

Hurtownie danych Laboratorium Czw 11:15

Lista 1

Kajetan Pynka 254495

Zad 1.1

Reg/05 - Produkt może być sprzedawany w wielu sklepach

Reg/06 - Zakup może dotyczyć wielu produktów

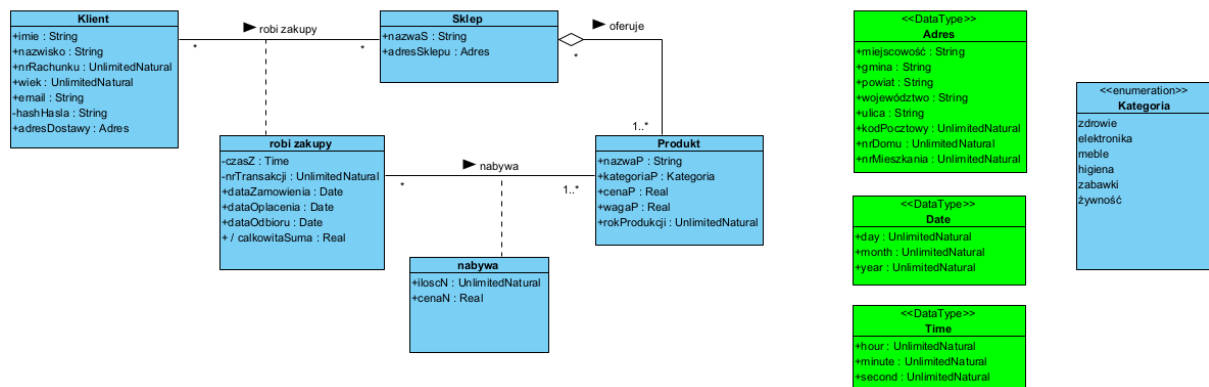
Reg/07 - Zakup musi dotyczyć przynajmniej jednego produktu

Reg/08 - Zakup każdego produktu obarczony jest nabyciem określonej ilości danego produktu za konkretną cenę

Reg/09 - Produkt może być kupowany wiele razy

Reg/10 – Klient może robić zakupy w wielu sklepach

Zad 1.2



Zad 1.3

CREATE TABLE Adres (

idA INTEGER PRIMARY KEY,
miejscowosc VARCHAR(50) NOT NULL,
gmina VARCHAR(50) NOT NULL,
powiat VARCHAR(50) NOT NULL,
województwo VARCHAR(50) NOT NULL,
ulica VARCHAR(50) NOT NULL,
kodPocztowy INTEGER NOT NULL,
nrDomu INTEGER NOT NULL,
nrMieszkania INTEGER

);

CREATE TABLE Klient (

idK INTEGER PRIMARY KEY,
imie VARCHAR(50) NOT NULL,
nazwisko VARCHAR(50) NOT NULL,
nrRachunku VARCHAR(20) UNIQUE,
wiek INTEGER,
email VARCHAR(30) NOT NULL UNIQUE,
hashHasla VARCHAR(256) NOT NULL UNIQUE,
adresDostawy INTEGER FOREIGN KEY REFERENCES Adres(idA) UNIQUE

);

CREATE TABLE Sklep (

idS INTEGER PRIMARY KEY,
nazwaS VARCHAR(30) NOT NULL,
adresSklepu INTEGER FOREIGN KEY REFERENCES Adres(idA) NOT NULL UNIQUE

);

```
CREATE TABLE Kategoria (  
    nazwaKat VARCHAR(30) PRIMARY KEY  
);
```

```
CREATE TABLE Produkt(  
    idP INTEGER PRIMARY KEY,  
    nazwaP VARCHAR(50) NOT NULL,  
    kategoriaP VARCHAR(30) REFERENCES Kategoria(nazwaKat) NOT NULL,  
    cenaP DECIMAL(30,4) NOT NULL,  
    wagaP REAL,  
    rokProdukcji INTEGER  
);
```

```
CREATE TABLE ProduktSklepowy(  
    idPS INTEGER PRIMARY KEY,  
    idSklepu INTEGER REFERENCES Sklep(idS) NOT NULL,  
    idProduktu INTEGER REFERENCES Produkt(idP) NOT NULL  
);
```

```
CREATE TABLE RobiZakupy (  
    nrTransakcji INTEGER PRIMARY KEY,  
    idKlienta INTEGER REFERENCES Klient(idK) NOT NULL,  
    idSklepu INTEGER REFERENCES Sklep(idS) NOT NULL,  
    czasZ TIME NOT NULL,  
    dataZamowienia DATE NOT NULL,  
    dataOplacenia DATE,  
    dataOdbioru DATE,  
    calkowitaSuma DECIMAL(30,4) NOT NULL  
);
```

```

CREATE TABLE Nabywa (

    idN INTEGER PRIMARY KEY,

    idZakupu INTEGER REFERENCES RobiZakupy(nrTransakcji) NOT NULL,

    idProd INTEGER REFERENCES Produkt(idP) NOT NULL,

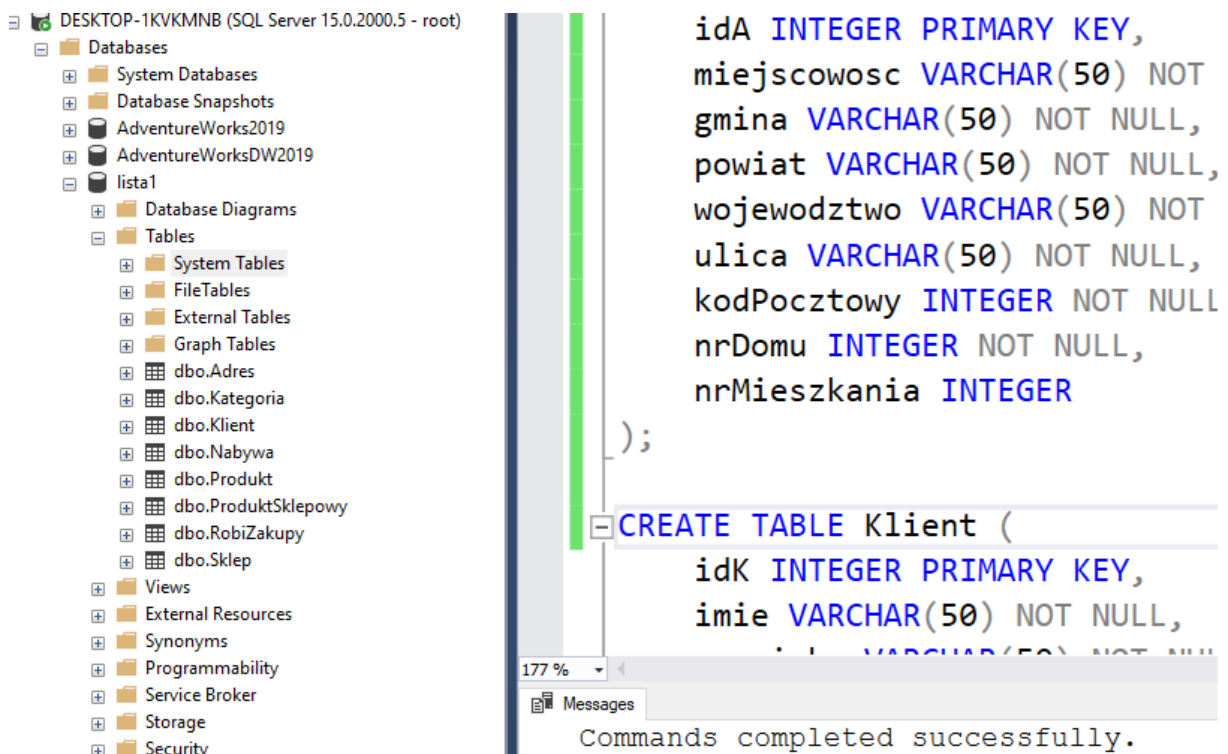
    iloscN INTEGER NOT NULL,

    cenaN DECIMAL(30,4) NOT NULL

);

```

Zad 1.4



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Databases' folder is expanded, showing the 'lista1' database. Under 'Tables', the following tables are listed: System Tables, FileTables, External Tables, Graph Tables, dbo.Adres, dbo.Kategoria, dbo.Klient, dbo.Nabywa, dbo.Produkt, dbo.ProduktSklepowy, dbo.RobiZakupy, and dbo.Sklep. On the right, a query window is open, displaying the following SQL code:

```

idA INTEGER PRIMARY KEY,
miestowosc VARCHAR(50) NOT
gmina VARCHAR(50) NOT NULL,
powiat VARCHAR(50) NOT NULL,
województwo VARCHAR(50) NOT
ulica VARCHAR(50) NOT NULL,
kodPocztowy INTEGER NOT NULL
nrDomu INTEGER NOT NULL,
nrMieszkania INTEGER

);

CREATE TABLE Klient (
    idK INTEGER PRIMARY KEY,
    imie VARCHAR(50) NOT NULL,
    ...

```

At the bottom of the query window, a status bar indicates 'Commands completed successfully.'

Zad 1.5

```

INSERT INTO Adres VALUES(1, 'Wrocław', 'Wrocław', 'wrocławski', 'dolnośląskie', 'Kardynała Stefana
Wyszyńskiego', 50332, 19, NULL);

```

```

INSERT INTO Kategoria VALUES('zdrowie');

```

```

INSERT INTO Kategoria VALUES('elektronika');

```

```

INSERT INTO Kategoria VALUES('meble');

```

```

INSERT INTO Kategoria VALUES('zabawki');

```

```
INSERT INTO Klient VALUES(1, 'Jan', 'Brzechwa', NULL, 23, 'jbrze@onet.pl', 'F$AFJGA$GAG!j32i4b', NULL);
```

```
INSERT INTO Sklep VALUES(1, 'H&M', 1);
```

```
INSERT INTO Produkt VALUES(1, 'lalka Barbie', 'zabawki', 19.95, 50, 2017);
```

```
INSERT INTO ProduktSklepowy VALUES(1, 1, 1);
```

```
INSERT INTO RobiZakupy VALUES(1, 1, 1, '19:43', '2022-01-20', NULL, NULL, 79.8);
```

```
INSERT INTO Nabywa VALUES(1, 1, 1, 4, 19.95);
```

Zad 2.1

```
SELECT COUNT(*) 'Liczba produktów' FROM Production.Product;
```

```
SELECT COUNT(*) 'Liczba kategorii' FROM Production.ProductCategory;
```

```
SELECT COUNT(*) 'Liczba podkategorii' FROM Production.ProductSubcategory;
```

Zad 2.2

```
SELECT Name FROM Production.Product WHERE Color IS NULL;
```

Zad 2.3

```
SELECT Year(DueDate) 'Rok' , SUM(TotalDue) 'Roczna kwota transakcji'  
FROM Sales.SalesOrderHeader GROUP BY Year(DueDate);
```

Zad 2.4

```
SELECT COUNT(*) 'Liczba klientów' FROM Sales.Customer;
```

```
SELECT COUNT(*) 'Liczba sprzedawców' FROM Sales.SalesPerson;
```

```
SELECT T.Name 'Region', COUNT(C.TerritoryID) 'Liczba klientów'  
FROM Sales.Customer C LEFT JOIN Sales.SalesTerritory T ON T.TerritoryID=C.TerritoryID  
GROUP BY T.Name;
```

```
SELECT T.Name 'Region', COUNT(SP.TerritoryID) 'Liczba sprzedawców'  
FROM Sales.SalesPerson SP LEFT JOIN Sales.SalesTerritory T ON T.TerritoryID=SP.TerritoryID  
GROUP BY T.Name;
```

Zad 2.5

```
SELECT T.rok 'Rok', COUNT(TransactionID) 'Liczba transakcji' FROM  
(SELECT Year(TransactionDate) rok, TransactionID FROM Production.TransactionHistory  
UNION SELECT Year(TransactionDate), TransactionID FROM Production.TransactionHistoryArchive) T  
GROUP BY T.rok;
```

Zad 2.6

```
SELECT PC.Name, PSC.Name, MIN(P.Name) FROM
    (SELECT ProductID FROM Production.Product
    EXCEPT SELECT ProductID FROM Sales.SalesOrderDetail) PID
LEFT JOIN Production.Product P ON P.ProductID=PID.ProductID
LEFT JOIN Production.ProductSubcategory PSC ON PSC.ProductSubcategoryID=P.ProductSubcategoryID
LEFT JOIN Production.ProductCategory PC ON PC.ProductCategoryID=PSC.ProductCategoryID
GROUP BY PC.Name, PSC.Name;
```

Zad 2.7

```
SELECT PSC.Name 'Produkt',
MIN(SO.DiscountPct * P.ListPrice) 'Najmniejsza obnizka',
MAX(SO.DiscountPct * P.ListPrice) 'Najwieksza obnizka'
FROM Production.Product P
    LEFT JOIN Production.ProductSubcategory PSC ON
PSC.ProductSubcategoryID=P.ProductSubcategoryID
    LEFT JOIN Sales.SpecialOfferProduct SOP ON SOP.ProductID=P.ProductID
    LEFT JOIN Sales.SpecialOffer SO ON SO.SpecialOfferID=SOP.SpecialOfferID
WHERE SO.DiscountPct > 0
GROUP BY PSC.Name;
```

Zad 2.8

```
SELECT ProductID 'ID', Name 'Produkt', ListPrice 'Cena' FROM Production.Product
    WHERE ListPrice > (SELECT AVG(ListPrice) FROM Production.Product);
```

Zad 2.9

```
SELECT Month(SOH.OrderDate) 'Miesiąc', AVG(SOH.SalesOrderID) 'Średnia liczba sztuk'
    FROM Sales.SalesOrderHeader SOH
    GROUP BY Month(SOH.OrderDate) ORDER BY Month(SOH.OrderDate);
```


Zad 2.10

```
SELECT ST.CountryRegionCode 'Kod kraju', AVG(DATEDIFF(DAY, OrderDate, ShipDate)) 'Dni oczekiwania'  
  
FROM Sales.SalesOrderHeader SOH  
  
LEFT JOIN Sales.SalesTerritory ST ON ST.TerritoryID=SOH.TerritoryID  
  
GROUP BY ST.CountryRegionCode;
```

Zadanie z zajęć:

```
SELECT ST.CountryRegionCode 'Region', DateDiff(YEAR, E.BirthDate, SYSDATETIME()) 'Wiek',  
  
SUM(LineTotal) 'Suma sprzedana' FROM Sales.SalesOrderDetail SOD  
  
JOIN Sales.SalesOrderHeader SOH ON SOD.SalesOrderID=SOH.SalesOrderID  
  
JOIN Sales.SalesTerritory ST ON ST.TerritoryID=SOH.TerritoryID  
  
JOIN Sales.SalesPerson SP ON SP.TerritoryID=ST.TerritoryID  
  
JOIN HumanResources.Employee E ON E.BusinessEntityID=SP.BusinessEntityID  
  
GROUP BY ST.CountryRegionCode, DateDiff(YEAR, E.BirthDate, SYSDATETIME()) ORDER BY 3;
```

Wnioski: