عنوان: **مروری بر چند تجربه‌ی کاری با SQLite**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۲/۲۷ ۱۲:۴۳:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: SQLite

SQLite
بانک اطلاعاتی سریع، کم حجم و سورس بازی است که استفاده از آن در دات نت فریم ورک بسیار ساده است. فقط کافی است پروایدر مربوط به آن را دریافت کنید و در کدهای قدیمی خود هر جایی مثلا sqlconnection داشتید آن‌را تبدیل به sqliteconnection کنید و امثال آن (به بیان دیگر، پروایدر تهیه شده از معماری ADO.NET پیروی می‌کند و عملا دانش قبلی شما به سادگی قابل استفاده و ارتقاء است). علاوه بر آن پروایدر ADO.NET تهیه شده برای آن، پشتیبانی از Entity framework را هم ارائه می‌دهد.
این دیتابیس تحت سیستم عامل‌های مختلف مهیا است و مهم‌ترین مزیت آن عدم نیاز به نصب آن می‌باشد.

مزایا:
- سورس باز و رایگان
- مهیا بودن آن در سایر پلتفرم‌ها (ویندوز، لینوکس و ...)
- نیازی به نصب ندارد و فقط یک DLL بومی است. این مورد برای کاربرانی که در مدیریت بانک‌های اطلاعاتی پیچیده مشکل دارند، یک مزیت مهم است.
- امکان تشکیل دیتابیس در حافظه. این نکته و توانایی، در آزمون‌های واحد بسیار جالب توجه است. می‌توانید با سرعت بالا دیتابیسی واقعی را در حافظه تشکیل داده، کلیه آزمون‌های واحد خود را اجرا کرده و پس از پایان کار، اثری از دیتابیس و تغییر داده‌ها و مشکلات بازگردانی اطلاعات به حالت اول وجود نخواهد داشت.

ملاحظات:
الف) مرتب سازی SQLite حساس به حروف کوچک و بزرگ است.
برای برگشت به عادت متداولی که وجود دارد می‌شود به صورت زیر عمل کرد:
`select f1 from tbl1 order by f1 COLLATE NOCASE`
یک COLLATE NOCASE اضافه شده است.

ب) رعایت نکات مرتبط با سیستم‌های 64 بیتی
در مورد سیستم‌های 64 بیتی و دات نت قبلا مطلبی را نوشته بودم : { + }. این مطلب دقیقا اینجا کاربرد پیدا می‌کند، از این لحاظ که SQLite یک بانک اطلاعاتی Native است. اگر برنامه‌ی دات نت شما برای حالت Any CPU تهیه شده است، در سیستم‌های 32 بیتی نیاز است تا DLL مرتبط SQLite را توزیع کنید و در سیستم‌های 64 بیتی DLL مرتبط 64 بیتی آن نیاز خواهد بود. در غیراینصورت برنامه‌ی شما در بدو امر کرش کرده و اجرا نخواهد شد.

مشکلات:

الف) کلید خارجی بی خاصیت!
SQLite از کلید خارجی پشتیبانی می‌کند اما آن‌را اعمال نمی‌کند! برای اینکه کلید خارجی را اعمال کنید باید خودتان تریگر بنویسید تا این‌کار را انجام دهد.

ب) پشتیبانی در حد صفر از مباحث همزمانی و تردینگ.
اگر برنامه شما مالتی ترد است، در بد مخمصه‌ای گرفتار شده‌اید. مدام با پیغام database is locked مواجه خواهید شد. (چه انتظاری داشتید؟ یک dll کمتر از 2 مگابایت که قرار نیست کار غول‌های دیتابیسی را انجام دهد)
بنابراین اصلا تصورش را هم نکنید که از این دیتابیس به عنوان بانک اطلاعاتی یک سایت (و محیط‌های چند کاربره) بتوان استفاده کرد و کاربران دچار مشکل نشوند.

ج)حجم بالای دیتابیس و عدم کش
از مباحث caching که در دیتابیس‌های معظم دیگر به صورت توکار وجود دارد خبری نیست. برای مثال اگر یک کوئری قرار است تعدادی را شمارش نماید، حاصلی کش نشده و اگر صدبار هم به صورت متوالی آن‌را فراخوانی کنید باز هم از نو محاسبات آن انجام خواهد شد.
این مورد در حجم بالای دیتابیس واقعا مهم است و نمودش را با دیتابیسی با حجم بالای یک گیگ به وضوح مشاهده خواهید. افت کارآیی و همچنین قرچ و قرچ مداوم هارد دیسک سیستم! (چون به کش رجوع نمی‌شود)

د) امنیت
روی بانک‌های اطلاعاتی اکسس حداقل می‌توان یک کلمه‌ی عبور را قرار داد (که در کسری از ثانیه قابل شکستن است!). در SQLite استاندارد هیچ خبری از این مباحث نبوده و امنیت را باید خودتان تامین کنید. (البته یک نسخه‌ی تجاری هم از این بانک اطلاعاتی با پشتیبانی از رمزنگاری اطلاعات موجود است : +)

ه) مرتب سازی فارسی
هر چند SQLite هیچ مشکلی در ثبت اطلاعات یونیکد و خصوصا متون فارسی ندارد، اما با مرتب سازی کلمات یونیکد مشکل داشته و بر اساس کد اسکی آن‌ها عمل می‌کند. هر چند امکان تعریف Collation سفارشی در آن ممکن است : + (البته ممکن بودن با موجود بودن متفاوت است)

نتیجه گیری:
- SQLite برای دیتابیسی تا حدود یک گیگ که فقط یک نفر قرار باشد از آن استفاده کند انتخاب بسیار مناسبی است (برای مثال فایرفاکس از آن برای ذخیره سازی تنظیمات خودش استفاده می‌کند).

نظرات خوانندگان

نویسنده: Amir
تاریخ: ۱۳۸۹/۰۲/۲۷ ۱۵:۲۵:۲۲

سلام جناب نصیری عزیز
مطلب بسیار خوبی بود و تلنگر خوبی بود برای من. از مطلب بسیار خوبتون و نکات و ملاحظاتتون در مورد استفاده از این ابزار سپاسگزارم و براتون بهترین آرزوها رو دارم.

نویسنده: Abolfazl Hosnaddinov
تاریخ: ۱۳۸۹/۰۲/۲۸ ۰۹:۱۴:۳۳

با سلام خدمت جناب نصیری
جای تقدیر و تشکر است ازین وبلاگ و حرکت بزرگ. ازین همه تجربه ای که با دیگران به اشتراگ می گذارید کمال تشکر رو دارم. امیدوارم در این مسیر همواره موفق بوده باشین. با بهترین آرزوها

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۲/۲۸ ۱۳:۲۹:۳۱

سلام، ممنون. لطف دارید.

نویسنده: RaminMjj
تاریخ: ۱۳۸۹/۰۲/۲۹ ۱۸:۳۸:۵۹

با سلام و تشکر ا مطالب خوبتون.
در مورد کلمه عبور وقتی که شما از Provider دات نت استفاده میکنید به شما اجازه استفاده از یک کلمه عبور برای Encryption را میدهد که کل اطلاعات را کد گذاری میکند که تجربه من نشان میدهد بهتر است این کار قبل از ورود اطلاعات صورت گیرد .مخصوصا وقتی که فیلد از نوع باینری دارید. درباره Access هم میدونم که درباره Access 2003 و به پایین منظورتون بوده که کلمه عبور را میتوان برداشت چون تا اونجا که من میدونم در Access 2007 امنیت به مراتب بالاتر هستش.
در کل SQLite برای کارهای کوچک مناسب هستش ولی ون خودم VistaDB را بیشتر ترجیح میدهم.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۲/۲۹ ۱۹:۲۶:۰۱

- خیر. تمام نگارش‌های Access مد نظر بودند. برای نمونه به برنامه معروف زیر مراجعه کنید :
http://www.elcomsoft.com/aopr.html
- در مورد رمزنگاری هم بله. یک ماژول تجاری همانطور که عرض کردم توسط نویسنده‌ی اصلی آن فروخته می‌شود یا پروایدری که شما عنوان کردید هم با استفاده از windows crypto api ، رمزنگاری را اعمال می‌کند که در این حالت دیتابیس شما فقط منحصر به ویندوز خواهد شد و برای استفاده از آن در MONO مشکل خواهید داشت.

نویسنده: RaminMjj
تاریخ: ۱۳۸۹/۰۲/۳۰ ۱۶:۳۵:۵۸

آقا وحید سلام.
من برنلمه شرکت elcom نسخه Pro با سریال را تهیه کردم و حسابی تستش کردم.
حق با من بود برنامه‌ای نیست که بتونه کلمه عبور Access 2007 را بلافاصله پیدا کنه.این برنامه هم که مدعی این کار بود از تکنیک جستجو،حدس،و کتابخانه برای پیدا کردن کلمه عبور استفاده میکنه که البته سرعت باورنکردنی داره ولی اگر شما کلمه عبور بالای 6 حرف در نظر بگیرید یا برای اطمینان 100% از پیدا نشدن کلمه عبور آن را فارسی در نظر بگیرید این برنامه هم کاری نمیتونه انجام بده و یا حداقل از حوصله شما خارج خواهد شد.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۲/۳۰ ۱۸:۱۳:۲۳

بله. به نظر در 2007 این کسری از ثانیه کمی بیشتر شده:)
چون از یک الگوریتم رمزنگاری به نام RC4 استفاده می‌کنه و پسورد هم همانند نگارش‌های قبلی در این فایل ذخیره نمی‌شه. البته این شرکت مدعی شده که تا 48 ساعت هر طولی رو می‌تونه بشکنه:
lastbit.com/access

نویسنده: RaminMjj
تاریخ: ۱۳۸۹/۰۲/۳۰ ۲۰:۰۳:۵۰

من امتحان کردم ولی نتونست یک رمز 5 کارکتری را که البته فارسی هم بود پیدا کنه.تو تنظیماتش هم اجازه دادم تمامی کارکترها را استفاده کنه.راستی نظرتون در مورد Firebird چی هستش به نظر گزینه خوبی میاد.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۲/۳۰ ۲۰:۵۵:۵۶

آقای مصباحی یک مقایسه در این مورد انجام دادن که جالب است:
بررسی و ارزیابی چند دیتابیس مدفون شده

نویسنده: علی اقدم
تاریخ: ۱۳۸۹/۰۶/۱۳ ۲۲:۰۸:۵۶

سلام آقای نصیری
من در حال نوشتن نرم افزاری هستم که در بالاترین حد حجم اطلاعاتی اون از 500 مگابایت افزایش پیدا نیمکنه ولی تو اون اطلاعاتش کوئری های سنگینی داره
چون من تجربه کار با SQLLiteرو ندارم آیا شما استفاده از اون رو در این برنامه توصیه می کنید ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۶/۱۳ ۲۲:۱۵:۵۸

سلام،
بله. تا این حد رو خوب جواب میده. البته مکانیزم‌های کش کردن اطلاعات رو باید خودتون در نظر داشته باشید و پیاده سازی کنید.
ضمنا استفاده از SQL Server Compact Edition را هم مدنظر داشته باشید (اگر کار شما فقط ویندوزی است)؛ نسخه‌ی جدید آن قرار است از Entity framework پشتیبانی کند و مشکلات استفاده چند کاربری را هم نخواهد داشت و برای ASP.NET بهینه سازی شده؛ هر چند برای SQLite هم اکنون پروایدر EF موجود است.

نویسنده: علی اقدم
تاریخ: ۱۳۸۹/۰۶/۱۴ ۱۸:۵۸:۳۲

بله نرم افزار فعلا فقط ویندوزی است ولی در آینده انشالله با Qt دوباره می نویسمش 8)
فکر کنم از پروایدر DevArt استفاده کنم جواب بده.
ممنون از توجهتون

نویسنده: فرهاد
تاریخ: ۱۳۹۱/۰۵/۲۷ ۲:۴۰

با سلام خدمت استاد عزیز.
من یه برنامه نوشتم که از بانک sqlite استفاده میکنه.

وقتی برناه رو توزیع میکنم روی بعضی از سیستم‌ها پیغامی مبنی بر عدم پیدا کردن dll مذکور رو میده، به نظر شما مشکل از کجاست ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۷ ۸:۲۱

پیشنیاز : « آیا برنامه نویس‌های دات نت باید نگران دنیای 64 بیتی باشند؟ »
SQLite یک بانک اطلاعاتی native است. بنابراین نیاز است دو نسخه 64 بیتی و 32 بیتی آن به همراه برنامه شما توزیع شود.
یا اینکه می‌تونید در تنظیمات پروژه، target platform رو روی X86 قرار بدید. به این صورت روی تمام سیستم‌ها X86 اجرا می‌شود و نیازی به توزیع X64 آن نیست.

عنوان: **استفاده از اسمبلی‌های دات نت 2 در یک پروژه دات نت 4**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۸/۱۰ ۱۴:۴۰:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: SQLite

تنظیمات برنامه BloggerToChm را به دات نت 4 تغییر دادم و بدون مشکل کامپایل شد. اما حین اجرا بلافاصله با خطای زیر برنامه اجرا نمی‌شد:

```
Mixed mode assembly is built against version 'v2.0.50727' of the runtime and cannot be loaded in the 4.0 runtime without additional configuration information.
```

مشکل هم از اسمبلی‌های مرتبط با SQLite است که هنوز برای دات نت 4 کامپایل نشده‌اند. برای رفع این مشکل باید تغییر زیر را (تنظیم گزینه useLegacyV2RuntimeActivationPolicy) به فایل app.config برنامه اضافه کرد:

```
<configuration>
  <startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0"/>
  </startup>
</configuration>
```

برای برنامه‌های ASP.NET نیز به همین صورت است. در آنجا این تغییرات باید به Web.Config اضافه شوند.

نظرات خوانندگان

نویسنده: DPNU Scientific Assocciation
تاریخ: ۱۳۸۹/۰۸/۱۱ ۰۰:۱۵:۳۷

سلام جناب نصیری.
با اجازه بنده می خواهم برای بلاگفا همچنین کاری انجام بدهم.
در مور تبدیلش به CHM می خواستم راهنمایی بگیرم
با کسب اجازه از ایده شما
متشکرم

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۸/۱۱ ۰۰:۲۷:۵۸

در مورد CHM پیشتر مطلب نوشتم: (+)
برای استفاده از آن در یک برنامه مستقل، تمام این‌کارها از طریق خط فرمان این برنامه (html help workshop) هم قابل انجام است. فقط فایل‌های پروژه و ایندکس و غیره آن‌را برنامه شما باید تولید کرده و به صورت پارامتر خط فرمان به آن ارسال کند.

نویسنده: mohsen
تاریخ: ۱۳۸۹/۰۸/۱۱ ۰۰:۳۳:۰۲

سلام استاد نصیری
واقعا ممنونم.
چند ماه پیش به خاطر استفاده از یک اسمبلی که با فریم ورک 2 کامپایل شده بود بشدت دنبال این موضوع بودم ولی به نتیجه ای نرسیدم(تصورم این بود که اسمبلی مجددا باید برای ورژن 4 کامپایل بشه).

ممنون از زحمات شما

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۸/۱۱ ۱۶:۴۹:۳۲

جهت تکمیل بحث در مورد mixed-mode assemblies به این آدرس مراجعه کنید : (+)
به عبارتی اسمبلی‌هایی هستند که حاوی کدهای managed و unmanaged می‌باشند مانند اسمبلی ساخته شده برای SQLite که هم کدهای دات نتی دارند و هم کدهای اصل مرتبط با خود SQLite که با زبان C نوشته شده.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۹/۲۶ ۱۲:۰۱:۱۱

- یک نکته‌ی دیگر:
اگر حین کار با NHibernate و SQLite در یک برنامه‌ی دات نت 4 به خطای "Could not create the driver from NHibernate.Driver.SQLite20Driver" برخوردید، راه حل همان مورد فوق است (اضافه کردن useLegacyV2RuntimeActivationPolicy به فایل کانفیگ برنامه).

عنوان: **مکان دریافت درایورهای جدید SQLite مخصوص دات نت**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۳/۱۵ ۲۱:۵۶:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: SQLite

کتابخانه‌ی System.Data.SQLite مدت مدیدی توسط Robert Simpson در آدرس http://sourceforge.net/projects/sqlite-dotnet2 به روز می‌شد و قابل دسترسی بود. اما این پروژه از سال 2010 متوقف شده و آخرین نسخه‌ی موجود آن 1.0.66.0 است. به همین جهت fork جدیدی از این پروژه در آدرس ذیل (که جزو دومین SQLite نیز می‌باشد) جهت به روز نگه داشتن آن تشکیل شده است:

http://system.data.sqlite.org

در زمان نگارش این مطلب، نگارش 1.0.73.0 این پروژه که در بردارنده‌ی SQLite 3.7.6.3 است، از آدرس فوق قابل دریافت می‌باشد.

نظرات خوانندگان

نویسنده: امین
تاریخ: ۱۳۹۰/۰۳/۱۶ ۱۳:۲۴:۳۵

ممنون از معرفی لینک من از هنوز داشتم لینک اولی که گفته بودین تو برنامه هام استفاده میکردم.
بازم ممنون

عنوان: **استفاده از پروایدر SQLite در Entity Framework 7**
نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۰۸/۲۶ ۱۹:۳۰
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, SQLite, Entity Framework 7

Entity Framework در نگارش 7 خود از منابع داده‌ایی جدیدی پشتیبانی میکند(+) . یعنی از Windows Store، Windows Phone و همچنین ASP.NET 5 (اپلیکیشن‌هایی که از .NET Core استفاده می‌کنند) پشتیبانی خواهد کرد. در این نسخه از دیتابیس‌های non-relational نیز پشتیبانی می‌شود. پروایدر SQLite به صورت رسمی توسط تیم EF ارائه شده است که در ادامه نحوه‌ی استفاده از آن را در یک برنامه کنسول ساده بررسی خواهیم کرد.
کلاس‌های برنامه:

```
using Microsoft.Data.Entity;
using Microsoft.Data.Entity.Metadata;
using System.Collections.Generic;
using System.Linq;

namespace UsingEF7WithSQLite
{
    public class Blog
    {
        public int BlogId { get; set; }
        public string Url { get; set; }

        public List<Post> Posts { get; set; }
    }

    public class Post
    {
        public int PostId { get; set; }
        public string Title { get; set; }
        public string Content { get; set; }

        public int BlogId { get; set; }
        public Blog Blog { get; set; }
    }
}
```

خب تا اینجا مدل‌های برنامه را تعریف کردیم، قدم بعدی افزودن پکیج مربوط به پروایدر SQLite به پروژه است، با دستور زیر این پکیج را نصب می‌کنیم:

```
PM> Install-Package EntityFramework.SQLite –Pre
```

اکنون کلاس کانتکست برنامه را به صورت زیر تعریف می‌کنیم:

```
namespace UsingEF7SQLiteProvider
{
    public class BloggingContext : DbContext
    {
        public DbSet<Blog> Blogs { get; set; }
        public DbSet<Post> Posts { get; set; }

        protected override void OnConfiguring(DbContextOptions builder)
        {
            builder.UseSQLite(@"Data Source=.\BloggingDatabae.db");
        }

        protected override void OnModelCreating(ModelBuilder builder)
        {
            builder.Entity<Blog>()
                .OneToMany(b => b.Posts, p => p.Blog)
                .ForeignKey(p => p.BlogId);

            // The EF7 SQLite provider currently doesn't support generated values
            // so setting the keys to be generated from developer code
            builder.Entity<Blog>()
                .Property(b => b.BlogId)
                .GenerateValueOnAdd(false);
```

```
            builder.Entity<Post>()
                .Property(b => b.BlogId)
                .GenerateValueOnAdd(false);
        }
    }
}
```

کار را با بازنویسی متد OnConfiguration شروع می‌کنیم، در این قسمت باید به EF بگوئیم که می‌خواهیم از SQLite استفاده کنیم برای اینکار از یک Extension Method با نام UseSQLite و پاس دادن کانکشتن استرینگ به آن استفاده می‌کنیم. **نکته:** پروایدر فعلی SQLite در حال حاضر از Generated values پشتیبانی نمی‌کند، برای این منظور باید درون متد OnModelCreating این قابلیت را غیرفعال کنیم.
اکنون می‌توانیم از طریق پاورشل نیوگت دیتابیس را ایجاد کنیم، برای اینکار باید پکیج زیر را به پروژه اضافه کنید:

```
Install-Package EntityFramework.Commands -Pre
```

سپس دستورات زیر را اجرا می‌کنیم:

```
Add-Migration MyFirstMigration
Apply-Migration
```

توسط دستور Apply-Migrate دیتابیس برای شما ایجاد خواهد شد. البته این دستور زمانی استفاده می‌شود که برنامه شما یک اپلیکیشن دسکتاپ باشد. اگر اپلیکیشن شما یک Windows Phone Application است باید در زمان اجرای برنامه این کد را بنویسید:

```
using (var db = new BloggingContext())
{
   db.Database.AsMigrationsEnabled().ApplyMigrations();
}
```

در نهایت می‌توانید با دستور زیر از کانتکست برنامه استفاده کرده و خروجی را مشاهده کنید:

```
using (var db = new Models.BloggingContext())
{
     db.Blogs.Add(new Models.Blog { Url = "http://dotnettips.info" });
     db.SaveChanges();

     foreach (var item in db.Blogs)
     {
         Console.WriteLine(item.Url);
     }
}
Console.ReadLine();
```

نظرات خوانندگان

نویسنده: حمید حسین وند
تاریخ: ۱۳۹۳/۱۰/۰۴ ۲۰:۱

سلام
آیا شما خودتون از این پروایدر استفاده کردید؟
وقتی من میخوام استفاده کنم و از طریق nuGet نصب کنم ارور میده.

```
Install-Package : 'EntityFramework.SQLite' already has a dependency defined for
'EntityFramework.Migrations'.
```

هیچ جوری نتونستم من نصبش کنم.

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۱۰/۰۴ ۲۳:۳۵

- حتما نیاز است که از آخرین نگارش NuGet استفاده کنید (NuGet 2.8.3 یا بالاتر).
- **کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید.** UsingEF7WithSQLite.rar

عنوان: **ساخت دیتابیس sqlite با EF6 Code First**
نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۸/۰۴ ۲۰:۲۵
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, SQLite

تا نسخه EF6 و minorهای آن به دلیل عدم پشتیبانی داریور sqlite از migration، ساخت دیتابیس با code first ممکن نیست برای همین مجبور هستند از پیاده سازی‌های خودشان و موجود بودن دیتابیس از قبل با استفاده از EF با آن کار کنند که یکی از مثال‌های آن در این آدرس قرار دارد و سعی دارد کلاسی مشابه sqlitehelper در اندروید که کار ساخت دیتابیس و مدیریت نسخه را دارد بسازد و از آن استفاده کند. البته در EF7 این مشکل حل شده است و تیم دات نت تمهیداتی را برای آن اندیشیده‌اند. در این نوشتار قصد داریم با استفاده از یک کتابخانه که توسط آقای مارک سالین نوشته شده است کار ساخت دیتابیس را آسانتر کنیم. این کتابخانه که با دات نت 4 به بعد کار میکند خیلی راحت می‌تواند دیتابیس شما را به روش Code First ایجاد کند.

در حال حاضر این کتابخانه از مفاهیم زیر پشتیبانی می‌کند:

تبدیل کلاس به جدول با پشتیبانی از خصوصیت Table
تبدیل پراپرتی‌ها به ستون با پشتیبانی از خصوصیت هایی چون
Column,Key,MaxLength,Required,Notmapped,DatabaseGenerated,Index
پشتیبانی از primarykey و کلیدهای ترکیبی
کلید خارجی و روابط یک به چند و پشتیبانی از cascade on delete
فیلد غیر نال

برای شروع ابتدا کتابخانه مورد نظر را از Nuget با دستور زیر دریافت کنید:

```
Install-Package SQLite.CodeFirst
```

خود این دستور باعث می‌شود که وابستگی‌هایش از قبیل sqlite providerها نیز دریافت گردند.
solution من شامل سه پروژه است یکی برای مدل‌ها که شامل کلاس‌های زیر برای تهیه یک دفترچه تلفن ساده است:

Person

```
public class Person
    {
        public int Id { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }

        public virtual ICollection<PhoneBook> Numbers { get; set; }

    }
```

PhoneBook

```
public class PhoneBook
    {
        public int Id { get; set; }

        public string Field{ get; set; }

        public string Number { get; set; }

        public virtual Person Person { get; set; }
    }
```

پروژه بعدی به نام سرویس که جهت پیاده سازی کلاس‌های EF است و دیگری هم یک پروژه‌ی WPF جهت تست برنامه.
در پروژه‌ی سرویس ما یک کلاس به نام Context داریم که مفاهیم مربوط به پیاده سازی Context در آن انجام شده است:

```
public class Context:DbContext
    {
        public Context():base("constr")
        {
        }

        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            modelBuilder.Conventions.Remove<PluralizingTableNameConvention>();
            var initializer = new InitialDb(modelBuilder);
            Database.SetInitializer(initializer);
        }

        public DbSet<PhoneBook> PhoneBook { get; set; }
        public DbSet<Person> Persons { get; set; }
    }
```

تا به الان چیز جدیدی نداشتیم و همه چیز طبق روال صورت گرفته است؛ ولی دو نکته‌ی مهم در این کد نهفته است:

اول اینکه در سطر اول متد بازنویسی شده onModelCreating، قرارداد مربوط به نامگذاری جداول را حذف می‌کنیم چرا که در صورت نبودن این خط، اسامی که کلاس sqllite برای آن در نظر خواهد گرفت با اسامی که برای انجام عملیات CURD استفاده می‌شوند متفاوت خواهد بود. برای مثال برای Person جدولی به اسم People خواهد ساخت ولی برای درج، آن را در جدول Person انجام می‌دهد که به خاطر نبودن جدول با خطای چنین جدولی موجود نیست روبرو می‌شویم.

نکته‌ی دوم اینکه در همین کلاس Context ما یک پیاده سازی جدید بر روی کلاس InitialDb داشته ایم که در زیر نمونه کد آن را می‌بینید:

```
public class InitialDb:SQLite.CodeFirst.SqliteCreateDatabaseIfNotExists<Context>
    {
        public InitialDb(DbModelBuilder modelBuilder) : base(modelBuilder)
        {
        }

        protected override void Seed(Context context)
        {
            var person = new Person()
            {
                FirstName = "ali",
                LastName = "yeganeh",
                Numbers = new List<PhoneBook>()
                {
                    new PhoneBook()
                    {
                        Field = "Work",
                        Number = "031551234"
                    },
                    new PhoneBook()
                    {
                        Field = "Mobile",
                        Number = "09123456789"
                    },
                    new PhoneBook()
                    {
                        Field = "Home",
                        Number = "031554321"
                    }
                }
            };

            context.Persons.Add(person);
            base.Seed(context);
        }
    }
```

در این کد کلاس InitialDb از کلاس SqliteCreateDatabaseIfNotExists ارث بری کرده‌است و متد seed آن را هم بازنویسی کرده‌ایم. کلاس SqliteCreateDatabaseIfNotExists برای زمانی کاربر دارد که اگر دیتابیس موجود نیست آن را ایجاد کند، در غیر اینصورت خیر. به غیر از آن، کلاس دیگری به نام SqliteDropCreateDatabaseAlways هم وجود دارد که با هر بار اجرا، جداول قبلی را حذف و مجددا آن‌ها را ایجاد میکند.
سپس در پروژه‌ی اصلی WPF در فایل AppConfig رشته اتصالی مورد نظر را وارد نمایید:

```
<connectionStrings>
    <add name="constr" connectionString="data source=.\phonebook.sqlite;foreign keys=true"
providerName="System.Data.SQLite" />
  </connectionStrings>
```

نکته‌ی مهم اینکه با افزودن کتابخانه از طریق nuget فایل app.config به روز می‌شود؛ ولی به نظر می‌رسد که تنظیمات به درستی انجام نمی‌شوند. در صورتیکه به مشکل زیر برخوردید و نتوانستید برنامه را اجرا کنید، کد زیر را که قسمتی از فایل app.config است، مطالعه فرمایید و موارد مربوط به آن را اصلاح کنید:

خطا:

```
The ADO.NET provider with invariant name 'System.Data.SQLite' is either not registered in the machine
or application config file, or could not be loaded
```

قسمتی از فایل app.config:

```
<entityFramework>
    <defaultConnectionFactory type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory,
EntityFramework">
      <parameters>
        <parameter value="mssqllocaldb" />
      </parameters>
    </defaultConnectionFactory>
    <providers>
         <provider invariantName="System.Data.SqlClient"
type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
      <provider invariantName="System.Data.SQLite" type="System.Data.SQLite.EF6.SQLiteProviderServices,
System.Data.SQLite.EF6" />

    </providers>
  </entityFramework>
  <system.data>
    <DbProviderFactories>
      <remove invariant="System.Data.SQLite.EF6" />
      <remove invariant="System.Data.SQLite" />
       <add name="SQLite Data Provider" invariant="System.Data.SQLite" description=".Net Framework Data
Provider for SQLite" type="System.Data.SQLite.SQLiteFactory, System.Data.SQLite" />
    </DbProviderFactories>
  </system.data>
```

کد Load پروژه WPF:

```
public MainWindow()
        {
            InitializeComponent();
           var context=new Context();

            var list= context.Persons.ToList();

            var s = "";

            foreach (var person in list)
            {
                s += person.FirstName + " " + person.LastName +
            " has these numbers:" + Environment.NewLine;

                foreach (var number in person.Numbers)
                {
                    s += number.Field + " : " + number.Number + Environment.NewLine;
                }
```

```
                s += Environment.NewLine;
            }
            MessageBox.Show(s);

        }
```

دانلود مثال