

Hurtownie danych – Spr. 1.

PWr. WIZ, Data: 10.03.2021

Student	Email: 251526@student.pwr.edu.pl	Ocena
Indeks	<u>251526</u>	
Imię	<u>Volodymyr</u>	
Nazwisko	<u>Zakhovaiko</u>	

Zestaw składa się z 1 zadania. Jeżeli nie potrafisz rozwiązać zadania, to próbuj podać, chociaż częściowe rozwiązanie lub uzasadnienie przyczyny braku rozwiązania. Pamiętaj o podaniu nr. indeksu oraz imienia i nazwiska.

SZBD: MS SQL 2017 lub 2019

Baza danych: **251526**

Zad. 1.

Proszę przeanalizować konceptualny model danych „Usługi” (Rys. 1.), który jest niekompletny, ale klasy i relacje między nimi mogą reprezentować rozpatrywany wycinek rzeczywistości. Następnie należy wykonać następujące zadania:

1. Zweryfikować model danych w kontekście podanego zbioru reguł i ograniczeń dziedzinowych modyfikując zbiór reguł i ograniczeń (uzupełniając ich definicję) lub korygując konceptualny model danych (uzasadniając!)
2. Przedstawić uzupełnioną i poprawioną wersję modelu danych (kompletny diagram klas UML)
3. Utworzyć logiczny/fizyczny model danych w postaci skryptu w języku DDL SQL (uwzględniając reguły i ograniczenia dziedzinowe), starając się zachować zgodność ze standardem języka SQL (pomijając, o ile to możliwe, natywne konstrukcje implementacji języków SQL)
4. Utworzyć bazę danych w systemie MS SQL 2017 lub 2019, która jest fizycznym modelem danych modelowanego wycinka rzeczywistości
5. Wprowadzić kilka rekordów do każdej tabeli sprawdzając poprawność implementacji (zarówno poprawne dane, jak i niezgodne z obowiązującymi regułami – komentując i wyjaśniając uzyskane komunikaty z systemu SZBD)

Reguły i graniczenia dziedzinowe:

Reg/01 – klient może wielokrotnie robić zakupy w tym samym sklepie

Reg/02 – W sklepie może robić zakupy dowolny klient

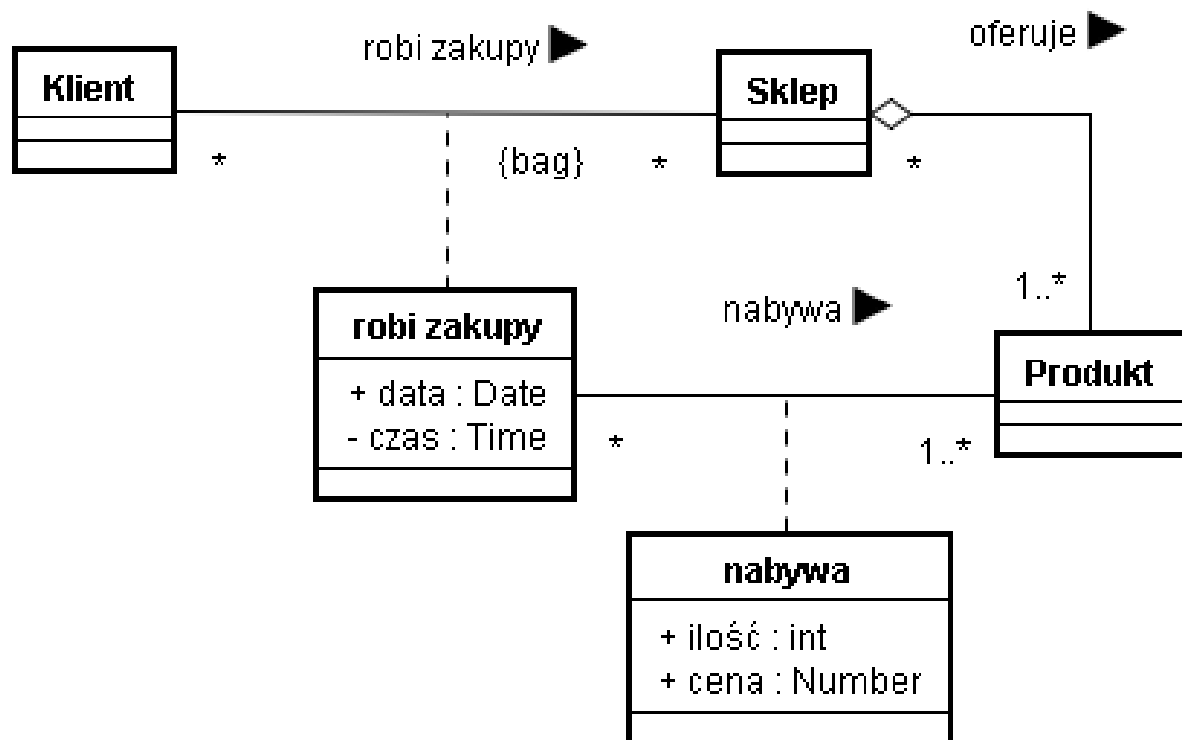
Reg/03 – Każdy zakup realizowany jest przez klienta w sklepie w określonym dniu i godzinie

Reg/04 – Sklep musi oferować co najmniej jeden produkt

Reg/05 – Ten sam produkt (rodzaj) może być oferowany przez wiele sklepów

Reg/06 – Każdy sklep może indywidualnie proponować cenę i ilość oferowanego produktu

Reg/07 - ...



Rysunek 1. Konceptualny model danych „Usługi”

Rozwiązanie:

1. Zweryfikować model danych w kontekście podanego zbioru reguł i ograniczeń dziedzinowych modyfikując zbiór reguł i ograniczeń (uzupełniając ich definicję) lub korygując konceptualny model danych (uzasadniając!)

Reg/01 – klient może wielokrotnie robić zakupy w tym samym sklepie

~~Reg/02 – W sklepie może robić zakupy dowolny klient~~

Reg/02 – Klient musi mieć co najmniej 18 lat.

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 – Ten sam produkt (rodzaj) może być oferowany przez wiele sklepów

Reg/06 – Każdy sklep może indywidualnie zaproponować cenę i ilość oferowanego produktu

Reg/07 – Cena produktu nie może być ujemna.

Reg/08 – Czas zakupu nie może być w przyszłości.

Reg/09 – Ilość dostępnego towaru nie może być ujemna.

Reg/10 – Pesel powinien mieć 11 znaków.

Reg/11 – Uniwersalny międzynarodowy kod produktu (UPC) powinien składać się z 12 cyfr.

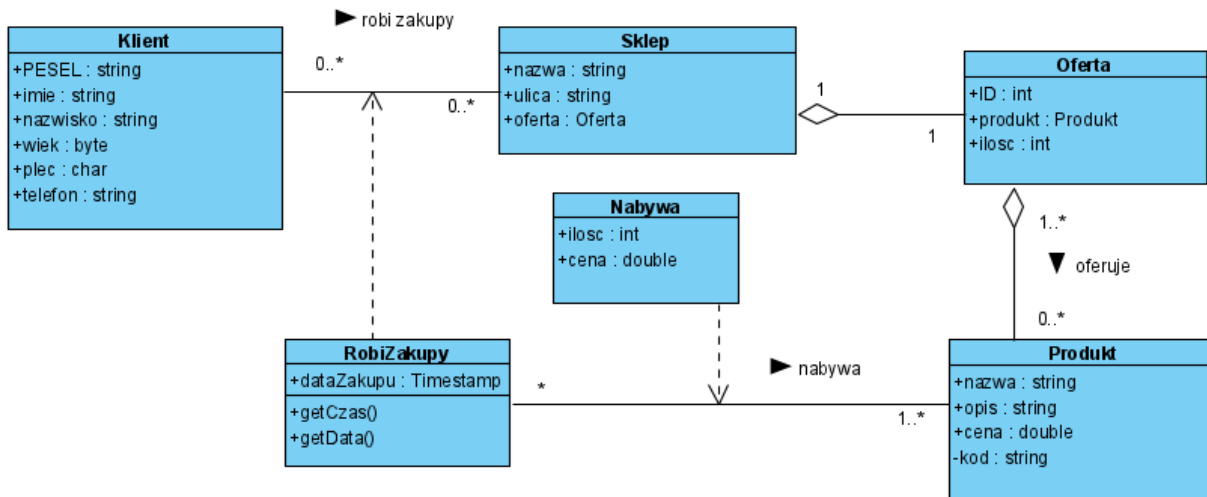
Reg/12 – Płeć człowieka musi być lub literą ‘m’, lub ‘k’.

PESEL jest prawidłowy tylko w przypadku, kiedy ma 11 znaków.

Ilość dostępnego towaru, oferowanego przez sklep musi być nie ujemna, bo sklep nie może oferować to, czego nie ma.

Klient nie może kupować coś w przyszłości, dlatego czas zakupu musi być wcześniejszy niż moment zakupu.

2.



Do klienta zostały dodane atrybuty: PESEL, imię, nazwisko, wiek, płeć i numer telefonu. Używamy typu string dla PESEL, bo jest on złożony z kilku części, które będzie ciężko znaleźć w przypadku typu PESELa jako liczby.

Do sklepu dodałem atrybuty: nazwa sklepu, ulica (null w przypadku sklepu internetowego), oferta. Oferta składa się z atrybutów ID, listy produktów i ich ilości. ID jest niezbędne dla identyfikacji oferty do którego sklepu należy.

Dla produktu dodałem atrybuty: nazwa, opis, cena, kod produktu dla identyfikacji, jeśli nazwy są jednakowe w kilku takich.

Datę i czas zakupu zmieniłem na jeden typ Timestamp i dodałem dwie metody dla otrzymania czasu i daty oddzielnie.

3. Kod SQL

```
-- CREATE DATABASE "251526";

CREATE TABLE [251526].dbo.Klient
(
    PESEL CHAR(11) NOT NULL,
    imie VARCHAR(255) NOT NULL,
    nazwisko VARCHAR(255) NOT NULL,
    wiek TINYINT NOT NULL,
    plec CHAR(1) CONSTRAINT ck_plec CHECK (plec in ('m', 'k')),
    telefon CHAR(11)
);

CREATE TABLE [251526].dbo.Produkt
```

```

(
    kod CHAR(12) NOT NULL,
    nazwa VARCHAR(255) NOT NULL,
    opis VARCHAR(255) NOT NULL,
    cena REAL NOT NULL,
    kategoria VARCHAR(255) NOT NULL
);

CREATE TABLE [251526].dbo.Oferta
(
    ofertaID BIGINT IDENTITY(1,1) NOT NULL,
    produktID CHAR(12) NOT NULL,
    ilosc INT DEFAULT 0
);

CREATE TABLE [251526].dbo.Sklep
(
    nazwa VARCHAR(255) NOT NULL,
    ulica VARCHAR(255) NOT NULL,
    ofertaID BIGINT NOT NULL
);

CREATE TABLE [251526].dbo.Nabywa
(
    nabywaID BIGINT IDENTITY(1,1) NOT NULL,
    robizakupyID BIGINT NOT NULL,
    ilosc INT NOT NULL,
    cena REAL NOT NULL
);

CREATE TABLE [251526].dbo.Robizakupy
(
    robizakupyID BIGINT IDENTITY(1, 1) NOT NULL,
    klientID CHAR(11) NOT NULL,
    produktID CHAR(12) NOT NULL,
    dataZakupu DATE DEFAULT CURRENT_TIMESTAMP
);

ALTER TABLE [251526].dbo.Klient
ADD CONSTRAINT pk_Klient PRIMARY KEY (PESEL),
    CONSTRAINT ck_plec CHECK (plec in ('m', 'k')),
    CONSTRAINT ck_wiek CHECK (wiek >= 18);

ALTER TABLE [251526].dbo.Produkt
ADD CONSTRAINT pk_Produkt PRIMARY KEY (kod),
    CONSTRAINT ck_Produkt_cena CHECK (cena >= 0);

ALTER TABLE [251526].dbo.Oferta
ADD CONSTRAINT pk_Oferta PRIMARY KEY (ofertaID),
    CONSTRAINT ck_Oferta_ilosc CHECK (ilosc >= 0),
    CONSTRAINT fk_Oferta_Produkt FOREIGN KEY (produktID)
    REFERENCES [251526].dbo.Produkt(kod);

ALTER TABLE [251526].dbo.Sklep
ADD CONSTRAINT pk_Sklep PRIMARY KEY (nazwa),
    CONSTRAINT fk_Sklep_Oferta FOREIGN KEY (ofertaID)
    REFERENCES [251526].dbo.Oferta(ofertaID);

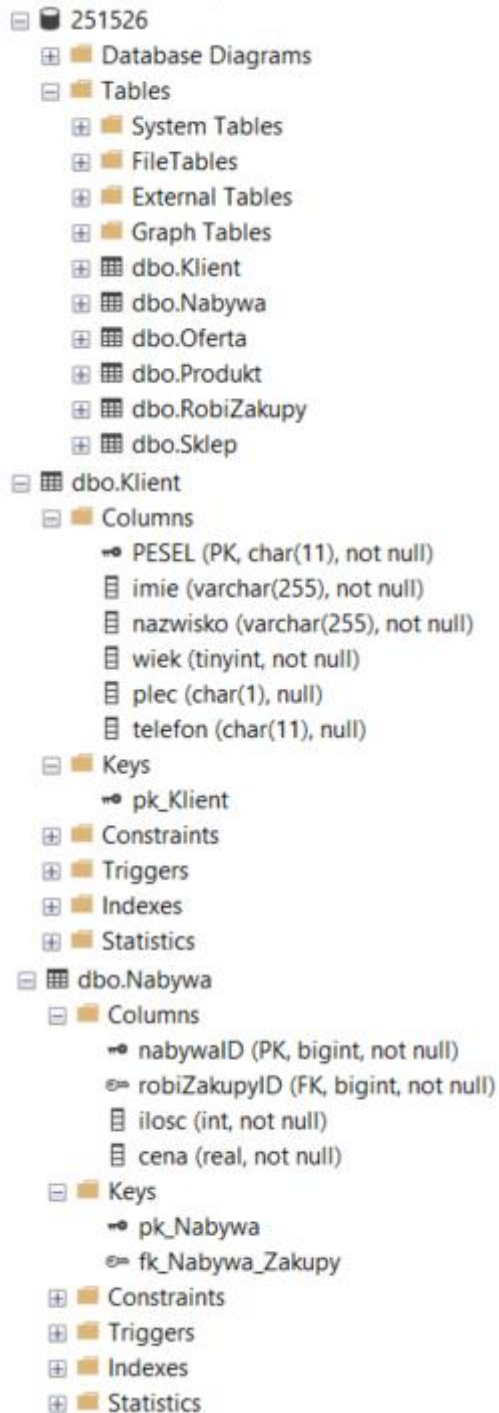
ALTER TABLE [251526].dbo.Nabywa
ADD CONSTRAINT pk_Nabywa PRIMARY KEY (nabywaID),
    CONSTRAINT ck_Nabywa_ilosc CHECK (ilosc > 0),
    CONSTRAINT ck_Nabywa_cena CHECK (cena > 0),
    CONSTRAINT fk_Nabywa_Produkt FOREIGN KEY (produktID)

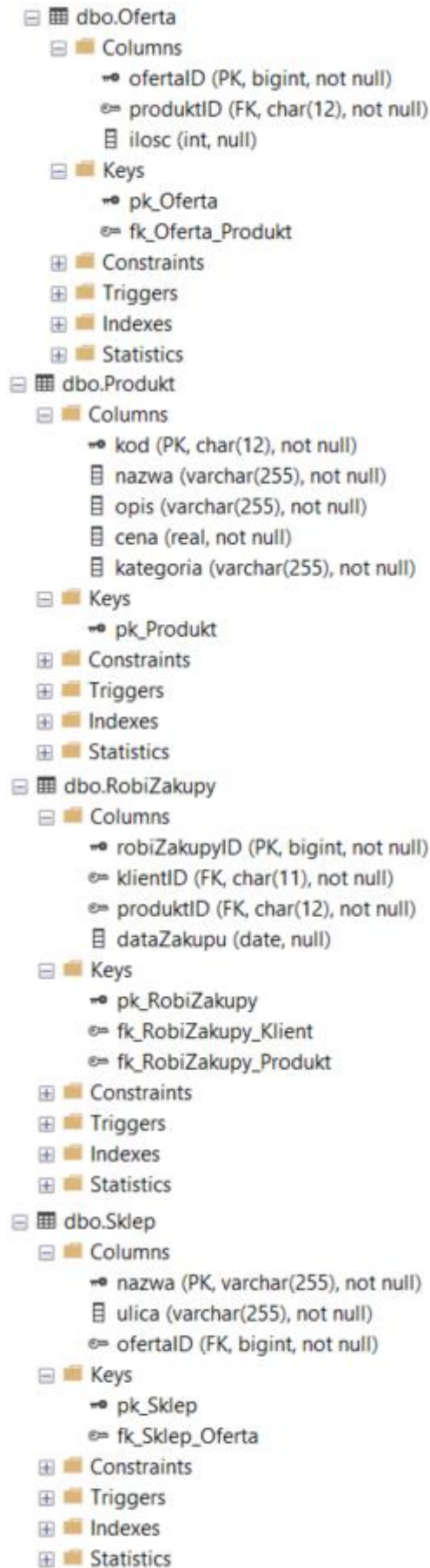
```

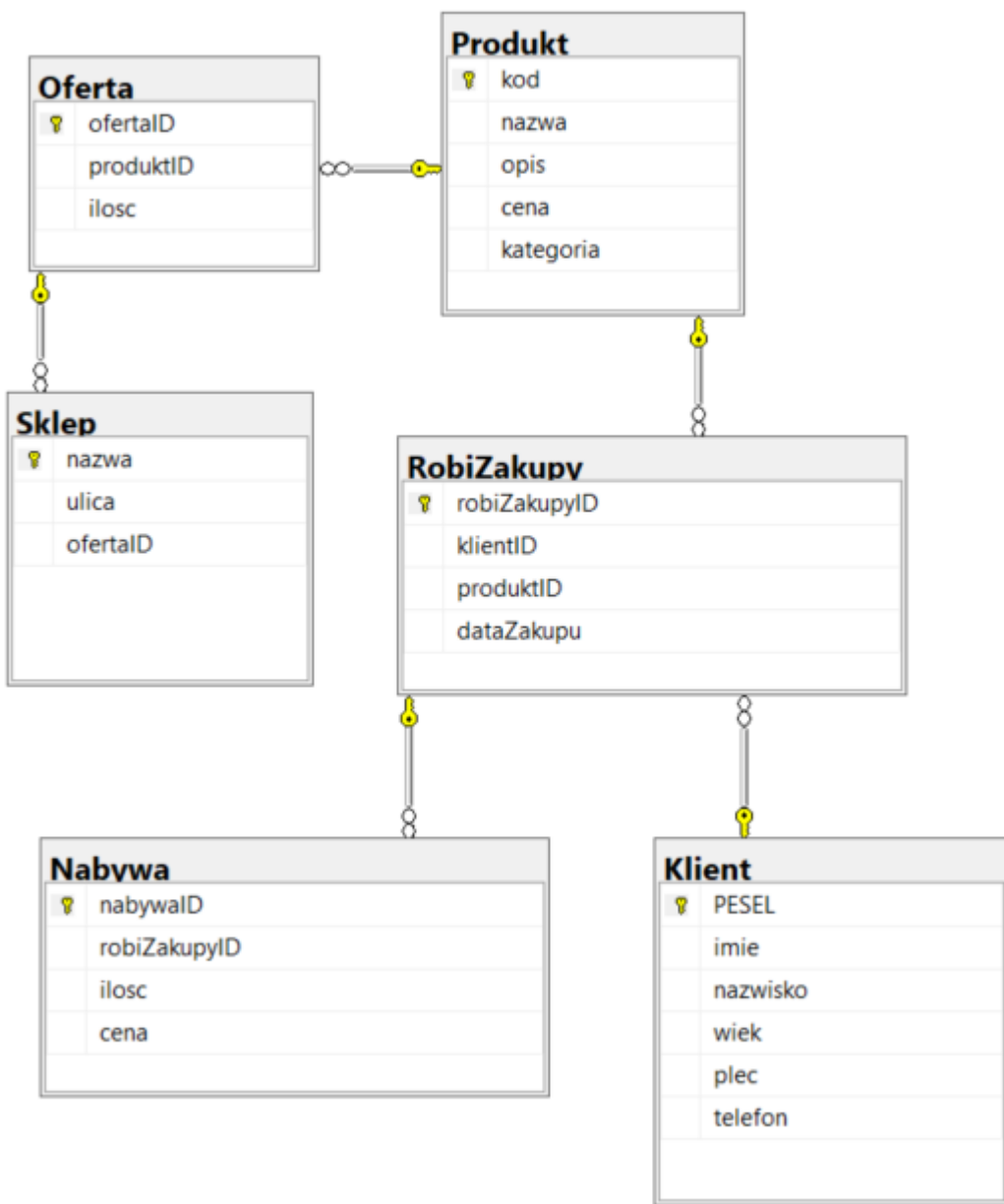
```
REFERENCES [251526].dbo.Produkt(kod);
```

```
ALTER TABLE [251526].dbo.RobiZakupy  
ADD CONSTRAINT pk_RobiZakupy PRIMARY KEY (robiZakupyID),  
CONSTRAINT fk_RobiZakupy_Klient FOREIGN KEY (klientID)  
REFERENCES [251526].dbo.Klient(PESEL),  
CONSTRAINT fk_RobiZakupy_Produkt FOREIGN KEY (produktID)  
REFERENCES [251526].dbo.Produkt(kod);
```

4.







5.

Przykład z dodawaniem jednakowych PESELów.

```

INSERT INTO [251526].dbo.Klient VALUES ('12345678901', 'Volodymyr', 'Zakhovaiko', 19, 'm',
'48881035771');
INSERT INTO [251526].dbo.Klient VALUES ('12345678900', 'Anna', 'Kowalska', 20, 'k',
'48881035772');
INSERT INTO [251526].dbo.Klient VALUES ('12345678901', 'Adam', 'Malewski', 20, 'm',
'48881035773');
  
```

Violation of PRIMARY KEY constraint 'pk_Klient'. Cannot insert duplicate key in object 'dbo.Klient'. The duplicate key value is (12345678901). The statement has been terminated.

Nie możemy dodawać dwa lub więcej jednakowych PESELi, bo ukazaliśmy, że jest to znaczenie unikalne.

Próbujemy dodać sklep bez dodawania oferty dla niego.

```
INSERT INTO [251526].dbo.Sklep VALUES ('24/7 Spożywczy', 'Wroclawska 42', 1);
```

The INSERT statement conflicted with the FOREIGN KEY constraint "fk_Sklep_Oferta". The conflict occurred in database "251526", table "dbo.Oferta", column 'ofertaID'. The statement has been terminated.

Dodajmy kilka produktów:

```
INSERT INTO [251526].dbo.Produkt VALUES ('000011112222', 'Xiaomi Mi Bend 3', 'Fitness gadzet', 120.87, 'akcesoria');
INSERT INTO [251526].dbo.Produkt VALUES ('000011112223', 'Samsung Galaxy 2', 'Mobile phone', 1200, 'telefony');
INSERT INTO [251526].dbo.Produkt VALUES ('000011112224', 'Lenovo 430s', 'Laptop ultrathin', 8000, 'laptopy');
```

Przy dodaniu tego samego produktu otrzymujemy błąd.

```
INSERT INTO [251526].dbo.Produkt VALUES ('000011112224', 'Lenovo 430s', 'Laptop ultrathin', 8000, 'laptopy');
```

Violation of PRIMARY KEY constraint 'pk_Produkt'. Cannot insert duplicate key in object 'dbo.Produkt'. The duplicate key value is (000011112224).

Teraz dodajmy kilka ofert:

```
INSERT INTO [251526].dbo.Oferta VALUES ('000011112222', 12);
INSERT INTO [251526].dbo.Oferta VALUES ('000011112223', 190);
```

Dodamy dwa sklepy, które mają wyższe oferty:

```
INSERT INTO [251526].dbo.Sklep VALUES ('24/7 Spożywczy', 'Wroclawska 42', 1);
INSERT INTO [251526].dbo.Sklep VALUES ('IKEA', 'Gdanska 43', 2);
```

Przy ponownym uruchomieniu powyższych komend otrzymamy błąd:

```
INSERT INTO [251526].dbo.Sklep VALUES ('IKEA', 'Gdanska 43', 2);
```

Violation of PRIMARY KEY constraint 'pk_Sklep'. Cannot insert duplicate key in object 'dbo.Sklep'. The duplicate key value is (IKEA).

Wszystkie błędy są otrzymane zgodnie z regułami. Dalej robimy zakup i dodajemy informację o ilości zakupionego towaru.

```
INSERT INTO [251526].dbo.RobiZakupy VALUES ('12345678901', '000011112222');
INSERT INTO [251526].dbo.Nabywa VALUES (12, 2100);
```

Msg 213, Level 16, State 1, Line 99
Column name or number of supplied values does not match table definition.

Otrzymujemy błąd, bo nie ukazaliśmy numer zakupu, dla którego wskazujemy dodatkowe informację.

Wnioski:

1. Dobrze zaprojektowana baza danych zapewnia bezpieczeństwo przechowywania prawidłowych danych i chroni nas przed wklejaniem i wprowadzeniem niepoprawnych.
2. W tym zadaniu do zaprojektowania bazy danych lepiej używać Entity Relationship Diagram, który pozwala dokładniej opisać wymagania do pól i typów tych pól.
3. W przyszłości do zaprojektowanej bazy można zastosować optymalizację, różne postaci normalne. Przykładem jest związek tabel Klient – Produkt. Wiele klientów może kupować wiele produktów, czyli jeden klient może kupować wiele produktów. To właśnie, dlaczego nam powtarzać i chronić te same dane dużą ilość razy? Możemy uniknąć tego przez wprowadzenie dodatkowych tablic wspomagających.
4. Także poprawność danych sprawdzamy w kilku różnych sposobów, przez użycie opcji NOT NULL, CONSTRAINT, UNSIGNED i innych.

Uwaga:

- **Sprawozdanie bez wniosków końcowych nie będzie sprawdzane i tym samym ocena jest negatywna!**
- Plik ze sprawozdaniem powinien mieć nazwę **Spr01HD-index-Nazwisko - 2021** i format **pdf**
- Oprogramowanie MS SQL 2017 lub 2019
- Model koncepcyjny danych należy przygotować wykorzystując narzędzie wspierające język UML np. Visual Paradigm

Tableau:

Students can continue using Tableau after the class is over by individually requesting their own one-year license through the [Tableau for Students program](#).

1. [Download Tableau Desktop and Tableau Prep here](#)
2. Select each product download link to get started. When prompted, enter your school email address for Business E-mail and enter the name of your school for Organization.
3. Activate with your product key:

Are your students new to Tableau? Share our free [Data Analytics for University Students guide](#) to help them get started.