Backend építése Laravelben

## Összes eddigi tananyag felfűzve [itt](https://docs.google.com/document/d/1ybnWk0tEwyhRDPbPCpyBNf4YEybYqK1jbMC2fahuEJ4/edit?usp=sharing) található (folyamatosan bővül).

A múlt heti tananyag már beépítve található [itt](https://github.com/Agnes-milia/book_with_multiply_keys.git) (többszörös primary key megoldva!), használd bátran kiindulásként a mai órához!

Az importálandó json fájlt megtalálod a json mappában a fenti linken.

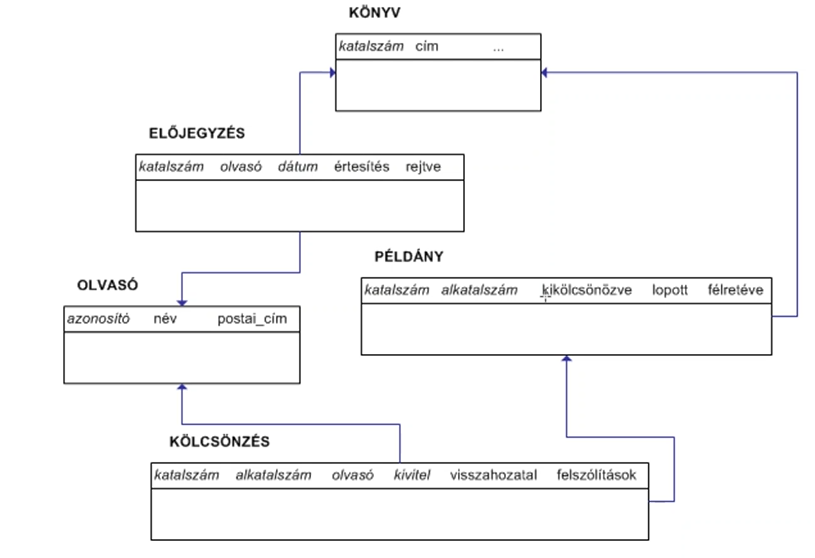
Egy újabb táblával bővítsük az adatbázisunkat: előjegyzés (*reservation*).

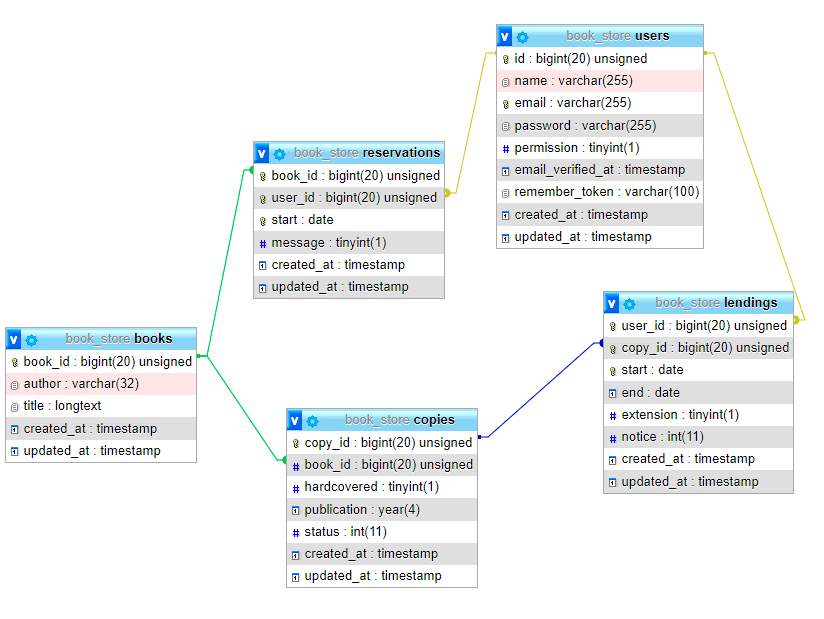
Itt is összetett kulcsot állítsatok be a lendings táblához hasonlóan! (kép a következő oldalon)

A kölcsönzés táblában is vegyünk fel újabb mezőket, ne felejtsük el, hogy mi mindent kell még hozzá beállítani (Model és Controller fájlokban is lesz változás).

Érdemes még egy *messages* táblával is bővíteni a továbbiakban, ahol az előjegyzéshez tartozó üzenetek, értesítéseket tárolnánk dátummal, és egy *dictionary* táblával, ahol az üzenetek sablon szövegeit tárolhatjuk pl.

Alább egy elképzelés található, a miénk kicsit más, de hasonló (az eredeti Kupcsikné Ica terve).





## Lekérdezések Laravelben[[1]](#footnote-0)

1. Eloquant használatával (**with** segítségével)
2. DB::table …
   1. **select (->get()),**
   2. **where**, whereNot, orWhere, whereNull, whereIn, whereDate, whereColumn…
   3. **aggregáló függvények (pl. ->count()),**
   4. **groupBy, having**, orderBy
   5. **raw**, azaz nyers sql használata
   6. **join**-ok használata
   7. unió,
   8. view-k sql-ben,
   9. triggerek,
   10. constraintek

## 

## Lekérdezések DB::table segítségével

importálás: use Illuminate\Support\Facades\DB;

Alap szintaxis:

// egy adott könyvhöz hány példány tartozik

public function book\_copy\_count($title){

$copies = DB::table('copies as c') //egy tábla lehet csak

//->select('mezo\_neve') //itt nem szükséges

->join('books as b' ,'c.book\_id','=','b.book\_id') //kapcsolat leírása, akár több join is lehet

->where('b.title','=', $title) //esetleges szűrés

->count(); //esetleges aggregálás; ha select, akkor get() a vége

return $copies;

}

## Konkrét lekérdezések az adatbázisban (DB::table gyakorlása)

**Elsőként írjuk meg a fent megírt lekérdezéseket ezzel a módszerrel is!**

1. Hány darab előjegyzése van a bejelentkezett felhasználónak?
2. Csoportosítsd szerzőnként a könyveket (nem példányokat) a szerzők ABC szerinti növekvő sorrendjében!
3. Határozd meg a könyvtár nyilvántartásában legalább 2 könyvvel rendelkező szerzőket!
4. A B betűvel kezdődő szerzőket add meg!
5. **A bejelentkezett felhasználó 3 napnál régebbi előjegyzéseit add meg! (együtt)**
6. **Bejelentkezett felhasználó azon kölcsönzéseit add meg (copy\_id és db), ahol egy példányt legalább db-szor (paraméteres fg) kölcsönzött ki! (együtt)**
7. **Hosszabbítsd meg a könyvet, ha nincs rá előjegyzés! (együtt)**

A lekérdezésekhez létrehozott útvonalat teszteld TC-ben is, a böngészőben is (mind GET kérés)!

## Validációs feladat

Tegyük fel, hogy csak 1000 vagy azután kiadott könyvek találhatóak a könyvtárban.

Csak akkor engedd módosítani a copies táblában a publication nevű mezőt, ha 1-sel vagy 2-sel kezdődik az év, és 4 jegyű a szám!

(ehhez nem lesz jó a year típus, mert nem engedi a régebbi éveket, helyette:  
$table->integer('publication')->length(4)->default(2000);)

**Készíts legalább 5 működő Unit tesztet a meglévők alapján a lekérdezésekhez!**

1. <https://laravel.com/docs/9.x/queries> [↑](#footnote-ref-0)