

## Hurtownie danych – Spr. 2.

PWr. WIZ, Data: 26.03.2025

Student	-----	Ocena
Indeks	<u>272677</u>	
Imię	<u>Michał</u>	
Nazwisko	<u>Dudniczenko</u>	

Zestaw składa się z 5 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:

Baza danych: **AdventureWorks**

Narzędzia:

- **Microsoft SQL Server Management Studio, Tableau lub Power BI1.**

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:

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

Tabele źródłowe:

- SalesTerritory
- Customer
- Person

**Rozwiązanie:**

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

Tabele źródłowe:

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

**Rozwiązanie:**

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

Tabele źródłowe:

- Sales.SalesOrderDetail
- Sales.SalesOrderHeader

Rozwiązanie:

```
CREATE TABLE DimCustomer (  
    CustomerID INT,  
    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,  
  
    CONSTRAINT pk_dimCustomer PRIMARY KEY (CustomerID),  
);  
  
CREATE TABLE DimProduct (  
    ProductID INT,  
    [Name] NVARCHAR(50) NOT NULL,  
    ListPrice MONEY NOT NULL,  
    Color NVARCHAR(15),  
    SubCategoryName NVARCHAR(50) NOT NULL,  
    CategoryName NVARCHAR(50) NOT NULL,  
  
    CONSTRAINT pk_dimProduct PRIMARY KEY (ProductID),  
);  
  
CREATE TABLE FactOrders (  
    ProductID INT NOT NULL,  
    CustomerID INT,  
    OrderDate DATETIME,  
    ShipDate DATETIME,  
    OrderQty SMALLINT,  
    UnitPrice MONEY,  
    UnitPriceDiscount MONEY,  
    LineTotal DECIMAL(38, 6),  
  
    CONSTRAINT fk_orders_product FOREIGN KEY (ProductID) REFERENCES DimProduct(ProductID),  
    CONSTRAINT fk_orders_customer FOREIGN KEY (CustomerID) REFERENCES DimCustomer(CustomerID),  
);
```

2. Zdefiniować ograniczenia referencyjne dla tabeli Fact\_Orders

**Rozwiązanie:**

```
CREATE TABLE FactOrders (
    ProductID INT NOT NULL,
    CustomerID INT,
    OrderDate DATETIME,
    ShipDate DATETIME,
    OrderQty SMALLINT,
    UnitPrice MONEY,
    UnitPriceDiscount MONEY,
    LineTotal DECIMAL(38, 6),

    CONSTRAINT fk_orders_product FOREIGN KEY (ProductID) REFERENCES DimProduct(ProductID),
    CONSTRAINT fk_orders_customer FOREIGN KEY (CustomerID) REFERENCES DimCustomer(CustomerID),
);
```

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

**Rozwiązanie:**

```
INSERT INTO DimCustomer (CustomerID, FirstName, LastName, TerritoryName, CountryRegionCode, [Group])
SELECT
    c.CustomerID,
    p.FirstName,
    p.LastName,
    t.[Name],
    t.CountryRegionCode,
    t.[Group]
FROM AdventureWorks2022.Sales.Customer c
JOIN AdventureWorks2022.Person.Person p ON c.PersonID = p.BusinessEntityID
JOIN AdventureWorks2022.Sales.SalesTerritory t ON c.TerritoryID = t.TerritoryID;
```

```
INSERT INTO DimProduct (ProductID, [Name], ListPrice, Color, SubCategoryName, CategoryName)
SELECT
    p.ProductID,
    p.[Name],
    p.ListPrice,
    p.Color,
    ps.[Name],
    pc.[Name]
FROM AdventureWorks2022.Production.[Product] p
JOIN AdventureWorks2022.Production.ProductSubcategory ps ON p.ProductSubcategoryID = ps.ProductSubcategoryID
JOIN AdventureWorks2022.Production.ProductCategory pc ON ps.ProductCategoryID = pc.ProductCategoryID;
```

```
INSERT INTO FactOrders (ProductID, CustomerID, OrderDate, ShipDate, OrderQty, UnitPrice, UnitPriceDiscount, LineTotal)
SELECT
    d.ProductID,
    h.CustomerID,
    h.OrderDate,
    h.ShipDate,
    d.OrderQty,
    d.UnitPrice,
    d.UnitPriceDiscount,
    d.LineTotal
FROM AdventureWorks2022.Sales.SalesOrderDetail d
JOIN AdventureWorks2022.Sales.SalesOrderHeader h ON d.SalesOrderID = h.SalesOrderID;
```

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 wartościowe diagramy w kontekście planowanych analiz, przygotowane w Tableau wraz z analizą wyników i wnioskami).

**Rozwiązanie 1.4.1. (zapytanie SQL + fragment wyniku 4-5 rekordów):**

```
SELECT
    CONCAT(c.LastName, ', ', c.FirstName) AS [Nazwisko, imię],
    p.CategoryName AS [Kategoria produktu],
    p.[Name] AS [Nazwa produktu],
    o.UnitPrice AS [Cena]
FROM FactOrders o
JOIN DimProduct p ON o.ProductID = p.ProductID
JOIN DimCustomer c ON o.CustomerID = c.CustomerID;
```

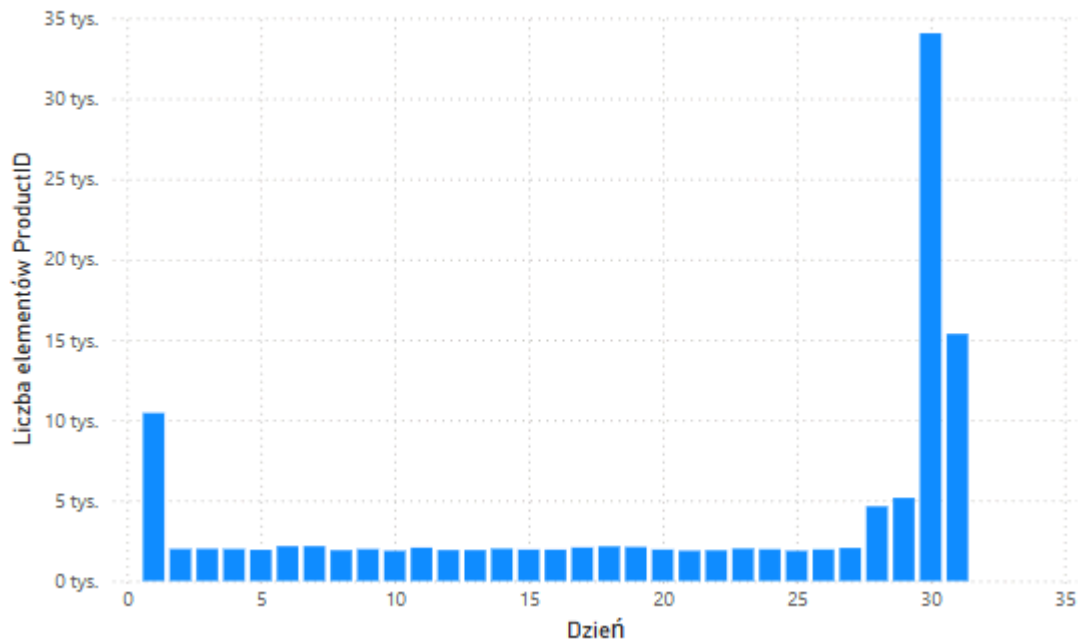
Fragment wyniku zapytania:

Nazwisko, imię	Kategoria produktu	Nazwa produktu	Cena
Bremer, Ted	Accessories	Sport-100 Helmet, Red	20,1865
Calafato, Ryan	Accessories	Sport-100 Helmet, Red	20,1865
Adams, Carla	Accessories	Sport-100 Helmet, Red	20,1865
Heaney, Brenda	Accessories	Sport-100 Helmet, Red	20,1865
Hirota, Nancy	Accessories	Sport-100 Helmet, Red	20,1865

Rek.: 5/121317

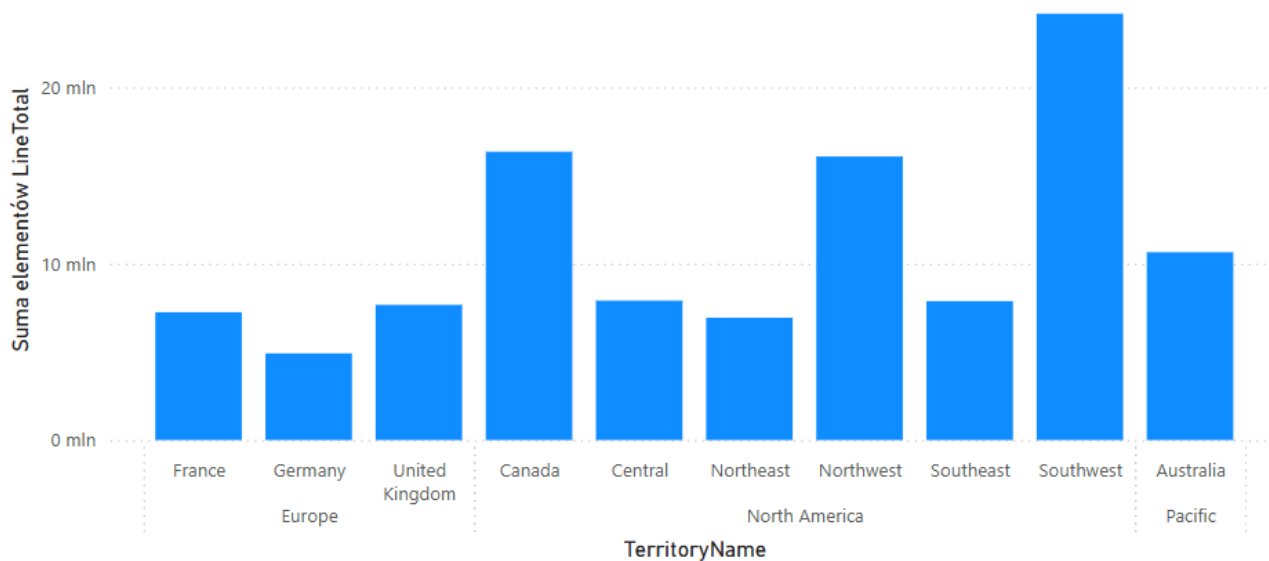
#### Rozwiązanie 1.4.2. (prezentacja wyników analizy danych – Tableau lub Power BI):

Liczba elementów ProductID wg Dzień



**Analiza wyników i wnioski:** zdecydowana większość klientów dokonuje zamówień w ostatnich dniach miesiąca. Może to być wartościowa informacja, w celu dostosowania strategii marketingowych, np. więcej reklam pod koniec miesiąca oraz w celu zarezerwowania większej mocy obliczeniowej serwerów na ten okres wyjątkowego obciążenia.

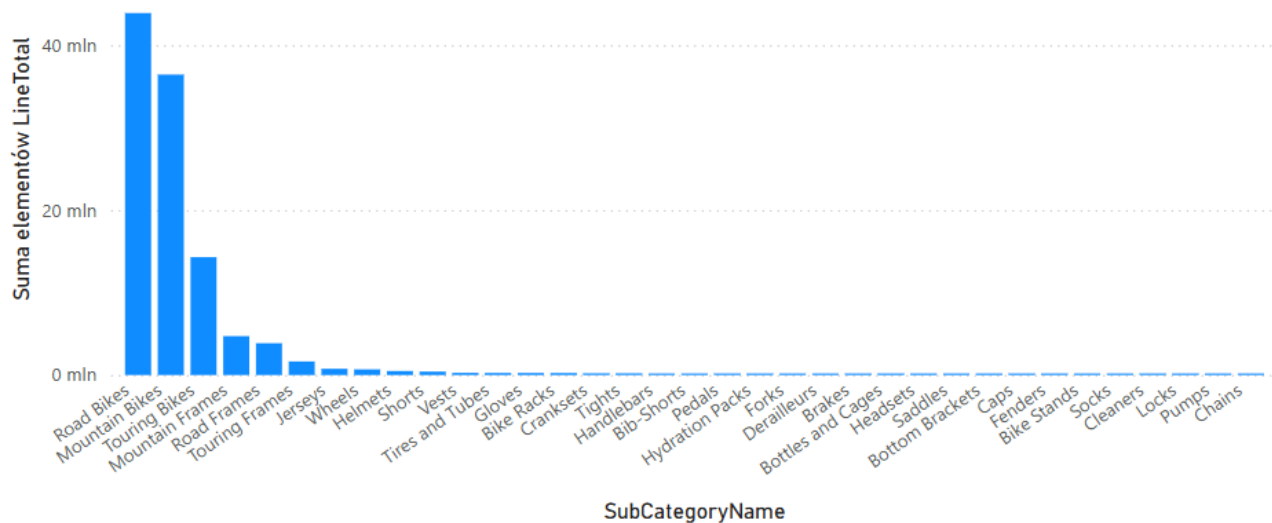
Suma elementów LineTotal wg Group i TerritoryName



**Analiza wyników i wnioski:** większość przychodu pochodzi od klientów z południowo-zachodniej części USA, Kanady, Australii i innych części USA. Również w UK i we Francji mamy duży przychód, możemy z tego powodu rozważyć agresywną ekspansję w tych krajach. Możemy również tak umiejscowić logistykę (magazyny, wysyłki itp.), żeby

klienci z regionów o największych przychodach czekali jak najkrócej na realizację zamówień.

Suma elementów LineTotal wg SubCategoryName



**Analiza wyników i wnioski:** większość przychodu pochodzi ze sprzedaży rowerów szosowych i górskich. Ta informacja może pomóc przy doborze strategii marketingowej.

## Zad. 2.

Proszę wyznaczyć klientów, którzy mieli największą liczbę zamówień w poszczególnych latach i przekroczyli roczną kwotę zamówień 130 000 brutto. Przykładowy wynik zapytania przedstawiony jest poniżej w tabeli 1.

Tabela 1. Fragment wyników zapytania zad. 2.

Rok	Klient Id	Nazwisko	Liczba zam.	Kwota
2011	29580	Bready, Richard	3	151771.60
2011	29898	Katyal, Sandeep	3	138317.84
2011	29614	Calafato, Ryan	3	134394.47
2012	29716	Dockter, Blaine	5	459031.55
2012	30117	Vessa, Robert	5	422961.29
2012	30074	Tian, Yuping	5	375760.05

Rek.: 6/?

## Rozwiązanie:

```
WITH PelnePodsumowanie AS (
SELECT
    YEAR(oh.OrderDate) AS [Rok],
    oh.CustomerID AS [Klient Id],
    CONCAT(p.LastName, ' ', p.FirstName) AS [Nazwisko],
    COUNT(oh.SalesOrderID) AS [Liczba zam.],
    SUM(oh.TotalDue) AS [Kwota],
    DENSE_RANK() OVER (PARTITION BY YEAR(oh.OrderDate) ORDER BY COUNT(oh.SalesOrderID) DESC) AS [Ranking]
FROM Sales.SalesOrderHeader oh
JOIN Sales.Customer c ON oh.CustomerID = c.CustomerID
LEFT JOIN Person.Person p ON p.BusinessEntityID = c.PersonID
GROUP BY YEAR(oh.OrderDate), oh.CustomerID, CONCAT(p.LastName, ' ', p.FirstName)
HAVING SUM(oh.TotalDue) > 130000
)
SELECT [Rok], [Klient Id], [Nazwisko], [Liczba zam.], ROUND([Kwota], 2)
FROM PelnePodsumowanie
WHERE Ranking = 1
ORDER BY [Rok] ASC, [Liczba zam.] DESC, [Kwota] DESC;
```

Fragment wyników zapytania:

Rok	Klient Id	Nazwisko	Liczba zam.	Kwota
2011	29580	Bready, Richard	3	151771,60
2011	29898	Katyal, Sandeep	3	138317,84
2011	29614	Calafato, Ryan	3	134394,47
2012	29716	Dockter, Blaine	5	459031,55

Rek.: 4/128

### Zad. 3.

- a) Proszę wyznaczyć liczbę dni pomiędzy pierwszym i ostatnim złożonym przez klientów zamówieniem (tab. 2.)

Tabela 2. Liczba dni pomiędzy pierwszym i ostatnim zamówieniem

Pierwsze	Ostatnie	Liczba dni
2011-05-31	2014-06-30	1126

Rek.:1/?

```
SELECT
    CAST(MIN(OrderDate) AS DATE) AS [Pierwsze],
    CAST(MAX(OrderDate) AS DATE) AS [Ostatnie],
    DATEDIFF(DAY, MIN(OrderDate), MAX(OrderDate)) AS [Liczba dni]
FROM Sales.SalesOrderHeader;
```

Wynik zapytania:

Pierwsze	Ostatnie	Liczba dni
2011-05-31	2014-06-30	1126

Rek.: 1/1



- b) Proszę wyznaczyć dla poszczególnych klientów liczbę zamówień oraz dni pomiędzy pierwszym i ostatnim złożonym przez nich zamówieniem (tab. 3.)

**Tabela 3. Liczba dni pomiędzy pierwszym i ostatnim zamówieniem klientów wraz z ich liczbą złożonych zamówień**

Klient	Od	Do	Liczba dni	Liczba zam.
11725	2013-09-01	2013-10-02	31	2
13798	2014-05-07	2014-06-07	31	2
14061	2013-07-30	2013-08-30	31	2
16937	2013-05-18	2013-06-18	31	2

Rek.: 4/?

**Rozwiązanie:**

```
SELECT
    CustomerID AS [Klient],
    CAST(MIN(OrderDate) AS DATE) AS [Od],
    CAST(MAX(OrderDate) AS DATE) AS [Do],
    DATEDIFF(DAY, MIN(OrderDate), MAX(OrderDate)) AS [Liczba dni],
    COUNT(SalesOrderID) AS [Liczba zam.]
FROM Sales.SalesOrderHeader
GROUP BY CustomerID
ORDER BY [Klient];
```

Fragment wyników zapytania:

Klient	Od	Do	Liczba dni	Liczba zam.
11000	2011-06-21	2013-10-03	835	3
11001	2011-06-17	2014-05-12	1060	3
11002	2011-06-09	2013-07-26	778	3
11003	2011-05-31	2013-10-10	863	3

Rek.: 4/19119

---

**Zad. 4.**

Proszę wyznaczyć sumaryczną liczbę zamówień obsługanych przez pracowników w kolejnych latach (licząc od początku działalności). Fragment wyniku zawiera tabela 4.

**Tabela 4. Sumaryczną liczbę zamówień obsługanych przez pracowników**

Pracownik	Rok	Liczba	Razem
274	2011	4	4
274	2012	22	26
274	2013	14	40
274	2014	8	48
275	2011	65	65
275	2012	148	213

Rek.: 6/?

**Rozwiązanie:**

```
SELECT
    SalesPersonID AS [Pracownik],
    YEAR(OrderDate) AS [Rok],
    COUNT(*) AS [Liczba],
    SUM(COUNT(*)) OVER (PARTITION BY SalesPersonID ORDER BY YEAR(OrderDate) ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS [Razem]
FROM Sales.SalesOrderHeader
WHERE SalesPersonID IS NOT NULL
GROUP BY SalesPersonID, YEAR(OrderDate)
ORDER BY [Pracownik] ASC, [Rok] ASC;
```

Fragment wyników zapytania:

Pracownik	Rok	Liczba	Razem
274	2011	4	4
274	2012	22	26
274	2013	14	40
274	2014	8	48

Rek.: 4/58

---

**Zad. 5.**

Proszę wyznaczyć liczbę oraz kwotę zamówionych, w poszczególnych latach, kategorii produktów. Fragment wyniku przedstawiono w tab. 5.

**Tabela 5. Liczba oraz kwota zamówionych kategorii produktów**

Year	Category	Kwota	Liczba zamówień
2011	Accessories	14836129.61	360
2012	Accessories	69183322.41	1339
2013	Accessories	128142537.66	20684
2014	Accessories	51924553.04	18811
2011	Bikes	94334765.91	3826
2012	Bikes	410662322.01	10776

**Rozwiązanie:**

```
SELECT
    YEAR(oh.OrderDate) AS [Year],
    pc.[Name] AS [Category],
    CAST(SUM(od.LineTotal) AS DECIMAL(20, 2)) AS [Kwota],
    COUNT(od.ProductID) AS [Liczba zamówień]
FROM Sales.SalesOrderDetail od
JOIN Production.[Product] p ON p.ProductID = od.ProductID
JOIN Sales.SalesOrderHeader oh ON oh.SalesOrderID = od.SalesOrderID
JOIN Production.ProductSubcategory ps ON ps.ProductSubcategoryID = p.ProductSubcategoryID
JOIN Production.ProductCategory pc ON pc.ProductCategoryID = ps.ProductCategoryID
GROUP BY YEAR(oh.OrderDate), pc.[Name]
ORDER BY [Category] ASC, [Year] ASC;
```

Fragment wyników zapytania:

Year	Category	Kwota	Liczba zamówień
2011	Accessories	20820.77	360
2012	Accessories	102439.82	1339
2013	Accessories	675024.65	20684
2014	Accessories	473787.64	18811

---

## Wnioski:

Udało mi się utworzyć bazę danych bez większych problemów, jedynie dalej poruszanie się po interfejsie SSMS zajmuje mi stosunkowo sporo czasu. Zacząłem korzystać z wbudowanego generatora diagramów dla tabel żeby czytelniej móc przeglądać powiązania między tabelami w bazie. Ustawienie kluczy obcych, wypełnienie danymi oraz wygenerowanie raportu z zapytania przebiegło bez problemów. Przy generowaniu wykresów w Power BI musiałem oswoić się z interfejsem aplikacji, ostatecznie tworzenie tych wizualizacji trochę przypomina mi korzystanie z tabel przestawnych w Excelu podczas przygotowań do matury z informatyki kilka lat temu. Zadania z zapytaniami przebiegły dość sprawnie, jedynie 2 zadania gdzie zdecydowałem się wykorzystać funkcje okienkowe zajęły mi sporo czasu, ponieważ był to chyba pierwszy raz kiedy ich używałem i szczerze dalej trochę nie do końca rozumiem jak działają, do prawidłowych wyników dotarłem trochę metodą prób i błędów. We wszystkich zadaniach otrzymane wyniki były dokładnie takie same jak podane w instrukcji, jedynie w jednym zadaniu kwoty dla kategorii, które otrzymałem były całkowicie różne od tych przykładowych. Nie wiem dlaczego, dość długo analizowałem kod mojego zapytania, strukturę tabel w bazie i wydaje mi się, że moje zapytanie jest prawidłowe i nie wiem w jaki sposób dojść do podanych wyników. Przy realizacji tej listy zadań nie korzystałem za dużo ze sztucznej inteligencji, jedynie pytania dotyczące specyfiki dialektu sql server np. "jak obliczyć liczbę dni między datami", "jak wyświetlić datetime jako yyyy-MM-dd" lub pytania o składnię np. funkcji okienkowych.

## Uwaga:

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