

Hurtownie danych – Spr. 3.

PWr. WIZ, Informatyka, Data: 24.03.2020

Student	Email: 242493@student.pwr.edu.pl	Ocena
Indeks	<u>242493</u>	
Imię	<u>Arkadiusz</u>	
Nazwisko	<u>Rasz</u>	

Spis treści

Zad. 1. Przygotowanie schematu	2
Zad. 2. Tworzenie tabel wymiarów i tabeli faktów	2
Zad. 3. Wypełnianie danych – denormalizacja źródłowej bazy	3
Zad. 4. Więzy integralności	6
Sprawdzanie więzów integralności	7
Klucze główne:	7
Klucze obce:	8
Zad. 5. Tworzenie kostki	9
Zad. 6. Uruchomienie kostki	11
Zad. 7. Proste raporty	12
Podsumowanie i wnioski.....	13

Zad. 1. Przygotowanie schematu

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

```
CREATE SCHEMA Rasz;
```

Zad. 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)
- **FACT_SALES** (ProductID, CustomerID, SalesPersonID, OrderDate, ShipDate, OrderQty, UnitPrice, UnitPriceDiscount, LineTotal)

```
CREATE TABLE [Rasz].[Dim_Customer](
    [CustomerID] INT NOT NULL,
    [FirstName] NVARCHAR(50) NULL,
    [LastName] NVARCHAR(50) NULL,
    [Title] NVARCHAR(50) NULL,
    [City] NVARCHAR(30) NULL,
    [TerritoryName] NVARCHAR(50),
    [CountryRegionCode] NVARCHAR(3) NULL,
    [Group] NVARCHAR(50) NULL
);
```

```
CREATE TABLE [Rasz].[Dim_Salesperson] (
    [SalesPersonID] INT NOT NULL,
    [FirstName] NVARCHAR(50) NULL,
    [LastName] NVARCHAR(50),
    [Title] NVARCHAR(50) NULL,
    [Gender] NCHAR(1) NULL,
    [CountryRegionCode] NVARCHAR(3) NULL,
    [Group] NVARCHAR(50) NULL
);
```

```
CREATE TABLE [Rasz].[Dim_Product](
    [ProductID] INT NOT NULL,
    [Name] NVARCHAR(50) NOT NULL,
    [ListPrice] MONEY NOT NULL,
    [Color] NVARCHAR(15) NULL,
    [SubCategoryName] NVARCHAR(50) NULL,
    [CategoryName] NVARCHAR(50) NULL,
    [Weight] DECIMAL(8, 2) NULL,
    [Size] NVARCHAR(5) NULL,
    [IsPurchased] BIT
);
```

```
CREATE TABLE [Rasz].[Fact_Sales] (
    [ProductID] INT NOT NULL,
    [CustomerID] INT NOT NULL,
    [SalesPersonID] INT NULL,
    [OrderDate] INT NOT NULL,
    [ShipDate] INT NULL,
    [OrderQty] SMALLINT NOT NULL,
    [UnitPrice] MONEY NOT NULL,
    [UnitPriceDiscount] MONEY NOT NULL,
    [LineTotal] NUMERIC(38, 6) NOT NULL
);
```

Zad. 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.

```
INSERT INTO Rasz.Dim_Product(ProductID, Name, ListPrice, Color, SubCategoryName,
                             CategoryName, Weight, Size, IsPurchased)
(
    select distinct P.ProductID, P.Name, P.ListPrice, P.Color, S.Name, C.Name,
                   P.Weight, P.Size, IIF(D.SalesOrderDetailID is null, 0, 1)
    from Production.Product P
    left join Production.ProductSubcategory S
        on P.ProductSubcategoryID = S.ProductSubcategoryID
    left join Production.ProductCategory C
        on S.ProductCategoryID = C.ProductCategoryID
    left join Sales.SalesOrderDetail D
        on P.ProductID = D.ProductID
);
```

```
INSERT INTO [Rasz].[Dim_Customer]([CustomerID], [FirstName], [LastName], [Title],
                                   [City], [TerritoryName], [CountryRegionCode], [Group])
(
    select C.CustomerID, P.FirstName, P.LastName, P.Title, A.City, T.Name,
           S.CountryRegionCode, T.[Group]
    from Sales.Customer C
    left join Person.Person P
        on C.PersonID = P.BusinessEntityID
    left join Person.BusinessEntityAddress B
        on B.BusinessEntityID = P.BusinessEntityID
    left join Person.Address A
        on A.AddressID = B.AddressID AND B.AddressTypeID = 2
    left join Sales.SalesTerritory T
        on T.TerritoryID = C.TerritoryID
    left join Person.StateProvince S
        on S.StateProvinceID = A.StateProvinceID
);
```

```
INSERT INTO [Rasz].[Dim_Salesperson]([SalesPersonID], [FirstName], [LastName],
                                      [Title], [Gender], [CountryRegionCode], [Group])
(
    select SP.BusinessEntityID, P.FirstName, P.LastName, P.Title, E.Gender,
           ST.CountryRegionCode, ST.[Group]
    from Sales.SalesPerson SP
    left join Sales.SalesTerritory ST
        on SP.TerritoryID = ST.TerritoryID
    left join Person.Person P
        on SP.BusinessEntityID = P.BusinessEntityID
    left join HumanResources.Employee E
        on SP.BusinessEntityID = E.BusinessEntityID
);
```

```

INSERT INTO [Rasz].[Fact_Sales]([ProductID],[CustomerID],[SalesPersonID],[OrderDate],
    [ShipDate], [OrderQty], [UnitPrice], [UnitPriceDiscount], [LineTotal])
(
    select SOD.ProductID, SOH.CustomerID, SOH.SalesPersonID,
        DATEPART(YEAR, SOH.OrderDate) * 10000 +
            DATEPART(MONTH, SOH.OrderDate) * 100 +
            DATEPART(DAY, SOH.OrderDate),
        DATEPART(YEAR, SOH.ShipDate) * 10000 +
            DATEPART(MONTH, SOH.ShipDate) * 100 +
            DATEPART(DAY, SOH.ShipDate),
        SOD.OrderQty, SOD.UnitPrice, SOD.UnitPriceDiscount, SOD.LineTotal
    from Sales.SalesOrderHeader SOH
    join Sales.SalesOrderDetail SOD
        on SOH.SalesOrderID = SOD.SalesOrderID
);

```

Wyniki wstawiania danych do wszystkich tabel:

```

Messages

(504 rows affected)

(19844 rows affected)

(17 rows affected)

(121317 rows affected)

```

Dim_Customer:

	CustomerID	FirstName	LastName	Title	City	TerritoryName	CountryRegionCode	Group
18...	12130	Suzanne	Zhou	NULL	Pad...	Germany	DE	Europe
18...	12131	Randall	Doming...	NULL	Dun...	France	FR	Europe
18...	12132	Kaitlyn	Henders...	NULL	Tre...	France	FR	Europe
18...	12133	Colleen	Xie	NULL	Lan...	United Kingdom	GB	Europe
18...	12134	Hector	Alonso	NULL	Stok...	United Kingdom	GB	Europe
18...	12135	Fernando	Taylor	NULL	El C...	Southwest	US	North America
18...	12136	Blake	Turner	NULL	Cliff...	Canada	CA	North America
18...	12137	Jack	Green	NULL	Nov...	Southwest	US	North America
18...	12138	Colin	Jai	NULL	Bea...	Northwest	US	North America
18...	12139	Franklin	Yang	NULL	Belli...	Northwest	US	North America
18...	12140	Oscar	Russell	NULL	Wes...	Canada	CA	North America
18...	12141	Kaitlyn	Lee	NULL	Bea...	Northwest	US	North America
18...	12142	Gabriella	Torres	NULL	Mil...	Northwest	US	North America
18...	12143	Lozano	Turner	NULL	Port...	Northwest	US	North America

Dim_Product:

	ProductID	Name	ListPrice	Color	SubCategoryName	CategoryName	Weight	Size	IsPurchased
235	730	LL Road Frame - Re...	337.22	Red	Road Frames	Components	2.50	62	1
236	731	ML Road Frame - R...	594.83	Red	Road Frames	Components	2.22	44	0
237	732	ML Road Frame - R...	594.83	Red	Road Frames	Components	2.26	48	1
238	733	ML Road Frame - R...	594.83	Red	Road Frames	Components	2.30	52	1
239	734	ML Road Frame - R...	594.83	Red	Road Frames	Components	2.36	58	0
240	735	ML Road Frame - R...	594.83	Red	Road Frames	Components	2.38	60	0
241	736	LL Road Frame - Bla...	337.22	Black	Road Frames	Components	2.32	44	1
242	737	LL Road Frame - Bla...	337.22	Black	Road Frames	Components	2.36	48	0
243	738	LL Road Frame - Bla...	337.22	Black	Road Frames	Components	2.40	52	1
244	739	HL Mountain Frame ...	1364.50	Silver	Mountain Frames	Components	2.72	42	1
245	740	HL Mountain Frame ...	1364.50	Silver	Mountain Frames	Components	2.76	44	0
246	741	HL Mountain Frame ...	1364.50	Silver	Mountain Frames	Components	2.80	48	1
247	742	HL Mountain Frame ...	1364.50	Silver	Mountain Frames	Components	2.84	46	1
248	743	HL Mountain Frame ...	1349.60	Black	Mountain Frames	Components	2.72	42	1
249	744	HL Mountain Frame ...	1349.60	Black	Mountain Frames	Components	2.76	44	1
250	745	HL Mountain Frame ...	1349.60	Black	Mountain Frames	Components	2.80	48	1

Dim_Salesperson:

SalesPersonID	FirstName	LastName	Title	Gender	CountryRegionCode	Group
274	Stephen	Jiang	NULL	M	NULL	NULL
275	Michael	Blythe	NULL	M	US	North America
276	Linda	Mitchell	NULL	F	US	North America
277	Jillian	Carson	NULL	F	US	North America
278	Garrett	Vargas	NULL	M	CA	North America
279	Tsvi	Reiter	NULL	M	US	North America
280	Pamela	Ansman-Wolfe	NULL	F	US	North America
281	Shu	Ito	NULL	M	US	North America
282	José	Saraiva	NULL	M	CA	North America
283	David	Campbell	NULL	M	US	North America
284	Tete	Mensa-Annan	Mr.	M	US	North America
285	Syed	Abbas	Mr.	M	NULL	NULL
286	Lynn	Tsoflias	NULL	F	AU	Pacific
287	Amy	Alberts	NULL	F	NULL	NULL
288	Rachel	Valdez	NULL	F	DE	Europe

Fact_Sales:

	ProductID	CustomerID	SalesPersonID	OrderDate	ShipDate	OrderQty	UnitPrice	UnitPriceDiscount	LineTotal
1	776	29825	279	20110531	20110607	1	2024.994	0.00	2024.994000
2	777	29825	279	20110531	20110607	3	2024.994	0.00	6074.982000
3	778	29825	279	20110531	20110607	1	2024.994	0.00	2024.994000
4	771	29825	279	20110531	20110607	1	2039.994	0.00	2039.994000
5	772	29825	279	20110531	20110607	1	2039.994	0.00	2039.994000
6	773	29825	279	20110531	20110607	2	2039.994	0.00	4079.988000
7	774	29825	279	20110531	20110607	1	2039.994	0.00	2039.994000
8	714	29825	279	20110531	20110607	3	28.8404	0.00	86.521200
9	716	29825	279	20110531	20110607	1	28.8404	0.00	28.840400
10	709	29825	279	20110531	20110607	6	5.70	0.00	34.200000
11	712	29825	279	20110531	20110607	2	5.1865	0.00	10.373000
12	711	29825	279	20110531	20110607	4	20.1865	0.00	80.746000
13	762	29672	279	20110531	20110607	1	419.4589	0.00	419.458900
14	758	29672	279	20110531	20110607	1	874.794	0.00	874.794000
15	745	29734	282	20110531	20110607	1	809.76	0.00	809.760000

W wynikach pojawia się wiele wartości NULL. W części wynika to z braku danych dla tych atrybutów w odpowiednich tabelach, częściej jest to spowodowane łączeniami zewnętrznymi.

Zad. 4. Więzy integralności

1. Dodać integralność referencyjną i klucze główne do tabel już zdefiniowanych.
2. Przygotować instrukcję INSERT INTO, która sprawdzi poprawność integralności referencyjnej oraz klucze główne

```
ALTER TABLE [Rasz].[Dim_Customer]  
ADD CONSTRAINT PK_Dim_Customer_CustomerID PRIMARY KEY ([CustomerID]);
```

```
ALTER TABLE [Rasz].[Dim_Product]  
ADD CONSTRAINT PK_Dim_Product_ProductID PRIMARY KEY ([ProductID]);
```

```
ALTER TABLE [Rasz].[Dim_Salesperson]  
ADD CONSTRAINT PK_Dim_Salesperson PRIMARY KEY ([SalesPersonID]);
```

```
ALTER TABLE [Rasz].[Fact_Sales]  
ADD CONSTRAINT FK_Fact_Sales_Dim_Product_ProductID  
FOREIGN KEY ([ProductID])  
REFERENCES [Rasz].[Dim_Product]([ProductID])
```

```
ALTER TABLE [Rasz].[Fact_Sales]  
ADD CONSTRAINT FK_Fact_Sales_Dim_Customer_CustomerID  
FOREIGN KEY ([CustomerID])  
REFERENCES [Rasz].[Dim_Customer]([CustomerID])
```

```
ALTER TABLE [Rasz].[Fact_Sales]  
ADD CONSTRAINT FK_Fact_Sales_Dim_Salesperson_SalesPersonID  
FOREIGN KEY ([SalesPersonID])  
REFERENCES [Rasz].[Dim_Salesperson]([SalesPersonID])
```

Dodanie kluczy obcych powoduje błąd:

The CREATE UNIQUE INDEX statement terminated because a duplicate key was found for the object name 'Rasz.Dim_Customer' and the index name 'PK_Dim_Customer_CustomerID'. The duplicate key value is (11279).

Jak się okazuje, część klientów ma podane w bazie danych 2 adresy – domowy oraz dostawy.

CustomerID	FirstName	LastName	Title	City	TerritoryName	CountryRegionCode	Group
11279	Amanda	Cook	NULL	NULL	Northwest	NULL	North America
11279	Amanda	Cook	NULL	Everett	Northwest	US	North America
18689	Miguel	Miller	NULL	NULL	Northwest	NULL	North America
18689	Miguel	Miller	NULL	Everett	Northwest	US	North America
18083	Osarumwense	Agbonile	NULL	NULL	Northwest	NULL	North America
18083	Osarumwense	Agbonile	NULL	Issaquah	Northwest	US	North America
20281	Karl	Xie	NULL	NULL	Northwest	NULL	North America
20281	Karl	Xie	NULL	Ballard	Northwest	US	North America
15841	Luis	Shan	NULL	NULL	Southwest	NULL	North America

Przypadków tych jest 24, każdy z nich ma po jednym z rekordów posiadających miasto oraz bez miasta. Dane te wyczyszczono zapytaniem:

```
delete from Rasz.Dim_Customer
where City is null and CustomerID in (
    select CustomerID
    from Rasz.Dim_Customer
    group by CustomerID
    having COUNT(*) > 1
);
```

Po usunięciu duplikatów, wszystkie polecenia wykonują się.

Sprawdzanie więzów integralności

Poniższe komendy wykonywane są na wyczyszczonych tabelach.

Klucze główne:

```
INSERT INTO Rasz.Dim_Customer VALUES
```

```
(1, 'Thomas', 'Sanchez', 'Mr.', 'Colma', 'Southwest', 'US', 'North America'),
(1, 'Noah', 'Lee', 'Mr.', 'Concord', 'Southwest', 'US', 'North America');
```

Violation of PRIMARY KEY constraint 'PK_Dim_Customer_CustomerID'. Cannot insert duplicate key in object 'Rasz.Dim_Customer'. The duplicate key value is (1).

```
INSERT INTO Rasz.Dim_Product VALUES
```

```
(2, 'Blade', 2.00, 'Black', 'Road Frames', 'Components', 2.12, 66, 1),
(2, 'Blade1', 1.00, 'Black', 'Road Frames', 'Components', 2.13, 67, 1);
```

Violation of PRIMARY KEY constraint 'PK_Dim_Product_ProductID'. Cannot insert duplicate key in object 'Rasz.Dim_Product'. The duplicate key value is (2).

```
INSERT INTO Rasz.Dim_Salesperson VALUES
```

```
(3, 'Stephen', 'Jiang', 'Mr.', 'M', 'US', 'North America'),
(3, 'Amy', 'Alberts', 'Ms.', 'F', 'US', 'North America');
```

Violation of PRIMARY KEY constraint 'PK_Dim_Salesperson'. Cannot insert duplicate key in object 'Rasz.Dim_Salesperson'. The duplicate key value is (3).

Klucze obce:

Tabele wymiarów wypełniono pojedynczymi wierszami:

```
INSERT INTO Rasz.Dim_Product VALUES
(1, 'Blade', 2.00, 'Black', 'Road Frames', 'Components', 2.12, 66, 1);
INSERT INTO Rasz.Dim_Customer VALUES
(2, 'Thomas', 'Sanchez', 'Mr.', 'Colma', 'Southwest', 'US', 'North America');
INSERT INTO Rasz.Dim_Salesperson VALUES
(3, 'Stephen', 'Jiang', 'Mr.', 'M', 'US', 'North America');
```

Wstawianie poprawnego rekordu:

```
INSERT INTO Rasz.Fact_Sales VALUES
(1, 2, 3, 20110531, 20110607, 1, 2024.994, 0.00, 2024.994);

(1 row affected)
```

Niepoprawny identyfikator ProductID:

```
INSERT INTO Rasz.Fact_Sales VALUES
(100, 2, 3, 20110531, 20110607, 1, 2024.994, 0.00, 2024.994);
```

The INSERT statement conflicted with the FOREIGN KEY constraint "FK_Fact_Sales_Dim_Product_ProductID". The conflict occurred in database "AdventureWorksOLTP", table "Rasz.Dim_Product", column 'ProductID'.

Niepoprawny identyfikator CustomerID:

```
INSERT INTO Rasz.Fact_Sales VALUES
(1, 100, 3, 20110531, 20110607, 1, 2024.994, 0.00, 2024.994);
```

The INSERT statement conflicted with the FOREIGN KEY constraint "FK_Fact_Sales_Dim_Customer_CustomerID". The conflict occurred in database "AdventureWorksOLTP", table "Rasz.Dim_Customer", column 'CustomerID'.

Niepoprawny identyfikator SalespersonID:

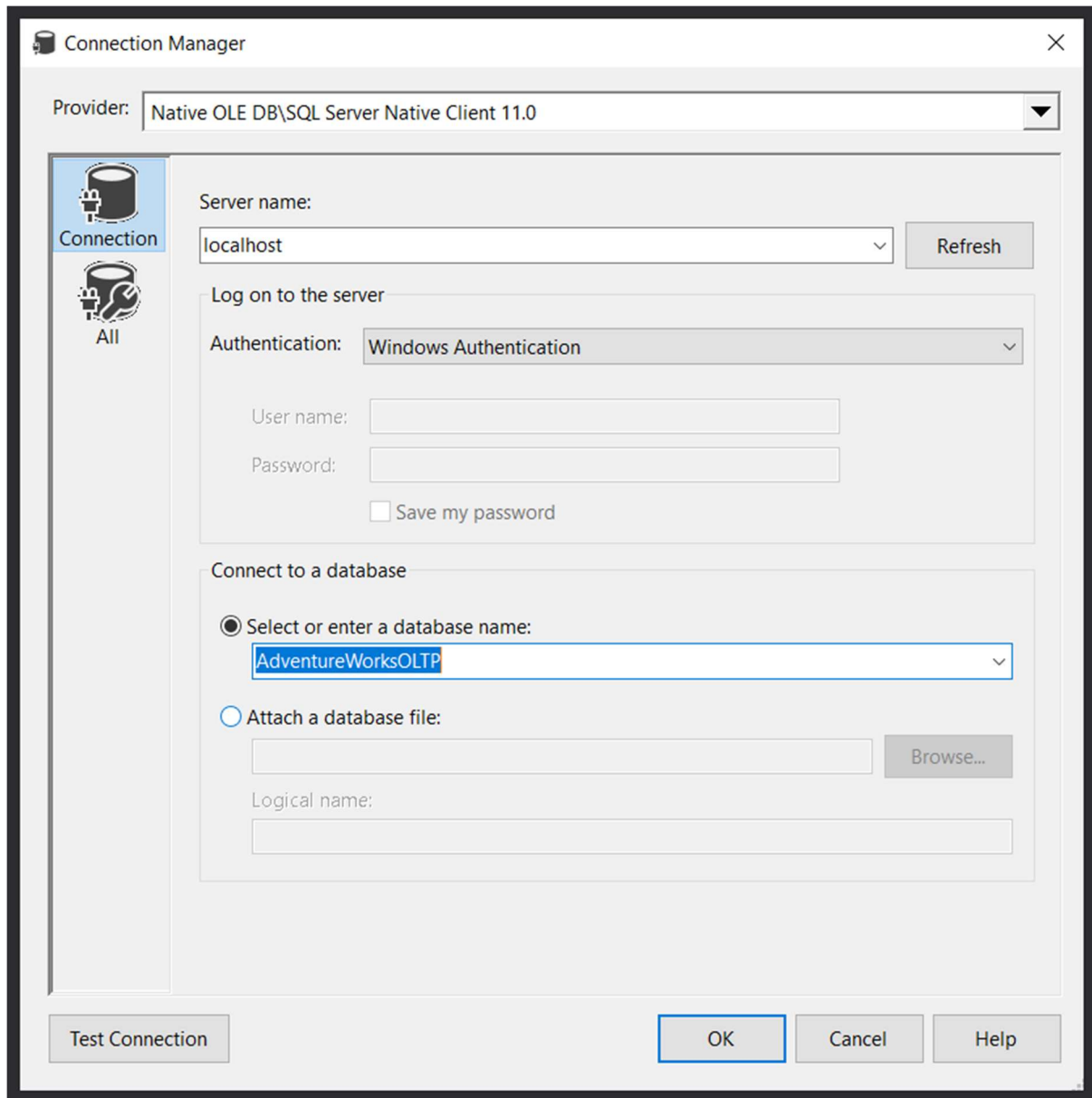
```
INSERT INTO Rasz.Fact_Sales VALUES
(1, 2, 100, 20110531, 20110607, 1, 2024.994, 0.00, 2024.994);
```

The INSERT statement conflicted with the FOREIGN KEY constraint "FK_Fact_Sales_Dim_Salesperson_SalesPersonID". The conflict occurred in database "AdventureWorksOLTP", table "Rasz.Dim_Salesperson", column 'SalesPersonID'.

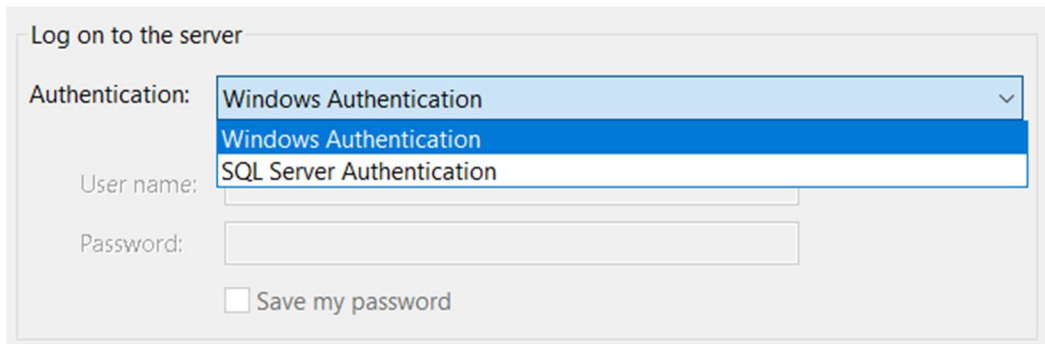
Zad. 5. Tworzenie kostki

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

Do bazy danych podłączono się za pomocą SQL Server Native Client. Co ciekawe, serwer nie był automatycznie znaleziony, adres *localhost* musiał zostać wpisany manualnie.



Do uwierzytelnienia użytkownika dostępne są dwie opcje: Windows Authentication oraz SQL Server Authentication. Z powodu braku konfiguracji bezpieczeństwa w lokalnym serwerze, do połączenia użyte jest Windows Authentication z kontem administratora.



Log on to the server

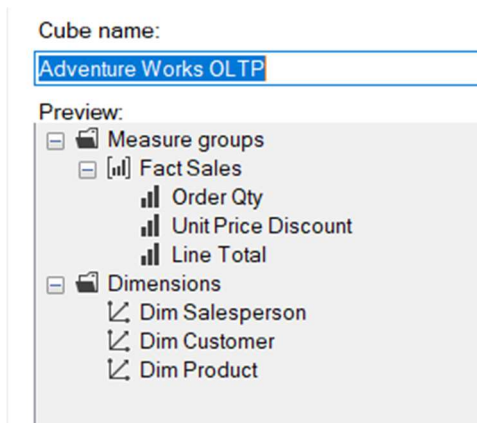
Authentication: Windows Authentication

User name:

Password:

☐ Save my password

Po połączeniu się z bazą danych i wybraniu odpowiednich wymiarów i miar z dostępnych tabel, możemy utworzyć kostkę. Podsumowanie:



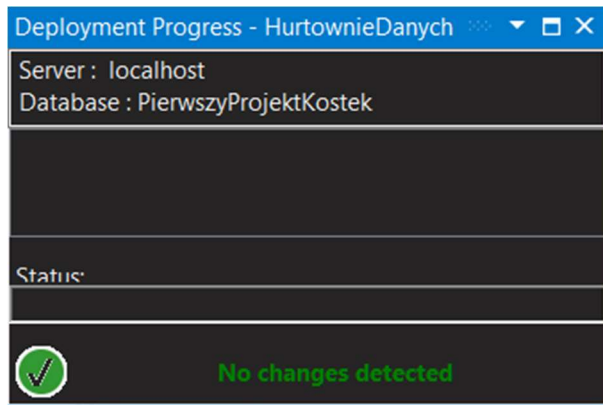
Cube name:

Preview:

- [-] Measure groups
 - [+] Fact Sales
 - Order Qty
 - Unit Price Discount
 - Line Total
- [-] Dimensions
 - Dim Salesperson
 - Dim Customer
 - Dim Product

Zad. 6. Uruchomienie kostki

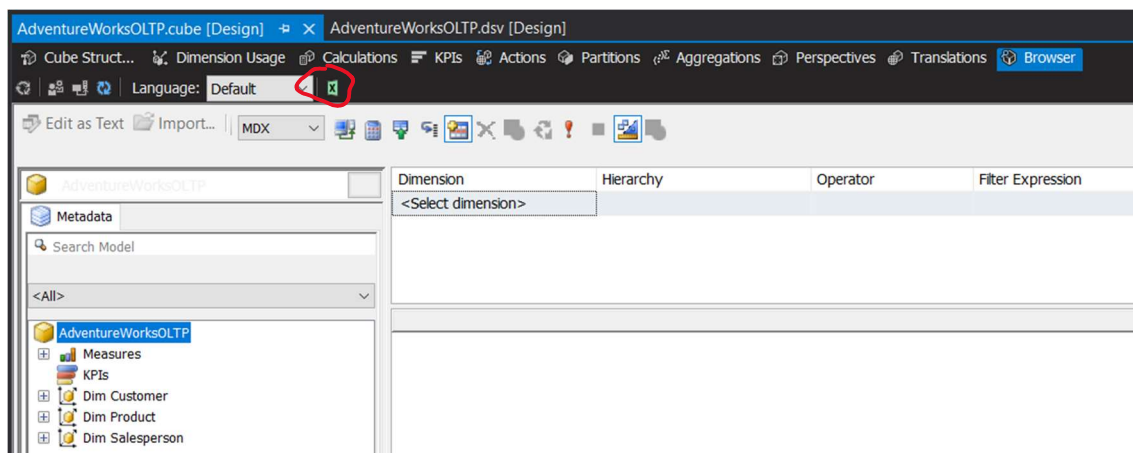
Po utworzeniu kostki i poprawnej instalacji wszystkich komponentów, możemy uruchomić kostkę. Wynik uruchomienia za każdym kolejnym razem, kiedy w bazie danych nie wykryto zmian w tabelach wymiarów oraz faktów:



Zad. 7. Proste raporty

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

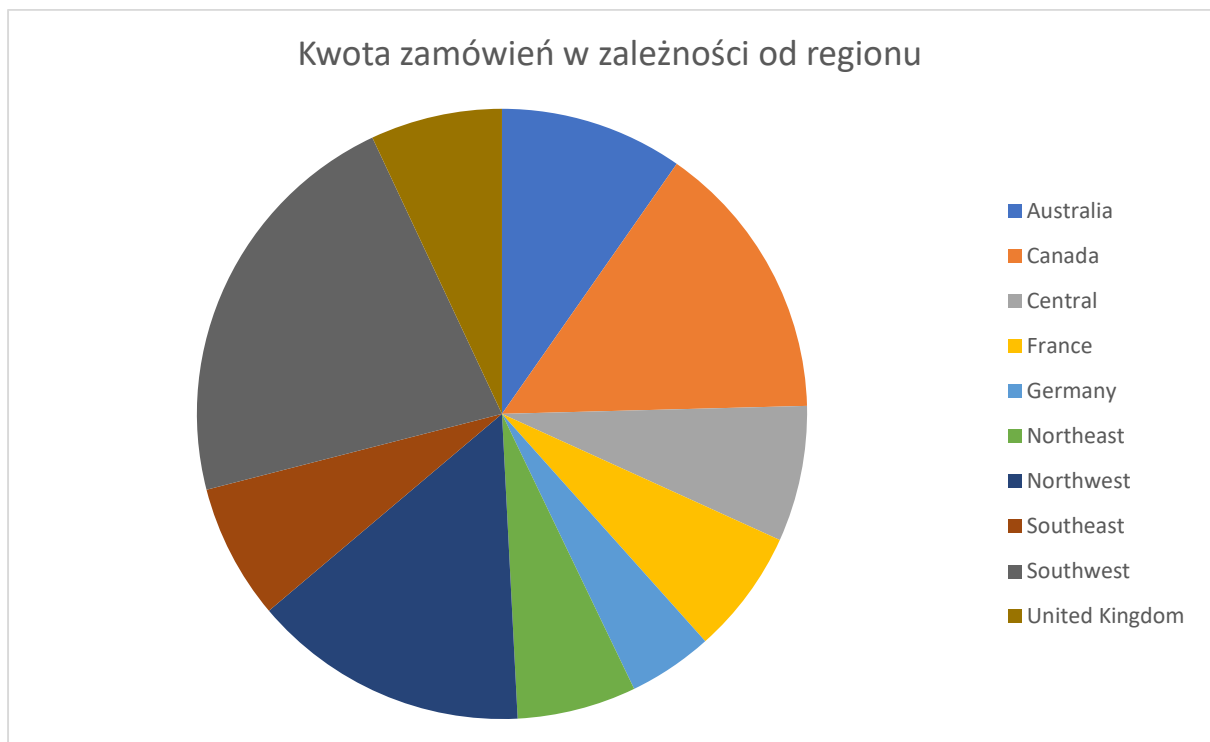
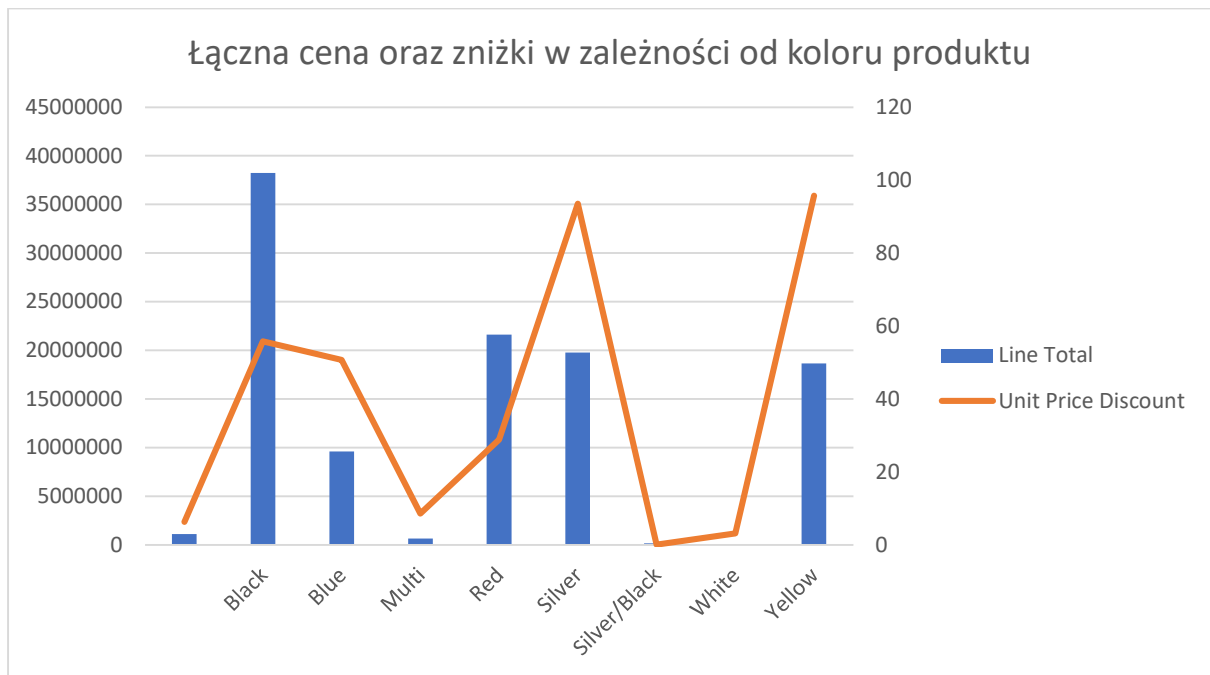
Po uruchomieniu kostki, możemy ją przeglądać i tworzyć raporty w Visual Studio. Dane można przekierować do Excel wybierając ikonę jak na rysunku niżej:



Liczba zamówień produktów z każdej kategorii w zależności od ich koloru:

Order Qty									
Row Labels	Black	Blue	Multi	Red	Silver	Silver/Black	White	Yellow	Grand Total
Accessories	6532	6743		6266	2761				61932
Bikes	32396	8456		18088	13271			18057	90268
Clothing	27568	6738	25073				5217	9074	73670
Components	15441	1722		4875	8991	3931		5425	49044
Grand Total	81937	23659	25073	29229	25023	3931	5217	32556	274914





Podsumowanie i wnioski

Utworzenie kostki zdecydowanie ułatwia analizę danych. Praca na przetworzonych danych jest łatwiejsza oraz szybsza, niż analiza wyników zapytań SQL. Integracja z zewnętrznymi narzędziami, jak Excel, pozwala na wiele opcji przetwarzania danych.

Nie wszystkie atrybuty tabel faktów mogą być użyte jako miary. W przypadku tabeli użytej w zadaniu, analiza identyfikatorów czy dat w postaci liczbowej nie ma sensu z perspektywy biznesowej.