

<https://github.com/FAQGURU/FAQGURU/blob/master/topics/en/mongodb.md>

### **Támogatja a MongoDB az idegen kulcs korlátozásait?**

Nem. A MongoDB nem támogatja az ilyen kapcsolatokat. Az adatbázis semmilyen korlátozást nem alkalmaz a rendszerre (azaz: idegen kulcs korlátozásai). Alapvetően a NoSQL adatbázisban Önön múlik, hogy hogyan szervezi az adatokat és azok kapcsolatait, ha vannak ilyenek.

### **Magyarázza el, mi az a MongoDB?**

A MongoDB egy nyílt forráskódú dokumentum adatbázis, amely nagy teljesítményt, magas rendelkezésre állást és automatikus skálázást biztosít. Főbb jellemzői:

- Dokumentumorientált és NoSQL adatbázis.
- Támogatja az aggregálást
- BSON formátumot használ
- Sharding (Helps in Horizontal Scalability)
- Támogatja az Ad Hoc Queries
- Schema Less
- Capped Collection
- Indexelés (A MongoDB bármely mezője indexelhető)
- MongoDB replika készlet (magas rendelkezésre állást biztosít)
- Támogatja a több tárolómotort

### **Hány indexet hoz létre a MongoDB alapértelmezés szerint egy új gyűjteményhez?**

Alapértelmezés szerint a MongoDB minden gyűjteményhez létrehozta a `_id` gyűjteményt.

### **Mi a replikáció a MongoDB-ben?**

A replikáció az adatok szinkronizálása több szerveren keresztül. A replikáció redundanciát biztosít és növeli az adatok elérhetőségét. Ha a különböző adatbázis-kiszolgálókon többször másolja az adatokat, a replikáció megvédi az adatbázist egyetlen szerver elvesztésétől. A replikáció lehetővé teszi a hardverhibák és a szolgáltatás megszakításainak helyreállítását is

### **Mi a „Névtér” a MongoDB-ben?**

A MongoDB BSON (bináris csere- és struktúraobjektum-jelölés) objektumokat tárol a gyűjteményben. A gyűjtemény és az adatbázis nevének összefűzését névtérnek nevezzük

### **Mi a BSON a MongoDB-ben?**

A BSON egy bináris sorosítási formátum, amelyet dokumentumok tárolására és távoli eljárás hívatások kezdeményezésére használnak a MongoDB-ben. A BSON kibővíti a JSON modellt, hogy további adattípusokat, rendezett mezőket biztosítson, és hatékony legyen a kódoláshoz és a dekódoláshoz különböző nyelveken belül.

### **Szüksége van-e a MongoDB-nek sok RAM-ra?**

Nem. A MongoDB kis szabad RAM memóriaterületen futtatható.

### **Mi az a replika készlet?**

Ez egy olyan mongo példányok csoportja, amelyek ugyanazt az adatsort tartják fenn. A replikakészletek redundanciát és magas rendelkezésre állást biztosítanak, és ezek az összes production deployment alapját képezik.

### **Hogyan hajthatom végre az SQL JOIN megfelelőt a MongoDB-ben?**

A Mongo nem relációs adatbázis. A \$lookup operátor lényegében megegyezik egy left outer join-nal:

```
{
  $lookup:
  {
    from: <collection to join>,
    localField: <field from the input documents>,
    foreignField: <field from the documents of the "from" collection>,
    as: <output array field>
  }
}
```

### **Mikor kell használni a MongoDB vagy más dokumentumorientált adatbázis-rendszereket?**

A strukturálatlan adatok tárolására a MongoDB a legalkalmasabb. És ez az adatokat dokumentum formátumba rendezheti. Ezek az adattárak nem érvényesítik az ACID tulajdonságokat és a sémákat. Ez nem nyújt semmilyen tranzakciós képességet. Tehát ez nagy méreteket ölthet, és gyorsabb hozzáférést érhetünk el (olvasás és írás egyaránt). Ha strukturált adatokkal akar dolgozni, az RDBM-et válassza.

### **Mik azok a NoSQL adatbázisok? Melyek a különböző típusú NoSQL adatbázisok?**

A NoSQL adatbázis olyan mechanizmust biztosít az adatok tárolására és visszakeresésére, amelyet a relációs adatbázisokban (például SQL, Oracle stb.) Használt táblázatos relációktól eltérő módon modelleznek.

A NoSQL adatbázis típusai:

- Document Oriented
- Key Value
- Graph
- Column Oriented

### **Mi a covered lekérdezés a MongoDB-ben?**

A covered lekérdezés az, amelyben:

- a lekérdezésben használt mezők a lekérdezésben használt index részét képezik, és
- az eredményekben visszaadott mezők ugyanabban az indexben vannak

### **Miért nagy méretűek a MongoDB adatfájlok?**

A MongoDB az adatfájlokat előre lefoglalja, hogy helyet foglaljon és elkerülje a fájlrendszer széttöredezését a szerver telepítésekor.

### **Hogyan biztosítja a MongoDB az egyidejűséget?**

A MongoDB olvasó-író zárat használ, amelyek lehetővé teszik az olvasók egyidejű hozzáférését egy erőforráshoz, például adatbázishoz vagy gyűjteményhez, de kizárólagos hozzáférést biztosítanak egyetlen írási művelethez.

### **MongoDB kapcsolatok. Mit kell használni - beágyazás vagy hivatkozás?**

Általában,

- a beágyazás jó, ha az entitások között van egy-egy vagy egy-sok kapcsolat, és
- a hivatkozás jó, ha sok-sok kapcsolat van.

Fontolja meg főszabályként azt is, hogy ha sok [gyermekdokumentum] van, vagy ha nagyok, akkor a legjobb lehet egy külön gyűjtés. A kisebb és / vagy kevesebb dokumentum általában alkalmas a beágyazásra.

### **Mi a különbség a MongoDB és a MySQL között?**

A fő különbségek a MongoDB és a MySQL között a következők:

- Különbség van az adatok ábrázolásában a két adatbázisban. A MongoDB-ben az adatok a JSON-dokumentumok gyűjteményében jelennek meg, míg a MySQL-ben az adatok táblákban és sorokban vannak. A JSON-dokumentumok összehasonlíthatják az asszociatív tömböket, ha PHP-t és könyvtárobjektumokat használnak Python használatakor.
- Amikor a lekérdezésről van szó, be kell írnia egy karakterláncot a lekérdezési nyelvbe, amelyet a DB rendszer elemez. A lekérdezés nyelvét Strukturált lekérdezési nyelvnek vagy SQL-nek hívják, ahonnan a MySQL megkapja a nevét. Ez kiszolgáltatja a DB-t, amely hajlamos az SQL-támadásokra. Másrészt a MongoDB lekérdezése objektum-orientált, ami azt jelenti, hogy átad egy MongoDB-nek egy dokumentumot, amely elmagyarázza, hogy mit kérdez. Nincs semmiféle elemzés, amihez egy kis idő kell a felhasználónak, amíg megszokja, ha már használja az SQL-t.
- Az olyan relációs adatbázisok egyik legnagyobb előnye, mint a MySQL, a JOIN művelet. A művelet lehetővé teszi több tábla közötti lekérdezést. Bár a MongoDB nem támogatja a join-t, támogatja a többdimenziós adattípusokat, mint dokumentumok és tömbök.
- A MySQL támogatja az atomi tranzakciókat. Több műveletet végezhet egy tranzakción belül, és visszagörgethet, mintha egyetlen műveletet végezne. A MongoDB-ben nem támogatnak tranzakciókat, és az egyetlen művelet atomi.
- Az egyik legjobb dolog a MongoDB-ben, hogy nem Ön felelős a séma meghatározásáért. Csak annyit kell tennie, hogy bedobja a dokumentumokat. A gyűjteményben lévő 2 dokumentumnak nem kell ugyanabban a mezőben lennie. A MySQL-ben történő tárolás előtt meg kell határoznia a táblázatokat és oszlopokat. A táblázat minden sora ugyanazokkal az oszlopokkal rendelkezik.
- A MongoDB teljesítménye jobb, mint a MySQL és más relációs DB-ké. Ennek oka, hogy a MongoDB feláldozza a JOINS-ot és egyéb dolgokat, és kiváló teljesítményelemző eszközökkel rendelkezik. Vegye figyelembe, hogy még mindig indexelnie kell az adatokat, és a legtöbb alkalmazás adatai nem elegendőek ahhoz, hogy különbséget láthassanak. A MySQL-t kritizálják a gyenge teljesítmény miatt, különösen az ORM alkalmazásban.
- A MySQL egyik előnye a NoSQL-hez, például a MongoDB-hez képest, hogy a MySQL közösség sokkal jobb, mint a NoSQL. Ez főleg azért van, mert a NoSQL viszonylag új, míg a MySQL több éve létezik.
- Az RDBS-k az ACID nevű paradigmán működnek, amely az (Atomicitás, Konzisztencia, Izoláció és Tartósság) rövidítése. Ez nincs a MongoDB adatbázisban.
- A MongoDB rendelkezik Map Reduce funkcióval, amely megkönnyíti a méretezhetőséget. Ez azt jelenti, hogy a MongoDB adatbázis teljes funkcionalitását még akkor is elérheti, ha olcsó hardvert használ.
- Nem kell részletes DB modellt kidolgoznia a MongoDB-vel, mivel az nem relációs. ezáltal megtakarítva a fejlesztési időt és költségeket.