

پایگاه داده‌ی

SQLite

برای برنامه‌نویسان اندروید و .NET

ترجمه و تألیف:

مهندس سید علیرضا قمصری

انتشارات پندار پارس

سرشناسه	: قمصری جویانی، سیدعلیرضا، ۱۳۶۱ -
عنوان و نام پدیدآور	: پایگاه داده‌ی SQLite، برای برنامه‌نویسان اندروید و NET / ترجمه و تالیف سیدعلیرضا قمصری.
مشخصات نشر	: تهران: پندار پارس، ۱۳۹۳.
مشخصات ظاهری	: ۳۱۲ ص.؛ مصور، جدول.
شابک	: 978-600-6529-57-8 : ۱۴۸۰۰۰ ریال
وضعیت فهرست نویسی	: فیپا
موضوع	: اس. کیو. ال. (زبان برنامه نویسی کامپیوتر)
موضوع	: پایگاه‌های اطلاعاتی رابطه‌ای
موضوع	: پایگاه‌های اطلاعاتی -- مدیریت
رده بندی کنگره	: ۷۳/۷۶QA / الف ۱۳۹۳۴۶۸
رده بندی دیویی	: ۷۵۶۵/۰۰۵
شماره کتابشناسی ملی	: ۳۴۵۸۳۱۷

توجه: برای دسترسی به کدهای پروژه‌های کتاب، به صفحه‌ی این کتاب در سایت پندارپارس یا سایت مؤلف مراجعه فرمایید.

انتشارات پندارپارس



دفتر فروش: انقلاب، ابتدای کارگر جنوبی، کوی رشتچی، شماره ۱۴، واحد ۱۶ www.pendarepars.com
 تلفن: ۶۶۵۷۲۳۳۵ - تلفکس: ۶۶۹۲۶۵۷۸ همراه: ۰۹۱۲۲۴۵۲۳۴۸ info@pendarepars.com



نام کتاب : پایگاه‌داده‌ی SQLite، برای برنامه‌نویسان ANDROID و NET.

ناشر : انتشارات پندار پارس

ترجمه و تالیف : سید علیرضا قمصری جویانی

چاپ نخست : اردیبهشت ۹۳

شمارگان : ۵۰۰ نسخه

لیتوگرافی : ترام‌سنج

چاپ، صحافی : فرشویه، خیام

قیمت : ۱۴۸۰۰ تومان شابک : ۹۷۸-۶۰۰-۶۵۲۹-۵۷-۸



*هرگونه کپی برداری، تکثیر و چاپ کاغذی یا الکترونیکی از این کتاب بدون اجازه ناشر تخلف بوده و پیگرد قانونی دارد *

فهرست

2	بخش نخست: SQLITE و مفاهیم SQL
3	فصل 1: SQLITE چیست؟
4	جامعیت و عدم نیاز به سرور
6	کل پایگاه داده در یک فایل
7	عدم نیاز به تنظیمات
7	پشتیبانی توسط انواع ابزارها
8	ویژگی‌های منحصر به فرد
9	مجوزها
9	قابلیت اطمینان بالا
10	خلاصه
11	فصل 2: کاربرد SQLITE
11	یک پایگاه داده کوچک
11	فایل‌های برنامه
13	حافظه‌ی پنهان برنامه
13	آرشیو نمودن و انباره‌ی داده
14	جایگزین معماری کلاینت
14	ابزار آموزش
15	موتور SQL
15	کاربردهای نامناسب
15	حجم بالای تراکنش
16	حجم بالای داده‌ها
16	کنترل دسترسی
16	کلاینت/سروری
17	Replication
17	برخی از استفاده کنندگان معروف
18	خلاصه
19	فصل 3: تهیه و نصب SQLITE
20	محصولات SQLite
20	هسته‌ی SQLite
20	ابزار خط فرمان sqlite3
20	افزونه‌ی Tcl
21	ابزار آنالیز SQLite
21	نسخه‌های از پیش کامپایل شده
22	مستندات
22	سورس کدها
22	Amalgamation
23	سورس فایل‌ها
23	نصب SQLite
28	خلاصه
29	فصل 4: استفاده از CLP و SQLITE ANALYZER
29	استفاده از sqlite3

29 استفاده از CLP در مد Shell
32 استفاده از CLP در خط فرمان
33 مدیریت پایگاه داده
33 تولید یک پایگاه داده
36 دسترسی به اطلاعات اسکیمای پایگاه داده
38 صادر نمودن داده ها
39 وارد نمودن داده ها
40 فرمت بندی
41 استخراج محدودهای از داده ها
42 اجرای خودکار فرآیندها
43 پشتیبان گیری از پایگاه داده
44 استخراج اطلاعات فایل پایگاه داده
46 ابزارهای دیگر SQLite
46 خلاصه
47	فصل 5: SQL برای SQLite، بخش نخست
47 نمونه‌ی پایگاه داده
48 نصب
49 اجرای مثال ها
50 گرامر و ساختار نحوی
51 دستورها
51 لفظها
52 کلیدواژه‌ها و شناسه‌ها
52 کامنت‌ها
53 تولید پایگاه داده
53 ایجاد جدول‌ها
55 ویرایش جدول‌ها
56 پرسوجو بر روی پایگاه داده
57 عملیات رابطه‌ای
58 Select و عملیات pipeline
61 فیلتر نمودن
62 مقادیر
62 عملگرها
63 عملگرهای دوتایی
65 عملگرهای منطقی
66 عملگر like و Glob
67 محدودسازی و مرتب‌سازی
69 توابع و توابع تجمعی
71 گروه بندی
75 حذف موارد تکراری
76 Join جدول‌ها
78 Inner join
79 Cross join
80 Outer join

81	Natural join
81	گرامرهای ترجیحی
82	نامها و اسامی مستعار
84	زیر پرسو جوها
86	پرسو جوهای ترکیبی
89	نتایج شرطی
91	مدیریت null در SQLite
93	خلاصه
95	فصل 6: SQL برای SQLite، بخش دوم
95	ویرایش اطلاعات
95	درج رکورد جدید
95	درج یک سطر
97	درج لیستی از سطرها
100	ویرایش رکوردها
101	حذف رکوردها
102	یکپارچگی دادهها
103	یکپارچگی موجودیت
103	قید یکتایی
104	قید کلید اصلی
108	یکپارچگی دامین
108	مقادیر پیش فرض
110	قید not null
110	قید check
111	یکپارچگی رابطه‌ای
113	Collation
114	کلاس‌های ذخیره‌سازی
117	نماها
118	ایندکس‌ها
120	collation
120	استفاده از ایندکس
122	تریگرها
122	تریگرهای ویرایش
123	مدیریت خطاها
124	نماهای ویرایش پذیر
126	تراکنش‌ها
126	محدوده‌ی تراکنش‌ها
127	خلاصه
۱۲۹	بخش دوم: استفاده از SQLite در برنامه‌نویسی
131	فصل 7: برنامه‌نویسی آندروید با SQLite
131	پیش‌نیازها
132	JDK
132	IDE
132	SDK

133	ADT
133	تهیه‌ی ابزارها
135	دانلود و نصب پکیج استارتر SDK
137	نصب ADT
139	جایگزین‌های Eclipse
139	مدیریت ابزار مجازی
141	کلاس‌های اندروید برای SQLite
141	استفاده از کلاس پایه‌ی کمکی SQLiteOpenHelper
143	کار با کلاس SQLiteDatabase
143	باز و بسته نمودن SQLiteDatabase
144	اجرای پرس‌وجوهای عمومی با SQLiteDatabase
146	استفاده از متدهای کمکی در SQLiteDatabase
147	مدیریت تراکنش‌ها با SQLiteDatabase
148	استفاده از دیگر متدهای SQLiteDatabase
148	پیاده‌سازی SQLiteOpenHelper
152	استفاده از SQLiteQueryBuilder
154	تولید برنامه‌ی مبتنی بر SQLite
171	استفاده از واسط کاربری
174	استفاده از adb
175	خلاصه
177	فصل 8: برنامه‌نویسی NET با SQLITE
177	پیش‌نیازها
178	.NET Framework
178	Visual Studio
180	کلاس‌های NET برای SQLite
180	کلاس SQLiteConnection
181	کلاس SQLiteCommand
183	کلاس SQLiteDataReader
184	کلاس SQLiteDataAdapter
185	تولید برنامه مبتنی بر SQLite
195	استفاده از واسط کاربری
198	خلاصه

به نام تنها نگارنده

در سیستم‌های نرم‌افزاری، همیشه چیزهایی !!! برای ذخیره‌سازی و بازیابی وجود دارد. حجم این نوع داده‌ها نیز نسبت به نوع برنامه و کاربرد آن، متفاوت است. در اوایل انقلاب نرم‌افزاری، استفاده از فایل‌های متنی برای ذخیره‌سازی این نوع داده‌ها کافی بود. به مرور با گسترش نرم‌افزارها، حجم داده‌های مورد استفاده گسترش پیدا نمود و نیاز به مکانیزمی قوی‌تر برای ذخیره‌ی آنها احساس شد. روش‌های قدیمی که مبتنی بر حرکت رو به جلو در فایل و خواندن رکوردها به صورت ترتیبی و یکی پس از دیگری بود، برای حجم بالای اطلاعات چندان کاربردی نبود. در کنار این موضوع نیازهای جدید مانند گزارش‌ها، آمارگیری‌ها، مرتب‌سازی‌ها و فرآیندایی این چنین که در ارتباط مستقیم با اطلاعات انجام می‌گرفت احساس گردید.

در این زمان سیستم‌های مدیریت پایگاه داده ایجاد شدند و وظیفه مدیریت و نگهداری اطلاعات را برعهده گرفتند. با گذشت زمان، رفته رفته این نوع سیستم‌ها نیز گسترش پیدا نمودند و به حدی پیچیده شدند که شاخه‌ای جدید در علوم کامپیوتر و نرم‌افزار را بنا نهادند. مباحثی مانند مدل رابطه‌ای، تراکنش‌ها، همزمانی، روش‌های ذخیره‌سازی و بازیابی اطلاعات، تکثیر و تجمیع و ادغام داده‌ها و مواردی از این دست، هریک بسط یافته و در پیچه‌ای تازه از مباحث مربوط به انباره‌های اطلاعات و همچنین تحلیل و آنالیز آنها را به روی مهندسان نرم‌افزار گشودند.

این موضوع هرچند به گسترش محصولات و علوم مرتبط با کامپیوتر و تکنولوژی اطلاعات کمک نمود، ولی اثرات دیگری هم داشت، پیچیده شدن این برنامه‌ها استفاده از آنها را تخصصی‌تر نمود. این موضوع بدان معنی بود که برای استفاده از ساده‌ترین مکانیزم یک سیستم ذخیره‌ی اطلاعات، لازم بود تا تنظیم‌ها و پیش‌نیازهای گسترده‌ای در نظر گرفته شود، زیرا پیچیدگی سیستم سرویس دهنده، این موارد را ایجاب می‌نمود. برای نمونه‌ای ساده، می‌توان به مکانیزم مبتنی بر معماری کلاینت/سرور اشاره نمود که در سیستم‌های اطلاعاتی، موضوعی عادی و مرسوم است.

اینک فرض کنید قصد داشتید یک برنامه‌ی ساده‌ی واژه‌نامه را تولید کنید. با توجه به ماهیت برنامه، لزوم استفاده از پایگاه داده غیرقابل انکار است. ولی آیا این قابل قبول خواهد بود که برای استفاده از یک واژه‌نامه‌ی کامپیوتری، نیاز به برنامه‌ای مانند MS SQL Server، Oracle، و یا MS Access باشد؟ در مورد ابزارهایی با منابع سخت‌افزاری و نرم‌افزاری محدود نیز که امکان اجرای این نوع موتورهای پایگاه داده را ندارند چه راهی می‌توان پیشنهاد داد؟

این نوع پرسش‌ها و نیازهایی که بدون پاسخ مانده بودند موجب تولید نوعی از موتور پایگاه داده شد که SQLite نام گرفت. این سیستم پایگاه داده‌ای، تمامی قابلیت‌های مورد انتظار از یک پایگاه داده‌ی استاندارد را ارائه می‌دهد ولی پیچیدگی‌های این نوع سیستم‌ها را ندارد. حجم کم، منابع کم مورد نیاز و عدم نیاز به تنظیم‌های خاص، موجب شده تا بتوان از آن در ابزارهای ضعیف نیز همانند سیستم‌های قدرتمند، استفاده نمود.

نباید انتظار معجزه از SQLite داشت، بلکه باید به آن به عنوان قهرمانی برای کمک به موتورهای پایگاه داده‌ی قوی نگریست. SQLite رقیب آنها نیست بلکه همکاری است که همانند دیگر سیستم‌های اطلاعاتی، امکان استفاده از پایگاه داده‌ی رابطه‌ای را برای سرویس گیرنده‌ها (حتی ابزارهای ضعیف) مهیا می‌سازد.

این کتاب دیباچه‌ای است بر SQLite و در کنار توضیح معماری این نرم‌افزار، شیوه‌ی استفاده از آن را مورد بررسی قرار می‌دهد. آنچه پیش روی شماست، نتیجه‌ی کناره‌م قرار دادن مطالب پراکنده‌ای است که از منابع مختلف استخراج نموده‌ام. البته تجربه‌ی شخصی خود را نیز در تدوین مطالب دخالت داده و سعی کردم موارد فنی غیر ضروری را از مطالب حذف و به خواست خدا برای جلد دوم کتاب کنار بگذارم. کتاب را به دو بخش تقسیم کرده‌ام،

بخش نخست به توضیح SQLite و ابزارهای کار با آن و همچنین بررسی نگارش SQL مربوط به آن می‌پردازد و بخش دوم کتاب که نگاهی کاربردی به این محصول دارد و استفاده از آن در محیط برنامه‌نویسی را تشریح می‌نماید.

هرچند تجربه‌ای مختصر در ترجمه و جمع‌آوری کتب علوم کامپیوتر دارم ولی از این نکته نیز غافل نیستم که هنوز در ابتدای راه بوده و نیازمند کوشش و یادگیری بیشتری می‌باشم. لذا پیشاپیش از خواننده درمورد بخش‌هایی از کتاب که دارای متونی گنگ و نامفهوم می‌باشد، کمال پوزش را دارم و خود را نیازمند توصیه‌ها، پیشنهادها و انتقادهای شما دوست عزیز می‌دانم. خوشحال خواهم شد در تالیفات و نگارش‌های بعدی کتاب، از نظرهای شما بهرمنند شوم.

در پایان شکرگذار پروردگارم که عنایتی به من فرمود تا بتوانم این کتاب را گردآوری نمایم. همچنین از تمامی اساتید خود و دوستانی که در این مسیر مرا یاری کردند، به‌ویژه همسر مهربانم - که سنگ صبور، پشتوانه و الهام بخش من بوده و هست - کمال تشکر و قدردانی را دارم.

سید علی رضا قمصری جویانی / بهار 93

s.alireza@qamsari.com

بخش نخست

SQLite و مفاهیم SQL

- فصل نخست : SQLite چیست ؟
- فصل دوم : کاربرد SQLite
- فصل سوم : تهیه و نصب SQLite
- فصل چهارم : استفاده از CLP و SQLite Analyzer
- فصل پنجم : SQL برای SQLite، بخش نخست
- فصل ششم : SQL برای SQLite، بخش دوم

فصل 1

SQLite چیست؟

در یک توصیف کوتاه، SQLite یک پکیج نرم‌افزاری است که یک سیستم مدیریت پایگاه‌داده‌ی رابطه‌ای یا RDBMS¹ را ارائه می‌دهد. این نوع سیستم‌ها، برای ذخیره‌ی اطلاعات و داده‌های کاربران در قالب مجموعه‌ای از رکوردها درون جداول مورد استفاده قرار می‌گیرند. البته در کنار ذخیره کردن داده‌ها، موتور پایگاه‌داده² قادر است دستورها و پرس و جوهای پیچیده را نیز برای تولید گزارش‌ها و خلاصه‌سازی‌ها و تهیه‌ی خروجی‌های مورد نظر، پردازش نماید. این دستورها می‌توانند از یک یا ترکیبی از چند جدول برای تولید خروجی استفاده نمایند. از انواع دیگر این نوع RDBMSها می‌توان در بخش تجاری به Oracle، DB2 محصول IBM و SQL Server محصول Microsoft اشاره نمود. همچنین مواردی مانند MySQL و PostgreSQL نیز که به صورت متن باز⁴ ارائه شده‌اند، از محبوبیت بالایی برخوردار بوده و در محصولات تجاری نیز مورد استفاده قرار می‌گیرند.

عبارت Lite در SQLite که در لغت به معنای سبک بوده و کارایی و توانایی کم را القا می‌سازد، به قابلیت این ابزار اشاره ندارد، بلکه به سادگی و سهولت مکانیزم‌های موجود در حین نصب، مدیریت و استفاده از منابع در این برنامه اشاره دارد. به صورت کلی SQLite با این ویژگی‌ها شناخته می‌شود:

- **عدم نیاز به سرور:** SQLite برای فعالیت، به یک سرور و یا پردازنده‌ی مجزا و مستقل نیاز ندارد و موتور پایگاه‌داده به صورت مستقیم به فایل‌های داده‌ای خود دسترسی پیدا می‌کند.
- **عدم نیاز به تنظیمات:** عدم وجود سرور به معنی عدم نیاز به نصب و تنظیمات مربوط می‌باشد و تولید یک نمونه پایگاه‌داده در SQLite به سادگی تولید و باز کردن یک فایل معمولی است.
- **استقلال از سکوی کاری:**⁵ تمامی پایگاه‌داده درون یک فایل ذخیره می‌شود که مستقل از سیستم‌عامل بوده و بدون نیاز به تنظیمات مدیر سیستم، قابل انتقال به یک سکوی کاری دیگر است.
- **جامعیت:** یک کتابخانه به تنهایی، تمامی سیستم پایگاه‌داده را شامل می‌گردد و همین کتابخانه به صورت مستقیم با برنامه‌ی سرویس گیرنده در تعامل خواهد بود.
- **نیاز منابع محدود:** نسخه‌ی پایانی و کامپایل شده، چیزی حدود یک مگابایت حجم خواهد داشت که در زمان اجرا تنها چند مگابایت حافظه اشغال می‌کند. البته با اعمال تغییرات و تنظیماتی در حین کامپایل و استفاده از برنامه، این حجم نیز به خوبی قابل کاهش است.

¹ Relational Database Management System

² Database Engine

³ Query

⁴ Open Source

⁵ Platform

4/ فصل نخست؛ SQLite چیست؟

- **تراکنشی¹**: تراکنش‌های SQLite به صورت کامل منطبق بر ACID بوده و امکان دسترسی امن به داده را از طریق چندین thread و یا پردازش ارائه می‌دهند.
 - **پشتیبانی SQL**: SQLite از بیشتر دستورهای ارائه شده در استاندارد SQL92 یا SQL2 پشتیبانی می‌نماید.
 - **قابلیت اطمینان بالا**: تیم تولید و توسعه‌ی SQLite به طور جدی و دقیق، فرآیند بررسی، چک کردن و تست این نرم‌افزار را انجام داده‌اند و می‌توان از بدون خطا بودن آن تا حد بسیار بالایی اطمینان داشت.
- روی هم، SQLite محیطی کاربردی و انعطاف‌پذیر را برای کار با پایگاه‌داده‌ی رابطه‌ای مهیا نموده است که افزون بر نیاز کم به منابع سیستم، برای کاربران و برنامه‌نویسان به آسانی قابل استفاده می‌باشد.

جامعیت و عدم نیاز به سرور

برخلاف بیشتر محصولات RDBMS، SQLite دارای معماری کلاینت/سروری نیست. بیشتر سیستم‌های پایگاه‌داده در ابعاد بزرگ، دارای یک پکیج حجیم سرویس دهنده هستند که موتور پایگاه‌داده را ارائه می‌دهد. سرور داده، اغلب متشکل از چندین پردازش است که به صورت همزمان به مواردی مانند ارتباطات کلاینت‌ها، تعاملات با فایل‌ها، cache کردن داده‌ها، بهینه کردن پرس و جوها و پردازش آنها می‌پردازند. یک پایگاه‌داده، عموماً از تعداد زیادی فایل تشکیل شده است که در یک یا چند دایرکتوری از سرور ذخیره شده‌اند. برای دسترسی به پایگاه‌داده، تمامی این فایل‌ها باید موجود و سالم باشند که این نیاز، فرآیند جابجایی و تهیه‌ی نسخه پشتیبان از یک پایگاه‌داده را سخت می‌نماید.

همچنین تمامی ماژول‌های سرور نیاز به منابعی مشخص دارند که می‌بایست توسط سیستم میزبان² تأمین گردد. تجربه نشان داده است که مواردی مانند حساب کاربری اختصاصی برای سرویس‌ها، اسکریپت‌های زمان شروع به کار، حافظه‌ی اختصاصی برای سیستم میزبان برنامه‌ی سرور، عملاً سرور داده یا دیتابیس سرور را به هزینه‌برترین بخش سیستم نرم‌افزار تبدیل می‌نماید. این موارد معمولاً منجر به این رویداد می‌شود که به مشتری پیشنهاد می‌گردد از یک سیستم مستقل برای میزبانی برنامه‌ی سرور استفاده کند.

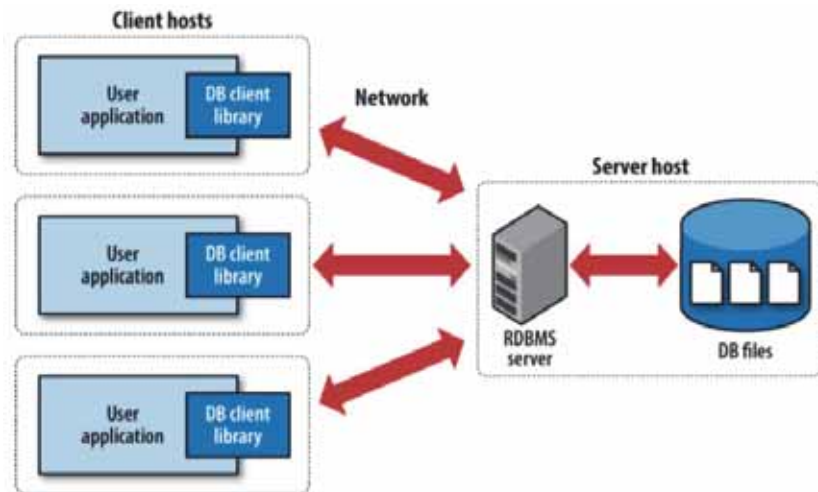
برای دسترسی به پایگاه داده، کتابخانه‌های نرم‌افزاری سمت کلاینت که عموماً توسط تولیدکنندگان پایگاه داده ارائه می‌شود مورد نیاز است. این کتابخانه‌ها می‌بایست توسط برنامه‌هایی که قصد تعامل و دسترسی به سرور پایگاه داده را دارند مورد استفاده قرار گیرند. آنها³ API یا واسط‌های برنامه‌نویسی را ارائه می‌دهند که به وسیله‌ی آنها می‌توان سرور پایگاه داده را یافت و به آن دسترسی پیدا کرد. همچنین پرس‌وجوها و دستورها را در ارتباط با پایگاه داده اجرا می‌نمایند. شکل 1-1 شیوه‌ی قرارگیری المان‌های مختلف در یک مدل عمومی کلاینت/سروری در ارتباط با RDBMS را نشان می‌دهد.

¹ Transactional

² Host System

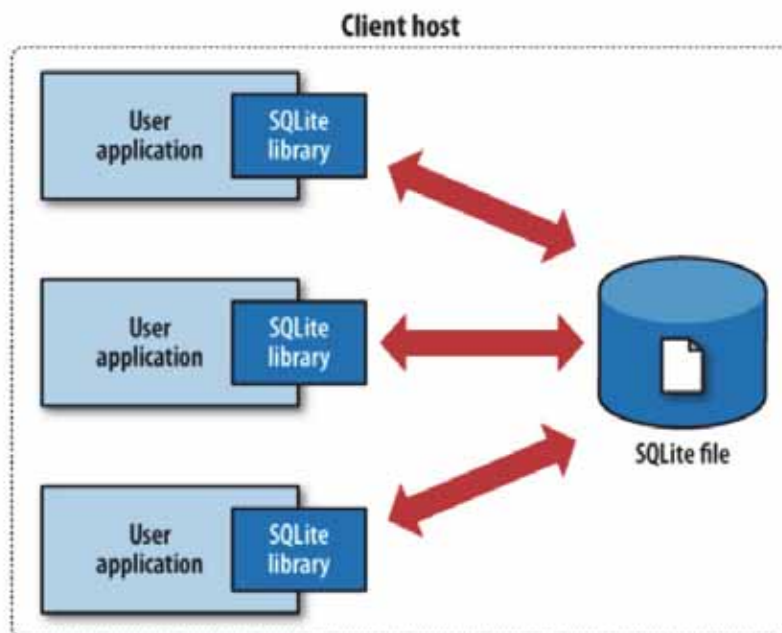
³ Application Programming Interface

پایگاه داده‌ی SQLite / 5



شکل 1-1 مدل عمومی کلاینت/سروری در ارتباط با RDBMS

درمقابل، SQLite یک سرور مجزا ندارد. تمام موتور پایگاه داده در تعامل با برنامه‌ای است که قصد دسترسی به پایگاه داده را دارد. تنها منبعی که در میان برنامه‌های مختلف به اشتراک گذاشته می‌شود، همان تک فایل پایگاه داده می‌باشد که بر روی دیسک ذخیره شده است. چنانچه قصد تغییر نسخه‌ی پشتیبان و یا جابجا کردن پایگاه داده را داشته باشید کافی است همان فایل را کپی نمایید. شکل 1-2 معماری مبتنی به SQLite را نشان می‌دهد.



شکل 1-2 معماری بدون سرور SQLite

6/ فصل نخست؛ SQLite چیست؟

با حذف سرور، بخش زیادی از پیچیدگی برطرف می‌شود که این خود باعث سادگی ماژول‌های برنامه و محدود شدن به آن دسته از احتیاجاتی می‌گردد که در ارتباط با سیستم‌عامل و تنظیمات آن است. برخلاف RDBMS‌های مرسوم که نیازمند تعاملات پیچیده‌ی چند پردازنده‌ای و چند نسخه‌ای هستند SQLite تنها به قابلیتی در حد خواندن و نوشتن در یک مخزن و انباره¹ نیاز دارد.

این سادگی باعث شده که ملزومات پشتیبانی در SQLite کم بوده و در بیشتر محیط‌ها همانند تلفن همراه، پخش کننده‌های مدیای قابل حمل، کنسول‌های بازی و دیگر ابزارها به‌آسانی قابل استفاده باشد؛ محیط‌هایی که دیگر RDBMS‌ها توان ورود به آنها را ندارند.

اگرچه SQLite از مدل مرسوم کلاینت/سروری استفاده نمی‌کند ولی گاهی برنامه‌ی کاربردی را تحت عنوان کلاینت SQLite معرفی می‌نماید. این مفهوم، بیشتر برای بیان این موضوع به کار برده می‌شود که برنامه‌های مستقل می‌توانند به‌صورت همزمان به یک فایل پایگاه داده‌ی SQLite دسترسی پیدا کنند و بدین معنی نیست که هرکدام از یک سرور مجزا استفاده می‌نمایند. SQLite برای تعامل مستقیم با برنامه‌ی اجرایی طراحی گردیده است و این فرض نیز باعث حذف کتابخانه‌های اضافی و در نتیجه، ساده‌تر شدن روال نصب، شده است.

حذف وابستگی‌های خارجی، خود باعث حذف مشکلات حاصل از وجود نسخه‌ها و نگارش‌های متفاوت می‌گردد. اگر کدهای SQLite به‌صورت مستقیم درون برنامه‌ی شما قرار گیرد، هیچ‌گاه نگران استفاده از نسخه‌ی مناسب توسط سرویس گیرنده در حین اجرای برنامه نخواهید بود.

هرچند، حذف سرور، محدودیت‌هایی را نیز به همراه خواهد داشت. SQLite برای آدرس‌دهی انباره به‌صورت محلی طراحی شده است و بدین معناست که حتی سرور وب نیز تنها قادر به اتصال به پایگاه داده‌ی محلی خود خواهد بود. این مدل برای مواردی که قرار است چندین کلاینت از ماشین‌های مختلف به یک پایگاه داده‌ی متمرکز متصل شوند مناسب نخواهد بود. این مدل، یک مدل کلاینت/سروری بوده و بهتر است در طراحی آن از سیستم‌های پایگاه داده-ای که این معماری را پشتیبانی می‌نمایند استفاده شود.

کل پایگاه داده در یک فایل

SQLite تمامی پکیج پایگاه داده را درون یک فایل قرار می‌دهد. این فایل شامل ساختار پایگاه داده و همچنین تمامی داده‌هایی است که درون جداول مختلف ذخیره گردیده است. فرمت فایل به صورت مستقل از سیستم‌عامل طراحی شده و ورای از اینکه طول یک واژه و یا ترتیب چیدمان بایت‌ها در آن چگونه است، قابل استفاده خواهد بود.

¹ Storage

7 / پایگاه داده‌ی SQLite

نگهداری تمامی پایگاه داده در یک فایل، روال تولید، کپی و پشتیبان‌گیری از پایگاه داده را سهولت می‌بخشد. کل پایگاه داده می‌تواند به صورت یک‌جا جایجا شده و حتی توسط سیستم‌های مدیریت نسخ¹، کنترل و همانند یک فایل معمولی ویرایش شده و به اشتراک گذاشته شود. این احتمال نیز که به دلیل از بین رفتن یکی از چندین فایل پایگاه داده (که در سیستم‌های پایگاه‌داده‌ی دیگر مرسوم است)، اطلاعات به‌طور کلی از دست برود نیز منتفی خواهد بود. تجمع داده‌ها در یک فایل، این مزیت را نیز خواهد داشت که برای مدیریت آن نیاز به مهارت بالایی از سمت کاربر نخواهد بود.

عدم نیاز به تنظیمات²

از دیدگاه کاربر، SQLite نیاز به نصب ندارد و هیچ تنظیماتی لازم نبوده و نیازی به نگرانی در این زمینه نیست. اگر چه پارامترهایی برای بهینه‌سازی روال عملکرد SQLite وجود دارد، ولی این پارامترها صرفاً برای برنامه‌نویسان بوده و از دید کاربر نهایی مخفی است. با حذف سرور و ادغام موتور پایگاه‌داده با برنامه، دیگر کاربر نیازی ندارد از این موضوع آگاهی داشته باشد که برنامه از یک پایگاه داده استفاده می‌کند. این یک روال کاربردی مناسب است و احتمالاً نهایت کاری که او انجام خواهد داد انتخاب یک فایل (پایگاه‌داده) خواهد بود.

پشتیبانی توسط انواع ابزارها

اندازه‌ی کوچک کد و همچنین میزان پایین منابع مورد نیاز برای اجرا، SQLite را برای استفاده در ابزارهایی که دارای سیستم‌عامل‌های محدود و ساده می‌باشند، به گزینه‌ای مناسب و محبوب تبدیل نموده است. کدهای ANSI C که بیشتر به سیستم‌های اولیه و قدیمی متمایلند، دارای ساختاری است که توسط اغلب کامپایلرهای ابزارهای الکترونیکی پشتیبانی می‌شود. با تنظیمات پیش فرض، کتابخانه‌ی کامپایل شده‌ی SQLite در بیشتر سیستم‌عامل‌ها حجمی کمتر از 700 کیلو بایت و برای اجرا به حافظه‌ای حدود 4 مگا بایت نیاز دارد. با حذف ویژگی‌های اضافه و پیشرفته، می‌توان حجم کد را تا حدود 300 کیلو بایت کاهش داد. در پایان، با حداقل تنظیمات، حجم کد تا 250 کیلو بایت و میزان حافظه‌ی مصرفی تا نیم مگا بایت قابل کاهش می‌باشد. البته حجم پایگاه داده را در این آمار در نظر نگرفته‌ایم که بنا به حجم داده، متغیر است.

SQLite به کمترین میزان پشتیبانی از محیط میزبان برای انجام عملیات خود نیازمند است و به صورت کاملاً ماژولار³ تولید شده است. سیستم مدیریت تخصیص حافظه‌ی داخلی به‌آسانی قابل ویرایش و حتی جایگزینی است. دسترسی به فایل‌ها و انباره با استفاده از واسطی به‌نام سیستم فایل مجازی⁴ یا VFS انجام می‌شود که برای همخوانی

¹ Revision Control System

² Configuration

³ Modular

⁴ Virtual File System

8/ فصل نخست؛ SQLite چیست؟

بیشتر با محیط‌های مختلف، قابل تغییر و توسعه می‌باشد. در پایان باید به این نکته اشاره نمود که SQLite در هر ابزاری با یک پردازنده‌ی 32 بیتی قابل اجرا و استفاده می‌باشد.

ویژگی‌های منحصر به فرد

SQLite ویژگی‌هایی را ارائه می‌دهد که در بسیاری از دیگر سیستم‌های پایگاه‌داده ارائه نمی‌شود. مهم‌ترین آنها، استفاده از مکانیزم نوع داده داینامیک¹ برای جداول است. موتور SQLite به شما این امکان را می‌دهد که هر نوع داده‌ای² را در هر ستون از جدول، مستقل از نوع داده‌ی آن، قرار دهید. این یک تفاوت اساسی با دیگر سیستم‌های پایگاه‌داده می‌باشد. زیرا در آنها اغلب مکانیزمی سخت‌گیرانه برای درج داده با توجه به نوع داده لحاظ می‌گردد.

از بسیاری جهات سیستم نوع داده‌ی پویای موجود در SQLite همانند فرآیندی است که در زبان‌های اسکریپتی محبوب مورد استفاده قرار می‌گیرد. در این فرآیند، یک نوع داده‌ی مینا وجود دارد که می‌تواند هر نوع مقداری را از یک عدد صحیح گرفته تا یک رشته عبارت در خود جای دهد. این مکانیزم می‌تواند بسیاری از محدودیت‌های موجود در حین کدنویسی را برطرف نماید.

دیگر ویژگی مفید، قابلیت دستکاری نمودن و تعامل با بیش از یک پایگاه‌داده در آن واحد است. SQLite این امکان را فراهم می‌آورد که یک موتور پایگاه‌داده به صورت همزمان به چندین فایل پایگاه‌داده متصل شود. این ویژگی باعث می‌شود SQLite بتواند عبارات SQLی را پردازش نماید که میان چند پایگاه‌داده ارتباط برقرار می‌نماید. مواردی مانند اتصال بین جداول دو یا چند پایگاه‌داده در حین پردازش در یک پرس‌وجو و یا کپی کردن دسته‌ای از داده‌ها با استفاده از یک دستور می‌تواند نمونه‌هایی از کاربرد این ویژگی باشد.

SQLite همچنین قابلیت تولید یک پایگاه‌داده‌ی درون حافظه‌ای به صورت کامل را دارا است. در واقع این پایگاه‌داده‌ی فایل‌هایی است که همتای خارجی و فیزیکی درون انباره نداشته و تمامی چرخه‌ی عمر آن درون فایل حافظه‌ی پنهان³ قرار دارد. با وجود اینکه این مدل پایگاه‌داده‌ی موجود در حافظه، دارای دوام نبوده و همچنین از مکانیزم‌های تراکنش به طور کامل پشتیبانی نمی‌نماید ولی سرعت بالای آنها به دلیل انجام عملیات درون حافظه، آنها را به گزینه‌ی مناسبی برای ذخیره‌ی داده‌های موقت و دیگر اطلاعات انتقالی نموده است.

موارد دیگری نیز وجود دارد که باعث انعطاف‌پذیری بالای SQLite می‌شود. برخی از آنها همانند جداول مجازی⁴ دارای معادل در سیستم‌های مرسوم پایگاه‌داده هستند که البته شخصی‌سازی‌هایی با توجه به معماری کلی SQLite در

¹ Dynamic-Type System

² Data Type

³ Cache

⁴ Virtual Tables

پایگاه داده‌ی SQLite / 9

آنها انجام گرفته است. این ویژگی‌ها و ضمایم¹ ابزارهایی قدرتمند را برای بالابردن کارایی SQLite در زمینه‌ی نیاز و موقعیت خاص شما ارائه نموده‌اند.

مجوزها

SQLite و همچنین کد آن نیاز به هیچ نوع مجوز خاصی ندارد. حتی تحت لیسانس GPL² و یا هیچ نوع لیسانس متن باز نیز نیستند. تولیدکنندگان SQLite کد آن را در معرض استفاده‌ی همگان قرار داده‌اند و این بدان معناست که آنها ادعایی در ارتباط با استفاده از کد و یا خود SQLite نداشته و آنها را بدون هیچ‌گونه چشم‌داشتی در معرض استفاده‌ی دیگران قرار داده‌اند.

به بیانی دیگر، می‌توانید با کدهای SQLite به هر شکلی که مایلید رفتار نمایید. آنها را مناسب با نیاز خود تغییر داده و یا از نسخه‌ی موجود آن استفاده نمایید. کد و خروجی کامپایل شده‌ی آن به هر شکل توسط شما قابل استفاده بوده و تغییر و عرضه‌ی آن و حتی فروش نسخه‌ی تولیدی شما بدون مانع است. هیچ‌گونه محدودیت و هماهنگی با تولید کنندگان اولیه لازم نیست. البته مواردی که به انحصاری بودن الگوریتم و ثبت اختراع مربوط شود و یا تعاملات تجاری و استانداردسازی‌ها می‌بایست در ارتباط مستقیم با تیم تولید کننده و زیر نظر آنها انجام گیرد که این نیز تنها برای سهولت در توسعه‌ی SQLite می‌باشد.

قابلیت اطمینان بالا

هدف از استفاده از یک پایگاه داده، نگه‌داری اطلاعات به صورت امن و سازمان یافته است. تیم تولید SQLite به این نکته آگاهی دارند که یک پایگاه داده که به دارا بودن خطا و عدم قابلیت اطمینان شهرت پیدا کند، توسط هیچ کس مورد استفاده قرار نخواهد گرفت. برای بالا بردن سطح قابلیت اطمینان، هر نگارش از SQLite پیش از نهایی شدن و انتشار، با یک فرایند طولانی و پیچیده‌ی تست، یعنی بیش از 10 میلیون تست و پرس‌وجوی تستی، مورد بررسی و اعتبارسنجی قرار می‌گیرد.

تست Soak نیز پیش از انتشار نسخه‌ی پایانی، با بیش از 2/5 میلیون تست که موارد مختلف همانند عبارت، کمبود حافظه و کمبود فضای ذخیره‌سازی را زیر پوشش قرار می‌دهد انجام می‌شود. این تست‌ها بیشتر به شکلی طراحی می‌شوند که سیستم دارای منابع محدود بوده و در مقابل، از بالاترین قابلیت برنامه استفاده گردد.

این حجم بالای عملیات تست باعث می‌شود که تعداد خطاهای SQLite بسیار کم و محدود باشد. البته باید توجه داشت که مواردی که به خرابی پایگاه داده³ و یا پاک شدن اطلاعات مربوط شود همچنان باقی بوده و در این موارد نیاز به کنترل توسط خود برنامه‌نویس است.

¹ Extensions

² General Public License

³ Database Corruption

10/ فصل نخست؛ SQLite چیست؟

همچنین طراحی‌های جدید نباید مکانیزم پشتیبانی از نگارش‌های قدیمی را خدشه‌دار نماید و این خود درجین فرآیند تولید و حتی خطایابی و تست برنامه، مشکلاتی را ایجاد می‌نماید و مدیریت آن کار دشواری است.

همانند دیگر مفاهیمی که در مورد SQLite بیان شد می‌توان گفت که خطاهای کمتر به معنی مشکلات برنامه‌نویسی کمتر و کاهش نگرانی در مورد آنها است و اطمینان بیشتر به معنی محبوبیت و کاربردی شدن محصول SQLite است.

خلاصه

در این فصل به توضیحی پیرامون SQLite پرداختیم. با معماری آن آشنا شده و تفاوت آن با دیگر سیستم‌های مرسوم مدیریت پایگاه‌داده‌ی رابطه‌ای را بحث نمودیم. ویژگی‌هایی که توسط SQLite ارائه می‌شود را بررسی و مزایا و معایب استفاده از آن را اشاره نمودیم. در فصل بعدی به بررسی موارد کاربرد SQLite خواهیم پرداخت.

فصل 2

کاربرد SQLite

SQLite هم در اجرا و هم در استفاده، به عنوان یک محصول انعطاف پذیر شناخته می شود. در این فصل در مورد کاربردهایی که SQLite می تواند برای ما به همراه داشته باشد صحبت خواهیم نمود. برخی از این نقش ها همانند سیستم های RDBMS¹ مرسوم کلاینت/سروری است و بخشی در ارتباط با محدوده و سادگی استفاده از SQLite است که شامل راه حل هایی می شود که به طور کامل در معماری پایگاه داده کلاینت/سرور قرار نمی گیرد.

یک پایگاه داده کوچک

سالها تجربه این موضوع را به برنامه نویسان آموخت که یک سکوی کاری RDBMS مبتنی بر رفتار کلاینت/سروری، یک مکانیزم قدرتمند برای ذخیره سازی و دستکاری امن و سازمان یافته ی اطلاعات است. متأسفانه بیشتر محصولات بزرگ RDBMS نیاز به منابع گسترده ی سخت افزاری داشته و برای نگه داری، هزینه بر هستند. آنها عملکرد و ظرفیت ما را افزایش می دهند، ولی این موضوع را، که در کجا و به چه صورت می توان سیستم را پیاده سازی و مستقر نمود، محدود می کنند.

SQLite برای پر نمودن این فضای خالی طراحی و تولید شده است. ابزارهایی قدرتمند و مشابه را برای ذخیره سازی و دستکاری امن و سازماندهی شده برای داده ارائه می دهد که برای استفاده در محیط های محدودتر از نظر منابع سخت افزاری بسیار مناسب است.

SQLite برای تکمیل و نه جایگزینی سیستم های بزرگ RDBMS طراحی شده و در شرایطی که سادگی و پیاده سازی بیش از ظرفیت مورد نیاز است کاربرد دارد. این نقش مکمل، برنامه ها و ابزارها را قادر می سازد تا از امکان مدیریت داده های رابطه ای² و سالها تجربه ی مرتبط با آن بهره مند شوند؛ حتی اگر محیط اجرا فاقد یک سیستم مدیریتی متمرکز باشد. مسلماً نیاز به نصب MySQL بر روی کامپیوتر و یا حتی یک تلفن همراه، تنها برای استفاده از یک برنامه ی دفترچه تلفن، خنده دار خواهد بود. با استفاده از SQLite دیگر نیاز به انجام این فرآیند نصب وجود نخواهد داشت.

فایل های برنامه

امروز برنامه های مدرن کامپیوتری به طور معمول با تعداد قابل توجهی فایل ارائه می شوند. این نوع برنامه ها به صورت پیش فرض یک یا چند فایل پیکربندی³ و تنظیمات دارند. این فایل ها معمولاً برای ذخیره سازی اطلاعات پیکربندی

¹ Relational Database Management System

² Relational Data

³ Configuration Files

12/ فصل دوم: کاربرد SQLite

در سطح سیستم و یا کاربر، داده‌های موجود در حافظه‌ی پنهان و همچنین انواع دیگر داده‌های مورد نیاز، استفاده می‌شوند.

استفاده از SQLite به‌عنوان یک لایه‌ی انتزاعی می‌تواند یک مزیت محسوب شود. زیرا حجم بالایی از داده‌های موجود در یک برنامه همانند داده‌های موجود در حافظه‌ی پنهان، داده‌های ارائه دهنده‌ی وضعیت برنامه و داده‌های پیکربندی، در قالب یک مدل داده‌ای رابطه‌ای قابل مدیریت هستند. استفاده از SQLite می‌تواند روال نگاشت این نوع داده‌ها را به مدل داده‌ای درون برنامه، ساده‌تر نماید.

SQLite همچنین می‌تواند در قالب یک فرمت خاص از اسناد، مورد استفاده قرار گیرد. به جای ایجاد یک فرمت جدید برای ارائه‌ی اسناد در برنامه‌ی خود، می‌توانید از فایل‌های پایگاه‌داده‌ی SQLite برای ارائه‌ی فایل سند دلخواه استفاده کنید. SQLite از انواع فراوانی "نوع داده" پشتیبانی می‌نماید، مواردی مانند متون یونیکد¹ و همچنین داده‌های باینری که می‌توانند برای ذخیره‌ی تصاویر و یا داده‌های خام مورد استفاده قرار گیرند.

حتی اگر در برنامه، نیازمندی قوی برای استفاده از ذخیره‌سازی رابطه‌ای دیده نشود، هنوز هم مزایای قابل توجهی برای استفاده از کتابخانه‌ی SQLite به‌عنوان یک ظرف ذخیره‌سازی وجود دارد.

کتابخانه‌ی SQLite روال افزایشی برای به‌روزرسانی ارائه می‌دهد که ذخیره کردن تغییرات کوچک را سریع و آسان می‌نماید. سیستم تراکنشی آن، تمامی فرآیندهای I/O در ارتباط با فایل را مدیریت نموده و صدمه دیدن فایل‌ها را در مقابل رخدادهایی مانند قطع برق و یا متوقف نمودن پردازش (به‌صورت پیش‌بینی نشده) محافظت می‌نماید. همچنین SQLite دارای لایه‌ای داخلی برای cache نمودن فایل می‌باشد. بدین ترتیب فایل‌های حجیم و بزرگ می‌توانند بدون تغییر محسوس در استفاده از RAM، مورد پردازش قرار گیرند و این کار بدون نیاز به انجام فعالیتی خاص در سمت برنامه روی می‌دهد.

فایل‌های پایگاه‌داده‌ی SQLite مستقل از سکوی کاری بوده و به‌آسانی، قابلیت جابجایی را دارا می‌باشند. محتوای یک فایل می‌تواند به‌آسانی و بدون دردسر و به‌صورت امن، میان برنامه‌های مختلف به اشتراک گذاشته شود؛ بدون نگرانی در مورد اینکه دیگر برنامه‌ها آیا قادر به استفاده از داده‌های موجود در آن هستند یا خیر. وجود یک روال عمومی برای کار با فایل، فرآیند خطایابی و همچنین تهیه‌ی ابزارهایی جانبی برای کار با آن را به راحتی فراهم خواهد نمود. همچنین چندین برنامه می‌توانند به صورت همزمان به فایل دسترسی پیدا کرده و موتور پایگاه‌داده، وظیفه‌ی مدیریت، cache کردن و تراکنش‌های مورد نیاز را بر عهده خواهد داشت. استفاده از SQLite همچنین روال خطایابی و اشکال‌زدایی را ساده می‌نماید. زیرا فایل توسط یک ابزار استاندارد در ارتباط با پایگاه‌داده مدیریت می‌گردد که بدون خطا است. همچنین می‌توانید از ابزارهای جانبی برای کار با فایل پایگاه‌داده در کنار برنامه‌ی کاربردی استفاده نمایید. به‌طور مشابه فایل‌های پایگاه‌داده می‌تواند توسط برنامه به‌صورت یک آبجکت، تولید و مورد استفاده قرار گیرد.

¹ Unicode

استفاده از یک نمونه پایگاه داده برای ارائه‌ی یک سند می‌تواند تا حدودی غیر معمول به نظر برسد؛ ولی به‌عنوان یک ایده، قابل بررسی است. مزیت این روش بسیار روشن است. برنامه‌نویس به‌جای نگران بودن در مورد فرمت فایل، کش کردن داده‌ها و یا همزمان‌سازی¹، می‌تواند تمرکزش را بر روی هسته و عملکرد اصلی سیستم گذارد و فرآیند تولید برنامه را سرعت دهد.

حافظه‌ی پنهان برنامه

SQLite دارای قابلیت است که می‌تواند یک پایگاه داده را به‌طور کامل درون حافظه ایجاد نماید. این فرآیند برای تولید یک پایگاه داده‌ی موقت و کوچک که نیاز به ذخیره‌سازی دائمی ندارد مناسب و کاربردی است.

پایگاه داده‌های درون حافظه، بیشتر برای کش کردن نتایج استخراج شده از سیستم‌های مرسوم RDBMS مورد استفاده قرار می‌گیرند. یک برنامه ممکن است زیرمجموعه‌ای از داده‌ها را از یک سرور راه دور² دریافت کرده و آن را درون یک پایگاه داده‌ی موقت قرار دهد. سپس عملیات پردازش و یا جست‌وجوی خود را به دفعات، بر روی این داده‌های محلی انجام دهد. این روال در مواردی که پاسخ‌دهی سریع مدنظر است، همانند ارائه‌ی پیشنهاد در حین تایپ عبارات، کاربرد دارد.

پایگاه داده‌های موقت همچنین می‌توانند برای ایندکس‌گذاری و لینک کردن داده‌هایی که دارای ارجاعات فراوان باشند مورد استفاده قرار گیرند. بدین ترتیب، به‌جای نیاز به تولید نوع داده‌های پیچیده برای مدیریت این ساختارهای داده‌ای که شامل مواردی همچون Hash Table، درخت‌ها و اشاره‌گرها می‌باشد، برنامه‌نویس می‌تواند یک پایگاه داده طراحی و داده‌ها را درون آن بارگذاری نماید.

اگرچه ممکن است استفاده از عبارات SQL برای استخراج داده‌ها از اطلاعات موجود در حافظه، کار عجیبی به نظر برسد ولی افزون بر بالا بردن سرعت و کارایی، می‌تواند روال تولید برنامه را سریع‌تر و پیچیدگی آن را کاهش دهد. همچنین یک پایگاه داده، قابلیت افزایش حجم دارد و این کار توسط مکانیزم‌هایی انجام می‌شود که می‌تواند به صورت موقت پایگاه داده را در فایل نیز ذخیره نماید و بدین ترتیب، کمبود RAM، محدودیتی برای حجم پایگاه داده‌ی درون حافظه نخواهد بود.

آرشیو نمودن و انباری داده

SQLite این امکان را فراهم می‌آورد که مجموعه‌ای از داده‌های پیچیده، به‌آسانی در قالب یک فایل با قابلیت دسترسی بالا و مستقل از سکوی کاری ذخیره گردند. داشتن تمامی اطلاعات درون یک فایل، اشتراک‌گذاری، تکثیر و حتی بارگذاری حجم بالایی از جداول را برای مواردی مانند واژه‌نامه و یا مجموعه‌ای از نقاط جغرافیایی، ساده می‌نماید.

¹ Synchronization

² Remote

برخلاف بسیاری از محصولات RDBMS، کتابخانه‌ی SQLite قادر است به فایل پایگاه‌داده در قالب تنها خواندنی متصل شود. این مکانیزم برای سیستم‌هایی که از لحاظ فضای ذخیره‌سازی در محدودیت هستند همانند بازی‌های ویدئویی مفید خواهد بود.

جایگزین معماری کلاینت

SQLite می‌تواند به‌عنوان یک گزینه‌ی مکمل برای حالاتی که سیستم‌های مرسوم RDBMS مورد استفاده قرار می‌گیرند به کار برده شود. به‌ویژه در مورد ارزیابی و نمایش امکانات نرم‌افزارهایی که برای فعالیت، وابسته به پایگاه‌داده می‌باشند.

یک محصول آنالیز داده را در نظر بگیرید که داده‌ها را از یک پایگاه‌داده‌ی رابطه‌ای استخراج و برای تولید گزارش‌ها و گراف‌ها، مورد استفاده قرار می‌دهد. ارائه کردن روال برای دانلود نسخه‌ی آزمایشی برنامه و چک کردن و نمایش امکانات آن، کار چندان آسانی نخواهد بود. زیرا اگر قابلیت دانلود مهیا باشد برنامه می‌بایست برای اتصال به پایگاه‌داده و استخراج اطلاعات تستی از آن، پیکربندی و تنظیم گردد و روال‌های اعتبارسنجی ورود کاربر به پایگاه‌داده نیز مورد نیاز خواهد بود. اینها مواردی است که برای مشتریان نه چندان حرفه‌ای، موضوع مطلوبی نخواهد بود.

اینک فرض کنید برنامه در نسخه‌ای دارای قابلیت اتصال به پایگاه‌داده‌ی SQLite برای کار با داده‌های تستی باشد. با دانلود و اجرای برنامه، مشتری می‌تواند با قابلیت‌های نرم‌افزار و شیوه‌ی کارکردن با آن آشنا شود. در روال‌هایی از این دست که نیاز به اتصال به پایگاه‌داده و استفاده از آن برای فعالیت برنامه لازم است، استفاده از یک مکانیزم سریع و بدون پیچیدگی برای حل مشکل می‌تواند یکی از کاربردهای SQLite باشد.

ابزار آموزش

برای دانش‌پژوهانی که علاقمند به آموزش SQL و مبانی مدل رابطه‌ای هستند، SQLite یک محیط مناسب تهیه نموده است که هم به سادگی قابل نصب می‌باشد و هم به‌آسانی قابلیت استفاده و به اشتراک گذاری را ارائه می‌دهد. SQLite یک سیستم رابطه‌ای را به‌طور کامل پشتیبانی نموده و تقریباً تمامی ویژگی‌های زبان SQL را پشتیبانی می‌نماید. به هیچ پیش‌فرض خاصی برای اجرا نیاز ندارد و همچنین بدون نیاز به سرور و تنظیمات مدیریتی است. این ویژگی باعث می‌شود که دانش‌پژوه بتواند به‌صورت مستقیم بر روی یادگیری SQL و دستکاری داده‌ها، بدون نیاز به تنظیمات اضافی تمرکز نماید.

حجم کم، وجود نسخه‌های مربوط به تمامی سیستم‌عامل‌ها از جمله Windows، Mac OS X و Linux در کنار ابزارهای کمکی موجود، عدم نیاز به فرآیند نصب و وجود قابلیت مستقل از سکوی کاری بودن، ویژگی‌هایی است که در کنار هم SQLite را به ابزاری مناسب برای آموزش تبدیل نموده است. همچنین امتیاز وجود پایگاه‌داده در یک فایل، به اشتراک‌گذاری نتیجه‌ی کار دانش‌پژوه و همچنین تهیه‌ی نسخه‌ی پشتیبان از آن را به سهولت مهیا نموده است.

موتور SQL

جداول مجازی SQLite این امکان را به برنامه‌نویس می‌دهند که محتوای جداول را درون کد تعریف نماید. با تعریف مجموعه‌ای از توابع قابل فراخوانی¹ که داده‌ها را استخراج و باز می‌گرداند، می‌توان ارتباطی بین موتور پردازش داده و منبع داده تولید نمود. این قابلیت به SQLite این امکان را می‌دهد که پرس‌وجوهای خود را بر روی منابع اطلاعات انجام دهد؛ بدون آنکه داده‌ها را در جداول استاندارد قرار داده باشد.

جداول مجازی، یک راه کاربردی مناسب برای تولید گزارش‌ها و یا انجام پرس‌وجوهای موردی بر روی لاگ‌ها و یا هر مجموعه داده‌ای با ساختاری جدولی است. به جای تولید مجموعه‌ای از جست‌وجوهای سفارشی و یا ابزارهای گزارش‌سازی، داده‌ها می‌توانند به طور مستقیم برای موتور SQLite ارسال شوند و این بدان معنی است که اجرای پرس‌وجوها با استفاده از SQL انجام می‌گیرد. SQL زبانی است که بیشتر برنامه‌نویسان با آن آشنایی دارند. بنابراین این امکان فراهم می‌شود که از یک ابزار شناخته شده و عمومی (Generic) یعنی همان زبان SQL، برای تعامل با پایگاه داده و تولید گزارش استفاده گردد و این کار دارای مزایای بسیاری است.

کاربردهای نامناسب

اگرچه SQLite به صورت بسیار انعطاف‌پذیر طراحی شده است اما مواردی نیز وجود دارد که خارج از اهداف طراحی آن است. با وجود اینکه SQLite را می‌توان در این موارد نیز مورد استفاده قرار داد ولی انتخاب چندان مناسبی نخواهد بود. چنانچه در برنامه‌های خود به یکی از مواردی که در ادامه معرفی می‌شود نیاز داشتید بهتر است از محصولات مشابه دیگر RDBMS که از معماری کلاینت/سروری پشتیبانی می‌کند استفاده نمایید.

حجم بالای تراکنش

اگر چه SQLite حجم قابل قبولی از تراکنش‌ها را پشتیبانی می‌کند ولی برای شمار بالای تراکنش‌های همزمان² که توسط دیگر RDBMS‌ها پشتیبانی می‌گردد طراحی نشده است. بیشتر این نوع سیستم‌ها قابلیت قفل‌گذاری در سطح جدول و یا سطر را نیز دارا هستند و این امکان را فراهم می‌آورند که چندین تراکنش به صورت همزمان و بدون نگرانی درمورد از دست دادن داده، اعمال شوند.

مدیریت همزمانی ارائه شده در SQLite برای حفاظت از داده‌ها در سطح فایل ارائه می‌شود. در این روش، کل پایگاه داده برای خواندن توسط چند تراکنش به صورت همزمان قابل دسترسی می‌باشد ولی برای اعمال تغییرات در پایگاه داده، کل آن می‌بایست قفل شود. در نتیجه تراکنش‌های نوشتنی به صورت یک لیست در پشت سر هم قرار می‌گیرند که این فرآیند، نرخ کلی تراکنش‌های سیستم را کاهش می‌دهد.

¹ Callback

² Concurrent

بنا به میزان و پیچیدگی به روزرسانی، SQLite قادر به مدیریت چندصد تراکنش از پردازش‌ها و یا Threadهای متفاوت در دقیقه می‌باشد. اما اگر شما به نرخ تراکنش بالاتری نیاز دارید، دیگر سیستم‌های مرسوم کلاینت/سروری بهره‌وری بالاتری را ارائه می‌دهند.

حجم بالای داده‌ها

این نکته‌ی عجیبی نخواهد بود که با یک پایگاه‌داده SQLite با حجم چندین گیگابایت یا بیشتر مواجه شوید. اما برخی محدودیت‌های عملی در مورد میزان دیتای قابل ارائه‌ی درون پایگاه‌داده‌ی SQLite وجود دارد. از آنجا که SQLite همه چیز را درون یک فایل قرار می‌دهد، حجم بالای داده به معنای فشار بر روی سیستم فایل¹ و سیستم‌عامل خواهد بود. اگر چه سیستم فایل‌های جدید، حجم بالا تا حدود ترا بایت و بیشتر را پشتیبانی می‌کنند ولی این به معنای خوب بودن روش نیست. بسیاری از این سیستم فایل‌ها به دلیل بالا رفتن حجم فایل و الگوهای خاص دسترسی به محتوای فایل در حجم‌های چند گیگا بایتی، دچار افت کارایی و بهره‌وری می‌گردند. چنانچه تخمین حجم داده‌ی شما چندین گیگا بایت باشد بهتر است از سیستم‌های پایگاه‌داده‌ی دیگر استفاده نمایید.

کنترل دسترسی

SQLite مکانیزم‌های احراز هویت² و مجوزدهی³ ندارد. در مقابل برای شناسایی مجوز، از حق دسترسی مربوط به فایل پایگاه‌داده استفاده می‌نماید. در این حالت محدودیت دسترسی در یکی از این سه سطح تعریف می‌گردد.

- دسترسی برای خواندن و نوشتن
- دسترسی تنها برای خواندن
- نداشتن دسترسی

چنانچه دسترسی به صورت کامل باشد امکان نوشتن در پایگاه‌داده و یا ویرایش آن وجود خواهد داشت. چنانچه این میزان کنترل دسترسی مطابق با خواست شما نباشد بهتر است از نمونه‌های دیگر RDBMS برای برنامه‌ی خود استفاده نمایید.

کلاینت/سروری

SQLite بدون کامپوننت‌های مربوط به شبکه ارائه شده است و تنها برای استفاده از منابع محلی مناسب است. هیچ پشتیبانی طبیعی برای دسترسی به کامپیوترها از طریق شبکه در آن دیده نشده است و این، استفاده از آن در معماری‌های کلاینت/سروری را غیرممکن می‌سازد.

¹ Filesystems

² Authentication

³ Authorization

حتی دسترسی چند سیستم به فایل پایگاه داده از طریق به اشتراک گذاری آن مسئله دار است. زیرا برخی از سیستم فایل‌هایی که در شبکه استفاده می‌شوند دارای قابلیت قفل گذاری قدرتمند برای فایل نیستند و این به معنی پتانسیل از دست رفتن داده‌ها و خراب شدن فایل پایگاه داده است.

البته این به معنی آن نیست که سیستم‌های کلاینت/سروری از SQLite در معماری خود استفاده نمی‌کنند. برای نمونه، بسیاری از وب‌سرویس‌ها از این پایگاه داده برای انجام فعالیت و سرویس‌دهی بهره می‌برند زیرا کلیه پردازش‌های وب سرویس بر روی یک سیستم انجام می‌پذیرد و دسترسی به داده‌ها نیز به صورت محلی است.

Replication

SQLite به صورت داخلی از Replication پشتیبانی نمی‌کند. این روال می‌تواند به سادگی توسط کپی کردن فایل پایگاه داده انجام شود. البته این کار باید در زمانی انجام گردد که هیچ پردازشی در حال ویرایش آن نباشد.

سیستم‌های Replication بر روی API‌های پایه‌ی پایگاه داده بنا می‌شوند ولی مشکلات مربوط به خود را نیز دارا هستند. در کل، چنانچه در طراحی خود نیاز به استفاده از این نوع سرویس را دارید باید به سراغ RDBMS‌های پیچیده‌تر بروید.

بیشتر نیازهایی که تا اینجا بیان شد باعث بالا رفتن پیچیدگی و سرباره‌های¹ مدیریتی می‌شوند و این به معنی نیاز به استفاده از یک سیستم بزرگ پایگاه داده است. در مقابل، SQLite تلاش در ساده‌تر کردن موضوع و حذف مواردی دارد که برای سیستم ساده مورد نیاز نیست. اما مهم‌ترین موضوع، انتخاب ابزار مناسب برای جلوگیری از مشکلات آینده است.

برخی از استفاده کنندگان معروف

سایت SQLite عنوان می‌کند که "SQLite از نظر گستردگی استفاده، دارای بالاترین رتبه در بین موتورهای پایگاه داده می‌باشد". این ادعایی بزرگ است! به‌ویژه با این نگرش که در حین صحبت در مورد پایگاه داده‌های رابطه‌ای، عموماً فکر افراد به سمت محصولاتمانند Oracle، SQL Server و MySQL می‌رود.

همچنین در مورد تعداد هم می‌توان این صحبت را تنها یک ادعا در نظر گرفت، زیرا به صورت دقیق نمی‌توان تعداد کاربران را مشخص نمود. از آنجا که لیسانس و یا مجوزی برای استفاده از SQLite وجود ندارد، آمار دقیقی از تعداد کاربران آن نیز نمی‌توان ارائه داد. هیچ کس حتی اعضای تیم توسعه‌ی SQLite نیز از اینکه چه کسانی و به چه منظوری از SQLite استفاده می‌کنند خبر ندارد.

¹ Overhead

با وجود این می‌توان تا حدودی لیستی از استفاده‌کنندگان از SQLite ارائه داد. مرورگر Firefox و برنامه Thunderbird، هر دو از SQLite برای ذخیره‌ی کوکی‌ها¹، تاریخچه، تنظیمات کاربری و دیگر داده‌ها استفاده می‌نمایند.

محصولات زیادی از Adobe، Skype و McAfee نیز از موتور SQLite بهره می‌برند. همچنین کتابخانه‌ی SQLite با برخی از زبان‌های اسکریپت‌نویسی همانند PHP و Python یکپارچه شده است. کمپانی Apple به‌صورت گسترده‌ای از SQLite در محصولات خود مانند iPad و iPhone استفاده می‌نماید. BlackBerry، Android، Symbian و دیگر انواع سیستم‌عامل‌ها به صورت گسترده‌ای از آن پشتیبانی می‌نمایند. آنچه عنوان شد اگر به بلیون نرسد، در حد میلیون نمونه از پایگاه‌داده‌ی SQLite خواهد بود که مورد استفاده قرار می‌گیرد. برخی از این نمونه‌ها ممکن است تنها چند صد کیلوبایت داده در خود داشته باشند و این همان چیزی است که SQLite برای آن تولید شده است.

سیستم‌های بزرگ کلاینت/سروری RDBMS که توسط هزاران برنامه‌نویس تولید شده‌اند قدرت سیستم‌های پایگاه‌داده‌ی رابطه‌ای را نشان می‌دهد و SQLite برای نمایش این قدرت، در ابزارهای کوچک و موبایل و سیستم‌های رومیزی به‌جای محبوب کردن آن در اتاق‌های سرور، تولید شده است.

خلاصه

در این فصل در مورد کاربردهای SQLite صحبت شد. مزایا و معایب آن در مقابل دیگر سیستم‌های RDBMS که به‌عنوان ابزارهایی حرفه‌ای برای کار با پایگاه‌داده‌های رابطه‌ای شناخته می‌شوند ارائه گردید تا بتوانید تصمیم بگیرید در چه دسته از برنامه‌های خود از آن استفاده نمایید. در فصل بعدی در مورد شیوه‌ی به‌دست آوردن منابع مورد نیاز برای استفاده از SQLite صحبت خواهیم نمود.

¹ Cookies

فصل 3

تهیه و نصب SQLite

در این فصل در مورد شیوهی تهیه و نصب SQLite صحبت خواهیم نمود. همان‌گونه که گفته شد SQLite یک پروژه‌ی متن باز بوده و به آسانی می‌توانید بدون نیاز به هیچ نوع لیسانس و مجوزی از آن استفاده نمایید. کد پایه‌ی ارائه شده در SQLite بیشتر سیستم‌عامل‌های مرسوم را پشتیبانی کرده و کدها و کتابخانه‌های کامپایل نشده و همچنین فایل اجرایی آن برای تمامی سکوهای کاری شامل Linux، Mac OS X و Windows در وب سایت SQLite موجود است.

همان‌گونه که گفته شد تمامی موارد قابل دانلود از جمله سورس کد و فایل قابل اجرا و کتابخانه‌های مورد نیاز، در صفحه دانلود موجود در وب سایت SQLite همانند شکل 1-3 به آدرس زیر در دسترس می‌باشد.

<http://www.sqlite.org/download.html>

Source Code

sqlite-amalgamation-3080402.zip (1.42 MiB)	This ZIP archive contains all C source code for SQLite 3.8.4.2 combined into a single source file (the amalgamation). (sha1: 511422bc5bb6d7ce36751f9f51ce2b0fa4f1d987)
sqlite-autoconf-3080402.tar.gz (1.85 MiB)	A tarball containing the <i>amalgamation</i> for SQLite 3.8.4.2 together with an configure script and makefile for building it. This tarball also contains in the "tea" subdirectory a separate configure script and makefile compatible with the <i>Tcl Extension Architecture (TEA)</i> for building the TCL bindings to SQLite. (sha1: 97d727b46ce67c90eb7e0fa3f57a1cfeb7f08f74)

Documentation

sqlite-doc-3080402.zip (4.41 MiB)	This ZIP archive contains most of the static HTML files that comprise this website, including all of the SQL Syntax and the C/C++ interface specs and other miscellaneous documentation. (sha1: 3f615f9fa81b737e652b06eddf8307235d84f2c)
--	---

Precompiled Binaries for Linux

sqlite-shell-linux-x86-3080402.zip (316.02 KiB)	A <i>command-line shell</i> for accessing and modifying SQLite databases. This program is compatible with all versions of SQLite through 3.8.4.2 and beyond. (sha1: c95d1180efa3db11de3ab09c80d704d938be5376)
sqlite-analyzer-linux-x86-3080402.zip (939.97 KiB)	An analysis program for database files compatible with all SQLite versions through 3.8.4.2 and beyond. (sha1: 37d312f9a5a8977b0e864016d6e0d67ca0ce7304)

Precompiled Binaries for Mac OS X (x86)

شکل 1-3 صفحه دانلود مربوط به سایت SQLite

همچنین برای دریافت کد مربوط به تمرینات کتاب، اطلاعات بیشتر، دریافت فیلم‌های آموزشی، راهنمایی و بیان پرسش‌هایتان می‌توانید از سایت مؤلف به آدرس زیر و یا صفحه‌ی این کتاب در سایت انتشارات پندارپارس استفاده نمایید.

<http://www.programmerf1.com>

محصولات SQLite

پروژه‌ی SQLite شامل چهار محصول اصلی می‌باشد.

هسته‌ی SQLite

هسته‌ی SQLite شامل موتور اصلی پایگاه‌داده به همراه مجموعه API‌های عمومی می‌باشد. این هسته می‌تواند برای تولید کتابخانه‌ی ایستا و یا پویا استفاده شود و یا به صورت مستقیم در برنامه‌ی سرویس‌گیرنده اضافه شود. شکل 2-3 لینک دانلود هسته‌ی SQLite برای سیستم‌عامل ویندوز را که در صفحه‌ی دانلود سایت SQLite قرار دارد، نمایش می‌دهد.

[sqlite-dll-win32-x86-3080402.zip](#) (323.29 KiB) This ZIP archive contains a DLL for the Win32 API. The DLL is built using Ruby on Rails. (sha1: e518343bb2a6e333d8096aa3)

شکل 2-3 لینک دانلود کتابخانه SQLite

ابزار خط فرمان sqlite3

برنامه‌ی sqlite3 یک ابزار خط فرمان¹ می‌باشد که بر روی هسته‌ی SQLite قرار می‌گیرد. با این ابزار، برنامه‌نویس می‌تواند به وسیله‌ی دستورهای SQL با هسته‌ی SQLite در تعامل باشد. این ابزار برای تولید و خطایابی پرس‌وجوها ابزار بسیار مناسبی است. شکل 3-3 لینک دانلود این پوسته² را برای سیستم‌عامل ویندوز، که در صفحه‌ی دانلود سایت SQLite قرار دارد، نمایش می‌دهد.

[sqlite-shell-win32-x86-3080402.zip](#) (289.42 KiB) A command-line shell for accessing and modifying versions of SQLite through 3.8.4.2 and beyond. (sha1: ca9b09765596bc42eb7433c89c366e3)

تصویر 3-4 لینک دانلود برنامه sqlite3

افزونه‌ی Tcl

SQLite به صورت تاریخی با زبان Tcl در ارتباط بوده است. این کتابخانه، یک کپی از هسته‌ی SQLite می‌باشد که به Tcl متصل شده است. پس از کامپایل کردن برنامه، خروجی این مرحله کتابخانه‌ای است که واسطی از SQLite را

¹ Command-Line Tool

² Shell

برای زبان Tcl با استفاده از معماری افزونه‌ی Tcl ارائه می‌دهد. خارج از محدوده‌ی APIهای زبان C، این اتصال Tcl تنها یک واسط برنامه‌نویسی رسمی است که مستقیماً توسط تیم توسعه‌ی SQLite پشتیبانی می‌شود.

ابزار آنالیز SQLite

این ابزار برای آنالیز پایگاه‌داده مورد بررسی قرار می‌گیرد و آمار و ارقام و اطلاعاتی را درباره‌ی سبک فایل پایگاه‌داده، قطعه‌بندی¹ آن، فضای خالی موجود در آن و دیگر داده‌هایی از این دست ارائه می‌دهد. این ابزار برای عملیات اشکال‌زدایی، بهره‌وری و مدیریت ساختار فیزیکی پایگاه‌داده بسیار سودمند است.

بیشتر برنامه‌نویسان در ابتدا از دو محصول نخست، یعنی هسته‌ی SQLite و ابزار خط فرمان Sqlite3 استفاده می‌نمایند. در فصل‌های دیگر کتاب، در مورد این دو محصول صحبت خواهد شد. فرآیند کامپایل کردن افزونه‌ی TCL وابسته به این است که هسته‌ی SQLite را به صورت یک کتابخانه‌ی دینامیک، کامپایل نمایید. ابزار آنالیز نیز عموماً به صورت از پیش کامپایل شده ارائه می‌گردد؛ ولی شما می‌توانید سورس کد آن را دانلود و بر اساس نیاز خود آن را کامپایل نمایید. برای استفاده از این ابزار می‌بایست روال مدون و مدیریت شده‌ای در پروژه‌ی خود داشته باشید. شکل 3-4 لینک دانلود این برنامه را برای سیستم‌عامل ویندوز، که در صفحه‌ی دانلود سایت SQLite قرار دارد، نمایش می‌دهد.

[sqlite-analyzer-osx-x86-3080402.zip](#) An analysis program for database files
(sha1: ebe24b747b2a7c7301af258d65f...)
(814.84 KIB)

تصویر 3-4 لینک دانلود برنامه آنالیزگر SQLite

نسخه‌های از پیش کامپایل شده

صفحه‌ی مربوط به دانلود SQLite شامل نسخه‌ی کامپایل شده‌ی ابزار خط فرمان sqlite3 برای سیستم‌عامل‌های لینوکس، ویندوز و Mac OS X می‌باشد. چنانچه قصد دارید تجربه‌ی خود را با SQLite آغاز کنید می‌توانید ابزار خط فرمان را دانلود نموده، پس از خارج کردن آن از حالت فشرده و اجرای آن، دستورهای SQL را بر روی آن تست نمایید. در مورد سیستم‌عامل Mac OS X و بیشتر نسخه‌های لینوکس، حتی نیاز به دانلود ابزار نیست زیرا sqlite3 قابل بخشی از سیستم‌عامل، از پیش نصب شده است. صفحه‌ی دانلود SQLite همچنین شامل نسخه‌ای از ابزار آنالیز تحت عنوان sqlite3_analyzer برای سه نوع سیستم‌عامل بیان شده نیز می‌باشد.

نسخه‌های از پیش کامپایل شده‌ی هسته‌ی SQLite و همچنین افزونه‌ی TCL برای سیستم‌عامل ویندوز و لینوکس برای دانلود وجود دارد. فایل‌های لینوکسی در قالب فایل‌های با پسوند .so و فایل‌های ویندوزی با پسوند .dll ارائه

¹ Fragmentation

22/ فصل سوم؛ تهیه و نصب SQLite

می‌شوند. فایل از پیش کامپایل شده برای Mac OS X وجود ندارد. این کتابخانه‌ها در مواردی کاربرد دارند که قصد ایجاد یک برنامه را داشته باشید و نخواهید کدهای هسته‌ی SQLite را به صورت مستقیم درون برنامه‌ی خود وارد نمایید.

مستندات

صفحه دانلود SQLite همچنین شامل مستندات مورد نیاز شما نیز می‌باشد. فایل `sqlite-doc-x.zip` (تصویر 2-3) حاوی یک کپی از مطالبی است که در سایت SQLite ارائه شده است. محتوای سایت همیشه با توجه به آخرین به‌روزرسانی‌های محصولات تغییر می‌کند و دارای ساختاری برای نمایش مستندات نگارش‌های مختلف نیست. به همین دلیل توصیه می‌شود مستندات مربوط به نگارش مورد نظر خود را از صفحه دانلود دریافت نمایید.

Documentation

[sqlite-doc-3080402.zip](#) This ZIP archive contains most of the SQLite documentation, including the SQL Syntax and the C/C++ interfaces.
(4.41 MiB) (sha1: 3f615f9fa81b737e652b0...)

تصویر 2-3 لینک دانلود فایل مستندات

سورس کدها

بیشتر پروژه‌های متن باز، روالی ارائه می‌دهند که با استفاده از آن به سورس کد پروژه دسترسی پیدا کرده، با کامپایل کردن آن، به خروجی قابل استفاده‌ی پروژه دسترسی خواهید داشت. اما SQLite به صورت دیگری عمل نموده است. از آنجا که یک روال عمومی برای استفاده از SQLite، اضافه کردن سورس کد آن به پروژه‌ی خود به صورت مستقیم است، سورس قابل دانلود به صورتی تنظیم گردیده است که به کمترین تنظیمات برای یکپارچه شدن با برنامه‌ی میزبان نیاز داشته باشد. این روال در بیشتر سورس کدها لحاظ گردیده است تا در دسر برنامه‌نویسان در تجمیع آنها با پروژه‌ی خودشان کمینه شود. بدین ترتیب اگر قصد داشته باشید خود `sqlite3` را کامپایل و ایجاد نمایید می‌بایست بخشی از کار را به صورت دستی پس از دانلود سورس کد انجام دهید.

Amalgamation

نسخه‌ی رسمی کد که ارائه می‌شود Amalgamation نام دارد که یک فایل سورس کد به زبان C است و شامل تمامی کدهای مربوط به هسته‌ی SQLite می‌باشد. این فایل حاصل ادغام چندین فایل برنامه‌نویسی مختلف است که در حدود 4 مگا بایت حجم داشته و بیش از 100000 خط کد را شامل می‌گردد. شکل 5-3 لینک دانلود این فایل را در صفحه مربوطه در سایت SQLite نمایش می‌دهد.

Source Code

sqlite-amalgamation-3080402.zip (1.42 MiB)	This ZIP archive contains all amalgamation . (sha1: 511422bc5bb6d7ce3)
sqlite-autoconf-3080402.tar.gz (1.85 MiB)	A tarball containing the amalgamation for building it. This tarball also contains a makefile compatible with the make utility. (sha1: 97d727b46ca67c98e)

شکل 5-3 لینک دانلود سورس کد SQLite

این فایل به همراه فایل هدر مربوطه‌اش تمامی آن چیزی است که شما برای کار با SQLite در پروژه‌ی خود نیاز دارید. این عملیات ادغام دارای مزایایی است. نخست اینکه همه چیز درون یک فایل قرار دارد که باعث می‌شود افزوده شدن SQLite به برنامه‌ی میزبان، به آسانی انجام شود. این کار از طریق افزودن فایل به پروژه و یا تولید یک کتابخانه‌ی مستقل و استفاده از آن در پروژه انجام می‌شود. دوم اینکه این کار، بهره‌وری را افزایش می‌دهد. برخی از فرایندهای بهینه‌سازی کامپایلرها به یک واحد ترجمه محدود می‌باشند. وجود کل ساختار درون یک فایل، روال بهینه نمودن توسط کامپایلر را سرعت می‌بخشد. با این کار میزان بهینه‌سازی در حدود 5% و یا بیشتر افزایش می‌یابد. تنها نکته‌ی منفی استفاده از یک فایل، دشواری عملیات خطایابی است. زیرا جست‌وجو در بین 100000 خط کد برای مشخص نمودن بهترین مکان برای Breakpoint کار آسانی نخواهد بود.

سورس فایل‌ها

هنگامی که با Amalgamation کار می‌کنید با چهار فایل مهم سروکار دارید.

- `sqlite3.c` : سورس فایل مربوط به هسته‌ی SQLite به همراه افزونه‌های عمومی
- `sqlite3.h` : هدر فایل که APIهای هسته‌ی SQLite را معرفی می‌نماید.
- `sqlite3ext.h` : هدر فایل افزونه که برای تولید افزونه‌های SQLite مورد استفاده قرار می‌گیرد.
- `shell.c` : سورس کد `sqlite3` که یک پوسته‌ی خط فرمانی را برای تعامل با SQLite ارائه می‌دهد.

دو فایل نخست، یعنی `sqlite3.c` و `sqlite3.h` تمام آن چیزی است که برای ادغام SQLite با برنامه‌ی خود بدان نیاز دارید. فایل `sqlite3ext.h` برای تولید افزونه‌ها و ماژول‌ها مورد استفاده قرار می‌گیرد. فایل `shell.c` حاوی سورس کد ابزار خط فرمان `sqlite3` می‌باشد. تمامی این فایل‌ها را می‌توان درون سیستم‌عامل‌های لینوکس، ویندوز و یا Mac OS X بدون نیاز به فایل‌های پیکربندی اضافی کامپایل نمود.

نصب SQLite

نخستین گام برای کار با SQLite نصب آن است. همان‌گونه که پیش‌تر نیز گفته شد، این برنامه نیازی به نصب و تنظیمات آغازین ندارد و روال نصب چیزی بیشتر از کپی کردن فایل‌های برنامه بر روی سیستم و اجرای آن نیست. برای شروع، به صفحه دانلود در وبسایت SQLite به آدرسی که پیش‌تر معرفی شده رفته و نسخه‌ی مناسب با سیستم‌عامل خود را دانلود نمایید. سیستم‌عامل در نظر گرفته شده برای کار با SQLite که در ادامه‌ی کتاب مورد

24/ فصل سوم؛ تهیه و نصب SQLite

استفاده قرار می‌گیرد، ویندوز خواهد بود. تنظیمات ذکر شده بر روی تمامی نسخه‌های این سیستم‌عامل قابل اجرا خواهد بود. برای راهنمایی در مورد نصب SQLite بر روی دیگر سیستم‌عامل‌ها، به سایت SQLite مراجعه کرده و یا از طریق سایت programmerF1 نسبت به طرح پرسش خود اقدام نمایید. برای نصب، ابتدا می‌بایست فایل‌های زیر را دانلود نمایید.

sqlite-shell-win32-*.zip

sqlite-dll-win32-*.zip

sqlite-analyzer-win32-x86-*.zip

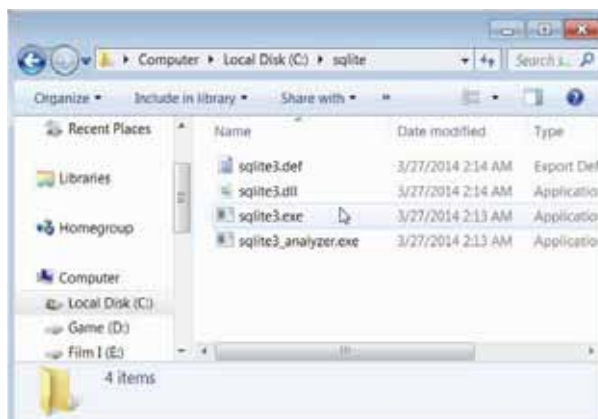
توجه داشته باشید که کاراکتر * در لیست بالا به نگارش جاری SQLite اشاره دارد که به‌عنوان مثال همان‌گونه که در شکل 6-3، لینک مربوط به دانلود فایل‌ها در صفحه دانلود سایت SQLite نشان می‌دهد، نگارش کنونی 3.8.4.2 می‌باشد که به‌صورت 3080402 آورده شده است. البته به این نکته نیز توجه داشته باشید که فایل آخر یا همان برنامه‌ی آنالیزگر برای کار با SQLite مورد نیاز نیست و می‌توانید از دانلود آن چشم‌پوشی نمایید. طبق توضیحاتی که در فصل بعدی در مورد برنامه‌ی sqlite3 ارائه خواهد شد، برای کار با آن نیاز به دانلود فایل کتابخانه‌ی SQLite نیز نخواهد بود. دانلود کتابخانه‌ی SQLite تنها در صورتی مفید خواهد بود که قصد داشته باشید از آن در برنامه‌های تولیدی خود استفاده نمایید. در فصل‌های بعدی در مورد استفاده از کتابخانه‌ی SQLite در برنامه‌نویسی، صحبت خواهد شد. با این وجود در ادامه، این فرض دنبال می‌شود که شما هر سه فایل را دانلود نموده‌اید.

Precompiled Binaries for Windows

sqlite-shell-win32-x86-3080402.zip (289.42 KIB)	A command-line shell for accessing and modifying versions of SQLite through 3.8.4.2 and beyond. (sha1: ca9b09765596bc42eb7433c89c366e3)
sqlite-dll-win32-x86-3080402.zip (323.29 KIB)	This ZIP archive contains a DLL for the SQLite3 the Win32 API. The DLL is built using SQLite3 with Ruby on Rails. (sha1: e518343bb2a6e333d8096aa365252ef)
sqlite-analyzer-win32-x86-3080402.zip (679.86 KIB)	An analysis program for database files compiled with Ruby on Rails. (sha1: b9988618f4afb372765b1ef5b6813695)

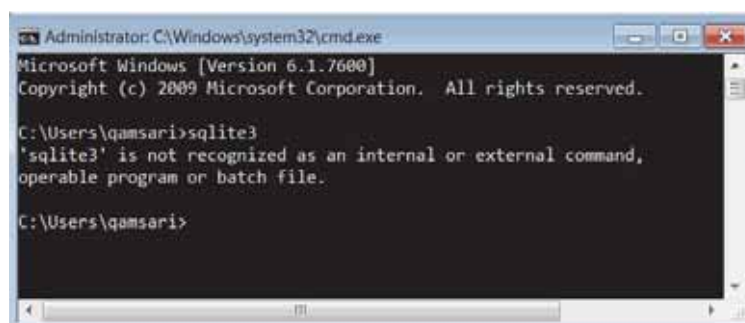
شکل 6-3 لینک فایل‌های مورد نیاز برای دانلود

یک پوشه در درایو C یا هر مسیر دلخواه دیگر با نام SQLite ایجاد و محتوای فایل‌های فشرده‌ای را که در مرحله‌ی پیش‌دانلود نموده‌اید در این پوشه کپی کنید. اکنون باید مطابق شکل 7-3 پوشه‌ی شما دارای چهار فایل `sqlite3.exe`، `sqlite3.dell`، `sqlite3.def` و `sqlite3_analyzer.exe` باشد.



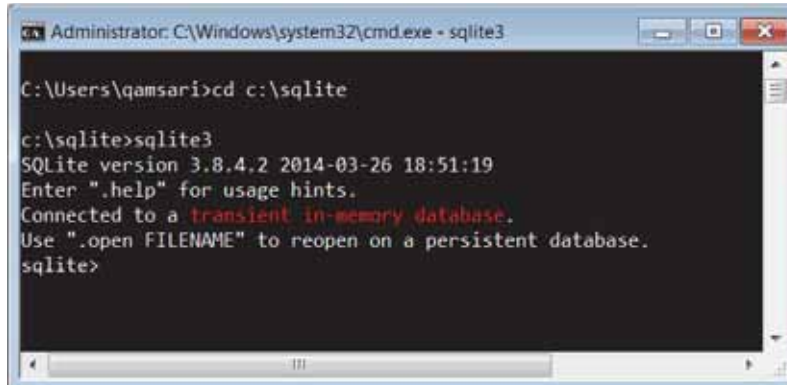
شکل 3-7 محتوای پوشه‌ی انتخابی پس از انتقال فایل‌ها

مرحله‌ی بعدی یک گام اختیاری است و می‌توانید از آن چشم‌پوشی نمایید. هدف از این گام اضافه کردن مسیر فایل اجرایی `sqlite3.exe` به متغیرهای محیطی سیستم‌عامل است تا از طریق خط فرمان DOS امکان دسترسی سریع به برنامه وجود داشته باشد؛ وگرنه برای اجرای برنامه‌ی `sqlite3` می‌بایست ابتدا به درون پوشه‌ی حاوی فایل اجرایی `sqlite3.exe` رفته و سپس `sqlite3` را تایپ و برنامه را اجرا نمایید. زیرا سیستم‌عامل از محل قرارگیری آن آگاه نیست. به شکل 3-8 توجه نمایید.



شکل 3-8 خطای اجرای `sqlite3.exe`

همان‌گونه که در شکل 3-8 مشاهده می‌نمایید، با اجرای دستور `sqlite3` خطایی نمایش داده می‌شود که نشان از عدم شناسایی دستور `sqlite3` دارد. در این حالت سیستم‌عامل از وجود یک فایل اجرایی با این نام بی‌اطلاع است. در صورتی که نخواهید مسیر فایل را به سیستم‌عامل معرفی نمایید می‌بایست همانند آنچه در تصویر 3-9 نمایش داده شده است، ابتدا به پوشه‌ی محل قرارگیری فایل `sqlite3.exe` رفته و سپس دستور `sqlite3` را اجرا نمایید.



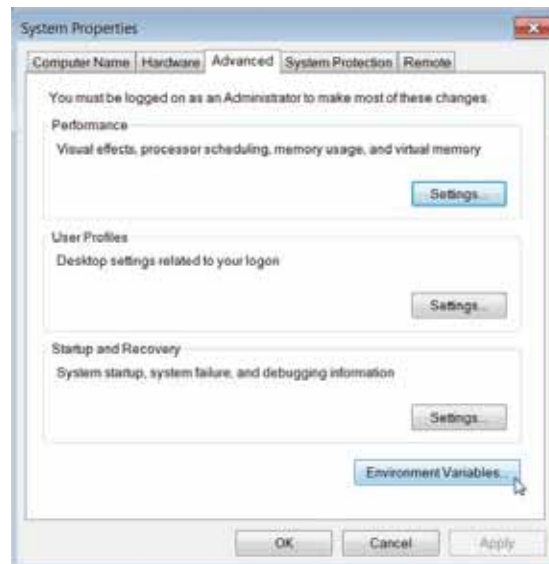
شکل 9-3 تغییر مسیر و اجرای موفقیت آمیز برنامه

همان‌گونه که در شکل 9-3 دیده می‌شود، ابتدا با استفاده از دستور cd مسیر جاری خط فرمان را به محل قرارگیری برنامه تغییر داده و سپس برنامه را اجرا می‌نماییم. همان‌گونه که مشاهده می‌نمایید استفاده از این روش چندان سریع نیست. برای اینکه در هر بار اجرای برنامه نیاز به مشخص نمودن مسیر فایل نباشد، می‌بایست این مسیر را به متغیر محیطی Path در سیستم‌عامل اضافه نمایید. برای افزودن مسیر فایل اجرای Sqlite3.exe به متغیر محیطی Path سیستم‌عامل به صورت زیر عمل نمایید.

1. در Run عبارت زیر را تایپ کرده و اجرا نمایید.

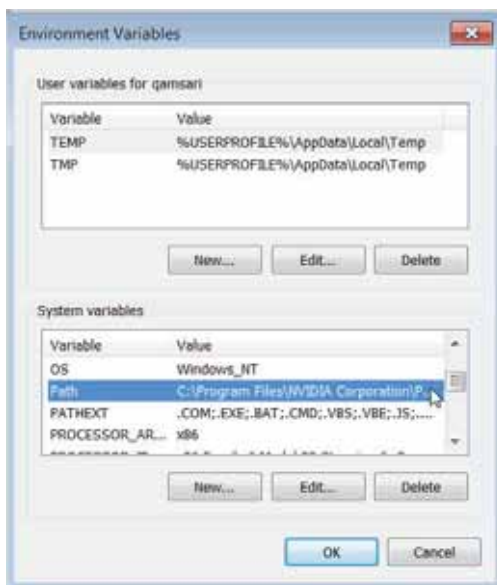
SystemPropertiesAdvanced.exe

2. فرم خصوصیات سیستم همانند شکل 10-3 نمایش داده می‌شود. در زبانه Advance بر روی دکمه‌ی Environment Variables کلیک نمایید.



شکل 10-3 فرم خصوصیات سیستم

3. فرم متغیرهای محیطی همانند شکل 3-11 نمایش داده می‌شود. در بخش System variables متغیر Path را انتخاب و بر روی دکمه‌ی Edit کلیک نمایید.

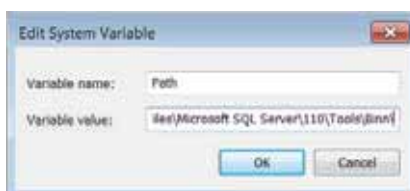


شکل 3-11 فرم متغیرهای محیطی

4. فرم ویرایش متغیر سیستمی همانند شکل 3-12 نمایان می‌شود. در بخش variable value به انتهای عبارت موجود عبارت زیر را اضافه نمایید و بر روی دکمه‌ی Ok کلیک نمایید.

;C:\Sqlite\

توجه داشته باشید که مسیر می‌بایست با مسیر قرارگیری فایل sqlite3.exe مطابقت داشته باشد که در مثال ما مسیر پیش فرض c:\sqlite\ می‌باشد. همچنین کاراکتر ; نیز پیش از مسیر تایپ شود.

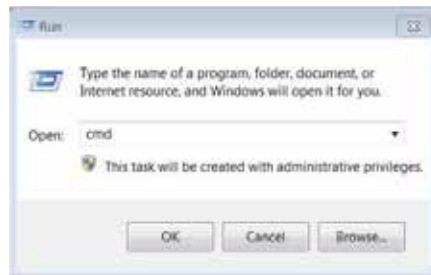


شکل 3-12 فرم ویرایش متغیر سیستمی

5. با کلیک بر روی دکمه‌ی Ok، تمامی فرم‌های باز شده را ببندید.

اکنون روال نصب SQLite به پایان رسیده است و می‌توانید آن را اجرا و از آن استفاده نمایید. البته توجه داشته باشید که چون به صورت مستقیم از SQLite استفاده می‌نماییم و برنامه‌ای برای استفاده از آن تولید نکرده‌ایم می‌بایست از ابزار خط فرمان sqlite3.exe برای تعامل با موتور پایگاه داده استفاده نماییم. گام آخر از فرآیند نصب نیز برای سادگی در کار با این ابزار بود و جزء روال اصلی نصب نیست. برای اطمینان از اینکه روال ما بدون اشکال و درست بوده است می‌توانید به صورت زیر عمل نمایید.

در Run عبارت cmd را تایپ کنید تا خط فرمان Dos آشکار شود (شکل 3-13). در خط فرمان، عبارت sqlite3 را تایپ و اجرا نمایید. باید خروجی همانند شکل 3-9 باشد که نشان دهنده اجرای پوسته یا ابزار خط فرمان sqlite3 می‌باشد. البته چنانچه مسیر فایل sqlite3.exe را با استفاده از روشی که عنوان شد به سیستم معرفی کرده باشید، می‌توانید مستقیماً sqlite3 را در Run تایپ و برنامه را اجرا نمایید و لزومی به استفاده از cmd نیست.



شکل 3-13 استفاده از Run

تبریک می‌گوییم، روال نصب را به درستی انجام داده‌اید و این موتور پایگاه‌داده اکنون در اختیار شما است. چنانچه خروجی به غیر از شکل 3-9 نمایش داده شد، روال نصب را دوباره تکرار نمایید. عملیات به قدری ساده و بدون ابهام است که بعید به نظر می‌رسد در فرآیند نصب دچار خطا و اشکال شده باشید. از جمله مواردی که ممکن است باعث بروز خطا شده باشد غلط‌های تایپی در حین مشخص نمودن مسیر فایل sqlite3.exe می‌باشد.

خلاصه

در این فصل با شیوه‌ی دسترسی به فایل‌های مورد نیاز برای کار با پایگاه‌داده‌ی SQLite آشنا شدید. همچنین روال نصب SQLite که در واقع چیزی بیشتر از کپی کردن فایل‌ها نیست را نیز آموختید. در فصل بعدی با شیوه‌ی کار با ابزار خط فرمان sqlite3 و sqlite3_analyzer آشنا خواهید شد.

فصل 4

استفاده از CLP و SQLite Analyzer

همان‌گونه که در فصل پیش گفتیم، SQLite دارای پکیج‌های مختلفی است که هر یک کاربردی خاص دارد. این پکیج‌ها عبارتند از:

- ابزار خط فرمان sqlite3 یا CLP¹
- ابزار خط فرمان sqlite3_analyzer
- کتابخانه‌ی اشتراکی SQLite (dll یا .so)
- افزونه‌ی Tcl

این فصل به بررسی دو ابزار خط فرمانی می‌پردازد که در پروژه‌ی SQLite قرار گرفته و برای سهولت در استفاده از SQLite ارائه شده است.

استفاده از sqlite3

برنامه‌ی خط فرمان sqlite3 که جزئی از پروژه‌ی SQLite می‌باشد و با نام CLP نیز شناخته می‌شود، ابزاری برای کار با پایگاه‌داده‌های SQLite و مدیریت آنهاست. این برنامه به‌صورت توکار دارای نسخه‌ای از کتابخانه‌ی SQLite است و لذا لازم نیست برای استفاده از آن، کتابخانه‌ی SQLite در سیستم نصب بوده و یا در کنار فایل اجرایی وجود داشته باشد. این خود نمونه‌ای از یک برنامه‌ی کاربردی است که از کدهای SQLite درون آن استفاده شده است.

ابزار خط فرمان sqlite3 روش بسیار عالی برای آشنایی با پایگاه‌داده SQLite می‌باشد و همانند خود SQLite برای سیستم‌عامل‌های متفاوت ارائه شده است. به همین دلیل یادگیری کار با آن می‌تواند شما را مطمئن سازد که همیشه یک مسیر مشخص برای کار با پایگاه‌داده وجود دارد. این ابزار در واقع حاصل ادغام دو برنامه می‌باشد. نخست، ابزار خط فرمانی که می‌تواند برای مدیریت پایگاه‌داده مورد استفاده قرار گیرد و دیگری پوسته (Shell) و واسطی است که با استفاده از آن می‌توانید فرآیند پردازش پرس‌وجوهای خود را دنبال کنید و از پایگاه‌داده، اطلاعات استخراج نمایید.

استفاده از CLP در مد Shell

برای اجرای sqlite3 در حالت پوسته، در Run عبارت cmd را تایپ و اجرا نمایید. پس از آشکار شدن خط فرمان Dos می‌توانید با تایپ و اجرای دستور sqlite3، برنامه را فراخوانی نمایید. البته چنانچه متغیر Path در سیستم‌عامل را تنظیم نموده باشید، می‌توانید برنامه‌ی sqlite3 را به‌صورت مستقیم از درون Run فراخوانی نمایید. همچنین می‌توانید

¹ Command-Line Program