

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat: Állatkerthálózat

Készítette: **Martinák Mátyás**
Neptunkód: **KLNSPG**
Dátum: **2023. 10. 25.**

Miskolc, 2023

Tartalomjegyzék

| | |
|--|-----------|
| 1. A feladat leírása | 2 |
| 2. I. feladat - XML/XSD létrehozás | 4 |
| 2.1. ER modell | 4 |
| 2.2. XDM modell | 4 |
| 2.3. Az XML dokumentum | 5 |
| 2.4. Az XML dokumentum alapján XMLSchema készítése | 8 |
| 3. II. feladat - DOM | 14 |
| 3.1. Adatolvasás | 14 |
| 3.2. Adatmódosítás | 14 |
| 3.3. Adatlekérdezés | 14 |
| 3.4. Adatírás | 14 |

1. fejezet

A feladat leírása

A feladat egy vagy több állatkert hálózatát mutatja be, amiben helyet kapnak az egyes állatkeretekben dolgozók, azok feladatai, az állatok és élőhelyeik, eledelük, az eledelt gyártó cégek, illetve az állatok örökbefogadói, ha vannak. Mind az ER modell tervezésben és mind az XML megvalósításban angol nyelvet használtam, ugyanis ez a legelterjedtebb nyelv a programozásban. Összesen 6 egyedet hoztam létre, melyek a következők:

- Employee,
- Site,
- Habitat,
- Animal,
- Food,
- User

Legelőször is érdemes pár szót szólni a **Site** egyedről. Innen indul ki minden. Ez az egyed tárolja el az egyes állatkertek legfőbb tulajdonságait, mint pl. név, terület vagy éppen nyitva tartás. Elsődleges kulcsa a `site_id`, ami az állatpark azonosítója.

A Site és az **Employee** egyed között egy 1:N kapcsolat van, mivel egy állatkerthez több dolgozó is tartozhat, de egy dolgozó, csak egy állatkerthez tartozhat. Az 1:N kapcsolat neve: **Works**. Egy dolgozónak van azonosítója, vezeték és keresztnéve (ami ER modellben egy többágú tulajdonság), neme, születési dátuma és ami a legfontosabb, a dolgozó feladatai, posztjai, amiből lehet egy vagy több, így ez egy többértékű tulajdonság lesz. Ez azért fontos, mivel a relációs modellnél ez a tulajdonság egy külön táblát kap majd, amiben lesz a posztnak egy id-ja, a poszt neve, illetve, hogy kihez tartozik.

Egy állatkerthez több élőhely is tartoztat, de egy élőhely csak egy állatkerthez tartozik. Ezt ábrázolja a **Manage** kapcsolat, ami 1:N kapcsolattal köti össze a Site és a **Habitat** egyedeket. Az

élőhelynek nincsenek „extra” tulajdonságai, van egy azonosítója, neve, térképen való elhelyezkedése, leírása és kapacitása, hogy mennyi állatot képes egyszerre befogadni.

Az **Occupy** kapcsolat szintén 1:N kapcsolattal köti össze a Habitat-ot az **Animal**-lél. Az állatnak van azonosítója, neve, faja és leírása.

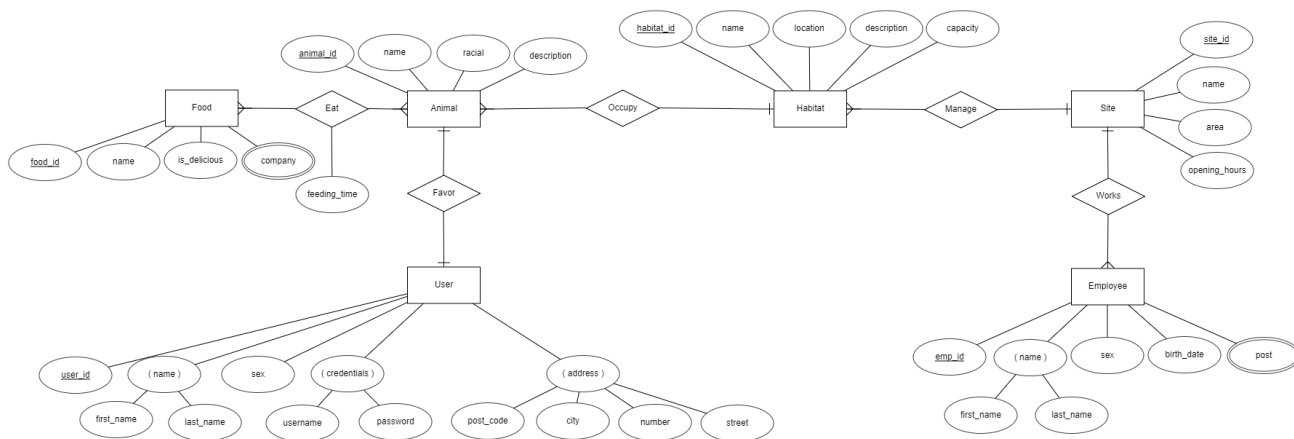
Itt jön a legelső N:M kapcsolat, az **Eat**, aminek lesz tulajdonsága, a **feeding_time**, az etetési idő. Fontos, hogy megjegyezzük, az N:M kapcsolat külön kapcsolótáblát fog kapni a relációs modellben. Az Eat köti össze az Animalt a **Food**-dal, ami az állat eledelét modellező egyed. Ennek van azonosítója, neve, egy **boolean** (logikai) értéke, ami azt dönti el, hogy finom-e az adott eledel, vagy sem. Ezen kívül van egy többértékű tulajdonsága is, az eledeleket gyártó cégek, amik szintén külön táblát fognak majd kapni a relációs modellben.

Az állatokat örökbe is lehet fogani bizonyos **User**-eknek, ezt a **Favor 1:1** kapcsolat modellezi. Talán a Usernek van a legtöbb tulajdonsága ebben az adatbázisban. Van természetesen azonosítója, két neve (vezeték és keresztnév), neme, bejelentkezési adatai (felhasználónév, jelszó), mivel online szeretnénk lebonyolítani az állatok örökbefogadását. Ezen kívül címe is van a felhasználónak, ami az irányítószám, város, utca, házszám tulajdonságokból tevődik össze.

2. fejezet

I. feladat - XML/XSD létrehozás

2.1. A feladat ER modellje

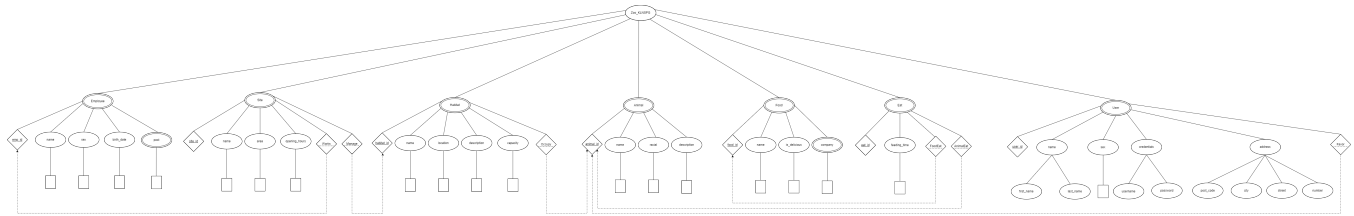


2.1. ábra. A feladat ER modellje

2.2. A feladat XDM modellje

A konvertáláskor figyelembe kell venni az ER modell során definiált kapcsolatokat, azok típusait (1:1, 1:N, N:M), illetve az entitások elsődleges kulcsait is. Minden *egy-több* kapcsolat esetében ahhoz az elsődleges kulcshoz kerül a szaggatott nyíl, ahol az ER modellben a „több” szerepel. Az „Eat” és a „Favor” kapcsolatokat kivéve, mindenhol 1:N kapcsolat szerepel az ER modellben, így az XDM mindenhol majdnem hasonlóan fog kinézni. Az N:M kapcsolat esetében egy új modellt

veszünk fel, tulajdonsággal és *primary key*-el együtt természetesen, ahonnan a nyilakat a fő entitásokhoz húzzuk. A többágú tulajdonságok itt is több tulajdonsággal rendelkeznek, a többértékű tulajdonságok itt nem kapnak külön modellt. Az XDM modell gyökéréleme: **Zoo_KLNSPG**



2.2. ábra. A feladat XDM modellje

A „Works” kapcsolat 1:N, ahol a több érték az *Employee*-hoz kerül, így a kapcsolatot is az *emp_id*-hez húzzuk. Szintén ugyan ez a helyzet a „Manage” kapcsolatnál is, ahol a rombuszt a *habitat_id*-hoz húzzuk. Az *Animal* modell egy különleges, ide 3 kapcsolatot is húzunk, melyek:

- „Manage” a *Habitat*-ból
- „AnimalEat” az *Eat*-ből és végül
- „Favor” a *User*-ből

Kettő 1:N kapcsolat és egy 1:1 kapcsolat húz ide.

2.3. Az XDM modell alapján XML dokumentum készítése

Az XMLKLNSPG.xml dokumentumot *Visual Studio Code*-ban hoztam létre, és XML 1.0 szabvány szerint készült el. A dokumentumhoz hozzá kötöttem az XMLSchemaKLNSPG.xsd XSD file-t, és definiáltam az egyedeket az XML szabályainak megfelelően. Ahol szükséges volt, gyermek elemeket, valamint attribútumokat használtam a tagok azonosításához.

```
<?xml version="1.0" encoding="UTF-8"?>

<Zoo_KLNSPG xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="XMLSchemaKLNSPG.xsd">

  <!-- Employee példányok-->
  <Employee emp_id="1">
    <first_name>Kovacs</first_name>
    <last_name>Janos</last_name>
    <birth_date>1979-11-02</birth_date>
    <sex>M</sex>
  </Employee>
```

```
<Employee emp_id="2">
<first_name>Jakab</first_name>
<last_name>Jozsef</last_name>
<birth_date>1954-12-06</birth_date>
<sex>M</sex>
</Employee>
<Employee emp_id="3">
<first_name>Balogh</first_name>
<last_name>Boglarka</last_name>
<birth_date>2000-11-04</birth_date>
<sex>F</sex>
</Employee>

<!-- Site peldanyok-->
<Site site_id="1" Works="3" Manage="1">
<name>Miskolc Allatkert</name>
<area>212000</area>
<opening_hours>09:00 - 17:00</opening_hours>
</Site>
<Site site_id="2" Works="1" Manage="2">
<name>Fovaros Allat-es Novenykert</name>
<area>184001</area>
<opening_hours>09:00 - 17:30</opening_hours>
</Site>
<Site site_id="3" Works="2" Manage="3">
<name>Debreceni Allatkert es Vidampark</name>
<area>170000</area>
<opening_hours>09:00 - 15:30</opening_hours>
</Site>

<!-- Habitat peldanyok-->
<Habitat habitat_id="1" Occupy="3">
<name>Medve park</name>
<location>#3</location>
<description>Az allatkerti medvek elohelye. Jelenleg harom medve talalhato
itt, Jazmin, Andor es Matyko. Szeretik a latogatokat, mindig erdeklodve
nezelodnek.</description>
</Habitat>
<Habitat habitat_id="2" Occupy="1">
<name>Muflonok dombja</name>
<location>#22</location>
<description>Vadasparkunk muflonjai itt talalhatoak. Baratsagosak, turista
kedvelok, szeretik a finom falatokat.</description>
</Habitat>
<Habitat habitat_id="3" Occupy="2">
<name>Szurikatak szigete</name>
<location>#18</location>
```

```
<description>Ki ne imadna a kis erdeklo do szurikatak at. Nalunk ro gton 4-et  
is orok be fogad hat, vagy csak latogathat is.</description>  
</Habitat>
```

```
<!-- Animal peldanyok-->  
<Animal animal_id="1">  
<name>Matyko</name>  
<racial>Medve</racial>  
<description>Az allatkert egyik kan medve je</description>  
</Animal>  
<Animal animal_id="2">  
<name>Kis Hegyes</name>  
<racial>Muflon</racial>  
<description>Az allatkert nosten y muflona</description>  
</Animal>  
<Animal animal_id="3">  
<name>Mokas</name>  
<racial>Szurikata</racial>  
<description>Az allatkert legfiatalabb szurikata ja</description>  
</Animal>
```

```
<!-- Food peldanyok-->  
<Food food_id="1">  
<name>Fagyasztott nyers hus</name>  
<is_delicious>>false</is_delicious>  
<company>Family Frost</company>  
</Food>  
<Food food_id="2">  
<name>Sargarepa</name>  
<is_delicious>>true</is_delicious>  
<company>Magyar Zoldseg</company>  
</Food>  
<Food food_id="3">  
<name>Sult husi</name>  
<is_delicious>>true</is_delicious>  
<company>Family Frost</company>  
</Food>
```

```
<!-- Eat peldanyok-->  
<Eat eat_id="1" FoodEat="1" AnimalEat="3">  
<feeding_time>09:00:00 18:00:00</feeding_time>  
</Eat>  
<Eat eat_id="2" FoodEat="3" AnimalEat="1">  
<feeding_time>07:00:00 20:00:00</feeding_time>  
</Eat>  
<Eat eat_id="3" FoodEat="2" AnimalEat="2">  
<feeding_time>06:00:00 14:00:00 20:00:00</feeding_time>  
</Eat>
```



```
<!-- User peldanyok-->
<User user_id="1" Favor="3">
  <username>Allatbarat</username>
  <password>allat123</password>
  <sex>M</sex>
  <first_name>Kiss</first_name>
  <last_name>Sandor</last_name>
  <post_code>8200</post_code>
  <city>Veszprem</city>
  <street>Petofi Sandor utca</street>
  <number>3</number>
</User>
<User user_id="2" Favor="1">
  <username>Vadoc</username>
  <password>fegyo02</password>
  <sex>M</sex>
  <first_name>Fegyver</first_name>
  <last_name>Sandor</last_name>
  <post_code>4024</post_code>
  <city>Debrecen</city>
  <street>Kossuth utca</street>
  <number>26</number>
</User>
<User user_id="3" Favor="2">
  <username>Possumluvr</username>
  <password>possumlover</password>
  <sex>F</sex>
  <first_name>Kazai</first_name>
  <last_name>Eszter</last_name>
  <post_code>3521</post_code>
  <city>Miskolc</city>
  <street>Uj elet utca</street>
  <number>24</number>
</User>

</Zoo_KLNSPG>
```

Listing 2.1. Az XML dokumentum

2.4. Az XML dokumentum alapján XMLSchema készítése

Az XMLSchemaKLNPG.xsd séma file leírja mindazon megkötéseket, amelyeknek az XML dokumentumnak meg kell felelnie. Itt definiálunk minden típust, amit az XML file-ban használni szeretnénk, valamint az adatbázis kapcsolatait `xs:unique` és `xs:keyref` bejegyzésekkel hozom létre.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- Saját egyszeru tipusok definialasa -->

  <!-- Altalanos saját tipusok-->
  <xs:element name="name" type="xs:string"/>
  <xs:element name="first_name" type="xs:string"/>
  <xs:element name="last_name" type="xs:string"/>
  <xs:element name="sex" type="sexType"/>
  <xs:element name="description" type="xs:string"/>

  <!-- Employee saját tipus-->
  <xs:element name="birth_date" type="dateType"/>

  <!-- Site saját tipusok-->
  <xs:element name="area" type="xs:integer"/>
  <xs:element name="opening_hours" type="xs:string"/>

  <!-- Habitat saját tipus-->
  <xs:element name="location" type="xs:string"/>

  <!-- Animal saját tipus-->
  <xs:element name="racial" type="xs:string"/>

  <!-- Food saját tipus-->
  <xs:element name="is_delicious" type="xs:boolean"/>
  <xs:element name="company" type="xs:string"/>

  <!-- Eat saját tipus-->
  <xs:element name="feeding_time" type="timeListType"/>

  <!-- User saját tipusok-->
  <xs:element name="username" type="xs:string"/>
  <xs:element name="password" type="xs:string"/>
  <xs:element name="post_code" type="xs:string"/>
  <xs:element name="city" type="xs:string"/>
  <xs:element name="street" type="xs:string"/>
  <xs:element name="number" type="xs:string"/>

  <!-- Simple types-->
```

```
<xs:simpleType name="dateType">
  <xs:restriction base="xs:date">
    <xs:minInclusive value="1940-01-01"/>
    <xs:maxInclusive value="2000-12-31"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="sexType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="M"/>
    <xs:enumeration value="F"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="timeListType">
  <xs:list itemType="xs:time"/>
</xs:simpleType>

<!-- Complex types-->
<xs:complexType name="employeeType">
  <xs:sequence>
    <xs:element ref="first_name"/>
    <xs:element ref="last_name"/>
    <xs:element ref="birth_date"/>
    <xs:element ref="sex"/>
  </xs:sequence>
  <xs:attribute name="emp_id" type="xs:integer" use="required"/>
</xs:complexType>

<xs:complexType name="siteType">
  <xs:sequence>
    <xs:element ref="name"/>
    <xs:element ref="area"/>
    <xs:element ref="opening_hours"/>
  </xs:sequence>
  <xs:attribute name="site_id" type="xs:integer" use="required"/>
  <xs:attribute name="Works" type="xs:integer" use="required"/>
  <xs:attribute name="Manage" type="xs:integer" use="required"/>
</xs:complexType>

<xs:complexType name="habitatType">
  <xs:sequence>
    <xs:element ref="name"/>
    <xs:element ref="location"/>
    <xs:element ref="description"/>
  </xs:sequence>
  <xs:attribute name="habitat_id" type="xs:integer" use="required"/>
  <xs:attribute name="Occupy" type="xs:integer" use="required"/>
```

```
</xs:complexType>

<xs:complexType name="animalType">
  <xs:sequence>
    <xs:element ref="name"/>
    <xs:element ref="racial"/>
    <xs:element ref="description"/>
  </xs:sequence>
  <xs:attribute name="animal_id" type="xs:integer" use="required"/>
</xs:complexType>

<xs:complexType name="foodType">
  <xs:sequence>
    <xs:element ref="name"/>
    <xs:element ref="is_delicious"/>
    <xs:element ref="company"/>
  </xs:sequence>
  <xs:attribute name="food_id" type="xs:integer" use="required"/>
</xs:complexType>

<xs:complexType name="eatType">
  <xs:sequence>
    <xs:element ref="feeding_time"/>
  </xs:sequence>
  <xs:attribute name="eat_id" type="xs:integer" use="required"/>
  <xs:attribute name="FoodEat" type="xs:integer" use="required"/>
  <xs:attribute name="AnimalEat" type="xs:integer" use="required"/>
</xs:complexType>

<xs:complexType name="userType">
  <xs:sequence>
    <xs:element ref="username"/>
    <xs:element ref="password"/>
    <xs:element ref="sex"/>
    <xs:element ref="first_name"/>
    <xs:element ref="last_name"/>
    <xs:element ref="post_code"/>
    <xs:element ref="city"/>
    <xs:element ref="street"/>
    <xs:element ref="number"/>
  </xs:sequence>
  <xs:attribute name="user_id" type="xs:integer" use="required"/>
  <xs:attribute name="Favor" type="xs:integer" use="required"/>
</xs:complexType>

<!-- A gyokerelem összetett tipusa -->
<xs:complexType name="zooType">
  <xs:sequence>
```

```
<xs:element name="Employee" type="employeeType" minOccurs="3"
  maxOccurs="unbounded"/>
<xs:element name="Site" type="siteType" minOccurs="3" maxOccurs="unbounded"/>
<xs:element name="Habitat" type="habitatType" minOccurs="3"
  maxOccurs="unbounded"/>
<xs:element name="Animal" type="animalType" minOccurs="3"
  maxOccurs="unbounded"/>
<xs:element name="Food" type="foodType" minOccurs="3" maxOccurs="unbounded"/>
<xs:element name="Eat" type="eatType" minOccurs="3" maxOccurs="unbounded"/>
<xs:element name="User" type="userType" minOccurs="3" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>

<!-- A gyokerelem definicioja -->
<xs:element name="Zoo_KLNSPG" type="zooType">

  <!-- Elsodleges kulcsok -->
  <xs:key name="EmployeeKey">
    <xs:selector xpath="Employee"/>
    <xs:field xpath="@emp_id"/>
  </xs:key>

  <xs:key name="SiteKey">
    <xs:selector xpath="Site"/>
    <xs:field xpath="@site_id"/>
  </xs:key>

  <xs:key name="HabitatKey">
    <xs:selector xpath="Habitat"/>
    <xs:field xpath="@habitat_id"/>
  </xs:key>

  <xs:key name="AnimalKey">
    <xs:selector xpath="Animal"/>
    <xs:field xpath="@animal_id"/>
  </xs:key>

  <xs:key name="FoodKey">
    <xs:selector xpath="Food"/>
    <xs:field xpath="@food_id"/>
  </xs:key>

  <xs:key name="EatKey">
    <xs:selector xpath="Eat"/>
    <xs:field xpath="@eat_id"/>
  </xs:key>

  <xs:key name="UserKey">
```

```
<xs:selector xpath="User"/>
<xs:field xpath="@user_id"/>
</xs:key>

<!-- Idegen kulcsok -->
<xs:keyref name="SiteWork" refer="EmployeeKey">
<xs:selector xpath="Site"/>
<xs:field xpath="@Works"/>
</xs:keyref>

<xs:keyref name="SiteManage" refer="HabitatKey">
<xs:selector xpath="Site"/>
<xs:field xpath="@Manage"/>
</xs:keyref>

<xs:keyref name="HabitatOccupy" refer="AnimalKey">
<xs:selector xpath="Habitat"/>
<xs:field xpath="@Occupy"/>
</xs:keyref>

<xs:keyref name="EatFood" refer="FoodKey">
<xs:selector xpath="Eat"/>
<xs:field xpath="@FoodEat"/>
</xs:keyref>

<xs:keyref name="EatAnimal" refer="AnimalKey">
<xs:selector xpath="Eat"/>
<xs:field xpath="@AnimalEat"/>
</xs:keyref>

<xs:keyref name="UserAnimal" refer="AnimalKey">
<xs:selector xpath="User"/>
<xs:field xpath="@Favor"/>
</xs:keyref>

<!-- Az 1:1 kapcsolat -->
<xs:unique name="UserAnimalConnect">
<xs:selector xpath="UserKey"/>
<xs:field xpath="@Favor"/>
</xs:unique>

</xs:element>

</xs:schema>
```

Listing 2.2. Az XSD dokumentum

3. fejezet

II. feladat - DOM

3.1. Adatolvasás

3.2. Adatmódosítás

3.3. Adatlekérdezés

3.4. Adatírás