

Adatbázistrendek és technológiák: Hagyományos és modern megközelítések

Összefoglalás: Az adatbáziskezelési technológiák fejlődése az elmúlt évtizedekben forradalmasította az adattárolás és -kezelés módszereit. A hagyományos SQL-alapú relációs adatbázisok évtizedek óta dominálnak, de a megváltozott adatigények és a Big Data korában megjelenő új kihívások miatt a NoSQL-adatbázisok egyre nagyobb népszerűségnek örvendenek. A cikk áttekinti az SQL- és NoSQL-adatbázisok közötti főbb különbségeket, különös figyelmet fordítva az ACID- és a CAP-elvekre, amelyek meghatározzák a különböző adatbázisok teljesítményét és megbízhatóságát. Bemutatásra kerülnek a hibrid adatbázisok, amelyek a két technológia előnyeit ötvözik, új megoldásokat kínálnak, alkalmazva a relációs modellek struktúráját és a NoSQL rugalmasságát. A hibrid rendszerek nemcsak költséghatékony megoldásokat jelentenek, hanem lehetőséget adnak a különböző adattípusok integrált kezelésére is. A cikk végül bemutatja az egyes adatbázistípusok leggyakoribb alkalmazási területeit, hangsúlyozva, hogy a megfelelő megoldás kiválasztása az adott üzleti igények és kihívások alapján történik.

Kulcsszavak: SQL, NoSQL, relációs adatbázis, hibrid adatbázis, dokumentum-alapú adatbázis, gráfolapú adatbázis, kulcsérték, ACID, CAP.

Abstract: The development of database management technologies in recent decades has revolutionized the methods of data storage and management. Traditional SQL-based relational databases have dominated for decades, but due to changed data requirements and new challenges in the era of Big Data, NoSQL databases are gaining more and more popularity. The article reviews the main differences between SQL and NoSQL databases, paying particular attention to the ACID and CAP principles that determine the performance and reliability of the different databases. Hybrid databases are presented, which combine the advantages of the two technologies and

* Dunaújvárosi Egyetem, Informatikai Intézet, Szoftverfejlesztési és Alkalmazási Tanszék
E-mail: dudasn@uniduna.hu

offer new solutions, applying the structure of relational models and the flexibility of NoSQL. Hybrid systems are not only cost-effective solutions, but also offer the possibility of integrated management of different data types. Finally, the article presents the most common application areas of each database type, emphasizing that the right solution is selected based on the specific business needs and challenges. **Keywords:** SQL, NoSQL, relational database, hybrid database, document-based database, graph-based database, key-value, ACID, CAP.

Bevezető

Az elmúlt évtizedekben az adatbáziskezelési technológiák jelentős fejlődésen mentek keresztül. Míg az SQL-alapú (Structured Query Language) relációs adatbázisok hosszú ideig uralták az adattárolás világát, az utóbbi években a NoSQL-adatbázisok (Not Only SQL) is egyre népszerűbbé váltak a dinamikus, nagy adatmennyiséggel dolgozó rendszerek igényeinek kielégítésére.

Az SQL- és NoSQL-adatbázisok közötti választás sokáig kompromisszumot jelentett a struktúra, rugalmasság, skálázhatóság és teljesítmény terén.

Azonban a technológiai fejlődés lehetővé tette az olyan hibrid adatbázisok megjelenését, amelyek egyesítik az SQL- és NoSQL-rendszerek előnyeit. Az olyan adatbázisok, mint a CockroachDB és az ArangoDB, mind a relációs, mind a nem-relációs adatok kezelésére alkalmasak, így lehetővé teszik, hogy a felhasználók a legjobbat kapják mindkét technológiából.

A cikk célja, hogy bemutassa a hagyományos SQL- és a NoSQL-alapú adatbázisok közötti főbb különbségeket, illetve ismertesse az új hibrid adatbázisok adta előnyöket és kihívásokat. Valamint rávilágítson arra, hogy milyen helyeken érdemes alkalmazni ezeket a megoldásokat.

Relációs adatbázisok

A relációs adatbázisok (RDBMS) az 1970-es években váltak népszerűvé, miután Edgar Frank „Ted” Codd kidolgozta a relációs modell alapjait. Ted Codd új elgondolása matematikailag megalapozott eszközöket használ, ezáltal pontosabb leírást tesz lehetővé. Megalkotta a relációs algebrát, mely a relációkon végezhető műveleteket definiálja. Fő gondolata az volt, hogy az adatokat kétdimenziós táblázatok (relációk) formájában kell tárolni, így az egymással logikai kapcsolatban lévő táblázatok rendszere alkotja majd magát az adatbázist. Tehát amíg a korai modellek – a hálós és a hierarchikus modell – megvalósításában az egyes rekordok között fizikai kapcsolat van (mutatókon keresztül), addig a táblázatok közötti kapcsolat logikai.

Természetesen akkoriban – ahogy ma is – a szakmai világ kétkedve fogadta az újdonságokat. Az IBM eleinte nem is alkalmazta Codd kutatási eredményeit, nem fogadták el az ötleteit, azaz az adatbáziskezelés relációs elvű megközelítését. Minimum 10 évébe került, hogy meggyőzze a szakmai közvéleményt. Pedig az általa lefektetett alapelvek még a mai napig érvényesek, és szinte egyeduralkodóvá vált a relációs elvű adatbáziskezelés. Nem igazán terjedt el a más szemléletű adatbáziskezelés, kivéve a legutóbbi időben.

NoSQL-adatbázisok

A NoSQL-adatbázisok (Not Only SQL) olyan adatbázis-technológiák, amelyek a relációs adatmodellek alternatívájaként jöttek létre a 2000-es évek elején. A NoSQL-adatbázisok elsősorban nagy méretű, gyorsan változó, és nem strukturált adatok kezelésére specializálódtak.

Ezek az adatbázisok nem kötődnek szigorúan strukturált séma követelményekhez, és nagyobb rugalmasságot kínálnak az adattárolás és kezelés terén is.

A NoSQL-adatbázisok négy fő kategóriába sorolhatók:

- Dokumentum-alapú adatbázisok, ahol az adatok dokumentumok formájában tárolódnak (pl. JSON, BSON formátumban).
- Kulcsérték tárolók, amelyek egyszerű kulcsérték párok alapján tárolják az adatokat.
- Oszlopalapú adatbázisok, amelyek táblázatos formában, de oszlop-orientáltan tárolják az adatokat.
- Gráfalapú adatbázisok, amelyek az adatok közötti kapcsolatokat gráfként kezelik. Az adatokat csomópontok és élek formájában tárolják, ahol a csomópontok entitásokat képviselnek, az élek pedig az ezek közötti kapcsolatokat jelölik.

[1] Buza A. (2015):
Az adatbáziskezelés
alapjai. Dunaújváros:
Rolling.

1. táblázat. NoSQL-alapú adatbázistípusok

Adatbázis típus	Adatbáziskezelő	Alkalmazási területek
Dokumen- tum-alapú	– MongoDB – CouchDB – RavenDB	Webalkalmazások, E-kereskedelem (pl: termékín- formációk), Mobilalkalmazások, Keresőmotorok, Tartalomkezelő rendszerek (CMS)
Kulcsértéktároló	– Redis – Amazon Dyna- moDB – Riak – Firebase Realtime Database	Gyorsítótárak (pl. webes alkalmazások gyorsítása), Munkamenet-kezelés (pl. felhasználói munkamene- tek tárolása), Valós idejű adatok (pl. hőmérsékleti szenzoradatok) feldolgozása, Konfigurációs beállítások kezelése
Oszlopalapú	– Apache Cassand- ra, – HBase – Google Bigtable	Big Data elemzések (pl.: Webes elemzések), Napló- fájlok feldolgozása (pl.: szerverlogok) Valós idejű analitika (pl.: pénzügyi tranzakciók elemzése)
Gráfalapú	– Neo4j	Szociális hálózatok (pl. kapcsolatok és interakciók modellezése), Hálózatelemzés (pl. kiberbiztonsági elemzés), Ajánló rendszerek (pl. termék- vagy tartalomajánlás)

Az 1. táblázatban látható néhány adatbáziskezelő, és a főbb alkalmazási terü-
leteik. Elmondható, hogy NoSQL-adatbázisok rugalmas adatmodellt használnak,
amely lehetővé teszi az adatok strukturálatlan (pl.: különféle bináris fájlok: képek,
hangfájlok, videók, ...) vagy félig strukturált (pl.: JSON formátumú dokumentu-
mok) formában történő tárolását. De akár a strukturált adatok (oszlopalapú ada-
tok) tárolására is alkalmasak. Az egyik legfontosabb jellemzőjük a sémamentes-
ség, amely azt jelenti, hogy az adatok szerkezetének előzetes meghatározása nem
szükséges, és különböző típusú adatok tárolhatók ugyanabban az adatbázisban. [1]
A NoSQL-rendszerek általában elosztott architektúrát használnak, amely le-
hetővé teszi az adatok több szerveren vagy adatközpontban történő tárolását és
elosztását.

Ez növeli a skálázhatóságot, és lehetővé teszi a nagy adatmennyiségekkel (Big Data) történő hatékony munkavégzést. Például a webes alkalmazások, a közösségi média és az online szolgáltatások mind-mind hatalmas adatmennyiséget generálnak, melyet a relációs adatbázisok már nem tudnak hatékonyan kezelni.

[2] Ullman Widom–Jennifer Widom (Szerk.) (1998): *Adatbázis-rendszerek alapvetés*, Budapest: Panem.

Hagyományos SQL- és NoSQL-adatbázisok főbb különbségei

Az SQL-adatbázisok, mint például a MySQL, PostgreSQL, és az Oracle Database, a relációs modellre épülnek, és olyan környezetekben használjuk őket, ahol az adatok szerkezete jól meghatározott, és ahol a tranzakciós integritás kritikus.

A relációs adatbázisokat az SQL (Structured Query Language) nyelven keresztül kezeljük, amely lehetővé teszi az adatok hatékony lekérdezését, manipulálását és karbantartását. Az SQL támogatja a tranzakciók ACID (Atomicity, Consistency, Isolation, Durability) tulajdonságait, amelyek biztosítják a megbízható és konzisztens adatkezelést. Azaz egy vagy több adatbázisműveletet egy tranzakcióba csoportosíthatunk, ez lesz egy olyan munkaegység, amit atomosan (oszthatatlan módon) és más tranzakciótól látszólag elkülönítve (elzártan, rejtve) kell végrehajtani. Miközben minden művelet végrehajtása során rendszerünk egy konzisztens állapotból egy konzisztens állapotba kerül. Valamint az adatbáziskezelő a tartósságot is garantálja, mely során befejezett, sikeres tranzakciók eredményei nem veszhetnek el. Ha egy tranzakció sikertelen, akkor a kezdeti állapotnak kell megmaradnia. [2]

A NoSQL-adatbázisok nem kötődnek a relációs modellhez, és nem igényelnek előre meghatározott sémát. Ez lehetővé teszi számukra, hogy sokkal rugalmasabban kezeljék a strukturálatlan vagy félig strukturált adatokat. Emellett a NoSQL-rendszerek nagy előnye a horizontális skálázhatóság, amely lehetővé teszi, hogy a rendszerek nagy mennyiségű adatot kezeljenek elosztott architektúrákon keresztül.

A NoSQL-adatbázisok leggyakrabban a CAP-tétel (Consistency, Availability, Partition tolerance) alapján működnek, és inkább a rendelkezésre állást és a partíciótoleranciát preferálják, szemben a teljes adatkonzisztenciával, azaz kompromisszumot kötnék az adatok konzisztenciájában. Ez lehetővé teszi az adatok gyors hozzáférését, még akkor is, ha a rendszer egyes részei nem elérhetők.

[3] <https://db.bme.hu/~gajdos/2012adatb2/3.%20eloadas%20NoSQL%20adat-b%E1zisok%20doc.pdf>,” [Online]. Available: <https://db.bme.hu/~gajdos/2012adatb2/3.%20eloadas%20NoSQL%20adatb%E1zisok%20doc.pdf>. [Hozzáférés dátuma: 2024. október 17.]

Maga a CAP-tétel elosztott adatbázisokra vonatkozik, és arra mutat rá, hogy egy elosztott rendszerben a konzisztencia (Consistency), rendelkezésre állás (Availability) és partíciótürés (Partition tolerance) három tulajdonságából egyszerre legfeljebb kettőt lehet biztosítani. [3]

- 1. Konzisztencia:** Az adatok minden lekérdezéskor ugyanazok, tehát a frissítések azonnal láthatók a teljes rendszerben.
- 2. Rendelkezésre állás:** A rendszer minden lekérdezésre választ ad, még akkor is, ha nem tudja garantálni a legfrissebb adatokat.
- 3. Partíciótürés:** A rendszer képes tovább működni akkor is, ha hálózati szakadás történik, és a csomópontok közötti kommunikáció megszakad.

Skálázhatóság

A relációs adatbázisok hagyományosan vertikálisan (függőlegesen) skálázhatók, ami azt jelenti, hogy a teljesítmény növelése érdekében a szerver hardveres kapacitását kell növelni (pl. több memória, gyorsabb CPU). Ez a megközelítés azonban költséges és gyakran nehéz, különösen, ha az adatbázis mérete és a felhasználói terhelése gyorsan növekszik.

A relációs adatbáziskezelő rendszerek jól teljesítenek kisebb adatmennyiségeknél és egyszerű tranzakcióknál, de problémák merülhetnek fel, amikor a rendszer terhelése növekszik. Az ACID-tulajdonságok fenntartása nagy adathalmazok esetén lassíthatja a rendszert, mivel az adatok konzisztenciáját és integritását minden egyes tranzakció során biztosítani kell.

A NoSQL-adatbázisok egyik legnagyobb előnye a horizontális (vízszintes) skálázhatóság, ami azt jelenti, hogy a teljesítmény és a tárhelykapacitás növelése érdekében további szerverek adhatók a rendszerhez. Ez lehetővé teszi, hogy a NoSQL-adatbázisok könnyen kezeljenek hatalmas adatmennyiségeket és nagy terhelést, anélkül, hogy drága hardverfrissítésekre, nagyobb szerverek beszerzésére lenne szükség. Ezáltal lehetővé vált az olcsóbb hardvereken való elosztott adatkezelés.

Hibrid adatbázisok: Az SQL és NoSQL integrációja

A hagyományos SQL-alapú és NoSQL-rendszerek különböző célokat szolgálnak, és gyakran kiegészítik egymást a különböző alkalmazási területeken. A hibrid adatbázisok olyan megoldások, amelyek az SQL- és NoSQL-adatbázis-technológiák egyesítésével működnek, kihasználva és kombinálva a két adatkezelési módszer előnyeit. A hibrid adatbázisok kialakulása a 2010-es évek környékén kezdődött, amikor a vállalatok egyre nagyobb adatkezelési kihívásokkal szembesültek. A relációs adatbázisok már nem voltak elegendők a Big Data korszakában jelentkező, skálázhatósági és teljesítményigények kielégítésére. Azonban az SQL-alapú adatbázisok robusztus tranzakciókezelése és adatkonzisztenciája továbbra is nélkülözhetetlen volt. Ezért alakultak ki olyan adatbázisok, amelyek egyesítik a hagyományos relációs rendszerek és a NoSQL rugalmasságát.

ELŐNYÖK ÉS HÁTRÁNYOK

Ha röviden kellene összefoglalni a hibrid rendszerek előnyeit, akkor ezek a következők lennének:

1. *Rugalmasság:* A hibrid adatbázisok képesek strukturált – relációs adatbázisokban megszokott – és nem strukturált vagy félig strukturált – NoSQL-alapú – adatokat is kezelni. Így egy rendszer képes különböző típusú adatok feldolgozására és tárolására.
2. *Skálázhatóság és teljesítmény:* A NoSQL-technológia biztosítja a hibrid adatbázisok számára a horizontális skálázást (több csomópont hozzáadásával), ami lehetővé teszi a nagy méretű adatforgalom kezelését.
3. *Tranzakciók és konzisztencia:* A hagyományos SQL-alapú adatbázisok által kínált ACID-tulajdonságok a hibrid rendszerekben is elérhetőek, így biztosított a biztonságos és megbízható tranzakciókezelés.
4. *Egyszerű integráció:* A hibrid rendszerek lehetővé teszik, hogy a különböző típusú adatok (strukturált és nem strukturált) egyetlen platformon keresztül legyenek elérhetőek és elemezhetőek, ami egyszerűíti a fejlesztést és az adatkezelést.
5. *Költséghatékonyság:* A hibrid megoldások csökkenthetik az adatkezelési költségeket, mivel egy platformon lehet integrálni a különféle adatokat, ami megszünteti az igényt több rendszer fenntartására.

Viszont akadnak a hibrid rendszereknek hátrányai is:

1. *Komplexitás:* A hibrid adatbázisok kezelése és karbantartása bonyolultabb lehet, mint a tisztán SQL- vagy NoSQL-rendszereké, mivel mindkét technológiát integrálni kell. Ez növeli a fejlesztés és karbantartás költségeit.

2. *Optimalizálás nehézségei*: Az eltérő adattípusok és használati esetek miatt a hibrid rendszerek optimalizálása kihívást jelenthet, különösen akkor, ha az adatbázisnak egyszerre kell kezelnie nagyméretű strukturált és nem strukturált adatokat.
3. *Magasabb belépési küszöb*: A fejlesztőknek és rendszergazdáknak mind SQL-, mind NoSQL-technológiák ismeretére is szükségük van, ami növeli az oktatási és szakértelmi követelményeket.
4. *Teljesítményproblémák*: A hibrid rendszerek esetenként kompromisszumokat kötnek a teljesítmény terén, ha mindkét típusú adatkezelési technológia előnyeit akarják kihasználni.

Alkalmazási területek adatbázis típusonként

A relációs adatbázisok tipikus alkalmazási területei:

- *Pénzügyi rendszerek*: ahol az adatkonzisztencia és a tranzakciók integritása kritikus.
- *Üzleti alkalmazások*: pl. ERP-rendszerek (Enterprise Resource Planning), ahol a tárolt adatok struktúrája szigorúan szabályozott.
- *Kis- és középállalkozások webes alkalmazásai*: kisebb adatbázisok, ahol a relációs modell jól kezelhető.

A NoSQL-adatbázisok leggyakoribb alkalmazási területei:

- *Big Data rendszerek*: ahol hatalmas adatokat kell tárolni és gyorsan elérni (pl. social media platformok, streaming szolgáltatások).
- *IoT (Internet of Things)*: ahol az eszközök folyamatosan hatalmas mennyiségű, különböző típusú adatokat generálnak.
- *E-kereskedelem és felhasználóiinterakció-alapú rendszerek*: ahol a felhasználói adatok nagy mennyiségben és dinamikusan változnak (pl. Amazon, eBay).

A hibrid adatbázisok leggyakoribb alkalmazási területei:

- *E-kereskedelmi platformok*: A termékinformációk relációs adatok, míg az ügyfélértékelések és felhasználói tartalmak félig vagy nem strukturáltak.
- *IoT-rendszerek*: Nagy mennyiségű, változó struktúrájú érzékelőadatok és rögzített, strukturált meta-információk kezelése.
- *Adatvezérelt marketing rendszerek*: Strukturált ügyféladatok és nem strukturált interakciós adatok kombinálása valós idejű analitika és személyre szabott ajánlatok létrehozása céljából.

2. táblázat. Néhány hibrid adatbázis kezelő

Adatbázis kezelő	Típus	Alkalmazási Területek
ArangoDB [4]	Multimodel-adatbázis (dokumentum, kulcsérték, gráf)	Szociális hálózatok, összetett elemzések, webalkalmazások.
Couchbase [5]	Dokumentumalapú és kulcsérték tároló. Képes SQL-alapú lekérdezések futtatására is.	Mobilalkalmazások, e-kereskedelem.
Microsoft Azure Cosmos DB [6]	Globális elosztású multimodel adatbázis (dokumentum, kulcsérték, gráf)	Nagy léptékű webalkalmazások, IoT-alkalmazások, Big Data-elemzések.
OrientDB [7]	Multimodel-adatbázis (dokumentum, gráf, reláció)	Hálózatelemzés, szociális hálózatok, összetett adatkapcsolatok modellezése.

Összegzés

A relációs, NoSQL- és a hibrid adatbázis kezelők mind jelentős szerepet játszanak az adatok kezelésében, de különböző igényekhez és alkalmazási területekhez illeszkednek. Relációs-adatbázisok klaszikus, jól bevált megoldások, amelyek szigorú adatmodellezést és tranzakcióbiztonságot kínálnak az ACID-tulajdonságok révén.

Ideálisak olyan alkalmazásokhoz, ahol a strukturált adatok és a magas fokú adatkonzisztencia elengedhetetlenek, mint például pénzügyi rendszerek vagy vállalati erőforrás-tervezési (ERP) megoldások.

A NoSQL-adatbázisok viszont rugalmasabb megközelítést kínálnak, lehetővé téve a strukturált, félig strukturált és nem strukturált adatok hatékony kezelését. Az horizontálisan skálázható NoSQL-rendszerek kiválóan alkalmasak a Big Data-kihívások kezelésére, mint például a közösségi média, valós idejű analitika és IoT-alkalmazások, ahol a sebesség és a rugalmasság kulcsszerepet játszik.

A hibrid adatbázisok az SQL és NoSQL előnyeit ötvözik, lehetővé téve a felhasználók számára, hogy egyszerre kezeljenek különböző típusú adatokat. Ezek a rendszerek ideálisak a modern, összetett alkalmazások számára, amelyek változatos adatkezelési igényekkel rendelkeznek. Rugalmas adatmodellezésük és a tranzakciók biztonságának megőrzése segít a különböző adatok hatékony integrálásában.

Mindezek a megoldások különböző kihívásokat és előnyöket kínálnak, és a választásuk mindig az adott alkalmazás igényeitől, a tervezett adatmennyiségtől, a szükséges skálázhatóságtól és a teljesítmény elvárásoktól függ. A jövőben valószínű, hogy a különböző adatbáziskezelő rendszerek további fejlődésen mennek keresztül, reagálva a folyamatosan változó adatmennyiségi és kezelési igényekre. Az optimális megoldás kiválasztása érdekében a fejlesztőknek alaposan meg kell érteniük a rendelkezésre álló lehetőségeket, és figyelembe kell venniük a specifikus követelményeket és környezeti tényezőket.

