**HURTOWNIE DANYCH**

Laboratorium 3

Maciej Kopiński 254578

**Zad. 1. Wykorzystanie funkcji grupujących (rollup, cube, grouping sets)**

1. Przygotować zestawienie przedstawiające, ile pieniędzy wydali klienci na zamówienia na przestrzeni poszczególnych lat.

Wykonaj zestawienie przy użyciu poleceń rollup, cube, grouping sets.

1. Przygotować zestawienie przedstawiające łączną kwotę zniżek z podziałem na kategorie, produkty oraz lata.

**Zad. 2. Wykorzystanie funkcji okienkowych (over, over partition by, row\_number, rank, dense\_rank, ntile)**

1. 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.

Wykonaj podobne zestawienia dla pozostałych kategorii.

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

Ogranicz wynik do 10 najlepszych klientów.

1. 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.
2. Przygotuj zestawienie dla kategorii produktów prezentujące sumę maksymalnych cen produktów w poszczególnych podkategoriach należących danej tej kategorii. W zadaniu wykorzystaj funkcje okna.
3. Przygotuj ranking klientów w zależności od liczby zakupionych produktów. Porównaj rozwiązania uzyskane przez funkcje rank i dense\_rank.
4. Przygotuj ranking produktów w zależności od średniej liczby sprzedanych sztuk. Wyróżnij 3 (prawie równoliczne) grupy produktów: sprzedających się najlepiej, średnio i najsłabiej.

**Rozwiązania:**

**Ad.1**

**1)**

**ROLLUP:**

SELECT p.FirstName + ' ' + p.LastName AS Klient, YEAR(soh.OrderDate) AS Rok, SUM(soh.TotalDue) AS Kwota

FROM Person.Person p

JOIN Sales.Customer c ON c.PersonID = p.BusinessEntityID

JOIN Sales.SalesOrderHeader soh ON soh.CustomerID = c.CustomerID

GROUP BY ROLLUP(p.FirstName + ' ' + p.LastName, YEAR(soh.OrderDate))

ORDER BY 1;

**CUBE:**

SELECT p.FirstName + ' ' + p.LastName AS Klient, YEAR(soh.OrderDate) AS Rok, SUM(soh.TotalDue) AS Kwota

FROM Person.Person p

JOIN Sales.Customer c ON c.PersonID = p.BusinessEntityID

JOIN Sales.SalesOrderHeader soh ON soh.CustomerID = c.CustomerID

GROUP BY CUBE(p.FirstName + ' ' + p.LastName, YEAR(soh.OrderDate))

ORDER BY 1, 2, 3;

**GROUPING SETS:**

SELECT p.FirstName + ' ' + p.LastName AS Klient, YEAR(soh.OrderDate) AS Rok, SUM(soh.TotalDue) AS Kwota

FROM Person.Person p

JOIN Sales.Customer c ON c.PersonID = p.BusinessEntityID

JOIN Sales.SalesOrderHeader soh ON soh.CustomerID = c.CustomerID

GROUP BY GROUPING SETS(

(),

(YEAR(soh.OrderDate)),

(p.FirstName + ' ' + p.LastName),

(p.FirstName + ' ' + p.LastName, YEAR(soh.OrderDate))

)

ORDER BY 1, 2, 3;

**2)**

SELECT pc.Name, p.Name, YEAR(soh.OrderDate), SUM(sod.UnitPriceDiscount \* sod.UnitPrice \* sod.OrderQty)

FROM Production.ProductCategory pc

JOIN Production.ProductSubcategory ps ON ps.ProductCategoryID = pc.ProductCategoryID

JOIN Production.Product p ON p.ProductSubcategoryID = ps.ProductSubcategoryID

JOIN Sales.SalesOrderDetail sod ON sod.ProductID = p.ProductID

JOIN Sales.SalesOrderHeader soh ON soh.SalesOrderID = sod.SalesOrderID

GROUP BY GROUPING SETS(

(pc.Name, p.Name, YEAR(soh.OrderDate)),

(pc.Name, p.Name)

)

ORDER BY 1, 2, 3;

**Ad.2**

**1)**

SELECT DISTINCT pc.Name AS Nazwa, YEAR(soh.OrderDate) AS Rok, 100 \* SUM(sod.LineTotal) OVER(PARTITION BY YEAR(soh.OrderDate))/SUM(sod.LineTotal) OVER() AS Procent

FROM Production.ProductCategory pc

JOIN Production.ProductSubcategory ps ON ps.ProductCategoryID = pc.ProductCategoryID

JOIN Production.Product p ON p.ProductSubcategoryID = ps.ProductSubcategoryID

JOIN Sales.SalesOrderDetail sod ON sod.ProductID = p.ProductID

JOIN Sales.SalesOrderHeader soh ON soh.SalesOrderID = sod.SalesOrderID

WHERE pc.Name='Bikes'

ORDER BY 2;

SELECT DISTINCT pc.Name AS Nazwa, YEAR(soh.OrderDate) AS Rok, 100 \* SUM(sod.LineTotal) OVER(PARTITION BY YEAR(soh.OrderDate))/SUM(sod.LineTotal) OVER() AS Procent

FROM Production.ProductCategory pc

JOIN Production.ProductSubcategory ps ON ps.ProductCategoryID = pc.ProductCategoryID

JOIN Production.Product p ON p.ProductSubcategoryID = ps.ProductSubcategoryID

JOIN Sales.SalesOrderDetail sod ON sod.ProductID = p.ProductID

JOIN Sales.SalesOrderHeader soh ON soh.SalesOrderID = sod.SalesOrderID

WHERE pc.Name='Accessories'

ORDER BY 2;

SELECT DISTINCT pc.Name AS Nazwa, YEAR(soh.OrderDate) AS Rok, 100\* SUM(sod.LineTotal) OVER(PARTITION BY YEAR(soh.OrderDate))/SUM(sod.LineTotal) OVER() AS Procent

FROM Production.ProductCategory pc

JOIN Production.ProductSubcategory ps ON ps.ProductCategoryID = pc.ProductCategoryID

JOIN Production.Product p ON p.ProductSubcategoryID = ps.ProductSubcategoryID

JOIN Sales.SalesOrderDetail sod ON sod.ProductID = p.ProductID

JOIN Sales.SalesOrderHeader soh ON soh.SalesOrderID = sod.SalesOrderID

WHERE pc.Name='Clothing'

ORDER BY 2;

SELECT DISTINCT pc.Name AS Nazwa, YEAR(soh.OrderDate) AS Rok, 100 \* SUM(sod.LineTotal) OVER(PARTITION BY YEAR(soh.OrderDate))/SUM(sod.LineTotal) OVER() AS Procent

FROM Production.ProductCategory pc

JOIN Production.ProductSubcategory ps ON ps.ProductCategoryID = pc.ProductCategoryID

JOIN Production.Product p ON p.ProductSubcategoryID = ps.ProductSubcategoryID

JOIN Sales.SalesOrderDetail sod ON sod.ProductID = p.ProductID

JOIN Sales.SalesOrderHeader soh ON soh.SalesOrderID = sod.SalesOrderID

WHERE pc.Name='Components'

ORDER BY 2;

**2)**

SELECT DISTINCT p.FirstName + ' ' + p.LastName AS Klient, YEAR(soh.OrderDate) AS Rok, COUNT(soh.SalesOrderID) OVER(PARTITION BY c.CustomerID ORDER BY YEAR(soh.OrderDate)) AS Liczba

FROM Person.Person p

JOIN Sales.Customer c ON c.PersonID = p.BusinessEntityID

JOIN Sales.SalesOrderHeader soh ON soh.CustomerID = c.CustomerID

WHERE c.CustomerID IN (

SELECT CustomerID FROM(

SELECT DISTINCT TOP 10 x.CustomerID, COUNT(x.CustomerID) OVER(PARTITION BY x.CustomerID) y

FROM Sales.SalesOrderHeader x

ORDER BY 2 DESC) P)

ORDER BY 1, 2;

**3)**

SELECT "Imię i nazwisko", Rok, Miesiąc, "W miesiącu", "W roku", "W roku narastająco",

SUM("W miesiącu") OVER (PARTITION BY "Imię i nazwisko", Rok ORDER BY Miesiąc ROWS BETWEEN 1 PRECEDING AND CURRENT ROW)

AS "Obecny i poprzedni miesiąc"

FROM(

SELECT DISTINCT p.FirstName + ' ' + p.LastName AS "Imię i nazwisko", YEAR(soh.OrderDate) AS Rok, MONTH(soh.OrderDate) AS Miesiąc,

COUNT(soh.SalesOrderID) OVER (PARTITION BY p.FirstName + ' ' + p.LastName, YEAR(soh.OrderDate), MONTH(soh.OrderDate)) AS "W miesiącu",

COUNT(soh.SalesOrderID) OVER (PARTITION BY p.FirstName + ' ' + p.LastName, YEAR(soh.OrderDate)) AS "W roku",

COUNT(soh.SalesOrderID) OVER (PARTITION BY p.FirstName + ' ' + p.LastName, YEAR(soh.OrderDate) ORDER BY MONTH(soh.OrderDate)) AS "W roku narastająco"

FROM Person.Person p

JOIN HumanResources.Employee e ON e.BusinessEntityID = p.BusinessEntityID

JOIN Sales.SalesPerson sp ON sp.BusinessEntityID = e.BusinessEntityID

JOIN Sales.SalesOrderHeader soh ON soh.SalesPersonID = sp.BusinessEntityID

) P;

**4)**

SELECT DISTINCT Kategoria, SUM("Cena max") OVER (PARTITION BY Kategoria) Suma

FROM(

SELECT DISTINCT cat.Name AS Kategoria, subcat.Name AS Podkategoria,

MAX(p.ListPrice) OVER (PARTITION BY cat.Name, subcat.Name) AS "Cena max"

FROM Production.ProductCategory cat

JOIN Production.ProductSubcategory subcat ON subcat.ProductCategoryID = cat.ProductCategoryID

JOIN Production.Product p ON p.ProductSubcategoryID = subcat.ProductSubcategoryID

) P;

**5)**

**RANK:**

SELECT Klient, RANK() OVER(ORDER BY Liczba DESC) AS Pozycja

FROM (

SELECT DISTINCT p.FirstName + ' ' + p.LastName AS Klient, SUM(sod.OrderQty) OVER (PARTITION BY p.FirstName + ' ' + p.LastName) Liczba

FROM Person.Person p

JOIN Sales.Customer c ON c.PersonID = p.BusinessEntityID

JOIN Sales.SalesOrderHeader soh ON soh.CustomerID = c.CustomerID

JOIN Sales.SalesOrderDetail sod ON sod.SalesOrderID = soh.SalesOrderID

) P

ORDER BY 2;

**DENSE\_RANK:**

SELECT Klient, DENSE\_RANK() OVER(ORDER BY Liczba DESC) AS Pozycja

FROM (

SELECT DISTINCT p.FirstName + ' ' + p.LastName AS Klient, SUM(sod.OrderQty) OVER (PARTITION BY p.FirstName + ' ' + p.LastName) Liczba

FROM Person.Person p

JOIN Sales.Customer c ON c.PersonID = p.BusinessEntityID

JOIN Sales.SalesOrderHeader soh ON soh.CustomerID = c.CustomerID

JOIN Sales.SalesOrderDetail sod ON sod.SalesOrderID = soh.SalesOrderID

) P

ORDER BY 2;

**6)**

SELECT Produkt, Tile AS Grupa, RANK() OVER(PARTITION BY Tile ORDER BY Srednia DESC) Pozycja, "Śreania liczba sztuk"

FROM (

SELECT Produkt, NTILE(3) OVER(ORDER BY Srednia DESC) AS Tile, Srednia, SUM(Srednia) OVER (PARTITION BY Produkt) "Śreania liczba sztuk"

FROM(

SELECT DISTINCT p.Name Produkt, AVG(sod.OrderQty) OVER (PARTITION BY p.Name) AS Srednia

FROM Production.Product p

JOIN Sales.SalesOrderDetail sod ON sod.ProductID = p.ProductID

) A) b;

**Wnioski:**

**Ad.1**

**1)**

**Obraz zawierający stół

Opis wygenerowany automatycznie**

Z powyższego zrzutu ekranu możemy wyraźnie stwierdzić, że największa sprzedaż miała miejsce w 2013 roku. Trzeba oczywiście wziąć pod uwagę, że dane z 2014 roku są niekompletne.

**2)**

**Obraz zawierający stół

Opis wygenerowany automatycznie**

Możemy zauważyć, że największe sumy udzielonych zniżek wystąpiły w 2013 roku – wynika to z faktu najwyższej sprzedaży właśnie w tym roku.

**Ad.2**

**Obraz zawierający stół

Opis wygenerowany automatycznieObraz zawierający stół

Opis wygenerowany automatycznieObraz zawierający stół

Opis wygenerowany automatycznie1)**

**Obraz zawierający stół

Opis wygenerowany automatycznie**

Najwyższy udział w sprzedaży wyglądał tak samo w przypadku każdej kategorii – najwyższy wynik w roku 2013, najniższy w 2011, następnie w 2014 i 2012 (jedyna różnica jest widoczna w przypadku akcesoriów – udział w roku 2014 był większy niż w roku 2011 i 2012).

**2)**

**Obraz zawierający stół

Opis wygenerowany automatycznie**

Tutaj także widzimy, że większa część zakupionych przez klienta produktów była zakupiona w 2013 roku. Widzimy także, że klienci, którzy kupowali najwięcej, kupowali głownie w latach 2013 oraz 2014.

**3)**

**Obraz zawierający stół

Opis wygenerowany automatycznie**

Widzimy, że pracownicy wykazywali największą sprzedaż w roku 2013. Dla powyższego przykładu można także powiedzieć, że pracownik sprzedał w 2013 roku większość produktów z 2013 w drugiej połowie roku.

**4)**

**Obraz zawierający stół

Opis wygenerowany automatycznie**

Suma maksymalnych cen w poszczególnych kategoriach pokazuje nam, że najwyższe ceny mają produkty należące do kategorii ‘Bikes’, następnie ‘Components’, ‘Accessories’, a na końcu ‘Clothing’.

**5)**

**Obraz zawierający stół

Opis wygenerowany automatycznieObraz zawierający stół

Opis wygenerowany automatycznie**

Na powyższych zrzutach ekranu możemy zobaczyć różnicę w funkcjach RANK oraz DENSE\_RANK – brak ciągłości dla funkcji RANK w przeciwieństwie do DENSE\_RANK.

**6)**

Obraz zawierający stół

Opis wygenerowany automatycznie

Rękawiczki w rozmiarze L są kupowane średnio w największych ilościach w poszczególnym zamówieniu przez klientów.

**Zadanie na zajęciach:**

SELECT DISTINCT e.Gender, DATEPART(q, soh.OrderDate) Kwartal, MONTH(soh.OrderDate) Miesiąc,

SUM(sod.OrderQty) OVER (PARTITION BY e.Gender, DATEPART(q, soh.OrderDate)ORDER BY MONTH(soh.OrderDate)) AS Suma,

SUM(sod.OrderQty) OVER (PARTITION BY MONTH(soh.OrderDate), e.Gender)

FROM HumanResources.Employee e

JOIN Sales.SalesPerson sp ON sp.BusinessEntityID = e.BusinessEntityID

JOIN Sales.SalesOrderHeader soh ON soh.SalesPersonID = sp.BusinessEntityID

JOIN Sales.SalesOrderDetail sod ON sod.SalesOrderID = soh.SalesOrderID

ORDER BY 2, 3, 4;

**Wnioski:**

Obraz zawierający stół

Opis wygenerowany automatycznie

Widzimy, że mężczyźni kupują więcej produktów niż kobiety. Można także zauważyć, że najwyższa sprzedaż występuje w ciepłych miesiącach.