NoSQL adatbáziskezelők

Szathmáry László
Debreceni Egyetem
Informatikai Kar

- sématervezés

(utolsó módosítás: 2018. nov. 29.)

Sématervezés

A MongoDB használata során a nagy kérdés: az adatokat ágyazzuk be, vagy tegyük őket külön kollekcióba?
Relációs adatbázis-kezelő rendszerek esetén a 3NF használata preferált. A MongoDB esetében viszont azt kell átgondolni, hogy az alkalmazásban hogyan akarjuk felhasználni az adatokat. Vagyis az alkalmazáshoz igazítjuk hozzá a sémát (*Application-Driven Schema*).

Milyen adatokat használunk együtt?
Mely adatokat olvassuk sokat?
Mely adatokat módosítjuk sokat?

} ezeket a szempontokat is figyelembe kell venni

Jellemzői:

- beágyazott adatok használata
- nincs join művelet
- atomiak a műveletek
- egy kollekciónak nincs deklarált sémája (habár a dokumentumok általában eléggé hasonlóak)

Ha a sémánk úgy néz ki, mint egy RDBMS séma, akkor a MongoDB-t valószínűleg nem megfelelően használjuk.

Vegyünk pl. egy blogot. Egy RDBMS rendszerben lenne egy *post* táblánk, ami a blogbejegyzéseket tartalmazza. Lehetséges mezők: cím, tartalom, létrehozási dátum. Egy post-hoz tartoznak kulcsszavak is, de a tag-eket már egy másik táblába kellene tenni (*tags*). Egy post-hoz tartoznak kommentek is, melyek egy harmadik táblába kerülnének (*comments*). Vagyis ha meg akarunk jeleníteni egy blogposztot, akkor ehhez legalább 3 táblát kellene felhasználnunk, ill. join művelettel összekapcsolni.

MongoDB esetében egy post dokumentumba be lehet illeszteni a tag-eket egy listába. Ugyanígy a kommentek is beágyazhatók az adott posztba, elvégre egy post-hoz nem szokott túl sok komment tartozni. Itt a post-ok megjelenítéséhez csak egyetlen kollekció dokumentumait kell lekérdezni, s egy dokumentum minden szükséges adatot tartalmaz.

1:1 kapcsolat (One to One)

Példa 1:1 kapcsolatra: *Employee: Resume*. Egy alkalmazottnak egy önéletrajza van, ill. egy önéletrajz egy alkalmazotthoz tartozik.

Két lehetőségünk is van:

1) Különválasztjuk őket két kollekcióba. Az Alkalmazott tartalmaz egy *resume_id* –t, míg az Önéletrajz tartalmaz egy *employee_id* –t.
2) Beágyazás. Az önéletrajzot beágyazzuk az alkalmazott dokumentumába.

Melyiket válasszuk?

- Egy dokumentum mérete a MongoDB-ben max. 16 MB lehet. Ha az alkalmazott és az önéletrajz összmérete ezt meghaladja, akkor nem lehetnek együtt.
- Ha az önéletrajz nagy (pl. képeket is tartalmaz), de ritkán van rá szükség, az alkalmazottat viszont gyakran olvassuk, akkor szét lehet őket választani.
- Ha az önéletrajz nagyon gyakran frissül, akkor is lehet külön.
- 1:1 kapcsolat esetében viszont a legtermészetesebb a beágyazás. Ha az előző speciális esetek nem állnak fenn, akkor nyugodtan be lehet ágyazni az önéletrajzot az alkalmazott dokumentumába.

1:N kapcsolat (One to Many / One to Few)

Példa 1:N kapcsolatra: *City: Person*. Egy városnak sok lakója van, ill. egy személynek csak egy állandó lakhelye van.

A város lakóit be lehetne tenni egy listába. Viszont egy város lakossága nagyon sok lehet, így ez a lista nagyon nagy méretűvé válna. Ebben az esetben a szétválasztás tűnik jó megoldásnak.

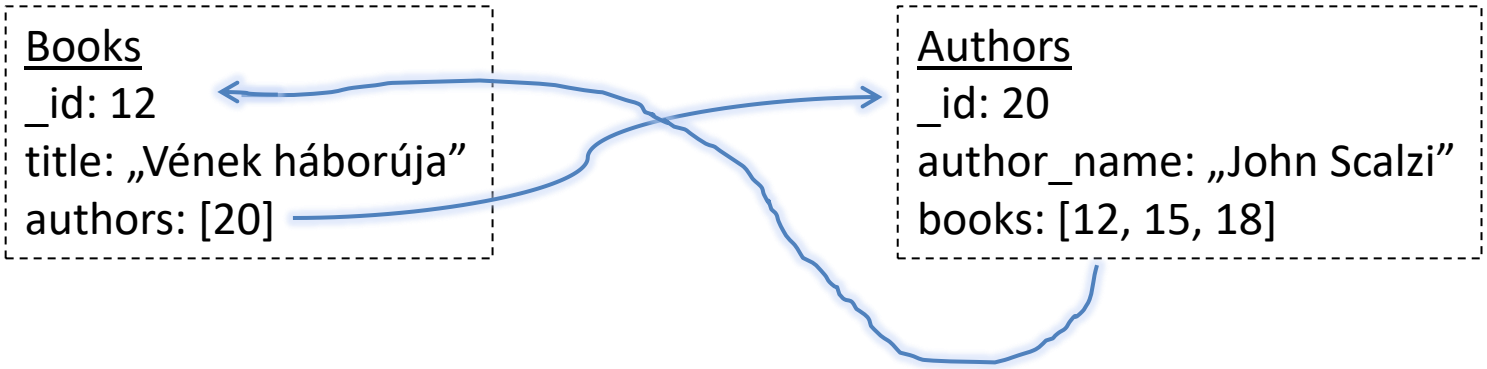
Viszont mi a helyzet akkor, ha nem *One to Many*, hanem csak *One to Few* kapcsolatról van szó, vagyis amikor egy elemhez csak néhány elem tartozik? Gondoljunk vissza a blogposztos példára: egy poszthoz általában nem szokott túl sok komment tartozni. Ebben az esetben a kommentek beágyazhatók a blogposztba egy lista formájában.

Összefoglalva:

One to Many:	két kollekció használata lesz talán a legjobb megoldás
One to Few:	beágyazás

N:M kapcsolat (Many to Many)

Példa N:M kapcsolatra: *Books: Authors*. Egy könyvnek több szerzője is lehet, ill. egy szerzőnek lehet több könyve is.



Egy másik megoldás lehetne a beágyazás, de ez anomáliákhoz vezethet. Pl. az Authors alá tesszük be a könyveket. Ekkor ugyanaz a könyv több szerző alatt is ugyanúgy jelenne meg. Mi van akkor, ha módosítani akarjuk egy könyv adatait? Az összes helyen frissíteni kellene...