#1 *Lépjen be a MongoPlayground oldalára, majd a melléklet 1. pontjában szereplő gyűjteményt másolja be a Configuration részbe!*

1. *Készítsen lekérdezést, amely csak az user\_id, firstName és lastname oszlopokat jeleníti meg!*
2. *A lekérdezés kódját adja meg válaszként!*

db.collection.find({},

{

user\_id: 1,

firstName: 1,

lastname: 1,

\_id: 0

})

*#2 A MongoPlayground-on az előző feladatban létrehozott gyűjteményből kérdezze le a Grace keresztnevű felhasználó email-címét és jelszavát (csak ez a két mező jelenjen meg)!*

db.collection.find({

firstName: "Grace"

},

{

user\_id: 1,

firstName: 1,

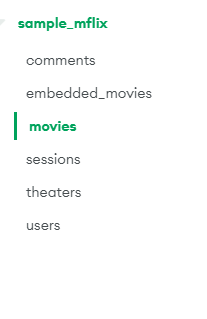
lastname: 1,

\_id: 0

})

*#3 Lépjen be a MongoDB Atlas-ba, majd értelemszerűen hozzon létre új szervezetet, projektet és cluster-t!*

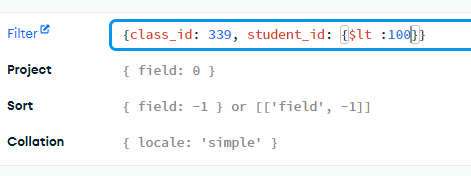
1. *A Sandbox részben a három pontra (...) kattintva válassza ki a Load Sample Dataset lehetőséget! (A dataset betöltése több percet is igénybe vehet)*
2. *Kattintson a Browse Collections gombra!*
3. *Kattintson az adatbázisok közül a sample\_training-re, majd azon belül a grades gyűjteményre!*
4. *A gyűjtemény dokumentumaiból a képernyőn látható részt csatolja válaszként kép formájában!*



*#4 Az előző feladatban kiválasztott grades gyűjteményből kérdezze le a 339-es azonosítójú osztály eredményeit!*

*A listában csak azok a dokumentumok jelenjenek meg, ahol a tanuló azonosítója 100 alatt van!*

*A lekérdezést adjuk meg válaszként!*

**

*#5 A MongoDB Atlas-ban a + Create Database gomb lenyomásával hozzunk létre új adatbázist Gyak8 néven, azon belül pedig egy új gyűjteményt szemelyek néven! A létrehozás után tegye aktuálissá a Gyak8.szemelyek gyűjteményt!*

*Az Insert Document gomb lenyomása után szúrjunk be új dokumentumot a szemelyek gyűjteménybe a következő tartalommal (értéknek mindenhova a saját adatait adja meg): neptunkod, nev, evfolyam*

*A neptunkod legyen string típusú*

*A név két mezőből álljon: vezeteknev és keresztnev (Object típus)*

*Az evfolyam egész szám legyen (Int32)*

*A beszúrás után a rögzített adatokat megjelenő képernyőrészletet csatoljuk válaszkét kép formájában!*

**

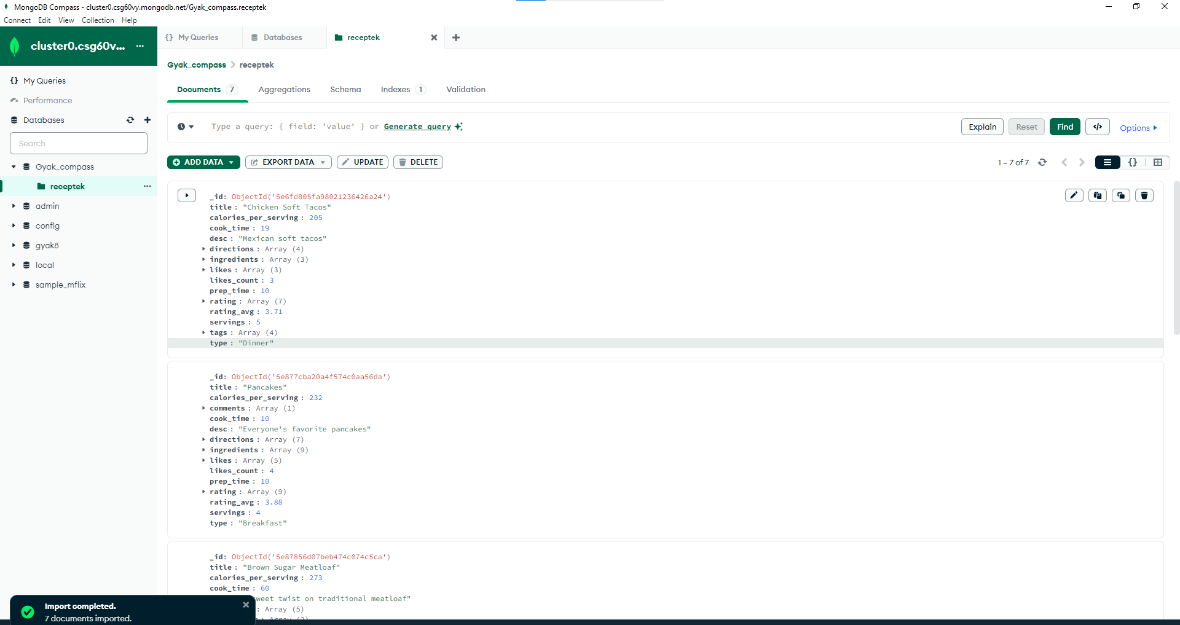
*#6 Indítsa el a MongoDB Compass alkalmazást, majd csatlakozzon a MongoDB cluster-hez!*

*Ha múltkor sikerült a kapcsolódás, akkor baloldalt a Recent részben kiválaszthatja a kapcsolatot Sikertelen kapcsolat esetén a baloldalon lévő Network Access részben adja hozzá az aktuális IP-címét a tűzfal kivételekhez!*

*Hozzon létre új adatbázist Gyak\_compass néven, azon belül egy új gyújteményt receptek néven!*

*A receptek gyűjteménybe importálja be a 2. melléklet tartalmát (Add data / Import File, majd Select File, végül Import)!*

*Az importálás után az adatokat megjelenítő képernyőrészletet csatolja válaszként kép formájában!*

**

*#7 A MongoDB Compass segítségével kérdezze le az előző feladatban létrehozott receptek gyűjtemény azon dokumentumait, amelyre teljesül:*

*A lájkok száma több, mint 2!*

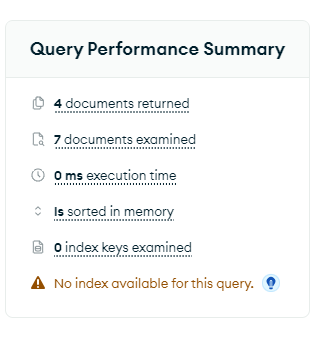
*A lista legyen sorbarendezve a főzési idő szerint csökkenő sorrendben! (A rendezés funkció az Options gomb lenyomása után érhető el)*

*A listában ne jelenjenek meg az ingredients és a rating mezők (Project szakasznál kell beállítani)!*

*A lekérdezést összeállító képernyőrészletet csatolja válaszként kép formájában!*

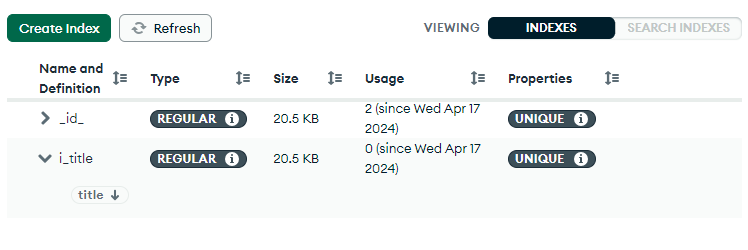


*#8 Az előző feladatban létrehozott lekérdezésre hajtsa végre az Explain Plan funkciót!*

1. *Az eredményként megjelenő Query Performance Summary-t mutató képernyőrészletet csatolja válaszként kép formájában!*

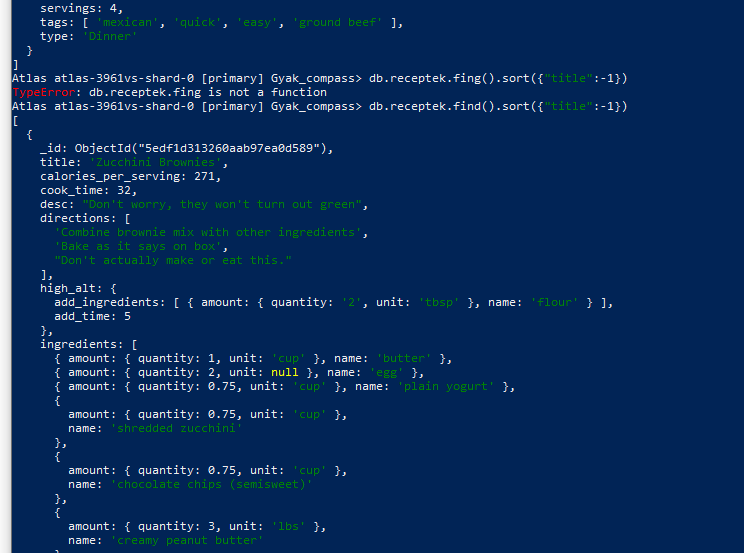
*#9 A MongoDB Compass-ban készítsen új indexet a 7. feladatban importált receptek gyűjteményhez az Indexes rész Create Index funkciójának segítségével!*

1. *Az index neve legyen i\_title, és a title mező szerint csökkenő legyen*
2. *Az index egyedi (unique) legyen (Options rész)!*
3. *Az index létrehozása után megjelenő képernyőt csatoljuk a válaszhoz kép formájában!*

**

*#9 A MongoDB Atlas-ban navigáljunk a Cluster-hez, majd válasszuk a Connect lehetőséget, ezen belül pedig a "Connect with the mongodb shell" opciót!*

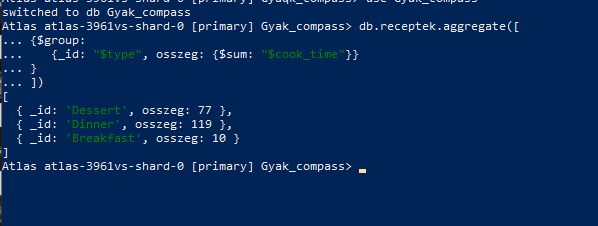
1. *Töltsük le a mongo shell állományt, majd tömörítsük ki egy mappába (pl. Dokumentumok)*
2. *A fájlkezelőben lépjünk be a mongo shell bin mappájába, majd nyissunk egy parancssort!*
3. *Csatlakozzunk a Cluster-hez a Connect to Cluster ablakban megjelenő connection string segítségével!*
4. *Adjuk ki a show dbs parancsot, majd a parancssor ablakának tartalmát kép formájában adjuk meg válaszként!*

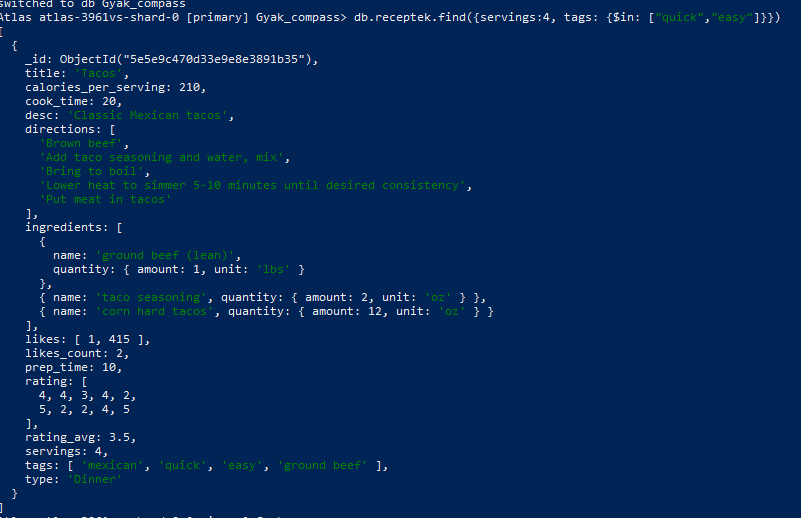
**zh-ban: grafikus felületen egy lekérdezés + shellben parancsokat összerakni, fontos a kapcsolat (felhő)!!!

*#12 A mongo shell-ben kérdezzük le, hogy recept típusonként (type) mennyi a főzési idők (cook\_time) összege!*

1. *A lekérdezés kódját adjuk meg válaszként!*

db.receptek.aggregate([ {$group: {\_id: "$type", osszeg: {$sum: "$cook\_time"}} } ])

**

*#13*

*#17*

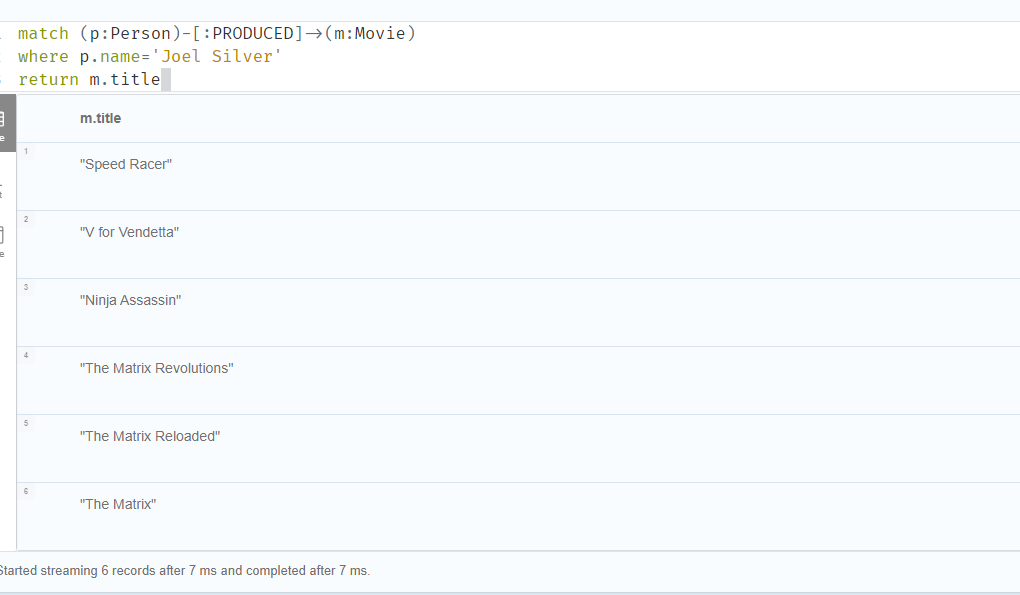
match (p:Person) where p.born=1965 or p.born=1964 return p.name, p.born

*#18 A Neo4J Sandbox Movie adatbázisából kérdezze le azon filmek címét és megjelenési évét, amelyek címe A-betűvel kezdődik! (WHERE, STARTS WITH).*

1. *A listát rendezzük a megjelenési év szerint csökkenő sorrendbe (ORDER BY)!*
2. *Az utasítás kódját adjuk meg válaszként!*

**

*#19 A Neo4J Sandbox Movie adatbázisából kérdezze le, hogy milyen filmeket készített (:PROCUCED) Joel Silver!*

1. *Csak a filmek címe jelenjen meg*
2. *A lekérdezés eredményét mutató képernyőrészletet csatolja válaszként kép formájában!*
3. **

*#20 A Neo4J Sandbox Movie adatbázisából kérdezze le, hogy melyik rendező hány filmet rendezett! (:DIRECTED).*

*Csak azokat a rendezőket jelenítsük meg, akik 1-nél több filmet rendeztek! (WHERE)*

*A lekérdezés kódját adja meg válaszként!*

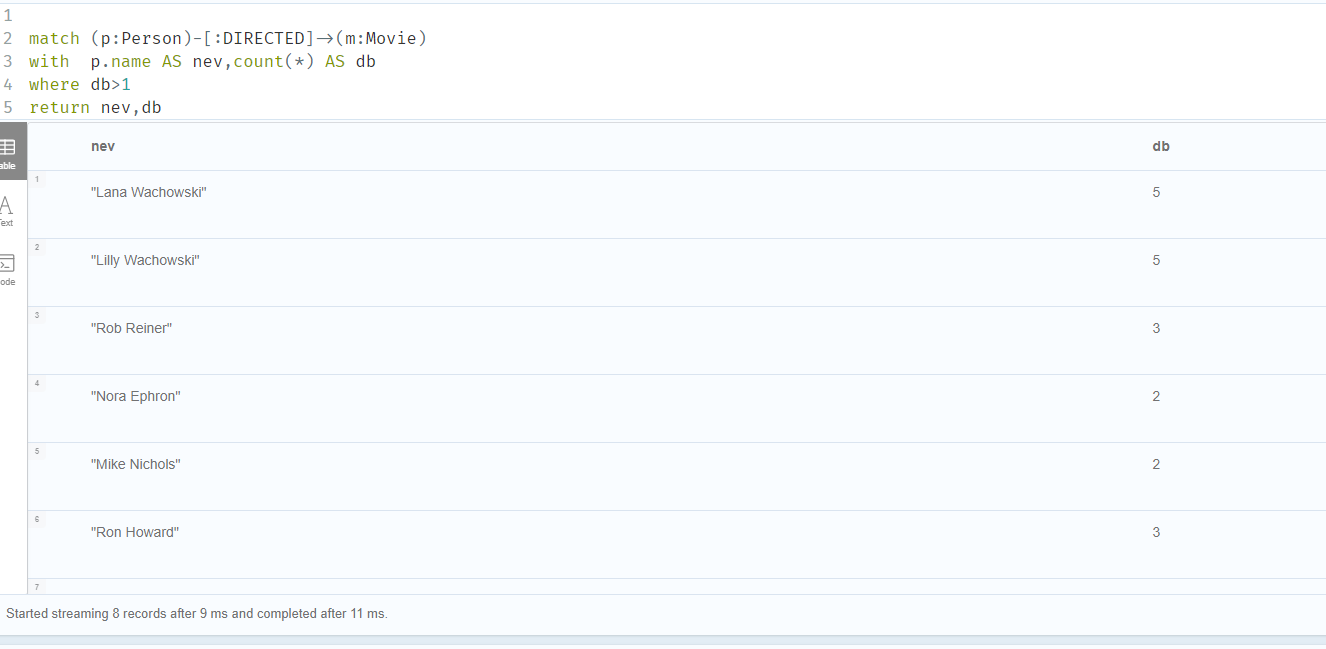
(NINCS HAVING!!!!)

match (p:Person)-[:DIRECTED]->(m:Movie)

with  p.name AS `nev`,count(\*) AS `db`

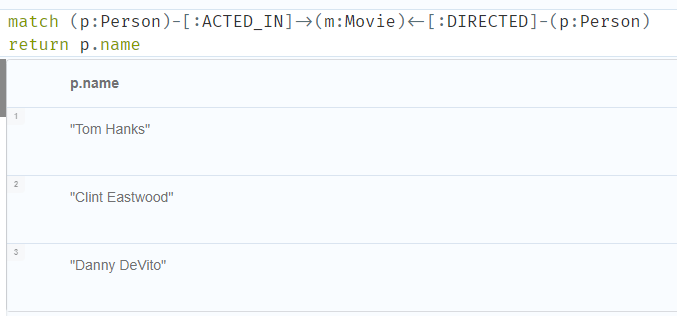
where db>1

return nev,db

**

*#21 A Neo4J Sandbox Movie adatbázisából jelenítsük meg azokat a személyeket, akik egyszerre voltak szereplők és rendezők is valamely filmben!*

1. *A lekérdezés kódját adjuk meg válaszként!*

**