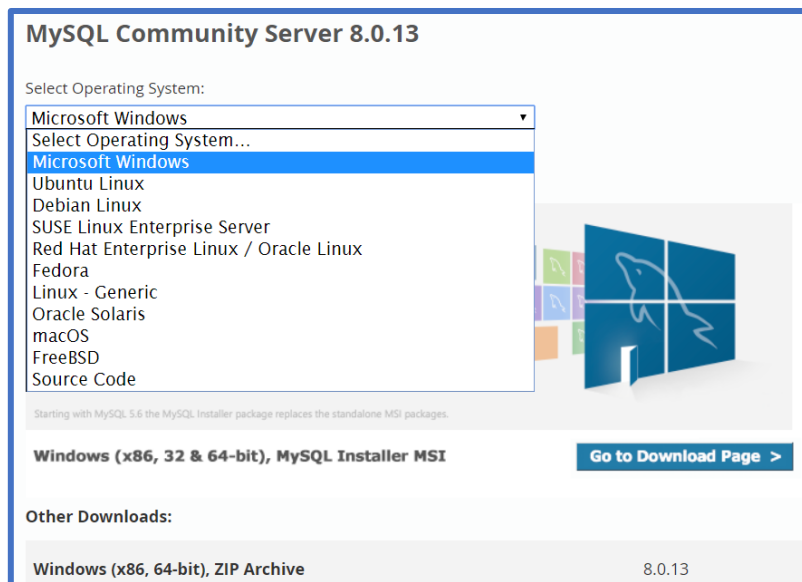


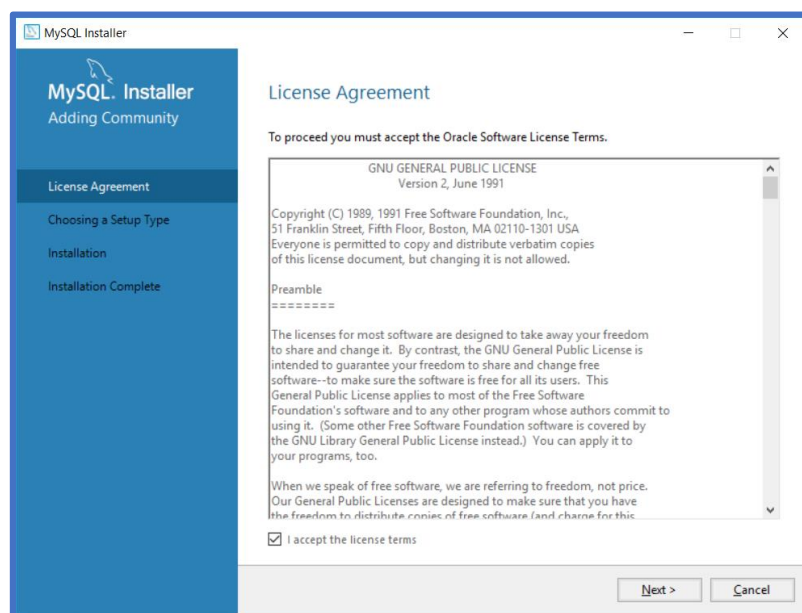
Bevezetés - A munkakörnyezet kialakítása

A gyakorlatok többségén a MySQL¹ adatbáziskezelő rendszert használjuk. A MySQL jelenleg a Oracle tulajdonában van, létezik kereskedelmi és GPL-es verziója is. A MySQL Community Server különböző verziói letölthető a <https://dev.mysql.com/downloads/mysql/> oldalról.



1. ábra MySQL letöltése

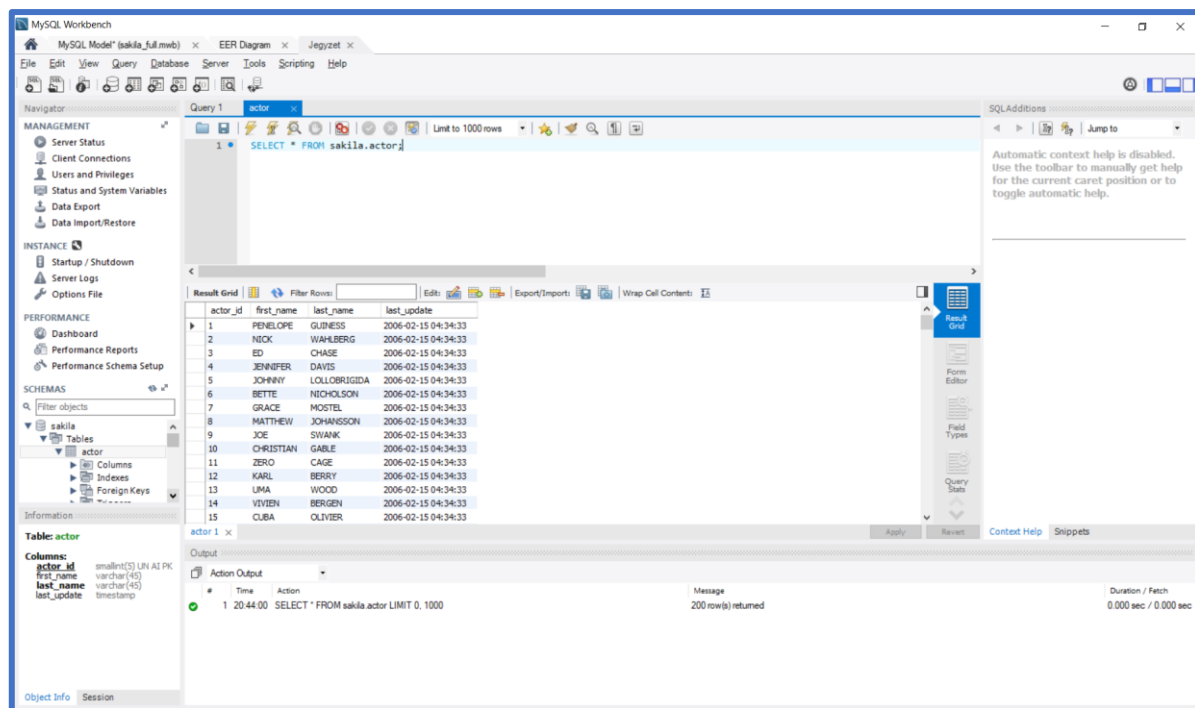
A szoftver letöltése után szokásos telepítési folyamaton kell végig menni. A részletes menetet a https://www.youtube.com/watch?v=fy_cKks4cDM megtekinthető.



2. ábra MySQL telepítése

¹ <https://www.mysql.com/>

Amennyiben Setup Type ablaknál a „Developer Default”-t választjuk, akkor többek között egy adminisztrációs segédsoftver a MySQL Workbench is telepítésre kerül. Ez egy nagyon hasznos segédeszköz, de a tanulás során azt tartjuk szem előtt, hogy az elsajátított ismeretek ne csak egy adott környezetben lehessen használni, hanem minél szélesebb körben alkalmazható tudást sajátítsunk el. Így nem lehet cél egy segédprogram használatának megismerése még akkor sem, ha az a későbbiekben (pl. szakdolgozat készítésekor) megkönnyíti és meggyorsítja az informatikus munkáját.



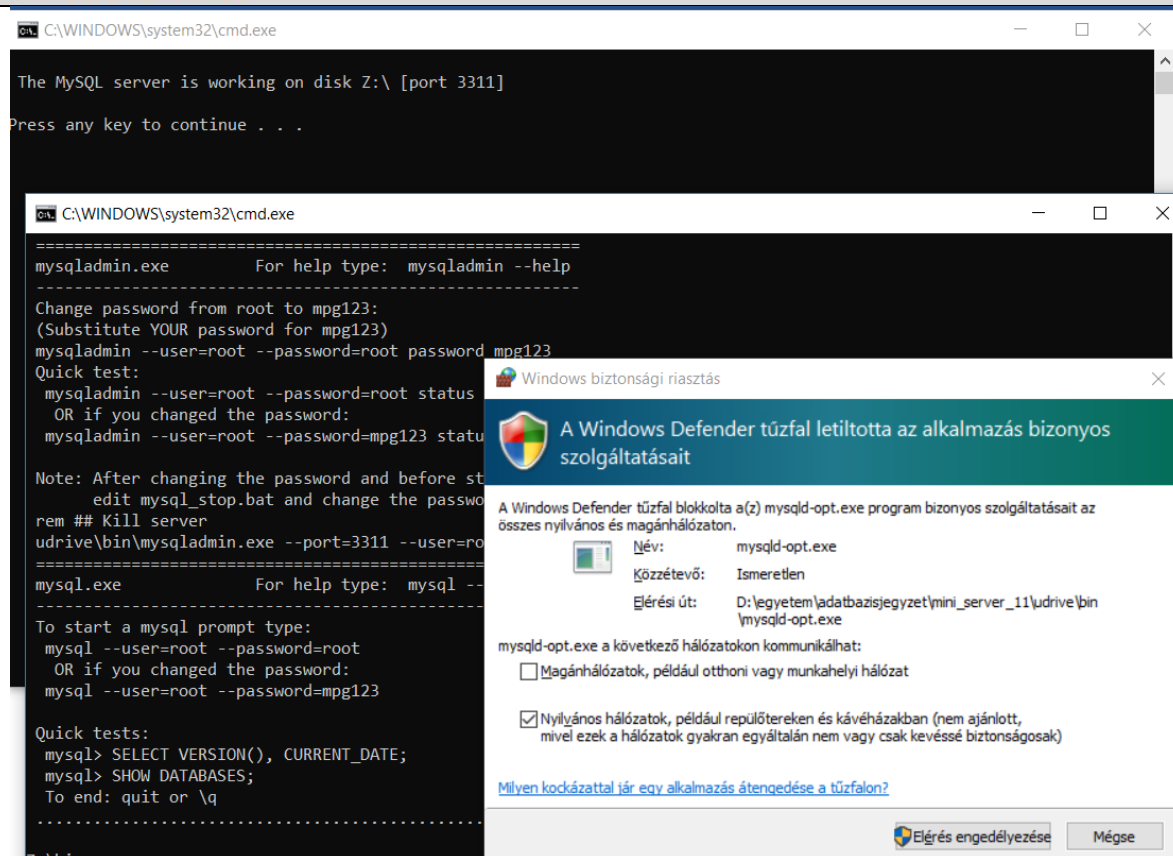
3. ábra MySQL Workbench

Az előbbieket és az önálló gyakorlás szem előtt tartva használjunk egy portable verziót. Ezt könnyen tudjuk telepíteni bármilyen helyzetben. A MySQL mini server letöltése a https://sourceforge.net/projects/miniserver/files/MiniServer/MiniServer_%20MySQL%205.0.67%20Portable/ oldalról lehetséges. Itt tömörített zip, illetve önkicsomagolásos formában exe-ként találhatjuk meg.

Home / MiniServer / MiniServer_MySQL 5.0.67 Portable			
Name	Modified	Size	Downloads / Week
Parent folder			
mini_server_11.zip	2008-10-05	4.0 MB	67
mini_server_11.exe	2008-10-05	2.0 MB	14

4. ábra MySQL Mini Server

Letöltés után csomagoljuk ki egy könyvtárban, ahol dolgozunk a továbbiakban (pl.:d:\munka\mysql). Ezek után indítsuk el a mysql_start.bat állományt. Ezzel indíthatjuk el a szerverünket és a mysql_stop.bat állománnyal pedig leállíthatjuk amennyiben befejeztük a munkát.



5. ábra MySQL Mini Server indítása

Első indításkor 3 ablak jelenik meg. A Windows Defender ablak engedélyt kér a szerver futtatáshoz. Tapasztalat alapján a szolgáltatás akkor is működni fog, ha nem engedélyezzük. (Ha nincs rendszergazdai jogunk a használni kívánt géphez, akkor ez hasznos.) A képen a leghátsó ablakban láthatjuk, hogy létrejön egy Z meghajtó, illetve az, hogy a szerverünk a 3311-es portot fogja használni. A Z meghajtóhoz a MySQL mappánk udrive mappája lett rendelve. A port eltér a MySQL alap 3306-s portjától, amennyiben ezt tesztelni szeretnénk szabni akkor a udrive mappabában lévő my.conf állományban megtehetjük. A másik parancssoros ablakban információkat találunk a szerver használatához, gyors teszteléséhez. Teljes jogosultsággal a *root* felhasználóval és *root* jelszóval léphetünk be. Ekkor a szerverünk konzolján találjuk magunkat.

```
Z:\bin>mysql --user=root --password=root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.67-community-nt MySQL Community Edition (GPL)

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

mysql>
```

6. ábra MySQL konzol

Hozzunk létre egy adatbázist, amiben az első alkalommal fogunk dolgozni (create database ora01).

```
mysql> create database ora01;  
Query OK, 1 row affected (0.01 sec)  
  
mysql>
```

7. ábra Adatbázis létrehozása

Ellenőrizzük a létrehozott adatbázisunkat (show databases).

```
mysql> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| ora01 |  
+-----+  
3 rows in set (0.00 sec)
```

8. ábra Adatbázisok megjelenítése

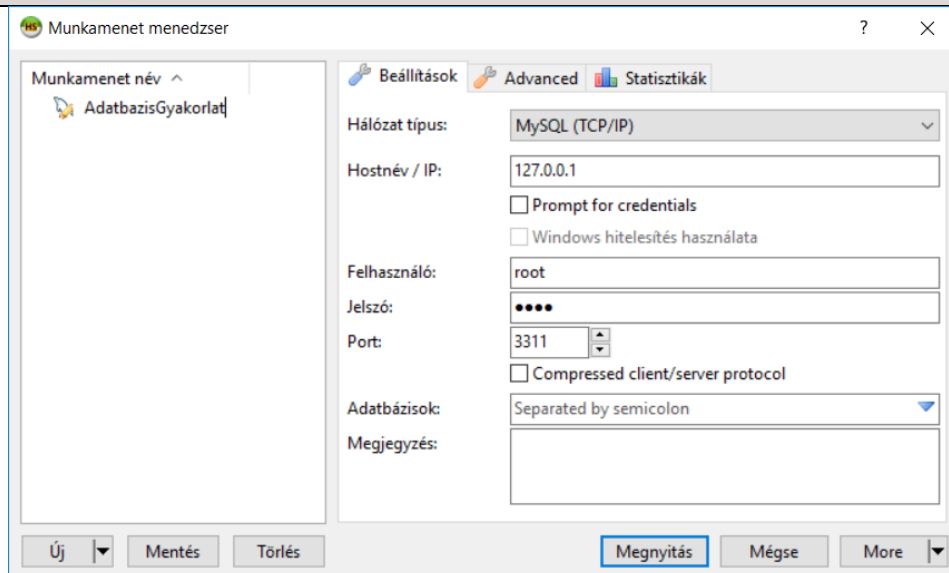
Minden parancsot „;”-vel kell lezárunk. Amennyiben ezt nem tesszük meg, akkor új sorban folytathatjuk a parancsunk gépelését. Így akár pótolhatjuk is az elmaradt parancsvége jelet.

```
mysql> select current_date  
-> ;  
+-----+  
| current_date |  
+-----+  
| 2018-11-03 |  
+-----+  
1 row in set (0.00 sec)  
  
mysql>
```

9. ábra Többsoros parancsbevitel

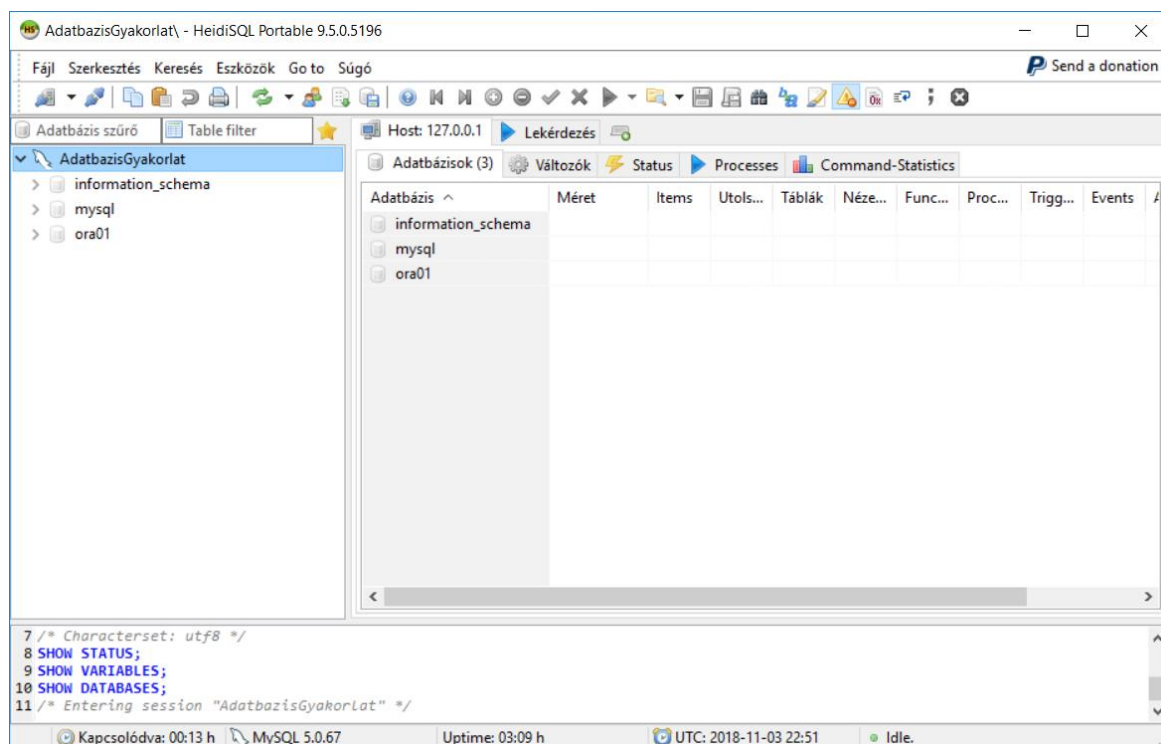
Már ennyi gépelésből az látszik, hogy nem egy barátságos felület itt parancsokat kiadni, tesztelni, tanulni. Használjunk egy olyan felületet, ami „barátságosabb” a tanuláshoz, teszteléshez. Az egyik ilyen ingyenes kliens a HeidiSQL (<https://www.heidisql.com/>). Itt is található portable verzió, amit a szoftver honlapjáról letölthetők. Ezt is csomagoljuk ki a munka mappánkba a mini szerver mellé. Ezt az adminisztrációs szoftvert lehet használni MariaDB, MS SQL, PostgreSQL esetén is.

Az alkalmazás első indításakor meg kell adni a bejelentkezési munkamenet adatait (root, root, 3311).



10. ábra HeidiSQL munkamenet

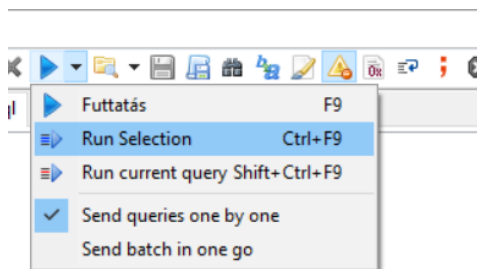
Ezt csak egyszer kell megadnunk, ha elmentjük a munkamenetünket (AdatbázisGyakorlat). A bejelentkezés után következő ablak fogad bennünket.



11. ábra HeidiSQL nyitóablak

Ekkor láthatjuk a konzolon létrehozott adatbázisunkat. A program vizuálisan is sok segítséget, információt ad nekünk, de mi igazából csak az utolsó „Lekérdezés” fület fogjuk használni. Itt gépeljük be a lekérdezéseinket és kapuk meg a lekérdezéseink kimenetét. Minden órai gyakorlást érdemes külön-külön elmenteni: ora01.sql, ora02.sql, stb. A lekérdezéseinket három módon futtathatjuk le. Lefuttathatjuk az állomány összes parancsát (F9) vagy a kijelölt parancso(ka)t (Ctrl+F9), illetve az aktuális parancsot

(Shift+Ctrl+F9). Ha szeretnénk a gyorsgombokat megváltoztatni, akkor ezt az Eszközök/Beállítások alatt megtehetjük.



12. ábra HeidiSQL lekérdezések futtatása

Adjuk ki a *use* parancsot. Ezzel kijelölhetjük, hogy melyik adatbázissal szeretnénk dolgozni.

```
1. use ora01;
```

A HeidiSQL alsó sávjában láthatjuk, hogy milyen parancsot adtunk ki és ennek milyen hatása volt. Az első megállapításunk az lehet az alsó sávban látott parancsokból következően, hogy az SQL nem különbözteti meg a nagy és kisbetűt.

CREATE TABLE

Példa

Hozzon létre egy kurzus adattáblát a tanfolyam adatbázisban, a mezők között szerepeljen a megnevezés, létszám adatok, kezdés ideje és a díjadatok.

A feladat leírása pontosabban táblázatos formában.

Kurzus		
Id	szöveges	A kurzus azonosítója
Megnevezes	szöveges	A kurzus megnevezése
Min_letszam	egész	A kurzus minimális létszáma
Max_letszam	egész	A kurzus maximális létszáma
Kezdes_ideje	dátum	A kurzus indítási ideje
Dij	egész	A kurzus díja

Megoldás

```
1. CREATE TABLE Kurzus (
2.     Id VARCHAR(8) NOT NULL,
3.     Megnevezes VARCHAR(30) NOT NULL,
4.     Min_letszam INT(2) NOT NULL,
5.     Max_letszam INT(3) NOT NULL,
6.     Kezdes_ideje DATE NOT NULL,
7.     Dij INT(6) NOT NULL,
8.     PRIMARY KEY (Id)
9. );
```

Vegyük végig (keressük meg önállóan) a parancs egyes részeit: NOT NULL, PRIMARY KEY.

Vigyünk fel adatokat az előbbi táblába!

INSERT INTO

```
1. INSERT INTO Kurzus
2.   VALUES('12345678','PLC programozása', 10,20, '2018.10.12', 200000);
3. INSERT INTO Kurzus
4.   VALUES('12345678','HMI programozása',10,18,NULL,200000);
```

Ha szemügyre vesszük a megoldást, akkor több megállapítást tehetünk. Látható, hogy az oszlopok sorrendje meghatározza a bevittet, ha ettől el akarunk térni, akkor a tábla neve után meg kell adnunk az oszlopok nevét és ez alapján kell megadni a sor adatait. Kísérletezzünk bátran az adatbevitellel. Ne ijedjünk meg a hibáktól. Azért jó a portable változatot használni, hiszen így bármit bátran kipróbálhatunk. Például a második megoldásnál is kapunk egy hibaüzenetet. Értelmezzük ezt és javítsuk ki a hibát.

```
1. INSERT INTO Kurzus
2.   VALUES('11111111','CNC programozása',10,18,NULL,200000),
3.   ('2222222','CAD',10,18,NULL,200000);
4. SELECT * FROM Kurzus;
```

Adattárolási módok

MySQL estén több tárolási mód létezik, ezek befolyásolják a kiadható parancsok körét. A szerver beállításainál adhatjuk meg, hogy melyik legyen az alapértelmezett tárolási mód. A *myISAM* adattáblák egy egyszerűbb tárolási módja, ami szűkebb lehetőséget nyújt, de sokszor elegendő, illetve fontos szempont a gyorsasága. Az *innoDB* összetett tárolási mód, bővebb lehetőségeink vannak: a tranzakció kezelés, a rekord szintű zárolás, az idegen kulcs, megkötés érvényesítése. Hátránya, hogy lassabb, de sokkal több funkcionalitást nyújt. A *memory table* a memóriában tárolt adattáblák lehetőségét biztosítja. Ideiglenes adatok kezelésére megfelelő.

Adattípusok

Az egyes adatbáziskezelő rendszerek esetén különböző adattípusokat használhatunk. A MySQL használat esetén ezek a következők.

Numerikus típusok

A numerikus adattípusoknál az INTEGER (egész) két féle lehet: SIGNED vagy UNSIGNED (előjeles vagy előjel nélküli). Ha az adattípusunk előjeles, akkor negatív számokat is tárolhatunk benne, ha pedig előjel nélküli, akkor kétszer akkora számokat tárolhatunk benne, mint előjeles változatában. Ha a táblák készítésekor nem mondjuk meg, hogy milyen legyen, előjeles lesz.

A főbb egész adattípusok

- TINYINT – 1 byte
- SMALLINT – 2 byte
- MEDIUMINT – 3 byte
- INT – 4 byte
- BIGINT – 8 byte.

Használható a SERIAL rövidítés is, ami a BIGINT UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE rövidítése. Ez a gyakori használat miatt indokolt. Ugyanilyen álnévnek tekinthető a

BOOL, BOOLEAN is, amivel megvalósíthatjuk a logikai adattípust. Itt a háttérben egy TINYINT(1) valósul meg, amiben 0,1 érték tárolódik el a logikai true, false-nak megfelelően.

Valós adattípusok

- FLOAT
- DOUBLE
- DECIMAL

Dátum és idő adattípusok

- DATE
- DATETIME
- TIMESTAMP
- TIME
- YEAR

Szöveges adattípusok

- CHAR
- VARCHAR
- BLOB
- BIT
- TEXT
- ENUM
- SET

CHAR

CHAR(méret) fix hosszúságú karakterláncot tartalmazhat. Minden esetben adott hosszúságot tárol, azaz, ha nem használjuk fel a teljes méretet, akkor szóközökkel tölti fel az adott szélességig. Ez gyorsítja a keresést, de a tárolás pazarló.

VARCHAR

VARCHAR(méret) változó hosszúságú karakterláncot tartalmazhat. Ez csak annyi helyet foglal, amit az aktuális adat megkövetel. Így a tárolás takarékosabb, a keresés lassabb mint a CHAR esetén.

BLOB

Bináris adatok tárolását teszi lehetővé.

BIT

1-64 bit hosszúságú bináris adat tárolását teszi lehetővé. (Megjegyzés: BIT=BIT(1))

TEXT

Nagy mennyiségű (max 65535 karakter) szöveges információ tárolását teszi lehetővé.

ENUM

Lehetőségünk van olyan értékhalmozot is megadni, amik közül vihetünk be értéket az adattáblába. Az értékek felsorolásakor a sorrend fontos, indexekkel látja el a rendszer, így ezt felhasználva matematikai (átlag, összeg) függvényeket is használhatunk.

Példa

```
1. CREATE TABLE ruhak (
2.   megnevezés VARCHAR(40),
3.   meret ENUM('x-small', 'small', 'medium', 'large', 'x-large')
4. );
```

SET


Amennyiben egy adott halmazból származó információkat akarunk eltárolni akár többet is, akkor ezt megadhatjuk a táblalétrehozáshoz a SET kulcsszó segítségével. Példa SET-re.

```
1. CREATE TABLE jegyek(
2.   jegyAz INT PRIMARY KEY AUTO_INCREMENT,
3.   hallgatoAz INT,
4.   datum DATE,
5.   jegy SET("elégtelen", "elégséges", "közepes", "jó", "jeles")
6. );
```

```
1. INSERT INTO jegyek VALUES(null,1,"2018.11.11","jó");
2. INSERT INTO jegyek VALUES(null,2,"2018.11.11","jó,jeles");
3. INSERT INTO jegyek VALUES(null,2,"2018.11.11","jó,jeles,1");
4. INSERT INTO jegyek VALUES(null,3,"2018.11.11","4,jó,jeles,1");
5. SELECT * FROM jegyek;
```

Amennyiben nem a halmaznak megfelelő értékeket adunk meg az adatbevitelnél, akkor figyelmeztetést kapunk, illetve a rendszer csonkolja a bevitelt, amit a képen jól látható.

AdatbázisGyakorlat: Figyelmeztetés

 Your query produced 1 warnings.

Warnings from last query:
Warning: Data truncated for column 'jegy' at row 1

jegyek (4x4)

jegyAz	hallgatoAz	datum	jegy
1	1	2018-11-11	jó
2	2	2018-11-11	jó,jeles
3	2	2018-11-11	jó,jeles
4	3	2018-11-11	jó,jeles

Megjegyzés: Az INSERT INTO-t más alakban is használhatjuk. Lehetőség van akár SELECT kimenetét egy létező adattáblába irányítani.