

### Hurtownie danych – Spr. 3.

PWr. WIZ, Data: 24-26.03.2021

Student		Ocena
Indeks	<u>251526</u>	
Imię	<u>Volodymyr</u>	
Nazwisko	<u>Zakhovaiko</u>	

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: **AdventureWorks2019**

#### **Zad. 1.**

Należy przygotować raport umożliwiający ocenę pracowników na podstawie sumarycznej sprzedaży zrealizowanej przez pracowników w poszczególnych latach.

Raport powinien zawierać dane zgodnie z tabelą 1. Raport należy przygotować bez użycia funkcji okienkowych (klauzula OVER)

Tab. 1. Wykaz pracowników oraz wypracowane przez nich kwoty w poszczególnych latach wraz z liczbą obsłużonych zamówień

Pracownik	pracID	Rok zamówienia	Kwota	Liczba zamówień
Jiang, Stephen	274	2011	28926.25	4
Jiang, Stephen	274	2012	453524.52	22
...	...	...	...	...
Blythe, Michael	275	2011	875823.83	65
Blythe, Michael	275	2012	3375456.89	148
...	...	...	...	...

#### **Rozwiązanie:**

```
SELECT MIN(PP.LastName + ', ' + PP.FirstName) Pracownik,
       SOH.SalesPersonID pracID,
       YEAR(SOH.OrderDate) "Rok zamówienia",
       SUM(SOH.SubTotal) Kwota,
       COUNT(SOH.SalesOrderID) "Liczba zamówien"
FROM Sales.SalesOrderHeader AS SOH
     JOIN Person.Person AS PP
     ON PP.BusinessEntityID = SOH.SalesPersonID
GROUP BY YEAR(SOH.OrderDate), SOH.SalesPersonID
ORDER BY pracID, "Rok zamówienia";
```

Pracownik	pracID	Rom zamówienia	Kwota	Liczba zamówien
Jiang, Stephen	274	2011	28926,2465	4
Jiang, Stephen	274	2012	453524,5233	22
Jiang, Stephen	274	2013	431088,7238	14
Jiang, Stephen	274	2014	178584,3625	8

Rek.: 4/58

#### **Zad 2.**

Należy przygotować raport umożliwiający ocenę pracowników na podstawie sumarycznej sprzedaży zrealizowanej przez pracowników w poszczególnych latach (raport powinien zawierać dane zgodnie z tabelą 1.). Raport należy przygotować wykorzystując klauzule OVER

### Rozwiązanie:

```
SELECT DISTINCT PP.LastName + ', ' + PP.FirstName Pracownik,
               SOH.SalesPersonID pracID,
               YEAR(SOH.OrderDate) "Rok zamówienia",
               SUM(SOH.SubTotal) OVER(PARTITION BY SOH.SalesPersonID, YEAR(SOH.OrderDate)) Kwota,
               COUNT(SOH.SalesOrderID) OVER(PARTITION BY SOH.SalesPersonID, YEAR(SOH.OrderDate))
               "Liczba zamówien"
FROM Sales.SalesOrderHeader AS SOH
     INNER JOIN Person.Person AS PP
     ON PP.BusinessEntityID = SOH.SalesPersonID
ORDER BY pracID, "Rok zamówienia";
```

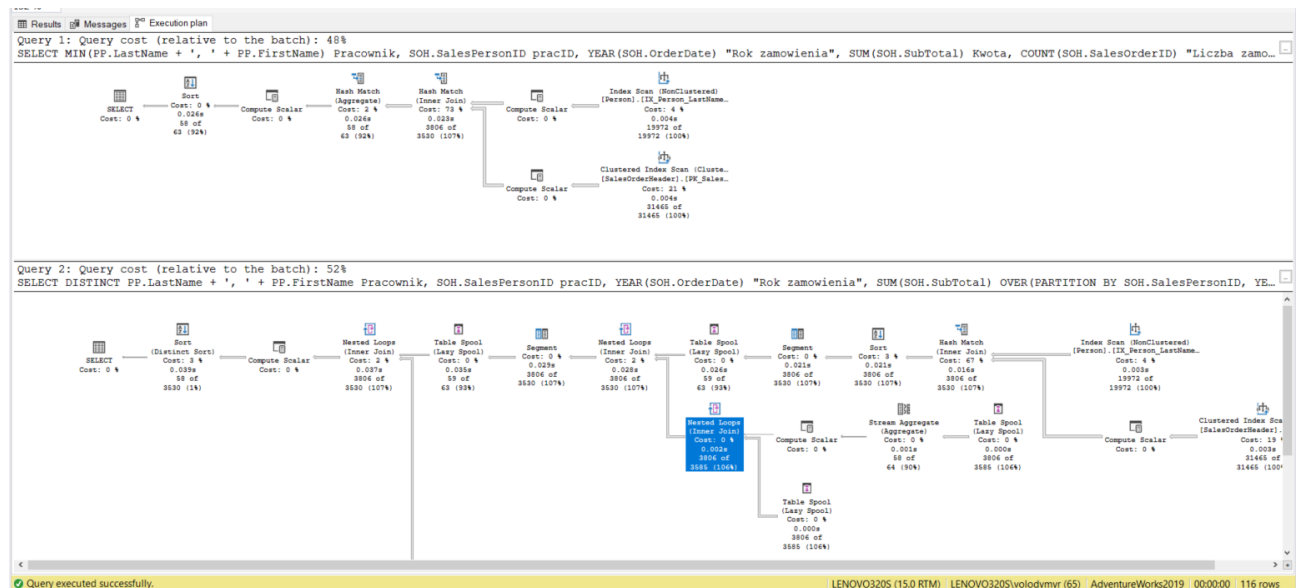
Pracownik	pracID	Rok zamówienia	Kwota	Liczba zamówien
Jiang, Stephen	274	2011	28926,2465	4
Jiang, Stephen	274	2012	453524,5233	22
Jiang, Stephen	274	2013	431088,7238	14
Jiang, Stephen	274	2014	178584,3625	8

Rek.: 4/58

### Zad 3.

Oceń jakość rozwiązań zaproponowanych w poprzednich dwóch zadaniach. Które z nich jest korzystniejsze i dlaczego, biorąc pod uwagę plany wykonania obu zapytań

### Rozwiązanie:



Jak widzimy z obrazku, przy użyciu klauzy OVER czas ządania i potrzebnosc resursów jest znacznie większa, niż przy użyciu GROUP BY. Natomiast można jeszcze bardziej skrócić czas ządania za pomocą CTE.

#### Zad. 4.

Źródło danych:

Baza danych: **AdventureWorks2019**

Narzędzia:

- **Microsoft SQL Server Management Studio, Tableau**

#### Etap 1.

Utworzyć bazę danych (o ile nie istnieje) o nazwie zgodnej z identyfikatorem studenta, a następnie utworzyć tabele (skrypt CREATE TABLE) zgodnie z poniższą specyfikacją:

- Dim\_Customer (CustomerID, FirstName, LastName, TerritoryName, CountryRegionCode, Group)

Tabele źródłowe:

- SalesTerritory
- Customer
- Person

#### Rozwiązanie:

```
CREATE DATABASE [251526];
```

```
CREATE TABLE Dim_Customer (  
    CustomerID INT NOT NULL CONSTRAINT PK_Dim_Customer PRIMARY KEY,  
    FirstName NVARCHAR(50) NOT NULL,  
    LastName NVARCHAR(50) NOT NULL,  
    TerritoryName NVARCHAR(50) NOT NULL,  
    CountryRegionCode NVARCHAR(3) NOT NULL,  
    "Group" NVARCHAR(50) NOT NULL  
);
```

- Dim\_Product (**ProductID**, Name, ListPrice, Color, SubCategoryName, CategoryName)

Tabele źródłowe:

- Production.Product
- Production.ProductSubcategory
- Production.ProductCategory

#### Rozwiązanie:

```
CREATE TABLE Dim_Product (  
    ProductID INT NOT NULL CONSTRAINT PK_Dim_Product PRIMARY KEY,  
    "Name" NVARCHAR(50) NOT NULL,  
    ListPrice MONEY NOT NULL,  
    Color NVARCHAR(15) NULL,  
    SubCategoryName NVARCHAR(50) NOT NULL,  
    CategoryName NVARCHAR(50) NOT NULL  
);
```

- Fact\_Orders (ProductID, CustomerID, OrderDate, ShipDate, OrderQty, UnitPrice, UnitPriceDiscount, LineTotal)

Tabele źródłowe:

- Sales.SalesOrderDetail
- Sales.SalesOrderHeader

**Rozwiązanie:**

```
CREATE TABLE Fact_Orders (  
    ProductID INT NOT NULL,  
    CustomerID INT NOT NULL,  
    OrderDate DATETIME NOT NULL,  
    ShipDate DATETIME NULL,  
    OrderQty SMALLINT NOT NULL,  
    UnitPrice MONEY NOT NULL,  
    UnitPriceDiscount MONEY NOT NULL,  
    LineTotal AS ISNULL(([UnitPrice]*((1.0)-[UnitPriceDiscount]))*[OrderQty],(0.0))  
);
```

2. Zdefiniować ograniczenia referencyjne dla tabeli Fact\_Orders

**Rozwiązanie:**

```
ALTER TABLE dbo.Fact_Orders  
ADD  
    CONSTRAINT FK_ProductID FOREIGN KEY (ProductID) REFERENCES Dim_Product(ProductID),  
    CONSTRAINT FK_CustomerID FOREIGN KEY (CustomerID) REFERENCES  
Dim_Customer(CustomerID);
```

3. Utworzone tabele napełnić danymi wykorzystując dane w podanych tabelach źródłowych

**Rozwiązanie:**

```
INSERT INTO [251526].dbo.Dim_Customer  
SELECT C.CustomerID,  
    PP.FirstName,  
    PP.LastName,  
    ST.[Name],  
    ST.[CountryRegionCode],  
    ST.[Group]  
FROM AdventureWorks2019.Sales.Customer AS C  
JOIN AdventureWorks2019.Person.Person AS PP  
ON PP.BusinessEntityID = C.PersonID  
JOIN AdventureWorks2019.Sales.SalesTerritory AS ST  
ON ST.TerritoryID = C.TerritoryID;
```

```
SELECT * FROM [251526].dbo.Dim_Customer;
```

CustomerID	FirstName	LastName	TerritoryName	CountryRegionCode	Group
11000	Jon	Yang	Australia	AU	Pacific
11001	Eugene	Huang	Australia	AU	Pacific
11002	Ruben	Torres	Australia	AU	Pacific
11003	Christy	Zhu	Australia	AU	Pacific

Rek.: 4/19119

```
INSERT INTO [251526].dbo.Dim_Product  
SELECT DISTINCT PP.ProductID,  
    PP.[Name],  
    PP.ListPrice,  
    PP.Color,
```

```

PSC.[Name],
PC.[Name]
FROM AdventureWorks2019.Production.Product AS PP
JOIN AdventureWorks2019.Production.ProductSubcategory AS PSC
ON PP.ProductSubcategoryID = PSC.ProductSubcategoryID
JOIN AdventureWorks2019.Production.ProductCategory AS PC
ON PSC.ProductCategoryID = PC.ProductCategoryID;

SELECT * FROM [251526].dbo.Dim_Product;

```

ProductID	Name	ListPrice	Color	SubCategoryName	CategoryName
680	HL Road Frame - Black, 58	1431,50	Black	Road Frames	Components
706	HL Road Frame - Red, 58	1431,50	Red	Road Frames	Components
707	Sport-100 Helmet, Red	34,99	Red	Helmets	Accessories
708	Sport-100 Helmet, Black	34,99	Black	Helmets	Accessories

Rek.: 4/295

```

INSERT INTO [251526].dbo.Fact_Orders
SELECT SOD.ProductID,
      SOH.CustomerID,
      SOH.OrderDate,
      SOH.ShipDate,
      SOD.OrderQty,
      SOD.UnitPrice,
      SOD.UnitPriceDiscount
FROM AdventureWorks2019.Sales.SalesOrderHeader AS SOH
JOIN AdventureWorks2019.Sales.SalesOrderDetail AS SOD
ON SOH.SalesOrderID = SOD.SalesOrderID;

SELECT * FROM [251526].dbo.Fact_Orders;

```

ProductID	CustomerID	OrderDate	ShipDate	OrderQty	UnitPrice	UnitPriceDiscount	LineTotal
707	29582	2011-07-01 00:00:00.000	2011-07-08 00:00:00.000	1	20,1865	0,00	20.186500
707	29614	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	2	20,1865	0,00	40.373000
707	29491	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	2	20,1865	0,00	40.373000
707	29824	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	1	20,1865	0,00	20.186500

Rek.: 4/121317

4. Wygenerować raport z danych w utworzonej bazie danych zgodnie z podaną definicją:

**Zamówienia(„Nazwisko, imię”, Kategoria produktu, Nazwa produktu, Cena)**

Rozwiązanie przedstawić w formie tabelarycznej (definicja zapytania i kopia wyniku) oraz w formie wizualnej (2-3 diagramy przygotowane w Tableau wraz z analizą wyników i wnioskami).

Uwaga:

*Cena powinna być wyznaczona z uwzględnieniem ceny jednostkowej, zniżki oraz ilości*

**Rozwiązanie 1. (zapytanie SQL + fragment wyniku 4-5 rekordów):**

```
SELECT DC.LastName + ', ' + DC.FirstName AS "Nazwisko, imie",
       DP.CategoryName "Kategoria Produktu",
       DP.[Name] "Nazwa produktu",
       DP.ListPrice Cena
FROM [251526].dbo.Fact_Orders AS FO
JOIN [251526].dbo.Dim_Customer AS DC
ON DC.CustomerID = FO.CustomerID
JOIN [251526].dbo.Dim_Product AS DP
ON DP.ProductID = FO.ProductID;
```

Nazwisko, imie	Kategoria Produktu	Nazwa produktu	Cena
Bremer, Ted	Accessories	Sport-100 Helmet, Red	34,99
Calafato, Ryan	Accessories	Sport-100 Helmet, Red	34,99
Adams, Carla	Accessories	Sport-100 Helmet, Red	34,99
Heaney, Brenda	Accessories	Sport-100 Helmet, Red	34,99

Rek.: 4/121317

**Rozwiązanie 2. (prezentacja wyników analizy danych - Tableau):**

Tableau - Book1

File Data Worksheet Dashboard Story Analysis Map Format Server Window Help

Standard Show Me

Data Analytics Pages

Fact\_Orders+ (251526)

Search

Tables

- Dim\_Customer
  - Nazwisko, imie
  - Dim\_Customer (Count)
- Dim\_Product
  - CategoryName
  - Name
  - ListPrice
  - Dim\_Product (Count)
- Fact\_Orders
  - CustomerID
  - OrderDate
  - ProductID
  - ShipDate
  - LineTotal
  - OrderQty
  - UnitPrice
  - UnitPriceDiscount
  - Fact\_Orders (Count)
- Measure Names
- Measure Values

Columns

Rows

Nazwisko, imie CategoryName Name ListPrice

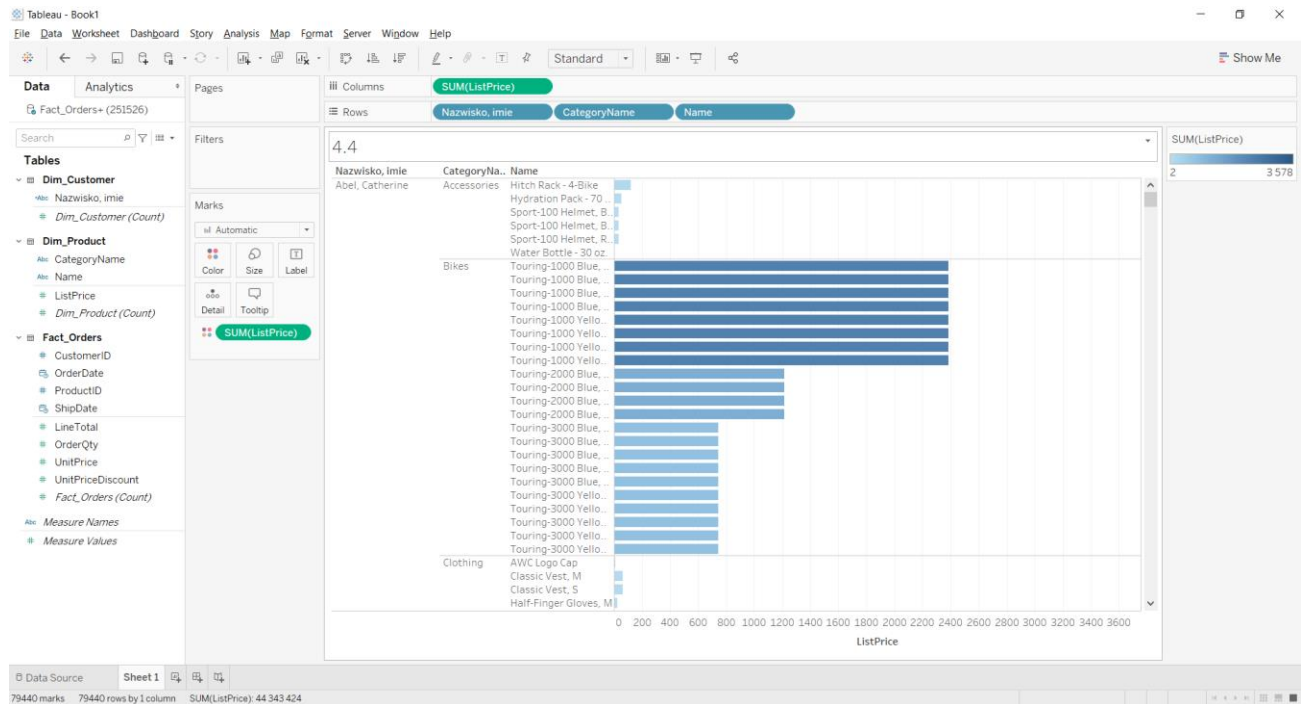
4,4

Nazwisko, imie	CategoryName	Name	ListPrice
Abel, Catherine	Accessories	Bike Wash - Dissolver	7.95
		Hitch Rack - 4-Bike	120
		Hydration Pack - 70 oz	54.99
		Sport-100 Helmet, Black	34.99
		Sport-100 Helmet, Blue	34.99
		Sport-100 Helmet, Red	34.99
		Water Bottle - 30 oz	4.99
	Bikes	Touring-1000 Blue, 46	2384.07
		Touring-1000 Blue, 50	2384.07
		Touring-1000 Blue, 54	2384.07
		Touring-1000 Blue, 60	2384.07
		Touring-1000 Yellow, 46	2384.07
		Touring-1000 Yellow, 50	2384.07
		Touring-1000 Yellow, 54	2384.07
		Touring-1000 Yellow, 60	2384.07
		Touring-2000 Blue, 46	1214.85
		Touring-2000 Blue, 50	1214.85
		Touring-2000 Blue, 54	1214.85
		Touring-2000 Blue, 60	1214.85
		Touring-3000 Blue, 44	742.35
		Touring-3000 Blue, 50	742.35
		Touring-3000 Blue, 54	742.35
		Touring-3000 Blue, 58	742.35
		Touring-3000 Blue, 62	742.35
		Touring-3000 Yellow, 44	742.35
		Touring-3000 Yellow, 50	742.35
		Touring-3000 Yellow, 54	742.35
		Touring-3000 Yellow, 58	742.35
		Touring-3000 Yellow, 62	742.35
	Clothing	AWC Logo Cap	8.99
		Classic Vest, M	63.5
		Classic Vest, S	63.5
		Half-Finger Gloves, M	24.49
		Half-Finger Gloves, S	24.49
		Long-Sleeve Logo Jersey, L	49.99

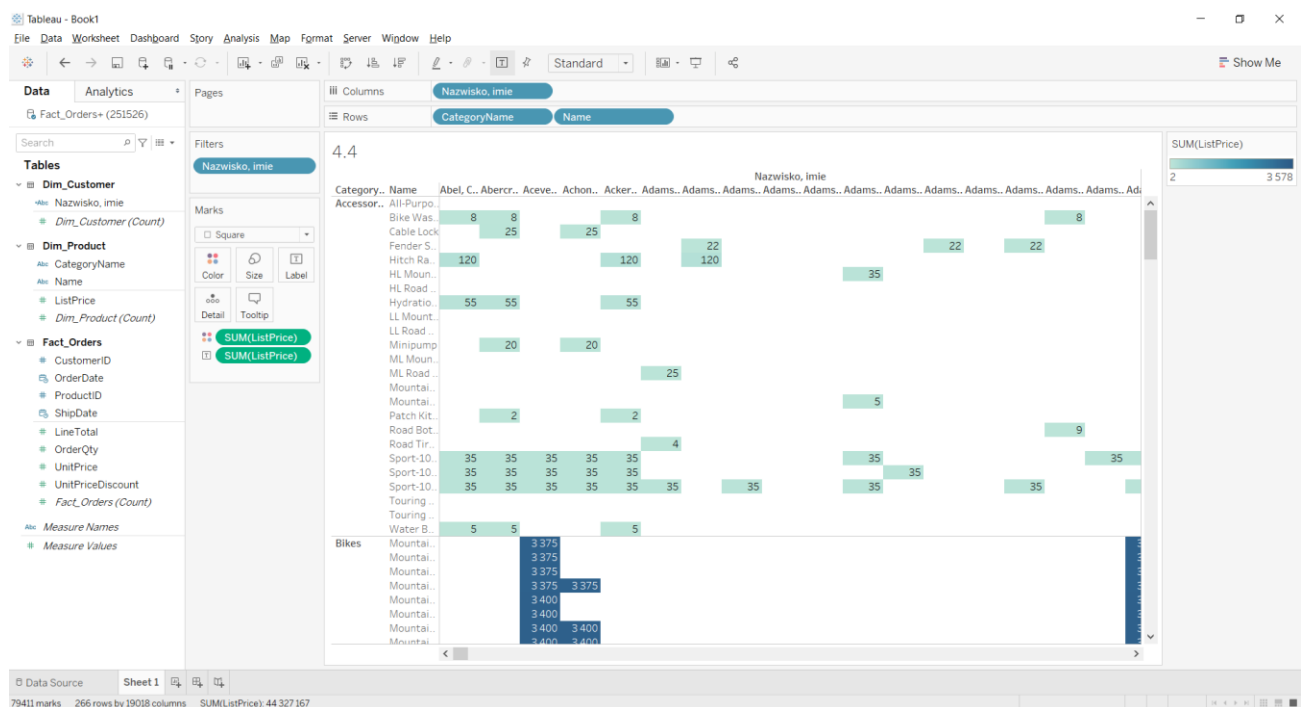
Sheet 1

79411 marks 79411 rows by 1 column

Wynik w formacie tabeli jest podobny do otrzymanego w SQL serwerze tylko zgrupowany dla łatwiejszej analizy danych.



W ten sposób jest łatwo przeanalizować popularne produkty kupione przez ludzi.



Jeszcze jeden dobry widok dla analizy, różny kolor klatek pozwala na szybszą analizę wielkiej ilości danych.

## Wnioski:

1. Klauzula OVER w SQL Server jest używana z PARTITION BY do dzielenia danych na partycje. Aby użyć klauzul OVER i PARTITION BY, wystarczy określić kolumnę, według której chcemy podzielić zagregowane wyniki.

2. Przy wypełnieniu tabeli Dim\_Customer JOIN musimy robić po polach BusinessEntityID oraz PersonID (a nie CustomerID), bo inaczej wyjdzie tak, że nie wszystkie osobniki zostały zapisane.
3. Program Tableau pozwala prowadzić analizy wielkich danych w bardzo wygodny i łatwy sposób bez żadnych problemów. Program ma różne diagramy do wyświetlenia żądań. Da on możliwość zwiększenie ilości przeanalizowanych danych niż zwykły przegląd wyników SQL serwera.

*Uwaga:*

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