

# 4. sprawozdanie z laboratorium Hurtownie Danych

Mikołaj Kubś, 272662

10 kwietnia 2025

1. DDL (Data Definition Language) - tworzenie struktur danych i schema z użyciem poleceń CREATE, ALTER, DROP
2. DML (Data Manipulation Language) - manipulacja danymi w tabelach z użyciem poleceń INSERT, UPDATE, DELETE
3. DCL (Data Control Language) - zarządzanie uprawnieniami, za pomocą poleceń GRANT, DENY, REVOKE
4. DQL (Data Query Language) - pozyskiwanie danych z bazy, za pomocą polecenia SELECT

## 1 Zadanie 1 - przygotowanie schematu

```
1 CREATE SCHEMA Kubs
```

Utworzenie dedykowanego schematu pozwala na logiczne odseparowanie obiektów stworzonych na potrzeby laboratorium od pozostałych struktur bazy danych.

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

```
1  
2 CREATE TABLE Kubs.DIM_CUSTOMER (  
3     CustomerID INT NOT NULL,  
4     FirstName NVARCHAR(50) NULL,  
5     LastName NVARCHAR(50) NULL,
```

```

6      Title NVARCHAR(8) NULL,
7      City NVARCHAR(30) NULL,
8      TerritoryName NVARCHAR(50) NULL,
9      CountryRegionCode NVARCHAR(3) NULL,
10     [Group] NVARCHAR(50) NULL
11 );
12
13 CREATE TABLE Kubs.DIM_PRODUCT (
14     ProductID INT NOT NULL,
15     Name NVARCHAR(50) NOT NULL,
16     ListPrice MONEY NULL,
17     Color NVARCHAR(15) NULL,
18     SubCategoryName NVARCHAR(50) NULL,
19     CategoryName NVARCHAR(50) NULL,
20     Weight DECIMAL(8, 2) NULL,
21     Size NVARCHAR(5) NULL,
22     IsPurchased BIT NULL DEFAULT 0
23 );
24
25 CREATE TABLE Kubs.DIM SALESPERSON (
26     SalesPersonID INT NOT NULL,
27     FirstName NVARCHAR(50) NULL,
28     LastName NVARCHAR(50) NULL,
29     Title NVARCHAR(8) NULL,
30     Gender NCHAR(1) NULL,
31     CountryRegionCode NVARCHAR(3) NULL,
32     [Group] NVARCHAR(50) NULL
33 );
34
35 CREATE TABLE Kubs.FACT_SALES (
36     ProductID INT NOT NULL,
37     CustomerID INT NOT NULL,
38     SalesPersonID INT NULL,
39     OrderDate INT NOT NULL,
40     ShipDate INT NULL,
41     OrderQty SMALLINT NOT NULL,
42     UnitPrice MONEY NOT NULL,
43     UnitPriceDiscount DECIMAL(8, 4) NOT NULL DEFAULT 0,
44     LineTotal DECIMAL(19, 4) NOT NULL
45 );

```

Zgodnie z poleceniem, kolumny OrderDate oraz ShipDate będą przechowywać dane typu całkowitego. W tabeli faktów sprzedawca nie jest wymagany. W tabeli wymiaru dla produktu kategoria i podkategoria również mogą

być NULL. Tabele DIM\_ pełnią oczywiście role wymiarów, a FACT\_SALES jest tabelą faktów zawierającą miary dotyczące transakcji sprzedaży.

Wartości NULL w niektórych kolumnach wynikają z opcjonalności tych danych w oryginalnym źródle danych lub mogą być nieobecne (również z powodu braku połączenia z inną tabelą, np. produkt bez podkategorii nie może mieć kategorii).

### 3 Zadanie 3 - wypełnianie danych

```
1  INSERT INTO Kubs.DIM_CUSTOMER (
2      CustomerID,
3      FirstName,
4      LastName,
5      Title,
6      City,
7      TerritoryName,
8      CountryRegionCode,
9      [Group]
10 )
11 SELECT DISTINCT
12     c.CustomerID,
13     p.FirstName,
14     p.LastName,
15     p.Title,
16     a.City,
17     st.Name AS TerritoryName,
18     st.CountryRegionCode,
19     st.[Group]
20 FROM Sales.Customer AS c
21 LEFT JOIN Person.Person AS p ON c.PersonID = p.BusinessEntityID
22 LEFT JOIN Sales.SalesTerritory AS st
23     ON c.TerritoryID = st.TerritoryID
24 LEFT JOIN Person.BusinessEntityAddress bea
25     ON p.BusinessEntityID = bea.BusinessEntityID
26 LEFT JOIN Person.Address AS a ON bea.AddressID = a.AddressID
27 WHERE c.PersonID IS NOT NULL;
28
29 INSERT INTO Kubs.DIM_PRODUCT (
30     ProductID,
31     Name,
32     ListPrice,
33     Color,
```

```

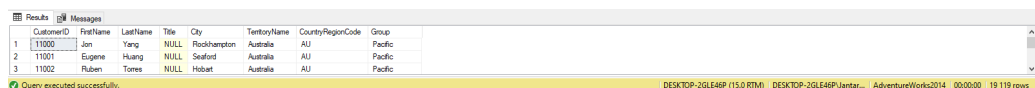
34     SubCategoryName,
35     CategoryName,
36     Weight,
37     Size,
38     IsPurchased
39 )
40 SELECT DISTINCT
41     p.ProductID,
42     p.Name,
43     p.ListPrice,
44     p.Color,
45     psc.Name AS SubCategoryName,
46     pc.Name AS CategoryName,
47     p.Weight,
48     p.Size,
49     1 AS IsPurchased
50 FROM Production.Product AS p
51 INNER JOIN Sales.SalesOrderDetail AS sod
52     ON p.ProductID = sod.ProductID
53 LEFT JOIN Production.ProductSubcategory AS psc
54     ON p.ProductSubcategoryID = psc.ProductSubcategoryID
55 LEFT JOIN Production.ProductCategory AS pc
56     ON psc.ProductCategoryID = pc.ProductCategoryID;
57
58 INSERT INTO Kubs.DIM_SALESPERSON (
59     SalesPersonID,
60     FirstName,
61     LastName,
62     Title,
63     Gender,
64     CountryRegionCode,
65     [Group]
66 )
67 SELECT
68     sp.BusinessEntityID AS SalesPersonID,
69     p.FirstName,
70     p.LastName,
71     p.Title,
72     e.Gender,
73     st.CountryRegionCode,
74     st.[Group]
75 FROM Sales.SalesPerson AS sp
76 INNER JOIN Person.Person AS p

```

```

77     ON sp.BusinessEntityID = p.BusinessEntityID
78 INNER JOIN HumanResources.Employee AS e
79     ON sp.BusinessEntityID = e.BusinessEntityID
80 LEFT JOIN Sales.SalesTerritory AS st
81     ON sp.TerritoryID = st.TerritoryID;
82
83 INSERT INTO Kubs.FACT_SALES (
84     ProductID,
85     CustomerID,
86     SalesPersonID,
87     OrderDate,
88     ShipDate,
89     OrderQty,
90     UnitPrice,
91     UnitPriceDiscount,
92     LineTotal
93 )
94 SELECT
95     sod.ProductID,
96     soh.CustomerID,
97     soh.SalesPersonID,
98     DATEPART(YEAR, soh.OrderDate) * 10000 +
99     DATEPART(MONTH, soh.OrderDate) * 100 +
100    DATEPART(DAY, soh.OrderDate) AS OrderDate,
101    DATEPART(YEAR, soh.ShipDate) * 10000 +
102    DATEPART(MONTH, soh.ShipDate) * 100 +
103    DATEPART(DAY, soh.ShipDate) AS ShipDate,
104    sod.OrderQty,
105    sod.UnitPrice,
106    sod.UnitPriceDiscount,
107    sod.LineTotal
108 FROM Sales.SalesOrderDetail AS sod
109 INNER JOIN Sales.SalesOrderHeader AS soh
110     ON sod.SalesOrderID = soh.SalesOrderID;

```



	CustomerID	FirstName	LastName	Title	City	TerritoryName	CountryRegionCode	Group
1	11000	Ken	Yang	NULL	Roshampton	Australia	AU	Pacific
2	11001	Eugene	Huang	NULL	Seaford	Australia	AU	Pacific
3	11002	Ruben	Torres	NULL	Hobart	Australia	AU	Pacific

Rysunek 1: Pierwsze 3 wiersze DIM\_CUSTOMER i liczba wierszy - 19119

ProductID	Name	ListPrice	Color	SubCategoryName	CategoryName	Weight	Size	IsPurchased
984	Mountain-500 Silver, 40	564.99	Silver	Mountain Bikes	Bikes	27.35	40	1
722	LL Road Frame - Black, 58	337.22	Black	Road Frames	Components	2.46	58	1
864	Classic Vest, S	63.90	Blue	Vests	Clothing	NULL	S	1

Rysunek 2: Pierwsze 3 wiersze DIM\_PRODUCT i liczba wierszy - 266

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

Rysunek 3: Pierwsze 3 wiersze DIM\_SALESPERSON i liczba wierszy - 17

ProductID	CustomerID	SalesPersonID	OrderDate	ShipDate	OrderQty	UnitPrice	UnitPriceDiscount	LineTotal
870	29485	276	20140129	20140205	5	2.994	0.0000	14.9700
864	29485	276	20140129	20140205	4	38.10	0.0000	152.4000
370	29485	276	20140129	20140205	3	729.91	0.0000	2189.7300

Rysunek 4: Pierwsze 3 wiersze FACT\_SALES i liczba wierszy - 121317

Zgodnie z poleceniem wypełniono tabele danymi z bazy danych AdventureWorks2014. Ponieważ analizy będą dotyczyć sprzedaży produktów, pominięto wszystkie produkty, które nigdy nie zostały kupione. Okazało się, że wszystkie produkty, które nie miały kategorii lub podkategorii, nigdy nie były kupione.

Liczby wierszy:

DIM\_CUSTOMER - 19119

DIM\_PRODUCT - 266

DIM\_SALESPERSON - 17

FACT\_SALES - 121317

## 4 Zadanie 4 - Wieży integralności

### 4.1 Definiowanie kluczy głównych i obcych

Poniższy kod SQL dodaje klucze główne do tabel wymiarów oraz klucze obce do tabeli faktów, łącząc ją z wymiarami.

```

1 ALTER TABLE Kubs.DIM_CUSTOMER
2 ADD CONSTRAINT PK_DIM_CUSTOMER PRIMARY KEY (CustomerID);
3
4 ALTER TABLE Kubs.DIM_PRODUCT
5 ADD CONSTRAINT PK_DIM_PRODUCT PRIMARY KEY (ProductID);
6
7 ALTER TABLE Kubs.DIM SALESPERSON
8 ADD CONSTRAINT PK_DIM SALESPERSON PRIMARY KEY (SalesPersonID);
9
10 ALTER TABLE Kubs.FACT_SALES
11 ADD CONSTRAINT FK_FACT_SALES_DIM_CUSTOMER
12 FOREIGN KEY (CustomerID) REFERENCES Kubs.DIM_CUSTOMER(CustomerID);
13
14 ALTER TABLE Kubs.FACT_SALES
15 ADD CONSTRAINT FK_FACT_SALES_DIM_PRODUCT
16 FOREIGN KEY (ProductID) REFERENCES Kubs.DIM_PRODUCT(ProductID);
17
18 ALTER TABLE Kubs.FACT_SALES
19 ADD CONSTRAINT FK_FACT_SALES_DIM SALESPERSON
20 FOREIGN KEY (SalesPersonID)
21 REFERENCES Kubs.DIM SALESPERSON(SalesPersonID);

```

## 4.2 Testowanie więzów integralności

Poniższe instrukcje INSERT INTO mają na celu sprawdzenie działania zdefiniowanych kluczy głównych i obcych. Oczekujemy, że próby wstawienia niepoprawnych danych zakończą się błędami przechwyconymi w blokach CATCH.

```

1 PRINT 'Test 1: Próba naruszenia PK w DIM_CUSTOMER';
2 BEGIN TRY
3     INSERT INTO Kubs.DIM_CUSTOMER (CustomerID, FirstName, LastName)
4     VALUES (11000, 'Test', 'DuplicatePK');
5
6     PRINT 'BŁĄD - duplikat PK';
7 END TRY
8 BEGIN CATCH
9     PRINT 'SUKCES';
10    PRINT ERROR_MESSAGE();
11 END CATCH
12 GO
13
14 PRINT 'Test 2: Próba naruszenia FK (nieistniejący ProductID) w FACT_SALES';
15 BEGIN TRY

```

```

16     INSERT INTO Kubs.FACT_SALES (
17         ProductID, CustomerID, SalesPersonID, OrderDate, ShipDate,
18         OrderQty, UnitPrice, UnitPriceDiscount, LineTotal
19     ) VALUES (
20         -999,
21         11000,
22         NULL,
23         20250101, 20250102, 1, 10.0, 0, 10.0
24     );
25
26     PRINT 'BŁĄD: Nieistniejące ProductID';
27 END TRY
28 BEGIN CATCH
29     PRINT 'SUKCES';
30     PRINT ERROR_MESSAGE();
31 END CATCH
32 GO
33
34 PRINT 'Test 3: Próba naruszenia FK (nieistniejący CustomerID) w FACT_SALES';
35 BEGIN TRY
36     INSERT INTO Kubs.FACT_SALES (
37         ProductID, CustomerID, SalesPersonID, OrderDate, ShipDate,
38         OrderQty, UnitPrice, UnitPriceDiscount, LineTotal
39     ) VALUES (
40         776,
41         -999,
42         NULL,
43         20240103, 20240104, 2, 20.0, 0, 40.0
44     );
45
46     PRINT 'BŁĄD: nieistniejący CustomerID';
47 END TRY
48 BEGIN CATCH
49     PRINT 'SUKCES';
50     PRINT ERROR_MESSAGE();
51 END CATCH
52 GO
53
54 PRINT 'Koniec testów'

```



```
Messages
Test 1: Próba naruszenia PK w DIM_CUSTOMER
(0 rows affected)
SUCCESS
Violation of PRIMARY KEY constraint 'PK_DIM_CUSTOMER'. Cannot insert duplicate key in object 'Kuba.DIM_CUSTOMER'. The duplicate key value is (11000).
Test 2: Próba naruszenia FK (nieistniejący ProductID) w FACT_SALES
(0 rows affected)
SUCCESS
The INSERT statement conflicted with the FOREIGN KEY constraint "FK_FACT_SALES_DIM_PRODUCT". The conflict occurred in database "AdventureWorks2014", table "Kuba.DIM_PRODUCT", column "ProductID".
Test 3: Próba naruszenia FK (nieistniejący CustomerID) w FACT_SALES
(0 rows affected)
SUCCESS
The INSERT statement conflicted with the FOREIGN KEY constraint "FK_FACT_SALES_DIM_CUSTOMER". The conflict occurred in database "AdventureWorks2014", table "Kuba.DIM_CUSTOMER", column "CustomerID".
Koniec testów
```

## Rysunek 5: Wykonanie testów więzów integralności

Testowe instrukcje zakończyły się przechwyceniem błędów związanych z naruszeniem więzów integralności.