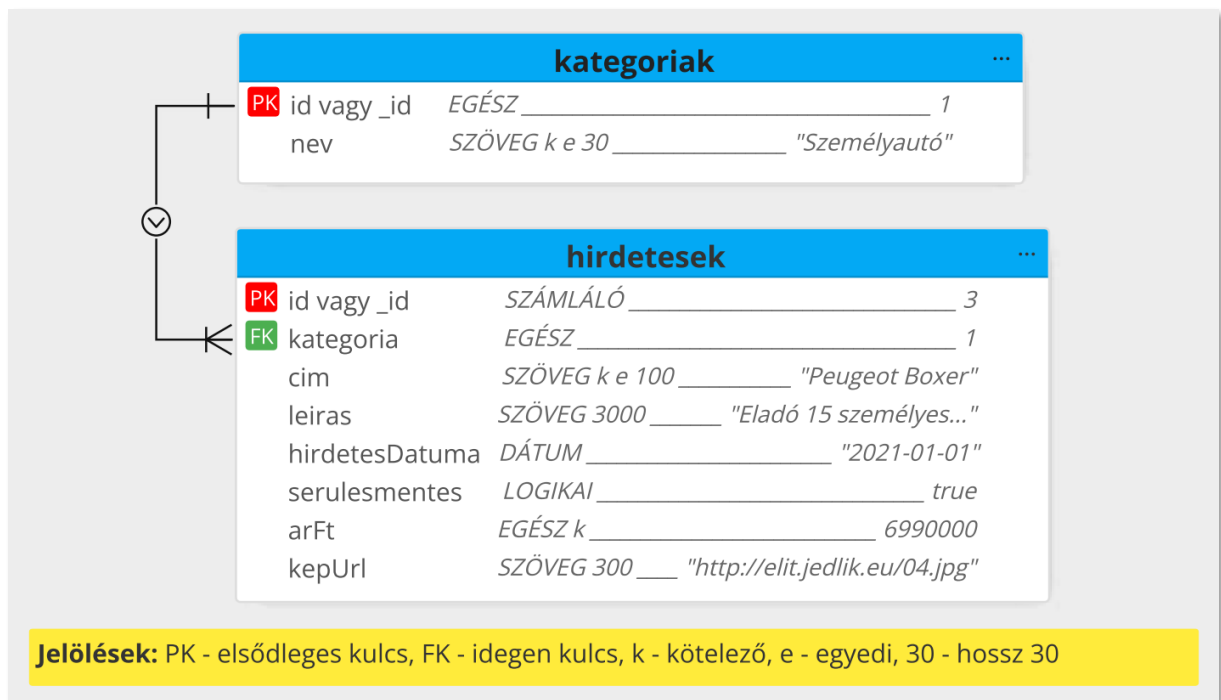


## Mintafeladat – Hahu - Backend programozás – NL

A következő feladatban egy használtautó kereskedés backend serverét kell elkészítenie a leírás és a minták szerint! Megoldása csak úgy lesz teljes értékű, ha a projektjében a modell(ek) és kontroller(ek) pontosan a feladatspecifikáció szerintiek, felesleges kódot nem tartalmaznak.

1. Hozzon létre backend server projektet a választott technológia szerint! A projektmappát „Vezetéknév\_Keresztnév\_backend” formában nevezze el!
2. Indítsa el számítógépén az adatbázis servert, hozza létre az adatbázist `hahu` néven, majd töltse fel táblákkal és mintaadatokkal a kiadott forrásállományok felhasználásával! Az adatbázis felépítését és a táblák kapcsolatát a következő ábra szemlélteti.



3. Hozzon létre modelleket a fenti ábra alapján a táblákhoz!
4. A `hirdetesDatuma` mező alapértelmezett értéke az aktuális dátum legyen, ha az nem kerül megadásra! A `kategoria` mező alapértelmezett értéke 1 legyen, ha az nem kerül megadásra!
5. A modell segítségével ellenőrizze le, hogy az `arFt` mező értéke osztható-e 1000-el! Ha nem osztható, akkor „Az ár nem osztható 1000-el!” hibaüzenetet rögzítse a modellben!
6. Készítsen végpontot `/api/hahu` route-on, melynek használatával **POST** típusú kéréssel új hirdetést tud rögzíteni az adatbázisban! A végponthoz tartozó kezelő (handler) kódsorait védje `try-catch` szerkezettel és hiba esetén a beépített hibaüzenetet küldje vissza a válaszüzenetbe a 400-as státuskóddal! Sikeres rögzítés után térjen vissza a következő (200 OK) válasszal:

```
{
  "message": "A rekord rögzítése sikeres!"
}
```

Egy lehetséges válasz hiba (400 Bad Request) esetén :

```
{
  "error": "arFt: Az ár nem osztható 1000-el!"
}
```

7. Készítsen végpontot `/api/hahu/:mezo` route-on, melynek használatával **GET** típusú kéréssel a hirdetések tábla összes adatát tudja a megadott mező szerint rendezve lekérdezni! Oldja meg, hogy a `katgoria` mezőbe ne a kategória azonosítója, hanem a kategória neve kerüljön a kapcsolódó `kategoriak` táblából a válaszüzenetbe! Egy lehetséges sikeres (200 OK) válasz részlete:

```
[
  {
    "id": 1,
    "katgoria": {
      "nev": "Kishaszonjármű"
    },
    "cim": "Renault Trafic 1.9 dCi L1H2 Business",
    "leiras": "A HŰTŐ HIBÁTLANUL MŰKÖDIK. Az ...",
    "hirdetesDatuma": "2022-02-21T00:00:00.000Z",
    "serulesmentes": true,
    "arFt": 999999,
    "kepUrl": "http://elit.jedlik.eu/nits/hahu/01.jpg"
  },
  ...
]
```

8. Készítsen végpontot a `/api/hahu/:id` route-on, melynek használatával **DELETE** típusú kéréssel egy megadott azonosítójú hirdetést tud törölni! Törlés előtt, ellenőrizze, hogy létezik-e a megadott azonosítójú hirdetés! Ha a hirdetésnem létezik, akkor „A hirdetés `#{id}` azonosítóval nem létezik!” üzenetet és a 404-es státuskódot küldje válaszként! Sikeres törlés esetén a „A hirdetés `#{id}` azonosítóval törölve!” válaszüzenetet (message) és a 200-as státuskódot küldje válaszként! Minta válaszok:

```
{
  "message": "A hirdetés 19 azonosítóval törölve!"
}

{
  "error": "A hirdetés 2022 azonosítóval nem létezik!"
}
```

9. Hozzon létre **Thunder Client** kollekción „teszt” néven melyben az Ön által létrehozott végpontokat ellenőrzi! Exportálja a kollekción a projektmappába „thunder-collection\_teszt.json” néven!
10. **Projektmappáját** tömörítve másolja a „Z:” megható főkönyvtárába!  
(Vezetéknév\_Keresztnév\_backend.ZIP néven) Tömörítés előtt törölje a felesleges állományokat!  
**Ügyeljen arra, hogy feladatmegoldást tartalmazó mappát/állományt ne töröljön!**