



HURTOWNIE DANYCH

Lista 1 – Modelowanie danych i podstawy SQL



ALEKSANDER STEPANIUK

NR. INDEKSU: 272644

Politechnika Wrocławska, Informatyka Stosowana

Rozwiązania:

Zadanie 1.

Reguły i ograniczenia dziedzinowe:

Reg/01 – Klient może wielokrotnie robić zakupy w tym samym sklepie.

Reg.02 – W sklepie może robić zakupy dowolny klient.

Reg.03 – Każdy zakup realizowany jest przez klienta w sklepie w określonym dniu i godzinie.

Reg/04 – Sklep musi oferować co najmniej jeden produkt.

Reg/05 – Dany produkt może być sprzedawany w wielu sklepach.

Reg/06 – Dany produkt może mieć różne ceny w różnych sklepach.

Reg/07 – Klient robiący zakupy musi nabyć co najmniej jeden produkt.

Reg/08 – Klient może robić zakupy w wielu sklepach.

Reg/09 – Oferta w danym sklepie dotyczy wyłącznie jednego produktu.

Reg/10 – Cena oferty musi być większa od zera.

Reg/12 – Ilość nabytego produktu musi być większa od zera.

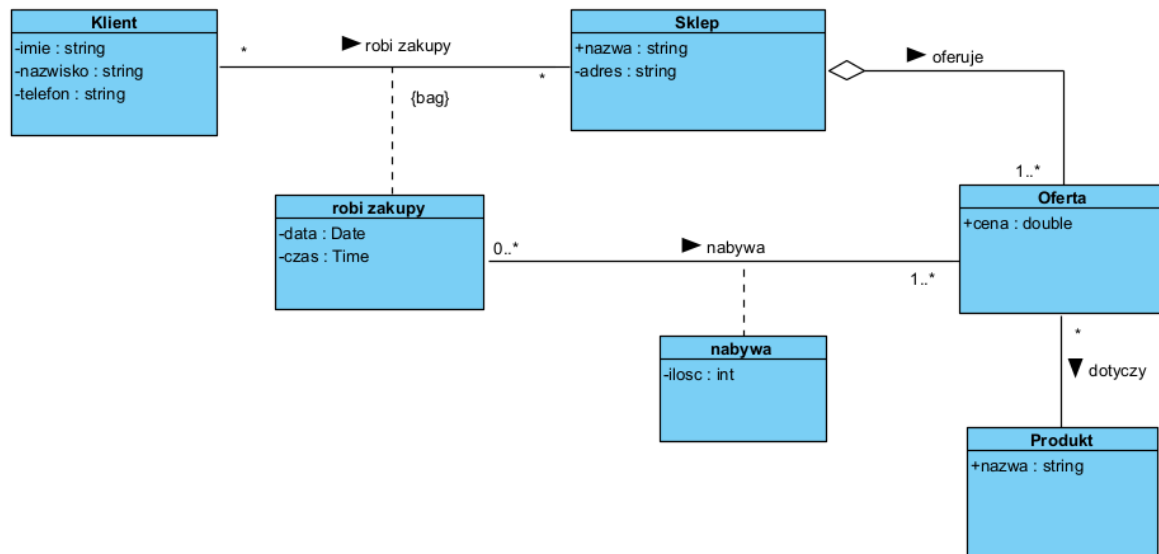
Reg/13 – Imię klienta nie może być puste.

Reg/14 – Nazwisko klienta nie może być puste.

Reg/15 – Nazwa sklepu nie może być pusta.

Reg/16 – Nazwa produktu nie może być pusta.

Diagram klas UML:



DDL SQL:

- [-] [Database Icon] Usługi
 - [+] [Folder Icon] Database Diagrams
 - [-] [Folder Icon] Tables
 - [+] [Folder Icon] System Tables
 - [+] [Folder Icon] FileTables
 - [+] [Folder Icon] External Tables
 - [+] [Folder Icon] Graph Tables
 - [+] [Table Icon] dbo.Klienci
 - [+] [Table Icon] dbo.Nabytki
 - [+] [Table Icon] dbo.Oferty
 - [+] [Table Icon] dbo.Produkty
 - [+] [Table Icon] dbo.Sklepy
 - [+] [Table Icon] dbo.Zakupy
 - [+] [Folder Icon] Dropped Ledger Tables

```
CREATE DATABASE Usługi;
GO

USE Usługi;
GO
```

```
-- TWORZYMY TABELE
```

```
CREATE TABLE Klienci (  
    IdKlienta INT IDENTITY(1, 1) PRIMARY KEY,  
    Imie VARCHAR(50) NOT NULL,  
    Nazwisko VARCHAR(50) NOT NULL,  
    Telefon VARCHAR(15),  
);
```

```
CREATE TABLE Sklepy (  
    IdSklepu INT IDENTITY(1, 1) PRIMARY KEY,  
    Nazwa VARCHAR(50) NOT NULL,  
    Adres VARCHAR(100) NOT NULL,  
);
```

```
CREATE TABLE Produkty (  
    IdProduktu INT IDENTITY(1, 1) PRIMARY KEY,  
    Nazwa VARCHAR(255) NOT NULL,  
);
```

```
CREATE TABLE Oferty (  
    IdOferty INT IDENTITY(1, 1) PRIMARY KEY,  
    Cena DECIMAL(10, 2) NOT NULL,  
    IdProduktu INT NOT NULL,  
    IdSklepu INT NOT NULL,  
    CONSTRAINT Cena_0 CHECK (Cena > 0),  
    CONSTRAINT FK_IdSklepu_0 FOREIGN KEY (IdSklepu) REFERENCES Sklepy(IdSklepu),  
    CONSTRAINT FK_IdProduktu_0 FOREIGN KEY (IdProduktu) REFERENCES Produkty(IdProduktu),  
);
```

```
CREATE TABLE Zakupy (  
    IdZakupu INT IDENTITY(1, 1) PRIMARY KEY,  
    DataZ date NOT NULL,  
    CzasZ time NOT NULL,  
    IdKlienta INT NOT NULL,  
    IdSklepu INT NOT NULL,  
    CONSTRAINT FK_IdKlienta_Z FOREIGN KEY (IdKlienta) REFERENCES Klienci(IdKlienta),  
    CONSTRAINT FK_IdSklepu_Z FOREIGN KEY (IdSklepu) REFERENCES Sklepy(IdSklepu),  
);
```

```
CREATE TABLE Nabytki (  
    IdNabytku INT IDENTITY(1, 1) PRIMARY KEY,  
    Ilosc INT NOT NULL,  
    IdOferty INT NOT NULL,  
    IdZakupu INT NOT NULL,  
    CONSTRAINT Ilosc_N CHECK (Ilosc > 0),  
    CONSTRAINT FK_IdOferty_N FOREIGN KEY (IdOferty) REFERENCES Oferty(IdOferty),  
    CONSTRAINT FK_IdZakupu_N FOREIGN KEY (IdZakupu) REFERENCES Zakupy(IdZakupu),  
);
```

Dodawanie danych poprawnych:

```
-- DODAJEMY DANE DO BAZY
```

```
INSERT INTO Klienci (Imie, Nazwisko, Telefon)
```

```
VALUES ( Imie 'Jan', Nazwisko 'Kowalski', Telefon '123456789'),  
        ( Imie 'Anna', Nazwisko 'Nowak', Telefon '987654321'),  
        ( Imie 'Piotr', Nazwisko 'Kowalczyk', Telefon '456123789');
```

```
INSERT INTO Sklepy (Nazwa, Adres)
```

```
VALUES ( Nazwa 'Biedronka', Adres 'ul. Kolejowa 1, 00-001 Warszawa'),  
        ( Nazwa 'Lidl', Adres 'ul. Dworcowa 2, 00-002 Warszawa'),  
        ( Nazwa 'Auchan', Adres 'ul. Szkolna 3, 00-003 Warszawa');
```

```
INSERT INTO Produkty (Nazwa)
```

```
VALUES ( Nazwa 'Mleko'),  
        ( Nazwa 'Chleb'),  
        ( Nazwa 'Jogurt'),  
        ( Nazwa 'Ser');
```

```
INSERT INTO Oferty (Cena, IdProduktu, IdSklepu)
```

```
VALUES (Cena 2.50, IdProduktu 1, IdSklepu 1),  
       (Cena 3.00, IdProduktu 2, IdSklepu 1),  
       (Cena 1.50, IdProduktu 3, IdSklepu 1),  
       (Cena 4.00, IdProduktu 4, IdSklepu 1),  
       (Cena 2.00, IdProduktu 1, IdSklepu 2),  
       (Cena 2.50, IdProduktu 2, IdSklepu 2),  
       (Cena 1.00, IdProduktu 3, IdSklepu 2),  
       (Cena 3.50, IdProduktu 4, IdSklepu 2),  
       (Cena 2.00, IdProduktu 1, IdSklepu 3),  
       (Cena 2.50, IdProduktu 2, IdSklepu 3),  
       (Cena 1.00, IdProduktu 3, IdSklepu 3),  
       (Cena 3.50, IdProduktu 4, IdSklepu 3);
```

```
INSERT INTO Zakupy (DataZ, CzasZ, IdKlienta, IdSklepu)
```

```
VALUES (DataZ '2021-01-01', CzasZ '12:00:00', IdKlienta 1, IdSklepu 1),  
       (DataZ '2021-01-02', CzasZ '13:00:00', IdKlienta 2, IdSklepu 2),  
       (DataZ '2021-01-03', CzasZ '14:00:00', IdKlienta 3, IdSklepu 3);
```

```
INSERT INTO Nabytki (Ilosc, IdOferty, IdZakupu)
```

```
VALUES (Ilosc 2, IdOferty 1, IdZakupu 1),  
       (Ilosc 1, IdOferty 2, IdZakupu 1),  
       (Ilosc 3, IdOferty 3, IdZakupu 2),  
       (Ilosc 1, IdOferty 4, IdZakupu 2),  
       (Ilosc 2, IdOferty 5, IdZakupu 3),  
       (Ilosc 1, IdOferty 6, IdZakupu 3),  
       (Ilosc 3, IdOferty 7, IdZakupu 3),  
       (Ilosc 1, IdOferty 8, IdZakupu 3),  
       (Ilosc 2, IdOferty 9, IdZakupu 3),  
       (Ilosc 1, IdOferty 10, IdZakupu 3),  
       (Ilosc 3, IdOferty 11, IdZakupu 3),  
       (Ilosc 1, IdOferty 12, IdZakupu 3);
```

Próba dodania niepoprawnych danych:

```
-- PRÓBA DODANIA NIEPOPRAWNYCH DANYCH

-- a) naruszenie ograniczenia NOT NULL
✓ INSERT INTO Klienci (Imie, Nazwisko, Telefon)
VALUES ( Imie NULL, Nazwisko 'Kowalski', Telefon NULL);

-- poprawnie byłoby tak:
✓ INSERT INTO Klienci (Imie, Nazwisko, Telefon)
VALUES ( Imie 'Jan', Nazwisko 'Kowalski', Telefon NULL);

-- b) naruszenie ograniczenia CHECK
✓ INSERT INTO Oferty (Cena, IdProduktu, IdSklepu)
VALUES ( Cena -1, IdProduktu 1, IdSklepu 1);

-- poprawnie byłoby tak:
✓ INSERT INTO Oferty (Cena, IdProduktu, IdSklepu)
VALUES ( Cena 2.50, IdProduktu 1, IdSklepu 1);

-- c) naruszenie ograniczenia FOREIGN KEY
✓ INSERT INTO Oferty (Cena, IdProduktu, IdSklepu)
VALUES ( Cena 2.50, IdProduktu 1, IdSklepu 4);

-- poprawnie byłoby tak:
✓ INSERT INTO Oferty (Cena, IdProduktu, IdSklepu)
VALUES ( Cena 2.50, IdProduktu 1, IdSklepu 1);
```

messages

```
(3 rows affected)

(3 rows affected)

(4 rows affected)

(12 rows affected)

(3 rows affected)

(12 rows affected)
Msg 515, Level 16, State 2, Line 111
Cannot insert the value NULL into column 'Imie', table 'Uslugi.dbo.Klienci'; column does not allow nulls. INSERT fails.
The statement has been terminated.

(1 row affected)
Msg 547, Level 16, State 0, Line 119
The INSERT statement conflicted with the CHECK constraint "Cena_0". The conflict occurred in database "Uslugi", table "dbo.Oferty", column 'Cena'.
The statement has been terminated.

(1 row affected)
Msg 547, Level 16, State 0, Line 127
The INSERT statement conflicted with the FOREIGN KEY constraint "FK_IdSklepu_O". The conflict occurred in database "Uslugi", table "dbo.Sklepy", column 'IdSklepu'.
The statement has been terminated.

(1 row affected)

Completion time: 2025-03-09T21:59:22.5948001+01:00
```

Zadanie 2.

1)

```
-- 1)
SELECT COUNT(*) FROM [Production].[Product] --504
SELECT COUNT(*) FROM [Production].[ProductCategory] --4
SELECT COUNT(*) FROM [Production].[ProductSubcategory] --37
```

Results		Messages
(No column name)		
1	504	
(No column name)		
1	4	
(No column name)		
1	37	
✓ Query executed successfully.		

2)

```
-- 2)
SELECT *
FROM [Production].[Product] P
WHERE P.Color is NULL;
```

ProductID	Name	ProductNumber	MakeFlag	FinishedGoodsFlag	Color	SafetyStockLevel	ReorderPoint	StandardCost	ListPrice	Size	SizeUnitMeasureCode	WeightUnitMeasureCode	Weight	DaysToManu
1	Adjustable Race	AR-5381	0	0	NULL	1000	750	0,00	0,00	NULL	NULL	NULL	NULL	0
2	Bearing Ball	BA-8327	0	0	NULL	1000	750	0,00	0,00	NULL	NULL	NULL	NULL	0
3	BB Ball Bearing	BE-2349	1	0	NULL	800	600	0,00	0,00	NULL	NULL	NULL	NULL	1
4	Headset Ball Bearings	BE-2908	0	0	NULL	800	600	0,00	0,00	NULL	NULL	NULL	NULL	0
5	Blade	BL-2036	1	0	NULL	800	600	0,00	0,00	NULL	NULL	NULL	NULL	1
6	Crown Race	CR-9981	0	0	NULL	1000	750	0,00	0,00	NULL	NULL	NULL	NULL	0
7	Chain Stays	CS-2812	1	0	NULL	1000	750	0,00	0,00	NULL	NULL	NULL	NULL	1
8	Decal 1	DC-8732	0	0	NULL	1000	750	0,00	0,00	NULL	NULL	NULL	NULL	0
9	Decal 2	DC-9824	0	0	NULL	1000	750	0,00	0,00	NULL	NULL	NULL	NULL	0
10	Down Tube	DT-2377	1	0	NULL	800	600	0,00	0,00	NULL	NULL	NULL	NULL	1
11	Mountain End Caps	EC-M092	1	0	NULL	1000	750	0,00	0,00	NULL	NULL	NULL	NULL	1
12	Road End Caps	EC-R098	1	0	NULL	1000	750	0,00	0,00	NULL	NULL	NULL	NULL	1
13	Touring End Caps	EC-T209	1	0	NULL	1000	750	0,00	0,00	NULL	NULL	NULL	NULL	1
14	Fork End	FE-3760	1	0	NULL	800	600	0,00	0,00	NULL	NULL	NULL	NULL	1
15	Flat Washer 1	FW-1000	0	0	NULL	1000	750	0,00	0,00	NULL	NULL	NULL	NULL	0
16	Flat Washer 6	FW-1200	0	0	NULL	1000	750	0,00	0,00	NULL	NULL	NULL	NULL	0
17	Flat Washer 2	FW-1400	0	0	NULL	1000	750	0,00	0,00	NULL	NULL	NULL	NULL	0
18	Flat Washer 9	FW-3400	0	0	NULL	1000	750	0,00	0,00	NULL	NULL	NULL	NULL	0

Query executed successfully. ZAWO-LAPTOP (16.0 RTM) ZAWO-LAPTOP\aliks (59) AdventureWorks2022 00:00:00 248 rows

3)

```
-- 3)
SELECT YEAR(OrderDate) as "Year", SUM(TotalDue) as "Total"
FROM [Sales].[SalesOrderHeader]
GROUP BY YEAR(OrderDate)
ORDER BY YEAR(OrderDate);
```

	Year	Total
1	2011	14155699,525
2	2012	37675700,312
3	2013	48965887,9632
4	2014	22419498,3157

4)

```
-- 4)
--a) ilu jest klientów
SELECT COUNT(DISTINCT(PersonID))
FROM [Sales].[Customer] C
    JOIN [Person].[Person] ON [Person].[BusinessEntityID] = C.PersonID;
```

100 %	
Results Messages	
	(No column name)
1	19119

```
--b) ilu jest klientów w danym regionie
SELECT
    Sales.SalesTerritory.Name as "Territory",
    COUNT(DISTINCT(PersonID)) as "Customers"
FROM Sales.Customer
    JOIN Person.Person ON Person.BusinessEntityID = Sales.Customer.PersonID
    JOIN Sales.SalesTerritory ON Sales.Customer.TerritoryID = Sales.SalesTerritory.TerritoryID
GROUP BY SalesTerritory.Name;
```

Results Messages		
	Territory	Customers
1	Australia	3625
2	Central	69
3	Canada	1677
4	France	1844
5	Northwest	3428
6	United Kingdom	1951
7	Southwest	4565
8	Southeast	91
9	Northeast	57
10	Germany	1812

```
--c) ilu jest sprzedawców
SELECT COUNT(*)
FROM [Sales].[SalesPerson] S;
```

Results		Messages
	(No column name)	
1	17	

```
--d) ilu jest sprzedawców w danym regionie (LEFT JOIN z nullem a join bez nulla)
SELECT
    Sales.SalesTerritory.Name as "Territory",
    COUNT(*) as "SalesPersons"
FROM Sales.SalesPerson
    LEFT JOIN Sales.SalesTerritory ON SalesPerson.TerritoryID = SalesTerritory.TerritoryID
GROUP BY SalesTerritory.Name;
```

Results			Messages
	Territory	SalesPersons	
1	NULL	3	
2	Australia	1	
3	Canada	2	
4	Central	1	
5	France	1	
6	Germany	1	
7	Northeast	1	
8	Northwest	3	
9	Southeast	1	
10	Southwest	2	
11	United Kingdom	1	

5)

```
-- 5)
SELECT
    Year(S.OrderDate) as "Year",
    COUNT(S.SalesOrderID) as "Orders"
FROM Sales.SalesOrderHeader S
GROUP BY Year(S.OrderDate)
ORDER BY Year(S.OrderDate);
```

100 %

Results		Messages
	Year	Orders
1	2011	1607
2	2012	3915
3	2013	14182
4	2014	11761

6)

```
-- 6) Podaj produkty, które nie zostały kupione przez żadnego klienta. Zestawienie pogrupuj według kategorii i podkategorii.
SELECT
    PC.Name as "Category",
    PSC.Name as "Subcategory",
    PP.ProductID as "ProductID",
    PP.Name as "Product"
FROM Production.Product PP
    LEFT JOIN Sales.SalesOrderDetail SOD ON PP.ProductID = SOD.ProductID
    LEFT JOIN Production.ProductSubcategory PSC ON PP.ProductSubcategoryID = PSC.ProductSubcategoryID
    LEFT JOIN Production.ProductCategory PC ON PSC.ProductCategoryID = PC.ProductCategoryID
WHERE SOD.ProductID IS NULL;
```

Results		Messages		
	Category	Subcategory	ProductID	Product
1	NULL	NULL	1	Adjustable Race
2	NULL	NULL	2	Bearing Ball
3	NULL	NULL	3	BB Ball Bearing
4	NULL	NULL	4	Headset Ball Bearings
5	NULL	NULL	316	Blade
6	NULL	NULL	317	LL Crankarm
7	NULL	NULL	318	ML Crankarm
8	NULL	NULL	319	HL Crankarm
9	NULL	NULL	320	Chainring Bolts
10	NULL	NULL	321	Chainring Nut
11	NULL	NULL	322	Chainring
12	NULL	NULL	323	Crown Race
13	NULL	NULL	324	Chain Stays
14	NULL	NULL	325	Decal 1
15	NULL	NULL	326	Decal 2
16	NULL	NULL	327	Down Tube
17	NULL	NULL	328	Mountain End Caps
18	NULL	NULL	329	Road End Caps
19	NULL	NULL	330	Touring End Caps
20	NULL	NULL	331	Fork End

Query executed successfully.

```

SELECT
    PC.Name as "Category",
    PSC.Name as "Subcategory",
    COUNT(PP.ProductID) as "Products"
FROM Production.Product PP
    LEFT JOIN Sales.SalesOrderDetail SOD ON PP.ProductID = SOD.ProductID
    LEFT JOIN Production.ProductSubcategory PSC ON PP.ProductSubcategoryID = PSC.ProductSubcategoryID
    LEFT JOIN Production.ProductCategory PC ON PSC.ProductCategoryID = PC.ProductCategoryID
WHERE SOD.ProductID IS NULL
GROUP BY PC.Name, PSC.Name;

```

Results		Messages	
	Category	Subcategory	Products
1	NULL	NULL	209
2	Components	Bottom Brackets	1
3	Components	Forks	1
4	Components	Handlebars	1
5	Accessories	Lights	3
6	Components	Mountain Frames	2
7	Accessories	Panniers	1
8	Accessories	Pumps	1
9	Components	Road Frames	13
10	Components	Saddles	1
11	Clothing	Shorts	1
12	Components	Touring Frames	1
13	Components	Wheels	3

7)

```
SELECT
    PSC.Name as "Subcategory",
    MIN(SOD.UnitPriceDiscount * SOD.OrderQty * SOD.UnitPrice) as "MinDiscount",
    MAX(SOD.UnitPriceDiscount * SOD.OrderQty * SOD.UnitPrice) as "MaxDiscount"
FROM Sales.SalesOrderDetail SOD
    JOIN Production.Product PP ON SOD.ProductID = PP.ProductID
    JOIN Production.ProductSubcategory PSC ON PP.ProductSubcategoryID = PSC.ProductSubcategoryID
GROUP BY PSC.Name;
```

100 %			
Results Messages			
	Subcategory	MinDiscount	MaxDiscount
1	Mountain Bikes	0,00	0,40
2	Road Bikes	0,00	0,30
3	Touring Bikes	0,00	0,20
4	Handlebars	0,00	0,05
5	Bottom Brackets	0,00	0,00
6	Brakes	0,00	0,02
7	Chains	0,00	0,02
8	Cranksets	0,00	0,02
9	Derailleurs	0,00	0,02
10	Forks	0,00	0,02
11	Headsets	0,00	0,05
12	Mountain Frames	0,00	0,02
13	Pedals	0,00	0,02
14	Road Frames	0,00	0,05
15	Saddles	0,00	0,02
16	Touring Frames	0,00	0,05

100 %			
Results Messages			
	Subcategory	MinDiscount	MaxDiscount
1	Mountain Bikes	0,00	3248,4279
2	Road Bikes	0,00	2551,485
3	Touring Bikes	0,00	4005,2376
4	Handlebars	0,00	19,5976
5	Bottom Brackets	0,00	0,00
6	Brakes	0,00	16,0602
7	Chains	0,00	2,8174
8	Cranksets	0,00	65,7704
9	Derailleurs	0,00	14,858
10	Forks	0,00	37,2692
11	Headsets	0,00	50,6336
12	Mountain Frames	0,00	174,1102
13	Pedals	0,00	13,1528
14	Road Frames	0,00	245,3674
15	Saddles	0,00	5,4483
16	Touring Frames	0,00	634,9731
17	Wheels	0,00	134,9762
18	Bib-Shorts	0,00	47,0198
19	Caps	0,00	12,1365

8)

```
-- 8) Podaj produkty, których cena jest wyższa od średniej ceny produktów w sklepie.  
SELECT  
    PP.ProductID as "ProductID",  
    PP.Name as "ProductName",  
    PP.ListPrice as "Price"  
FROM [Production].[Product] PP  
WHERE PP.ListPrice > (SELECT AVG(ListPrice) FROM [Production].[Product]);
```

Results		Messages	
	ProductID	ProductName	Price
1	680	HL Road Frame - Black, 58	1431,50
2	706	HL Road Frame - Red, 58	1431,50
3	717	HL Road Frame - Red, 62	1431,50
4	718	HL Road Frame - Red, 44	1431,50
5	719	HL Road Frame - Red, 48	1431,50
6	720	HL Road Frame - Red, 52	1431,50
7	721	HL Road Frame - Red, 56	1431,50
8	731	ML Road Frame - Red, 44	594,83
9	732	ML Road Frame - Red, 48	594,83
10	733	ML Road Frame - Red, 52	594,83
11	734	ML Road Frame - Red, 58	594,83
12	735	ML Road Frame - Red, 60	594,83
13	739	HL Mountain Frame - Silver, 42	1364,50
14	740	HL Mountain Frame - Silver, 44	1364,50
15	741	HL Mountain Frame - Silver, 48	1364,50
16	742	HL Mountain Frame - Silver, 46	1364,50
17	743	HL Mountain Frame - Black, 42	1349,60
18	744	HL Mountain Frame - Black, 44	1349,60

9)

```
-- 9) Ile średnio produktów w każdej kategorii sprzedaje się w poszczególnych miesiącach?
SELECT
    YEAR(S.OrderDate) as "Year",
    MONTH(S.OrderDate) as "Month",
    PC.Name as "Category",
    COUNT(SOD.ProductID) / COUNT(DISTINCT(SOD.SalesOrderID)) as "AvgProductsSold"
FROM [Sales].[SalesOrderDetail] SOD
    JOIN [Sales].[SalesOrderHeader] S ON SOD.SalesOrderID = S.SalesOrderID
    JOIN [Production].[Product] PP ON SOD.ProductID = PP.ProductID
    JOIN [Production].[ProductSubcategory] PSC ON PP.ProductSubcategoryID = PSC.ProductSubcategoryID
    JOIN [Production].[ProductCategory] PC ON PSC.ProductCategoryID = PC.ProductCategoryID
GROUP BY YEAR(S.OrderDate), MONTH(S.OrderDate), PC.Name
ORDER BY YEAR(S.OrderDate), MONTH(S.OrderDate), PC.Name;
```

Results		Messages		
	Year	Month	Category	AvgProductsSold
1	2011	5	Accessories	2
2	2011	5	Bikes	4
3	2011	5	Clothing	2
4	2011	5	Components	3
5	2011	6	Bikes	1
6	2011	7	Accessories	2
7	2011	7	Bikes	2
8	2011	7	Clothing	2
9	2011	7	Components	3
10	2011	8	Accessories	2
11	2011	8	Bikes	3
12	2011	8	Clothing	2
13	2011	8	Components	4
14	2011	9	Bikes	1
15	2011	10	Accessories	2
16	2011	10	Bikes	4
17	2011	10	Clothing	2

10)

```
-- 10) Ile średnio czasu Klient czeka na dostawę zamówionych produktów? Przygotuj zestawienie w zależności od kodu regionu (SalesTerritory.CountryRegionCode).  
SELECT  
    ST.CountryRegionCode as "CountryRegionCode",  
    AVG(CAST(DATEDIFF(DAY, S.OrderDate, S.ShipDate) as float)) as "AvgDaysToShip"  
FROM [Sales].[SalesOrderHeader] S  
JOIN [Sales].[SalesTerritory] ST ON S.TerritoryID = ST.TerritoryID  
GROUP BY ST.CountryRegionCode  
ORDER BY ST.CountryRegionCode;
```

Results		Messages
	CountryRegionCode	AvgDaysToShip
1	AU	7
2	CA	7
3	DE	7,00343118566527
4	FR	7
5	GB	7
6	US	7

Wnioski:

Konceptualny model danych „Usługi” był niekompletny, bo nie pozwalał na ustawienie różnych cen tych samych produktów w różnych sklepach. Dodałem więc dodatkową klasę „Oferta”, która dotyczy produktu i definiuje jego cenę. Pozostałe dane produktów, które pozostają niezmiennie we wszystkich sklepach (takie jak nazwa produktu) pozostała w klasie „Produkt”.

W sklepie istnieją 504 różne produkty, podzielone na 4 kategorie oraz 37 podkategorii. Można nadmienić prawie 20 tysięcy różnych klientów oraz 17 pracowników sklepu. Jest łącznie 10 regionów, w tym Australia, Francja, Kanada i tak dalej. Większość z nich posiada tysiące różnych klientów, natomiast pozostałe trzy są o wiele mniej liczne w klientów i posiadają ich mniej niż 100. Trzech sprzedawców nie posiada żadnego regionu wpisanego (NULL).

Wiele produktów nie ma zdefiniowanego koloru, kategorii lub podkategorii (czyli posiada wartość NULL).

Wraz z biegiem lat, sklep zdobywał coraz więcej łącznie wykonanych transakcji (liczba ta rosła z każdym rokiem), jedynie rok 2014 pokazuje mniejszą wartość niż jego poprzednik z roku 2013, jednak wynika to z tego, że dane dotyczące roku 2014 nie są kompletne, a więc dotyczą jedynie miesięcy od stycznia do czerwca 2014 roku.

Jest prawie 250 produktów, które nie zostały nigdy zakupione przez żadnego klienta (co stanowi prawie połowę wszystkich produktów). Większość sprzedaży następuje jednak w okresie letnim (wakacje, ciepła pogoda, a jako że jest to sklep rowerowy, wtedy następuje największa sprzedaż).

Klient czeka średnio tydzień na dostawę zamówionych produktów, niezależnie od kodu regionu.