# Hurtownie Danych

## Lista 1

Joanna Pszon

256767

K06-01e

**Zad 1.1.**

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 – Klient nabywa co najmniej jeden produkt

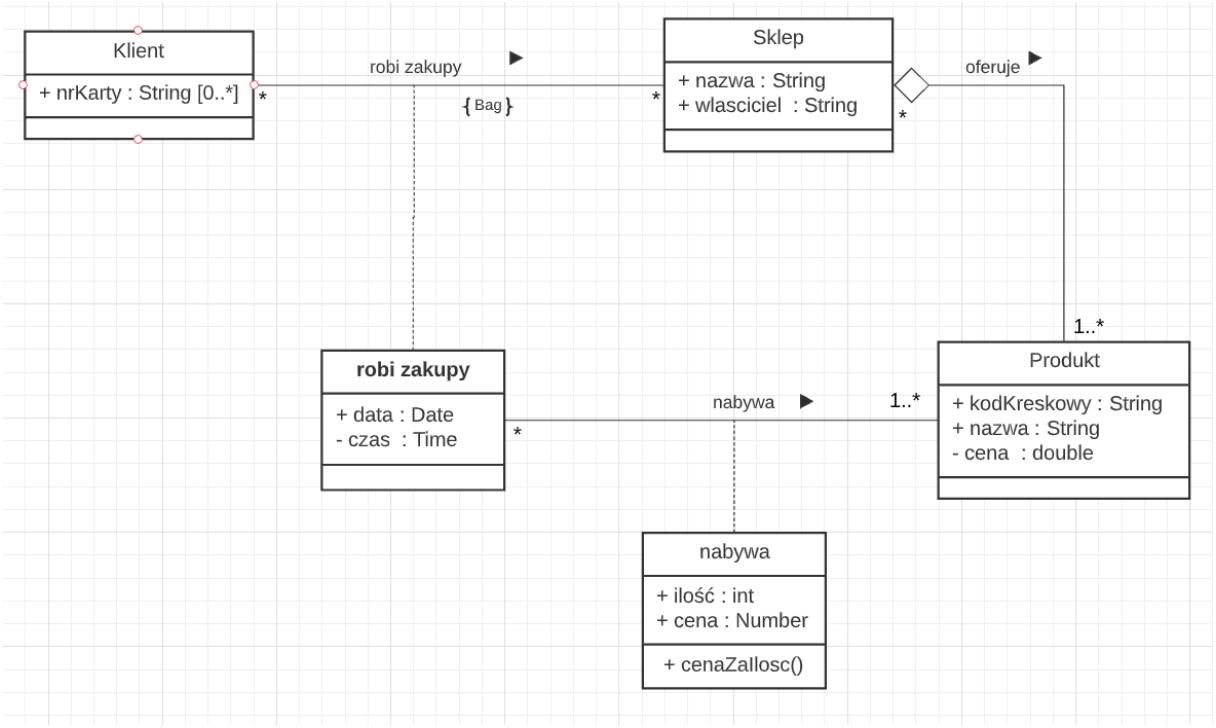
Reg/06 – Klient może robić zakupy w wielu sklepach

Reg/07 – Produkt może być w wielu zakupach

Reg/08 – Każdy produkt w ramach zakupów ma określoną ilość i cenę

Reg/09 – Produkt może być w wielu sklepach

**Zad 1.2.**



**Zad 1.3.**

```
create table Klient(
  idKlienta integer primary key,
  nrKarty varchar
);

create table Sklep(
  nazwa varchar primary key,
  wlasciciel varchar
);

create table robiZakupy(
  idZakupow integer primary key,
  idKlienta integer references Klient(idKlienta),
  nazwa varchar references Sklep(nazwa),
  Dataz date,
  czas time
);

create table Produkt(
  kodKreskowy varchar primary key,
  nazwa varchar,
  cena real
);

create table ProduktWSklepie(
  idProduktWSklepie integer primary key,
  kodKreskowy varchar references Produkt(kodKreskowy),
  nazwa varchar references Sklep(nazwa)
);
```

```
create table nabywa(
  idNabycia integer primary key,
  kodKreskowy varchar references Produkt(kodKreskowy),
  idZakupow integer references robiZakupy(idZakupow),
  ilosc integer,
  cena real
);
```

**Zad 2.1.**

```
SELECT COUNT(ProductID)
FROM Product;


SELECT COUNT(ProductSubcategoryID)
FROM ProductSubcategory;


SELECT COUNT(ProductCategoryID)
FROM ProductCategory;
```

**Zad 2.2.**

```
SELECT *
FROM Product
WHERE Color IS NULL;


SELECT Color
FROM Product
LEFT JOIN ProductSubcategory
ON Product.ProductSubcategoryID = ProductSubcategory.ProductSubcategoryID
WHERE (ProductSubcategory.ProductCategoryID IS NULL) OR
(Product.ProductSubcategoryID IS NULL)
```

**Zad 2.3.**

```
SELECT SUM(TotalDue)
FROM SalesOrderHeader
GROUP BY YEAR(DueDate);
```

**Zad 2.4.**

```
SELECT COUNT(CustomerID)
FROM Customer;

SELECT COUNT(BusinessEntityID)
FROM SalesPerson;

SELECT COUNT(CustomerID), SalesTerritory.Name
FROM Customer
LEFT JOIN SalesTerritory
ON Customer.TerritoryID = SalesTerritory.TerritoryID
GROUP BY Customer.TerritoryID;

SELECT COUNT(BusinessEntityID), SalesTerritory.Name
FROM SalesPerson
LEFT JOIN SalesTerritory
ON BusinessEntityID.TerritoryID = SalesTerritory.TerritoryID
GROUP BY BusinessEntityID.TerritoryID;
```

**Zad 2.5.**

```
SELECT COUNT(SalesOrderID)
FROM SalesOrderHeader
GROUP BY YEAR(OrderDate);
```

**Zad 2.6.**

```
SELECT ProductName, ProductSubcategoryID, ProductSubcategory.ProductCategoryID
FROM Product
INNER JOIN ProductSubcategory
ON Product.ProductSubcategoryID = ProductSubcategory.ProductSubcategoryID
WHERE NOT EXISTS (SELECT *
                  FROM SalesOrderDetail
                  WHERE SalesOrderDetail.ProductID = Product.ProductID)
GROUP BY ProductSubcategory.ProductCategoryID, ProductSubcategoryID;
```

**Zad 2.7.**

```
SELECT ProductID, MIN(UnitPriceDiscount * OrderQty), MAX(UnitPriceDiscount *
OrderQty), Product.ProductSubcategoryID

FROM SalesOrderDetail

LEFT JOIN Product

ON SalesOrderDetail.ProductID = Product.ProductID

GROUP BY Product.ProductSubcategoryID
```

**Zad 2.8.**

```
SELECT P1.ProductID

FROM Product P1

WHERE P1.ListPrice > (SELECT AVG(P2.ListPrice)

                      FROM Product P2)
```

**Zad 2.9.**

```
SELECT AVG(MonthsSales)

FROM

     (SELECT SUM(SalesOrderDetail.OrderQty) AS MonthsSales

     FROM SalesOrderDetail

     INNER JOIN SalesOrderHeader

     ON SalesOrderId = SalesOrderID

     GROUP BY MONTH(SalesOrderHeader.DueDate));
```

**Zad 2.10.**

```
SELECT AVG(SalesOrderHeader.ShipDate – SalesOrderHeader.OrderDate),
SalesTerritory.CountryRegionCode

FROM SalesOrderHeader

LEFT JOIN SalesTerritory

ON SalesTerritory.TerritoryID = SalesOrderHeader.TerritoryID

GROUP BY SalesTerritory.CountryRegionCode
```

**Wnioski:**

Diagram klas pomaga w stworzeniu projektu bazy danych. Szczególnie rozbudowany i dobrze opisany. W przypadku dużej ilości pól baza traci jednak na czytelności.

```
SELECT ProductCategory.ProductCategoryName, ((SalesOrderDetail.UnitPrice *
SalesOrderDetail.UnitPriceDiscount)/100) * SalesOrderDetail.OrderQty

      Employee.JobTitle

FROM SalesOrderDetail

LEFT JOIN Product ON SalesOrderDetail.ProductID = Product.ProductID

LEFT JOIN ProductSubcategory ON ProductSubcategory.ProductSubcategoryID =
Product.ProductSubcategoryID

LEFT JOIN ProductCategory ON ProductSubcategory.ProductCategoryID =
ProductCategory.ProductCategoryID

LEFT JOIN SalesOrderHeader ON SalesOrderHeader.SalesOrderID =
SalesOrderDetail.SalesOrderID

LEFT JOIN SalesPerson ON SalesOrderHeader.SalesPersonID =
SalesPerson.BusinessEntityID

LEFT JOIN Employee ON SalesPerson.BusinessEntityID = Employee.BusinessEntity

GROUP BY ProductCategory.ProductCategoryName, Employee.JobTitle
```