

WEBPROGRAMOZÁS III. GY.

(LBT_IM744G2)

Beadandó feladat dokumentáció

Kovács Norbert
ANOXWJ

2021. június 8.

Tartalomjegyzék

1. Weboldal koncepciója	3
1.1. Technikai részletek	3
1.2. Elképzelés	3
2. Adatbázis tervezése	4
2.1. ER modell	4
2.2. Relációs adatmodell	5
2.3. SQL parancsok	5
2.3.1. Táblák létrehozása	5
3. Összefoglaló	5

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, raktárpépület 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 fejleszttem 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

Szükségünk lesz, egy **épület** entitásra, egy **'szoba'** entitásra. Ennek a megnevezése, nem teljesen fedi le a pontos entitást, hiszen minden épületben ez mást reprezentál, azonban szükséges eltárolnunk, hogy melyik állat melyik kennelhez van rendelve, vagy hogy melyik alkalmazott melyik irodában dolgozik stb. Szükségünk lesz egy **állat** entitásra is, hogy eltároljuk a különféle állatokat, amik a szervezethez köthetőek. Tárolnunk kell az **alkalmazottakat** is, és az **örökbefogadók** személyeit. Amennyiben lesz elég idő a fejlesztés során, az alkalmazottak **képzettségi mátrixát** is szeretném tárolni, ami egy gyakorlatból átemelt problémára nyújt megoldást.

(A megvalósítandó projekthez tartozó adatmodell legalább 5 forgalmi adattáblát tartalmaz (a külső modulok által használt táblákon kívül).)

2.1. ER modell

Az ER modell tervezése közben vegyük szemügyre nagyobb részletességgel az entitásokat. Az egyedi azonosító mezőket, nem tüntetjük fel a tervezés során, csak az ER modellben. A grafikus szemléltetésre a *Draw.io* asztali alkalmazását használom.

Telep

Egy **Telep** eltárolásához, elegendő a telep *megnevezése*, ami gyakorlatilag csak a város neve lesz, és egy *leírás* mező. Egy **Telephez** több **Épület** tartozhat majd, értelemszerűen egy **Épület**, csak egy **Telephez**.

Épület

Az **Épület** entitás kapcsán, tárolnunk kell a *megnevezését*, a *telepet*, és a *típusát*. Általánosságban kijelenthetjük, hogy egy **Telephez**, több azonos típusú épület is tartozhat (két iroda, két raktár).

Szoba

Egy **szoba**, egy épület alegysége. Tárolnunk kell a *kapacitását*. Attól függetlenül, hogy egy épület meghatározza, hogy milyen típusú szobákkal rendelkezik, ettől függetlenül, lehetőséget kell biztosítanunk a variációknak. Például egy Állatotthon is rendelkezhet kenneleken kívül váróval stb. Ezért tárolnunk kell a szoba *típusát* is.

Állat

Egy állat esetén, tárolnunk kell a *nevét*, *típusát*, *születési évét*, *nemét*, és hogy melyik „szobához” tartozik. Itt a későbbiekben, valamilyen hibaellenőrzésre szükség lesz, hogy egy állatot, ne lehessen hozzárendelni, egy iroda szobához, vagy egy irodai alkalmazottat, egy kennelhez. Amennyiben tárolni akarjuk, az örökbefogadás eseményét, akkor egy *tulajdonos* mezőt is fel kell vennünk.

Alkalmazott

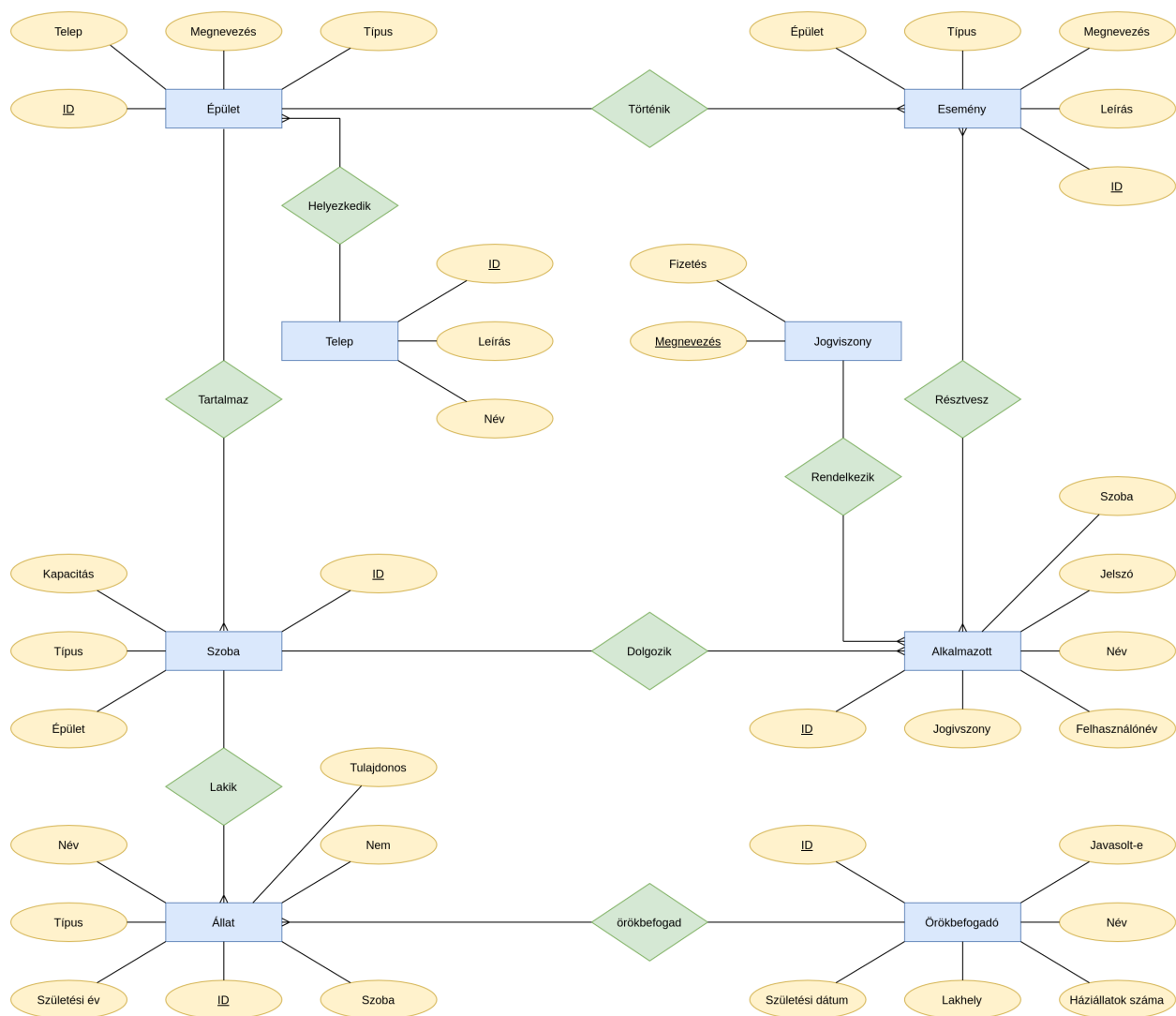
Az alkalmazottak esetén, tároljuk el az alkalmazott teljes *nevét*, *születési dátumát*, alkalmazotti *jogviszonyát*, weboldalon használható *felhasználónevét*, *jelszavát*, és az *épületet* amihez kirendelt. Az alkalmazott munkaköre, meghatározza majd a fizetést is, amit külön entitásként fogunk kezelni.

Jogviszony

A Jogviszonynak tároljuk a *megnevezését*, és a *fizetés értékét*. Mivel, nem lehet, két ugyanolyan megnevezésű munkakör, külön fizetéssel, ezért nincs szükségünk, külön azonosítómezőre.

Örökbefogadó

Egy örökbefogadó személy kapcsán, el kell tárolnunk az *nevét*, *születési évét*, *lakhelyét*, jelenleg birtokolt *háziállatok számát*, és hogy *javasolt-e örökbefogadásra*.



Az ER modell

2.2. Relációs adatmodell

2.3. SQL parancsok

2.3.1. Táblák létrehozása

3. Összefoglaló