

Nyílt forráskódú adatbázis-kezelők

Szabad szoftver keretrendszer

Készítette a Közigazgatási és Igazságügyi minisztérium E-közigazgatási
Szabad Szoftver Kompetencia Központja
Budapest, 2013



A projekt az Európai Unió támogatásával, az Európai Regionális Fejlesztési Alap társfinanszírozásával valósul meg.

Kódszám: EKOP–1.2.15

Ez a Mű a Creative Commons Nevezd meg! – Így add tovább! 3.0 Unported
Licenc feltételeinek megfelelően szabadon felhasználható.

A dokumentum legfrissebb változata letölthető a honlapunkról:

<http://szabadszoftver.kormany.hu/szabad-szoftver-keretrendszer/>

Tartalomjegyzék

Történeti áttekintés.....	2
MySQL.....	3
Telepítése és beállítása.....	4
A MySQL mentése.....	6
Referenciák.....	7
MariaDB.....	7
Referenciák.....	8
PostgreSQL.....	8
Referenciák.....	9
SQLite.....	9
Referenciák.....	10
Hogyan válasszunk?.....	10

Nyílt forráskódú adatbázis-kezelők

Az általános definíció szerint az adatbázis-kezelő szoftverek olyan speciális alkalmazások, amelyek adatok tárolására és elemzésére használhatók. Az adatbázis-kezelő rendszerek gyakran hálózatos környezetben többfelhasználós, párhuzamos hozzáférést biztosítanak adatbázisokhoz. Angol megnevezésük: DBMS – Database Management System.

Történeti áttekintés

Már a 20. század legelején mutatkozott igény a népesség nyilvántartási adatok kezelése miatt adatbázisba szervezésre. Valójában azonban csak a 1960-as években alakult ki az a struktúra, amelynek nyomait a ma működő adatbázis-kezelők működési modelljeiben is megtaláljuk. Elsősorban bérszámfejtésre és adatsorok, statisztikák elkészítésére használták a lyukkártyás technológiát. A ma használt legnépszerűbb adatbázis-modell működési logikáját 1970-ben Edgar F. Codd¹ publikálta az IBM munkatársaként. Az első modern adatbázis-kezelőt is az IBM hozta létre System-R néven (ma DB/2 néven ismert). Az első nagyszabású adatbázis-kezelő mégis az Oracle² nevéhez fűződik, amelyet ma is széles körben használnak. Fejlesztésének motivációi között többek között a CIA igényei is szerepeltek. A mai SQL alapjait azonban a System-R fejlesztői fektették le, amelyet SEQUEL nyelvnek neveztek el. Ma már sok kereskedelmi termék és számos kiváló nyílt forráskódú rendszer is a felhasználók rendelkezésére áll. A legelterjedtebb szabad szoftverek:

- MariaDB³
- MySQL⁴
- PostgreSQL⁵
- SQLite⁶

¹ http://hu.wikipedia.org/wiki/Edgar_F._Codd

² http://hu.wikipedia.org/wiki/Oracle_Corporation

³ <https://mariadb.org/>

⁴ <http://www.mysql.com/about/legal/licensing/index.html>

⁵ <http://www.postgresql.org/>

⁶ <http://www.sqlite.org/copyright.html>

Rendelkezésre áll még több olyan adatbázis-kezelő is, amely akár bizonyos felhasználószámig, vagy bizonyos nem kereskedelmi esetekben ingyenesen használhatóak, de mivel ezek nem szabad szoftverek, és vagy funkció-, vagy felhasználószám-korlátozással rendelkeznek, nem szerepelnek ebben a felsorolásban. Egy napjainkban üzemeltetett átlagos LAMP (mely a Linux, Apache, MariaDB és PHP stack közös rövidítése) szerveren biztosan telepítve van egy adatbázis-szerver, hiszen nem csak a modern CMS, CRM és egyéb webes technológiák követelik meg a DB meglétét, hanem nagy felhasználó szám esetén az FTP, MAIL és egyéb felhasználói profilokat, jelszavakat is célszerű adatbázisban tartani.

Speciális igények, például nagy számú hálózati hozzáférés ideiglenes adatainak tárolása, vagy rendkívül nagy mennyiségű adat (big data) gyors, párhuzamos feldolgozása speciális adattároló- és kezelő alkalmazásokat hozott létre, amelyek a hagyományos adatbázis-kezelők kiegészítésére, illetve akár kiváltására is használhatók. Az első esetben a meglévő adatbázis-kezelők gyorsítótárazó képességeinek teljesítményét is meghaladva tömörebb, csak memóriában történő tárolást, és ezzel több felhasználó egyidejű kiszolgálását teszik lehetővé olyan gyors kulcs-érték kezelők, mint a népszerű nyílt forráskódú memcached vagy Redis. A második speciális esetben a noSQL adatbázis-kezelők és a kapcsolódó map-reduce feldolgozó-rendszerek, mint például a szintén nyílt forráskódú MongoDB és a hozzá kapcsolható Hadoop jöhetnek számításba.

MySQL

A MySQL⁷ jelenleg az egyik leginkább ismert és elterjedt adatbázis-motor, amelynek közösségi változata (MySQL Community Edition) szabad szoftver. A jogtulajdonos Oracle nagyvállalati részre elérhetővé teszi zárt forráskóddal, klasszikus licenc vásárlási konstrukcióban a közösségi változat kibővített funkcionális variánsát. Feltehetőleg ez a fő oka, hogy a közösség a nyílt licenc alatt hozzáférhető változat alapjain elkezdte fejleszteni a MariaDB nevű adatbázis kezelőt, mely a jelenlegi trendek szerint tempósan elkezdte átvenni a MySQL helyét a szabad operációs rendszer terjesztésekben és számos felhasználónál. De erről majd a következő fejezetben.

A MySQL fejlesztését a svéd MySQL AB kezdte meg kettős licencelés alatt. Lehetősége volt a felhasználóknak GPL szabad szoftver licenc⁸ alatti üzleti felhasználásra vagy zárt (tulajdonosi) licenccel egyéb üzleti modell megvalósítására is. A MySQL-t 2008-ban felvásárolta a Sun, amelyet két évvel később az Oracle vásárolt fel. Így mára a MySQL fejlesztését az Oracle végzi, de az előre nyílt forráskódú maradt. Ez egyrészt kedvező, mivel az Oracle egyike az adatbázis-fejlesztő cégek legnagyobbikának, másrészt pedig sok emberben kételyeket ébreszt, hogy az Oracle-nek szüksége van-e egy nyílt forráskódú, konkurens termékre. Mivel azonban a MySQL továbbra is GPL alatt elérhető, így a lehetőség adott egy esetleges kódbezárás utáni a rendszer fejlesztésének szabad folytatására. Jelenleg a legtöbb platform alatt futtatható (Linux és Unix klónok, illetve Windows és Mac OS X alatt is). Üzemeltetési szempontból a kezelése egyszerűen elsajátítható és az elérhető dokumentációk tekintetében is igen versenyképes. Főbb verzióit tekintve hatalmas fejlődés ment keresztül az elmúlt években és egyre inkább kezdenek nem kis lekérdező motorként, hanem igen komoly feladatok ellátására alkalmas DB-ként gondolni rá. Adminisztrálását számos eszköz könnyíti, többek között van általános célra használható parancssoros felülete, ahol gyakorlatilag majdnem mindent könnyen el lehet érni (az adminisztrátorok inkább ezt szeretik és használják, mivel SSH-n keresztül távolról is könnyen és biztonságosan kezelhető). A szabad szoftver MySQL GUI Tools a funkciók grafikus elérését teszi lehetővé, de ennek fejlesztését az Oracle leállította és a részben szabad MySQL Workbench használatát ajánlják helyette, mindkét eszköz megtalálható a

⁷ <http://www.mysql.com/about/legal/licensing/index.html>

⁸ GPL v2, szerkesztési (linking) kivétellel, amely még nagyobb teret enged az üzleti célú felhasználásnak.

fontosabb Linux terjesztésekben, de elérhetők a MySQL honlapjáról is. Sajnos a legtöbb esetben a phpMyAdmin⁹ felületen keresztül adminisztrálják, amely PHP-alapú, nyílt forrású eszköz. Jellemzően a rendszer-üzemeltetők (az adatbázis üzemeltetőkkel szemben) nem kedvelik, mivel a biztonsági hiányosságai és azok befoltozásának gyakorisága és gyorsasága jellemzően problémás, folyamatosan veszélyezteti az őt futtató szerver és az adatbázis biztonságát egyaránt. Ennek ellenére, megfelelő környezetben használva (SSL mögött, Htaccess és egyéb szabályok a hozzáférés korlátozására, nem kitalálható URL stb.) ajánlható és remek eszköz a MySQL DB-adminok számára. Az adatbázisok mentése viszonylag egyszerű és a gyári segédprogramokkal jól megoldható, a legtöbb mentő program képes kezelni natívan, illetve bizonyos esetben lehetőség van MySQL Hotcopy-t csinálni az éppen futó adatbázisról. Az alaptelepítés minden esetben egy általános `my.conf` állománnyal érkezik, amely finomhangolása feltétlen szükséges az adott rendszer várható terhelése és a hardverparaméterek függvényében. A MySQL számos olyan képességgel rendelkezik, ami kiemeli a mezőnyből.¹⁰ A MySQL-hez nagyvállalatoknak szánt megoldások is elérhetők, amely segítségével magas rendelkezésre állású rendszereket lehet építeni. Ilyen megoldás például a MySQL Cluster. Működése pont olyan, mint egy teljes klaszteré, azonban csak az adatbázis-motor van elosztva. Kisvállalati környezetben, illetőleg ahol nincs szükség a teljes funkcionalitás (pl. LAMP) klaszterbe szervezésére, jó terheléselosztó módszer lehet, illetve az adatok redundáns tárolását lehet vele megoldani. A funkció beállítása viszonylag egyszerű, és igazi prémium funkcionalitást biztosít. Nagyvállalati környezetben a MySQL mellett szóló érv, hogy mivel hazánkban is igen elterjedt, sok informatikai vállalkozás képes támogatást nyújtani hozzá.

Telepítése és beállítása

Mivel a MySQL minden ismertebb platformra fordítható és természetesen a legnépszerűbb terjesztések csomagolt részét képezi, ezért igen egyszerű:

```
apt-get install mysql-server mysql-client
```

a függőségek kezelése miatt, az apt-get inentől felrakja a szükséges csomagokat. Természetesen, ha PHP alól is el akarjuk majd érni, akkor szükséges a php5-mysql csomag telepítése is, de erről bővebben olvashatunk a WEB fejezetben.

A telepítő tehát letölti a függőségeket és beállítja egy alap működésre a MySQL-t, kérni fogja az installáció végén a MySQL root jelszavát. Ezt érdemes legalább 10 karakteres véletlen jelszónak beállítani, amelyet majd a MySQL backup során is fel kell hogy használjunk, így kénytelenek leszünk a MySQL gépen tárolni, ezért olyan állományba és könyvtárba rakjuk, ahol csak a root felhasználó olvashatja (igaz nem ez lesz az Achilles-sarka a MySQL rendszerünknek :)

Ezt követően a MySQL készen áll a működésre, azaz csak teljesen alap funkcionalitással. Ha szeretnénk alap esetben egy kicsit biztonságosabbá tenni, akkor erre egy remek lehetőséget kínál a `mysql_secure_installation` script. Szintén rootként futtatva, interaktívan tudunk pár igen hasznos opciót tiltani vagy engedni: tilthatjuk az anonymous felhasználó hozzáférését, a távoli root hozzáférést, törölhetjük vagy korlátozhatjuk a teszt adatbázishoz való hozzáférést. Ezek azért fontos dolgok, mivel többnyire egy LAMP környezet gyenge pontja (a PHP-n kívül) általánosságban az SQL szokott lenni.

A további konfigurálást a `/etc/mysql/my.cnf` állományban tudjuk megtenni. A fontosabb opciók:

```
[mysqld]
user      = mysql
pid-file  = /var/run/mysqld/mysqld.pid
socket    = /var/run/mysqld/mysqld.sock
```

⁹ <http://www.phpmyadmin.net/>

¹⁰ <http://hu.wikipedia.org/wiki/MySQL>

```
port      = 3306
basedir   = /usr
datadir   = /var/lib/mysql
tmpdir    = /tmp
lc_messages_dir = /usr/share/mysql
bind-address = 127.0.0.1
```

A fenti alap beállítások megváltoztatására csak indokolt esetben van szükség. Ilyen lehet, ha a nem a standard 3306-os porton akarjuk hogy válaszoljon, vagy ha DATADIR-t át akarjuk rakni, pl. lemez I/O vagy telítettség miatt. Fontos megjegyezni, hogy a MySQL nagyon rosszul tolerálja, ha normál üzem közben elfogy alóla a lemezterület. Ilyenkor a táblák és a DB is sérülhetnek, ezért nagyon fontos jól méretezi a DATADIR területet, valamint érdemes folyamatosan egyedi scriptek, vagy nagios, vagy munin segítségével figyelemmel kísérni, nehogy kifusson a MySQL a neki fenntartott lemezterületből. A bind-address segítségével pedig beállíthatjuk, hogy a ne csak a localhoston hallgasson DB motor, hanem pl. a külső lábon is. Ehhez ugyan aztán a DB-n belül is engedélyezni kell. Mivel ez egy jellemző dolog ezért a többi konfiguráció előtt érdemes tárgyalni. Tehát ha egy távoli host részére szeretnénk engedni, hogy egy adott db vagy akár egy adott db táblájához hozzáférjen, akkor a következőket kell tennünk: a [bind-address](#)-t átírjuk a megfelelő IP-címre, pl.

```
bind-address = 192.168.1.1
```

majd újraindítjuk a MySQL-t: [service mysql restart](#)

Ezek után belépünk a `mysql -u root -p` parancs segítségével. Ha még nem hoztunk létre adatbázist, akkor a legegyszerűbb, ha rögtön úgy hozzuk létre, hogy ahhoz a megfelelő IP-ről hozzá is tudjanak férni:

```
mysql> CREATE DATABASE webmotor;
mysql> GRANT ALL ON webmotor.* TO webuser@'192.168.1.2' IDENTIFIED BY 'jelszo';
mysql> flush privileges;
mysql> quit;
```

jelen esetben tehát a webmotor db-hez a webuser felhasználónak engedjük a 192.168.1.2-ről a hozzáférést. Ezt később bármikor módosíthatjuk a GRANT parancs segítségével. A távoli SQL hozzáférések alap esetben titkosítatlan csatornán utaznak, azaz tökéletesen lehallgathatóak. Éppen ezért, ilyen távoli hozzáférést, csak VPN-en belül, vagy belső hálózaton belül, de mégis leginkább akár ssh port forward, vagy stunnel¹¹ segítségével csináljunk csak! A távoli management fejezetben mindkettő opció elolvasható.

A továbbiakban a MySQL finom hangolását végezhetjük el:

```
max_connections = 100
```

alap esetben 100 db párhuzamos csatlakozást enged a szerver felé, természetesen ez akár lehet párhuzamosan 100 db PHP kód, ami ír vagy olvas az SQL-ben. Éppen ezért, nagyon fontos, hogy ezt a paramétert szükség esetén emeljük fel a megfelelő számra. Nagyon gyakori hiba, hogy egy jól méretezett vason a MySQL hatalmas terhelést ad, egyszerűen azért, mert elfogynak a nyitható kapcsolatok és berakja őket egy várakozó sorba, ami az Apache+PHP oldalon is fennakadást okoz. Így előbb-utóbb megeszi az erőforrásokat, a memóriát és a CPU-t is. Egy átlagos és jól méretezett LAMP környezetben, átlagos vasteljesítmény mellett az ideális szám 900.

```
connect_timeout = 5
wait_timeout    = 600
max_allowed_packet = 16M
```

¹¹ <http://shellboy.com/secure-remote-mysql-connection-using-stunnel.html>

```
thread_cache_size = 128
sort_buffer_size = 4M
bulk_insert_buffer_size = 16M
tmp_table_size = 32M
max_heap_table_size = 32M
```

Jellemzően a timeout és cache értékek beállításai megfelelőek szoktak lenni egy teljesen alap rendszernek. Megint csak egy megfelelően megválasztott timeout vagy cache mérettel nagy gyorsulást lehet elérni, de ugyan akkor egy túl rövidre vagy hosszúra választott timeout, szépen le tudja ültetni a rendszert, akár csak a túl kicsi vagy nagy cache. Ezeket szépen fokozatosan érdemes igazítani az aktuális rendszerhez, a programozókkal együttműködve. Érdemes folyamatosan monitorozni egy-egy átállított paraméter hatását pl. a munin segítségével.

```
slow_query_log_file = /var/log/mysql/mysql-slow.log
long_query_time = 10
```

a slow query log beállítása egy befektetés a jövőbe, hiszen utólag itt lehet majd visszakeresni azokat a lekérdezéseket, amelyek valamiért akár percekig, vagy órákig futottak. Itt megint nagy segítséget jelenthet a Munin Slow-Query mrtg diagramja, ahol látható hogy volt vagy nem volt ilyen. Utána pedig oda lehet adni a programozónak a logot elemzésre. Ezen kívül természetesen a MySQL teljes és részleges log rögzítésre is képes, amely szintén megtalálható az alap konfigurációban, viszont ezzel nagyon csínján bányjunk, ugyan is nagyon hamar a telítődhet a teljes log partíció, ha nem csak adott periódusra kapcsoljuk be a teljes logot.

A MySQL mentése

Értelemszerűen az installálás utolsó lépésjében érdemes egy alap mentést azonnal beállítani, amit aztán lehet bővíteni, vagy mentési rendszerbe kötni. A MySQL beépített mentési mechanizmusát felhasználva a következőképpen lehet a legegyszerűbben lementeni a DB-t:

```
/usr/bin/mysqldump -prootjelszo webdb | /bin/gzip > /backup/mysql-webdb-$1.tgz
/usr/bin/mysqldump -prootjelszo information_schema | /bin/gzip >
/backup/mysql-information_schema-$1.tgz
/usr/bin/mysqldump --all-databases -prootjelszo | /bin/gzip > /backup/mysql-ALLDATABASE-$1.tgz
```

Mint az látható, az első két sor adott adatbázisokat ment, amelyeket a jelszó után nevesítettünk, a harmadik sor pedig az összes adatbázist menti egyben. A mentő sorok gzip tömörítést is alkalmaznak, mivel a DB állományok akár 80-90%-os méret csökkenéssel is tárolhatóak, hiszen azok jellemzően sima text állományok. A fenti példában azért van külön és egyben is mentve az összes használt DB, mivel egy esetleges részleges adat veszteség vagy korrupció esetén, nagyon nagy segítség, ha nem a teljes mentésből kell visszaállítani 1 táblát vagy 1 DB-t. Viszont fontos, hogy egy teljes dump is rendelkezésre álljon. Fontos továbbá az information_schema DB mentési is, mint az látható, hiszen itt tárolódnak a DB közötti hozzáférések és egyéb fontos információk. A teljes mentés természetesen ezt tartalmazza. A fenti sorokat érdemes tehát egy héjprogramba belemásolni és a megfelelő időben, amely jellemzően hajnalban üresjáratú időben van, futtatni. Akár napi, heti, havi rotációs tárolással.

Egy alternatív mentési lehetőség, amikor a MysqlHotCopy¹² Perl script gyűjteményt használjuk. Jellemzően ezt a módszert nem az előzőekben leírt mysqldump helyett, hanem annak kiegészítésére, akár napközbeni vagy tesztelés előtti gyors DB-mentésre használhatjuk bizonyos megkötésekkel. Pl. csak MyISAM DB és Archive db típusok esetében használható.

¹² <http://dev.mysql.com/doc/refman/5.0/en/mysqlhotcopy.html>

Természetesen a komplexebb mentő eszközök kezelik a MySQL-t, de erről a Backup fejezetben olvashatunk.

Referenciák

Néhány fontosabb referencia, ahol a MySQL-t meglelégedéssel használják¹³:

- WhiteHouse.Gov
- NASA
- Department of Homeland Security
- Swedish National Police
- Ministère de la Justice (France)
- Bank of Finland
- Swedish National Police
- US Navy

MariaDB

A MariaDB¹⁴ a MySQL forráskódjából kiindult fejlesztés, mely jelenleg a MySQL-lel teljes mértékben kompatibilis, közösség által fejlesztett és támogatott adatbázis szerver. A rendszer egyben kiválóan szemlélteti a szabad szoftver licencelésben rejlő hatalmas lehetőséget is. A MariaDB-t az Oracle üzleti célokat szem előtt tartó lépései miatt aggódó közösség hozta létre a GPL licenc előírásait betartva, a MySQL forráskódjának továbbfejlesztésével. A legtöbb platformra elérhető binárisal vagy forráskóddal rendelkeznek. Több klienssel használható, és rendelkezik a saját fejlesztésű MariaDB Client Library for Java Applicationsszel is. 2012 decemberében egy alapítványt is létrehozta, amelynek célja a fejlesztés felügyelete, valamint annak háttérének megteremtése. Jelenleg azonban nem minden Linux terjesztésnek része (bár ez 2013 elején gyorsan változik, szinte minden fontosabb terjesztés átállt rá MySQL-ről), viszont minden lényeges terjesztéshez elérhető, binárisan formában lévő programcsomagok szintjén (pl. Debian és Ubuntu rendszerekhez [apt.source](http://apt.source.intel.com) szinten is). Minden jel arra mutat, hogy **a MariaDB a jövő szabad szoftveres adatbázis-kezelője**, hiszen az Oracle MySQL fejlesztése nagyon könnyen vehet nem várt fordulatot, így a felhasználók jelentős része tart a MySQL-től.

A MariaDB installálása majdnem pontosan megegyezik a MySQL beállításával, csupán jelenleg nem alap része a terjesztések szoftver kínálatának. Ezért első közelítésben szükséges felvenni a MariaDB csomag tárolóit¹⁵ a rendszerbe:

```
sudo apt-get install python-software-properties
sudo apt-key adv --recv-keys --keyserver keyserver.ubuntu.com 0xc9cb082a1bb943db
sudo add-apt-repository 'deb http://ftp.heanet.ie/mirrors/mariadb/repo/5.5/ubuntu precise main'
```

Innentől pedig már csak telepíteni kell:

```
sudo apt-get update
sudo apt-get install mariadb-server
```

¹³ <http://www.mysql.com/customers/>

¹⁴ <https://mariadb.org/>

¹⁵ <https://downloads.mariadb.org/mariadb/repositories/>

A továbbiakban pontosan ugyanolyan parancsok és lehetőségek segítségével állíthatjuk be, mint a MySQL esetében.

Referenciák

Számos ismert sikertörténetről olvashatunk a MariaDB honlapján¹⁶, ezek közül néhány nagyobb, érdekesebb:

- A Wikipédia angol és német kiadásai valamint a WikiData 2013 április 22-én befejezte a MariaDB-re való átállást¹⁷, melyet azóta is elégedetten használnak. A rendszer terhelésére jellemző, hogy az intenzív gyorstárazás ellenére (melyet memcached és Redis segítségével valósítottak meg) az angol Wikipédia csúcsterhelése 50k lekérdezés másodpercenként.
- A 150 millió felhasználós Nimbuzz háttérrendszerén átállt MariaDB-re¹⁸.
- Az OLX, mely napi 40 millió lapot szolgál ki, szintén MariaDB-t használ¹⁹.

PostgreSQL

A PostgreSQL²⁰ fejlesztését az 1970-as években kezdték meg a kaliforniai Berkeley egyetemen Ingres néven. A 80-as évek közepétől több különböző kereskedelmi termék (mint a Sybase vagy a Microsoft SQL Server), és akkor még Postgres (Post Ingres) néven ingyenes verzióként élt tovább a projekt. 1996-ban kapta a manapság használt PostgreSQL nevet (a 6.0-ás verziótól). Alapvetően egy olyan adatbázis-kezelő rendszer, amely szabványos SQL-felületen vezérelhető, azonban a tudása jóval komplexebb a függvényrendszernek köszönhetően. A függvények megvalósítása több nyelven is lehetséges (pl. Java, Perl, Python, Ruby, Tcl, C/C++, PL/pgSQL). További előnyei a triggerrek²¹ definiálása, a tranzakciókezelés, a komplex adattípusok használhatósága, a sémák és a replikáció. Természetesen rendelkezik a magyar nyelv teljes körű támogatásával DB szinten. Komplexitásának és tudásának köszönhetően gyakran választják nagy teherbírású és magas rendelkezésre állást igénylő feladatokhoz. A PostgreSQL is szabad szoftver, amelyet a MIT-stílusú (a módosított változatok zárt licencelését is megengedő) PostgreSQL²² licenc alatt tesznek közzé.

Jelenleg a következő, igencsak meggyőző korlátokkal rendelkezik:

- maximális adatbázisméret korlátlan
- maximális táblaméret 32 TB
- maximális sorméret 1,6 TB
- maximális mezőméret 1 GB
- sorok maximális száma táblánként korlátlan
- oszlopok maximális száma táblánként 250–1600, oszloptípustól függően
- indexek maximális száma táblánként korlátlan

Rendelkezik a természetesnek számító parancssoros kezelést lehetővé tevő eszközzel, és számtalan támogatott és igen magas szinten elkészített GUI segédprogrammal²³ is, amelyek megkönnyítik

¹⁶ <https://kb.askmonty.org/en/mariadb-case-studies/>

¹⁷ <http://blog.wikimedia.org/2013/04/22/wikipedia-adopts-mariadb/>

¹⁸ <https://kb.askmonty.org/en/nimbuzz-uses-mariadb/>

¹⁹ https://kb.askmonty.org/en/olx_serves_40_million_page_views_daily_MariaDB/

²⁰ <http://www.postgresql.org/>

²¹ http://hu.wikipedia.org/wiki/Trigger_%28adatb%C3%A1zisok%29

²² <http://opensource.org/licenses/postgresql>

²³ http://wiki.postgresql.org/wiki/Community_Guide_to_PostgreSQL_GUI_Tools

az adatbázis-adminisztrátorok munkáját. Az egyik ilyen igen népszerű segédprogram a phpPgAdmin²⁴, amely akárcsak a MySQL esetében a phpMyAdmin, egy PHP alapú GUI. Érdemes megnézni a tudásmátrixot²⁵, amely jól szemlélteti, hogy az elmúlt években hatalmas fejlődésen ment keresztül, amelyet folyamatosan honorálnak is a különféle díjakkal²⁶. A kételkedőknek pedig érdemes átfutni a nagyobb felhasználók²⁷ listáját tartalmazó weboldalt, ahol meggyőződhetünk arról, hogy mekkora cégek döntöttek a PostgreSQL mellett. Természetesen rendelkezik az összes ilyenkor elvárható support és információs felülettel, Wiki-oldallal, célzott fórumrendszerrel. Inkább érdekesség, mint fontos információ, de a PostgreSQL team kizárólag Debian GNU/Linuxot használ a saját²⁸ projektjén belül. Számos európai cég vállal 7×24-es támogatást a PostgreSQL-re, itthon is több közepes méretű cég vállal Enterprise supportot rá. Érdemes megemlíteni a számos gyári kiterjesztés közül a PostGIS²⁹ kiterjesztést, amely kifejezetten földrajzi adatok kezeléséhez nyújt hathatós segítséget. Gyakorlati tapasztalat, hogy a PostgreSQL a komplexebb Oracle-szintű DB kiváltására kiválóan alkalmas.

Referenciák

A rendszer honlapján számos komoly referencia van feltüntetve³⁰, ezek közül néhány fontosabb:

- U.S. Department of Labor
- U.S. General Services Administration
- U.S. State Department
- National Physical Laboratory of India
- United Nations Industrial Development Organisation
- City of Garden Grove, California
- BASF, Agricultural Product Division
- IMDB.com
- Creative Commons
- Greenpeace
- Fujitsu
- Cisco

SQLite

Az SQLite³¹ egy igen széles körben használt (a többi felsorolt DB-hez képest apró) adatbázis-kezelő függvénykönyvtár, mely egyszerű kezelését egy parancssori adminisztrációs eszköz és egy Firefox kiegészítő (SQLite Manager) segíti. Egyéni felhasználásra szánt (bár helyben párhuzamosan több folyamat és szál is használhatja), nem biztosít hálózati hozzáférést az egyetlen állományként tárolt adatbázisaihoz. Jellemzően kliens oldali szoftverekbe építik a rendszer adatainak és

²⁴ <http://sourceforge.net/projects/phpPgAdmin/>

²⁵ <http://www.postgresql.org/about/featurematrix/>

²⁶ <http://www.postgresql.org/about/awards/>

²⁷ <http://www.postgresql.org/about/users/>

²⁸ <http://www.postgresql.org/about/servers/>

²⁹ <http://www.postgresql.org/about/news/1387/>

³⁰ <http://www.postgresql.org/about/users/>

³¹ <http://www.sqlite.org/>

a gyorsító tárnak egyszerű kezelésére. Jó példa erre a Firefox³² böngésző és Thunderbird³³ levelező-kliens, ahol az elsődleges adatok tárolására használják. Számos más cég építi bele³⁴ kereskedelmi vagy szabad szoftveres termékébe. Felhasználási feltételei nincsenek³⁵, hivatalosan közkinccs (public domain)³⁶, amely szabad kezet ad a felhasználás tekintetében. Tudása folyamatosan fejlődik,³⁷ és méreteihez képest igen figyelemre méltó. Az adatbázis fájlok kényelmes elérésére használható még az Sqliteman³⁸, az SQLite Studio³⁹ és a (PHP-ban íródott) SQLiteManager⁴⁰.

Számos további adatbázis-kezelő megoldás létezik szabad szoftverként, azonban egy tipikus vállalati környezetben (ide nem értve az olyan felhasználási területet, ahol párhuzamosan több ezer, százezer, vagy millió felhasználó férhet hozzá egy időben ugyanahhoz az adatbázishoz, mint pl. a Google vagy a Facebook esetében) jellemzően a több főkiszolgálós replikációt⁴¹ tudó adatbázis-motorokat használjuk.

Referenciák

Meglepő, hogy az SQLite teljesen észrevétlenül mennyire részévé vált a mindennapjainknak. A projekt honlapján található fontosabb referenciák közül⁴² kiemelünk néhány fontosabbat:

- Mozilla Firefox, Thunderbird, FirefoxOS - ezek az óriási felhasználói bázissal rendelkező programok és rendszerek a belső adataik tárolására használják az SQLite-ot.
- A Google számos helyen használja az SQLite-ot. A leginkább ismert felhasználási terület az Android rendszer, de ez végzi Google egyre népszerűbb böngészőjének, a Chrome-nak az adattárolását is.
- Az Apple is használja a rendszert a MacOS X, Apple Mail, Safari, iTunes szoftverekben, valamint az iPhone rendszerben.
- A széles körben használt Dropbox adatmentő és szinkronizáló szolgáltatás kliens oldalán ebben tárolják az adatokat.

Hogyan válasszunk?

Felmerül a kérdés, hogy a felsoroltak közül melyik az, amelyik egy adott feladatra megfelelő. Ha hálózaton is elérhető, többfelhasználós adatbázis kezelőre van szükség, akkor a kérdés a PostgreSQL és a MariaDB között fog eldőlni. Általánosságban elmondható, hogy mindkét rendszer kiválóan használható általános feladatokra. De a rendszer használatát jól ismerők nagyobb száma (korábbi MySQL DBA-k), elterjedtsége (a MySQL-t is beleértve), illetve népszerűségének gyors növekedése miatt a MariaDB⁴³ a legjobb választás. Ha mindenképpen szeretnénk nagyvállalati rutinnal licencvásárlással egybekötni a támogatási igényünket, akkor javasolt a MySQL nagyvállalatoknak szánt változatai közül választani. A legjobb tanács, ami adható, hogy a feladathoz válasszunk esz-

³² <http://www.mozilla.org>

³³ <https://www.mozilla.org/hu/thunderbird/>

³⁴ <http://www.sqlite.org/famous.html>

³⁵ <http://www.sqlite.org/copyright.html>

³⁶ http://en.wikipedia.org/wiki/Public_Domain

³⁷ <http://www.sqlite.org/features.html>

³⁸ <http://sqliteman.com>

³⁹ <http://sqlitestudio.pl/index.rvt>

⁴⁰ <http://www.sqlitemanager.org/>

⁴¹ http://hu.wikipedia.org/wiki/Multi-master_replication

⁴² <http://www.sqlite.org/famous.html>

⁴³ <http://db-engines.com/en/system/MariaDB%3BPostgreSQL>

közt, és ne fordítva. Azaz, nézzük meg, hogy közép távon mire van szükségünk, mekkora adatmennyiség, milyen adatbázis-felépítés, milyen típusú adatbázis-kiterhelés várható, és ezt az ajánlásokkal és a két projekt weboldalán feltüntetett tudásmátrixszal összevetve döntünk. Fontos további szempont, hogy a MariaDB azt az utat járja, amit előtte a MySQL sok sok éven át, mielőtt felvásárolták, így aki biztos szeretne lenni abban, hogy még sokáig a közösség támogatását fogja élvezni, annak érdemes megfontolni a MariaDB-re való váltást.