

Jegyzőkönyv

Adatkezelés XML-ben

I. beadandó feladat

Gyakorlatvezető: Agárdi Anita, tanszéki mérnök



1. A feladat ismertetése

A feladat elvégzéseként egy adott könyvtári rendszer ER-modelljéből tervezek és készítek XDM-modellt, amelyből a feladat utolsó lépéseként XMLSchema formátumú dokumentum készül.

1.1 Az ER-modell egyedeinek bemutatása

Egyed megnevezése	Egyed megnevezése a modellben	Attribútumok (PK – Primary key, M – Multivalued, C – Composite, O – Optional, D – Derived) ¹
Olvasó	<i>Olvaso</i>	Könyvtari_szam ^(PK) Vezeteknev Keresztnev Beir_datum Foglalkozas Beiratkozott Telefonszam ^(O) Lakcim ^(C) { Orszag Ir_szam Varos Utca Hazszam
Könyv	<i>Konyv</i>	ISBN ^(PK) Cím Kiad_datum
Példány	<i>Peldany</i>	Lelt_szam ^(PK) Ar Kolcs_e
Szerző	<i>Szerzo</i>	Szerzo_azon ^(PK) Vezeteknev Keresztnev
Kiadó	<i>Kiado</i>	Kiado_azon ^(PK) Nev Cim ^(C) { Orszag Ir_szam Varos

Az **Olvasó** egyed tulajdonságai egyértelműen következnek. Elsődleges kulcsa a könyvtári szám [*Könyvtari_szam*], amelyet a könyvtár ad beiratkozáskor. A Foglalkozás [*Foglalkozas*], Vezeték- és keresztnév [*Vezeteknev*] és [*Keresztnev*], illetve Lakcím [*Lakcim*] egyértelmű, melyek közül utóbbi összetett adat. A Lakcím elem öt másik adatot tartalmaz (ország [*Orszag*], irányítószám [*Ir_szam*], város [*Varos*], utca [*Utca*], házszám [*Hazszam*]). A beiratkozási dátum [*Beir_datum*] dátum formátumú, az adatok

¹ Magyar megfelelőik: PK – Elsődleges kulcs, M – Többértékű, C – Összetett, O – Opcionális, D – Származtatott

felvitelének időpontját tartalmazza. A *[Beiratkozott]* egy igaz-hamis típus, amely megadja, hogy az olvasó beiratkozott-e az adott félvére.

A **Könyv** egyed elsődleges kulcsa az ISBN szám, amely egy nemzetközi jelölés. Ezen kívül szükség van még a **Könyv** címére *[Cim]*, illetve kiadásának idejére *[Kiad_datum]*.

A **Példány** egyed egy adott könyv, amelyből több is lehet, ezáltal saját kulcsot kap; ez lesz a leltári szám *[Lelt_szam]*. Mivel ezek minőségüket tekintve eltérőek lehetnek, árazva vannak *[Ar]*. Mindemellett látható, hogy az adott példány kölcsönzött-e *[Kolcs_e]* éppen.

A **Szerző** egyed kulcsa az azonosítója *[Szerzo_azon]*, illetve Vezeték- és keresztnév *[Vezeteknev]* és *[Keresztnev]* egyértelműen szükségesek.

A **Kiadó** egyed a könyv kiadóját adja meg. Azonosítása a kiadó azonosító *[Kiado_azon]* alapján történik, ez lesz az egyed elsődleges kulcsa. A kiadóknál a kiadó nevét *[Nev]* és a címet *[Cim]* emeljük ki, ahol kiadták.

1.2 Az ER modell relációinak bemutatása

Reláció megnevezése	Reláció megnevezése a modellben	Reláció iránya, számossága	Attribútumok (PK – Primary key, M – Multivalued, C – Composite, O – Optional, D – Derived)
Előjegyzés	<i>Elojegyzes</i>	Olvaso ^(N) ↔ ^(M) Konyv	Eloj_datum
Kölcsönzés	<i>Kolcsonzes</i>	Olvaso ⁽¹⁾ ↔ ^(N) Peldany	Kolcs_datum
Kiadta	<i>Kiadta</i>	Kiado ⁽¹⁾ ↔ ^(N) Konyv	
Írta	<i>Irta</i>	Szerzo ^(N) ↔ ^(M) Konyv	

A relációk is többnyire egyértelműek, mivel logikus, hogy egy olvasó ki tudja-e egy adott könyvnek egy példányát (hiszen egy könyvből több is lehet egy könyvtárban) kölcsönözni *[Kolcsonzes]*, amelynek mi az idejére vagyunk kíváncsiak. Ezért tulajdonsága a kölcsönzési dátum *[Kolcs_datum]*. Másrészt amennyiben egy könyvnek egyetlen példánya sincs a könyvtárban (mert nem tartják, vagy mind éppen kölcsönzés alatt áll), akkor előjegyezhetnek rá *[Elojegyzes]*, hogy elsőbbséggel juthassanak az adott könyv egy példányához. Az előjegyzésnél is a dátumra van szükségünk *[Eloj_datum]*, hogy egy esetleges sorrendet állíthassunk fel.

A relációk számosságát és irányát tekintve:

az **Előjegyzés** *[Elojegyzes]* kétirányú és mindkét irányba többszörös (N–M) kapcsolat, mivel egy olvasó tud több könyvet is előjegyezni, illetve ugyanazt a könyvet több olvasó is elő tudja jegyezni. a **Kölcsönzés** *[Kolcsonzes]* szintúgy kétirányú, de (1–N) kapcsolat, mert egy olvasó kikölcsönözhet több könyvet is, de ugyanazt a könyvet (példányt) nem tudja egyszerre több ember kikölcsönözni.

a **Kiadta** [*Kiadta*] megint kétirányú és (1–N) kapcsolat, mivel egy kiadó több könyvet is kiadhat, de egy könyvet csak egy kiadó adhat ki.

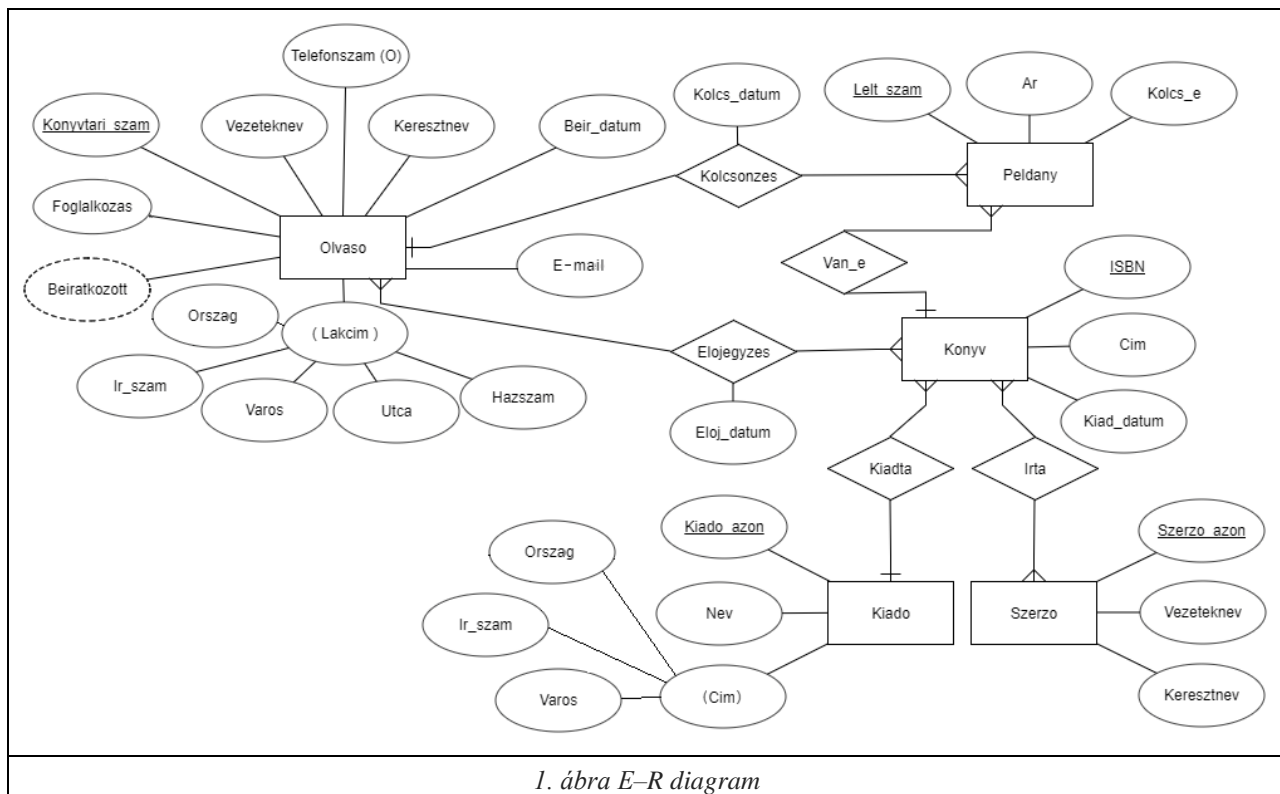
az **Írta** [*Írta*] kapcsolat kétirányú és mindkét irányba többszörös (N–M) kapcsolat, mert egy szerző írhat több könyvet, illetve egy könyvet írhat több szerző is.

2. Az adatbázis modellezése

Az adatbázisról először egy úgynevezett *E–R modell* [*Entity–Relationship model*] készült, amely az egyedek és kapcsolataik leírására szolgáló grafikus diagrammodell.

Majd ezen E–R diagramból *XDM-modell* készült, amely egy hierarchikus sémájú modell, hogy az XMLSchema-t könnyebben tudjuk elkészíteni.

2.1 Az E–R modell



2.2 Az E–R modell átalakítása XDM modellre

Első lépésként ki kell jelölni egy gyökérelemet, amely az egész adatmodell szülőeleme lesz. Mivel egyik elemünk sem felelne meg, így egy virtuális elemet hozunk létre, amely az egész hierarchikus struktúrát magába foglalja. Nevezzük ezt az elemet egyszerűen gyökérnek (<ROOT>).

Második lépés, hogy az összes elemhez egy befoglaló elemet hozunk létre, amellyel az adott elemcsoport létezését kívánjuk megadni. Ennek oka, hogy az ER-modell számosságot nem határoz meg semmilyen módon, de nekünk szükséges tudatnunk, hogy egy adathalmaz létezik még akkor is, ha jelenleg egyedünk nincsen benne.

Végezetül harmadik lépésként az ER-modell elemeit XDM modellbeli elemekké konvertáljuk.

2.2.1 Az egyedek átkonvertálása

Az **Olvasó** [*Reader*] egyed a **Readers** gyűjtő szülő gyermeke lesz és a következőket tartalmazza: a vezetéknév [*Last name*] és keresztnév [*First name*] egyetlen mezőbe kerül, a név [*Name*] mezőbe, a lakcím [*Address*] mező, mivel összetett adat, így szülőként kerül beépítésre a modellbe, alelemekkel, melyek: [*Country*], [*ZIP code*], [*City*]. Az utca és házszám [*Street, Number*] bővítéssel lesz hozzátoldva. A Beiratkozott [*Registered?*] mező üres tulajdonságként lesz jelen, illetve a Foglalkozás [*Occupation*], Telefonszám [*Telephone*], E-mail [*E-mail*], és Beiratkozási dátum [*Date of enrollment*] mezők egyszerű entitásként lesznek beépítve.

Külön kulcs attribútuma lesz a könyvtári azonosító [*ID*].

A **Könyv** [*Book*] egyed a **Books** gyűjtő szülő gyermeke lesz és a következőket tartalmazza: cím [*Title*] és kiadás ideje [*Date of publishing*] adatokat. Attribútumként pedig megjelenik egy saját kulcs attribútum [*ISBN*], illetve két új idegen kulcs [*Author, Publisher*], amely majd a kiadási és szerzői adatokat adja meg.

A **Példány** [*Copy*] egyed a **Books** egy alentitása lesz, amely a **Copies** szülő gyermeke és az árat [*Price*], illetve a kikölcsönzöttség állapotát [*Borrowed?*] tartja nyilván.

Attribútumként még egy kulcselem, a leltári szám [*Accession number*] kerül bele, amely a könyv egyeddel tart kapcsolatot.

A **Szerző** [*Author*] egyednél a cím [*Address*] bővítetlen, a vezetéknév [*Last name*] és keresztnévet [*First name*] tartalmazza, illetve attribútumként a szerző azonosítót [*Author ID*]

A **Kiadó** [*Publisher*] egyednél szintúgy nincs alakítás, a kiadó nevét [*Name*] és székhelyét [*City*] tartalmazza, attribútumként pedig a szerzőnél is látott azonosítót [*Publisher ID*].

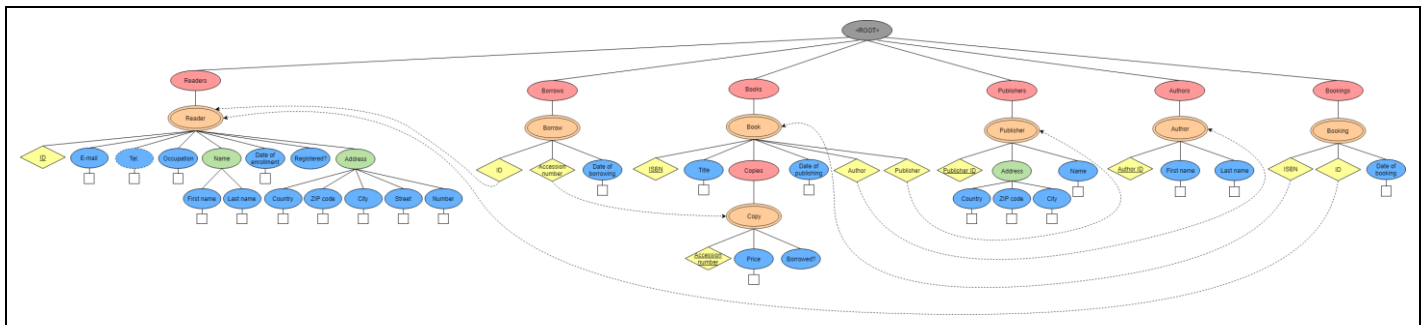
2.2.2 A kapcsolatok átkonvertálása

Az **Előjegyzés** [*Booking*] relációnk külön gyűjtőbe kerül, mégpedig a **Bookings**-ba úgy, hogy attribútumként két idegen kulcsot is tartalmaz. Ezek az ISBN [*ISBN*] illetve a könyvtári azonosító [*ID*]. Az előjegyzési dátum [*Date of booking*] mező egyszerű elemként kerül fel.

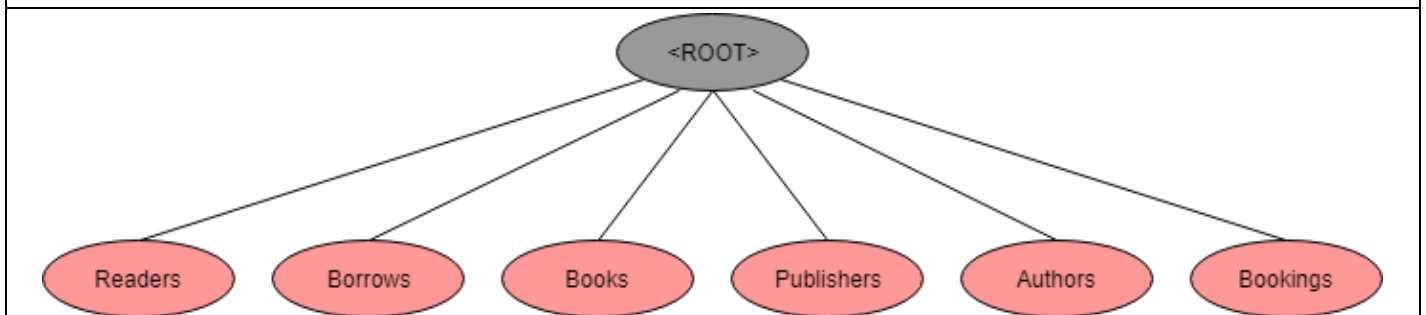
A **Kölcsönzés** [*Borrow*] hasonlóan úgy kerül külön elemként a modellbe, hogy szülője a **Borrows**, attribútumai a könyvtári azonosító [*ID*] az Olvasóval, a leltári szám [*Accession number*] pedig a könyvek példányával van relációban.

A **Kiadta és Írta** [-] reláció nem lesz külön elem, csupán idegen kulcsokkal tartják a kapcsolatot a többi elemmel.

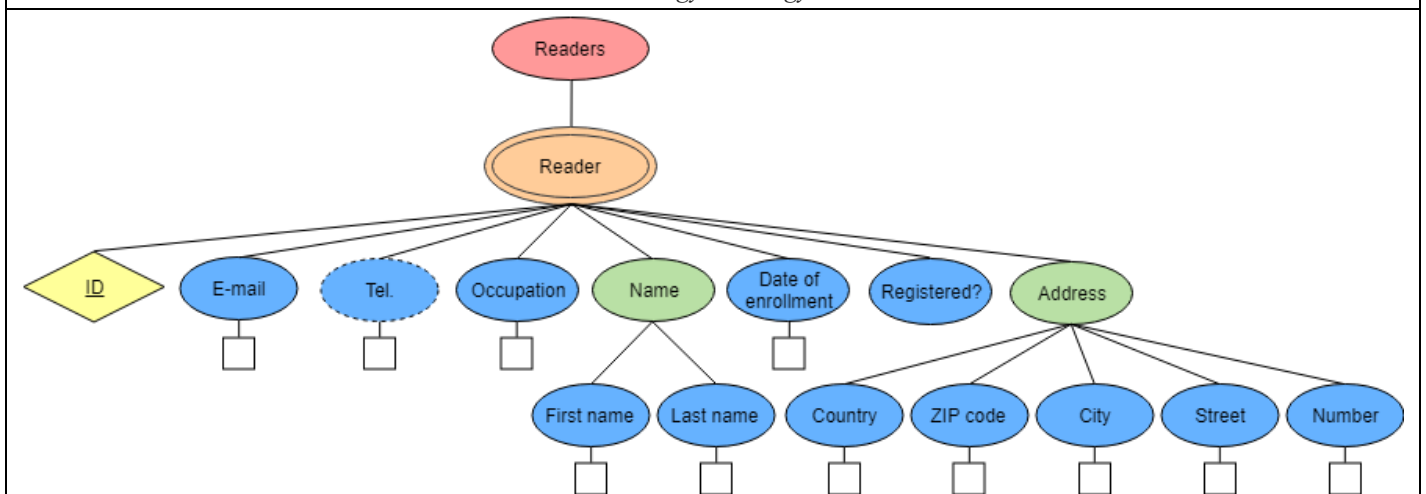
2.3 Az XDM-modell és részei



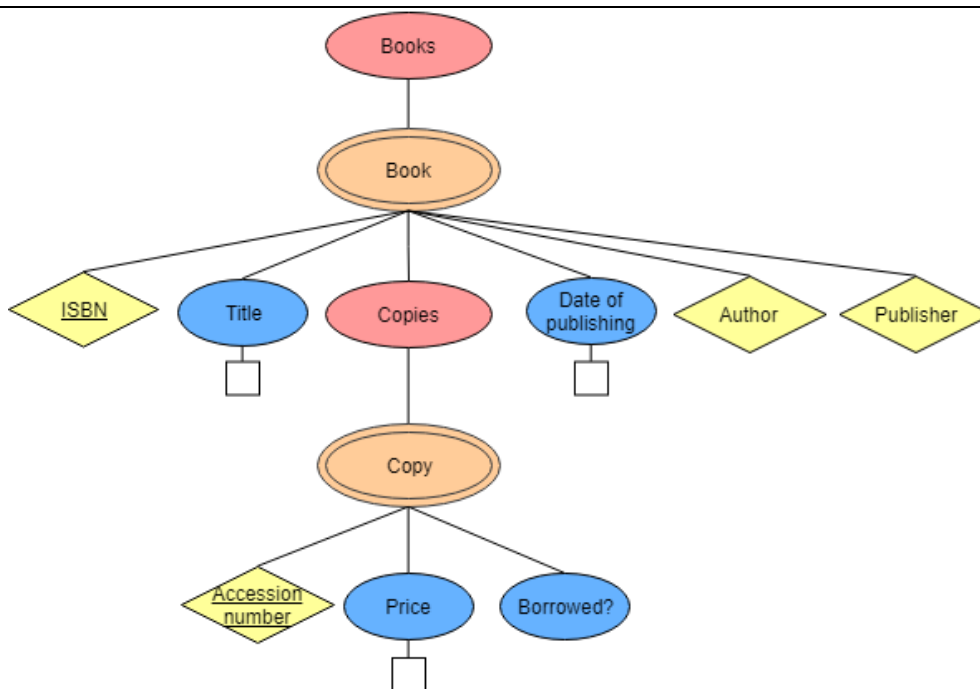
2. ábra XDM-modell



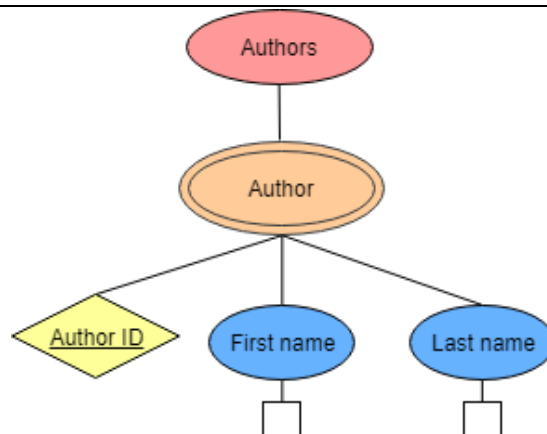
3. ábra Virtuális gyökér és gyökérelemek



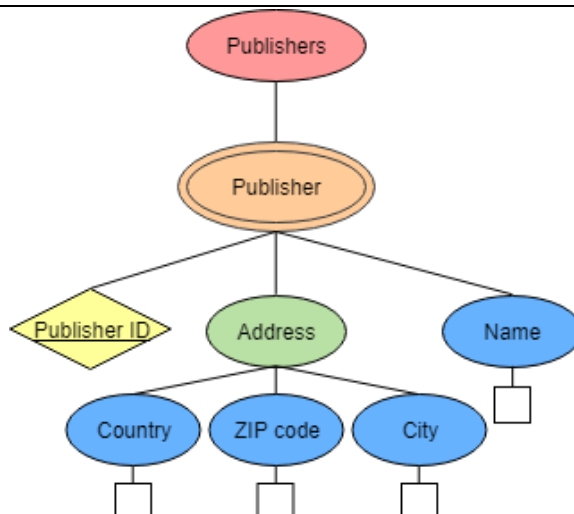
4. ábra Az olvasó elem és alelemei



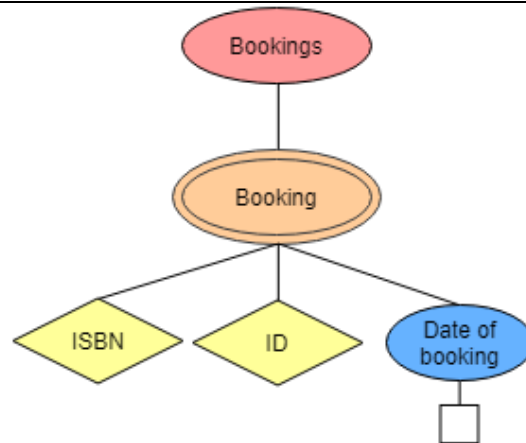
5. ábra A könyv elem és alelemei



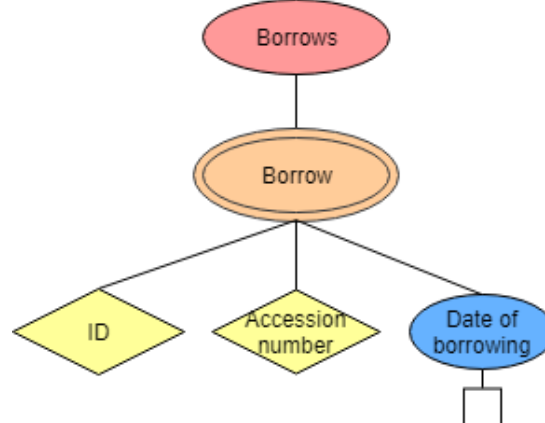
6. ábra A szerző elem és alelemei



7. ábra A kiadó elem és alelemei



8. ábra Az előjegyzés reláció és alelemei



9. ábra A kölcsönzés reláció és alelemei

3 Az XML séma létrehozása

3.1 A létrehozás szempontjai

A létrehozásnál fontos, hogy a séma visszaadja az XDM-modell szerkezete által definiált struktúrát. Ennek megkönnyítése érdekében egy globális típust hozunk létre, a lakcím tárolására. Ezt „Type_Address”-nek nevezzük és a következőket tartalmazza: irányítószám, város, utca, házszám. A későbbiekben bővítéssel az ország mezőt is tartalmazni fogja. A típus létrehozása a következőképpen valósult meg:

```
<xs:complexType name="Type_Address">
  <xs:sequence>
    <xs:element name="ZIP_Code" minOccurs="1" maxOccurs="1">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="[0-9]{4}"></xs:pattern>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Country" type="xs:string" minOccurs="1"
maxOccurs="1"></xs:element>
```



```
<xs:element name="City" type="xs:string" minOccurs="1"
maxOccurs="1"></xs:element>
</xs:sequence>
</xs:complexType>
```

A számosságot tekintve azon elemeket, melyeknek csak egyszer kell előfordulniuk, egyre korlátozzuk a számukat (*maxOccurs="1"*), illetve azoknak, melyeknek egyszer sem, ott alsó korlátnak 0-t, felső korlátnak 1-et (*minOccurs="0" maxOccurs="1"*), ha pedig többször, akkor felső korlátnak határozatlan (*maxOccurs="unbounded"*) számossággal definiáljuk.

Fontos továbbá, hogy az elsődleges kulcsokat a szülőkön belül kell megadnunk, hiszen így biztosítjuk, hogy a kulcsmező egyedisége csak azon elemek körében érvényesül. Az idegen kulcsokat, vagy hivatkozásokat már a gyökérelembe kell definiálni, mert máskülönben láthatósági problémákba ütközhetünk.

Üres elem azokból a mezőkből lesz, melyeknél elég feltüntetni magát, az üres elemet, nem szükséges bármilyen adatot is tartalmaznia. Ilyenek lesznek a Beiratkozott? [*Registered?*] illetve Kölcsönzött-e? [*Borrowed?*] elemek.

Megszorításokat számértékekre adunk meg, illetve az e-mail cím és telefonszám mintáját definiáljuk.

3.2 Az XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
xmlns:vc="http://www.w3.org/2007/XMLSchema-versioning" vc:minVersion="1.0"
vc:maxVersion="1.1">

  <xs:complexType name="Type_Address">
    <xs:sequence>
      <xs:element name="ZIP_Code" minOccurs="1" maxOccurs="1">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{4}"></xs:pattern>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="City" type="xs:string" minOccurs="1"
maxOccurs="1"></xs:element>
      <xs:element name="Street" type="xs:string" minOccurs="1"
maxOccurs="1"></xs:element>
      <xs:element name="Number" minOccurs="1" maxOccurs="1">
        <xs:simpleType>
```

```
<xs:restriction base="xs:integer">
  <xs:minInclusive value="1"></xs:minInclusive>
</xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:element name="ROOT">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Readers" minOccurs="1" maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Reader" minOccurs="0"
maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="E-mail" minOccurs="1"
maxOccurs="1">
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:pattern value="[_\\-a-zA-Z0-9\\.\\+]+@[a-zA-Z0-
9](\\.|?[-a-zA-Z0-9]*[a-zA-Z0-9])*"></xs:pattern>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element name="Telephone" minOccurs="0"
maxOccurs="1">
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:pattern value="[0-9]{2}-[0-9]{3}-[0-
9]{4}"></xs:pattern>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element name="Occupation" type="xs:string"
minOccurs="1" maxOccurs="1"></xs:element>
                  <xs:element name="Date_of_enrollment" type="xs:date"
minOccurs="1" maxOccurs="1"></xs:element>
                  <xs:element name="Registered" minOccurs="0"
maxOccurs="1">
                    <xs:complexType/>
                  </xs:element>
                  <xs:element name="Name" minOccurs="1"
maxOccurs="1">
                    <xs:complexType>
                      <xs:sequence>
```

```
<xs:element name="First_name" type="xs:string"
minOccurs="1" maxOccurs="1"></xs:element>
<xs:element name="Last_name" type="xs:string"
minOccurs="1" maxOccurs="1"></xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Address">
<xs:complexType>
<xs:complexContent>
<xs:extension base="Type_Address">
<xs:sequence>
<xs:element name="Street" minOccurs="0"
maxOccurs="1"></xs:element>
<xs:element name="Number" minOccurs="1"
maxOccurs="1">
<xs:simpleType>
<xs:restriction base="xs:integer">
<xs:minInclusive
value="1"></xs:minInclusive>
</xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="ReaderID" type="xs:string"
use="required"></xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:key name="ReaderPK">
<xs:selector xpath="Reader"></xs:selector>
<xs:field xpath="@ReaderID"></xs:field>
</xs:key>
</xs:element>
<xs:element name="Borrows" minOccurs="1" maxOccurs="1">
<xs:complexType>
<xs:sequence>
<xs:element name="Borrow" minOccurs="0"
maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="Date_of_borrowing" type="xs:date"
minOccurs="1" maxOccurs="1"></xs:element>
</xs:sequence>
```

```

        <xs:attribute name="ReaderID" type="xs:string"
use="required"></xs:attribute>
        <xs:attribute name="AccessionNumber" type="xs:string"
use="required"></xs:attribute>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Books" minOccurs="1" maxOccurs="1">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Book" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Title" type="xs:string" minOccurs="1"
maxOccurs="1"></xs:element>
                        <xs:element name="Date_of_publishing" minOccurs="1"
maxOccurs="1">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:pattern value="[0-9]{4}"></xs:pattern>
                                    </xs:restriction>
                                </xs:simpleType>
                                </xs:element>
                                <xs:element name="Copies" minOccurs="1"
maxOccurs="1">
                                        <xs:complexType>
                                            <xs:sequence>
                                                <xs:element name="Copy" minOccurs="0"
maxOccurs="unbounded">
                                                        <xs:complexType>
                                                            <xs:sequence>
                                                                <xs:element name="Price" minOccurs="1"
maxOccurs="1">
                                                                        <xs:simpleType>
                                                                            <xs:restriction base="xs:integer">
                                                                                <xs:minInclusive
value="0"></xs:minInclusive>
                                                                                        </xs:restriction>
                                                                                        </xs:simpleType>
                                                                                        </xs:element>
                                                                                        <xs:element name="Borrowed"
minOccurs="0" maxOccurs="1">
                                                                            <xs:complexType/>
                                                                            </xs:element>
                                                                        </xs:sequence>
                                                                        <xs:attribute name="AccessionNumber"
type="xs:string" use="required"></xs:attribute>

```

```
        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
<xs:key name="CopyPK">
    <xs:selector xpath="Copy"></xs:selector>
    <xs:field xpath="@AccessionNumber"></xs:field>
</xs:key>
</xs:element>
</xs:sequence>
<xs:attribute name="ISBN" type="xs:string"
use="required"></xs:attribute>
<xs:attribute name="Author" type="xs:string"
use="required"></xs:attribute>
<xs:attribute name="Publisher" type="xs:string"
use="required"></xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:key name="BookPK">
    <xs:selector xpath="Book"></xs:selector>
    <xs:field xpath="@ISBN"></xs:field>
</xs:key>
</xs:element>
<xs:element name="Bookings" minOccurs="1" maxOccurs="1">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Booking" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Date_of_booking" type="xs:date"
minOccurs="1" maxOccurs="1"></xs:element>
                    </xs:sequence>
                    <xs:attribute name="ISBN" type="xs:string"
use="required"></xs:attribute>
                    <xs:attribute name="ReaderID" type="xs:string"
use="required"></xs:attribute>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Publishers" minOccurs="1" maxOccurs="1">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Publisher" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
```

```
<xs:sequence>
  <xs:element name="City" type="xs:string" minOccurs="1"
maxOccurs="1"></xs:element>
  <xs:element name="Name" type="xs:string"
minOccurs="1" maxOccurs="1"></xs:element>
</xs:sequence>
<xs:attribute name="PublisherID" type="xs:string"
use="required"></xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:key name="PublisherPK">
  <xs:selector xpath="Publisher"></xs:selector>
  <xs:field xpath="@PublisherID"></xs:field>
</xs:key>
</xs:element>
<xs:element name="Authors" minOccurs="1" maxOccurs="1">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Author" minOccurs="0"
maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="First_name" type="xs:string"
minOccurs="1" maxOccurs="1"></xs:element>
            <xs:element name="Last_name" type="xs:string"
minOccurs="1" maxOccurs="1"></xs:element>
          </xs:sequence>
          <xs:attribute name="AuthorID" type="xs:string"
use="required"></xs:attribute>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:key name="AuthorPK">
    <xs:selector xpath="Author"></xs:selector>
    <xs:field xpath="@AuthorID"></xs:field>
  </xs:key>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:keyref name="Booking_ReaderFK" refer="ReaderPK">
  <xs:selector xpath="Bookings/Booking"></xs:selector>
  <xs:field xpath="@ReaderID"></xs:field>
</xs:keyref>
<xs:keyref name="Booking_BookFK" refer="BookPK">
  <xs:selector xpath="Bookings/Booking"></xs:selector>
  <xs:field xpath="@ISBN"></xs:field>
</xs:keyref>
```

```
</xs:keyref>
<xs:keyref name="Borrow_ReaderFK" refer="ReaderPK">
  <xs:selector xpath="Borrows/Borrow"></xs:selector>
  <xs:field xpath="@ReaderID"></xs:field>
</xs:keyref>
<xs:keyref name="Borrow_CopyFK" refer="CopyPK">
  <xs:selector xpath="Borrows/Borrow"></xs:selector>
  <xs:field xpath="@AccessionNumber"></xs:field>
</xs:keyref>
<xs:keyref name="Book_AuthorFK" refer="AuthorPK">
  <xs:selector xpath="Books/Book"></xs:selector>
  <xs:field xpath="@AuthorID"></xs:field>
</xs:keyref>
<xs:keyref name="Book_PublisherFK" refer="PublisherPK">
  <xs:selector xpath="Books/Book"></xs:selector>
  <xs:field xpath="@PublisherID"></xs:field>
</xs:keyref>
</xs:element>

</xs:schema>
```