# Adatbázis motor

## 1. SQLite bemutatása

Az SQLite egy beágyazott (embedded) relációs adatbázis kezelő, azaz az őt használó szoftverhez linkelve lehet használni, nem egy külön SQL szerver. Nincs user adatbázisa. Ettől eltekintve nagyrészt megvalósítja az SQL-92 szabványt. Olyan esetekben érdemes megfontolni az alkalmazását, amikor a helyi rendszeren szeretnénk az adatainkat tárolni. Az SQLite-ot neves szoftverek és eszközök használják: MAC OS, Solaris 10, Skype, iPhone, Firefox. A fejlesztők szlogenje szerint használjuk ott ezt a könyvtárat, ahol egyébként a fopen() parancsot használnánk. Gondoljunk arra, hogy egy hordozható eszközhöz írt szoftvernek általában pont ilyen adatbázis kezelőre van szüksége.

Adatbázisok esetén az ACID mozaikszó az Atomicity (atomicitás), Consistency (konzisztencia), Isolation (izoláció), és Durability (tartósság) rövidítése. Ezek az adatbázis kezelő rendszer tranzakció feldolgozó képességeinek alapelemei. Enélkül az adatbázis integritása nem garantálható, így a tranzakció kezelés támogatott ebben a környezetben is.

Részlegesen megvalósítja a triggereket és a legtöbb komplex/összetett lekérdezést. Az SQLite szokatlan típuskezelést használ az SQL adatbázis-kezelőhöz: egy adattípus nem a táblaoszlopaihoz, hanem egyedi értékekhez van hozzárendelve, más szóval dinamikus típuskezelést használ, gyengén típusos adatkezelés mellett. Amennyiben string típusú adat beilleszthető integer oszlopba, akkor a SQLite először a stringet integerre konvertálja, ha az oszlop preferált típusa integer. Ez nagyobb rugalmasságot ad az oszlopoknak, ami hasznos lehet dinamikus típuskezelésű script-nyelvekben való alkalmazás esetén, azonban ez a technika nem vihető át más SQL adatbázis-kezelőkbe. Az SQLite képtelen atipikus adatbázisokban található szigorúan típusos oszlopok kezelésére.

Ugyanazt az adatbázist több processz és szál használhatja egyidejűleg problémamentesen. Az olvasási kérelmek kiszolgálása párhuzamosan történik. Az írási kérelmek végrehajtása akkor történik meg, amikor nincs folyamatban más kérelem kiszolgálása, egyébként az írási kérelem sikertelen lesz és hibakóddal tér vissza, illetve lehetőség van egy beállítható várakozási időelteltével a kérelem ismétlésére. Ez a konkurens hozzáférési állapot megváltozható ideiglenes táblák használata esetén.

Az SQLite számos programnyelvből használható, így BASIC, C, C++, Common Lisp, Java,C#, Visual Basic .NET, Delphi, Curl, Lua, Tcl,R, PHP, Perl, Ruby, Objective-C (Mac OS Xen), Python, newLisp, Haskell, OCaml, Smalltalk és Scheme nyelvekhez rendelkezik illesztő felülettel.

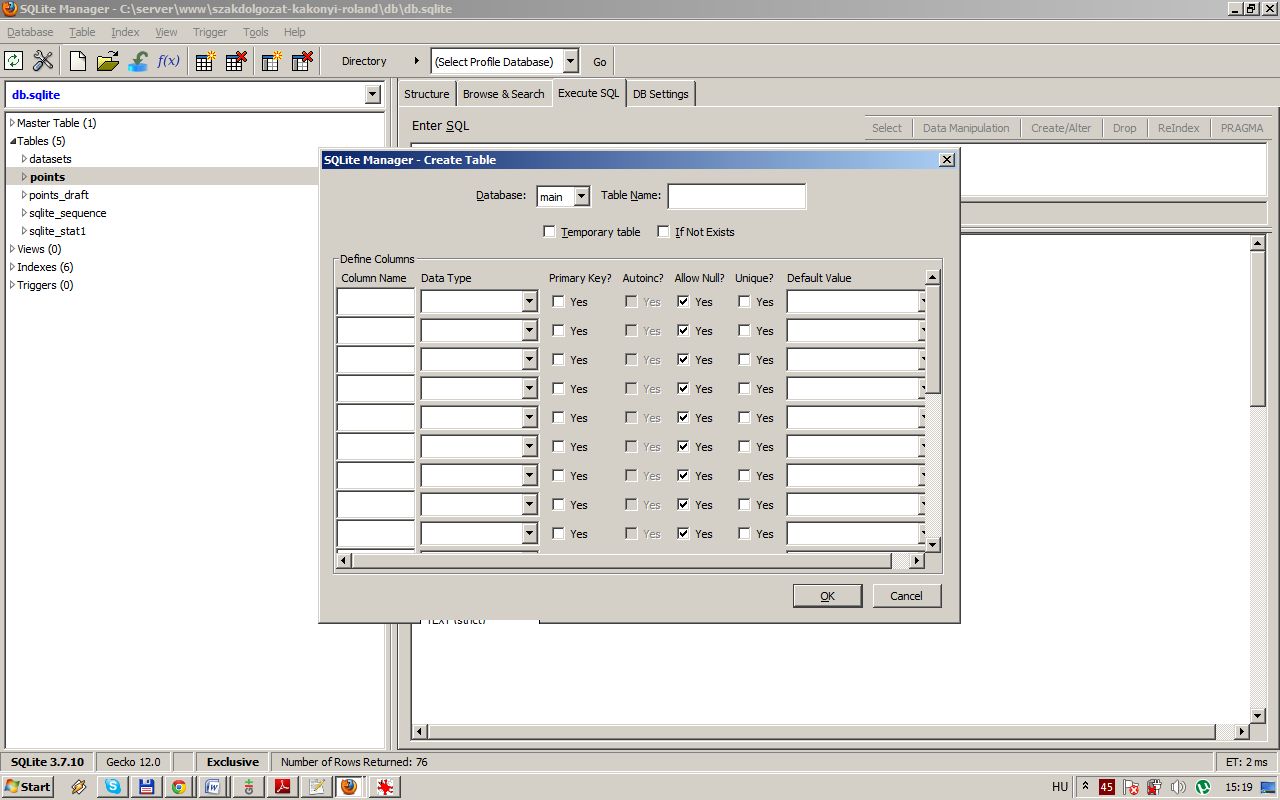
## 2. Választás indoklása

Egy olyan adatbázis motorra volt szükség, amely kis erőforrásigénnyel rendelkezik, de képes a megfelelő szolgáltatások biztosítására. A választás az SQLite adatbázis motorra esett. A használt php interpreterhez elérhető legmagasabb verziószámú, 2.8.17-es SQLite adatbázis motort választottam. Az SQLite választása mellett döntött továbbá a hordozhatósága, egyszerű kezelhetősége, külön szolgáltatástól, szervertől való függetlensége. Továbbá a szakdolgozatom keretein belül fejlesztett alkalmazás teljesítmény és a képesség igényeinek teljesen megfelel.

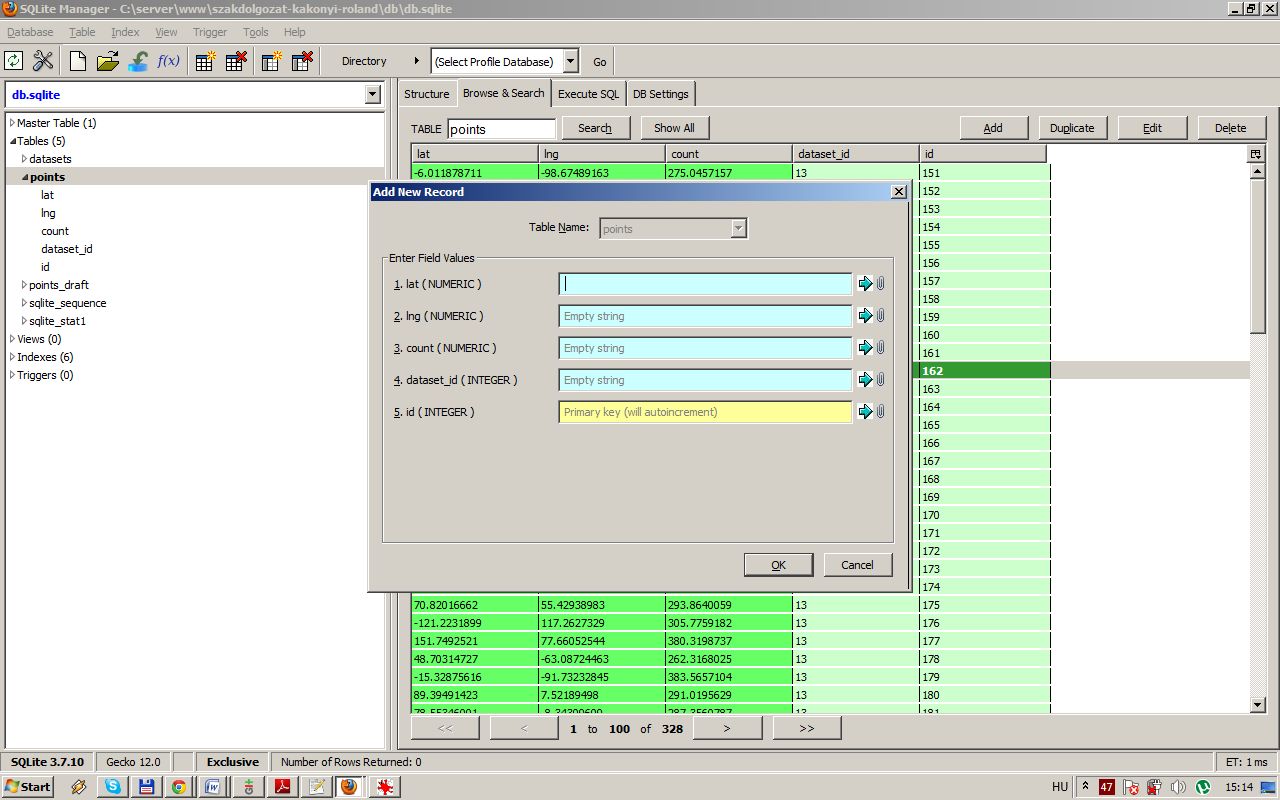
## 3. SQLite Manager, mint adatbázis kezelő felület

A szakdolgozatban bemutatott alkalmazás fejlesztése során a használt adatbázis létrehozására, módosítására, monitorozására és ellenőrzésére az SQLite Manager nevű ingyenes Firefox bővítmény 0.7.7-es változatát használtam [3]. Teljes körű támogatást nyújt az SQLite adatbázisok létrehozására és kezelésére.

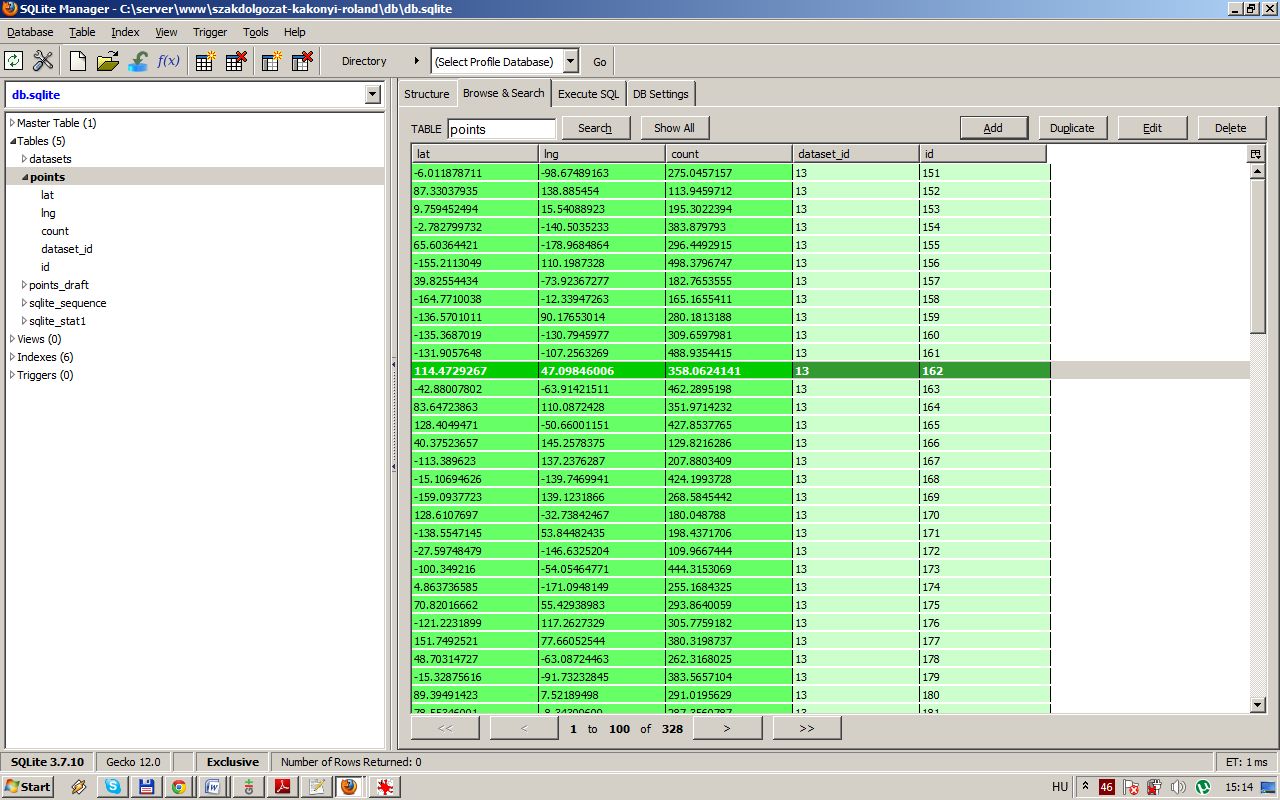
A legfontosabb felületei a következőek:



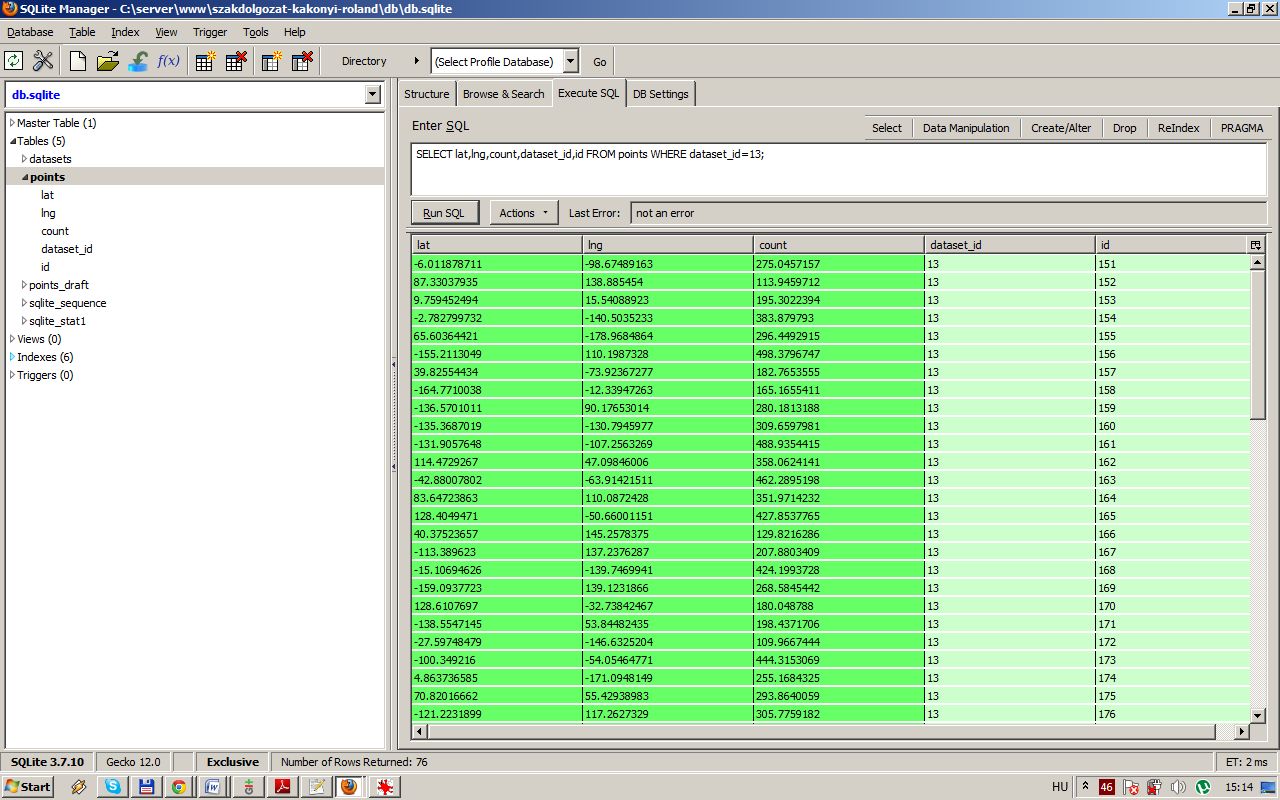
. ábra Tábla létrehozása



. ábra Rekord hozzáadása egy táblához



. ábra Tábla adatainak böngészése



4. ábra Lekérdezés futtatása és annak eredménye

[1] <http://www.w3schools.com/sql/sql_intro.asp>

[2] <http://www.sqlite.org/>

[3] <http://code.google.com/p/sqlite-manager/>

[4] <https://addons.mozilla.org/hu/firefox/addon/sqlite-manager/>