

Hurtownie danych  
Laboratorium Czw 11:15

Lista 3

Kajetan Pynka 254495

## Zad 1.1a

```
SELECT ISNULL(P.FirstName + ' ' + P.LastName, '') "Klient",
ISNULL(STR(YEAR(SOH.DueDate)), '') "Rok",
SUM(SOH.TotalDue) "Kwota"
FROM Sales.SalesOrderHeader SOH
JOIN Sales.Customer C ON C.CustomerID=SOH.CustomerID
JOIN Person.Person P ON P.BusinessEntityID=C.PersonID
GROUP BY GROUPING SETS (
    (YEAR(SOH.DueDate)),
    (P.FirstName + ' ' + P.LastName, YEAR(SOH.DueDate)),
    (P.FirstName + ' ' + P.LastName),
    ()
)
ORDER BY 1;
```

	Klient	Rok	Kwota
1			123216786,1159
2		2013	48181124,1984
3		2014	26072957,9986
4		2011	13903981,3182
5		2012	35058722,6007
6	A. Leonetti	2013	1814,1819
7	A. Leonetti	2014	1586,6583
8	A. Leonetti		3400,8402
9	Aaron Adams	2013	130,3458
10	Aaron Adams		130,3458
11	Aaron Alexander	2014	77,339
12	Aaron Alexander		77,339

## Zad 1.1b

```
SELECT ISNULL(P.FirstName + ' ' + P.LastName, '') "Klient",  
ISNULL(STR(YEAR(SOH.DueDate)), '') "Rok",  
SUM(SOH.TotalDue) "Kwota"  
FROM Sales.SalesOrderHeader SOH  
JOIN Sales.Customer C ON C.CustomerID=SOH.CustomerID  
JOIN Person.Person P ON P.BusinessEntityID=C.PersonID  
GROUP BY CUBE (YEAR(SOH.DueDate), P.FirstName + ' ' + P.LastName)  
ORDER BY 1;
```

	Klient	Rok	Kwota
1			123216786,1159
2		2013	48181124,1984
3		2014	26072957,9986
4		2011	13903981,3182
5		2012	35058722,6007
6	A. Leonetti	2013	1814,1819
7	A. Leonetti	2014	1586,6583
8	A. Leonetti		3400,8402
9	Aaron Adams	2013	130,3458
10	Aaron Adams		130,3458
11	Aaron Alexander	2014	77,339
12	Aaron Alexander		77,339

## Zad 1.1c

```
SELECT ISNULL(P.FirstName + ' ' + P.LastName, '') "Klient",  
ISNULL(STR(YEAR(SOH.DueDate)), '') "Rok",  
SUM(SOH.TotalDue) "Kwota"  
FROM Sales.SalesOrderHeader SOH  
JOIN Sales.Customer C ON C.CustomerID=SOH.CustomerID  
JOIN Person.Person P ON P.BusinessEntityID=C.PersonID  
GROUP BY ROLLUP (YEAR(SOH.DueDate), P.FirstName + ' ' + P.LastName)  
ORDER BY 1;
```

	Klient	Rok	Kwota
1		2012	35058722,6007
2		2013	48181124,1984
3		2011	13903981,3182
4		2014	26072957,9986
5			123216786,1159
6	A. Leonetti	2014	1586,6583
7	A. Leonetti	2013	1814,1819
8	Aaron Adams	2013	130,3458
9	Aaron Alexander	2014	77,339
10	Aaron Allen	2012	3756,989
11	Aaron Baker	2014	1934,8329
12	Aaron Bryant	2014	82,8529

## Zad 1.2

```
SELECT PC.Name "Kategoria", P.Name "Produkt", ISNULL(STR(YEAR(SOH.DueDate)),
'' ) "Rok",
SUM(SOD.UnitPrice * SOD.UnitPriceDiscount * SOD.OrderQty) "Kwota"
FROM Sales.SalesOrderHeader SOH
JOIN Sales.SalesOrderDetail SOD ON SOH.SalesOrderID=SOD.SalesOrderID
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
GROUP BY GROUPING SETS (
(PC.Name, P.Name, YEAR(SOH.DueDate)),
(PC.Name, P.Name)
) ORDER BY 1, 2;
```

	Kategoria	Produkt	Rok	Kwota
1	Accessories	All-Purpose Bike Stand	2013	0,00
2	Accessories	All-Purpose Bike Stand	2014	0,00
3	Accessories	All-Purpose Bike Stand		0,00
4	Accessories	Bike Wash - Dissolver	2013	83,8663
5	Accessories	Bike Wash - Dissolver	2014	27,7164
6	Accessories	Bike Wash - Dissolver		111,5827
7	Accessories	Cable Lock	2012	20,30
8	Accessories	Cable Lock	2013	3,48
9	Accessories	Cable Lock		23,78
10	Accessories	Fender Set - Mountain	2013	0,00
11	Accessories	Fender Set - Mountain	2014	0,00
12	Accessories	Fender Set - Mountain		0,00

## Zad 2.1a

```
SELECT PC.Name, STR(YEAR(SOH.OrderDate)) "Rok",  
       SUM(SOD.UnitPrice * SOD.OrderQty * (1 - SOD.UnitPriceDiscount)) OVER  
(PARTITION BY YEAR(SOH.OrderDate))  
       / SUM(SOD.UnitPrice * SOD.OrderQty * (1 - SOD.UnitPriceDiscount))  
OVER()*100 "Procent"  
FROM Sales.SalesOrderHeader SOH  
JOIN Sales.SalesOrderDetail SOD ON SOH.SalesOrderID=SOD.SalesOrderID  
JOIN Production.Product P ON P.ProductID=SOD.ProductID  
JOIN Production.ProductSubcategory PSC ON  
P.ProductSubcategoryID=PSC.ProductSubcategoryID  
JOIN Production.ProductCategory PC ON  
PC.ProductCategoryID=PSC.ProductCategoryID  
WHERE PC.Name='Bikes'  
UNION SELECT '', '', 100;
```

	Name	Rok	Procent
1	Bikes	2013	38,31
2	Bikes	2011	12,62
3	Bikes	2012	30,62
4	Bikes	2014	18,43
5			100,00

## Zad 2.1b

```
SELECT PC.Name, STR(YEAR(SOH.OrderDate)) "Rok",  
       SUM(SOD.UnitPrice * SOD.OrderQty * (1 - SOD.UnitPriceDiscount)) OVER  
(PARTITION BY YEAR(SOH.OrderDate))  
       / SUM(SOD.UnitPrice * SOD.OrderQty * (1 - SOD.UnitPriceDiscount))  
OVER()*100 "Procent"  
FROM Sales.SalesOrderHeader SOH  
JOIN Sales.SalesOrderDetail SOD ON SOH.SalesOrderID=SOD.SalesOrderID  
JOIN Production.Product P ON P.ProductID=SOD.ProductID  
JOIN Production.ProductSubcategory PSC ON  
P.ProductSubcategoryID=PSC.ProductSubcategoryID  
JOIN Production.ProductCategory PC ON  
PC.ProductCategoryID=PSC.ProductCategoryID  
WHERE PC.Name='Accessories'  
UNION SELECT '', '', 100;
```

	Name	Rok	Procent
1	Accessories	2013	53,06
2	Accessories	2012	8,05
3	Accessories	2011	1,63
4	Accessories	2014	37,24
5			100,00

## Zad 2.1c

```
SELECT PC.Name, STR(YEAR(SOH.OrderDate)) "Rok",  
       SUM(SOD.UnitPrice * SOD.OrderQty * (1 - SOD.UnitPriceDiscount)) OVER  
(PARTITION BY YEAR(SOH.OrderDate))  
       / SUM(SOD.UnitPrice * SOD.OrderQty * (1 - SOD.UnitPriceDiscount))  
OVER()*100 "Procent"  
FROM Sales.SalesOrderHeader SOH  
JOIN Sales.SalesOrderDetail SOD ON SOH.SalesOrderID=SOD.SalesOrderID  
JOIN Production.Product P ON P.ProductID=SOD.ProductID  
JOIN Production.ProductSubcategory PSC ON  
P.ProductSubcategoryID=PSC.ProductSubcategoryID  
JOIN Production.ProductCategory PC ON  
PC.ProductCategoryID=PSC.ProductCategoryID  
WHERE PC.Name='Clothing'  
UNION SELECT '', '', 100;
```

	Name	Rok	Procent
1	Clothing	2013	50,34
2	Clothing	2012	26,20
3	Clothing	2014	21,75
4	Clothing	2011	1,69
5			100,00



## Zad 2.1d

```
SELECT PC.Name, STR(YEAR(SOH.OrderDate)) "Rok",  
       SUM(SOD.UnitPrice * SOD.OrderQty * (1 - SOD.UnitPriceDiscount)) OVER  
(PARTITION BY YEAR(SOH.OrderDate))  
       / SUM(SOD.UnitPrice * SOD.OrderQty * (1 - SOD.UnitPriceDiscount))  
OVER()*100 "Procent"  
FROM Sales.SalesOrderHeader SOH  
JOIN Sales.SalesOrderDetail SOD ON SOH.SalesOrderID=SOD.SalesOrderID  
JOIN Production.Product P ON P.ProductID=SOD.ProductID  
JOIN Production.ProductSubcategory PSC ON  
P.ProductSubcategoryID=PSC.ProductSubcategoryID  
JOIN Production.ProductCategory PC ON  
PC.ProductCategoryID=PSC.ProductCategoryID  
WHERE PC.Name='Components'  
UNION SELECT '', '', 100;
```

	Name	Rok	Procent
1	Components	2014	14,14
2	Components	2012	32,88
3	Components	2013	47,55
4	Components	2011	5,41
5			100,00

## Zad 2.2

```
SELECT P.FirstName + ' ' + P.LastName "Klient", YEAR(SOH.DueDate) "Rok",  
       MAX(COUNT(SOH.SalesOrderID)) OVER(PARTITION BY SOH.CustomerID,  
       YEAR(SOH.DueDate)  
       ORDER BY SOH.CustomerID, YEAR(SOH.DueDate) ROWS UNBOUNDED  
PRECEDING) "Suma transakcji"  
FROM Sales.SalesOrderHeader SOH  
JOIN Sales.Customer C ON C.CustomerID=SOH.CustomerID  
JOIN Person.Person P ON P.BusinessEntityID=C.PersonID  
GROUP BY P.FirstName + ' ' + P.LastName, YEAR(SOH.DueDate), SOH.CustomerID  
ORDER BY 1,2;
```

	Klient	Rok	Suma transakcji
1	A. Leonetti	2013	1
2	A. Leonetti	2014	2
3	Aaron Adams	2013	1
4	Aaron Alexander	2014	1
5	Aaron Allen	2012	1
6	Aaron Baker	2014	1
7	Aaron Bryant	2013	1
8	Aaron Bryant	2014	1
9	Aaron Butler	2014	1
10	Aaron Campbell	2014	1
11	Aaron Carter	2014	1
12	Aaron Chen	2013	1

## Zad 2.3

```
SELECT "Imię i nazwisko", "Rok", "Miesiąc", "W miesiącu", "W roku",
      MAX("W roku narastająco") "W roku narastająco" ,
      ISNULL(LAG("W miesiącu") OVER (ORDER BY "Imię i nazwisko", "Rok",
"Miesiąc"), 0) + "W miesiącu" "Obecny i poprzedni miesiąc"
FROM (SELECT P.FirstName + ' ' + P.LastName "Imię i nazwisko",
      YEAR(SOH.OrderDate) "Rok", MONTH(SOH.OrderDate) "Miesiąc",
      COUNT(SOH.SalesOrderID) OVER (PARTITION BY SOH.SalesPersonID,
YEAR(SOH.OrderDate), MONTH(SOH.OrderDate)) "W miesiącu",
      COUNT(SOH.SalesOrderID) OVER (PARTITION BY SOH.SalesPersonID,
YEAR(SOH.OrderDate)) "W roku",
      COUNT(SOH.SalesOrderID)
      OVER (PARTITION BY SOH.SalesPersonID, YEAR(SOH.OrderDate) ORDER BY
SOH.SalesPersonID, YEAR(SOH.OrderDate) ROWS BETWEEN UNBOUNDED PRECEDING AND
CURRENT ROW) "W roku narastająco"
FROM Sales.SalesOrderHeader SOH JOIN Sales.SalesPerson SP ON
SP.BusinessEntityID = SOH.SalesPersonID
JOIN Person.Person P ON P.BusinessEntityID = SP.BusinessEntityID)
GROUP BY "Imię i nazwisko", "Rok", "Miesiąc", "W miesiącu", "W roku"
ORDER BY 1,2,3;
```

	Imię i nazwisko	Rok	Miesiąc	W miesiącu	W roku	W roku narastająco	Obecny i poprzedni miesiąc
1	Amy Alberts	2012	6	3	7	3	3
2	Amy Alberts	2012	9	2	7	5	5
3	Amy Alberts	2012	12	2	7	7	4
4	Amy Alberts	2013	1	1	29	1	3
5	Amy Alberts	2013	2	1	29	2	2
6	Amy Alberts	2013	3	1	29	3	2
7	Amy Alberts	2013	4	2	29	5	3
8	Amy Alberts	2013	5	1	29	6	3
9	Amy Alberts	2013	6	5	29	11	6
10	Amy Alberts	2013	7	3	29	14	8
11	Amy Alberts	2013	8	1	29	15	4
12	Amy Alberts	2013	9	4	29	19	5
13	Amy Alberts	2013	10	4	29	23	8

## Zad 2.4

```
SELECT "Kategoria", SUM("Kwota") "Suma maksymalnych" FROM (
SELECT DISTINCT PC.Name "Kategoria", MAX(P.ListPrice) OVER(PARTITION BY
PSC.ProductSubcategoryID) "Kwota"
FROM Production.ProductCategory PC
JOIN Production.ProductSubcategory PSC ON
PSC.ProductCategoryID=PC.ProductCategoryID
JOIN Production.Product P ON
P.ProductSubcategoryID=PSC.ProductSubcategoryID) INSIDE
GROUP BY "Kategoria";
```

	Kategoria	Suma maksymalnych
1	Accessories	663,88
2	Bikes	9362,33
3	Clothing	408,94
4	Components	5539,77

## Zad 2.5a

```
SELECT RANK() OVER(ORDER BY COUNT(SOD.OrderQty) DESC) "Ranga",  
       P.FirstName + ' ' + P.LastName "Imię i nazwisko", COUNT(SOD.OrderQty)  
"Liczba transakcji"  
FROM Sales.SalesOrderHeader SOH  
JOIN Sales.SalesOrderDetail SOD ON SOH.SalesOrderID=SOD.SalesOrderID  
JOIN Sales.Customer C ON C.CustomerID=SOH.CustomerID  
JOIN Person.Person P ON P.BusinessEntityID=C.PersonID  
GROUP BY P.FirstName + ' ' + P.LastName;
```

	Ranga	Imię i nazwisko	Liczba transakcji
1	1	Reuben D'sa	530
2	2	Richard Lum	482
3	3	Ryan Calafato	451
4	4	Yale Li	446
5	5	Marcia Sultan	441
6	6	Holly Dickson	440
7	7	Robert Vessa	436
8	7	Della Demott Jr	436
9	9	Sandra Maynard	432
10	10	Joseph Castellucio	429
11	11	Blaine Dockter	422
12	12	John Evans	418

## Zad 2.5b

```
SELECT DENSE_RANK() OVER(ORDER BY COUNT(SOD.OrderQty) DESC) "Ranga",  
       P.FirstName + ' ' + P.LastName "Imię i nazwisko", COUNT(SOD.OrderQty)  
"Liczba transakcji"  
FROM Sales.SalesOrderHeader SOH  
JOIN Sales.SalesOrderDetail SOD ON SOH.SalesOrderID=SOD.SalesOrderID  
JOIN Sales.Customer C ON C.CustomerID=SOH.CustomerID  
JOIN Person.Person P ON P.BusinessEntityID=C.PersonID  
GROUP BY P.FirstName + ' ' + P.LastName;
```

	Ranga	Imię i nazwisko	Liczba transakcji
1	1	Reuben D'sa	530
2	2	Richard Lum	482
3	3	Ryan Calafato	451
4	4	Yale Li	446
5	5	Marcia Sultan	441
6	6	Holly Dickson	440
7	7	Robert Vessa	436
8	7	Della Demott Jr	436
9	8	Sandra Maynard	432
10	9	Joseph Castellucio	429
11	10	Blaine Dockter	422
12	11	John Evans	418

## Zad 2.6

```
SELECT P.Name "Nazwa produktu", AVG(SOD.OrderQty) "Średnia liczba sztuk",  
       CASE NTILE(3) OVER(ORDER BY AVG(SOD.OrderQty) DESC)  
         WHEN 1 THEN 'Najlepiej'  
         WHEN 2 THEN 'Średnio'  
         WHEN 3 THEN 'Najsłabiej'  
       END "Ranga sprzedaży"  
FROM Sales.SalesOrderHeader SOH  
JOIN Sales.SalesOrderDetail SOD ON SOD.SalesOrderID=SOH.SalesOrderID  
JOIN Production.Product P ON P.ProductID=SOD.ProductID  
GROUP BY P.Name  
ORDER BY 2 DESC;
```

82	Rear Brakes	2	Najlepiej
83	Rear Derailleur	2	Najlepiej
84	Mountain-400-W Silver, 40	2	Najlepiej
85	Mountain-200 Silver, 38	2	Najlepiej
86	Mountain-200 Silver, 42	2	Najlepiej
87	Mountain-200 Silver, 46	2	Najlepiej
88	Mountain-500 Silver, 40	2	Najlepiej
89	Mountain-500 Silver, 42	2	Najlepiej
90	Road-250 Black, 44	2	Srednio
91	Road-250 Black, 48	2	Srednio
92	Road-250 Red, 44	2	Srednio
93	Road-250 Red, 48	2	Srednio

## **Wnioski:**

- Funkcje grupujące pozwalają nam w bardzo precyzyjny sposób dobrać dane, po których chcemy pogrupować wynik.  
Korzystając z GROUPING SETS możemy np. określić by w wyniku znalazły się rekordy pogrupowane po wszystkich kolumnach, tylko po jednej czy dwóch wybranych kolumnach.
- Funkcje okienkowe pozwalają nam zaimplementować w bardzo prosty i czytelny sposób pewne narastające wartości wśród danych grup jak i pewne rankingi, które możemy przypisać rekordom. Należy jednak rozróżniać funkcje okienkowe od agregujących, ponieważ ich wpływ na strukturę ostatecznego wyniku jest zgoła odmienny.