

# 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

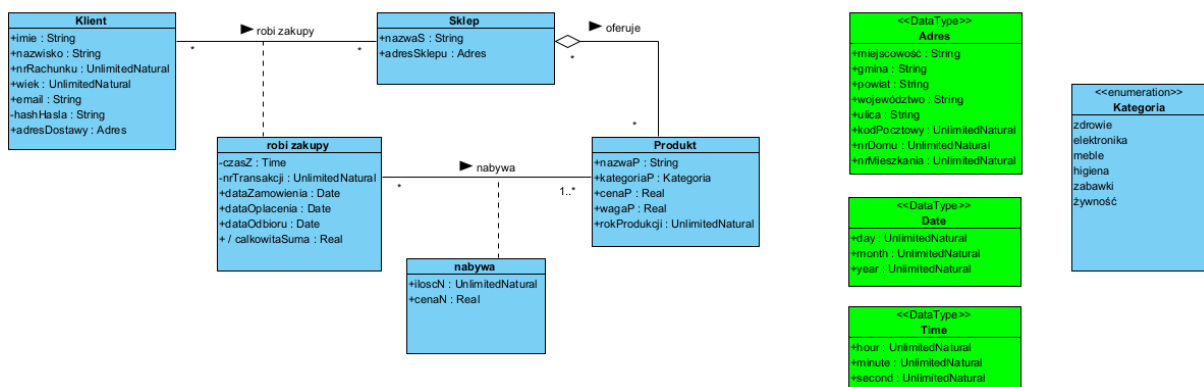
Reg/09 - Produkt może być kupowany wiele razy

Reg/10 – Klient może robić zakupy w wielu sklepach

Reg/11 – Sklep może posiadać wiele produktów

Pozbyłem się Reg/04

## Zad 1.2



Dodałem atrybuty do większości klas. Utworzyłem klasy Adres, Date, Time (które są stereotypu DataType, tzn. mogą być wykorzystane jako typy atrybutów w zwykłych klasach). Dodatkowo wzbogaciłem produkt o kategorię, która jest typem wyliczeniowym konkretnych wartości. Pozbyłem się ograniczenia na „przynajmniej jeden produkt w sklepie”.

### 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,

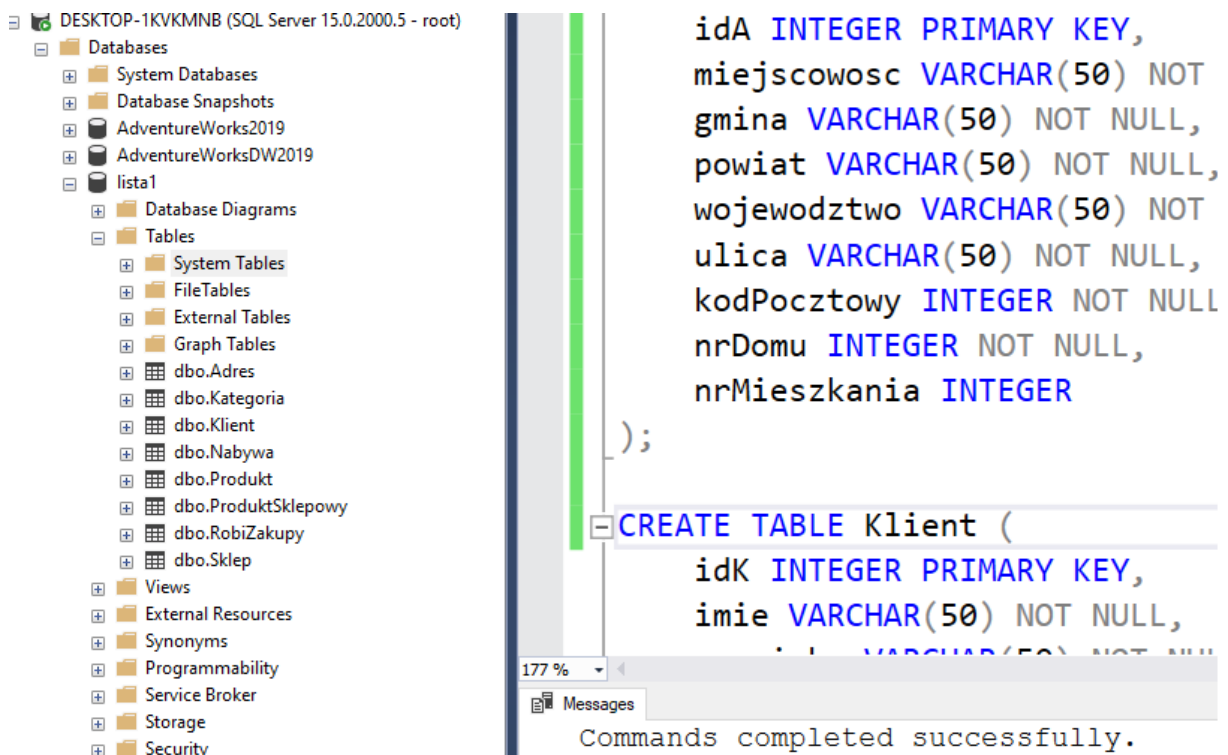
    iloscN INTEGER NOT NULL,

    cenaN DECIMAL(30,4) NOT NULL

);
```

## Zad 1.4

Na podstawie powyższych zapytań pomyślnie utworzyłem bazę danych w systemie MS SQL 2019:



## Zad 1.5

**Przykładowe wprowadzanie rekordów do bazy danych:**

```
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);
```

**Po wprowadzeniu błędnego rekordu (tutaj 'województwo' zostało podane jako NULL przy wprowadzaniu adresu):**

 Messages

Msg 213, Level 16, State 1, Line 1

Column name or number of supplied values does not match table definition.

Completion time: 2022-03-10T13:34:57.2231694+01:00

## Zad 2.1

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

Results		Messages
Liczba produktów		
1	504	

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

Liczba kategorii	
1	4

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

Liczba podkategorii	
1	37

## Zad 2.2

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

Produkt	
233	HL Touring Seat/Sa...
234	Mountain Tire Tube
235	Road Tire Tube
236	Touring Tire Tube
237	LL Mountain Tire
238	ML Mountain Tire
239	HL Mountain Tire
240	LL Road Tire
241	ML Road Tire
242	HL Road Tire
243	Touring Tire
244	LL Touring Handleb...
245	HL Touring Handleb...
246	LL Bottom Bracket
247	ML Bottom Bracket
248	HL Bottom Bracket

**Wniosek:** Prawie połowa produktów nie posiada określonego koloru.

## Zad 2.3

```
SELECT Year(DueDate) 'Rok' , SUM(TotalDue) 'Roczna kwota transakcji'
FROM Sales.SalesOrderHeader GROUP BY Year(DueDate) ORDER BY 1;
```

	Rok	Roczna kwota transakcji
1	2011	13903981,3182
2	2012	35058722,6007
3	2013	48181124,1984
4	2014	26072957,9986

**Wniosek:** Z każdym kolejnym rokiem sumaryczna kwota z transakcji wydaje się rosnąć (dla roku 2014 nie mamy transakcji ze wszystkich miesięcy, tylko do połowy roku -> w związku z tym można założyć, że jeszcze drugie tyle do tej sumy należałoby dodać)

## Zad 2.4

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

	Liczba klientów
1	19820

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

	Liczba sprzedawców
1	17

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

	Region	Liczba klientów
1	Germany	1852
2	Australia	3665
3	United Kingdom	1991
4	Northeast	113
5	Southwest	4696
6	Canada	1791
7	France	1884
8	Northwest	3520
9	Southeast	176
10	Central	132

**Wniosek:** Najwięcej klientów pochodzi ze Stanów Zjednoczonych (Northwest + Southeast + Central + Northeast + Southwest), a następnie z Australii. Z Francji, Niemiec, UK i Kanady mamy do czynienia z podobnymi liczbami.



```
SELECT T.Name 'Region', COUNT(SP.BusinessEntityID) 'Liczba sprzedawców'
```

```
FROM Sales.SalesPerson SP LEFT JOIN Sales.SalesTerritory T ON T.TerritoryID=SP.TerritoryID
```

```
GROUP BY T.Name;
```

	Region	Liczba sprzedawców
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

**Wniosek:** Trzech sprzedawców nie ma w ogóle przypisanego regionu. Ogólnie w USA pracuje najwięcej sprzedawców (co koreluje z liczbą klientów), dla pozostałych regionów są to pojedyncze osoby.

## Zad 2.5

```
SELECT YEAR(DueDate) 'Rok', COUNT(SalesOrderID) 'Liczba transakcji' FROM Sales.SalesOrderHeader
```

```
GROUP BY YEAR(DueDate) ORDER BY 1;
```

	Rok	Liczba transakcji
1	2011	1533
2	2012	3765
3	2013	13514
4	2014	12653

**Wniosek:** Z każdym kolejnym rokiem zwiększa się liczba transakcji (uwzględniając wcześniej wspomnianą niepełność danych dla 2014 roku).

## Zad 2.6

```
SELECT PC.Name 'Kategoria', PSC.Name 'Podkategoria', COUNT(*) 'Liczba produktów' 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;
```

	Kategoria	Podkategoria	Liczba produktów
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

**Wniosek:** Zdecydowana większość produktów, które nie zostały zakupione przez żadnego klienta, nie posiada ani kategorii ani podkategorii. Najwięcej niezakupionych produktów należy do kategorii 'Components' (części rowerowych) oraz 'Accessories' (czyli wszelakich akcesoriów). Ramy rowerowe zawierają najwięcej niezakupionych produktów ze wszystkich podkategorii.

## Zad 2.7

```
SELECT PSC.Name 'Podkategoria',  
  
MIN(SOD.UnitPriceDiscount * SOD.OrderQty * SOD.UnitPrice) 'Najmniejsza obniżka',  
  
MAX(SOD.UnitPriceDiscount * SOD.OrderQty * SOD.UnitPrice) 'Największa obniżka'  
  
FROM Production.Product P  
  
LEFT JOIN Production.ProductSubcategory PSC ON  
PSC.ProductSubcategoryID=P.ProductSubcategoryID  
  
LEFT JOIN Sales.SalesOrderDetail SOD ON SOD.ProductID=P.ProductID  
  
WHERE SOD.UnitPriceDiscount > 0  
  
GROUP BY PSC.Name;
```

	Podkategoria	Najmniejsza obniżka	Największa obniżka
17	Caps	1,103	12,1365
18	Brakes	13,5894	16,0602
19	Tights	9,5687	104,986
20	Wheels	9,1603	134,9762
21	Touring Frames	42,5444	634,9731
22	Road Frames	39,1175	245,3674
23	Socks	1,1471	15,20
24	Tires and Tubes	0,3188	0,3188
25	Mountain Bikes	45,1992	3248,4279
26	Chains	2,5826	2,8174
27	Bike Racks	15,312	174,00
28	Cranksets	56,3746	65,7704
29	Handlebars	5,6833	19,5976
30	Gloves	3,0047	112,8303
31	Headsets	13,0522	50,6336
32	Saddles	3,4605	5,4483

**Wniosek:** Największe wartości w maksymalnej obniżce dotyczą ram i rowerów górskich natomiast najmniejsze dla ubrań i akcesoriów. Wynika z tego, że im droższy dany produkt tym większe wartości obniżki (**w złotych** a nie procentowo!).

## Zad 2.8

SELECT Name 'Produkt', ListPrice 'Cena' FROM Production.Product

WHERE ListPrice > (SELECT AVG(ListPrice) FROM Production.Product);

	Produkt	Cena
121	Mountain-400-W Silver, 40	769,49
122	Mountain-400-W Silver, 42	769,49
123	Mountain-400-W Silver, 46	769,49
124	Mountain-500 Silver, 40	564,99
125	Mountain-500 Silver, 42	564,99
126	Mountain-500 Silver, 44	564,99
127	Mountain-500 Silver, 48	564,99
128	Mountain-500 Silver, 52	564,99
129	Mountain-500 Black, 40	539,99
130	Mountain-500 Black, 42	539,99
131	Mountain-500 Black, 44	539,99
132	Mountain-500 Black, 48	539,99
133	Mountain-500 Black, 52	539,99
134	Road-750 Black, 44	539,99
135	Road-750 Black, 48	539,99
136	Road-750 Black, 52	539,99

**Wniosek:** Produkty droższe od średniej to ramy i całe rowery.

## Zad 2.9

```
SELECT Month(SOH.OrderDate) 'Miesiąc', CAST(CAST(SUM(SOD.OrderQty) AS FLOAT) / COUNT(DISTINCT  
YEAR(OrderDate)) AS DECIMAL(10,2)) 'Średnia liczba sztuk'
```

```
FROM Sales.SalesOrderHeader SOH JOIN Sales.SalesOrderDetail SOD ON  
SOD.SalesOrderID=SOH.SalesOrderID
```

```
GROUP BY Month(SOH.OrderDate) ORDER BY 1;
```

	Miesiąc	Średnia liczba sztuk
1	1	6528.00
2	2	3793.33
3	3	11338.67
4	4	4722.33
5	5	8673.00
6	6	7544.25
7	7	9980.00
8	8	6673.00
9	9	7675.67
10	10	8971.67
11	11	4501.00
12	12	5831.33

**Wniosek:** Najmniej sprzedawanych sztuk produktów jest w lutym a najwięcej w marcu. Ciężko doszukać się jakichś trendów na podstawie kolejnych miesięcy czy kwartałów.

## Zad 2.10

SELECT ST.CountryRegionCode 'Kod kraju', AVG(CAST(DATEDIFF(DAY, OrderDate, ShipDate) AS FLOAT)) 'Dni oczekiwania'

FROM Sales.SalesOrderHeader SOH

LEFT JOIN Sales.SalesTerritory ST ON ST.TerritoryID=SOH.TerritoryID

GROUP BY ST.CountryRegionCode;

	Kod kraju	Dni oczekiwania
1	DE	7,00343118566527
2	GB	7
3	AU	7
4	CA	7
5	FR	7
6	US	7

**Wniosek:** W zasadzie w każdym kraju niezależnie od kontynentu średnio klient oczekuje tydzień na wysyłkę.

## 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;

	Region	Wiek	Suma sprzedana
1	DE	47	4915407.595885
2	FR	47	7251555.646926
3	GB	54	7670721.035475
4	US	60	7909009.005872
5	AU	45	10655335.959317
6	US	44	16084942.547585
7	CA	47	16355770.454862
8	CA	59	16355770.454862
9	US	42	24184609.600810
10	US	54	31123984.081815
11	US	48	40049540.167321

**Wniosek:** Pracownicy z Ameryki Północnej sprzedają więcej niż pracownicy z innych części świata. Wiek wydaje się nie mieć w tym przypadku znaczenia.

## Wnioski:

- Zestaw reguł i ograniczeń biznesowych jest bardzo pomocny przy tworzeniu modelu danych (np. diagramu klas UML)
- Sam diagram klas w sposób wizualny (za pomocą asocjacji i atrybutów) pozwala intuicyjnie zwizualizować przyszłą bazę danych (klucze główne/obce i relacje same w sobie). Należy na tym etapie zachować precyzję i dobrze przemyśleć wszystkie funkcjonalności gdyż przełoży się to potem na funkcjonowanie całego systemu/całej bazy.
- Utworzenie fizycznej bazy danych na podstawie diagramu UML'owego nie stanowi już większego problemu (odpowiednie zapytania DDL SQL pozwalają dokładnie odwzorować wszelkie powiązania i ograniczenia). SZBD będzie dokładnie przestrzegać wszystkich wyznaczonych przez nas reguł.
- Za pomocą SQL'a możemy „wyciągnąć” proste informacje z naszej bazy danych (o liczbie rekordów czy o konkretnych rekordach). Dodatkowo możemy tworzyć kwerendy, które pozwolą nam zaobserwować korelacje między pewnymi danymi naszej bazy. Należy w tym wypadku być ostrożnym, musimy być świadomi struktury naszej bazy, tego co chcemy tak naprawdę uzyskać oraz jak należy interpretować otrzymane wyniki.