

Temat bazy danych

1. Krótki opis bazy danych

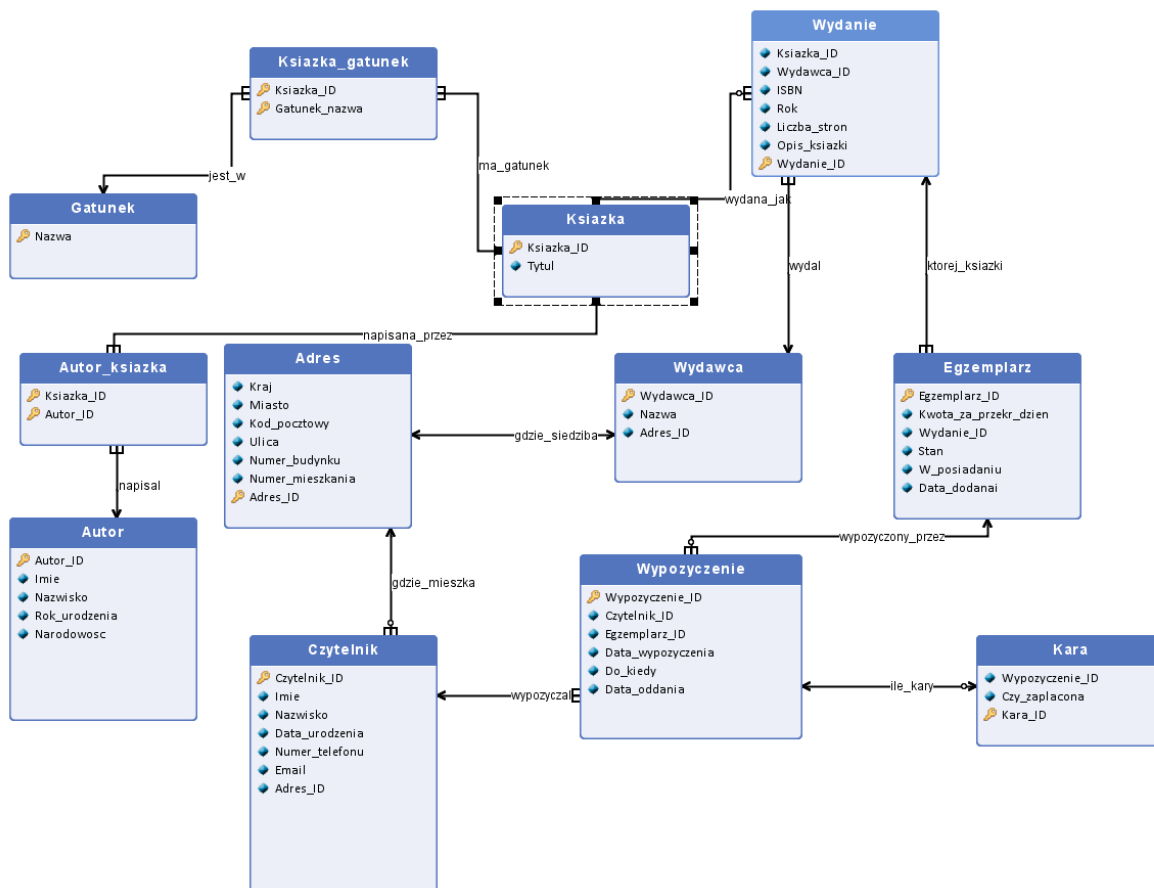
Baza danych służąca do obsługi biblioteki miejskiej.

Klientem jest właściciel miejskiej biblioteki, który zamierza mieć łatwy wgląd w to jakie książki posiada, jakie są dostępne a jakie aktualnie wypożyczone. Chce mieć również wgląd w to co wypożyczają czytelnicy i czy oddają książki w terminie.

Celem bazy jest przechowywanie wszystkich niezbędnych informacji, by ułatwić pracownikom proces wypożyczania czytelnikom książek. Może się również przydać do zautomatyzowania tego procesu.

Klient wymaga, by baza dawała możliwość wyszukiwania książek i filtrowania ich m. in. po gatunku, autorach, ilości wypożyczeń czy roku wydania. Kolejnym wymogiem jest łatwy wgląd w zachowania czytelników. Sprawdzenie średniego czasu oddawania książki, czy kwota jaką czytelnik powinien uregulować za książki oddane po terminie.

2. Schemat graficzny bazy danych (diagram ERD)



3. Opis zbioru encji (typy, klucze, ...)

Wydawca			
Tabela przechowująca zbiór wydawców, którzy wydają wydania. Każdy wydawca ma przydzielony indywidualny identyfikator, który jest kluczem głównym. Encja jest dodawana, gdy dodawane jest wydanie w tabeli wydanie, do której nie ma jeszcze odpowiedniego wydawcy. Encje nie są usuwane. Liczność: ok. 100, roczny przyrost: ok. 10.			
Nazwa	Klucz główny	Typ/Dziedzina	Opis
Wydawca_ID	Tak	INT	ID wydawcy
Wydawca_ID INT NOT NULL PRIMARY KEY IDENTITY,			
Nazwa	Nie	VARCHAR max 128 znaków	Nazwa wydawcy
Nazwa VARCHAR(128)			
Adres_ID	Nie	INT	ID adresu
Adres_ID INT REFERENCES Adres,			

Autor			
Tabela zawierająca informacje o autorach książek. Kluczem głównym jest indywidualny identyfikator nadany autorowi. Nowa encja powstaje, gdy do biblioteki trafi egzemplarz napisany przez autora, którego nie było wcześniej w bazie. Encje nie są usuwane. Liczność: ok. 200, roczny przyrost: ok. 10.			
Nazwa	Klucz główny	Typ/Dziedzina	Opis
Autor_ID	Tak	INT	ID autora
Autor_ID INT NOT NULL PRIMARY KEY IDENTITY,			
Imie	Nie	VARCHAR max 16 liter, pierwsza litera jest wielka	Imię autora
Imie VARCHAR(16),			
Nazwisko	Nie	VARCHAR max 32 litery, pierwsza litera jest wielka	Nazwisko autora
Nazwisko VARCHAR(32),			
Rok_urodzenia	Nie	INT max 4 cyfry	Rok urodzenia autora
Rok_urodzenia INT CHECK (Rok_urodzenia <= 9999),			
Narodowosc	Nie	VARCHAR max 32 znaki	Kraj w którym autor się urodził
Narodowosc VARCHAR(32)			

Gatunek

Zbiór gatunków książek. Kluczem głównym jest tu nazwa gatunku, ponieważ nie powtarzają się one. Nowa encja powstaje gdy w bibliotece pojawia się egzemplarz, który należy opisać gatunkiem, którego jeszcze nie ma. Encje nie są usuwane. Liczność: ok. 50, roczny przyrost: ok. 1.			
Nazwa	Klucz główny	Typ/Dziedzina	Opis
Nazwa	Tak	VARCHAR max 30 znaków	Nazwa gatunku książki
Nazwa VARCHAR(30) NOT NULL PRIMARY KEY			

Czytelnik			
Tabela zawiera zbiór użytkowników biblioteki, którzy wypożyczają książki. Każdy czytelnik otrzymuje indywidualne ID, które jest kluczem głównym. Nowa encja jest dodawana, gdy nowy czytelnik zarejestruje się w bibliotece. Encje nie są usuwane. Roczny przyrost: ok. 200.			
Nazwa	Klucz główny	Typ/Dziedzina	Opis
Czytelnik_ID	Tak	INT	ID czytelnika
Czytelnik_ID INT NOT NULL PRIMARY KEY IDENTITY,			
Imie	Nie	VARCHAR max 16 liter, pierwsza litera jest wielka	Pierwsze imię czytelnika
Imie VARCHAR(16) NOT NULL,			
Nazwisko	Nie	VARCHAR max 32 litery, pierwsza litera jest wielka	Nazwisko czytelnika
Nazwisko VARCHAR(32) NOT NULL,			
Data_urodzenia	Nie	DATE mm/dd/yyyy	Data urodzenia czytelnika
Data_urodzenia DATE,			
Numer_telefonu	Nie	CHAR 11 znaków, 9 cyfr, 2 -, cyfry zgrupowane po 3 rozdzielone (-)	Numer telefonu czytelnika
Numer_telefonu CHAR(11) NOT NULL CHECK (Numer_telefonu like '[0-9][0-9][0-9]-[0-9][0-9][0-9]-[0-9][0-9][0-9]'),			
Email	Nie	VARCHAR max 30 znaków, zawiera znak @ i zakończony sformułowaniem (.) i domena	Adres email czytelnika (opcjonalny)
Email VARCHAR(100) CHECK (Email like '%@%')			
Adres_ID	Nie	INT	ID adresu
Adres_ID INT NOT NULL REFERENCES Adres,			

Autor_ksiązka			
Zbiór encji łączący tabelę Autor i Książka. Klucz główny jest złożony z Książka_ID i Autor_ID. Nowa encja powstaje, gdy pojawia się nowa encja w tabeli książka. Encje nie są usuwane. Liczność: ok. 2000, roczny przyrost: ok. 100.			
Nazwa	Klucz główny	Typ/Dziedzina	Opis
Książka_ID	Tak	INT	ID książki

Ksiazka_ID INT NOT NULL REFERENCES Ksiazka ON DELETE CASCADE,
PRIMARY KEY (Ksiazka_ID, Autor_ID),

Autor_ID	Tak	INT	ID autora
----------	-----	-----	-----------

Autor_ID INT NOT NULL REFERENCES Autor ON DELETE CASCADE,

Wydanie

Tabela zawiera zbiór wydań które były/są dostępne w bibliotece. Wydanie to partia książek, która może, lecz nie musi posiadać ISBN. Dlatego kluczem głównym jest id książki z id wydawcy. Encja powstaje, gdy do biblioteki trafi egzemplarz książki nowego wydania. Encje nie są usuwane.
Liczność: ok. 2000, roczny przyrost: ok. 60.

Nazwa	Klucz główny	Typ/Dziedzina	Opis
Wydanie_ID	Tak	INT	ID wydania
Wydanie_ID INT NOT NULL PRIMARY KEY IDENTITY,			
Ksiazka_ID	Nie	INT	ID dzieła literackiego
Ksiazka_ID INT NOT NULL REFERENCES Ksiazka ON DELETE CASCADE,			
Wydawca_ID	Nie	INT	ID wydawcy
Wydawca_ID INT NOT NULL REFERENCES Wydawca ON DELETE CASCADE,			
ISBN	Nie	CHAR 13 cyfr rozdzielonych znakami (-)	Międzynarodowy numer wydania
ISBN CHAR(20),			
Rok	Nie	INT max 4 cyfry	Rok wydania
Rok INT CHECK (Rok < 10000),			
Liczba_stron	Nie	INT max 6	Liczba stron
Liczba_stron INT CHECK (Liczba_stron < 1000000),			
Opis_książki	Nie	VARCHAR max 2000	Krótki opis wydania
Opis_książki VARCHAR(2000)			

Ksiazka_gatunek

Tabela zawiera informacje na temat gatunków książek. Kluczem głównym jest tu klucz złożony z kluczy głównych tabel gatunek i ksiazka, t.j. nazwa_gatunku i ksiazka_id. Nowa encja jest tworzona w momencie dodawania książki. Encje nie są usuwane. Liczność: ok. 2000, roczny przyrost: ok. 100.

Nazwa	Klucz główny	Typ/Dziedzina	Opis
Ksiazka_ID	Tak	INT	ID książki
Ksiazka_ID INT NOT NULL REFERENCES Ksiazka ON DELETE CASCADE,			
Gatunek_nazwa	Tak	VARCHAR(30)	ID gatunku
Gatunek_nazwa VARCHAR(30) NOT NULL REFERENCES Gatunek ON DELETE CASCADE ON UPDATE CASCADE, PRIMARY KEY (Ksiazka_ID, Gatunek_nazwa)			

Wypożyczenie			
Tabela zawiera informację o tym, kto i kiedy, wypożyczył jaką książkę. Kluczem głównym jest indywidualny ID. Encja powstaje w momencie, gdy ktoś wypożycza książkę. Encje nie są usuwane. Roczny przyrost: ok. 5000.			
Nazwa	Klucz główny	Typ/Dziedzina	Opis
Wypożyczenie_ID	Tak	INT	ID wypożyczenia
Wypożyczenie_ID INT NOT NULL PRIMARY KEY IDENTITY,			
Czytelnik_ID	Nie	INT	ID czytelnika
Czytelnik_ID INT NOT NULL REFERENCES Czytelnik,			
Egzemplarz_ID	Nie	INT	ID egzemplarza
Egzemplarz_ID INT NOT NULL REFERENCES Egzemplarz,			
Data_wypożyczenia	Nie	DATE yyyy/mm/dd	Data wypożyczenia egzemplarza
Data_wypożyczenia DATE,			
Do_kiedy	Nie	DATE yyyy/mm/dd	Data do której należy zwrócić egzemplarz by nie została naliczona opłata
Do_kiedy DATE,			
Data_oddania	Nie	DATE yyyy/mm/dd	Data w której czytelnik zwrócił egzemplarz
Data_oddania DATE,			

Egzemplarz			
Tabela przechowująca informacje o egzemplarzach, które były lub są dostępne do wypożyczenia. Każda encja ma przydzielone ID, które jest kluczem głównym. Encja jest tworzona gdy do biblioteki trafia nowy egzemplarz. Gdy egzemplarz przestaje być wypożyczany, encja nie jest usuwana, lecz zmienia się wartość atrybutu "W_posiadaniu". Liczność: ok. 5000, roczny przyrost: ok. 200.			
Nazwa	Klucz główny	Typ/Dziedzina	Opis
Egzemplarz_ID	Tak	INT	ID egzemplarza
Egzemplarz_ID INT NOT NULL PRIMARY KEY IDENTITY,			
Kwota_za_przekr_dzien	Nie	DECIMAL do 4 miejsc przed przecinkiem i 2 po	Kwota, którą będzie musiał zapłacić czytelnik za każdy przekroczony dzień
Kwota_za_przekr_dzien DECIMAL(4,2),			
Wydanie_ID	Nie	INT	ID wydania
Wydanie_ID INT NOT NULL REFERENCES Wydanie ON DELETE CASCADE,			
Stan	Nie	INT od 1 do 10	Ocena wyglądu egzemplarza w skali od 1 do 10

Stan INT CHECK (Stan >0 AND Stan <=10),			
W_posiadaniu	Nie	BIT	Jeżeli egzemplarz jest w posiadaniu biblioteki i dalej jest wypożyczana, to przyjmuje wartość TRUE
W_posiadaniu BIT,			
Data_dodania	Nie	DATE yyyy/mm/dd	Data w której egzemplarz trafił do biblioteki
Data_dodania DATE			

Książka			
Tabela przechowująca książki, jako dzieła literackie, nie pojedyncze egzemplarze. Encje mają przydzielony indywidualny identyfikator. Kluczem nie może być tytuł i autor, ponieważ może zdarzyć się powtórzenie. Encja jest dodawana, gdy biblioteka pozyska egzemplarz nowej książki. Encje nigdy nie są usuwane. Liczność: ok. 1000, roczny przyrost: ok. 50.			
Nazwa	Klucz główny	Typ/Dziedzina	Opis
Książka_ID	Tak	INT	ID dzieła literackiego
Książka_ID INT NOT NULL PRIMARY KEY IDENTITY,			
Tytuł	Nie	VARCHAR max 200 znaków	Tytuł dzieła literackiego
Tytuł VARCHAR(200)			

Kara			
Tabela zawiera informacje o naliczonych karach. Kluczem głównym jest tutaj sztucznie utworzone indywidualne id. Encja jest tworzona, gdy ktoś odda książkę po terminie. Encje nie są usuwane. Roczny przyrost: ok. 300.			
Nazwa	Klucz główny	Typ/Dziedzina	Opis
Wypożyczenie_ID	Nie	INT	ID wypożyczenia
Wypożyczenie_ID INT NOT NULL REFERENCES Wypożyczenie ON DELETE CASCADE,			
Czy_zapłacona	Nie	BIT	Przyjmuje wartość 1, gdy kara, naliczona za oddanie po terminie, zostanie opłacona
Czy_zapłacona BIT,			
Kara_ID	Tak	INT	ID kary
Kara_ID INT NOT NULL PRIMARY KEY IDENTITY,			

Adres			
Zbiór adresów. Zapobiega redundancji adresów dla wydawców i czytelników. Kluczem głównym jest id przydzielony do encji. Nowa encja powstaje gdy nie ma w bazie adresu, którego potrzeba. Encje nie są usuwane. Roczny przyrost: ok. 50.			
Nazwa	Klucz główny	Typ/Dziedzina	Opis
Kraj	Nie	VARCHAR max 64 znaki	Nazwa kraju

Kraj VARCHAR(64),			
Miasto	Nie	VARCHAR max 64	Nazwa miasta
Miasto VARCHAR(64),			
Kod_pocztowy	Nie	CHAR(6) typu „xx-xxx” gdzie x to cyfry 0 do 9	Kod pocztowy miasta
Kod_pocztowy CHAR(6) CHECK (Kod_pocztowy like '[0-9][0-9]-[0-9][0-9][0-9]'),			
Ulica	Nie	VARCHAR max 64 znaki	Nazwa ulicy
Ulica VARCHAR(64),			
Numer_budynku	Nie	INT max 1000	Numer budynku
Numer_budynku INT CHECK (Numer_budynku <= 1000),			
Numer_mieszkania	Nie	INT max 1000	Numer mieszkania(jeśli inny niż budynku)
Numer_mieszkania CHAR(6)			
Adres_ID	Tak	INT	ID adresu
Adres_ID INT NOT NULL PRIMARY KEY IDENTITY,			

4. Schemat relacyjnej bazy danych

```
Table Ksiazka {
  Ksiazka_ID int pk
  Tytul varchar
}
```

```
Table Ksiazka_gatunek {
  Ksiazka_ID int pk
  Gatunek_nazwa int pk
}
```

```
Table Gatunek {
  Nazwa varchar pk
}
```

ref: Ksiazka_gatunek.Ksiazka_ID > Ksiazka.Ksiazka_ID

ref: Ksiazka_gatunek.Gatunek_nazwa > Gatunek.Nazwa

```
Table Wydanie {
  Ksiazka_ID int pk
```

```
Wydawca_ID int pk
ISBN char(13)
Rok int
Liczba_stron int
Opis_ksiazki varchar
}
```

```
Table Wydawca {
  Wydawca_ID int pk
  Adres_ID int
  Nazwa varchar
}
```

ref: Wydanie.Ksiazka_ID > Ksiazka.Ksiazka_ID

ref: Wydanie.Wydawca_ID > Wydawca.Wydawca_ID

```
Table Adres {
  Adres_ID int pk
  Kraj varchar
  Miasto varchar
  Kod_pocztowy varchar
  Ulica varchar
  Numer_budynku int
  Numer_mieszkania int
}
```

ref: Wydawca.Adres_ID > Adres.Adres_ID

```
Table Egzemplarz {
  Egzemplarz_ID int pk
  Wydanie_ID int
  Kwota_za_przekr_dzien decimal
  Stan int
  W_posiadaniu boolean
  Data_dodania date
}
```

ref: Egzemplarz.Wydanie_ID > Wydanie.Ksiazka_ID

ref: Egzemplarz.Wydanie_ID > Wydanie.Wydawca_ID


```
Table Wypozyczenie {  
  Wypozyczenie_ID int pk  
  Czytelnik_ID int  
  Egzemplarz_ID int  
  Data_wypozyczenia date  
  Do_kiedy date  
  Data_oddania date  
}
```

```
Table Czytelnik {  
  Czytelnik_ID int pk  
  Adres_ID int  
  Imie varchar  
  Nazwisko varchar  
  Data_urodzenia date  
  Numer_telefonu char(11)  
  Email varchar  
  Do_zaplaty float  
}
```

ref: Wypozyczenie.Czytelnik_ID > Czytelnik.Czytelnik_ID

ref: Wypozyczenie.Egzemplarz_ID > Egzemplarz.Egzemplarz_ID

ref: Czytelnik.Adres_ID > Adres.Adres_ID

```
Table Kara {  
  Kara_ID int pk  
  Wypozyczenie_ID int  
  Czy_zaplacona boolean  
}
```

ref: Kara.Wypozyczenie_ID > Wypozyczenie.Wypozyczenie_ID

```
Table Autor {  
  Autor_ID int pk  
  Imie varchar  
  Nazwisko varchar  
  Rok_urodzenia int  
  Narodowosc varchar  
}
```

```
Table Autor_ksiazka {
  Ksiazka_ID int pk
  Autor_ID int pk
}
```

ref: Autor_ksiazka.Ksiazka_ID > Ksiazka.Ksiazka_ID

ref: Autor_ksiazka.Autor_ID > Autor.Autor_ID

5. Szczegółowy opis utworzonych tabel pod kątem zastosowanych ograniczeń np. NOT NULL, UNIQUE, CHECK, DEFAULT, klucze ...

Jedynym możliwym update to update nazwy gatunku. W każdym innym przypadku kluczem głównym jest int z identity, co wyklucza używanie update.

Not null używam przy pk jako zabezpieczenie, mimo identity

```
CREATE TABLE Adres (
  Adres_ID INT NOT NULL PRIMARY KEY IDENTITY,
  Kraj VARCHAR(64),
  Miasto VARCHAR(64),
  Kod_pocztowy CHAR(6) CHECK (Kod_pocztowy like '[0-9][0-9]-[0-9][0-9][0-9]'),
  Ulica VARCHAR(64),
  Numer_budynku INT CHECK (Numer_budynku <= 1000),
  Numer_mieszkania CHAR(6)
);
```

- NOT NULL w Adres_ID ponieważ jest to klucz główny sztuczny z identity
- CHECK (Kod_pocztowy like '[0-9][0-9]-[0-9][0-9][0-9]') w Adres.Kod_pocztowy by wymusić polski kod pocztowy
- CHECK (Numer_budynku <= 1000), Adres.Numer_budynku musi być mniejszy niż 1000

```
CREATE TABLE Wydawca (
  Wydawca_ID INT NOT NULL PRIMARY KEY IDENTITY,
  Adres_ID INT REFERENCES Adres,
  Nazwa VARCHAR(128)
);
```

```
CREATE TABLE Ksiazka (
  Ksiazka_ID INT NOT NULL PRIMARY KEY IDENTITY,
  Tytul VARCHAR(200)
);
```

```
CREATE TABLE Gatunek (
    Nazwa VARCHAR(30) NOT NULL PRIMARY KEY
);
```

```
CREATE TABLE Ksiazka_gatunek (
    Ksiazka_ID INT NOT NULL REFERENCES Ksiazka ON DELETE CASCADE,
    Gatunek_nazwa VARCHAR(30) NOT NULL REFERENCES Gatunek ON DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY (Ksiazka_ID, Gatunek_nazwa)
);
```

```
CREATE TABLE Autor (
    Autor_ID INT NOT NULL PRIMARY KEY IDENTITY,
    Imie VARCHAR(16),
    Nazwisko VARCHAR(32),
    Rok_urodzenia INT CHECK (Rok_urodzenia <= 9999),
    Narodowosc VARCHAR(32)
);
```

Rok_urodzenia ograniczony do 4 cyfr za pomocą check

```
CREATE TABLE Wydanie (
    Wydanie_ID INT NOT NULL PRIMARY KEY IDENTITY,
    Ksiazka_ID INT NOT NULL REFERENCES Ksiazka ON DELETE CASCADE,
    Wydawca_ID INT NOT NULL REFERENCES Wydawca ON DELETE CASCADE,
    ISBN CHAR(20),
    Rok INT CHECK (Rok < 10000 ),
    Liczba_stron INT CHECK (Liczba_stron < 1000000),
    Opis_ksiazki VARCHAR(2000)
);
```

Maksymalny rok do 10000 za pomocą check. (Myślenie przyszłościowe)

Ograniczenie maksymalnej liczby stron do 1000000

```
CREATE TABLE Egzemplarz (
    Egzemplarz_ID INT NOT NULL PRIMARY KEY IDENTITY,
    Wydanie_ID INT NOT NULL REFERENCES Wydanie ON DELETE CASCADE,
    Kwota_za_przechr_dzien DECIMAL(4,2),
    Stan INT CHECK (Stan >0 AND Stan <=10),
    W_posiadaniu BIT,
    Data_dodania DATE
);
```

Wymuszenie dla stanu od 1 do 10

Not null przy kluczu obcym, ponieważ egzemplarz musi mieć wydanie.

```

CREATE TABLE Czytelnik (
    Czytelnik_ID INT NOT NULL PRIMARY KEY IDENTITY,
    Adres_ID INT NOT NULL REFERENCES Adres,
    Imie VARCHAR(16) NOT NULL,
    Nazwisko VARCHAR(32) NOT NULL,
    Data_urodzenia DATE,
    Numer_telefonu CHAR(11) NOT NULL CHECK (Numer_telefonu like '[0-9][0-9][0-9]-[0-9][0-9][0-9]-[0-9][0-9][0-9]'),
    Email VARCHAR(100) CHECK (Email like '%@%')
);

```

Wymuszenie telefonu w formacie 'xxx-xxx-xxx' gdzie x to cyfra

Wymuszenie maila z @ za pomocą check (Email like '%@%')

Not null przy imieniu, nazwisku i numerze telefonu by można było na nich robić selecta.

Zakładam, że pozostanie czytelnikiem biblioteki wymaga posiadanie imienia, nazwiska i numeru telefonu.

```

CREATE TABLE Wypozyczenie (
    Wypozyczenie_ID INT NOT NULL PRIMARY KEY IDENTITY,
    Czytelnik_ID INT NOT NULL REFERENCES Czytelnik,
    Egzemplarz_ID INT NOT NULL REFERENCES Egzemplarz,
    Data_wypozyczenia DATE,
    Do_kiedy DATE,
    Data_oddania DATE,
);

```

Not null przy czytelnik id, ponieważ nie może być takiego wypożyczenia, którego nikt nie dokonał.

Not null przy egzemplarz id, ponieważ nie może być wypożyczenia w którym nie jest wypożyczona żadna książka.

```

CREATE TABLE Kara (
    Kara_ID INT NOT NULL PRIMARY KEY IDENTITY,
    Wypozyczenie_ID INT NOT NULL REFERENCES Wypozyczenie ON DELETE CASCADE,
    Czy_zaplacona BIT,
);

```

Not null przy wypozyczenie id w karze, ponieważ kara naliczona jest do wypożyczenia

```

CREATE TABLE Autor_książka (
    Autor_ID INT NOT NULL REFERENCES Autor ON DELETE CASCADE,
    Książka_ID INT NOT NULL REFERENCES Książka ON DELETE CASCADE,
    PRIMARY KEY (Książka_ID, Autor_ID),
);

```

Not null przy autor_id ponieważ nie może być encji autor_książka bez autora

Not null przy książka_ID, ponieważ nie może być tej encji bez książki.