

# Hurtownie Danych – laboratorium lista 4

## Spis treści

Zadanie 1. Przygotowanie schematu.....	2
Zadanie 2. Tworzenie tabel wymiarów i tabeli faktów.....	2
Tabele wymiarów: .....	2
DIM_CUSTOMER .....	2
DIM_PRODUCT .....	3
DIM SALESPERSON .....	4
Tabela faktów: .....	4
FACT_SALES .....	4
Zadanie 3. Wypełnianie danych – denormalizacja źródłowej bazy .....	5
Tabele wymiarów: .....	5
DIM_CUSTOMER .....	5
DIM_PRODUCT .....	6
DIM SALESPERSON .....	7
Tabela faktów: .....	8
FACT_SALES .....	8
Zadanie 4. Więzy integralności .....	9
Zadanie 5. ....	11
DIM_PRODUCT .....	11
DIM_CUSTOMER .....	12
DIM SALESPERSON .....	12
Zadanie 6. Uruchomienie kostki .....	13
Zadanie 7. Proste raporty .....	14
Wnioski: .....	18

## Zadanie 1. Przygotowanie schematu.

W bazie danych należy utworzyć schemat, którego nazwa będzie odpowiadać nazwisku wykonującego ćwiczenie (zapisać zapytanie tworzące ten schemat).

```
CREATE SCHEMA Czech;
GO
```

## Zadanie 2. Tworzenie tabel wymiarów i tabeli faktów.

W nowo utworzonym schemacie utworzyć tabele wymiarów: klienta, produktu i sprzedawcy (zapisać skrypt CREATE TABLE), opisane w następujących schematach:

- DIM\_CUSTOMER (CustomerID, FirstName, LastName, Title, City, TerritoryName, CountryRegionCode, Group)
- DIM\_PRODUCT (ProductID, Name, ListPrice, Color, SubCategoryName, CategoryName, Weight, Size, IsPurchased)
- DIM SALESPERSON (SalesPersonID, FirstName, LastName, Title, Gender, CountryRegionCode, Group)

oraz tabelę faktów:

- FACT\_SALES (ProductID, CustomerID, SalesPersonID, OrderDate, ShipDate, OrderQty, UnitPrice, UnitPriceDiscount, LineTotal)

### Tabele wymiarów:

#### DIM\_CUSTOMER

```
-- Tworzenie tabel wymiarów
CREATE TABLE Czech.DIM_CUSTOMER (
    CustomerID INT PRIMARY KEY,
    FirstName NVARCHAR(50) NOT NULL,
    LastName NVARCHAR(50) NOT NULL,
    Title NVARCHAR(10),
    City NVARCHAR(50),
    TerritoryName NVARCHAR(50),
    CountryRegionCode NVARCHAR(10),
    [Group] NVARCHAR(50)
);
```

## Wyjaśnienia:

- Korzystamy z NVARCHAR zamiast VARCHAR aby poprawnie przechowywać znaki specjalne niezależnie od kodowania ASCII. Adventure Works prowadzi działalność w różnych regionach, dlatego jest to ważne.
- First i Last Name NOT NULL ponieważ wszyscy klienci którzy dokonali transakcji są przypisani do jakiejś osoby (Nie istnieje żaden zakup dokonany przez Customera z wartością NULL w PersonID). Oznacza to, że nawet zakupy firmowe są dokonywane przez fizycznych przedstawicieli.

```
SELECT COUNT(*) FROM Sales.SalesOrderHeader soh
JOIN Sales.Customer c ON soh.CustomerID = c.CustomerID
WHERE c.PersonID IS NULL;
```

	(No column name)
1	0

## DIM\_PRODUCT

```
CREATE TABLE Czech.DIM_PRODUCT (
    ProductID INT PRIMARY KEY,
    Name NVARCHAR(100) NOT NULL,
    ListPrice DECIMAL(10,2) NOT NULL CHECK (ListPrice >= 0),
    Color NVARCHAR(20),
    SubCategoryName NVARCHAR(50),
    CategoryName NVARCHAR(50),
    Weight DECIMAL(10,2) CHECK (Weight >= 0),
    Size NVARCHAR(10),
    IsPurchased BIT NOT NULL
);
```

## Wyjaśnienia:

- IsPurchased odpowiada za informację o tym, czy firma Adventure Works sama produkuje dany produkt. Czy kupuje go od firm trzecich.

## DIM\_SALESPERSON

```
CREATE TABLE Czech.DIM_SALESPERSON (
    SalesPersonID INT PRIMARY KEY,
    FirstName NVARCHAR(50) NOT NULL,
    LastName NVARCHAR(50) NOT NULL,
    Title NVARCHAR(10),
    Gender CHAR(1) CHECK (Gender IN ('M', 'F')),
    CountryRegionCode NVARCHAR(10),
    [Group] NVARCHAR(50)
);
```

Wyjaśnienia:

- Płeć sprzedawcy otrzymujemy z tabeli HumanResources.Employee

```
SELECT DISTINCT Gender FROM HumanResources.Employee;
```

	Gender
1	F
2	M

Tabela faktów:

## FACT\_SALES

```
-- Tworzenie tabeli faktów
CREATE TABLE Czech.FACT_SALES (
    ProductID INT NOT NULL,
    CustomerID INT,
    SalesPersonID INT,
    OrderDate INT NOT NULL CHECK (OrderDate BETWEEN 20000101 AND 21001231),
    ShipDate INT CHECK (ShipDate BETWEEN 20000101 AND 21001231),
    OrderQty INT NOT NULL CHECK (OrderQty > 0),
    UnitPrice DECIMAL(10,2) NOT NULL CHECK (UnitPrice >= 0),
    UnitPriceDiscount DECIMAL(5,2) DEFAULT 0 CHECK (UnitPriceDiscount >= 0),
    LineTotal DECIMAL(18,2) NOT NULL CHECK (LineTotal >= 0),
    CONSTRAINT FK_FACT_SALES_Product FOREIGN KEY (ProductID) REFERENCES Czech.DIM_PRODUCT(ProductID),
    CONSTRAINT FK_FACT_SALES_Customer FOREIGN KEY (CustomerID) REFERENCES Czech.DIM_CUSTOMER(CustomerID),
    CONSTRAINT FK_FACT_SALES_SalesPerson FOREIGN KEY (SalesPersonID) REFERENCES Czech.DIM_SALESPERSON(SalesPersonID)
);
```

Wyjaśnienia:

- CustomerID może być NULL -> w przypadku usunięcia klienta z DIM\_CUSTOMER, zamienimy wszystkie jego dane na NULL aby uniknąć tracenia rekordów.
- SalesPersonID może być NULL -> istnieją transakcje bez przypisanego sprzedawcy (prawdopodobnie transakcje internetowe).

OrderDate oraz ShipDate są zapisane w postaci liczby całkowitej w formacie yyyyMMdd. Daty muszą być z zakresu 2000.01.01 do 2100.12.31.

- UnitPrice, OrderQty oraz UnitPriceDiscount, LineTotal nie mogą być liczbami ujemnymi (OrderQty musi być ściśle dodatnie i całkowite).
- Klucze Obce:
  - ProductID odnosi się do ProductID z DIM\_PRODUCT
  - CustomerID odnosi się do CustomerID z DIM\_CUSTOMER
  - SalespersonID odnosi się do SalespersonID z DIM\_SALESPERSON

## Zadanie 3. Wypełnianie danych – denormalizacja źródłowej bazy

Wypełnić nowoutworzone tabele danymi znajdującymi się w tabelach źródłowych. Do wypełnienia użyć instrukcji INSERT INTO. Proszę sprawdzić liczbę skopiowanych rekordów.

### Tabele wymiarów:

#### DIM\_CUSTOMER

```
--Customer
WITH AddressPriority AS (
    SELECT
        bea.BusinessEntityID,
        a.City,
        ROW_NUMBER() OVER (
            PARTITION BY bea.BusinessEntityID
            ORDER BY
                CASE WHEN at.Name = 'Home' THEN 1 ELSE 2 END, -- Priorytet: Home → inny
                bea.AddressID -- Jeśli nie ma Home, wybierz najniższy AddressID
        ) AS rn
    FROM Person.BusinessEntityAddress bea
    JOIN Person.Address a ON bea.AddressID = a.AddressID
    JOIN Person.AddressType at ON bea.AddressTypeID = at.AddressTypeID
)
INSERT INTO Czech.DIM_CUSTOMER (CustomerID, FirstName, LastName, Title, City, TerritoryName, CountryRegionCode, [Group])
SELECT
    c.CustomerID,
    p.FirstName,
    p.LastName,
    p.Title,
    ap.City,
    st.Name AS TerritoryName,
    st.CountryRegionCode,
    st.[Group]
FROM Sales.Customer c
JOIN Person.Person p ON c.PersonID = p.BusinessEntityID
LEFT JOIN AddressPriority ap ON c.PersonID = ap.BusinessEntityID AND ap.rn = 1
LEFT JOIN Sales.SalesTerritory st ON c.TerritoryID = st.TerritoryID;
```

(19119 rows affected)

## Wyjaśnienia:

- Niektórzy klienci mają przypisane parę różnych adresów, dlatego posługujemy się CTE AddressPriority oraz funkcją ROW\_NUMBER, żeby dla każdego klienta wybrać adres domowy (jeśli posiada), w przeciwnym wypadku wybieramy dowolny adres.
- Korzystamy z LEFT JOIN z SalesTerritory oraz AddressPriority w przypadku gdyby któryś z klientów nie miał przypisanego adresu lub terytorium.

	CustomerID	FirstName	LastName	Title	City	TerritoryName	CountryRegionCode	Group
18...	29404	Chloe	Ross	NULL	Versailles	France	FR	Euro...
18...	29405	Meagan	Vance	NULL	Versailles	France	FR	Euro...
18...	29406	Melissa	Ward	NULL	Darmstadt	Germany	DE	Euro...
18...	29407	Samantha	Smith	Ms.	Cheltenham	United King...	GB	Euro...
18...	29408	Heather	Chen	NULL	Les Ulis	France	FR	Euro...
18...	29409	Kelsey	Pal	NULL	Hamburg	Germany	DE	Euro...
18...	29410	Lucas	Price	NULL	Hof	Germany	DE	Euro...
18...	29411	Cara	Lin	NULL	York	United King...	GB	Euro...
18...	29412	Melinda	Gutierrez	NULL	Chatou	France	FR	Euro...

## DIM\_PRODUCT

```

INSERT INTO Czech.DIM_PRODUCT (ProductID, Name, ListPrice, Color, SubCategoryName, CategoryName, Weight, Size, IsPurchased)
SELECT
    p.ProductID,
    p.Name,
    p.ListPrice,
    p.Color,
    psc.Name AS SubCategoryName,
    pc.Name AS CategoryName,
    p.Weight,
    p.Size,
    CASE WHEN p.MakeFlag = 0 THEN 1 ELSE 0 END AS IsPurchased
FROM Production.Product p
LEFT JOIN Production.ProductSubcategory psc ON p.ProductSubcategoryID = psc.ProductSubcategoryID
LEFT JOIN Production.ProductCategory pc ON psc.ProductCategoryID = pc.ProductCategoryID;

(504 rows affected)

```

## Wyjaśnienia:

- LEFT JOIN z ProductSubcategory oraz ProductCategory żeby uwzględnić produkty, które nie mają przypisanych kategorii i subkategorii.
- IsPurchased uzyskujemy za pomocą negacji informacji o tym, czy produkt jest wyrabiany przez firmę (p.MakeFlag).

	ProductID	Name	ListPrice	Color	SubCategoryName	CategoryName	Weight	Size	IsPurchased
203	530	Seat Post	0.00	NULL	NULL	NULL	NULL	NULL	1
204	531	Steerer	0.00	NULL	NULL	NULL	NULL	NULL	0
205	532	Seat Stays	0.00	NULL	NULL	NULL	NULL	NULL	0
206	533	Seat Tube	0.00	NULL	NULL	NULL	NULL	NULL	0
207	534	Top Tube	0.00	NULL	NULL	NULL	NULL	NULL	0
208	535	Tension Pulley	0.00	NULL	NULL	NULL	NULL	NULL	1
209	679	Rear Derailleur Cage	0.00	Silver	NULL	NULL	NULL	NULL	1
210	680	HL Road Frame - BL...	1431.50	Black	Road Frames	Components	2.24	58	0
211	706	HL Road Frame - R...	1431.50	Red	Road Frames	Components	2.24	58	0
212	707	Sport-100 Helmet, R...	34.99	Red	Helmets	Accessories	NULL	NULL	1
213	708	Sport-100 Helmet, B...	34.99	Black	Helmets	Accessories	NULL	NULL	1
214	709	Mountain Bike Sock...	9.50	White	Socks	Clothing	NULL	M	1
215	710	Mountain Bike Sock...	9.50	White	Socks	Clothing	NULL	L	1
216	711	Sport-100 Helmet, B...	34.99	Blue	Helmets	Accessories	NULL	NULL	1

## DIM\_SALESPERSON

```

INSERT INTO Czech.DIM_SALESPERSON (SalesPersonID, FirstName, LastName, Title, Gender, CountryRegionCode, [Group])
SELECT
    sp.BusinessEntityID AS SalesPersonID,
    p.FirstName,
    p.LastName,
    p.Title,
    e.Gender,
    st.CountryRegionCode,
    st.[Group]
FROM Sales.SalesPerson sp
JOIN Person.Person p ON sp.BusinessEntityID = p.BusinessEntityID
JOIN HumanResources.Employee e ON sp.BusinessEntityID = e.BusinessEntityID
LEFT JOIN Sales.SalesTerritory st ON sp.TerritoryID = st.TerritoryID;

```

(17 rows affected)

Wyjaśnienia:

- LEFT JOIN z SalesTerritory ponieważ istnieją klienci bez przypisanego terytorium sprzedaży.

	SalesPersonID	FirstName	LastName	Title	Gender	CountryRegionCode	Group
1	274	Stephen	Jiang	NULL	M	NULL	NULL
2	275	Michael	Blythe	NULL	M	US	North America
3	276	Linda	Mitchell	NULL	F	US	North America
4	277	Jillian	Carson	NULL	F	US	North America
5	278	Garrett	Vargas	NULL	M	CA	North America
6	279	Tsvi	Reiter	NULL	M	US	North America
7	280	Pamela	Ansman-Wolfe	NULL	F	US	North America

## Tabela faktów:

## FACT\_SALES

```

INSERT INTO Czech.FACT_SALES
(ProductID, CustomerID, SalesPersonID, OrderDate, ShipDate, OrderQty, UnitPrice, UnitPriceDiscount, LineTotal)
SELECT
    sod.ProductID,
    soh.CustomerID,
    soh.SalesPersonID,
    CAST(FORMAT(soh.OrderDate, 'yyyyMMdd') AS INT) AS OrderDate,
    CAST(FORMAT(soh.ShipDate, 'yyyyMMdd') AS INT) AS ShipDate,
    sod.OrderQty,
    sod.UnitPrice,
    sod.UnitPriceDiscount,
    sod.LineTotal
FROM Sales.SalesOrderDetail sod
JOIN Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID;

```

(121317 rows affected)

## Wyjaśnienia:

- Daty konwertujemy na format INT za pomocą CAST(FORMAT(Date, 'yyyyMMdd') AS INT)

	ProductID	CustomerID	SalesPersonID	OrderDate	ShipDate	OrderQty	UnitPrice	UnitPriceDiscount	LineTotal
1	749	21768	NULL	20110531	20110607	1	3578.27	0.00	3578.27
2	767	14501	NULL	20110531	20110607	1	699.10	0.00	699.10
3	773	25863	NULL	20110531	20110607	1	3399.99	0.00	3399.99
4	773	11003	NULL	20110531	20110607	1	3399.99	0.00	3399.99
5	773	28389	NULL	20110531	20110607	1	3399.99	0.00	3399.99
6	707	29844	275	20110531	20110607	4	20.19	0.00	80.75
7	712	29533	275	20110531	20110607	2	5.19	0.00	10.37
8	725	29718	275	20110531	20110607	1	183.94	0.00	183.94
9	773	29566	275	20110531	20110607	2	2039.99	0.00	4079.99
10	707	29491	275	20110531	20110607	2	20.19	0.00	40.37
11	712	29844	275	20110531	20110607	1	5.19	0.00	5.19
12	729	29718	275	20110531	20110607	2	183.94	0.00	367.88
13	772	29491	275	20110531	20110607	1	2039.99	0.00	2039.99
14	708	29491	275	20110531	20110607	2	20.19	0.00	40.37
15	715	29759	275	20110531	20110607	1	28.84	0.00	28.84



## Zadanie 4. Więzy integralności

- 1) Dodać integralność referencyjną i klucze główne do tabel już zdefiniowanych

```
ALTER TABLE Czech.FACT_SALES
ADD CONSTRAINT FK_FACT_SALES_Product
    FOREIGN KEY (ProductID) REFERENCES Czech.DIM_PRODUCT(ProductID) ON DELETE NO ACTION,
CONSTRAINT FK_FACT_SALES_Customer
    FOREIGN KEY (CustomerID) REFERENCES Czech.DIM_CUSTOMER(CustomerID) ON DELETE SET NULL, -- Usunięcie klienta ustawia NULL w sprzedaży
CONSTRAINT FK_FACT_SALES_SalesPerson
    FOREIGN KEY (SalesPersonID) REFERENCES Czech.DIM SALESPERSON(SalesPersonID) ON DELETE SET NULL; -- Usunięcie sprzedawcy ustawia NULL

ALTER TABLE Czech.DIM_PRODUCT
ADD CONSTRAINT PK_DIM_PRODUCT PRIMARY KEY (ProductID);

ALTER TABLE Czech.DIM_CUSTOMER
ADD CONSTRAINT PK_DIM_CUSTOMER PRIMARY KEY (CustomerID);

ALTER TABLE Czech.DIM SALESPERSON
ADD CONSTRAINT PK_DIM SALESPERSON PRIMARY KEY (SalesPersonID);
```

Dodajemy również indeksy, żeby usprawnić wydajność odczytów.

```
CREATE INDEX IDX_FACT_SALES_Product ON Czech.FACT_SALES(ProductID);
CREATE INDEX IDX_FACT_SALES_Customer ON Czech.FACT_SALES(CustomerID);
CREATE INDEX IDX_FACT_SALES_SalesPerson ON Czech.FACT_SALES(SalesPersonID);
```

Oraz ograniczenia zapewniające poprawność danych.

```
ALTER TABLE Czech.FACT_SALES
ADD CONSTRAINT CHK_FACT_SALES_ValidDates CHECK (ShipDate >= OrderDate);

ALTER TABLE Czech.DIM_PRODUCT
ADD CONSTRAINT CHK_DIM_PRODUCT_ValidPrice CHECK (ListPrice >= 0);
```

- 2) Przygotować instrukcję INSERT INTO, która sprawdzi poprawność integralności referencyjnej oraz klucze główne.

Poprawne dane:

```
-- Dodanie przykładowych danych do DIM_CUSTOMER
INSERT INTO Czech.DIM_CUSTOMER (CustomerID, FirstName, LastName, Title, City, TerritoryName, CountryRegionCode, [Group])
VALUES (1, 'Jan', 'Kowalski', 'Mr.', 'Warszawa', 'East Europe', 'PL', 'Retail');

-- Dodanie przykładowych danych do DIM_PRODUCT
INSERT INTO Czech.DIM_PRODUCT (ProductID, Name, ListPrice, Color, SubCategoryName, CategoryName, Weight, Size, IsPurchased)
VALUES (101, 'Laptop', 3500.00, 'Silver', 'Electronics', 'Computers', 2.5, '15"', 1);

-- Dodanie przykładowych danych do DIM SALESPERSON
INSERT INTO Czech.DIM SALESPERSON (SalesPersonID, FirstName, LastName, Title, Gender, CountryRegionCode, [Group])
VALUES (501, 'Anna', 'Nowak', 'Ms.', 'F', 'PL', 'Corporate');

-- Testowe dodanie sprzedaży do FACT_SALES
INSERT INTO Czech.FACT_SALES (ProductID, CustomerID, SalesPersonID, OrderDate, ShipDate, OrderQty, UnitPrice, UnitPriceDiscount)
VALUES (101, 1, 501, 20240315, 20240318, 2, 3500.00, 0.05);
```

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Poprawne dane zostały dodane bez problemów.

Niepoprawne dane:

```
--zły customerID
INSERT INTO Czech.FACT_SALES (ProductID, CustomerID, SalesPersonID, OrderDate, ShipDate, OrderQty, UnitPrice, UnitPriceDiscount, LineTotal)
VALUES (101, 9999, NULL, 20240301, 20240305, 2, 100.00, 0.1, 200.00);

--duplikacja
INSERT INTO Czech.DIM_CUSTOMER (CustomerID, FirstName, LastName, Title, City, TerritoryName, CountryRegionCode, [Group])
VALUES (1, 'John', 'Doe', 'Mr', 'Wroclaw', 'Central', 'PL', 'VIP');

INSERT INTO Czech.DIM_CUSTOMER (CustomerID, FirstName, LastName, Title, City, TerritoryName, CountryRegionCode, [Group])
VALUES (1, 'Jane', 'Doe', 'Ms', 'Krakow', 'South', 'PL', 'Regular');

--daty
INSERT INTO Czech.FACT_SALES (ProductID, CustomerID, SalesPersonID, OrderDate, ShipDate, OrderQty, UnitPrice, UnitPriceDiscount, LineTotal)
VALUES (101, 1, NULL, 18000101, 20240305, 2, 100.00, 0.1, 200.00);
```

---

Msg 547, Level 16, State 0, Line 209  
The INSERT statement conflicted with the FOREIGN KEY constraint "FK\_FACT\_SALES\_Customer". The conflict occurred in database "AdventureWorks2014", table "Czech.DIM\_CUSTOMER", column 'CustomerID'.  
The statement has been terminated.

(1 row affected)

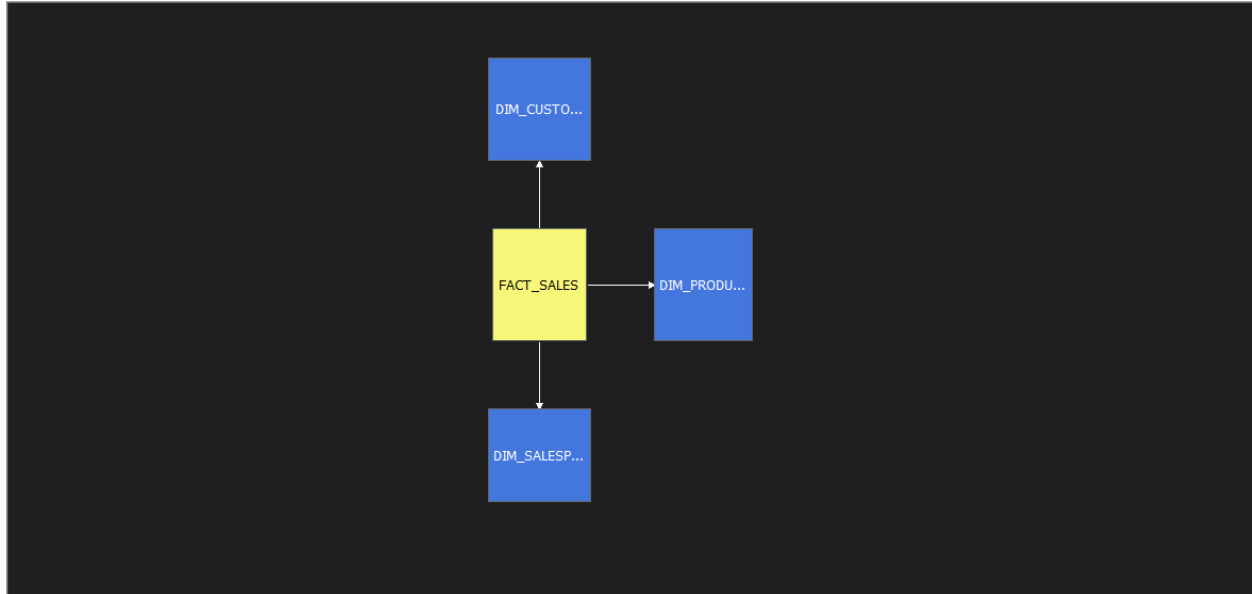
Msg 2627, Level 14, State 1, Line 216  
Violation of PRIMARY KEY constraint 'PK\_DIM\_CUST\_\_34AE64B887BEE36B'. Cannot insert duplicate key in object 'Czech.DIM\_CUSTOMER'. The duplicate key value is (1).  
The statement has been terminated.

Msg 547, Level 16, State 0, Line 220  
The INSERT statement conflicted with the CHECK constraint "CK\_FACT\_SALE\_Order\_\_71F1E3A2". The conflict occurred in database "AdventureWorks2014", table "Czech.FACT\_SALES", column 'OrderDate'.  
The statement has been terminated.

Widzimy, że dla niepoprawnych danych, nasze ustawione ograniczenia działają poprawnie.









## Zadanie 5.

Należy utworzyć projekt Analysis Services, w którym zostanie przygotowana kostka zawierająca utworzone wymiary (klienta, produktu i sprzedawcy) oraz tabelę faktów.



Po utworzeniu kostki dokonać edycji wymiarów (Solution Explorer -> Dimensions lub zakładka Cube Structure -> Dimensions).

### DIM\_PRODUCT










Attributes	
	<b>DIM PRODUCT</b>
	Category Name
	Color
	Is Purchased
	List Price
	Name
	Product ID
	Sub Category Name

Atrybuty:

- Nazwa kategorii
- Kolor
- Czy kupowany
- Cena jednostkowa
- Nazwa

- ProductID
- Nazwa podkategorii








## DIM\_CUSTOMER

Attributes	
	DIM_CUSTOMER
	City
	Country Region Code
	Customer ID
	First Name
	Group
	Last Name
	Territory Name
	Title

Atrybuty:

- Miasto
- Kod kraju
- CustomerID
- Imię
- Grupa geograficzna
- Nazwisko
- Terytorium
- Tytuł

## DIM SALESPERSON

Attributes	
	DIM SALESPERSON
	Country Region Code
	First Name
	Gender
	Group
	Last Name
	Sales Person ID

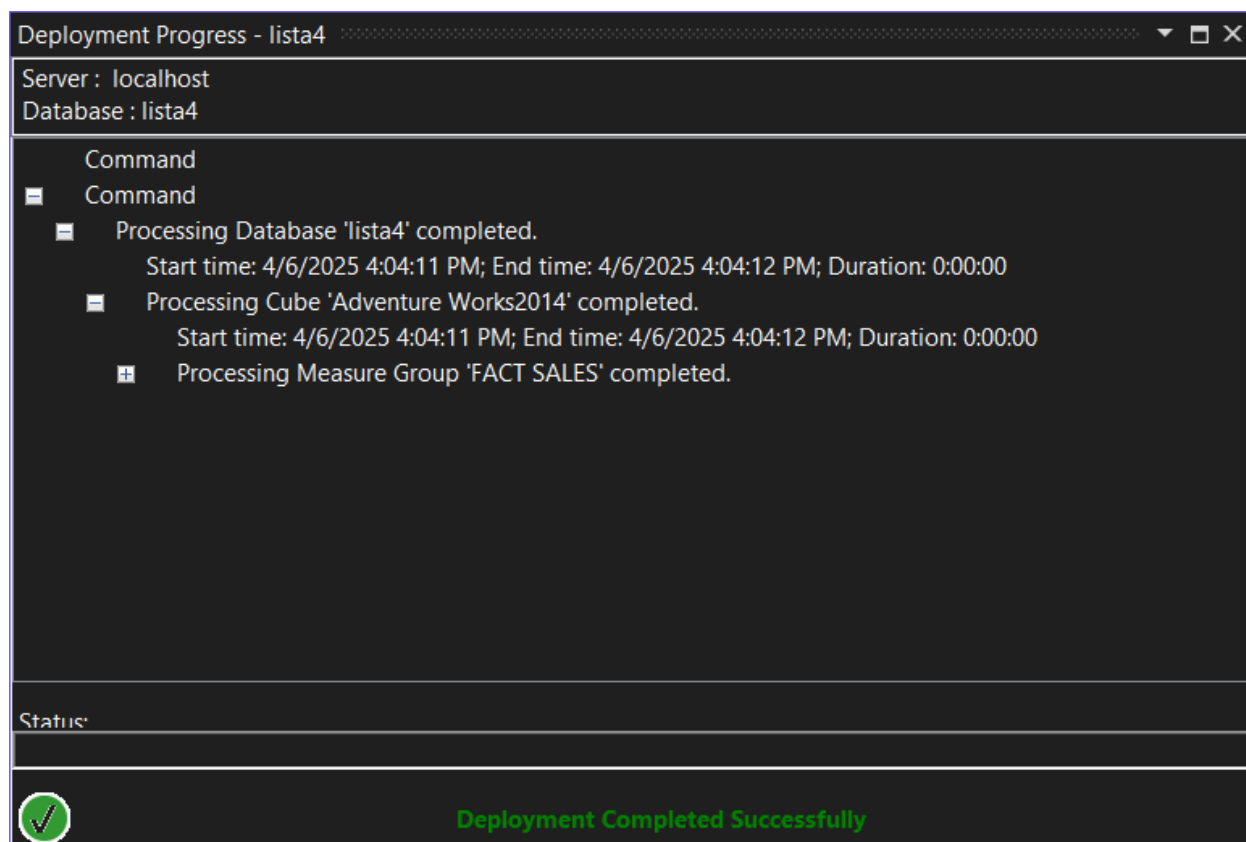
Atrybuty:

- Kod kraju
- Imię

- Płeć
- Grupa
- Nazwisko
- SalesPersonID

## Zadanie 6. Uruchomienie kostki

Przetworzyć kostkę (Przycisk Process).



Kostka została przetworzona poprawnie. Możemy połączyć naszą hurtownię danych z programem Excel, żeby przeprowadzić analizę danych.

## Zadanie 7. Proste raporty

Połączyć się z programem MS Excel i przygotować raporty (tabele i wykresy przestawne), w których zostaną ujęte ciekawe zależności pomiędzy danymi.

Data Connection Wizard


**Select Database and Table**

Select the Database and Table/Cube which contains the data you want.

Select the database that contains the data you want:

lista4

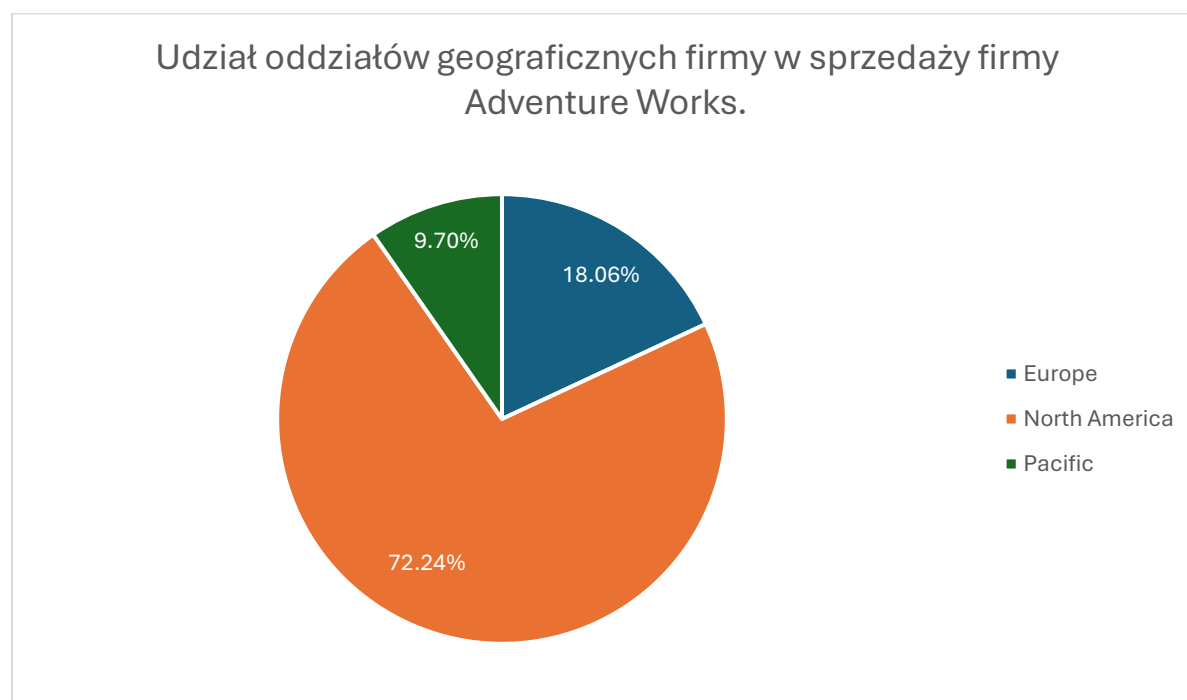
☒ Connect to a specific cube or table:

Name	Description	Modified	Created	Type
 Adventure Works2014		4/6/2025 4:04:12 PM		CUBE

< >

Cancel < Back Next > Finish

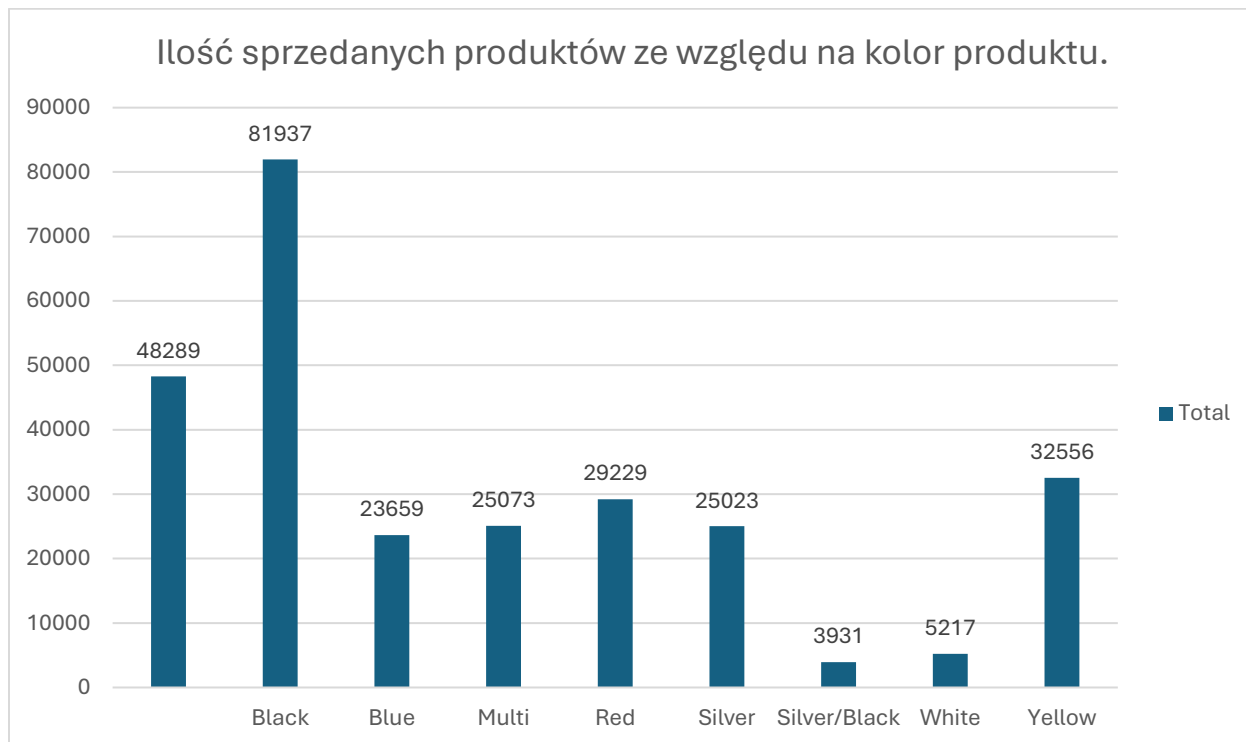
Udział oddziałów geograficznych w całkowitej sprzedaży firmy Adventure Works.



Wnioski:

- Największym rynkiem firmy Adventure Works jest Północna Ameryka (72%).

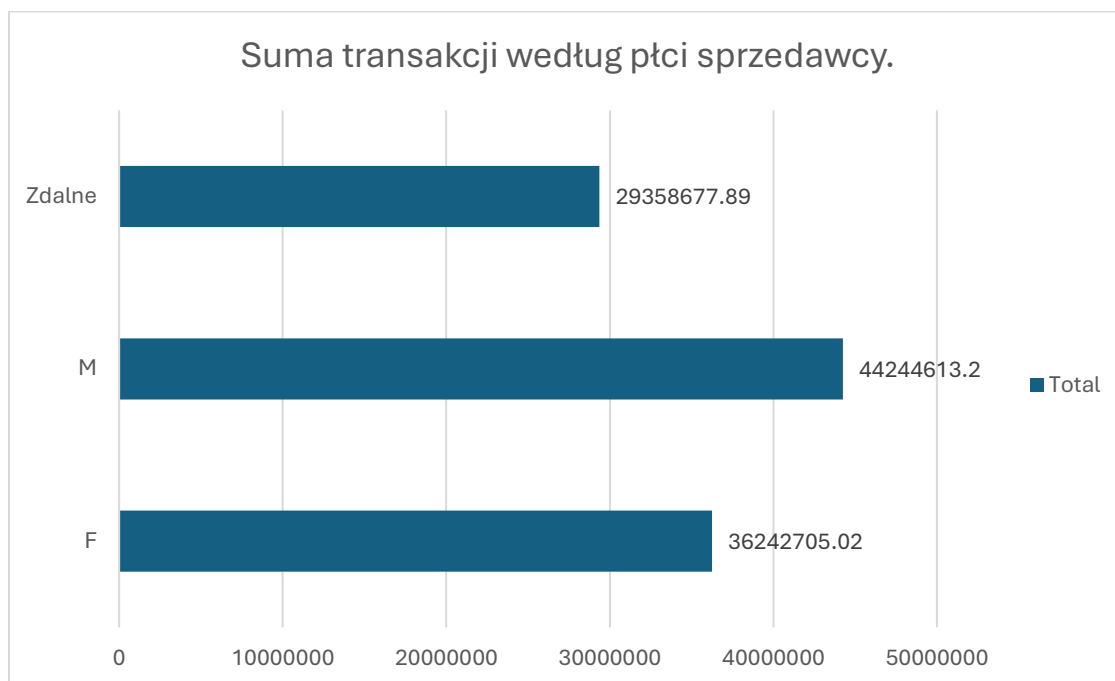
Ilość sprzedanych produktów ze względu na kolor produktu.



Wnioski:

- Najlepiej sprzedają się produkty w kolorze czarnym, bez koloru oraz w kolorze żółtym.

## Suma transakcji według płci sprzedawcy.



## Wnioski:

- Największą sumę transakcji firma osiągnęła na transakcjach dokonywanych przez mężczyzn, następnie kobiet, a na końcu na transakcjach dokonywanych bez udziału sprzedawcy.



## Liczba sprzedanych produktów ze względu na kategorię, podkategorię i kraj.



## Wnioski:

- Komponenty sprzedają się praktycznie tylko w Ameryce Północnej.
- Sprzedaż jest dominowana przez Stany Zjednoczone, w których zakupiono najwięcej produktów z każdej podkategorii.
- W krajach europejskich najlepiej sprzedają się rowery, opony, kaski oraz koszulki.
- Kategorią która sprzedaje się najlepiej są rowery, a następnie ubrania.

## Wnioski:

W ramach listy 4 udało mi się zaprojektować i zaimplementować pełen model hurtowni danych w oparciu o schemat gwiazdy. Obejmuje on tabele wymiarów (klienta, sprzedawcy, produktu) oraz tabelę faktów sprzedaży.

Na podstawie znormalizowanej bazy danych AdventureWorks dokonałem denormalizacji, integracji oraz przygotowania danych do dalszej analizy.

Dane zostały wprowadzone do kostki OLAP utworzonej w Visual Studio Analysis Services. Co umożliwiło połączenie jej z programem Excel aby w szybki i łatwy sposób umożliwić generowanie raportów w celu analizy danych.

Za pomocą programu Excel doszedłem do takich wniosków:

- Głównym rynkiem zbytu firmy Adventure Works jest Ameryka Północna, gdzie wygenerowano 72% całkowitej sprzedaży firmy.
- Najczęściej kupowane są produkty w kolorze czarnym, bez podanego koloru oraz w kolorze żółtym.
- Największe sumy transakcji firma osiągnęła na transakcjach obsługiwanych przez męskich sprzedawców, następnie przez kobiety, a na końcu w transakcjach bez udziału sprzedawcy.
- Najlepiej sprzedają się rowery, a następnie ubrania.