SQLite

Semetey Coşkun

Nisan, 2016

SQLite nedir? Bu soru ile başlayayım ve her zamanki gibi bir alıntı ile konuya giriş yapayım;

> SQLite, dünyada en çok dağıtılan ve tavsiye edilen kaynak kodları halka açık, tamamen C/C++ programlama dilleriyle geliştirilmiş sunucu yazılımı ve konfigurasyon gereksinimi olmayan, işlemsel ve ilişkisel bir SQL veritabanı motorudur.

Yukarıdaki tanım olduğu gibi http://tr.wikipedia.org/wiki/SQLite adresinden kopyala - yapıştır ile alınmıştır. Kaynak kodları açık denildiği için lisansından da söz edelim, SQLite "Public Domain" yani Vikipedi'de "Kamu Malı" olarak Türkçe'leştirilmiş bir lisansa sahiptir. Lisans ile ilgili olarak bir kopyala yapıştır yapmak istemiyorum. İlgisini çeken arkadaşlar için bağlatnı verip geçeceğim;

http://tr.wikipedia.org/wiki/Public_domain

Lisanstan sonra tanımdaki bir başka noktaya değinelim; "sunucu yazılımı ve konfigürasyon gereksinimi olmayan" ibaresi bulunuyor. Bu ne demek oluyor; örneğin bir uygulama geliştirdiniz ve veri tabanı olarak uygulamayı MySQL'e entegre ettiniz diyelim. Eğer ki bir son kullanıcı bu uygulamayı kullanmak istiyorsa bu demektir ki son kullanıcı bilgisayarına bir MySQL sunucu kurmalı (ben bu şekilde biliyorum ve bu bilgiyi Google'da biraz sınadım ve aksi bir bilgiye rastlamadım, yanlış bir bilgi varsa yani MySQL SQLite gibi tekil dosya olarak kullanılabiliyorsa 32. Sayı duyurusundan bu hatayı düzeltebilirsiniz). SQLite'de ise bir sunucu kurarak bu sunucuyu çalışma şekline göre ve istenilen seçeneklere göre optimize etmek için bir ayar dosyasına ihtiyaç yoktur. Çünkü söylendiği gibi bir sunucu kurmaya gerek yoktur.

Örnek olarak; Chrome, Chromium, Firefox gibi uygulamalar, geçmiş bilgileri, sık kullanılanlar, favoriler gibi bağlantı adreslerini ve kullanıcı ayarlarına yönelik diğer bilgileri SQLite veri tabanında saklıyorlar. Ubuntu için Firefox ön tanımlı geliyor, ya da Debian için Firefox çatallaması daha doğrusu sanırım sadece ismi değişik olan bu tarayıcı varsayılan olarak geliyor. Bu tarayıcılar gerekli bilgileri tutabilmek için bize sormadan sistemimize bir veri tabanı yönetim sistemi sunucusu kurarak ve bu sunucuda kullanabilecekleri bir kullanıcı oluşturmuyorlar. Bunlara hiç gerek olmaması için SQLite kullanıyorlar. Bu da tekil bir dosya demek.

Chromium için örnek verelim;

```
1 ~/.config/chromium/Default/History
```

Bu dosya tarayıcının geçmiş bilgilerini sakladığı dosyadır ve tahmin ettiğiniz üzere bir SQLite dosyasıdır.

Verileri bu şekilde saklamanın avantajı ne olabilir? Herhangi bir sunucuya gerek olmaksızın, sıradan bir ikili dosya içerisinde SQL'in avantajlarından yararlanabilip aynı zamanda veriler arasındaki ilişkiyi tablolar halinde oluşturabiliyorlar.

Bunun dışında ayrıca yedeklemesi de gördüğünüz gibi çok kolaylaşıyor bu şekilde; sağ tık -> kopyala.

Bir diğer avantajı da bir çok dil için ara yüz sunması. Bu programlama dillerinden Vikipedi'de verilmiş olanları şunlar;

BASIC, C, D, C++, Common Lisp, Java, C#, Visual Basic, Delphi, Curl, Lua, Tcl, REBOL, R, PHP, ASP, Perl, Ruby, Objective-C, Python, newLisp, Haskell, OCaml, Smaltalk

Yani bir çok dil ile geliştirdiğiniz uygulamada SQLite kullanabilirsiniz. Size ve son kullanıcıya rahatlık sağlayacaktır.

MySQL üzerinde SQL çalışırken hep komut satırını tercih ettim. Arayüzleri olan veri tabanı programlarından daha fazla şey öğrenildiğini düşünüyorum bu şekilde (uçbirimden el ile). Ayrıca aynı şekilde QT programlarken de QT Creator yerine bütün birimleri (Widget) elimle kodluyorum aynı sebepten dolayı; daha fazla şey öğrenildiğini düşünüyorum bu şekilde (el ile).

SQLite kullanan bir uygulama denerken ise oluşturduğum tabloları görme, üzerinde değişiklikler yapma, verilerde düzenleme gibi sebeplerden dolayı yazdığım uygulama dışında, harici olarak tabloları görebileceğim ve MySQL'de olduğu gibi uçbirimden veri tabanını yönetebileceğim bir uygulama aradım. Sonuç olarak sqlite3 isimli paketi buldum. Bu paketi kurduktan sonra uçbirim üzerinden istediğiniz gibi veri tabanları erişimini sağlayabiliyorsunuz (SQLite için tabii).

```
sudo apt-get install sqlite3
```

diyoruz ve paketi kuruyoruz. Uçbirimden veri tabanlarına erişebileceğimiz komutun adı sqlite3. Bunu da çok ufak bir ipucu olarak paylaşmak istiyorum; bir paket kurdunuz; hata yakalamak için ya da herhangi bir sebepten dolayı uygulamayı uçbirimden çalıştırmanız gerekti diyelim, ihtiyacınız olan komut çok büyük bir ihtimalle paketin adıdır.

```
sudo apt-get install pidgin    --> Pidgin'i uçbirimden çalıştırmak için "pidgin" komutunu
    vermelisiniz,
sudo apt-get install vlc       --> VLC'yi uçbirimden çalıştırmak için "vlc" komutunu
    vermelisiniz,
sudo apt-get install gimp      --> GIMP'i uçbirimden çalıştırmak için "gimp" komutunu
    vermelisiniz.
```

Dediğim ve küçük ipucundan tahmin ettiğiniz gibi komutumuz sqlite3. Bu komut şu şekilde çalışıyor;

```
sqlite3 /ilgili/dizinde/bulunan/veri/tabanı/dosyası
```

Yani sqlite3 komutundan sonra veri tabanı dosyamızın ismini belirtiyoruz. Örnek olarak;

```
sqlite3 ~/.config/chromium/Default/History
```

komutunu verdiğimizde Chromium'un geçmiş ile ilgili veri tabanı dosyasının içeriğini görebiliriz. Bu şekilde var olan bir veri tabanı dosyası içeriğini görebilirsiniz. Kendimiz yeni bir veri tabanı oluşturacaksak eğer aynı şekilde;

```
sqlite3 /ilgili/dizinde/oluşturmak/istediğimiz/veri/tabanı/dosyası
```

sqlite3 komutundan sonra bir yol (path) ismi giriyoruz.

Yani sqlite3 komutundan sonraki belirtilen veri tabanı dosyası mevcut ise bu dosya içeriğini görebiliriz eğer ki mevcut değilse yeni oluşturabiliriz. Chromium'un veri tabanı dosyasından gidelim şimdilik ve aşağıdaki komutu verelim;

```
sqlite3 ~/.config/chromium/Default/History
```

Bu komuttan sonra aşağıdaki gibi bir ekranla karşılaşırsanız sorun yok demektir.

```
SQLite version 3.7.3
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite>
```

MySQL'e alışık arkadaşlar için söyleyeyim "SHOW TABLES" komutunu vermeyin, SQLite'te "yemiyor" maalesef. Bağlandığımız mevcut veri tabanındaki tabloları görmek için .tables komutunu veriyoruz;

```
sqlite> .tables
downloads              presentation           urls
keyword_search_terms   segment_usage          visits
meta                   segments
```

Şeklinde tablo isimlerini görmemiz gerekiyor. Standart SQL komutlarını tabii ki kullanabiliyoruz;

```
sqlite> SELECT * FROM meta;
version|17
last_compatible_version|16
early_expiration_threshold|12951875310380757
```

Eğer ki MySQL'i de komut satırından kullanıyorsanız sanırım bu çıktı size biraz ilkel gelecektir. Tablolar halinde değil | ayırıcısı (separator) ile birbirinden ayrılmış olarak görüyoruz verileri.

> **NOT:** Ayırıcı (sonlandırıcı) karakter olarak ";" atomunun kullanıldığına dikkat edin; gireceğiniz SQL ifadelerinden sonra bu atomu kullanmak zorundasınız.

Kendi veri tabanımızı oluşturalım; önce mevcut veri tabanından çıkmak için ".exit" komutunu kullanıyoruz ve sonra;

```
1 sqlite3 SUDO
```

komutunu veriyoruz. Bu şekilde verdiğimiz komutta herhangi bir yol belirtilmediği için oluşturulacak olan veri tabanı dosyamız, uçbirimde o an bulunduğumuz dizin (pwd komutunun çıktısı) içerisinde oluşturulacaktır (mkdir, cd gibi komutların hepsinde olduğu gibi). Komutu verdikten sonra aşağıdaki şekilde bir tablo oluşturalım;

```
1 sqlite> CREATE TABLE users (
2    ...> id INTEGER PRIMARY KEY,
3    ...> name VARCHAR(100) UNIQUE NOT NULL,
4    ...> passwd VARCHAR(100),
5    ...> user_mod VARCHAR(10) NOT NULL);
6 sqlite>
```

Bu şekilde sqlite> satırına geçtiği için anlıyoruz ki SQL ifademizde herhangi bir söz dizim ya da bir başka hata yok ve başarılı bir şekilde tablomuz oluşturuluyor.

SQLite'de INTEGER PRIMARY KEY olarak belirttiğiniz sütun kendiliğinden artan olarak tanımlanmış oluyor. Yani MySQL'deki gibi ayrıca AUTO_INCREMENT olarak tanımlamamıza gerek kalmıyor. Yani bu tablodaki "id" sütunu kendiliğinden artan bir yapıya sahip olacak demek oluyor.

name ise benzersiz (UNIQUE) yani boş geçilemez alan (NOT NULL) olarak işaretleniyor.
user_mod alanı ise aynı şekilde boş geçilemez bir alan olarak işaretlenmiş.

> **NOT:** Bu kullanıcı bilgilerini saklayan bir tablo olarak düşünülmüştü. Doğal olarak aynı isimli iki kullanıcı istenen bir durum değil. user_mod sütunu ise kullanıcı yetkileri ile ilgili bilgileri tutan bir alan. İki kullanıcının sadece okuma hakkının olması (R–) olağan bir durumdur. Bu nedenlerle name alanı benzersiz olarak işaretlendiği halde user_mod tablosunun böyle bir şeye ihtiyacı yoktur, tam tersi bu user_mod alanı için sakıncalı bir durumdur.

Tabloya veri ekleyip kendiliğinden artan alanı görebiliriz. Veri eklemeyelim;

```
1 sqlite> insert into users values (null, 'root', 'ubuntu-tr', 'RWE');
```

Bu şekilde "null" olarak işaretleyip bir değer atamadık ve id değerimiz kendi kendine eklendi. Aynı sonucu şu şekilde de elde edebilirdik;

```
1 sqlite> insert into users (name, passwd, user_mod) values ('rootttt', 'ubuntu-tr', 'RWE');
```

Bu şekile de id numarasını alanlara ve değerlere eklemeden otomatik olarak artmasını sağlayabilirdik. Buradaki roottttt'un nedeni; o alanı benzersiz olarak işaretlediğimiz için root'dan farklı bir değer girmemiz gerekiyordu en hızlı bu şekilde değiştirdim.

Sonuçları şu şekilde test edebilirsiniz;

```
1 sqlite> select * from users;
2 1|root|ubuntu-tr|RWE
3 2|rootttt|ubuntu-tr|RWE
```

Benim bu konuda söyleyebileceklerim şimdilik bu kadar. Daha fazla ayrıntı için sqlite3 komutunu –help parametresi ile beraber verebilirsiniz. Ya da biraz daha ayrıntı için sqlite3 komutunu verdikten sonra .help komutunu verebilirsiniz.

Kaynak
SQLite'in sayfası; http://www.sqlite.org/ \
Dökümanlar; http://www.sqlite.org/docs.html