

# WEBPROGRAMOZÁS III. GY.

(LBT\_IM744G2)

Beadandó feladat dokumentáció

Kovács Norbert  
ANOXWJ

2021. június 8.

# Tartalomjegyzék

<b>1. Weboldal koncepciója</b>	<b>3</b>
1.1. Technikai részletek . . . . .	3
1.2. Elképzelés . . . . .	3
<b>2. Adatbázis tervezése</b>	<b>4</b>
2.1. ER modell . . . . .	4
2.2. Relációs adatmodell . . . . .	6
2.3. SQL parancsok . . . . .	6
<b>3. Összefoglaló</b>	<b>6</b>

# 1. Weboldal koncepciója

## 1.1. Technikai részletek

A beadandó feladatot, az órán is alkalmazott *xampp* szerverrel kívánom megvalósítani, *apache netbeans* fejlesztőkörnyezettel. A Mysql kezelését a videóban is használt, *phpmyadmin* oldalon intézem. Nem fontos részlet, a megvalósítás szempontjából, de a Windows operációs rendszer, virtuális gépként fut.

(Követelmény: A beadandó feladat PHP MVC kereterendszerben készül!)

- OS: Windows 10 Pro (21H1)
- Kernel: 10.0.19041.1023
- Mysql: 10.4.19-MariaDB
- PHP: 8.0.6
- Apache Netbeans IDE: 12.3
- CodeIgniter: 3.1.11

## 1.2. Elképzelés

A weboldal egy *fiktív* szervezet köré épül, aminek a feladata az *állategészségügy*, és *állatvédelem*. A fiktív szervezetnek több városban lesznek épületei országszerte. Egy városban több különféle épület is tartozhat a szervezethez. Például tartozhat állatmenhely, állatorvosi rendelő, raktárépület stb. Ezeknek az épületeknek vannak hozzájuk tartozó egységeik. Egy menhely állatok tárolására alkalmas kennelekkel, egy rendelő vizsgálóval, váróval, egy raktár szobákkal, szektorokkal rendelkezik stb.

Hasonlóképpen az egyetemi példafeladathoz, ahol

Kampusz -> Épület -> Terem

Pl.: Eger -> C épület -> 124-es terem,

felosztás található, itt is három rétegre kívánom felosztani a problémát, mégpedig:

Telep -> Épület -> 'Szoba'

Pl.: Miskolc -> Iroda -> 1-es iroda

vagy

Eger -> Egri állatmenhely -> 3-as Kennel.

A weboldal célja, tehát hogy a szervezet irányítani tudja a különböző városaiban található épületcsoportokat, pénzmozgásokat, alkalmazottakat stb. A fiktív szervezetet, az előző beadandóm állatotthonából fejlesztem tovább, részben idő spórolás céljából, részben a hasonló koncepcióból adódóan. Szervezet neve: **Kék mancs**.

Az *elképzelés* szerint, a felhasználókat több rétegben kezeljük majd. A weboldal nem a nyilvánosság felé közvetített oldal, hanem belső rendszerért felelős. Ettől függetlenül szükség lesz, telepenként admin, és felhasználói felületre.

## 2. Adatbázis tervezése

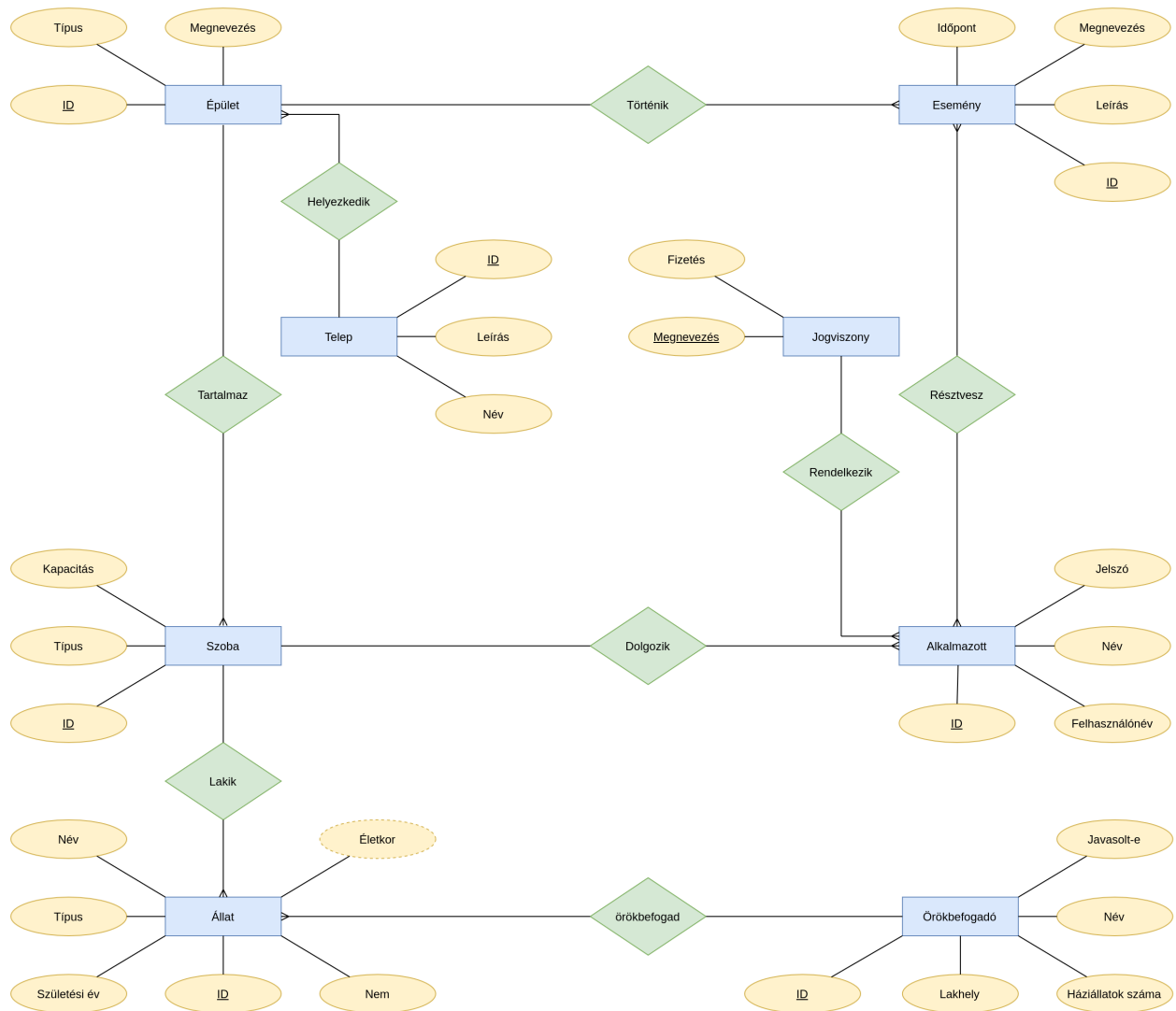
Az első entitásunk, egy **Telep** entitás lesz. Itt nem egy konkrét Telephelyről van szó, hanem egy olyan összefoglaló jelzőről, mint az egyetem kapcsán a Campus-ról. Ennél találóbb, vagy pontosabb megnevezést nem tudtam kitalálni, ami összefoglalja az egy városban található épületszolgáltatásokat. Szükségünk lesz, egy **épület** entitásra és egy **'szoba'** entitásra. Ennek a megnevezése, nem teljesen fedi le a pontos szerepet, hiszen minden épületben ez mást reprezentál, azonban szükséges eltárolnunk, hogy melyik állat melyik kennelhez van rendelve, vagy hogy melyik alkalmazott melyik irodában dolgozik stb. Itt majd valamilyen módon, implementálnunk kellene, egy hibakezelést, arra való tekintettel, hogy ne lehessen egy állatot, egy irodaszobához rendelni, vagy egy alkalmazottat, egy kennelhez. Szükségünk lesz egy **állat** entitásra is, hogy eltároljuk a különféle állatokat, amik a szervezethez köthetőek. Tárolnunk kell az **alkalmazottakat** is, és az **örökbefogadók** személyeit. Ezenkívül egy **Jogviszony** táblában, eltároljuk a munkaköröket, és a hozzájuk tartozó fizetést.

Ebben a felállásban is 8 forgalmi adattábláról van szó, ami túllépi a követelményben megjelölt minimális mennyiséget. Ahhoz, hogy a weboldal megközelítse a valóságban „alkalmazható” jelzőt, még rengeteg entitásra lenne szükségünk, azonban az egyszerűség kedvéért nem bővítjük tovább az adatbázist, el kell fogadnom, hogy a jövőben ez okozhat *hiányos funkciókat*.

### 2.1. ER modell

A pluszba kapott egyedi azonosító mezőket, és az idegen kulcsokat nem tüntetjük fel a tervezés során.

1. Telep  
Egy Telep eltárolásához, elegendő a telep *megnevezése* (ami gyakorlatilag a város neve), és egy *leírás* mező.
2. Épület  
Az épület entitás kapcsán, elég tárolnunk az épület *megnevezését*, és *típusát*.
3. Szoba  
Egy épület alegysége, a szoba entitás lesz. Elegendő tárolnunk a *típusát*, és a *kapacitását*.
4. Állat  
Egy állat esetén, tárolnunk kell a *nevét*, *típusát*, *születési évét* és *nemét*. Az állat *életkora*, származtatott tulajdonság lesz.
5. Alkalmazott  
Az alkalmazottak esetén, tároljuk el az alkalmazott teljes *nevét*, a weboldalon használható *felhasználónevét* és *jelszavát*.
6. Jogviszony  
A Jogviszonynak tároljuk a *megnevezését*, és a *fizetés értékét*. Mivel, nem lehet, két ugyanolyan megnevezésű munkakör, külön fizetéssel, ezért nincs szükségünk, külön azonosító mezőre.
7. Örökbefogadó  
Egy örökbefogadó személy kapcsán, el kell tárolnunk az *nevét*, *lakhelyét*, jelenleg birtokolt *háziállatok számát*, és hogy *javasolt-e örökbefogadásra*.
8. Esemény  
Egy eseményről eltároljuk a *megnevezését*, *időpontját*, és *leírását*.

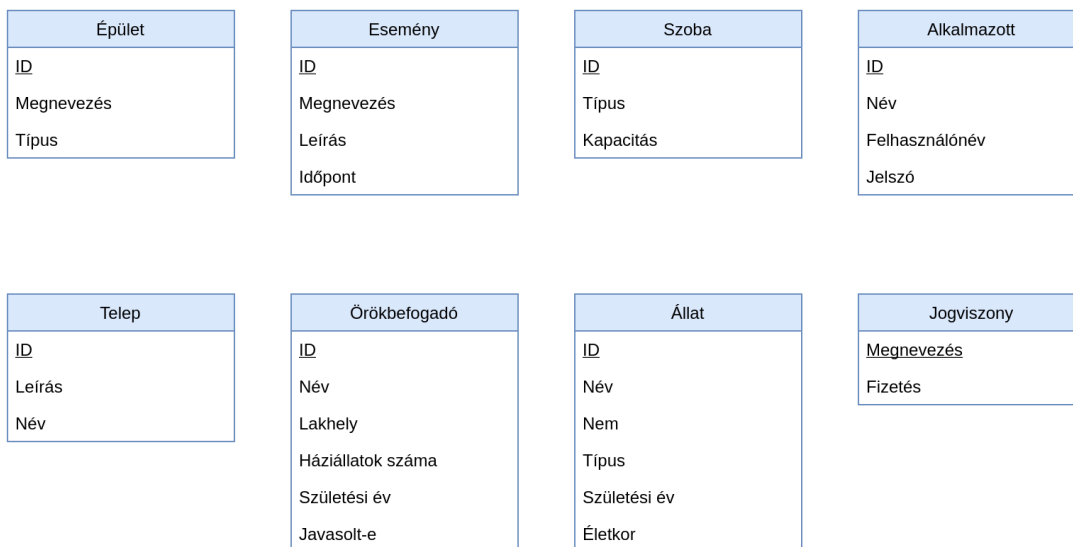


Az ER modell

„A megvalósítandó projekthez tartozó adatmodell legalább 5 forgalmi adattáblát tartalmaz (a külső modulok által használt táblákon kívül). Az adatmodell legalább 3. normálformában van és modellben megtalálható az egy-több, valamint a több-több típusú kapcsolat is. A technikai okokból létrehozott táblák (kapcsolótáblák) nem tekintendők forgalmi táblának.”

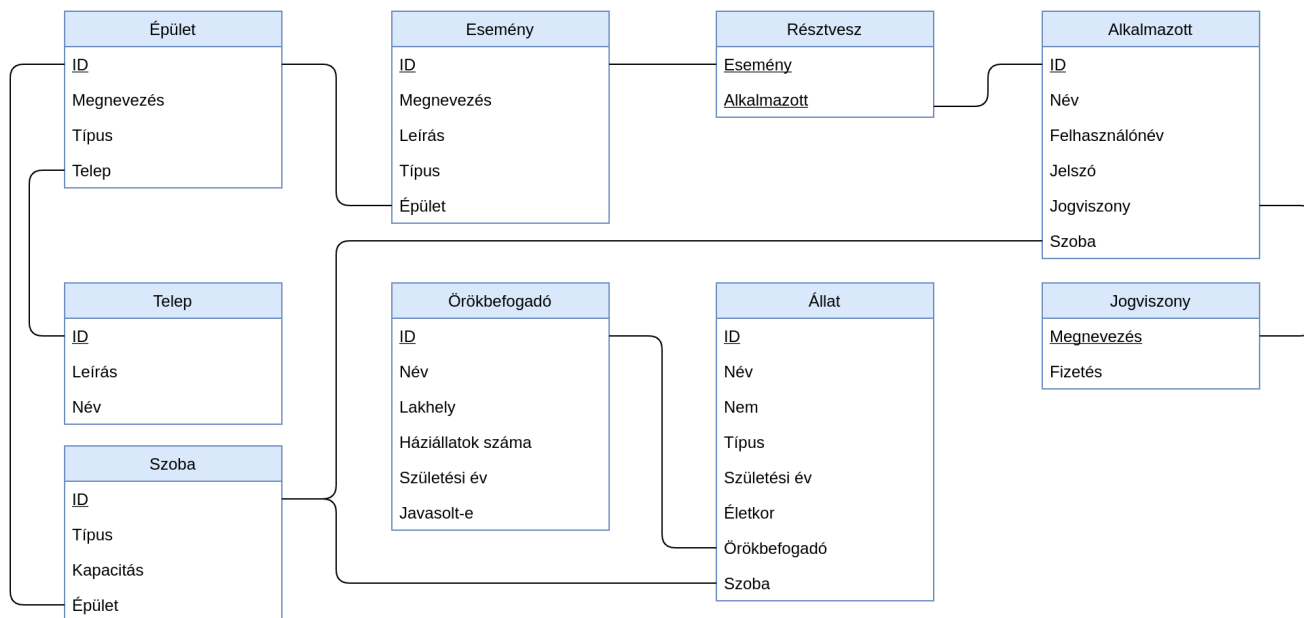
## 2.2. Relációs adatmodell

Az elkészült modell alapján, a következő táblák jöttek létre.



Az ER modell

Hozzuk létre a szükséges kapcsolótáblákat, és illesszük be az idegen kulcsokat, majd jelöljük a kapcsolatokat.



Az ER modell a kapcsolatokkal

## 2.3. SQL parancsok

Az SQL parancsok megtalálhatóak az application/create.sql fájlban.

## 3. Összefoglaló