

Hurtownie danych – Spr. 1.

PWr. WIZ, Data: 7.03.2022

Student	Email: 256305@student.pwr.edu.pl	Ocena
Indeks	<u>256305</u>	
Imię	<u>Grzegorz</u>	
Nazwisko	<u>Dzikowski</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: ?

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

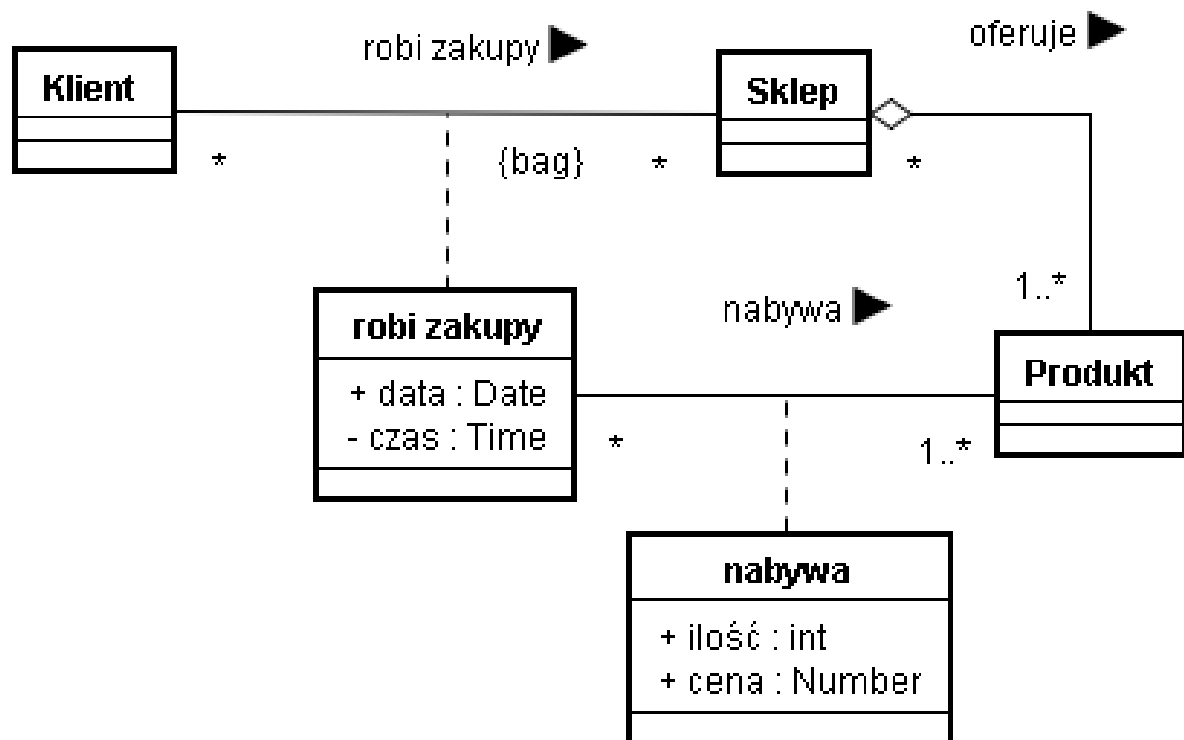
Reg.03 – Każdy zakup realizowany jest przez klienta w sklepie w określonym dniu i godzinie

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.

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 jednego klienta w sklepie w określonym dniu i godzinie

Reg/04 – Każdy 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 – Każdy zakup musi posiadać przynajmniej jeden produkt w ilości większej od zera

Reg/08 – Data każdego zakupu nie może być w przyszłości

Reg/09 – Jeden klient może robić zakupy w różnych sklepach

Reg/10 - Każdy klient musi posiadać typ (osoba fizyczna, firma lub instytucja) oraz posiadać nazwę (max 255 znaków), adres dostawy (max 1028 znaków) i numer telefonu (max 30 znaków)

Reg/11 – Każdy klient może posiadać email (max 255 znaków)

Reg/12 – Każdy sklep musi posiadać nazwę (max 255 znaków), adres (max 1028 znaków) oraz numer telefonu (max 30 znaków)

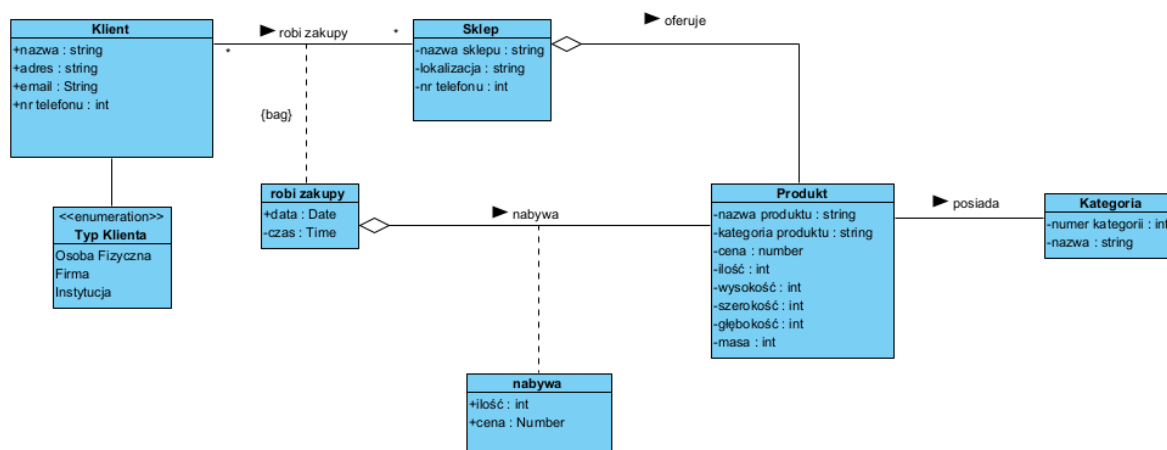
Reg/13 – Każdy produkt musi zawierać unikalny numer (liczba), nazwę (max 255 znaków), cenę (liczba zmiennoprzecinkowa), ilość (liczba) oraz kategorię

Reg/14 – Każdy produkt może posiadać długość (liczbę), szerokość (liczbę), głębokość (liczbę) w cm i masę w gramach (liczbę)

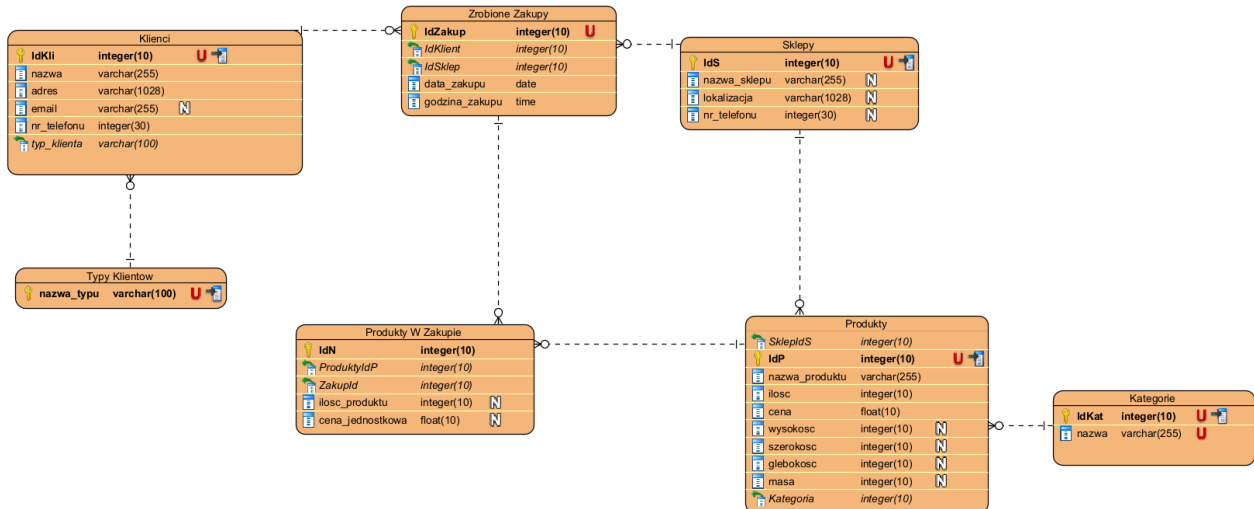
Reg/14 – Każda kategoria posiada unikalny numer (liczbę) i nazwę (max 255 znaków)

Reg/15 – Każda nazwa kategorii jest unikalna

2.



3.



CREATE TABLE "Typy Klientow" (

 nazwa_typu varchar(100) NOT NULL,

 PRIMARY KEY (nazwa_typu));

CREATE TABLE Kategorie (

 IdKat INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),

 nazwa varchar(255) NOT NULL UNIQUE);

CREATE TABLE Klienci (

 IdKli INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),

 nazwa varchar(255) NOT NULL,

 adres varchar(1028) NOT NULL,

 email varchar(255),

 nr_telefonu int NOT NULL,

 typ_klienta varchar(100) NOT NULL,

 FOREIGN KEY(typ_klienta) REFERENCES "Typy Klientow"(nazwa_typu));

CREATE TABLE Sklepy (

 IdS INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),

```
nazwa_sklepu varchar(255),  
lokalizacja varchar(1028),  
nr_telefonu int);
```

```
CREATE TABLE "Zrobione Zakupy" (  
    IdZakup    INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),  
    IdKlient   int NOT NULL,  
    IdSklep    int NOT NULL,  
    data_zakupu date NOT NULL,  
    godzina_zakupu time NOT NULL,  
    FOREIGN KEY(IdKlient) REFERENCES Klienci(IdKli),  
    FOREIGN KEY(IdSklep) REFERENCES Sklepy(IdS));
```

```
CREATE TABLE Produkty (  
    SklepIdS    int NOT NULL,  
    IdP         INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),  
    nazwa_produktu varchar(255) NOT NULL,  
    ilosc       int NOT NULL,  
    cena        float(10) NOT NULL,  
    wysokosc    int,  
    szerokosc   int,  
    glebokosc   int,  
    masa        int,  
    Kategoria   int NOT NULL,  
    FOREIGN KEY(SklepIdS) REFERENCES Sklepy(IdS),  
    FOREIGN KEY(Kategoria) REFERENCES Kategorie(IdKat));
```

```
CREATE TABLE "Produkty W Zakupie" (  
    IdZakup    INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),  
    IdKlient   int NOT NULL,  
    IdSklep    int NOT NULL,  
    data_zakupu date NOT NULL,  
    godzina_zakupu time NOT NULL,  
    FOREIGN KEY(IdKlient) REFERENCES Klienci(IdKli),  
    FOREIGN KEY(IdSklep) REFERENCES Sklepy(IdS));
```

```
IdN          INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),  
ProduktyIdP  int NOT NULL,  
ZakupId      int NOT NULL,  
ilosc_produktu  int,  
cena_jednostkowa float(10),  
FOREIGN KEY(ProduktyIdP) REFERENCES Produkty(IdP),  
FOREIGN KEY(ZakupId) REFERENCES "Zrobione Zakupy"(IdZakup));
```

```
CREATE UNIQUE INDEX Klienci_IdKli
```

```
ON Klienci (IdKli);
```

```
CREATE UNIQUE INDEX Sklepy_IdS
```

```
ON Sklepy (IdS);
```

```
CREATE UNIQUE INDEX Produkty_IdP
```

```
ON Produkty (IdP);
```

```
CREATE UNIQUE INDEX Kategorie_IdKat
```

```
ON Kategorie (IdKat);
```

```
CREATE UNIQUE INDEX "Typy Klientow_nazwa_typu"
```

```
ON "Typy Klientow" (nazwa_typu);
```

4.

Object Explorer: DESKTOP-1KVKMNB (SQL Server 15.0.2000.5 - root)

- Databases
 - System Databases
 - Database Snapshots
 - AdventureWorks2019
 - AdventureWorksDW2019
 - LABUslugi
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Kategorie
 - dbo.Klienci
 - dbo.Produkty
 - dbo.Produkty W Zakupie
 - dbo.Sklepy
 - dbo.Typy Klientow
 - dbo.Zrobione Zakupy
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Always On High Availability
 - Management
 - Integration Services Catalogs
 - SQL Server Agent (Agent XPs disabled)
 - XEvent Profiler

SQLQuery1.sql - DES...BUslugi (root (59))

```

FOREIGN KEY(Kategoria) REFERENCES Kategorie(IdKat));

CREATE TABLE "Produkty W Zakupie" (
    IdN          INTEGER NOT NULL PRIMARY KEY IDENTITY(1,1),
    ProduktIdP   int NOT NULL,
    ZakupId      int NOT NULL,
    ilosc_produktu int,
    cena_jednostkowa float(10),
    FOREIGN KEY(ProduktIdP) REFERENCES Produkty(IdP),
    FOREIGN KEY(ZakupId) REFERENCES "Zrobione Zakupy"(IdZakup));

CREATE UNIQUE INDEX Klienci_IdKli
    ON Klienci (IdKli);
CREATE UNIQUE INDEX Sklepy_IdS
    ON Sklepy (IdS);
CREATE UNIQUE INDEX Produkty_IdP
    ON Produkty (IdP);
CREATE UNIQUE INDEX Kategorie_IdKat
    ON Kategorie (IdKat);
CREATE UNIQUE INDEX "Typy Klientow_nazwa_typu"
    ON "Typy Klientow" (nazwa_typu);
  
```

100 %

Messages

Commands completed successfully.

Completion time: 2022-03-14T21:43:20.5217905+01:00

5.

Object Explorer: -1KVKMNB (SQL Server 15.0.2000.5 - root)

- Databases
 - System Databases
 - Database Snapshots
 - AdventureWorks2019
 - AdventureWorksDW2019
 - LABUslugi
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Kategorie
 - Columns
 - IdKat (PK, int, not null)
 - nazwa (varchar(255), not null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - dbo.Klienci
 - dbo.Produkty
 - dbo.Produkty W Zakupie
 - dbo.Sklepy
 - dbo.Typy Klientow
 - dbo.Zrobione Zakupy

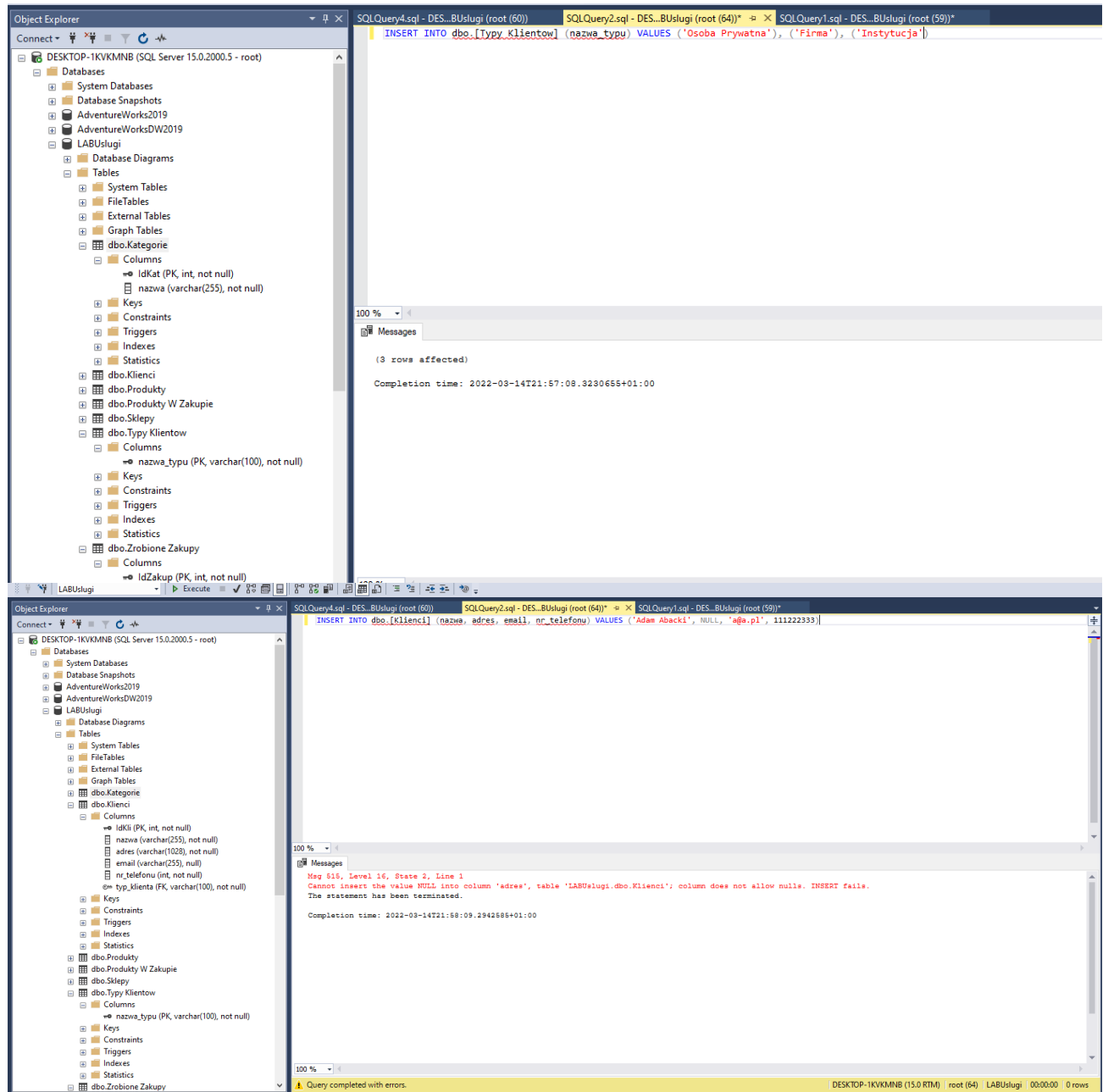
SQL Query: INSERT INTO dbo.Kategorie (nazwa) VALUES ('Karty graficzne'), ('Procesory'), ('Gry Komputerowe')

100 %

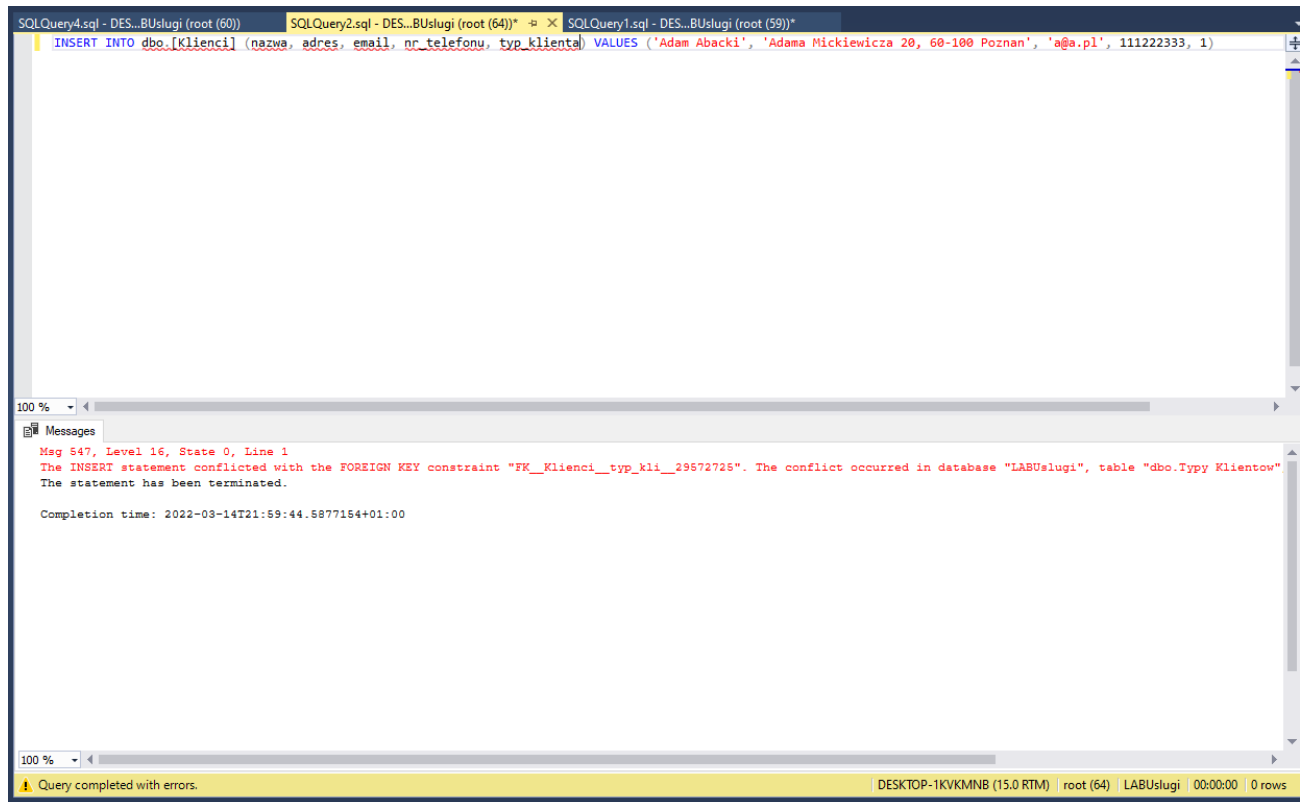
Messages

(3 rows affected)

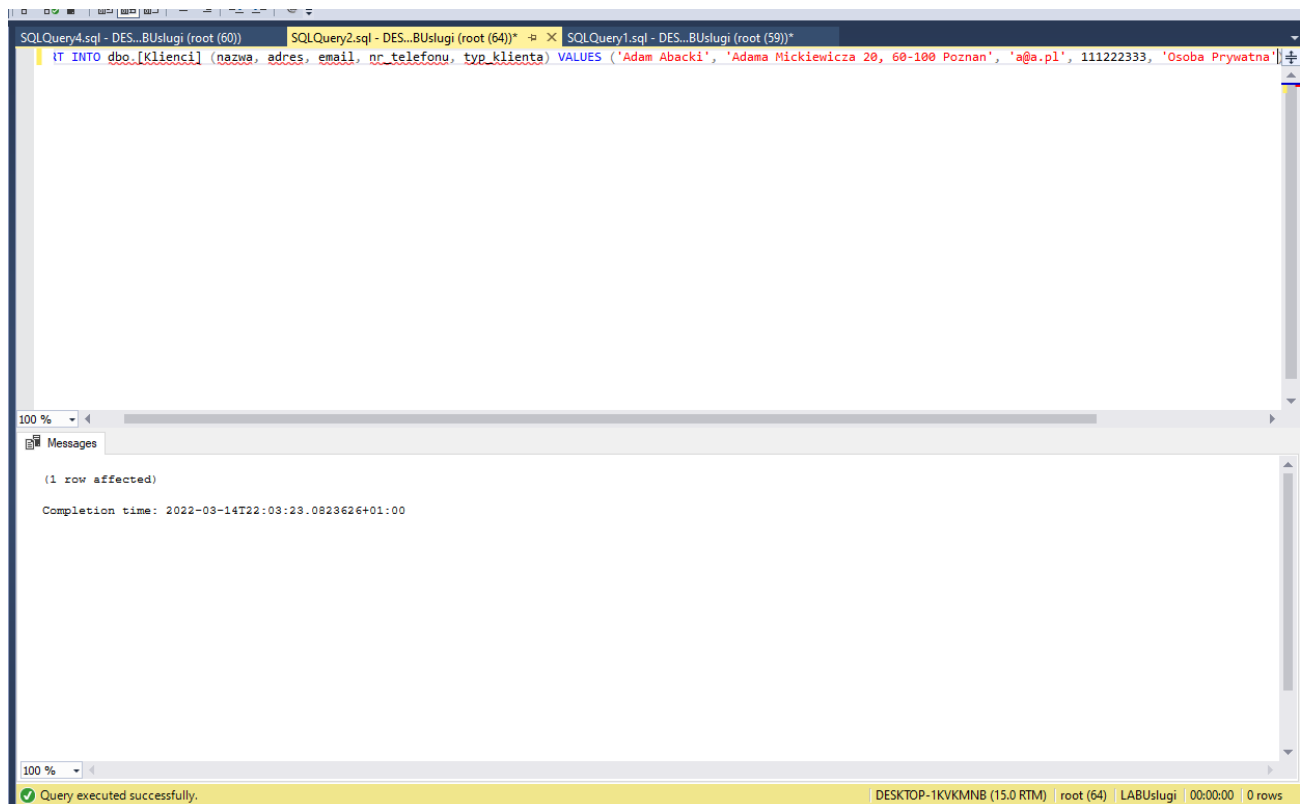
Completion time: 2022-03-14T21:54:07.0051729+01:00



Insert w bazę danych z niewłaściwymi danymi (NULL) nie działa



Niewłaściwy typ klienta wyrzuca błąd constaintów



Wnioski:

Projekt bazy danych, która wykonujemy na Hurtowni Danych, różni się od tych na poprzednich kursach, głównie poprzez skupienie na proces, nie na dane. Musiałem trochę inaczej podejść do projektowania systemu w tym zadaniu.

Dialekt MSSQL ma kilka różnic w stosunku do MySQL, np. zamiast AutoIncrement – IDENTITY. Dodatkowo, Management jest dużo potężniejszym narzędziem niż wszystkie inne, które kiedykolwiek używałem. Miałem problemy z połączeniem z MSSQL z maszyny wirtualnej, więc pewnie w przyszłości bym postawił lokalną kopię, ale na potrzeby zajęć pozostawie go na maszynie.

Uwaga:

- **Sprawozdanie bez wniosków końcowych nie będzie sprawdzane i tym samym ocena jest negatywna!**
- Plik ze sprawozdaniem powinien mieć nazwę **Spr01HD-index-Nazwisko - 2022** i format **pdf**
- Oprogramowanie MS SQL 2017 lub 2019
- Model konceptualny danych należy przygotować wykorzystując narzędzie wspierające język UML np. Visual Paradigm