**Hurtownie danych – Spr. 2.**

PWr. WIZ, Data: 14-15.03.2022

|  |  |  |
| --- | --- | --- |
| Student | ------------------------------------------------------------- | Ocena |
| Indeks | 256305 |  |
| Imię | Grzegorz |
| Nazwisko | Dzikowski |

Zestaw składa się z 4 zadań. Jeżeli nie potrafisz rozwiązać zadania, to próbuj podać, chociaż częściowe rozwiązanie lub uzasadnienie przyczyny braku rozwiązania. Pamiętaj o podaniu nr. indeksu oraz imienia i nazwiska.

Baza danych: **AdventureWorks**

**Zad. 1.**

Źródło danych: SalesOrderHeader

Zdefiniować następujące zapytania:

* 1. Wyznaczyć lata, w których zostały zarejestrowane zamówienia w bazie danych
  2. Utworzyć wykaz zamówień złożonych w pierwszym roku rejestracji zamówień (Identyfikator, Rok, Kwota zamówienia)
  3. Utworzyć wykaz zamówień złożonych w maju w poszczególnych latach (Rok, mc, Identyfikator, Kwota zamówienia)

**1.1**

**Rozwiązanie:**

USE AdventureWorks2019;

SELECT YEAR(OrderDate) as Lata FROM Sales.SalesOrderHeader GROUP BY YEAR(OrderDate) ORDER BY Lata;

|  |
| --- |
| Lata |
| 2011 |
| 2012 |
| 2013 |
| 2014 |

Rekordów: 5/5

1.2. Zapytanie SQL + fragment wyniku (4 rekordy z ?)

**Rozwiązanie:**

USE AdventureWorks2019;

SELECT

SalesOrderID as Identyfikator,

YEAR(OrderDate) as Rok,

SubTotal as [Kwota zamówienia]

FROM Sales.SalesOrderHeader

WHERE YEAR(OrderDate) =

(

SELECT TOP(1) YEAR(OrderDate) as Lata

FROM Sales.SalesOrderHeader

GROUP BY YEAR(OrderDate)

ORDER BY Lata

)

|  |  |  |
| --- | --- | --- |
| Identyfikator | Rok | Kwota zamówienia |
| 43659 | 2011 | 20565,6206 |
| 43660 | 2011 | 1294,2529 |
| 43661 | 2011 | 32726,4786 |
| 43662 | 2011 | 28832,5289 |

Rek.: 4/1607

* 1. Zapytanie SQL + fragment wyniku (4 rekordy z ?)

**Rozwiązanie:**

USE AdventureWorks2019;

SELECT

YEAR(OrderDate) as Rok,

MONTH(OrderDate) as Miesąc,

SalesOrderID as Identyfikator,

SubTotal as [Kwota zamówienia]

FROM Sales.SalesOrderHeader

WHERE MONTH(OrderDate) = 5

ORDER BY Rok;

|  |  |  |  |
| --- | --- | --- | --- |
| Rok | Miesąc | Identyfikator | Kwota zamówienia |
| 2011 | 5 | 43659 | 20565,6206 |
| 2011 | 5 | 43660 | 1294,2529 |
| 2011 | 5 | 43661 | 32726,4786 |
| 2011 | 5 | 43662 | 28832,5289 |

Rek.: 4/3175

**Zad 2.**

2.1. Utworzyć wykaz klientów, którzy mają więcej niż 25 zamówień (wykorzystać CTE). Przykładowy wynik zapytania przedstawiony jest poniżej w tabeli 2.1.:

**Rozwiązanie:**

USE AdventureWorks2019;

WITH Sales\_WD (CustomerID, PersonID, SalesID, FullName) AS

(

SELECT Sales.SalesOrderHeader.CustomerID, Sales.Customer.PersonID, Sales.SalesOrderHeader.SalesOrderID, CONCAT\_WS(' ', Person.FirstName, Person.LastName) as [Imie i Nazwisko]

FROM ((Sales.SalesOrderHeader

JOIN Sales.Customer ON Sales.Customer.CustomerID = Sales.SalesOrderHeader.CustomerID)

JOIN Person.Person ON Person.Person.BusinessEntityID = Sales.Customer.PersonID)

)

SELECT

CustomerID AS KlientID,

FullName AS [Imie i nazwisko],

COUNT(SalesID) AS [Liczba zamówień]

FROM Sales\_WD

GROUP BY CustomerID, FullName

HAVING COUNT(SalesID) > 25

ORDER BY KlientID;

|  |  |  |
| --- | --- | --- |
| KlientID | Imie i nazwisko | Liczba zamówień |
| 11091 | Dalton Perez | 28 |
| 11176 | Mason Roberts | 28 |
| 11185 | Ashley Henderson | 27 |
| 11200 | Jason Griffin | 27 |

Rek: 4/13

2.2. Ustalić, jakie czynniki mają wpływ na liczbę dokonanych zakupów. Przykładowy wynik zapytania przedstawiony jest poniżej w tabeli 2.2.

Źródło danych: SalesOrderHeaderSalesReason, ?

**Rozwiązanie:**

**USE AdventureWorks2019;**

WITH SalesReason (OrderID, SalesReason) AS (

SELECT Sales.SalesOrderHeader.SalesOrderID, Sales.SalesReason.[Name]

FROM Sales.SalesOrderHeader

RIGHT JOIN Sales.SalesOrderHeaderSalesReason ON Sales.SalesOrderHeaderSalesReason.SalesOrderID = Sales.SalesOrderHeader.SalesOrderID

RIGHT JOIN Sales.SalesReason ON SALES.SalesOrderHeaderSalesReason.SalesReasonID = Sales.SalesReason.SalesReasonID)

SELECT

SalesReason AS [Czynnik],

COUNT(OrderID) AS Dotyczy

FROM SalesReason

GROUP BY SalesReason

ORDER BY Dotyczy DESC

|  |  |
| --- | --- |
| Czynnik | Dotyczy |
| Price | 17473 |
| On Promotion | 3515 |
| Manufacturer | 1746 |
| Quality | 1551 |

Rek.: 4/10

**TABLEU**

Chart, bar chart

Description automatically generated

**Zad. 3.**

Zdefiniować zapytanie wyznaczające sprzedaż zrealizowaną przez pracowników poszczególnym klientom w latach zarejestrowanych w bazie danych.

**Rozwiązanie:**

* 1. Zapytanie SQL + fragment wyniku (4 rekordy z ?)

USE AdventureWorks2019;

SELECT

SalesPersonID,

CustomerID,

IIF([2011] is NULL, 'brak', str([2011])) as [2011],

IIF([2012] is NULL, 'brak', str([2012])) as [2012],

IIF([2013] is NULL, 'brak', str([2013])) as [2013],

IIF([2014] is NULL, 'brak', str([2014])) as [2014]

FROM

(

SELECT CustomerID, SalesPersonID, SubTotal, YEAR(OrderDate) AS Rok

FROM Sales.SalesOrderHeader

WHERE SalesPersonID is not NULL

) AS src

PIVOT (

SUM(SubTotal)

FOR ROK IN ([2011], [2012], [2013], [2014])

) AS PivotResult;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SalesPersonID | CustomerID | 2011 | 2012 | 2013 | 2014 |
| 274 | 29491 | brak | 33407 | brak | brak |
| 274 | 29493 | 2147 | brak | brak | brak |
| 274 | 29514 | brak | brak | 3405 | brak |
| 274 | 29523 | brak | brak | brak | 34349 |

Rek.: 4/860

* 1. Tableau – ten sam rezultat w formie graficznej

Obraz zawierający tekst, niebo, przybór do pisania, stacjonarne

Opis wygenerowany automatycznie

Fragment ~40 z 860 rekordów

**Zad. 4.**

Należy utworzyć tabelę przestawną prezentującą:

1. Średnią roczną kwotę zakupów dokonanych przez klientów w latach 2013-2014 wykorzystując operator PIVOT
2. Średnią roczną kwotę zakupów dokonanych przez klientów w latach 2013-2014 bez operatora PIVOT

**Rozwiązanie:**

* 1. Zapytanie SQL + fragment wyniku (4 rekordy z ?)

USE AdventureWorks2019;

WITH CustomerTable([Name], CustomerID, YearDate, SubTotal) AS (

SELECT

CONCAT\_WS(', ', Person.Person.LastName, Person.Person.FirstName) AS CustomerName,

Sales.SalesOrderHeader.CustomerID,

YEAR(OrderDate),

SubTotal

FROM Sales.SalesOrderHeader

JOIN Sales.Customer ON Sales.Customer.CustomerID = Sales.SalesOrderHeader.CustomerID

JOIN Person.Person ON Person.Person.BusinessEntityID = Sales.Customer.PersonID

)

SELECT \* FROM

(

SELECT \* FROM CustomerTable

) AS CustTable

PIVOT(

AVG(SubTotal)

FOR YearDate in ([2013], [2014])

) AS pvt\_source;

|  |  |  |  |
| --- | --- | --- | --- |
| Name | CustomerID | 2013 | 2014 |
| Abel, Catherine | 29485 | 28773,4492 | 27670,884 |
| Abercrombie, Kim | 29486 | 37776,8087 | 26765,9613 |
| Acevedo, Humberto | 29487 | 2461,7418 | 465,15 |
| Achong, Gustavo | 29484 | 30937,9131 | NULL |

Rek: 4/19119

1. Zapytanie SQL + fragment wyniku (4 rekordy z ?)

USE AdventureWorks2019;

WITH AVG\_YEAR ([Name], CustomerID, Rok, Average)

AS

(

SELECT CONCAT\_WS(', ', LastName, FirstName) as [Name], customer.CustomerID, YEAR([OrderDate]), AVG(SubTotal)

FROM Sales.Customer customer

JOIN Sales.SalesOrderHeader header ON customer.CustomerID=header.CustomerID

JOIN Person.Person person ON customer.PersonID=person.BusinessEntityID

GROUP BY customer.CustomerID, YEAR([OrderDate]), FirstName, LastName

)

SELECT [Name], CustomerID,

SUM(CASE

WHEN ROK = 2013 THEN Average

ELSE NULL

END) as [2013],

SUM(CASE

WHEN ROK = 2014 THEN Average

ELSE NULL

END) as [2014]

FROM AVG\_YEAR

GROUP BY [Name], CustomerID

ORDER BY [Name];

|  |  |  |  |
| --- | --- | --- | --- |
| Name | CustomerID | 2013 | 2014 |
| Abel, Catherine | 29485 | 28773,4492 | 27670,884 |
| Abercrombie, Kim | 29486 | 37776,8087 | 26765,9613 |
| Acevedo, Humberto | 29487 | 2461,7418 | 465,15 |
| Achong, Gustavo | 29484 | 30937,9131 | NULL |

Rek: 4/19119

**Wnioski:**

Tablau pozwala w bardzo prosty sposób wizualizować dane w bazie danych, co ułatwia ich pokazywanie w np. raportach

Użycie funkcji Pivot w zadaniu 4 dużo ułatwia realizację prezentacji danych, które normalnie byłyby jako kolejne wpisy w bazie danych. Było to dla mnie coś nowego, ponieważ nigdy wcześniej z funkcji pivot nie korzystałem.

With… as ułatwia tworzenie kwerend, dzięki czemu zamiast ogromnych tasiemców można rozdzielić prezentację danych od ich pobierania. Dzięki temu łatwiej debugować i analizować kwerendy.

Podsumowując, kwerendami można w bardzo prostu i przejrzysty sposób generować raporty i wizualizować dane. A wspomagając się Tablau można w bardzo dobry sposób wizualizować te dane.

*Uwaga:*

* Sprawozdanie bez wniosków końcowych nie będzie sprawdzane i tym samym ocena jest negatywna!