

Hatvany Béla Csaba

MySQL.NET

MySQL Server adatbázis-programozás
.NET környezetben

Hatvany Béla Csaba

MySQL.NET

**MySQL Server adatbázis-programozás
.NET környezetben**

BBS-INFO Kiadó - 2007.

Minden jog fenntartva! A könyv vagy annak oldalainak másolása, sokszorosítása csak a kiadó írásbeli hozzájárulásával történhet.

A könyv nagyobb mennyiségben megrendelhető a kiadónál:
BBS-INFO Kft. 1630 Bp. Pf. 21. Tel.: 407-17-07

A könyv megírásakor a szerkesztő és a kiadó a lehető legnagyobb gondossággal járt el. Ennek ellenére a könyvben előfordulhatnak hibák. Az ezen hibákból eredő esetleges károkért sem a szerkesztő sem a kiadó semmiféle felelősséggel nem tartozik, de a kiadó szívesen fogadja, ha ezen hibákra felhívják figyelmét.

ISBN 978-963-9425-16-3
E-book ISBN 9786156364159

Kiadja a BBS-INFO Kft.
1630 Budapest, Pf. 21.
Felelős kiadó: a BBS-INFO Kft. ügyvezetője

Nyomdai munkák: Biró Family Nyomda
Felelős vezető: Biró Krisztián

Tartalomjegyzék

Előszó.....	10
I. rész BEVEZETÉS.....	15
1. fejezet A MySQL Server madártávlattól.....	16
1.1. Idevonatkozó anyagok az interneten.....	22
2. fejezet A MySQL Server telepítése.....	24
2.1. A szerver letöltése és telepítése.....	24
2.2. A szerver segédeszközei: mysql, mysqladmin és mysqldump.....	37
2.2.1. A mysql konzolalkalmazás.....	39
2.2.2. A mysqladmin és a mysqldump segédeszközök.....	53
2.3. Idevonatkozó anyagok az interneten.....	55
3. fejezet Segédeszközök.....	56
3.1. A MySQL Administrator segédeszköz.....	57
3.2. A MySQL Query Browser.....	59
3.3. A MySQL Migration Toolkit.....	60
3.4. A MySQL Workbench.....	61
3.5. Az SQLyog segédeszköz.....	62
3.6. A DBManager Professional.....	65
3.7. A MyDB Studio.....	66
3.8. A Navicat MySQL.....	68
3.9. A phpMyAdmin webes segédeszköz.....	69
3.10. Idevonatkozó anyagok az interneten.....	75
II. rész - ALAPOK.....	77
4. fejezet A MySQL adatbázisok.....	78
4.1. Rövid bevezetés az adatbázis-elméletbe.....	78
4.1.1. Normálformák.....	79
4.1.2. Relációk.....	87
4.1.3. Kulcsok és referenciális integritás.....	89
4.1.4. Indexek.....	93

4.2.	MySQL adattípusok	95
4.2.1.	Az xxxINT adattípusok.....	95
4.2.2.	A FLOAT és DOUBLE vagy REAL adattípusok	98
4.2.3.	A DECIMAL vagy NUMERIC adattípus	99
4.2.4.	A DATE, TIME, DATETIME, YEAR és TIMESTAMP adattípusok.....	99
4.2.5.	A CHAR, VARCHAR és xxxTEXT adattípusok.....	101
4.2.6.	Az xxxBLOB adattípusok	104
4.2.7.	Az ENUM és SET adattípusok.....	107
4.3.	MySQL táblatípusok	110
4.3.1.	A MyISAM táblatípus	113
4.3.2.	Az ISAM táblatípus.....	114
4.3.3.	Az InnoDB táblatípus.....	114
4.3.4.	A BDB táblatípus	115
4.3.5.	A MERGE táblatípus.....	115
4.3.6.	A HEAP vagy MEMORY táblatípus.....	116
4.3.7.	Ideiglenes táblák	116
4.4.	Adatbázisok létrehozása.....	117
4.4.1.	Névadás és írásmód	117
4.4.2.	Adatbázisok létrehozása, kiválasztása és törlése	118
4.4.3.	Táblák létrehozása, módosítása és törlése.....	119
4.4.4.	Indexek létrehozása és törlése.....	121
4.4.5.	Információ lekérdezése adatbázisokról, táblákról és változóktól	121
4.5.	Példa adatbázisok.....	123
4.5.1.	A mysqlbooks példaadatbázis	124
4.6.	Idevonatkozó anyagok az interneten.....	132
5.	fejezet Bevezetés a MySQL SQL lekérdező nyelvbe	134
5.1.	Adatok lekérdezése: a SELECT utasítás	135
5.1.1.	Egyszerű lekérdezések.....	135
5.1.2.	Rendező lekérdezések.....	141
5.1.3.	Szűrő lekérdezések.....	144
5.1.4.	Több táblát érintő lekérdezések.....	152
5.1.5.	Lekérdezések egyesítése	162
5.1.6.	Lekérdezések csoportosítása aggregát függvényekkel	164
5.1.7.	Allekérdezések.....	169
5.2.	Adatok módosítása.....	185
5.2.1.	Adatok hozzáfűzése	185
5.2.2.	Adatok frissítése	189
5.2.3.	Adatok törlése	190
5.3.	Változók és függvények.....	192
5.3.1.	Változók.....	192
5.3.2.	Függvények.....	199
5.4.	Tranzakciók.....	204

5.5.	Idevonatkozó anyagok az interneten.....	210
6.	fejezet A MySQL Server 5.0 újdonságai.....	211
6.1.	Tárolt függvények és eljárások	211
6.1.1.	Függvények és eljárások létrehozása	211
6.1.2.	A változók használata.....	215
6.1.3.	A tárolt eljárások paramétertípusai.....	217
6.1.4.	Feltételes és iteratív utasítások	221
6.1.5.	Kezelők és feltételek.....	232
6.2.	Kurzorok.....	239
6.3.	Triggerek.....	243
6.4.	Nézetek	249
6.5.	A MySQL információs sémája	260
6.5.1.	A karakterkészlet metaadatai	262
6.5.2.	Az elsődleges objektumok metaadatai	264
6.5.3.	Az tárolt eljárások és triggerek metaadatai.....	272
6.5.4.	A jogosultságok metaadatai.....	276
6.5.5.	A statisztikai metaadatok	278
6.6.	Idevonatkozó anyagok az interneten.....	280
7.	fejezet Felügyeleti és biztonsági feladatok.....	282
7.1.	A MySQL Server beállítása	283
7.1.1.	Az alapmappa és az adatmappa beállítása	285
7.1.2.	A kommunikációs protokoll beállítása.....	286
7.1.3.	Az alapértelmezett táblatípus beállítása.....	288
7.1.4.	Az SQL üzemmód beállítása.....	288
7.1.5.	A memóriakezelés beállítása.....	290
7.1.6.	A gyorsítótár beállítása	292
7.1.7.	Az alapértelmezett karakterkészlet és egybevetés beállítása..	294
7.1.8.	A hibajelentések nyelvének beállítása	295
7.1.9.	A szerver paramétereinek beállítása a MySQL Administrator segédeszközzel.....	296
7.2.	A MySQL Server hozzáférési jogosultságrendszere.....	301
7.2.1.	Jogosultságok a MySQL szerveren.....	306
7.2.2.	A jogosultságok kezelése.....	315
7.2.3.	Első tennivalók a MySQL szerver biztonságának növelésére	318
7.3.	Biztonsági másolatok (<i>backup</i>) és az adatok helyreállítása.....	321
7.4.	Idevonatkozó anyagok az interneten.....	331
III rész	PROGRAMOZÁS.....	333
8.	fejezet A .NET technológia és az ADO.NET.....	335
8.1.	A .NET technológiáról dióhéjban.....	335
8.2.	ADO.NET - a .NET adatkezelő komponense	337
8.2.1.	A .NET adatszolgáltatói.....	338
8.3.	Az ADO.NET kapcsolat nélküli osztályai.....	340
8.3.1.	Kapcsolatteremtés az ADO.NET segítségével.....	340

8.3.2.	Az ADO.NET kapcsolt osztályai	341
8.3.	Idevonatkozó anyagok az interneten.....	342
9. fejezet	System.Data – a kapcsolat nélküli adatok világa	343
9.1.	A DataTable, a DataColumn és a DataRow osztályok	345
9.1.1.	A DataTable osztály konstruktorai és tulajdonságai	346
9.1.2.	A DataColumn osztály konstruktorai és tulajdonságai	348
9.1.3.	A DataRow osztály és sorok hozzáadása egy táblához.....	353
9.2.	Táblák rendezése és szűrése.....	357
9.3.	Sorok frissítése és törlése	357
9.4.	Sorok verziója és állapota	361
9.5.	Táblaesemények.....	363
9.6.	Adatnézetek.....	367
9.7.	A DataSet osztály.....	372
9.7.1.	A DataSet osztály konstruktorai és tulajdonságai	372
9.7.2.	Táblák közötti relációk.....	375
9.7.3.	Táblákra rótt megszorítások	381
9.8.	Idevonatkozó anyagok az interneten.....	388
10. fejezet	Kapcsolatteremtés.....	390
10.1.	Az egyetemes adatelérés rövid áttekintése	390
10.2.	Kapcsolatteremtés a MySQL Connector/ODBC segítségével.....	394
10.2.1.	A kapcsolat paraméterei.....	396
10.2.2.	Kapcsolat létrehozása DSN-nel	399
10.2.3.	Kapcsolat létrehozása a Visual Studio fejlesztőrendszerrel..	404
10.2.4.	Kapcsolat létrehozása programból.....	408
10.3.	Kapcsolatteremtés a MySQL Connector/Net segítségével.....	411
10.3.1.	A kapcsolat paraméterei.....	414
10.3.2.	Kapcsolat létrehozása a Visual Studio fejlesztőrendszerrel..	416
10.3.3.	Kapcsolat létrehozása programból.....	419
10.4.	Egy hasznos osztály a kapcsolatteremtésre.....	420
10.5.	Kapcsolatteremtés ASP.NET alkalmazásokból	427
10.6.	Idevonatkozó anyagok az interneten.....	431
11. fejezet	MySql.Data.MySqlClient – a kapcsolt adatok világa	432
11.1.	A MySql.Data.MySqlClient névtér.....	432
11.2.	A MySqlConnection osztály.....	435
11.3.	A MySqlCommand osztály	437
11.4.	A SqlParameter és SqlParameterCollection osztályok	440
11.5.	A MySqlDataReader osztály	444
11.6.	A MySqlDataAdapter osztály	448
11.7.	A MySqlCommandBuilder osztály	454
11.8.	A MySqlTransaction osztály	457
11.9.	A tárolt eljárások programozása.....	459
11.10.	Idevonatkozó anyagok az interneten.....	468
12. fejezet	Példaalkalmazások.....	469
12.1.	Adatnavigáció	469

12.1.1.	Előkészítések	469
12.1.2.	A program leírása	473
12.1.3.	A program felépítése	474
12.2.	Képállományok kezelése a MySQL adatbázisban	478
12.2.1.	A program leírása	478
12.2.2.	A program felépítése	481
12.2.3.	Egy alternatív megoldás	492
12.3.	Az interneten található képállományok tárolása a MySQL adatbázisba	495
12.3.1.	Előkészítések	495
12.3.2.	Az program leírása	499
12.3.1.	Az program felépítése	501
12.4.	Fő- és részletadatok megjelenítése	503
12.4.1.	A program leírása	504
12.4.2.	A program felépítése	505
12.5.	Egy MySQL adatbázis-vezérelt webalkalmazás	510
12.5.1.	Előkészítések	511
12.5.2.	A program leírása	517
12.5.3.	A program struktúrája	519
12.6.	Hogyan tovább?	525
12.7.	Idevonatkozó anyagok az interneten	526
Irodalomjegyzék		528
Tárgymutató		529

Előszó

Egy szakkönyv címe akkor jó, ha rövid, de eleve elárulja miről szól a könyv. *MySQL.NET* – a címe ennek a könyvnek s habár az olvasó máris sejtheti, hogy miről szól, a szokatlan párosítás miatt – MySQL adatbázisrendszer és .NET technológia – mégis úgy érzem, hogy mindjárt a könyv elején egy magyarázatra vár. Ezzel nem is szeretnék tovább adós maradni.

Magyarázat

A MySQL adatbázisrendszert a fejlesztők leggyakrabban a LAMP kifejezéssel hozzák kapcsolatba. A LAMP rövidítést legelőször Michael Kunze német szakújságíró alkalmazta a Linux (operációs rendszer), Apache (web szerver), MySQL (adatbázisrendszer) és PHP (programozási nyelv) négyes fogatra¹. Azóta több ilyen fejlesztési négykezet ismerünk, melyek a szakirodalomba a szoftveres verem (*software stack*) elnevezést kapták.

Két-három évvel ezelőtt a MySQL adatbázis-vezérelt (web) alkalmazásokat fejlesztése, annyit jelentett, mint a LAMP környezetben fejleszteni. Azóta a kép kezd megváltozni, helyesebben - kiegészülni. Az immár klasszikussá vált LAMP paradigma mellett, egyre jobban körvonalazódik egy másik fejlesztési környezet, amit a WIMA rövidítéssel írhatunk le: **Windows** (operációs rendszer), **Internet**

¹ <http://www.heise.de/ct/english/98/12/230/> - Michael Kunze *Lafst es leuchten LAMP: ein datenbankgestütztes Web-Publishing-System mit Freeware* című cikkének angol változata. Az eredeti a c't német számítógépes szakfolyóirat 1998/12 számában, a 230. oldalon jelent meg.

Information Services (web szerver), MySQL (adatbázisrendszer) és ASP.NET („programozási nyelv”²).

Tény, hogy egyre több .NET környezetben fejlesztett MySQL adatbázis-alkalmazásra van szükség és ezzel a folyamattal a szakirodalomnak is lépés kell tartania. Ennek ellenére csupán elvetve találunk ezzel a témával foglalkozó szakirodalmat: egy tucat cikket a világhálón, de mostanáig még nem jelent meg egy összefoglaló könyv.

Ezek az elgondolások vezettek ennek a könyvnek a megírásához. Tárnya: MySQL adatbázis-programozás .NET környezetben – innen a szokatlan cím.

Kihez szól ez a könyv?

Minden könyvnek megvan az ideális célközönsége. Ezt a könyvet két olvasócsoporthoz írtam meg. Reményeim szerint elsősorban azok a fejlesztők fogják a könyvet hasznosnak találni, akik ismerik a .NET 2.0 környezetben való programozást, de eddig nem foglalkoztak a MySQL programozásával. A második célcsoportba azok a fejlesztők tartoznak, akik jártasak a MySQL programozásban, viszont eddig nem fejlesztettek MySQL programokat a .NET 2.0 környezetben, de legalább közép-szinten ismerik a C# (vagy VB.NET) programozási nyelvet. Ezeknek a fejlesztőknek a könyv harmadik része hozza a legtöbb új információt.

Miről szól ez a könyv?

MySQL adatbázis-programozás .NET 2.0 környezetben – erről már szó esett. A két célcsoportra való tekintettel a könyv fejezeteit három részbe csoportosítottam.

Az első rész bevezető jellegű és viszonylag rövid három fejezetből áll. A MySQL Server általános tulajdonságainak és korlátainak áttekintése után, részletesen foglalkozom a szerver telepítésével. A javasolt operációs rendszer a Windows XP Professional. A harmadik fejezetben megismerünk egy sor segédeszközt is.

A MySQL Server alapjainak tárgyalása a könyv második részében kapott helyet. A 4. fejezetben a MySQL adatbázisokkal ismerkedünk meg. Egy rövid adatbázis-elméletbe való bevezetés után megismerjük a

² Itt most nem térek ki a rövidítés helyességére: természetesen az ASP.NET nem egy programozási nyelv. Helyesebb lett volna helyette a .NET keretrendszer nevét használni, de nem így történt. Mint ahogy a LAMP utolsó tagját néha Perl-re vagy Python-ra fordítják, úgy a WIMA utolsó tagja alatt egyesek .NET-et értenek.

MySQL adattípusait és táblatípusait, foglalkozunk az adatbázisok, táblák, indexek létrehozásával és módosításával és látni fogjuk, miként kérhetünk információt az adatbázisokról és a táblákról az adatbázisrendszerrel. A fejezet végén megismerkedünk a `mysqlbooks` példa-adatbázissal, mely végig fog kísérni a könyv végéig. Az 5. fejezet az SQL lekérdező nyelvet tárgyalja, különös tekintettel a MySQL Server adatbázisrendszerre. Ez egy hosszabb fejezet, melyben adatok lekérdezésekről és módosításáról, változókról és függvényekről, valamint tranzakciókról van szó. A MySQL Server 5.0 verziójában bevezetett újdonságokat a 6. fejezetben fogjuk megismerni: itt tárolt függvényekről és eljárásokról, kurzorokról és triggerokról, nézetekről és az információ sémáról olvashatunk. A második részt záró 7. fejezet a felügyeleti és biztonsági feladatokkal foglalkozik.

A könyv harmadik része a MySQL programozásának van szentelve. A nyitó 8. fejezet egy rövid bevezető a .NET technológiába és röviden foglalkozik az ADO.NET adatkezelő komponens szerepével és felépítésével is. A 9. fejezet részletesen tárgyalja az ADO.NET kapcsolat nélküli osztályait és a bemutatott anyagot számos példaprogrammal illusztrálja. A MySQL adatbázisrendszer konkrét programozásával a 10. fejezetben kezdünk foglalkozni, Itt részletesen tárgyaljuk a programból történő kapcsolatteremtést az adatbázisrendszerrel. A MySQL programozásához szükséges kapcsolt adatokat kezelő osztályokat a 11. fejezetben fogjuk megismerni. Több ilyen osztályokat definiáló adatszolgáltató létezik. A könyv csak a MySQL AB által kibocsátott **MySQL Connector/Net** adatszolgáltatóval foglalkozik. A könyv utolsó fejezete példaalkalmazásokban foglalja össze a könyvben tárgyalt anyagot. Az itt tárgyalt példaalkalmazások bonyolultsága fokozatosan nő, az utolsó két példát valódi adatbázis-vezérelt alkalmazások kissé leegyszerűsített változatainak tekinthetők. Az utolsó négy fejezetben a példaanyag C# programozási nyelven került megfogalmazásra. A letölthető archívumban ezek a példaprogramok VB.NET fogalmazásban is megtalálhatók. A programok előállítására a Visual Studio 2005 fejlesztőeszközt használhatjuk.

Minden egyes fejezet utolsó szakaszában, a fejezetben tárgyalt anyagra vonatkozó, az interneten lévő forrásokra mutató hivatkozások találhatóak. Az így elérhető források további információt nyújtanak az érdeklődő Olvasóak. Az interneten található forrás könnyebb elérésére érdekében ezen hivatkozások parancsikonjai megtalálhatók a szerző webhelyéről letölthető archívumban, a fejezeteknek megfelelő mappákban.

A könyvben szereplő példaanyag forráskódja letölthető a webhelyemről a következő címről: <http://www.hatvany-online.net/My-Books/MySQL.NET/download.aspx>.

Meggyőződésem, hogy egy szakkönyv szerzőjének munkája nem fejeződik be az utolsó fejezet megírásával és a kézirat elküldésével a kiadójának. Főleg ma nem, amikor számtalan kommunikációs lehetőség áll rendelkezésre szerző-olvasó vagy akár olvasó-olvasó párbeszédre. Ilyen párbeszédnek kívánok helyet adni a webhelyemen beállított fórummal is. Címe <http://www.hatvany-online.net/forums/mysql.net/>. Mindemellett szívesen fogadok az alábbi email-címemen észrevételeket, bírálatokat, kérdéseket.

Köszönetnyilvánítás

Ugyanúgy, mint a többi könyvemet, ezt sem írhattam volna meg feleségem támogatása nélkül. Köszönöm a végtelen türelmét.

Köszönöm továbbá a BBS-INFO könyvkiadónak, mindenekelőtt Bártfai Barnabás úrnak, hogy vállalta ennek a „szokatlan című” könyvnek a kiadását.

dr. Hatvany Béla Csaba
(csaba@hatvany-online.net)

2007. május 20.
Kuchen, Németország

I. rész

BEVEZETÉS

Ez a bevezető rész három rövid fejezetből áll.

Az első fejezetben (**A MySQL Server madártávtól**) röviden áttekintjük a MySQL Server képességeit és „hiányosságait”, továbbá megtudjuk, melyek a jelenleg létező verziók, illetve melyik verzióval foglalkozunk a továbbiakban.

A második fejezetben (**A MySQL Server telepítése**) már konkrétan foglalkozunk a szerverrel: megtudjuk honnan tölthetjük le, hogyan telepíthetjük a gépünkre és hogyan állíthatjuk be legfontosabb paramétereit. A fejezet végén, egyszerű utasításokkal kipróbáljuk a szerver működését.

A harmadik fejezet (**Segédeszközök**) a segédeszközökkel foglalkozik. Megismerkedünk a MySQL AB által rendelkezésünkre bocsátott segédeszközökkel, valamint egy sor más cégektől származó segédprogrammal is.

1. fejezet

A MySQL Server madártávlatból

Rajongói azt hangoztatják, hogy a leggyorsabb, legolcsóbb, legmegbízhatóbb - egy szóval, a legjobb relációs adatbázis-rendszer. Egyes ellenzői még relációs adatbázis-rendszer jellegét is megkérdőjelezik. Fejlesztői szerint „a legnépszerűbb nyitott forrású adatbázis” [1] (ld. az 1.1 ábrát).



1.1 ábra: A MySQL hivatalos webhelye

Tény, hogy a MySQL több mint tízmillió telepítéssel dicsekedhet, elkezdve egyéni kis alkalmazásoktól egészen a legnagyobb cégekig, lefedve a világ összes földrészét. „Igen, még az Antarktiszt is” – olvashatjuk a MySQL honlapján [2]. A MySQL mintegy félszáz kiváló ügyfele közül, elég ha csak a következő óriásokat említem meg: CERN, Friendster, Goole, NASA, Yahoo! [3].

Ez a sikertörténet 1994-ben kezdődött, amikor a svéd TcX cég munkatársai webalkalmazásaik részére egy SQL szervert kerestek. Miután sok „jelöltet” végigteszteltek és nem találtak megfelelőnek, Michael „Monty” Widenius, a cég egyik fejlesztője, elhatározta, hogy nekifog egy saját szerver fejlesztésébe. 1995-ben Michael Widenius, David Axmark és Allan Larsson létrehozzák a svédországi Upsalában a MySQL AB részvénytársaságot (AB, a svéd *Aktiebolag* rövidítése, a magyar Rt.-nek felel meg), mely azóta a MySQL Server relációs adatbázisrendszer fejlesztésével és terjesztésével foglalkozik. A cégnek jelenleg mintegy 350 alkalmazottja van, több mint 25 országban.

1995. május 23-án kibocsátásra került a MySQL Server (röviden MySQL) első nyilvános verziója, a 3.11.1. Azóta egy-két éves ritmusban váltják egymást a főverziók:

- 2001 január: 3.23
- 2003 március: 4.0
- 2004 október: 4.1
- 2005 október: 5.0
- 2005 november: 5.1
- 2007 február: 5.2
- 2007 március: 6.0

A siker tökéletesnek mutatkozik és ennek magyarázata a MySQL Server képességeiben keresendő.

A MySQL egy relációs adatbázisrendszer. Az adatok tárolása az adatbázisok tábláiban kapnak helyet és, egy adatbázison belül, relációkat definiálhatunk a táblák között.

A MySQL több tárológéppel (storage engine) rendelkezik. A tárológépeknek megfelelően, a szerver többféle típusú táblát kezel. A fejlesztő a számára legalkalmasabb táblatípust alkalmazhatja.

A MySQL nyitott forráskódú. Bárki letöltheti az adatbázisrendszer forráskódját, ingyen használhatja, és ha úgy kívánja, akár módosíthatja is. A MySQL AB a GPL (GNU General Public License [4]) licenc szerint

szabályozza az adatbázisrendszer használatát. A MySQL kereskedelmi használatára szükséges egy kereskedelmi licenc [5].

A MySQL használata nem költséges. Az adatbázisrendszer kereskedelmi használatának ár/teljesítmény aránya nagyon vonzó. A kereskedelmi licenc ára, a hozzávásárolt szolgáltatásokkal együtt, egy pár száz euró. Nem kereskedelmi használata teljesen ingyenes.

A MySQL differenciált szerver/kliens architektúrával rendelkezik. A szerver egy többszálú (*multi-threaded*) SQL szerver, mely sokféle háttér (*backend*), kliensalkalmazást, könyvtárat, alkalmazásprogramozói felületet (*application programming interface*, API) és segédeszközt támogat. A szerver és a kliens nem kell, hogy azonos operációs rendszer alatt fusson.

A MySQL kompatibilitása (compatibility) széles. Sok kliensalkalmazás egyszerre teremthet kapcsolatot a szerverrel, a kliensek pedig egyszerre használhatnak több adatbázist. A szerver egy egész sor interfésszel interaktívan is elérhető: parancssoros kliensek, böngészők, X Window System kliensek, stb.

A MySQL programozhatósága sokrétű. A szerver sok programozási nyelv részére nyújt programozási interfészt, mit például C, C++, C#, Perl, Python, VB.NET, Java, PHP, ASP.NET, stb.

A MySQL kapcsolhatósága (connectivity) jó. A szerver támogatja az ellenőrzött hálózati kapcsolást. A szükséges jogosultsággal rendelkező felhasználó a hálózatban bárhol, az interneten is, elérheti az adatait.

A MySQL biztonsága (security) szolid. Avatatlanok nem érhetik el az adatokat. A jogosultságok kiosztása egészen oszlopszintig finomítható és kliens IP címszintre tagolható. Az adatok könnyen biztosíthatók, illetve a biztosításokból könnyen visszaállíthatók.

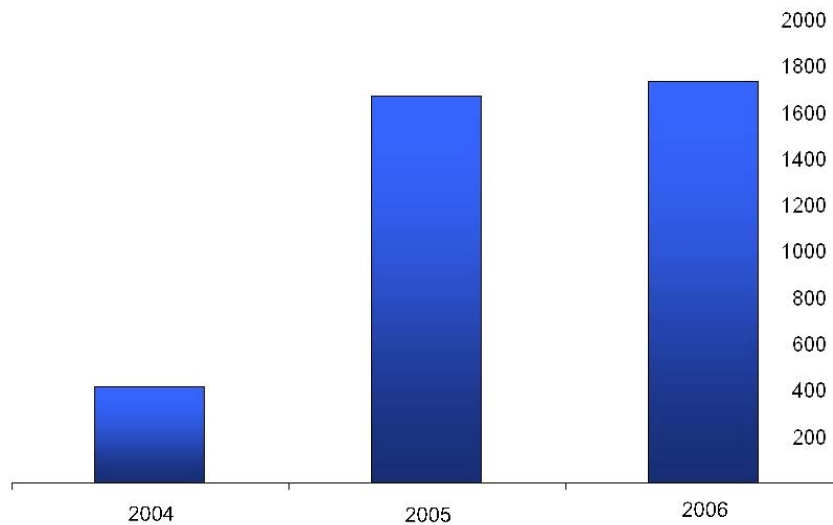
A MySQL hordozhatósága (portability) kiváló. A szerver sok UNIX dialektus és más operációs rendszer, mint a Windows, Mac OS X, IBM AIX, stb. alatt fut.

A MySQL nagy sebességű. Túlzás nélkül azt lehet állítani, hogy a MySQL a hatalmas adatmennyiséggel dolgozó és óriási számú lekérdezéseket lebonyolító adatbázis-vezérelt webalkalmazások *de facto* adatbázisa. Példa erre a Friendster webalkalmazás napi több mint másfél milliárd lekérdezéssel és szédületes 21 TB adatállománnyal [6].

Mindemellett, a MySQL használata egyszerű és könnyen elsajátítható, amiről ez a könyv is igyekszik a kedves Olvasót meggyőzni.

Ha a fenti tulajdonságokkal rendelkező adatbázisrendszert keresünk, akkor a MySQL Server a helyes választás.

A MySQL telepítése és fejlesztése kétségkívül a UNIX/Linux platformokon a legelterjedtebb. Az utóbbi években azonban, a Windows alatti telepítések és a .NET környezeti MySQL fejlesztések száma rohamosan növekedett. Erre vonatkozólag nem találtam semmilyen felmérést, de kollegáim és jómagam tapasztalata, valamint az 1.2 ábrán látható általam készített *ad hoc* felmérés, igazolja ezt az állítást. Az ábra a MySQL .NET fórum [7] utolsó három év bejegyzéseinek számát illusztrálja. 2005-ben egy négyszeres növekedést állapíthatunk meg az előző évhez viszonyítva.



1.2 ábra: A MySQL .NET fórum bejegyzéseinek száma az utolsó három évben

Ennek a folyamatnak több magyarázata lehet. Pontos felmérések hiányában, kénytelen vagyok saját tapasztalatomból következtetni. Sok kis és közepes cég kinőtte a Microsoft Access alkalmazásokat és így nagyobb teljesítményű, gyors és elfogadható költségekkel járó adatbázist keres. A MySQL az első adatbázisrendszer, amelyik ilyenkor szóba jön. Mivelhogy az említett cégeknél az irodai és prezentációs munka majdnem kizárólag a Microsoft Office termékcsomaggal lesz végrehajtva és mivelhogy ezek a cégek, gazdasági okoknál fogva, irtóznak több mint egy rendszer üzemeltetésétől, a Windows-ra telepített MySQL a természetes választásuk.

A fejlesztők részére ez a folyamat igen öröndetes, mert új tevékenységi területeket biztosít számukra. Az ilyen irányú tevékenységet kísérli meg elősegíteni ez a könyv is.

A MySQL Server verziói több változatot ismernek:

- Az Alpha változat egy fejlesztés alatt álló változatnak felel meg. A végleges alakjáig sok változáson megy át. Ez a változat nem szavatolja a szerver stabil működését. Általában csak hiba-keresésre és tesztelésre használjuk ezt a változatot.
- A Beta változat már az összes tervezett módosításokat tartalmazza, újabb lényeges módosítások nem várhatók. A változat tüzetes tesztelése még nem történt meg.
- A Gamma változat már egy kellőképpen tesztelt és messzemenően hibátlan és stabilnak számító szervernek felel meg. Hátra van a rejtett hibák keresése.
- A Production vagy Generally Available (GA) változat egy stabil, hibamentesnek bizonyult szervernek felel meg, amit a MySQL AB a termelésben való üzemeltetésre ajánl.

Jelenleg a következő verziókat találjuk a MySQL portálján:

- MySQL 4.1 GA
- MySQL 5.0 GA
- MySQL 5.1 Beta
- MySQL 6.0 Alpha – Falcon Preview

Mivelhogy a MySQL 5.0 GA verzió a jelenlegi legutolsó stabil verzió, ebben a könyvben ezzel a verzióval fogunk foglalkozni.

Kiváló tulajdonságai ellenére, természetesen a MySQL Server sem lehet egy csodaszer vagy egy csoda-adatbázis. Mint bármely más szoftvertermék, a MySQL sem adhat megoldást az összes elképzelhető feladatra. Nézzünk meg, mire nem képes a MySQL Server 5.0 GA!

A MySQL nem mindegyik táblatípusa támogatja a sor- (row locking) vagy rekordzárolást (record locking). Az adatbázis konzisztenciájának a megőrzésére, adatmódosításkor az adatokat ideiglenesen zárolni kell más felhasználók előtt, majd a módosítás elvégzése után, fel kell emelni a zárolást. Elméletileg a zárolás táblaszinten vagy sorszinten realizálható. A MyISAM típusú táblák csak táblazárolást (table locking) támogatnak, az InnoDB típusú táblák a táblazárolás mellett, a sorzárolást is támogatják.

A MySQL nem képes a MyISAM típusú táblákról forró biztonsági másolatot (*hot backup*) készíteni. A MyISAM típusú táblákról csak úgy tudunk biztonsági másolatot készíteni, ha először a táblát zároljuk. Ez nagy forgalmú és nagy adatmennyiséget tároló MyISAM típusú tábláknál természetesen hátrányos. Ha mégis szükségünk van forró biztonsági másolatokra, akkor használjuk az InnoDB típusú táblákat, amelyek támogatják a forró biztonsági másolatok elkészítését. Ehhez azonban, egy külön programot kell használnunk, mert a szerver ilyen programot nem tartalmaz.

A MySQL nem támogatja az egyéni adattípusokat (*user defined data type, UDT*). Némely adatbázisrendszer, mit például a MS SQL Server, Oracle vagy PostgreSQL, egyéni adattípusok definiálását és használatát teszik lehetővé. A MySQL adatbázisaiban nem definiálhatunk egyéni adattípust. Egyelőre ez a képesség nem szerepel a következő verziók tervezetében sem.

A MySQL a GIS adattípusok kezelésére viszonylag kevés függvényt támogat. A MySQL az Open Geospatial Consortium [8] OpenGIS® Simple Features Specifications For SQL specifikáció [9] alapján, térinformációk tárolására alkalmas, úgynevezett GIS (geographic information system) adattípust támogat [10]. Ilyen adatok kezelésére azonban, a MySQL viszonylag kevesebb függvényt támogat, mint más, erre az információtípusra szakosított adatbázisrendszer. A térinformációs adatokkal és a MySQL GIS adattípusával, speciális volta miatt, ebben a könyvben nem foglalkozunk. Anders Karlsson cikke [11] egy jó bevezető a témába.

A MySQL nem támogatja az XML dokumentumokat. A MySQL 5.0 GA verziója ugyan még nem támogatja az XML dokumentumokat, de a jelenleg Beta változatban levő 5.1 verzió márt tartalmaz két függvényt, amivel karakterláncként tárolt XML dokumentumokat kezelhetünk [12]. Alexander Barkov és Peter Gulutzan cikke [13] egy pár példával bemutatja, mit várhatunk a végleges 5.1 verziótól. Sajnos, sem az 5.1 sem a következő verziók nem tervezik egy natív XML adattípus bevezetését.

Ez a felsorolás, ugyanúgy mint a MySQL tulajdonságainak listája, távolról sem teljes, de segít eligazodni, ha döntenünk kell a MySQL mellett vagy ellene.

Már szó volt róla, hogy ez a könyv a MySQL 5.0 GA verzióval foglalkozik. Ettől a verziótól kezdve a szerver két változatban kerül kibocsátásra:

- Enterprise Server – előfizetés (*subscription*) alapján sok szolgáltatással együtt megvásárolható változat. Az előfizető havonta kap frissítéseket (*update*) és negyedévenként kiegészítő csomagokat (*service pack*).
- Community Server – szabadon letölthető változat bináris és forráskód alakban. Az újabb verziók megjelenése nincs előre ütemezve. Előfordulhat, hogy két teljes (bináris és forráskód alakú) kibocsátás között, egyes verziók csak forráskód alakban kerülnek kibocsátásra. Ez a változat nem hoz magával semmilyen szolgáltatást.

A két változat között csak a kibocsátás stratégiája és a szolgáltatások mennyisége eltérő. Mindkét változat, azonos verziószám esetében, ugyanazt a szerveret tartalmazza.

Ezeket figyelembe véve, most pontosíthatók: a továbbiakban a MySQL Community Server 5.0 GA szerverrel fogunk foglalkozni. A következő fejezetben látni fogjuk, hogyan telepíthetjük a MySQL Community Server 5.0 GA szervert a gépünkre.

1.1. Idevonatkozó anyagok az interneten

- [1] *MySQL AB The world's most popular open source database* - a MySQL AB honlapja
<http://www.mysql.com/>
- [2] *MySQL AB :: Why MySQL* - a MySQL AB honlapja a MySQL rövid ismertetésével és bemutató cikkekkel
<http://www.mysql.com/why-mysql/>
- [3] *MySQL AB :: MySQL Customers by Industry* - a MySQL fontosabb ügyfelei
<http://www.mysql.com/customers/>
- [4] *GNU General Public Licence* – a GPL licenc honlapja a Free Software Foundation webhelyén
<http://www.fsf.org/licensing/licenses/gpl.html>
- [5] *MySQL AB :: MySQL Licensing Policy* – a MySQL AB licencei a cég portálján
<http://www.mysql.com/company/legal/licensing/>

- [6] *Friendster - Home* – a Friendster portál
<http://www.friendster.com/>
- [7] *MySQL Forums:: .NET* – a MySQL .NET fóruma
<http://forums.mysql.com/list.php?38>
- [8] *Welcome to the OGC Website* OGC® – az Open Geospatial Consortium webhelye
<http://www.opengeospatial.org/>
- [9] *Simple Feature Access - Part 2 SQL Option* OGC® – az OpenGIS® specifikáció honlapja
<http://www.opengeospatial.org/standards/sfs>
- [10] *MySQL AB :: MySQL 5.0 Reference Manual 16 Spatial Extensions* – a MySQL dokumentáció a GIS adattípusról
<http://dev.mysql.com/doc/refman/5.0/en/spatial-extensions.html>
- [11] *MySQL AB :: GIS and Spatial Extensions with MySQL* – Anders Karlsson cikke a GIS adattípus támogatásáról a MySQL-ben
<http://dev.mysql.com/tech-resources/articles/4.1/gis-with-mysql.html>
- [12] *MySQL AB :: MySQL 5.1 Reference Manual 12.10 XML Functions* – a MySQL 5.1 dokumentáció az XML függvényekről
<http://dev.mysql.com/doc/refman/5.1/en/xml-functions.html>
- [13] *MySQL AB MySQL 5.1's New XML Functions* – Alexander Barkov és Peter Gulutzan cikke a MySQL 5.1 XML függvényeiről
<http://dev.mysql.com/tech-resources/articles/mysql-5.1-xml.html>

2. fejezet

A MySQL Server telepítése

A MySQL Server hivatalos honlapján [1] megtaláljuk a letölthető anyagok listáját. Minket pillanatnyilag a **MySQL Server 5.0.27 GA** [2] érdekel.

2.1. A szerver letöltése és telepítése

A **MySQL Server 5.0 GA** öt változatban tölthető le. A MySQL AB honlapján [2] **Windows downloads** cím alatt letölthetjük a következő telepítőt, illetve archívumokat:

- **Windows Essentials (x86)**: `mysql-essential-5.0.27-win32.msi` (16,8 MB) - Microsoft Windows Installer (MSI) telepítő, mely a szerver lényeges komponenseit, beleértve a konzolt vagy a parancssor eszközt és a **Server Instance Configuration Wizard** konfigurációs varázslót tartalmazza,
- **Windows (x86) ZIP/Setup.EXE**: `mysql-5.0.27-win32.zip` (40,3 MB) - zip archívum, mely a telepítőt és a **Server Instance Configuration Wizard** konfigurációs varázslót tartalmazza,
- **Without installer (unzip in C:)**: `mysql-noinstall-5.0.27-win32.zip` (53,2 MB) - zip archívum, mely az összes bináris állományt tartalmazza, telepítő nélkül.

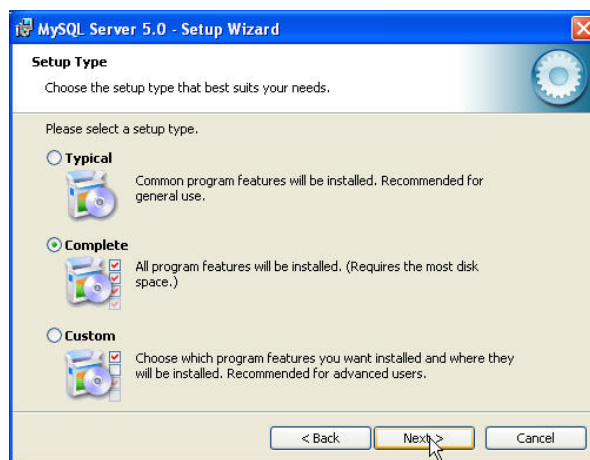
míg a honlap alján, a **Source downloads** cím alatt a forráskódot tartalmazó archívumokat tölthetjük le:

- **Windows Source (zip)**: `mysql-5.0.27-win-src.zip` (22,8 MB) - a forráskódot tartalmazó zip archívum,
- **Windows Source (tar.gz)**: `mysql-5.0.27-win-src.tar.gz` (19,6 MB) - a forráskódot tartalmazó tar archívum

Az 5.0 GA verzió esetében a fejlesztők a „lényeges” (nevükben az -essential- kifejezés szerepel) csomag telepítését javasolják és mi is ezt fogjuk használni.

Töltsük hát le a szóban forgó csomagot és nézzük meg, miként zajlik le a telepítésük. Ha esetleg a 4.1.5 verziónál korábbi verziót már telepítettünk a gépünkre, akkor azt távolítsuk el. Későbbi verziók esetében a telepítő sikeresen megbirkózik a meglévő verzió frissítésével. A könyv megjelenéséig újabb 5.0.x verziók jelenhetnek meg, de nem valószínű, hogy belátható időn belül lényeges változásokat hoznak. Jelenleg létezik egy 5.1.x Beta verzió is, de ezzel nem foglalkozunk, mivelhogy még sok módosításon eshet át.

Miután elindítjuk a telepítő programot, a felhasználót üdvöző ablakon nyomjuk meg a **Next >** gombot. A következő ablakon három lehetőség közül választhatunk (2.1 ábra):



2.1 ábra: Három változatban telepíthetjük a szervert

- **Typical** - az általános használatra szükséges szerverkomponenseket telepíti,
- **Comple**t - az összes szerverkomponenst telepíti,
- **Custom** - kiválaszthatjuk a telepítésre kerülő szerverkomponenseket.

Válasszuk ki a **Comple**t választógombot és nyomjuk meg a **Next >** gombot, hogy elindítsuk a telepítést.



2.2 ábra: Regisztrálhatunk a MySQL.com portálon

A következő ablakon (2.2 ábra) regisztrálhatjuk az email címünket a MySQL portálon, illetve ha már regisztráltuk, bejelentkezhethetünk a portálon. A regisztrálás díjmentes és nem kötelező. Regisztrált felhasználók rendszeresen kapják a portál hírleveleit és részt vehetnek a portál fórumain. Ha nem kívánunk regisztrálni vagy bejelentkezni, akkor válasszuk ki a **Skip Sign-Up** választógombot és nyomjuk meg a **Next >** gombot, hogy elindítsuk a tényleges telepítést.

A telepítés rendszerint problémamentesen zajlik le. Befejezését a 2.3 ábrán látható ablak hozza tudomásunkra és itt van lehetőségünk arra, hogy a telepített szerver beállításait egy varázsló (wizard) segítségével elvégezzük.



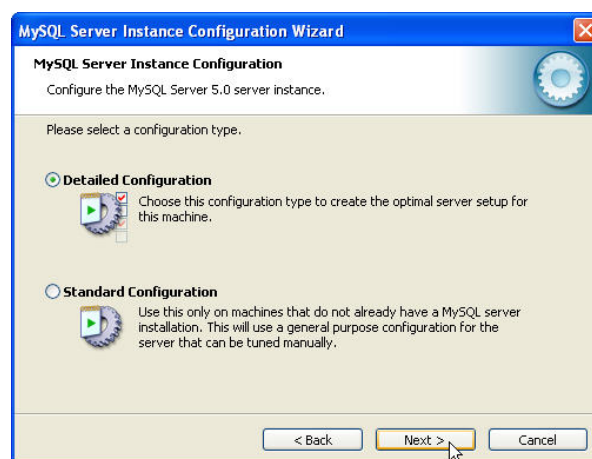
2.3 ábra: A szerver telepítése után, a beállításokat egy varázsló segítségével végezzük el

Jelöljük ki a **Configure the MySQL Server now** jelölőgombot, majd nyomjuk meg a **Finish** gombot, hogy befejezzük a telepítést és elindítsuk a **MySQL Server Instance Configuration Wizard** varázslót, amellyel beállíthatjuk a MySQL Server paramétereit.

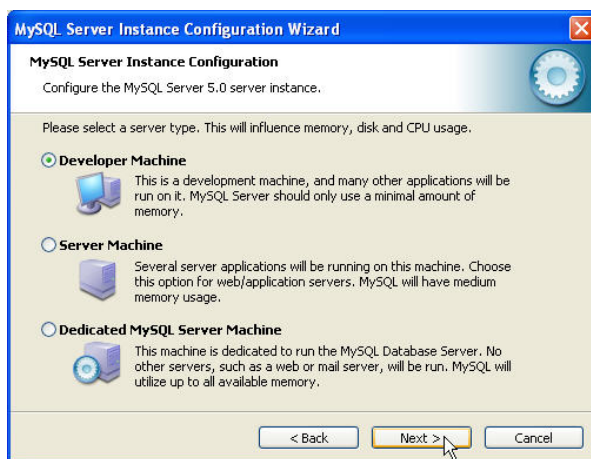
A varázsló üdvözlő ablakán, nyomjuk meg a **Next >** gombot. A következő ablakon (2.4 ábra) lehetőségünk van egy részletes (**Detailed Configuration**) és egy szabványos (**Standard Configuration**) beállítás között választani. Válasszuk ki a **Detailed Configuration** választógombot, hogy magunk szabjuk meg a beállítás részleteit és nyomjuk meg a **Next >** gombot.

Következő lépésként meghatározzuk a szerver típusát (2.5 ábra). Háromféle szervertípus között választhatunk:

- **Developer Machine** - a szerver a minimálisan szükséges memóriát használja. Válasszuk ezt a beállítást ha a szerver mellett más alkalmazások is futnak a gépen,
- **Server Machine** - ezt a beállítást akkor választjuk, ha a gépen nem fejlesztünk alkalmazásokat, mit például web szerverek esetében. Ezzel a beállítással a szerver közepes nagyságú memóriát vesz igénybe,
- **Dedicated MySQL Server Machine** - ha a gépünket csak a MySQL Server veszi igénybe, azaz egy dedikált szervert kívánunk beállítani, akkor ez a megfelelő beállítás.



2.4 ábra: Részletes vagy szabványos beállítás?

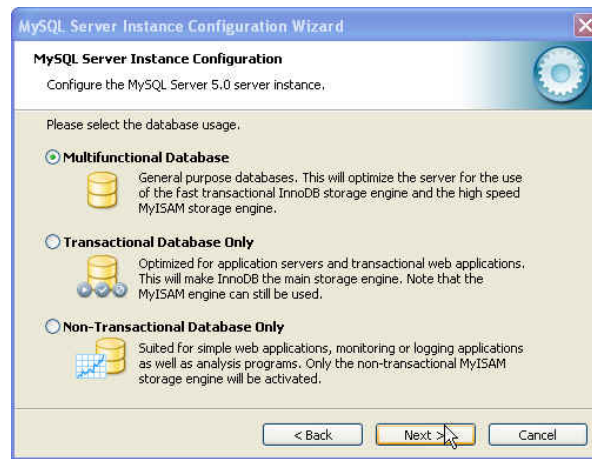


2.5 ábra: A három lehetséges szervertípus

Esetünkben fejlesztői gépről lévén szó, válasszuk ki az első beállítást és nyomjuk meg a **Next >** gombot.

A következő ablak segítségével eldönthetjük milyen típusú adatbázisokkal működjön a szerver (2.6 ábra). Itt is három lehetőség kínálkozik, amellyel a MySQL Server adatbázis-típusait választjuk ki (az adatbázis-típusokkal részletesen a 4. fejezetben foglalkozunk):

- **Multifunctional Database** - a szerver egyaránt támogatja az InnoDB és a MyISAM típusú adatbázisokat. Fejlesztői gép esetében ez a legmegfelelőbb beállítás.
- **Transactional Database Only** - a MyISAM típusú adatbázisok mellett, az InnoDB típusú adatbázisok képezik a szerver fő adatbázis típusát. Alapvetően tranzakciókat végrehajtó (webes) alkalmazásokat kiszolgáló szerverek esetében ezt a beállítást válasszuk.
- **Non-Transactional Database Only** - a szerver csak a MyISAM típusú adatbázisokat támogatja. Közöséges (webes) alkalmazásokat kiszolgáló szerver esetében emellett a beállítás mellett döntsünk.



2.6 ábra: Beállítjuk a támogatott adatbázistípusokat

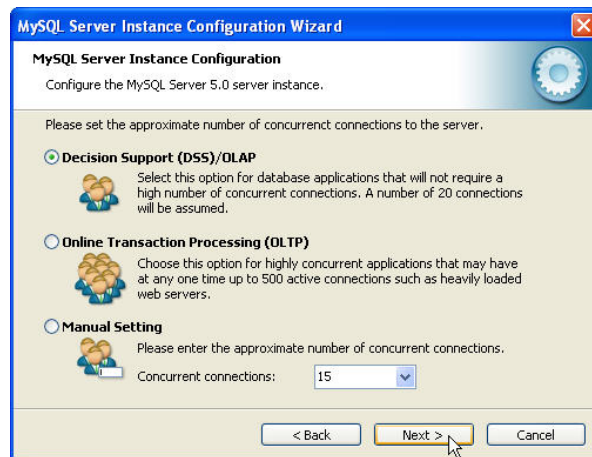
Mi természetesen a multifunkcionális adatbázisok mellett döntünk, mivelhogy ezt a telepítést tanulmányozási célokra fogjuk felhasználni. A **Next >** gombbal eljutunk a következő ablakhoz (2.7 ábra), ahol eldönthetjük, hova írja a szerver az InnoDB típusú adatbázisok adatállományát, ha nem kívánjuk hogy a szerver az alapértelmezett mappába írja. Az alapértelmezett mappa a telepítés mappája (**Installation Path**). A szerver szempontjából lényegtelen, hogy melyik mappába helyezi az InnoDB típusú adatbázisok adatállományát, nekünk viszont egyszerűbb a dolgunk, ha ezek az állományokat a telepítés mappájában keressük. Ezért fogadjuk el a javasolt mappát és nyomjuk meg a **Next >** gombot.

Ezután az egyidejűleg engedélyezett kapcsolatok számáról kell döntenünk. A 2.8 ábrán látható ablakon három választógombot látunk:

- **Decision Support (DSS)/OLAP** - válasszuk ki ezt a választógombot, ha legfeljebb 20 egyidejű kapcsolattal megelégszünk.
- **Online Transaction Processing (OLTP)** - válasszuk ki ezt a választógombot ha sok egyidejű kapcsolatra számítunk, mint például egy web szerver esetében. Az OLTP legfeljebb 500 kapcsolatot biztosít egyidejűleg.
- **Manual Setting** - ha ezt a választógombot választjuk ki, akkor megadhatjuk a feltételezett egyidejű kapcsolatok számát.



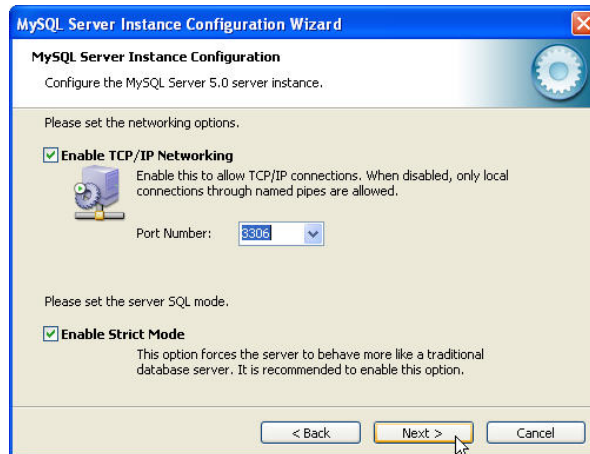
2.7 ábra: Beállítjuk az InnoDB típusú adatbázisok adatállományának helyét



2.8 ábra: Beállítjuk a kívánt egyidejű kapcsolatok maximális számát

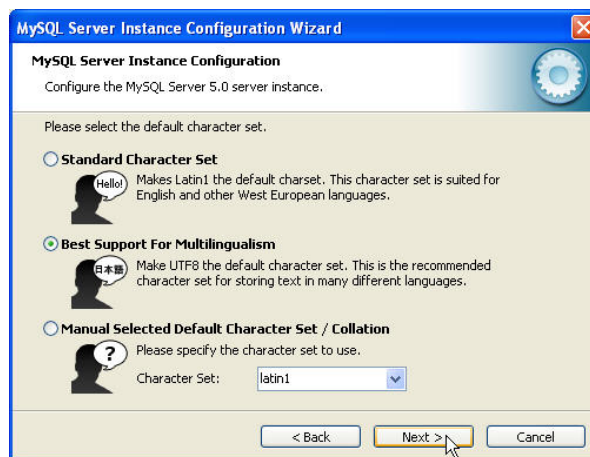
A MySQL Server tanulmányozására elegendő ha az OLAP beállítást választjuk. A **Next >** gombbal továbblépünk a hálózati kapcsolat beállításához (2.9 ábra). Itt eldönthetjük, hogy kívánunk-e TCP/IP kapcsolatot használni és megadhatjuk a használt portot. Windows alatt az alapértelmezett kapcsolat az TCP/IP kapcsolat, ezért hagyjuk az **Enable TCP/IP Networking** jelölőgombot kijelölve. Ha szabad a 3306 port a gépünkön, akkor itt is elfogadhatjuk a javasolt portot. Végül jelöljük ki az **Enable Strict Mode** jelölőgombot, amint azt a varázsló javasolja, hogy a szerver hagyományos (*traditional*) adatbázis-szerverként működjön. Ha a MySQL szervert nem hagyományos szerverként üzemeltetjük, akkor abban az esetben, amikor hibás adatot próbálunk

egy oszlopba elhelyezni, a szerver, bizonyos szabályokat követve, átalakítja a hibás adatot az oszlop adattípusára. Ezt az átalakítást a szerver egy figyelmeztetéssel hozza tudomásunkra. Mi ilyen „automatikus” átalakításokat el szeretnénk kerülni és ezért választjuk a hagyományos üzemmódot. A dokumentáció szerint ez azt jelenti hogy a szerver „egy hibát generál egy figyelmeztetés helyett” (*give an error instead a warning*), ha hibás adatot próbálunk egy oszlopba elhelyezni és nem fogadja el a hibás adatot.



2.9 ábra: A TCP/IP támogatás és a port beállítása

Ismét a **Next >** gombot használva, továbblépünk és a karakterkészlet beállításához jutunk el (2.10 ábra).



2.10 ábra: A karakterkészlet beállítása

Itt a következő lehetőségünk van

- **Standard Character Set** - alapértelmezett karakterkészletnek a latin1 karakterkészletet állítja be, ami a legtöbb nyugat-európai nyelveknek felel meg,
- **Best Support For Multilingualism** - alapértelmezett karakterkészletnek az UTF8 karakterkészletet állítja be, így gyakorlatilag bármilyen nyelvű szöveg használható,
- **Manual Selected Default Character Set/Collation** – alapértelmezett karakterkészletnek a **Character Set**: legördülő listából kiválasztott karakterkészletet állítja be.

Válasszuk ki a kívánt beállítást! Célszerű az UTF8 karakterkészletet választani alapértelmezett karakterkészletnek. Az itt beállított karakterkészlet lesz a szerver alapértelmezett karakterkészlete. Később látni fogjuk, hogy adatbázis, tábla és oszlop szinten is megadhatunk karakterkészletet, melyek rendre felülírják a szerver, az adatbázis és a tábla alapértelmezett karakterkészletét. Nyugtázzuk a választásunkat a **Next >** gombbal és térjünk át a Windows szolgáltatás beállítására (2.11 ábra).

A MySQL Server fejlesztői azt javasolják, hogy Windows alatt a szervert Windows szolgáltatásként futtassuk. Itt megadhatjuk a szolgáltatás nevét, például MySQL. Az **Include Bin Directory in Windows PATH** jelölőnégyzettel beállíthatjuk a PATH Windows rendszerváltozót úgy, hogy egy konzolalkalmazásból parancssorból elérhessük a szervert.



2.11 ábra: A Windows szolgáltatás beállítása

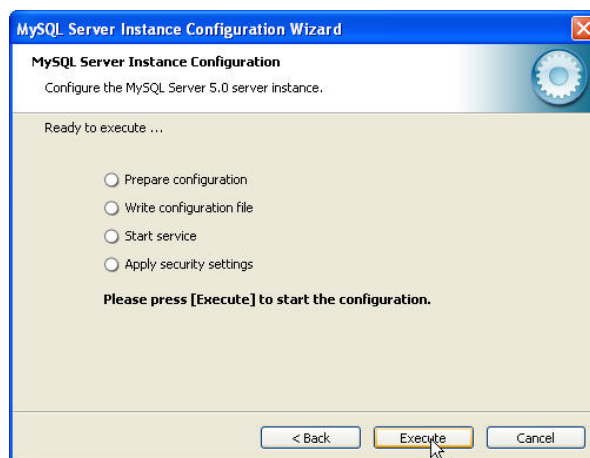
A következő ablakon (2.12 ábra) megadhatjuk az alapértelmezett root rendszergazda (*administrator*) jogosultsággal rendelkező felhasználó

ló jelszavát. Biztonsági okok miatt, amire később még visszatérünk, ne jelöljük ki az **Enable root access from remote host** és a **Create An Anonymous Account** jelölőnégyzeteket. Ezzel megakadályozzuk, hogy távoli gépekről a root rendszergazda, illetve névtelen felhasználók bejelentkezzenek a szerverre.



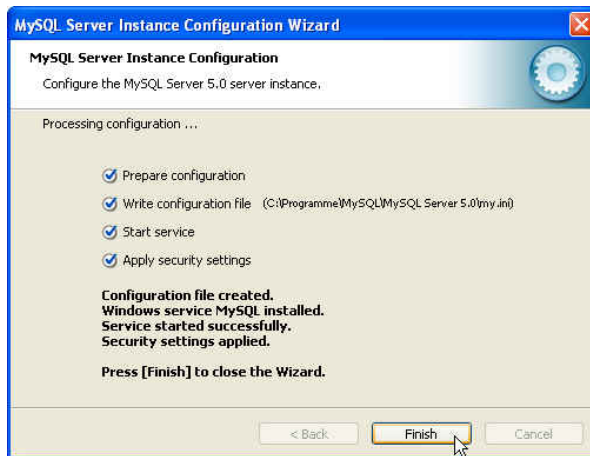
2.12 ábra: A root rendszergazda jelszavának megadása

Ezzel befejeztük a szerver indítóparamétereinek (*start parameter*) beállítását és ha megnyomjuk a **Next >** gombot, a varázsló kész a szervert a megadott paraméterek szerint beállítani (2.13 ábra). Nyomjuk meg ezen az ablakon az **Execute** gombot.



2.13 ábra: A varázsló kész a szerver beállítására

A varázsló egy ablakban mutatja a beállítás lépéseit és sikeres végrehajtását (2.14 ábra). Ha a beállítás alatt hibák keletkeztek, a hibajelentések ezen az ablakon jelennek meg. A beállítás rendszerint hibátlanul zajlik le.



2.14 ábra: A varázsló sikeresen befejezte a szerver beállítását

Alapértelmezés szerint a szervert a telepítő a C:\Program Files\MySQL\MySQL Server 5.0 mappába telepíti. Amint azt a 2.14 ábrán láthatjuk, a **MySQL Server Instance Configuration Wizard** varázsló ebbe a mappába helyezi a my.ini konfigurációs állományt. Ezzel az állománnyal többször fogunk találkozni. Ezen kívül a varázsló a következőket írja ki a ablakra:

Configuration file created. (Konfigurációs állomány létrehozva.)

Windows service MySQL installed. (MySQL Windows szolgáltatás telepítve.)

Service started successfully. (Szolgáltatás sikeresen indítva.)

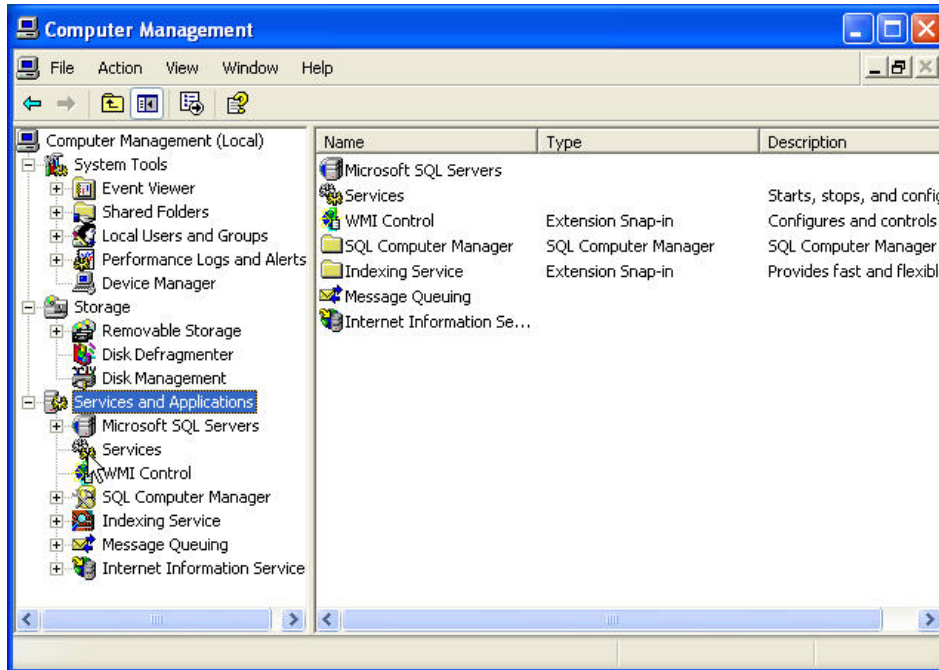
Security settings applied. (Biztonsági beállítások alkalmazva.)

Ezzel be is fejeztük a MySQL Server telepítését és beállítását. Mivelhogy szerverünk Windows szolgáltatásként működik, legegyszerűbben egy Windows konzolból tudjuk ellenőrizni, hogy működik-e:

```
>net start
These Windows services are started:
...
  MySQL
...
```

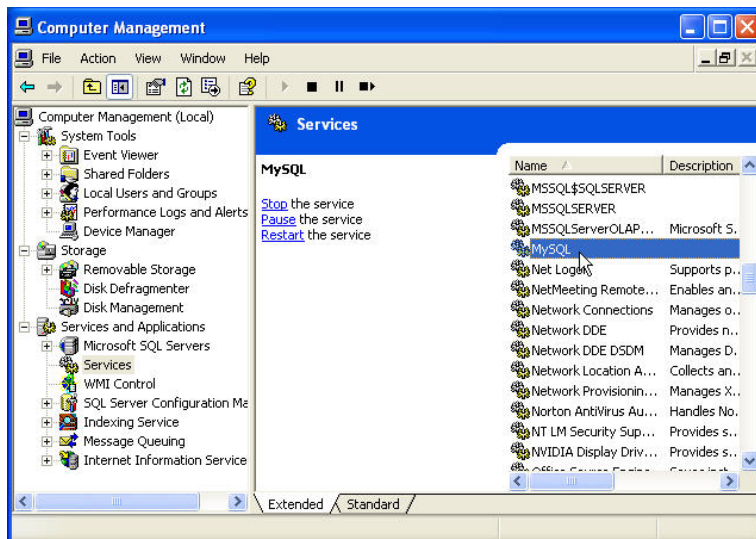
A `net stop mysql` utasítással leállíthatjuk és a `net start mysql` utasítással ismét elindíthatjuk a szervert. (Üzemelő szervert sohasem szabad a `net` utasítással leállítani! Erre a célra sokkal megfelelőbb módszerek állnak rendelkezésünkre.) A MySQL szerver telepítése után, a gépünk kikapcsolásával a szerver természetesen leáll és a gépünk bekapcsolásával automatikusan elindul.

A MySQL Windows szolgáltatást a **Computer Management** segédeszközzel is megnézhetjük. Kattintsunk a képernyőnkön az egér jobb oldali gombjával a **My Computer** ikonra. A megjelenő menüből válasszuk ki a **Manage** menüpontot. Ezzel előhívjuk a **Computer Management** ablakot (2.15 ábra).



2.15 ábra: A Computer Management ablak a Windows szolgáltatásokkal

Kattintsunk az ablak bal oldalán a **Services and Applications** ikonra, majd a **Services** ikonra. Az ablak jobb oldalán most megjelennek a Windows szolgáltatások és köztük láthatjuk a futó MySQL szolgáltatást is (2.16 ábra).



2.16 ábra: A MySQL Windows szolgáltatás

Később látni fogjuk, hogy más lehetőségek is rendelkezésünkre állnak, amelyekkel ellenőrizhetjük a MySQL Windows szolgáltatást.

Most nézzük meg, mi került a MySQL Server telepítésével a gépünkre!

Mint említettük, alapértelmezés szerint a telepítő program a szervert a `C:\Program Files\MySQL\MySQL Server 5.0` mappába telepíti. Ha megnézzük ennek a mappának a tartalmát, akkor több mappát (`bin`, `data`, `include`, `lib`, `scripts` és `share`) és állományt (`COPYING`, `EXCEPTIONS-CLIENT` és `*.ini`) találunk. Mindezekről később többet fogunk megtudni. Egyelőre jegyezzük meg a következőket:

- `bin` - ebben a mappába kaptak helyet a MySQL Server-hez tartozó bináris állományok: a `libmysql.dll` dinamikusan kapcsolódó eljáraskönyvtár (*dynamic link library*), a `mysqld.exe` szerveralkalmazás és más végrehajtható (*executable*) `*.exe` állományok. Az utóbbiak között felismerhetjük a **MySQL Server Instance Configuration Wizard** varázslót (`MySQLInstanceConfig.exe`, ld. a 2.3 ábrán). A többi állomány konzol alkalmazás és legtöbbjük fontos segítőeszköz, mint a `mysql.exe`, a `mysqladmin.exe` és a `mysqldump.exe`. A legfontosabbakról nemsokára többet fogunk megtudni.
- `data` - ebbe a mappába őrzi a szerver az adatbázisok leírását és adatait.

- `include`, `lib` - e két mappával nem foglalkozunk. Az első befoglalt állományokat (`include file`), a második függvénykönyvtárakat (`library`) tartalmaz.
- `scripts` - verziótól függően, ebben a mappában egy pár SQL szkript található.
- `share` - ide kerültek a különböző karakterkészleteket definiáló állományok és a támogatott természetes nyelveken megfogalmazott hibaüzenetek.
- `COPYING` - az állomány a GNU GENERAL PUBLIC LICENSE licenc második verziójának (Version 2, June 1991) szövegét tartalmazza, amely a MySQL Server használatát szabályozza.
- `EXCEPTION_CLIENT` - az állomány a MySQL FLOSS License Exception szövegét tartalmazza. Aktuális verziója a Version 0.4, 08 September 2005. Az okmány leírja milyen feltételek mellett használhatják a nyitott forráskódú alkalmazások (Free/Libre and Open Source Software ("FLOSS")) a MySQL kliensalkalmazásokat. Ha kereskedelmi alkalmazásban használjuk a MySQL Servert, akkor tanácsos ezt a két licencet tanulmányozni és ha szükséges, akkor akár szakértőhöz fordulni jogi tanácsért. Ha alkalmazásunk nem szolgál kereskedelmi célokat, akkor a MySQL Server, a GNU licencben foglaltak értelmében, szabadon használható.
- `my.ini`, `*.ini` - ez az állomány a MySQL Server beállítását meghatározó változókat definiálja és a 7. fejezetben részletesen foglalkozunk felépítésével. A többi `*.ini` állomány különböző példakonfigurációt tartalmaz.

A fent említett mappákkal és állományokkal a könyv folyamán többször fogunk találkozni. A következő szakaszban a `bin` mappában található `mysql.exe`, `mysqladmin.exe` és `mysqldump.exe` segédeszközökkel ismerkedünk meg.

2.2. A szerver segédeszközei: `mysql`, `mysqladmin` és `mysqldump`

A MySQL Server telepítésével konzolalkalmazás formájában három segédeszköz is a gépünkre került. Ezek a **`mysql`**, a **`mysqladmin`** és a **`mysqldump`** konzolalkalmazások. Ebben a szakaszban ezekkel a segédeszközökkel fogunk megismerkedni.

Kezdjük el egy előre beállított konzolalkalmazással, amelyből elérhetjük a MySQL szerveret! A **Start|Programs** menüponttal hívjuk elő

a programok listáját. A **MySQL** mappában megtaláljuk a **MySQL Server 5.0** mappát. Ebben a mappában rábukkanunk a **MySQL Command Line Client** ikonra. Ha erre az ikonra kattintunk, akkor megjelenik a MySQL konzol, ami nem más mint egy Windows konzol, és kéri, hogy jelentkezünk be:

```
Enter password:
```

Adjuk be a root felhasználó jelszavát, amit a MySQL Server telepítéskor megadtunk és nyomjuk meg az [Enter] billentyűt. A szerver a következő szöveggel jelzi, hogy készen áll utasítások elfogatására:

```
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 5.0.27-
community-nt
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the
buffer.
```

```
mysql>
```

A **STATUS** utasítással máris lekérdezhethetjük a szerver státusát:

```
mysql> STATUS
-----
C:\Program Files\MySQL\MySQL Server 5.0\bin\mysql.exe
Ver 14.12 Distrib 5.0.27, for Win32 (ia32)
Connection id:          1
Current database:
Current user:           root@localhost
SSL:                    Not in use
Using delimiter:       ;
Server version:        5.0.27-community-nt
Protocol version:      10
Connection:            localhost via TCP/IP
Server characterset:   utf8
Db characterset:       utf8
Client characterset:   utf8
Conn. characterset:    utf8
TCP port:              3306
Uptime:                16 min 11 sec

Threads: 1 Questions: 3 Slow queries: 0 Opens: 0
Flush tables: 1 Open tables: 6 Queries per second
avg: 0.001
-----
mysql>
```

A konzolt a quit utasítással zárjuk. Ezt a műveletet a továbbiakban sokszor fogjuk megismételni.

2.2.1. A mysql konzolalkalmazás

Amint az előbb láttuk, a **MySQL Command Line Client** konzolalkalmazás a **root** rendszergazda jelszavát várja és helyes jelszó megadása után csak a lokális MySQL Server-t kapcsolja. Ennek az a magyarázata, hogy a **MySQL Command Line Client** konzolalkalmazás indításával a `C:\Program Files\MySQL\MySQL Server 5.0\bin` mappában elhelyezett `mysql.exe` alkalmazást a következő paraméterekkel hívtuk meg:

```
mysql -u root -p
```

A `mysql.exe` alkalmazást, amit ezután **mysql** konzolalkalmazásnak vagy egyszerűen konzolnak fogunk nevezni, sokszor fogjuk használni és később többet fogunk megtudni róla. Egyelőre jegyezzük meg, hogy ha egy másik felhasználónévvel vagy nem egy lokális szerverre kívánunk bejelentkezni, akkor nem használhatjuk a **MySQL Command Line Client** konzolalkalmazást, hanem megfelelő paraméterek megadásával kell a **mysql** konzolt indítanunk.

A **mysql** konzolt egy Windows rendszerkonzolból a következő utasítással hívhatjuk meg

```
mysql [opciók] [adatbázis]
```

ahol az *opciók* a paramétereket, az *adatbázis* pedig a kapcsolandó adatbázist adja meg.

Ha a szerver telepítésénél kijelöltük az **Include Bin Directory in Windows PATH** jelölőnégyzetet (2.11 ábra), akkor a telepítőprogram hozzáadta a `C:\Program Files\MySQL\MySQL Server 5.0\bin` mappa elérési útját a **PATH** rendszerváltozóhoz. Ebben az esetben a `mysql` utasítással elérhetjük a **mysql** konzolt egy Windows konzolból. Ha azonban több szert, például különböző verziókat, telepítettünk a gépünkre, akkor a Windows konzolban beadott `mysql` utasítással csak az először telepített szert kapcsoljuk. Ezért mindenképpen előnyös ha elkészítünk magunknak egy egyszerű `.bat` állományt, a következő tartalommal:

```
"C:\Program Files\MySQL\MySQL Server  
x.y\bin\mysql.exe" ↵  
-u azonosító -p adatbázis
```

ahol *x.y* a szerver verziószáma, *azonosító* a felhasználó azonosítója és *adatbázis* a kapcsolni kívánt adatbázis. (Itt és a következőkben a ↵ jellel összekötött kódsorokat egyetlen egy kódsornak kell tekinteni.)

A paraméterek megadására általában két lehetőségünk van:

- `-p[]paraméterérték`, azaz egy `-` (kötőjel) karakterrel megelőzött betű, mely a paramétert azonosítja, és az ezt követő, a paraméterazonosítótól opcionálisan szóközzel elválasztott paraméterérték;
- `--paraméternév=paraméterérték`, azaz `--` (két kötőjel) karakterekkel megelőzött paraméternév, amit közvetlenül (szóköz nélkül) egy `=` (egyenlő) karakter és a paraméterérték követ.

A **mysql** konzol legfontosabb paraméterei a következők:

- `-u azonosító` vagy `-user=azonosító`: megadja a bejelentkező felhasználó azonosítóját. Ha még nincs felhasználó berendezve a szerverre vagy ha rendszergazdai feladatokat kívánunk végrehajtani, akkor a `root` azonosítóval kell bejelentkeznünk. Ha ezt a paramétert elhagyjuk, akkor a **mysql** konzol az ODBC azonosítót használja. Ez az azonosító azonban alapértelmezésben nincs a szerveren beállítva.
- `-p` vagy `--password=jelszó`: ezzel a paraméterrel jelezzük, hogy a bejelentkezés jelszóval történik és ezért válaszol a konzol a

Enter password:

felszólítással. Ha elhagyjuk a `-p` paramétert, a konzol jelszó nélkül próbál kapcsolatot teremteni a szerverrel, ami rendszerint kudarccal jár, ha a szerver biztonságosan van konfigurálva. Ha ezzel a paraméterrel megadjuk a jelszót: `--password=jelszó`, akkor az előbbi felszólítás elmarad. Nem tanácsos ezt a változatot használni, mivelhogy a jelszó kiolvasható a Windows folyamatok listájából (*process list*).

- `-h gazdagép` vagy `--host=gazdagép`: a **mysql** konzol egy másik gépen futó szerverrel is tud kommunikálni és ezzel a paraméterrel megadjuk a gazdagép (*host*, *hoszt*) *gazdagép* nevét vagy IP címét. Ha például a MySQL Server gazdagépe `mysql.myhost.com`, a kapcsolni kívánt adatbázis `mydatabase` és a felhasználói név `myusername`, akkor a következő paraméterekkel kell a **mysql** konzolt meghívunk:

```
mysql -h mysql.myhost.com --user= myusername
mydatabase -p
```


Ahhoz, hogy egy másik gépen futó szerveret elérhessünk, több feltételnek teljesülnie kell. Legelőször is a gazdagép a hálózaton elérhető kell legyen. Másodsor, a szerver TCP/IP portja (rendszerint 3306) is elérhető kell legyen. Ezt főleg tűzfal alkalmazásakor tanácsos ellenőrizni, hogy eleve elkerüljük a hosszas hibakeresést. Végül a szerver beállítása olyan kell legyen, hogy elfogadjon hálózati kapcsolatot. A 7. fejezetben többet fogunk megtudni ezzel kapcsolatban.

- `-P port` vagy `--port=port`: megadja a szerver portját, mely alapértelmezésben 3306.
- `--protocol=protokoll`: megadja a használandó protokollt. Ezt a paramétert ritkán kell használnunk, mivelhogy a **mysql** legtöbbször a helyes protokollt használja. Ha például megadjuk a `-h` paramétert, akkor a **mysql** automatikusan a `tcp` protokollt használja, azaz a kommunikáció TCP/IP csatornán történik. Ha nem használjuk a `-h` paramétert, akkor is az alapértelmezett protokoll a `tcp`, de ebben az esetben még két protokoll áll rendelkezésünkre: `pipe` és `memory`. Ezekről később még olvasni fogunk.

Az említett paramétereken kívül a **mysql** konzolalkalmazás sok más paramétert fogad el. (Ezekről majd a használatuk alkalmával leírjuk a legfontosabb tudnivalókat.) A `mysql -?`, `mysql -I` vagy `mysql --help` utasítással felsoroltathatjuk a **mysql** összes paramétereit. Érdeemes a

```
mysql -? >c:\mysql-help.txt
```

utasítással ezt a listát egy állományba elmenteni, hogy bármikor kéznél legyen.

Miután a **mysql** konzolban sikeresen bejelentkeztünk, az üdvözlő szöveg után a konzol a

```
mysql>
```

készenléti jellel (*prompt*, `prompt`) hozza tudomásunkra, hogy kész utasításokat a szerverre küldeni. A **mysql** konzolban gyakorlatilag mindent csinálhatunk, ami a MySQL Server-rel egyáltalán csinálható, de leggyakrabban SQL utasításokat küldünk a szerverre. Éppen ezért ez az eszköz nagyon hatékony, de nagyon veszélyes is: ha nem figyelünk oda egy egyszerű

```
DELETE FROM táblanév;
```

utasítással, minden figyelmeztetés nélkül kitörölhetünk egy nagy fáradtsággal és gonddal felépített táblát! A MySQL Server fejlesztői ilyen veszélyek elkerülésére bevezették a `-U`, `--i-am-a-dummy` vagy `--safe-updates` paramétert. Ha ezzel a paraméterrel indítjuk a **mysql** konzolt, akkor a **DELETE** vagy **UPDATE** utasításokat csak **WHERE** vagy **LIMIT** záradékkal hajthatunk végre. Az SQL utasításokkal az 5. fejezetben foglalkozunk részletesen. Itt csak annyit jegyezzünk meg, hogy az SQL utasításokat akár több sorban is írhatjuk és hogy `;` (pontosvessző) karakterrel kell lezárunk.

Ha egy hosszabb utasítást vagy több utasítást kívánunk elvégezni, akkor néha előnyös az utasításokat egy állományba írni és onnan a **mysql** konzolba beolvasni. Ilyenkor a **mysql** konzolt a következőképpen indítjuk:

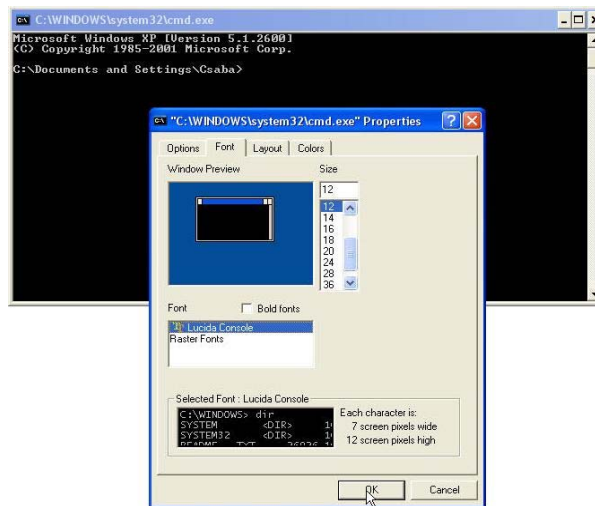
```
mysql [opciók] adatbázis < állomány.sql
```

ahol *állomány.sql* az SQL utasításokat tartalmazó állomány. Ez az állomány tetszőleges nevet viselhet és kiterjesztése sem kell, hogy *.sql* legyen.

A MySQL Server telepítésénél megadtunk egy alapértelmezett karakterkészletet (*character set*), például `utf8`. Ezen kívül egy egész sor más karakterkészletet, mint például a `latin1` (nyugat-európai nyelvek részére) vagy `latin2` (közép-európai nyelvek részére) is beállíthatunk a szerver alapértelmezett karakterkészletének. Sőt, amint ezt később látni fogjuk, adatbázis-, tábla- és oszlopszinten is definiálhatunk alapértelmezett karakterkészletet. Röviden: a szerveren az adatok a legkülönbözőbb karakterkészletek szerint lehetnek kódolva. A **mysql** konzol viszont egy DOS-kompatibilis kódlapot (*code page*), mint például 437 (amerikai), 850 (nyugat-európai) vagy 852 (közép-európai) stb., vesz igénybe. Ha a szerver által használt karakterkészlet és a konzol által használt kódlap nem „értik” meg egymást, akkor az úgynevezett különleges karakterek némelyike (például a magyar á, í, é, ó, ú, ö, ő, ü, ű vagy a német ä, ß stb.) hibásan lesz megjelenítve a konzolban és ezen túl fennáll a veszély, hogy hibásan kerül be az adatbázisba. Ezért gondoskodnunk kell arról, hogy a szerveren használt karakterkészletek és a **mysql** konzol kódlapja sikeresen együttműködjön és hogy a konzol egy olyan betűtípust (*font*) használjon, amelyik képes a különleges karaktereket megjeleníteni.

Nézzük meg, mit kell ehhez csinálnunk? Kezdjük el a helyes betűtípus (*font*) kiválasztásával!

Alapértelmezés szerint a Windows konzol a Raster Fonts betűtípust használja. Ez a betűtípus sajnos nem rendelkezik a különleges karakterekkel. Legelső dolgunk tehát az lesz, hogy megváltoztatjuk a Windows konzolban használt betűtípust Lucida Console-ra. Ezt a következőképpen tesszük: Indítsunk el egy Windows konzolt, majd kattintsunk az ablak címsávjában az ablak ikonjára! A legördülő menüből válasszuk ki a **Properties** menüparancsot. Ezzel megjelenik a **Properties** ablak. Kattintsunk a **Fonts** fülre, válasszuk ki a **Font** listából a Lucida Console bejegyzést és kattintsunk az **OK** gombra (2.17 ábra). A megjelenő ablakon két lehetőség között választhatunk két választógombbal. Az első választógomb felirata **Apply properties to current window only** és csak a jelen ablak részére állítja be a kiválasztott tulajdonságokat. A második választógomb felirata a konzol meghívásától függ. Ha a konzolt az **Start|Run** menüből a `cmd` utasítással indítottuk, akkor a felirat **Save properties for future windows with same title**, ha a konzolt egy parancsikkal (*shortcut*) indítottuk, akkor a felirat **Modify shortcut that started this window**. Mindkét esetben válasszuk ki a második választógombot. Ezzel a hasonlóképpen indított konzolok a beállított tulajdonsággal fognak megjelenni.



2.17 ábra: A betűtípus beállítása a Windows konzolban

Most már csak arról kell gondoskodnunk, hogy a konzol a megfelelő kódlapot használja. Ha még mindig nyitva van a `mysql` konzolunk, akkor megnézhetjük a `STATUS` utasítás eredményét, különben indítsuk még egyszer el a konzolt és nézzük meg a `STATUS` utasítással a szerver állapotát. Most minket csak a következő érdekel:

```

Server characterset:  utf8
Db characterset:      utf8
Client characterset:  utf8
Conn. characterset:   utf8

```

Az első két sor a szerver és az adatbázisok alapértelmezett karakterkészletét adja meg. Ez pillanatnyilag nem érdekes, mivelhogy, amint ezt nemsokára látni fogjuk, az adatbázisok tábláit és oszlopait az alapértelmezett karakterkészlettől eltérő karakterkészlettel láthatjuk el. A következő két sor a **mysql** konzol és a kapcsolat karakterkészletét adja meg. Ez a két adat érdekel minket, mivelhogy ennek a két karakterkészletnek kell az adatbázisok táblájával vagy oszlopaival kommunikálni. Természetes, ezek az adatok változókba vannak elmentve és a következő utasítással lekérdezhethetjük az értéküket (mindezeket az utasításokat és változókat később rendszeresen meg fogjuk ismerni):

```

mysql> SHOW VARIABLES LIKE '%_set_%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| character_set_client | utf8 |
| character_set_connection | utf8 |
| character_set_database | utf8 |
| character_set_filesystem | binary |
| character_set_results | utf8 |
| character_set_server | utf8 |
| character_set_system | utf8 |
| character_sets_dir | C:\Programme\...\share\charsets\ |
+-----+-----+
8 rows in set (0.00 sec)

```

A szerver válaszában felismerhetjük a

- character_set_server,
- character_set_database,
- character_set_client és
- character_set_connection

változókat, amelyek az utoljára beadott **STATUS** utasítás listájára utalnak. Ezen kívül a character_set_system és a character_set_results változókat látjuk. A character_set_system változó azt a karakterkészletet határozza meg, amivel a szerver a metaadatokat tárolja. Pillanatnyilag ez a változó minket nem érdekel. A character_set_results változó meghatározza, hogy milyen karakterkészletet használ a szerver, amikor visszaküldi a kliensalkalmazáshoz, esetünkben a **mysql** konzolhoz, a választ. Az eredmény helyes megjelenítése érdekében ennek a változónak az értéke nagyon fontos.

Ezek után nézzük meg, hogyan biztosíthatjuk az adatok helyes megjelenítését? Itt most csak azzal az esettel foglalkozunk, amikor a

MySQL táblák vagy oszlopok karakterkészlete `latin2` vagy `utf8`. Ebben az esetben a konzol kódlapja 852 vagy 65001 (UTF 8) kell legyen ahhoz, hogy a különleges karakterek helyesen jelenjenek meg a konzolban. Ezenkívül a szervernek is „tudnia” kell, hogy a konzol milyen kódlapot használ. Ha ez a két feltétel biztosítva van, akkor a szerver elvégzi a szükséges konverziókat és az adatok helyesen jönnek le a szervertől a konzolba, illetve helyesen kerülnek a konzolból fel a szerverre.

A Windows konzol által használt kódlapot a `chcp` utasítással állapíthatjuk meg:

```
>chcp
Active code page: 852
```

és a `chcp` *kódlap* utasítással állíthatjuk be:

```
>chcp 65001
Active code page: 65001
```

Ha a Windows konzol kódlapja nem 852 vagy 65001, akkor gondoskodunk kell a kódlap beállításáról és a kapcsolat által használt karakterkészletről. Vegyük sorra a lehetőségeket!

A jelen fejezetnek megfelelő mappában megtaláljuk a `huchr.sql` állományt. Ez az állomány azokat az SQL utasításokat tartalmazza, amelyekkel létrehozhatjuk a `huchr` adatbázist a `specchrs` táblával és a `latin2` és `utf8` oszlopokkal:

latin2	utf8
á	á
Á	Á
é	é
É	É
í	í
Í	Í
ó	ó
Ó	Ó
ö	ö
Ö	Ö
ő	ő
Ő	Ő
ú	ú
Ú	Ú
ű	ű
Ű	Ű

A latin2 oszlop a latin2 karakterkészlet szerint kódolt adatokat tartalmaz, az utf8 oszlop pedig utf8 kódolású karaktereket tartalmaz.

Egyelőre ne törődjünk az állomány tartalmával, később, miután megismertük az alapvető SQL utasításokat, megnézhetjük a bennfoglalt utasításokat. A huchr adatbázissal fogjuk ellenőrizni a **mysql** konzol beállítását.

Indítsunk el egy Windows konzolt és hívjuk meg a mysql.exe segédeszközt a következő paraméterekkel:

```
>mysql -u root -p <"C:\ MySQL.NET\2. fejezet\huchr.sql"
Enter password: *****
>
```

Ezzel létrehoztuk a huchr adatbázist. Most nézzük meg milyen kódlapot használ a Windows konzolunk:

```
>chcp
Active code page: 852
```

Indítsuk el a **mysql** konzolt, és nézzük meg a státusát:

```
>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or
\g.
Your MySQL connection id is 14 to server version:
5.0.27-community.nt

Type 'help;' or '\h' for help. Type '\c' to clear the
buffer.

mysql> STATUS
-----
mysql  Ver 14.12 Distrib 5.0.27, for Win32 (ia32)
Connection id:          14
Current database:
Current user:           root@localhost
SSL:                    Not in use
Using delimiter:       ;
Server version:        5.0.27-community.nt
Protocol version:      10
Connection:            localhost via TCP/IP
Server characterset:   utf8
Db      characterset:   utf8
Client characterset:  utf8
Conn.  characterset:  utf8
TCP port:              3306
Uptime:                2 hours 19 min 41 sec

Threads: 4  Questions: 151  Slow queries: 0  Opens: 0
Flush tables: 1  Open tables: 0  Queries per
second avg: 0.018
-----
```

Itt a félkövér szedéssel kiemelt két sor érdekelt. Amint látjuk, a kódlapok nem egyeznek meg: a Windows konzol a 852 kódlapot használja, míg a szerver az utf8 (Windows fogalmazásában a 65001) kódlapot tételezi fel és ezzel kommunikál a konzollal. Ezért ez eredmény nem megfelelő:

```
mysql> USE huchr;  
Database changed  
mysql> SELECT * FROM specchrs;  
+-----+-----+  
| latin2 | utf8 |  
+-----+-----+  
| í      | í    |  
| ü      | ü    |  
| é      | é    |  
| ë      | ë    |  
| ð      | ð    |  
| ñ      | ñ    |  
| ó      | ó    |  
| â      | â    |  
| ï      | ï    |  
| ð      | ð    |  
| é      | é    |  
| ù      | ù    |  
| ù      | ù    |  
| ù      | ù    |  
| ù      | ù    |  
+-----+-----+  
18 rows in set (0.00 sec)
```

Az első utasítással (**USE** huchr;) közöljük a szerverrel, hogy a huchr adatbázissal kívánunk a továbbiakban dolgozni, a másodikkal (**SELECT** * **FROM** specchrs;) pedig lekérdezzük a specchrs tábla adatait. A visszakapott eredmény, a különböző kódlapok miatt nem használható.

Zárjuk a kapcsolatot a quit utasítással, állítsuk be a Windows konzol kódlapját 65001-re és ismételjük meg a kísérletet:

```
mysql> quit  
Bye  
  
>chcp 65001  
Active code page: 65001  
  
>mysql -u root -p  
Enter password: *****  
Welcome to the MySQL monitor.  Commands end with ; or  
\g.  
Your MySQL connection id is 15 to server version:  
5.0.27-community.nt
```

```

Type 'help;' or '\h' for help. Type '\c' to clear the
buffer.

mysql> STATUS
-----
mysql Ver 14.12 Distrib 5.0.27, for Win32 (ia32)
...
Client characterset:  utf8
Conn.  characterset:  utf8
...
-----

mysql> USE huchr;
Database changed
mysql> SELECT * FROM specchrs;
+-----+-----+
| latin2 | utf8 |
+-----+-----+
| á      | á    |
| Ā      | Ā    |
| é      | é    |
| É      | É    |
| í      | í    |
| Í      | Í    |
| ó      | ó    |
| Ó      | Ó    |
| ö      | ö    |
| Ö      | Ö    |
| ó      | ó    |
| Ő      | Ő    |
| ú      | ú    |
| Ű      | Ű    |
| ü      | ü    |
| Ū      | Ū    |
+-----+-----+
18 rows in set (0.00 sec)

```

Ezúttal a különleges karakterek helyesen jelennek meg.

Ha ezzel a beállítással kívánunk dolgozni, akkor érdemes egy *.bat állománnyal (például mysql65001.bat) a **mysql** konzolt elindítani:

```
cmd /k "chcp 65001&&mysql -u root -p"
```

A cmd utasítás /k paramétere biztosítja, hogy az idézőjelekben foglalt utasítások elvégzése után a konzol ablaka nyitva marad. Az idézőjelekben, && karaktersorral elválasztva, két utasítást adunk meg: az első beállítja a 65001 kódlapot, a második meghívja a **mysql** konzolt.

Ha valamilyen oknál fogva nem kívánjuk a 65001 kódlapot használni és ragaszkodunk a 852 kódlaphoz, akkor két lehetőségünk van arra, hogy a különleges karakterek helyesen jelenjenek meg.

Első lehetőségként, a **mysql** konzol meghívása után, tudomására kell hoznunk a szervernek, hogy a 852 kódlapnak megfelelően kell az adatokat a konzolhoz küldenie. Ezt a **SET NAMES cp852;** utasítással végezzük el:

```
>chcp
Active code page: 852

>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or
\g.
Your MySQL connection id is 19 to server version:
5.0.27-community.nt

Type 'help;' or '\h' for help. Type '\c' to clear the
buffer.

mysql> STATUS
-----
mysql  Ver 14.12 Distrib 5.0.27, for Win32 (ia32)
...
Client characterset:  utf8
Conn.  characterset:  utf8
...
-----
mysql> SET NAMES cp852;
Query OK, 0 rows affected (0.00 sec)
mysql> STATUS
-----
mysql  Ver 14.12 Distrib 5.0.27, for Win32 (ia32)
...
Client characterset:  cp852
Conn.  characterset:  cp852
...
-----

mysql> USE huchr;
Database changed
mysql> SELECT * FROM specchrs;
+-----+-----+
| latin2 | utf8 |
+-----+-----+
| á      | á    |
| Á      | Á    |
| é      | é    |
| É      | É    |
| í      | í    |
+-----+-----+
```

```

| Í | | |
| Ó | | |
| Ö | | |
| ö | | |
| Ő | | |
| ő | | |
| Ú | | |
| ú | | |
| Ű | | |
| ű | | |
| Ű | | |
+---+---+
18 rows in set (0.00 sec)

```

Vegyük észre, hogy a **SET NAMES** *cp852*; utasítás végrehajtása után a státus lekérdezésére kapott válasz az új beállításnak felel meg. Az idevonatkozó sorokat félkövér szedéssel kiemeltük. A **SET NAMES** *kódlap*; utasítás egyébként egyenértékű a következő három utasítással

```

mysql> SET character_set_client = kódlap;
mysql> SET character_set_results = kódlap;
mysql> SET character_set_connection = kódlap;

```

Második lehetőségünk abban áll, hogy a **mysql** konzolalkalmazást eleve olyan paraméterekkel indítjuk, amelyek biztosítják a különleges karakterek megjelenítésére szükséges beállításokat:

```

>mysql -u root --character-sets-dir=
"C:\Program Files\MySQL\MySQL Server
5.0\share\charsets"
--default-character-set=cp852 -p

```

A `--character-sets-dir` paraméter a karakterkészletek definícióját tartalmazó mappát adja meg, míg a `--default-character-set` a kívánt karakterkészletet határozza meg.

Nézzük meg, miként működik ez a módszer!

```

>chcp
Active code page: 852

>mysql -u root --character-sets-dir=
"C:\Program Files\MySQL\MySQL Server 5.0\share\charsets"
--default-character-set=cp852 -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20 to server version: 5.0.27-
community.nt

```

```
Type 'help;' or '\h' for help. Type '\c' to clear the
buffer.
```

```
mysql> STATUS
```

```
-----
```

```
...
```

```
Client character set: cp852
```

```
Conn. character set: cp852
```

```
...
```

```
-----
```

```
mysql> USE huchr;
```

```
Database changed
```

```
mysql> SELECT * FROM specchrs;
```

latin2	utf8
á	á
Ā	Ā
ē	ē
Ē	Ē
í	í
Ī	Ī
ó	ó
Ō	Ō
ö	ö
Ö	Ö
ó	ó
Ŏ	Ŏ
ú	ú
Ū	Ū
ü	ü
Ū	Ū
ű	ű
Ű	Ű

```
18 rows in set (0.00 sec)
```

Ebben az esetben, amint ezt a **STATUS** utasítás eredménye tanúsítja, a szükséges karakterkészlet elejétől fogva be van állítva.

Ha ezzel a beállítással kívánunk dolgozni, akkor ismét érdemes egy *.bat állománnyal (például mysql852.bat) a **mysql** konzolt elindítani:

```
mysql -u root --character-sets-dir=↵
"C:\Program Files\MySQL\MySQL Server
5.0\share\charsets" ↵
--default-character-set=cp852 -p
```

Befejezésül, tisztázzunk még egy fontos dolgot! Miért tételezi fel a szerver, hogy a kliens karakterkészlete utf8? Ha újból elindítjuk a mysql konzolt és lekérdezzük a `character_set_client` változó értékét

```
mysql> SHOW VARIABLES LIKE 'character_set_client';
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| character_set_client | utf8  |
+-----+-----+
1 row in set (0.00 sec)
```

akkor az eredmény utf8. Honnan veszi a szerver ezt az értéket?

A `my.ini` konfigurációs állománnyal már találkoztuk és később részletesen foglalkozunk majd vele. Ezt a szöveges állományt a telepítő program hozta létre. Bármilyen ASCII szerkesztővel, például a **Jegyzettömbbel (Notepad)** megnézhetjük. Az állomány az `*.ini` állományoknál szokásos szakaszokra van felosztva. Minket most a `[mysql]` szakasz érdekel. Ha megnézzük az állomány tartalmát, akkor ebben a szakaszban a következőket találjuk:

```
[mysql]
default-character-set=utf8
```

A `default-character-set` startparaméter értéke meghatározza a `character_set_client` rendszerváltozó értékét. Ha a `default-character-set` indítóparaméter értékét a Windows konzol kódlapjára állítjuk be, akkor a `character_set_client` rendszerváltozó automatikusan a különleges karakterek helyes megjelenítésére szükséges értéket kapja. Módosítsuk a `[mysql]` szakasz a következőképpen:

```
[mysql]
character-sets-dir=↵
"C:/Program Files/MySQL/MySQL Server
5.0/share/charsets"
default-character-set=cp852
```

A `character-sets-dir` indítóparaméterrel megadjuk a karakterkészleteket definiáló mappa elérési útját, a `default-character-set` startparaméterrel megadjuk a konzol kódlapját. Ügyeljünk arra, hogy a mappa elérési útjában a Unix/Linux típusú jelölést használjuk (a MySQL Server számára a `\` karakter vezérlőkarakter (*escape character*)).

Ha most elindítjuk a **mysql** konzolt és lekérdezzük a `hchr` adatbázis `specchr` táblájának adatait, akkor a kívánt eredményt kapjuk.

Következésképpen, ahhoz, hogy a különleges karakterek helyesen jelenjenek meg a **mysql** konzolban, ha a szerver a `latin2` vagy az `utf8` karakterkészletet használja, akkor a Windows konzolnak, és így a **mysql** konzolnak is, a `65001` vagy a `852` kódlapot kell használnia és ezt valamilyen módon tudatnunk kell a szerverrel. Ha ezek a feltételek teljesültek, a szerver elvégzi a szükséges kódkonverziót, ami biztosítja az adatok helyes közlekedését mindét irányba.

2.2.2. A **mysqladmin** és a **mysqldump** segédeszközök

A **mysqladmin** és a **mysqldump** segédeszközökkel később még foglalkozni fogunk. Ebben az alszakaszban röviden ismertetjük ezeket a segédeszközöket. Mindkét segédeszköz konzolalkalmazás, amit a MySQL alapmappában található `bin` mappából indíthatunk a `mysqladmin.exe`, illetve a `mysqldump.exe` végrehajtható állománnyal. Mivelhogy a Windows `PATH` rendszerváltozóba felvettük a `bin` mappa elérési útját, ezeket az alkalmazásokat parancssorból indíthatjuk.

A **mysqladmin** segédeszközt felügyeleti feladatok elvégzésére használjuk, például új adatbázis létrehozására, adatbázis törlésére vagy jelszó megváltoztatására. Szintaxisa a következő:

```
>mysqladmin [opciók] utasítás
```

A **mysqladmin** opciói ugyanazok mint a **mysql** konzol opciói: `-u`, `-p`, `-h`, stb. Az *utasítás* meghatározza a végrehajtandó feladatot. A **mysqladmin** és más segédeszközzel a 3. fejezetben fogunk megismerkedni. Itt csak a következő két utasítást említjük meg:

- új adatbázis létrehozása: `create adatbázis`

```
>mysqladmin -u root -p create newdb
Enter password: *****
```

```
>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 21 to server
version: 5.0.27-community.nt
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the
buffer.
```

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| huchr |
| myimages |
| mysql |
| mysqlbooks |
| newdb |
| northwind |
| test |
| ttmysqlbooks |
+-----+
9 rows in set (0.02 sec)
```

- adatbázis törlése: drop *adatbázis*

```
> mysqladmin -u root -p drop newdb
Enter password: *****
Dropping the database is potentially a very bad thing to
do.
Any data stored in the database will be destroyed.

Do you really want to drop the 'newdb' database [y/N] y
Database "newdb" dropped
```

```
>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 23 to server version: 5.0.27-
community.nt
```

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| huchr |
| myimages |
| mysql |
| mysqlbooks |
| northwind |
| test |
| ttmysqlbooks |
+-----+
8 rows in set (0.00 sec)
```