + P.LastName, Year(H.OrderDate))
order by P.FirstName + ' ' + P.LastName, Year(H.OrderDate);
```

Wynik zapytania:

Klient	Rok	Kwota
NULL	NULL	19696496.0482
Aaron Adams	NULL	2381.94
Aaron Adams	2014	2381.94
Aaron Allen	NULL	1735.98
Aaron Allen	2013	1735.98
Aaron Baker	NULL	15.23
Aaron Baker	2013	15.23
Aaron Campbell	NULL	102.57
Aaron Campbell	2014	102.57
Aaron Carter	NULL	2959.0025
Aaron Carter	2012	2181.5625
Aaron Carter	2014	777.44
Aaron Collins	NULL	196.40

(24 108 wierszy)

Grupowanie ROLLUP spowodowało, że poza grupowaniem (Klient, Rok) odbywa się również grupowanie po samym (Klient) oraz po niczym(). Pozwala to na w prosty sposób wyświetlenie całkowitej kwoty danego klienta przez wszystkie lata oraz łączną sumę kwoty wszystkich klientów przez wszystkie lata.

CUBE:

```
select
    P.FirstName + ' ' + P.LastName as "Klient",
    Year(H.OrderDate) as "Rok",
    SUM(H.SubTotal) as "Kwota"
from Sales.SalesOrderHeader H
join Person.Person P
    on P.BusinessEntityID = H.CustomerID
group by cube(P.FirstName + ' ' + P.LastName, Year(H.OrderDate))
order by P.FirstName + ' ' + P.LastName, Year(H.OrderDate);
```

Wynik zapytania:

Klient	Rok	Kwota
NULL	NULL	19696496.0482
NULL	2011	2307186.6726
NULL	2012	4398597.7465
NULL	2013	7492180.9891
NULL	2014	5498530.64
Aaron Adams	NULL	2381.94
Aaron Adams	2014	2381.94
Aaron Allen	NULL	1735.98
Aaron Allen	2013	1735.98
Aaron Baker	NULL	15.23
Aaron Baker	2013	15.23
Aaron Campbell	NULL	102.57
Aaron Campbell	2014	102.57

(24 112 wierszy)

Operacja CUBE dodatkowo do grupowań podanych wyżej, używa kombinacji (Rok), co pozwala na wyświetlenie łącznej sumy kwot w każdym roku.

GROUPING SETS:

```
select
    P.FirstName + ' ' + P.LastName as "Klient",
    Year(H.OrderDate) as "Rok",
    SUM(H.SubTotal) as "Kwota"
from Sales.SalesOrderHeader H
join Person.Person P
    on P.BusinessEntityID = H.CustomerID
group by GROUPING sets(
    (P.FirstName + ' ' + P.LastName, Year(H.OrderDate)),
    ()
)
order by P.FirstName + ' ' + P.LastName, Year(H.OrderDate);
```

Wynik zapytania:

Klient	Rok	Kwota
NULL	NULL	19696496.0482
Aaron Adams	2014	2381.94
Aaron Allen	2013	1735.98
Aaron Baker	2013	15.23
Aaron Campbell	2014	102.57
Aaron Carter	2012	2181.5625
Aaron Carter	2014	777.44
Aaron Collins	2014	196.40
Aaron Edwards	2012	2181.5625
Aaron Edwards	2013	4723.04
Aaron Evans	2012	2181.5625
Aaron Evans	2013	4847.51
Aaron Gonzalez	2012	2181.5625

(14 353 wierszy)

Użycie grupowania GROUPING SETS pozwala na dowolność wyboru grup. W powyższym przykładzie użyto grup (Klient, Rok) oraz (), co pozwala na wyświetlenie łącznej kwoty wszystkich klientów oraz kwoty klientów w poszczególnych latach. Zapytanie nie byłoby możliwe z użyciem ROLLUP czy CUBE.

Zadanie 2.

Przygotować zestawienie przedstawiające łączną kwotę zniżek z podziałem na kategorie, produkty oraz lata. Wykonaj zestawienie przy użyciu polecenia:

- ROLLUP
- CUBE
- GROUPING SETS

ROLLUP:

```
select
    PC.Name "Kategoria",
    P.Name "Produkt",
    YEAR(SO.StartDate) "Rok",
    SUM(SO.DiscountPct * P.ListPrice) "Kwota"
from Sales.SpecialOffer SO
join Sales.SpecialOfferProduct SOP
    on SOP.SpecialOfferID = SO.SpecialOfferID
join Production.Product P
    on SOP.ProductID = P.ProductID
join Production.ProductSubcategory PS
    on P.ProductSubcategoryID = PS.ProductSubcategoryID
join Production.ProductCategory PC
    on PS.ProductCategoryID = PC.ProductCategoryID
group by ROLLUP(PC.Name, P.Name, YEAR(SO.StartDate))
order by PC.Name, P.Name, YEAR(SO.StartDate);
```

Wynik zapytania:

Kategoria	Produkt	Rok	Kwota
NULL	NULL	NULL	21695.289
Accessories	NULL	NULL	105.9751
Accessories	All-Purpose Bike Stand	NULL	0.00
Accessories	All-Purpose Bike Stand	2011	0.00
Accessories	Bike Wash - Dissolver	NULL	1.3515
Accessories	Bike Wash - Dissolver	2011	1.3515
Accessories	Cable Lock	NULL	0.50
Accessories	Cable Lock	2011	0.50
Accessories	Fender Set - Mountain	NULL	0.00
Accessories	Fender Set - Mountain	2011	0.00
Accessories	Headlights - Dual-Beam	NULL	0.00
Accessories	Headlights - Dual-Beam	2011	0.00
Accessories	Headlights - Weatherproof	NULL	0.00
Accessories	Headlights - Weatherproof	2011	0.00
Accessories	Hitch Rack - 4-Bike	NULL	20.40

(653 wierszy)

CUBE:

```
select
    PC.Name "Kategoria",
    P.Name "Produkt",
    YEAR(SO.StartDate) "Rok",
    SUM(SO.DiscountPct * P.ListPrice) "Kwota"
from Sales.SpecialOffer SO
join Sales.SpecialOfferProduct SOP
    on SOP.SpecialOfferID = SO.SpecialOfferID
join Production.Product P
    on SOP.ProductID = P.ProductID
join Production.ProductSubcategory PS
    on P.ProductSubcategoryID = PS.ProductSubcategoryID
join Production.ProductCategory PC
    on PS.ProductCategoryID = PC.ProductCategoryID
group by CUBE(PC.Name, P.Name, YEAR(SO.StartDate))
order by PC.Name, P.Name, YEAR(SO.StartDate);
```

Wynik zapytania:

Kategoria	Produkt	Rok	Kwota
NULL	NULL	NULL	21695.289
NULL	NULL	2011	4277.0455
NULL	NULL	2012	9730.366
NULL	NULL	2013	4721.9055
NULL	NULL	2014	2965.972
NULL	All-Purpose Bike Stand	NULL	0.00
NULL	All-Purpose Bike Stand	2011	0.00
NULL	AWC Logo Cap	NULL	1.5283
NULL	AWC Logo Cap	2011	1.5283
NULL	Bike Wash - Dissolver	NULL	1.3515
NULL	Bike Wash - Dissolver	2011	1.3515
NULL	Cable Lock	NULL	0.50
NULL	Cable Lock	2011	0.50
NULL	Chain	NULL	0.4048
NULL	Chain	2011	0.4048

(1315 wierszy)

GROUPING SETS:

```
select
    PC.Name "Kategoria",
    P.Name "Produkt",
    YEAR(SO.StartDate) "Rok",
    SUM(SO.DiscountPct * P.ListPrice) "Kwota"
from Sales.SpecialOffer SO
join Sales.SpecialOfferProduct SOP
    on SOP.SpecialOfferID = SO.SpecialOfferID
join Production.Product P
    on SOP.ProductID = P.ProductID
join Production.ProductSubcategory PS
    on P.ProductSubcategoryID = PS.ProductSubcategoryID
join Production.ProductCategory PC
    on PS.ProductCategoryID = PC.ProductCategoryID
group by GROUPING SETS(
    (PC.Name), (P.Name), (YEAR(SO.StartDate))
)
order by PC.Name, P.Name, YEAR(SO.StartDate);
```

Wynik zapytania:

Kategoria	Produkt	Rok
NULL	NULL	2011
NULL	NULL	2012
NULL	NULL	2013
NULL	NULL	2014
NULL	All-Purpose Bike Stand	NULL
NULL	AWC Logo Cap	NULL
NULL	Bike Wash - Dissolver	NULL
NULL	Cable Lock	NULL
NULL	Chain	NULL
NULL	Classic Vest, L	NULL
NULL	Classic Vest, M	NULL
NULL	Classic Vest, S	NULL
NULL	Fender Set - Mountain	NULL
NULL	Front Brakes	NULL
NULL	Front Derailleur	NULL

(303 wierszy)

Zadanie 3.

Dla kategorii 'Bikes' przygotuj zestawienie prezentujące procentowy udział kwot sprzedaży produktów tej kategorii w poszczególnych latach w stosunku do łącznej kwoty sprzedaży dla tej kategorii. W zadaniu wykorzystaj funkcje okna.

```
select distinct
    PC.Name,
    YEAR(H.OrderDate) as "Year",
    SUM(H.SubTotal) over (partition by YEAR(H.OrderDate))
    / SUM(H.SubTotal) over ()*100 as "Percentage"
from Sales.SalesOrderHeader H
join Sales.SalesOrderDetail D
    on H.SalesOrderID = D.SalesOrderID
join Production.Product P
    on D.ProductID = P.ProductID
join Production.ProductSubcategory PS
    on P.ProductSubcategoryID = PS.ProductSubcategoryID
join Production.ProductCategory PC
    on PS.ProductCategoryID = PC.ProductCategoryID
    and PC.Name = 'Bikes'
order by PC.Name, YEAR(H.OrderDate);
```

Wynik zapytania:

Name	Year	Percentage
Bikes	2011	7.94
Bikes	2012	34.51
Bikes	2013	42.56
Bikes	2014	14.97

Zadanie 4.

Przygotuj zestawienie prezentujące narastającą liczbę zamówień wykonanych przez klienta w poszczególnych latach. W zadaniu wykorzystaj funkcje okna.

```
with OrdersPerYear(CustomerId, OrderYear, OrderCount) as (  
    select  
        H.CustomerId,  
        YEAR(H.OrderDate),  
        Count(H.SalesOrderID)  
    from Sales.SalesOrderHeader H  
    join Person.Person P  
        on P.BusinessEntityID = H.CustomerID  
    group by H.CustomerId, YEAR(H.OrderDate)  
)  
select  
    CustomerId,  
    OrderYear,  
    Sum(OrderCount) over (partition by CustomerId  
                          order by OrderYear  
                          rows between unbounded preceding and current row)  
                          "Order count cumulatively"  
from OrdersPerYear  
order by CustomerId, OrderYear;
```

Wynik zapytania:

CustomerId	OrderYear	Order count cumulatively
11000	2011	1
11000	2013	3
11001	2011	1
11001	2013	2
11001	2014	3
11002	2011	1
11002	2013	3
11003	2011	1
11003	2013	3
11004	2011	1
11004	2013	3
11005	2011	1
11005	2013	3
11006	2011	1
11006	2013	3

(14 374 wierszy)

Zadanie 5.

Przygotuj zestawienie dla sprzedawców z podziałem na lata i miesiące prezentujące liczbę obsłużonych przez nich zamówień w ciągu roku, w ciągu roku narastająco oraz sumarycznie w obecnym i poprzednim miesiącu. W zadaniu wykorzystaj funkcje okna.

```
with SalesInMonth(SellerName, "Year", "Month", "CountOfSales") as (
    select
        P.FirstName + ' ' + P.LastName,
        Year(SOH.OrderDate),
        Month(SOH.OrderDate),
        Count(SOH.SalesOrderID)
    from Sales.SalesOrderHeader SOH
    join Person.Person P
        on P.BusinessEntityID = SOH.SalesPersonID
    group by
        P.FirstName + ' ' + P.LastName,
        Year(SOH.OrderDate),
        Month(SOH.OrderDate)
)
select
    SellerName as "Imie i nazwisko",
    "Year" as "Rok",
    "Month" as "Miesiac",
    CountOfSales as "W miesiacu",
    SUM(CountOfSales) over (partition by SellerName, "Year") as "W roku",
    SUM(CountOfSales) over (partition by SellerName, "Year"
                            order by "Month"
                            rows between unbounded preceding and current row)
        as "W roku narastajaco",
    SUM(CountOfSales) over (partition by SellerName, "Year"
                            order by "Month"
                            rows between 1 preceding and current row)
        as "Obecny i poprzedni miesiac"
from SalesInMonth
order by
    SellerName,
    "Year",
    "Month";
```

Wynik zapytania:

Imie i nazwisko	Rok	Miesiac
Amy Alberts	2012	6
Amy Alberts	2012	9
Amy Alberts	2012	12
Amy Alberts	2013	1
Amy Alberts	2013	2
Amy Alberts	2013	3
Amy Alberts	2013	4
Amy Alberts	2013	5
Amy Alberts	2013	6
Amy Alberts	2013	7
Amy Alberts	2013	8
Amy Alberts	2013	9
Amy Alberts	2013	10
Amy Alberts	2013	11
Amy Alberts	2013	12

(423 wierszy)

Zadanie 6.

Przygotuj zestawienie dla kategorii produktów prezentujące sumę maksymalnych cen produktów w poszczególnych podkategoriach należących danej tej kategorii. W zadaniu wykorzystaj funkcje okna.

```
with CategoryAndMaxPrices(CategoryName, MaxPrice) as (  
    select distinct  
        PC.Name,  
        MAX(P.ListPrice) over (partition by PS.ProductSubcategoryID)  
    from Production.ProductCategory PC  
    left join Production.ProductSubcategory PS  
        on PC.ProductCategoryID = PS.ProductCategoryID  
    left join Production.Product P  
        on P.ProductSubcategoryID = PS.ProductSubcategoryID  
)  
select CategoryName "Kategoria", SUM(MaxPrice) "Suma"  
from CategoryAndMaxPrices  
group by CategoryName;
```

Wynik zapytania:

Kategoria	Suma
Accessories	663.88
Bikes	9362.33
Clothing	408.94
Components	5539.77