# Táblák és egyedeik leírása

## **Kategóriák:**

* **Kat**\_**id** [PK] [AI] érték
* **Rendszam**\_**FK** [FK] ként, az ’Autok’ táblától kapja meg az adott autónak a rendszámát.
* **Kat**\_**besorolas** szerint 1-től 5-ig tart a besorolási szám, az alábbi kódrészlet alapján:
* private function kategoriaBesorolas(int $teljesitmeny): int
* {
* if ($teljesitmeny == 18) {
* return 1;
* } elseif ($teljesitmeny == 36) {
* return 2;
* } elseif ($teljesitmeny == 45) {
* return 3;
* } elseif ($teljesitmeny == 50) {
* return 4;
* } else {
* return 5;
* }
* }

## **FLOTTA\_TIPUSOK tábla**

* Flotta\_id [PK] [AI] érték
* Gyarto
* Tipus
* Teljesitmeny
* Vegsebesseg
* Gumimeret
* Hatotav

Az adatok **FIXEK** itt. Ismétlődés nélkül a járműpark felsorolása fajtánként, az alábbiak szerint:

[

    {

        "Flotta\_Flotta\_id": 1,

        "gyarto": "VW",

        "tipus": "e-up!",

        "teljesitmeny": 18,

        "vegsebesseg": 130,

        "gumimeret": "165|65-R15",

        "hatotav": 130

    },

    {

        "Flotta\_id": 2,

        "gyarto": "VW",

        "tipus": "e-up!",

        "teljesitmeny": 36,

        "vegsebesseg": 130,

        "gumimeret": "165|65-R15",

        "hatotav": 300

    },

    {

        "Flotta\_id": 3,

        "gyarto": "Skoda",

        "tipus": "Citigo-e-iV",

        "teljesitmeny": 36,

        "vegsebesseg": 130,

        "gumimeret": "165|65-R16",

        "hatotav": 300

    },

    {

        "Flotta\_id": 4,

        "gyarto": "Renault",

        "tipus": "Kangoo-Z.E.",

        "teljesitmeny": 45,

        "vegsebesseg": 130,

        "gumimeret": "165|65-R15",

        "hatotav": 285

    },

    {

        "Flotta\_id": 5,

        "gyarto": "Opel",

        "tipus": "Vivaro-e",

        "teljesitmeny": 50,

        "vegsebesseg": 192,

        "gumimeret": "165|65-R16",

        "hatotav": 320

    },

    {

        "Flotta\_id": 6,

        "gyarto": "KIA",

        "tipus": "Niro-EV",

        "teljesitmeny": 65,

        "vegsebesseg": 167,

        "gumimeret": "165|65-R17",

        "hatotav": 350

    }

]

Csak ez a 6 fajta autó lehet. Amikor egy Auto generálódik az Autok táblában, akkor az egyik (1-től 6-ig) tartó autót generálja, és annak adatait használja fel.

## **AUTOK tábla**

Autok\_id [PK] [AI] érték

Flotta\_id [FK] érték, a FLOTTA\_TIPUSOK táblától kapja meg.

Rendszam

Gyartasi\_ev

Km\_ora\_allas

A rendszám a generálás során egyedileg régi és új rendszám generálódik.

Gyártási év generálásánál 2019 és 2023 között random generálunk egy gyártási évet.

   $auto\_kivalasztas = array\_rand($auto\_adatok);

    $auto\_adat = $auto\_adatok[$auto\_kivalasztas];

    do {

        $rendszam\_uj\_regi = random\_int(0, 1);

        if ($rendszam\_uj\_regi > 0) {

            $rendszam = strtoupper($faker->regexify('AA[A-C][A-O]-[0-9]{3}'));

        } else {

            $rendszam = strtoupper($faker->regexify('(M|N|P|R|S|T)[A-Z]{2}-[0-9]{3}'));

        }

    } while (in\_array($rendszam, $rendszamok));

    $rendszamok[] = $rendszam;

    $gyartasi\_ev = random\_int(2019, 2023);

A ’ Km\_ora\_allas’ a ’ Gyartasi\_ev’ -re épülve generál neki adott km mennyiséget az autohoz.

private function kmOraAllasGeneralas(int $gyartasi\_ev): int

    {

        switch ($gyartasi\_ev) {

            case 2019:

                return random\_int(50\_000, 60\_000);

            case 2020:

                return random\_int(40\_000, 60\_000);

            case 2021:

                return random\_int(30\_000, 40\_000);

            case 2022:

                return random\_int(25\_000, 35\_000);

            case 2023:

                return random\_int(20\_000, 30\_000);

            default:

                return 0;

        }

    }

## **FELSZERELTSEGEK tábla**

* **Felsz\_id** [PK] [AI] érték
* **Rendszam\_FK** [FK] érték az ’Autok’ táblának a rendszám értékét veszi át.
* **Tolatokamera** (0 vagy 1-es érték, hogy van, vagy nincs az autóban),
* **Savtarto** (0 vagy 1-es érték, hogy van, vagy nincs az autóban),
* **Tempomat** (0 vagy 1-es érték, hogy van, vagy nincs az autóban),
* **Tolatoradar** (0 vagy 1-es érték, hogy van, vagy nincs az autóban),
* **Multif\_kormany** (0 vagy 1-es érték, hogy van, vagy nincs az autóban).

## **SZEMELYEK tábla**

* Szemely\_id [PK] [AI] érték
* Felh\_jelszo (csak 4 szám lehet)
* Szig\_szam
* Jogos\_szam
* Jogos\_erv\_kezdete (datum)
* Jogos\_erv\_vege (datum)
* V\_nev
* K\_nev
* Szul\_datum
* Telefon
* Email

## **FELHASZNALOK tábla**

* Felh\_id [PK] [AI] érték
* Szemely\_id\_FK [FK] érték a ’SZEMELYEK’ táblának a ’szemely\_id’ értékét veszi át.
* **Felh\_nev** a ’SZEMELYEK’ táblának a ’V\_nev’ értékéből illetve random generáljuk le, az alábbiak szerint:
* function felhasznaloNevGenerator(array &$felhNevek\_ell\_tar\_tomb, $V\_nev, $faker): string
* {
* do {
* $tipusValaszto = random\_int(0, 1);
* if ($tipusValaszto === 0) {
* $felhNev\_vege = $faker->numberBetween(1\_000, 9\_999);
* $keszFelhNev = $V\_nev . $felhNev\_vege;
* $keszFelhNev = strtr($keszFelhNev, 'áéíóöőúüűÁÉÍÓÖŐÚÜŰ', 'aeiooouuuAEIOOOUUU');
* } else {
* $felhNev\_eleje = $faker->word;
* $felhNev\_vege = $faker->numberBetween(100, 999);
* $keszFelhNev = $felhNev\_eleje . $felhNev\_vege;
* $keszFelhNev = strtr($keszFelhNev, 'áéíóöőúüűÁÉÍÓÖŐÚÜŰ', 'aeiooouuuAEIOOOUUU');
* }
* } while (in\_array($keszFelhNev, $felhNevek\_ell\_tar\_tomb));
* $felhNevek\_ell\_tar\_tomb[] = $keszFelhNev;
* return $keszFelhNev;
* **Elofiz\_kat** Random elővizetési kategóriába soroljuk a felhasználót, az alábbi értékekből választva:
* private array $elofizKategoriak = ["Power", "Power-Plus", "Power-Premium", "Power-VIP"];
* private function getRandomElofizKategoria(): string
* {
* $randomelofizetes = array\_rand($this->elofizKategoriak);
* return $this->elofizKategoriak[$randomelofizetes];
* }
* Jelszo\_masodik\_negyed a ’SZEMELYEK’ táblának a felh\_jelszo’ értékét veszi át és annak a 2. és 4. karaktere lesz összefűzve az értéke ennek.

# FOLYTATÁS

## **1. Laravel migrációk létrehozása**

* **Táblák létrehozása migrációval:**
  + **autok tábla:** Az autók alapvető adatai, mint a rendszám, flotta\_id, gyártási év és kilométeróra állás kerültek ide.
  + **flotta\_tipusok tábla:** A flotta azonosítója és a teljesítmény értékek meghatározására szolgál.
  + **kategoriak tábla:** Az autók rendszámának és kategóriabesorolásának tárolása.
  + **felszereltsegek tábla:** Az autók felszereltségi jellemzői, például tolatókamera, tempomat stb.

## **2. Seederek létrehozása és implementálása**

* **Seeder alapú adatok generálása Faker segítségével.**
* **Seederek felépítése és működése:**
  + **AutokSeeder:**
    - 200 autó generálása, ahol:
      * Rendszám kétféle (új és régi formátum) lehet.
      * A flotta\_id meghatározása az előre megadott valószínűségek alapján történik (pl. 85%-ban 1-3-as flotta\_id).
      * A gyártási év alapján a kilométeróra állás generálása.
  + **KategoriakSeeder:**
    - Az autókhoz tartozó kategóriabesorolás meghatározása a flotta\_id alapján. A teljesítmény (18, 36, 45, 50, 65) alapján kapják meg az 1-től 5-ig terjedő besorolási számot.
  + **FelszereltsegekSeeder:**
    - Az autók felszereltségi adatai a kategóriabesorolás alapján generálódnak. Például:
      * 1-es kategóriában kisebb az esély a teljes felszereltségre.
      * 5-ös kategóriában minden felszereltség automatikusan 1 (igen).

## **3. Helper osztályok a seedeléshez**

A seederek logikájának tisztán tartása érdekében minden komplex generálási folyamatot külön helper osztályba helyeztem:

* **AutokSeederHelper:**
  + Rendszám generálása (új és régi formátum).
  + Flotta\_id valószínűségi generálása.
  + Kilométeróra állás generálása a gyártási év alapján.
* **KategoriakSeederHelper:**
  + A kategóriabesorolás generálása az autók flotta\_id-ja és a teljesítmény alapján.
* **FelszereltsegekSeederHelper:**
  + A felszereltségek generálása kategóriabesorolás szerint.

## **4. Composer autoload konfiguráció**

* A composer.json fájlban a SeederHelperek namespace hozzáadása a database/seeder\_helperek/ mappához.
* Az autoload újragenerálása: composer dump-autoload.

## **5. Teljes adatbázis feltöltése**

* Az adatbázis feltöltése a seederek futtatásával történt:
  + php artisan db:seed --class=AutokSeeder
  + php artisan db:seed --class=KategoriakSeeder
  + php artisan db:seed --class=FelszereltsegekSeeder

## **Eredmény**

* A autok, kategoriak, és felszereltsegek táblák sikeresen feltöltésre kerültek.
* Az adatok helyesek és megfelelnek az előírt követelményeknek.
* Az adatok generálása teljesen dinamikus, Faker és valószínűségi logika alapján történik