

# JEGYZŐKÖNYV

## Adatkezelés XML környezetben

Féléves feladat: Állatkerthálózat

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

**Miskolc, 2023**

# Tartalomjegyzék

<b>1. A feladat leírása</b>	<b>2</b>
<b>2. I. feladat - XML/XSD létrehozás</b>	<b>4</b>
2.1. ER modell . . . . .	4
2.2. XDM modell . . . . .	4
2.3. Az XDM modell alapján XML dokumentum készítése . . . . .	5
2.4. Az XML dokumentum alapján XMLSchema készítése - saját típusok, ref, key, keyref, speciális elemek . . . . .	5
<b>3. II. feladat - DOM</b>	<b>6</b>
3.1. Adatolvasás . . . . .	6
3.2. Adatmódosítás . . . . .	6
3.3. Adatlekérdezés . . . . .	6
3.4. Adatírás . . . . .	6

# 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**-l. Az állatnak van azonosítója, neve, faja és leírása.

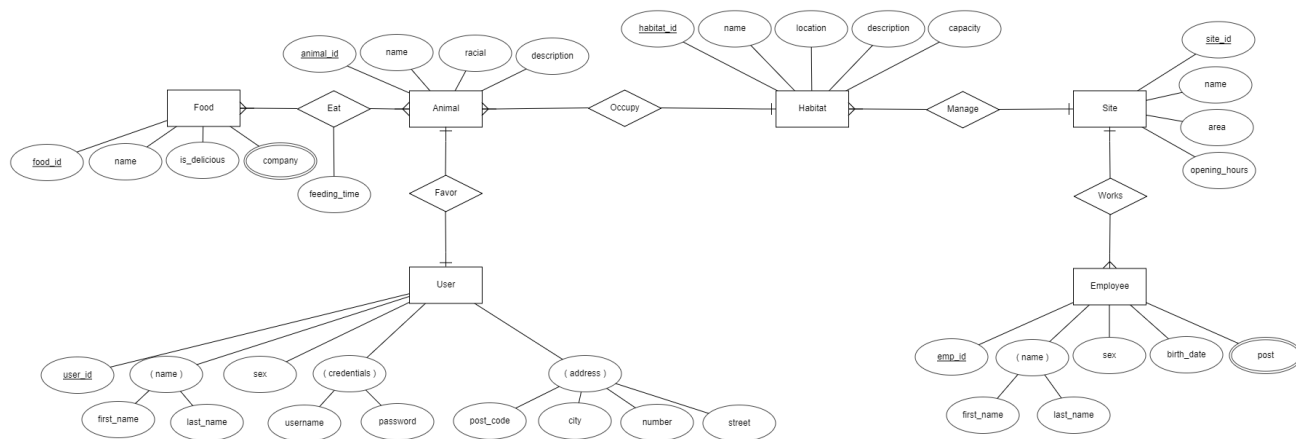
Itt jön a legelső N:M kapcsolat, az **Eat**, aminek lesz tulajdonsága, a **feeding\_time**, az etetési idő. Fontos, hogy megjegyezzük, az N:M kapcsolat külön kapcsolótáblát fog kapni a relációs modellben. Az Eat köti össze az Animalt a **Food**-dal, ami az állat eledelét modellező egyed. Ennek van azonosítója, 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

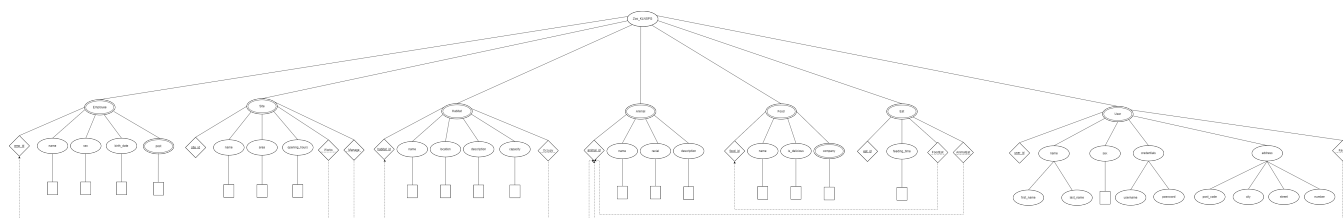


2.1. ábra. A feladat ER modellje

#### 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 kulchoz 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

## 2.4. Az XML dokumentum alapján XMLSchema készítése - saját típusok, ref, key, keyref, speciális elemek

## 3. fejezet

### II. feladat - DOM

#### 3.1. Adatolvasás

#### 3.2. Adatmódosítás

#### 3.3. Adatlekérdezés

#### 3.4. Adatírás