**W bazie danych znajdują się cztery tabele:**

* czytelnik - posiada informacje o czytelnikach zarejestrowanych w programie (Tabela 1.);
* zamowienie - posiada informacje o zamówionych, wypożyczonych czy oddanych książkach w bibliotece (Tabela 2.);
* kategoria - tabela zawierająca wszystkie kategorie książek w systemie (Tabela 3.);
* ksiazka - wszystkie książki dodane do bazy danych (Tabela 4.);

|  |  |
| --- | --- |
| * Tabela 1. czytelnik | |
| **Nazwa pola:** | **Opis:** |
| id\_czytelnik | Klucz główny przydzielony automatycznie |
| login | Nazwa czytelnika potrzeba przy logowaniu |
| haslo | Hasło niezaszyfrowane |
| imie | Imię |
| nazwisko | Nazwisko |
| adres | Adres zamieszkania np.: ul. Przykład 3/12 |
| miasto | Miasto |
| wojewodztwo | Nazwa województwa |
| telefon | Telefony |
| kod\_pocztowy | Kod pocztowy |
| email | Adres e-mail |

|  |  |
| --- | --- |
| Tabela 2. zamowienie | |
| **Nazwa pola:** | **Opis:** |
| id\_zamowienie | Klucz główny przydzielony automatycznie |
| id\_czytelnik | Klucz obcy z tabeli "czytelnik" |
| id\_ksiazka | Klucz obcy z tabeli "ksiazka" |
| data\_zamowienia | Data złożenia zamówienia w bibliotece |
| data\_odbioru | Data odbioru książki z biblioteki |
| data\_zwrotu | Data zwrotu książki do biblioteki |

|  |  |
| --- | --- |
| Tabela 3. kategoria | |
| **Nazwa pola:** | **Opis:** |
| id\_kategoria | Klucz główny przydzielony automatycznie |
| nazwa | Nazwa kategorii |

|  |  |
| --- | --- |
| Tabela 4. ksiazka | |
| **Nazwa pola:** | **Opis:** |
| id\_ksiazka | Klucz główny przydzielony automatycznie |
| id\_kategoria | Klucz obcy z tabeli "kategoria" |
| isbn | Niepowtarzalny 13-cyfrowy identyfikator książki |
| tytul | Tytuł książki |
| autor | Imię i Nazwisko autora książki |
| stron | Liczba stron książki |
| wydawnictwo | Nazwa wydawnictwa, w którym wydano książkę |
| rok\_wydania | Rok wydania książki |
| opis | Opis książki |

**2. Tworzenie bazy danych**

Po uruchomieniu MSSQLSMS należy utworzyć bazę danych odpowiedzialną za zarządzanie projektem biblioteki (Listing 1.).

**Listing 1. Polecenie tworzące bazę danych**

CREATE DATABASE biblioteka;

**3. Polecenie CREATE**

Po utworzeniu bazy danych należy utworzyć tabele za pomocą polecenia CREATE (listing 2.). Wymagany kod do utworzenia tabel został przedstawiony na listingu 6.

**Listing 2. Składnia polecenia CREATE**

CREATE TABLE nazwa\_tabeli (kolumny)

**Listing 3. Polecenia odpowiedzialne za dodanie wszystkich tabel do bazy danych**

Odpowiedzi

create table czytelnik(

id\_czytelnik int identity(1,1) not null,

login varchar(10) not null,

haslo varchar(20) not null,

imie varchar(20) not null,

constraint pk\_id\_czytelnik primary key(id\_czytelnik)

);

create table kategoria(

id\_kategoria int identity(10,1) not null,

nazwa varchar(100) not null,

constraint pk\_id\_kategoria primary key(id\_kategoria)

);

create table ksiazka(

id\_ksiazka int identity(1,1) not null,

id\_kategoria int not null,

isbn varchar(13) not null,

tytul varchar(100) not null,

autor varchar(70) not null,

stron int not null,

wydawnictwo varchar(50) not null,

rok int not null,

opis varchar(200),

constraint pk\_id\_ksiazka primary key (id\_ksiazka),

constraint fk\_id\_kategoria foreign key(id\_kategoria) references kategoria(id\_kategoria)

);

create table zamowienie(

id\_zamowienie int identity(1,1) not null,

id\_czytelnik int not null,

id\_ksiazka int not null,

data\_zamowienia date not null,

data\_odbioru date not null,

data\_zwrotu date not null,

constraint pk\_id\_zamowienie primary key(id\_zamowienie),

constraint fk\_id\_czytelnik foreign key(id\_czytelnik) references czytelnik(id\_czytelnik),

constraint fk\_id\_ksiazka foreign key(id\_ksiazka) references ksiazka(id\_ksiazka)

);

Przydatne skrypty do tworzenia tabeli

if OBJECT\_ID('dbo.Orders', 'U') is not null drop table dbo.Orders;

create table dbo.Orders(

orderid int not null constraint pk\_orders primary key,

orderdate date not null constraint dft\_orderdate default(sysdatetime()),

empid int not null

)

insert into dbo.Orders(orderid,orderdate,empid) values(10003,'19990503',3);

//lab 4

select \*

from SalesLT.ProductCategory;

insert into SalesLT.ProductCategory(Name) values('Herbs')

select \*

into #prod

from SalesLT.Product

where ListPrice>100

następnie

select \*

from #prod

select ProductID,[Name],ListPrice\*2 as DoubledPrice

into #OverPriced

from SalesLT.Product

następnie

select \*

from #OverPriced

select \*

into SalesLT.ProductV2

from SalesLT.Product

truncate table SalesLT.ProductV2

select \* into SalesLT.ProductV3 --kopiowanie definicji tabeli

from SalesLT.Product

where 1=0 --czyli kopiowanie bez danych

update SalesLT.ProductV3

set ListPrice=ListPrice-ListPrice\*0.1

where SellEndDate is not null --przecena na produkty wyprzedażowe o 10%

select \* from SalesLT.Customer

select FirstName, LastName

into Tab

from SalesLT.Customer

update Tab

set FirstName=LastName, LastName=FirstName

update P

set StandardCost \*= 0.85

from SalesLT.Product as P

join SalesLT.SalesOrderDetail as D

on P.ProductID=D.ProductID

where D.OrderQty>10