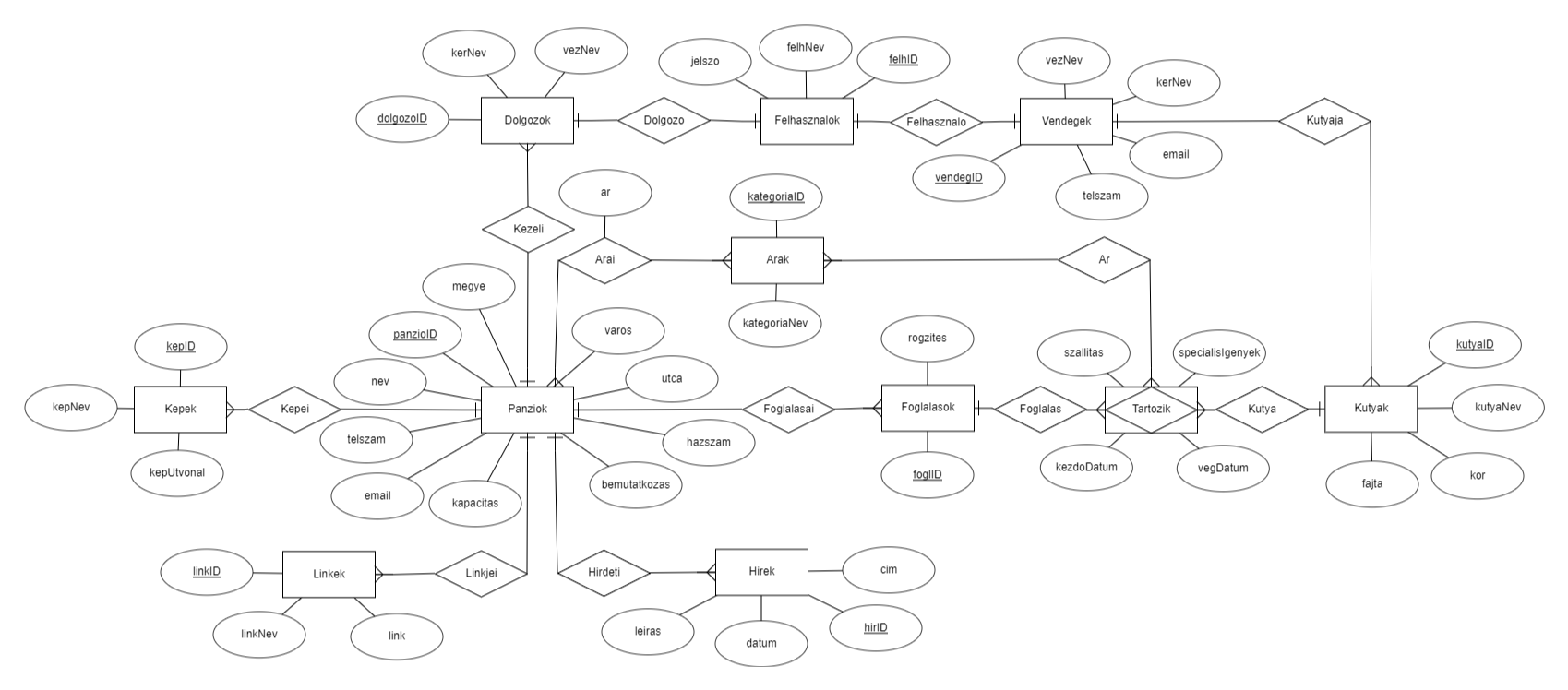
1. EK diagram
2. Leírás

Felhasznalok

Kötelező adat: felhNev, jelszo

Opcionális: jelenleg nincs, de bővíthető (profilkép, foglalkozás, stb…)

Automatikus: felhID

Vendegek (Foglaláshoz! Foglalás megvalósítható vendégként és felhasználóként is)

Kötelező adat: vezNev, kerNev, email, telszam

Opcionális: felh\_ID(ha van, akkor uniqe)

Automatikus: vendegID

Dolgozok

Kötelező adat: vezNev, kerNev, felh\_ID és panzio\_ID(dolgozó nem regisztrál, hanem fel kell venni az adatbázisba!)

Opcionális: jelenleg nincs

Automatikus: dolgozoID (Lehetne elsődleges kulcs a felh\_ID és panzio\_ID együtt, de a keresés a táblában így talán könnyebb.)

Panziok

Kötelező adat: nev, telszam, email, kapacitas, megye, varos (esetleges régióra szűrés miatt)

Opcionális: utca, hazszam, bemutatkozas

Automatikus: panzioID

Arak

Kötelező adat: kategoriaNev(pl. kis- vagy nagytestű, sétáltatás, külön kennel, stb.)

Opcionális: nincs

Automatikus: kategoriaID

Arai (Panzio-Arak N-M pl. fürdetés ugyanannyi 2 panzióban – ne legyen redundáns tárolás)

Kötelező adat: kategoria\_ID, panzio\_ID, ar(Ft)

Opcionális: nincs

Automatikus: nincs

Ar

Kötelező adat: kategoriaAr\_ID, foglAr\_ID, kutyaAr\_ID

Opcionális: nincs

Automatikus: nincs

Kepek

Kötelező adat: kepNev, kepUtvonal(oldal forrásainál), panzio\_ID

Opcionális: nincs

Automatikus: kepID

Linkek

Kötelező adat: linkNev (pl. facebook, instagram, stb.), link, panzio\_ID

Opcionális: nincs

Automatikus: linkID

Hirek

Kötelező adat: cim, leiras, panzio\_ID

Opcionális: jelenleg nincs

Automatikus: hirID, datum(current\_date)

Kutyak (ha vendég, akkor foglaláskor rögzül, ha felhasználó is, akkor felveheti a saját kutyáit)

Kötelező adat: kutyaNev, vendeg\_ID

Opcionális: kor, fajta

Automatikus: kutyaID

Foglalasok (Egy foglalás több kutyára szól, és egy kutyának több időpontot is lehet foglalni)

Kötelező adat: panzio\_ID

Opcionális: nincs

Automatikus: foglID, rogzites(now)

Tartozik

Kötelező adat: kezdoDatum, vegDatum, szallitas(kér/nem kér), fogl\_ID, kutya\_ID

Opcionális: specialisIgenyek(pl. fürdetés, sétáltatás, tanítás, stb.)

Automatikus: nincs

1. Modell elemei

Felhasznalok

felhID (szám) – elsődleges kulcs

felhNev (szöveg)

jelszo (szöveg)

Vendegek

vendegID (szám) – elsődleges kulcs

vezNev (szöveg)

kerNev (szöveg)

email (szöveg)

telszam (szöveg)

felh\_ID (szám) – külső kulcs

Dolgozok

dolgozoID (szám) – elsődleges kulcs

vezNev (szöveg)

kerNev (szöveg)

felh\_ID (szám) – külső kulcs

panzio\_ID (szám) – külső kulcs

Panziok

panzioID (szám) – elsődleges kulcs

nev (szöveg)

telszam (szöveg)

email (szöveg)

kapacitas (szám)

megye (szöveg)

varos (szöveg)

utca (szöveg)

hazszam (szöveg) (pl. 2/A miatt)

bemutatkozas (szöveg)

Arak

kategoriaID (szám) – elsődleges kulcs

kategoriaNev (szöveg)

Arai

kategoria\_ID (szám) – külső kulcs

panzio\_ID (szám) – külső kulcs

ar(szám)

kategoria\_ID, panzio\_ID – elsődleges kulcs

Ar

kategoriaAr\_ID (szám) – külső kulcs

foglAr\_ID (szám) – külső kulcs

kutyaAr\_ID (szám) – külső kulcs,

kategoriaAr\_ID, foglAr\_ID, kutyaAr\_ID – elsődleges kulcs

Kepek

kepID (szám) – elsődleges kulcs

kepNev (szöveg)

kepUtvonal (szöveg)

panzio\_ID (szám) – külső kulcs

Linkek

linkID (szám) – elsődleges kulcs

linkNev (szöveg)

link (szöveg)

panzio\_ID (szám) – külső kulcs

Hirek

hirID (szám) – elsődleges kulcs

cím (szöveg)

leiras (szöveg)

datum (dátum)

panzio\_ID (szám) – külső kulcs

Kutyak

kutyaID (szám) – elsődleges kulcs

kutyaNev (szöveg)

kor (szám)

fajta (szöveg)

vendeg\_ID (szám) – külső kulcs

Foglalasok

foglID (szám) – elsődleges kulcs

rogzites (dátum-idő)

panzio\_ID (szám) – külső kulcs

vendeg\_ID (szám) – külső kulcs

Tartozik

kezdoDatum (dátum)

vegDatum (dátum)

szallitas (boolean)

specialisIgenyek(szöveg)

fogl\_ID (szám) – külső kulcs

kutya\_ID (szám) – külső kulcs

fogl\_ID, kutya\_ID – elsődleges kulcs

1. SQL parancsok

CREATE TABLE IF NOT EXISTS Felhasznalok(

felhID int primary key auto\_increment,

felhNev varchar(50) not null,

jelszo varchar(100) not null

);

CREATE TABLE IF NOT EXISTS Vendegek(

vendegID int primary key auto\_increment,

vezNev varchar(50) not null,

kerNev varchar(50) not null,

email varchar(100) not null,

telszam varchar(20) not null,

felh\_ID int unique,

foreign key (felh\_ID) references Felhasznalok(felhID)

on update cascade on delete cascade

);

CREATE TABLE IF NOT EXISTS Panziok(

panzioID int primary key auto\_increment,

nev varchar(50) not null,

telszam varchar(20) not null,

email varchar(100) not null,

kapacitas int not null,

megye varchar(50) not null,

varos varchar(50) not null,

utca varchar(50),

hazszam varchar(10),

bemutatkozas varchar(1000)

);

CREATE TABLE IF NOT EXISTS Dolgozok(

dolgozoID int primary key auto\_increment,

vezNev varchar(50) not null,

kerNev varchar(50) not null,

felh\_ID int unique not null,

panzio\_ID int not null,

foreign key (felh\_ID) references Felhasznalok(felhID)

on update cascade on delete cascade,

foreign key (panzio\_ID) references Panziok(panzioID)

on update cascade on delete cascade

);

CREATE TABLE IF NOT EXISTS Arak (

kategoriaID int primary key auto\_increment,

kategoriaNev varchar(50) not null

);

CREATE TABLE IF NOT EXISTS Arai(

kategoria\_ID int not null,

panzio\_ID int not null,

ar int not null,

primary key(kategoria\_ID, panzio\_ID),

foreign key (kategoria\_ID) references Arak(kategoriaID)

on update cascade on delete cascade,

foreign key (panzio\_ID) references Panziok(panzioID)

on update cascade on delete cascade

);

CREATE TABLE IF NOT EXISTS Kepek (

kepID int primary key auto\_increment,

kepNev varchar(50) not null,

kepUtvonal varchar(100) not null,

panzio\_ID int not null,

foreign key (panzio\_ID) references Panziok(panzioID)

on update cascade on delete cascade

);

CREATE TABLE IF NOT EXISTS Linkek(

linkID int primary key auto\_increment,

linkNev varchar(50) not null,

link varchar(2050) not null,

panzio\_ID int not null,

foreign key (panzio\_ID) references Panziok(panzioID)

on update cascade on delete cascade

);

CREATE TABLE IF NOT EXISTS Hirek(

hirID int primary key auto\_increment,

cim varchar(50) not null,

leiras varchar(1000) not null,

datum date,

panzio\_ID int not null,

foreign key (panzio\_ID) references Panziok(panzioID)

on update cascade on delete cascade

);

CREATE TABLE IF NOT EXISTS Kutyak (

kutyaID int primary key auto\_increment,

kutyaNev varchar(30) not null,

kor int,

fajta varchar(30),

vendeg\_ID int not null,

foreign key (vendeg\_ID) references Vendegek(vendegID)

on update cascade on delete cascade

);

CREATE TABLE IF NOT EXISTS Foglalasok(

foglID int primary key auto\_increment,

rogzites datetime default now(),

panzio\_ID int not null,

foreign key (panzio\_ID) references Panziok(panzioID)

on update cascade on delete cascade

);

CREATE TABLE IF NOT EXISTS Tartozik(

kezdoDatum date not null,

vegDatum date not null,

szallitas boolean not null,

specialisIgenyek varchar(1000),

kutya\_ID int not null,

fogl\_ID int not null,

primary key(kutya\_ID, fogl\_ID),

foreign key (kutya\_ID) references Kutyak(kutyaID)

on update cascade on delete cascade,

foreign key (fogl\_ID) references Foglalasok(foglID)

on update cascade on delete cascade

);

CREATE TABLE IF NOT EXISTS Ar(

kategoriaAr\_ID int not null,

foglAr\_ID int not null,

kutyaAr\_ID int not null,

primary key(kategoriaAr\_ID, foglAr\_ID, kutyaAr\_ID),

foreign key (kategoriaAr\_ID) references Arak(kategoriaID)

on update cascade on delete cascade,

foreign key (foglAr\_ID) references Tartozik(fogl\_ID)

on update cascade on delete cascade,

foreign key (kutyaAr\_ID) references Tartozik(kutya\_ID)

on update cascade on delete cascade

);

1. Lekérdezések