# Adatbázis beadandó – blogfelület adatbázisa

## 1. relációs adatbázis leírása

A relációs adatbázis így épült volna fel:

1. tábla: felhasznalok
   1. felhasznalo\_id
   2. vezeteknev
   3. keresztnev
   4. email
2. tábla: rovatok
   1. rovat\_id
   2. rovat\_neve
3. tábla: bejegyzesek
   1. bejegyzes\_id
   2. rovat\_id
   3. felhasznalo\_id
   4. cim
   5. kozzetetel\_datum
   6. tartalom
   7. megtekintesek\_szama
4. tábla: kommentek
   1. felhasznalo\_id
   2. bejegyzes\_id
   3. iras\_datuma
   4. szoveg

A felhasznalo\_id mező összekapcsolja a regisztrált felhasználókat az általuk írt cikkekkel és kommentekkel (a felhasznalok táblát a bejegyzesek és a kommentek táblával).

A rovat\_id mező összekapcsolja a rovatokat az azokban szereplő bejegyzésekkel (a rovatok táblát a bejegyzesek táblával)

A bejegyzes\_id mező összekapcsolja a kommenteket azzal a bejegyzéssel, amihez lettek írva (a kommentek táblát a bejegyzesekkel)

## 2. NoSQL-adatbátis:

NoSQL adatbázisnak egy dokumentum adatbázist választottam, ami a következő szerkezetet követi. Az adatokat a MongoDB Compass segítségével töltöttem fel.

1. Collection: felhasznalok
   1. f\_felhasznaloID
   2. vezeteknev
   3. keresztnev
   4. email
2. Collection: bejegyzesek
   1. b\_bejegyzesID
   2. b\_felhasznaloID
   3. cim
   4. kozzetetel\_datum
   5. tartalom
   6. megtekintesek\_szama
   7. rovat:
      1. rovatID
      2. rovat\_neve
3. Collection: kommentek
   1. k\_felhasznaloID
   2. k\_bejegyzesID
   3. iras\_datuma
   4. szoveg

## 3. Lekérdezések

A legkérdezéseket a MongoShell használatával végeztem a létrehozott adatbázison.

1. Összes bejegyzés címének listázása

db.bejegyzesek.find({}, {cim: 1,\_id: 0})

1. A „szolgálati közlemény” rovatba tartozó bejegyzések minden adatának listázása

db.bejegyzesek.find({„rovat.rovat\_neve”: „szolgálati közlemény”}, {})

1. 1-es azonosítójú cikkhez tartozó kommentek szövegének listázása

db.kommentek.find({k\_bejegyzesID: 1}, {szoveg: 1, \_id: 0})

1. Azon rovatok és a hozzájuk tartozó legalacsonyabb megtekintési szám listázása, ahol a legalacsonyabb megtekintésű bejegyzést is 100-nál többször tekintették meg

db.bejegyzesek.aggregate([

{$group:

{\_id: "$rovat.rovatID", total: {$min: "$megtekintesek\_szama"}}

},

{$match: {total: {$gt: 100}}}

])

1. 1-es azonosítójú bejegyzéshez legutóbb írt komment összes adata

db.kommentek.find({k\_bejegyzesID: 1}, {}).sort({iras\_datuma: -1}).limit(1)

1. Felhasználók összes adatának listázása ímélcím szerint betűrendben:

db.felhasznalok.find({}, {}).sort({email: 1})

1. Bejegyzések átlagos megtekintési száma rovatok szerint az átlagos megtekintési szám szerint csökkenő sorrendben

db.bejegyzesek.aggregate([

{$group:

{\_id: "$rovat.rovatID", total: {$avg: "$megtekintesek\_szama"}}

}

]) .sort({total:-1})

1. Legkorábban írt első 10 bejegyzés közzétételének dátuma és rovatja

db.bejegyzesek.find({}, {„kozzetetel\_datum”: 1, „rovat.rovat\_neve”: 1, „\_id”: 0}).sort({kozzetetel\_datum: 1}).limit(10)

1. Legutóbbi 5 bejegyzés címe megtekintési száma az „érdekességek” rovatban

db.bejegyzesek.find({„rovat.rovat\_neve”: „érdekességek”}, {cim: 1, megtekintesek\_szama: 1, \_id: 0}).sort(kozzetetel\_datum: -1).limit(5)

1. Azokon bejegyzések közül a legtöbbször megnézett megtekintési száma és teljes címe, melyek címe tartalmazza a ’különleges’ szót.

db.bejegyzesek.find({cim: {$regex: /különleges/}}, {cim: 1, megtekintesek\_szama:1, \_id: 0}).sort(megtekintesek\_szama: -1).limit(1)