



قسمت یازدهم

این مار خوش خط و خال

پایگاه‌های داده

«احمد شریف پور»

برنامه نویسی در دنیای امروز که لبریز از داده‌های متفاوتی است که روز به روز بر تنوع و حجم آن‌ها افزوده می‌شود، بدون آشنایی با پایگاه‌های داده و استفاده از قابلیت‌های آن‌ها غیرممکن خواهد بود. در این قسمت و قسمت بعدی مجموعه مقاله‌های برنامه نویسی پایتون، به معرفی نحوه کار با پایگاه‌های داده از طریق کدهای پایتون خواهیم پرداخت. در بخش کنونی بیشتر به معرفی دستور زبان و نحوه کار پایگاه‌های داده SQL خواهیم پرداخت و در بخش بعدی نحوه استفاده از این زبان را در کدهای پایتون فرا خواهیم گرفت. در این دو قسمت ما از پایگاه داده SQLite و کتابخانه توابع python-apsw در محیط لینوکس (اوبونتو) استفاده خواهیم کرد.

یک داستان

ناگهان روزی، فرشته‌ای دوست‌داشتنی با بال‌هایی طلایی از بلندای کوهی ظاهر شد. نام او «سی کوئل» بود. فرشته ادعا کرد، می‌تواند مشکل کلنچار رفتن با این قفسه‌ها و پوشه‌ها را حل کند، البته، به شرطی که مردم به او و کامپیوترها اعتماد کنند. او این جادوی جدید را «پایگاه داده» نامید و گفت که پایگاه داده می‌تواند جایگزین تمام آن سیستم‌های بایگانی شود.

اما استفاده از این جادوی جدید، پیش‌نیازهایی داشت. مردم باید زبان جدیدی را می‌آموختند. البته، این زبان بسیار شبیه زبان عادی مردم بود و بیشتر به نوع متفاوتی از جمله‌بندی نیاز داشت و افراد باید پیش از به کار بردن این جمله‌های جادویی به دقت فکر می‌کردند. آن‌ها باید ابتدا طرحی از ارتباط میان داده‌ها و اطلاعات خود ترسیم می‌کردند. برخی از مردم حرف‌های او را پذیرفتند و زندگی برای آن‌ها بسیار لذت‌بخش و ساده شد و دیگری که او را به سخره گرفتند، در میان کوه‌های کاغذ مدفون شدند. پایان!

روزی روزگاری بود که دنیا به وسیله کاغذ اداره می‌شد. کاغذ، کاغذ و باز هم کاغذ. انسان‌ها مجبور بودند، به فکر ایجاد مکان‌های مناسبی برای نگهداری کاغذها باشند. این مکان‌ها را بایگانی می‌نامیدند و بایگانی کسب و کارهای بزرگ برای ذخیره تمام کاغذها، معمولاً چندین اتاق را اشغال می‌کرد. در هر بایگانی قفسه‌ها و کمدهایی وجود داشت که آن‌ها نیز به نوبه خود پر از پوشه‌های مختلف بودند. هر یک از این پوشه‌ها کاغذهای مربوط به یک موضوع خاص را نگهداری می‌کردند. اما پس از مدتی، ظرفیت همه آن‌ها به اتمام می‌رسید یا در اثر مرور زمان و باز و بسته شدن زیاد فرسوده شده و از هم می‌پاشیدند. استفاده درست از این سیستم بایگانی نیازمند مدارک دانشگاهی بود و گاه یافتن کل پرونده‌ها و کاغذهای مربوط به یک موضوع به چندین روز زمان نیاز داشت. کسب و کارها و سازمان‌ها از این وضعیت در عذاب بودند و آن دوران، دورانی تاریک در تاریخ بشریت محسوب می‌شد!

جدول طرز تهیه (Instructions) به نسبت ساده است و حاوی متن‌های طولانی و طرز تهیه غذاهای مختلف است. اما جدول مواد لازم (Ingredients) کمی پیچیده‌تر است، زیرا که به همراه نام ماده لازم مقدار آن‌ها نیز باید ذخیره شود.

برنامه‌های مورد نیاز

ابتدا مطمئن شوید، SQLite و APSW را نصب کرده‌اید. به خاطر داشته باشید، مخازن Canonical Partners و Independent باید در تنظیمات اوبونتو فعال شده باشند. در این صورت برای نصب برنامه‌های مورد نیاز در اوبونتو کافی است دستور زیر را در خط فرمان وارد کنید:

```
sudo apt-get install sqlite, python-apsw
```

مزیت SQLite این است که این موتور پایگاه داده برای اجرا به سرورهای جداگانه نیاز ندارد و به همین دلیل، برای برنامه‌های کوچک مناسب‌تر است. نکته مثبت دیگر این است که تعداد «انواع داده» یا Data Type های آن محدود و ساده است. این انواع عبارتند از Text (متن)، Numeric (عدد)، Blob (داده‌های دودویی) و Integer Primary Key (کلیدهای اصلی که بعدها باره آن‌ها توضیح خواهیم داد). مواد لازم، نام غذا و طرز تهیه همه داده‌هایی از نوع Text خواهند بود. اما Blob ها نوعی از داده است که می‌تواند شامل هر داده دیجیتالی نظیر عکس و... باشد که ما در این مثال با این نوع کاری نخواهیم داشت. موتور SQLite به صورت خودکار یک عدد صحیح منحصر به فرد را به انواع داده Integer Primary Key نسبت خواهد داد. کتابخانه APSW (سرنام Another Python SQLite Wrapper) نیز روش‌هایی ساده برای ارتباط با SQLite را فراهم خواهد آورد.

کمی SQL

پیش از شروع کار باید اندکی درباره کار با زبان SQL بیشتر بدانیم. در این زبان برای دریافت یک رکورد (یک سطر جدول) از یک پایگاه داده از دستور SELECT استفاده می‌کنیم. قالب اجرای این دستور به شکل زیر است:

```
SELECT [what] FROM [table] WHERE [constraints]
```

در این قالب کلی، تمام کلماتی که با حروف بزرگ نوشته شده‌اند باید عیناً به همین فرم آورده شوند. عبارت‌های داخل کروشه به ترتیب تعیین‌کننده رکورد مورد نظر (what)، جدولی یا جدول‌هایی که رکورد باید از آن‌ها استخراج شود (table) و شرایط انتخاب (constraints) هستند. مثلاً اگر بخواهیم همه فیلدهای جدول دستورالعمل‌ها را بدون هیچ شرطی استخراج کنیم، باید از دستوری به فرم زیر استفاده کنیم:

```
SELECT * FROM Recipes
```

و اگر بخواهیم رکوردی را تنها براساس شماره منحصر به فرد آن (مثلاً رکوردی با شماره ۲) استخراج کنیم، از عبارت زیر استفاده می‌کنیم:

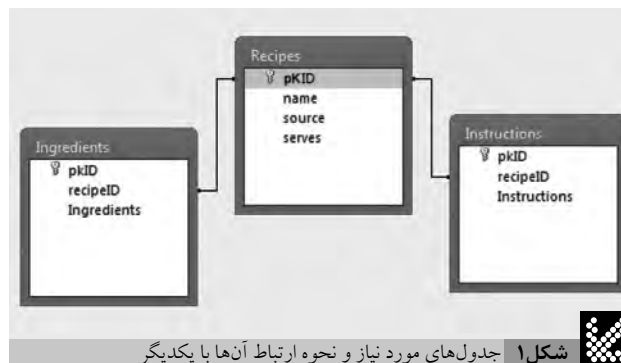
```
SELECT * FROM Recipes WHERE pkID=2
```

اگرچه این داستان چندان واقعی نبود، اما استفاده از پایگاه‌های داده SQL (بخوانید سی‌کوئل) زندگی ما را ساده‌تر خواهد کرد. پایگاه‌های داده همانند قفسه‌های بایگانی داستان بالا هستند. در این پایگاه‌های داده، جدول‌ها نقش پوشه‌ها را بازی می‌کنند و هر رکورد یا سطر این جدول‌ها در واقع حکم برگه‌های کاغذ در بایگانی‌های سنتی را دارند. هر تکه از اطلاعات یک فیلد نامیده می‌شود. برای کار با این سیستم ما از SQL (سرنام Structured Query language) یا «زبان پرس و جوی ساخت یافته» استفاده می‌کنیم. اگرچه این زبان از ابتدا برای ساده کردن کار با پایگاه‌های داده طراحی شده است، اما عبارت‌های مورد استفاده در آن می‌توانند بسیار پیچیده و طولانی شوند.

برای شروع بیایید به دستور پخت غذاهای مختلف فکر کنیم. این دستورهای پخت به فرم‌های گوناگون و در محل‌های مختلف نظیر کتاب‌های آشپزی، مجله‌ها، بسته‌بندی مواد غذایی و جاهای متنوع دیگر به چشم می‌خورند. اگرچه ظاهر آن‌ها ممکن است با هم متفاوت باشد، اما قالب همه آن‌ها یکسان است. در ابتدا نام غذا آورده شده است، پس از آن فهرست مواد لازم را خواهیم داشت و در انتها روش تهیه غذا توضیح داده شده است. پایگاه داده نمونه ما برای نگه‌داری این دستورالعمل‌های پخت غذا طراحی خواهد شد. ما از قالب کلی نوشتن این دستورالعمل‌ها برای طراحی جدول‌های پایگاه داده استفاده کرده و در این قسمت مقاله پایگاه داده را ایجاد می‌کنیم. در قسمت بعدی کدهای مربوط به خواندن و نوشتن اطلاعات روی پایگاه داده را خواهیم نوشت. هر چند که می‌توانیم جزئیات این پایگاه داده را تا حد دلخواه افزایش دهیم، اما برای سادگی کار ما تنها از سه جدول استفاده خواهیم کرد. جدول نخست، اطلاعات ابتدایی و خام دستورالعمل‌ها را (نظیر نام غذا، تعداد افرادی که با آن غذا سیر می‌شوند و نویسنده دستورالعمل) در خود نگه خواهد داشت؛ جدول دوم، اطلاعات مربوط به مواد لازم را در برخواهد داشت و در نهایت جدول سوم، حاوی دستورالعمل پخت غذاها خواهد بود.

ما می‌توانستیم کل پایگاه داده را با یک جدول نیز سروسامان دهیم، اما چنین جدولی بسیار بزرگ می‌شد و پراز اطلاعات تکراری می‌شد. جدول‌های مورد نظر ما شبیه شکل ۱ خواهند بود.

همه این جدول‌ها یک فیلد به نام pkID دارند که همان کلید اولیه منحصر به فرد است. وجود این فیلد بسیار مهم است، زیرا باعث می‌شود جدول‌های ما اطلاعات کاملاً تکراری نداشته باشند و همچنین ارتباط بین آن‌ها را فراهم می‌کند. این فیلد به صورت خودکار توسط موتور SQLite پر خواهد شد. در جدول Recipes ما از این فیلد برای استخراج اطلاعات مناسب از جدول‌های Instructions و Ingredients استفاده خواهیم کرد.



شکل ۱ جدول‌های مورد نیاز و نحوه ارتباط آن‌ها با یکدیگر

```

1 import apsw
2 # Opening/Creating database
3 connection = apsw.Connection("cookbook1.db3")
4 cursor=connection.cursor()
5
6 # Creating tables
7 sql='CREATE TABLE Recipes (pkID INTEGER PRIMARY\
8 KEY, name TEXT \
9 , serves TEXT, source TEXT)\'
10 cursor.execute(sql)
11 sql='CREATE TABLE Instructions (pkID INTEGER PRI-
12 MARY KEY, Instructions TEXT \
13 , recipeID NUMERIC)\'
14 cursor.execute(sql)
15 sql='CREATE TABLE Ingredients (pkID INTEGER PRIMARY
16 KEY, ingredients TEXT \
17 , recipeID NUMERIC)\'
18 cursor.execute(sql)
19
20 # Inserting sample data
21 sql='INSERT INTO Recipes (name,serves,source) VAL-
22 UES ("Spanish Rice" \
23 ,4,"Greg Walters")\'
24 cursor.execute(sql)
25
26 # Getting the last pkID
27 sql='SELECT last_insert_rowid()\'
28 cursor.execute(sql)
29 for x in cursor.execute(sql):
30     lastid=x[0]
31     print lastid
32
33 sql='INSERT INTO Instructions\
34 (recipeID,Instructions) VALUES \
35 (%s, "Brown hamburger. Stir in all other ingredi-
36 ents. Bring to boil.\
37 Stir. Lower to simmer. Cover and cook for 20 min\
38 utes or \
39 until all liquids absorbed.")\' %lastid
40 cursor.execute(sql)
41
42 sql='INSERT INTO Ingredients (recipeID , ingredi-
43 ents) VALUES \
44 (%s , "1 cup parboiled Rice (uncooked)")\' %lastid
45 cursor.execute(sql)

```

فهرست ۱ کدهای ساخت نخستین پایگاه داده



هنگامی که ما یک داده را در جدول Recipes وارد می‌کنیم باید عدد pkID منحصر به فرد آن را نیز داشته باشیم تا در سایر جدول‌ها نیز برای برقراری ارتباط از آن استفاده کنیم. این کار با دستور SQL زیر قابل انجام است:

```
SELECT last_insert_rowid()
```

اما این عملیات ساده، در برنامه ما در خط‌های ۲۷ تا ۳۰ پیاده شده است. دلیل هم این است که ما مقادیر برگشتی جدول را از apsw دریافت می‌کنیم و apsw آن‌ها را به صورت تویپل به ما باز می‌گرداند و ما باید عدد مورد نظر را از این تویپل استخراج کنیم. پس از به دست آوردن pkID آخرین رکورد وارد شده از آن برای درج مقادیر سایر جدول‌ها استفاده خواهیم کرد.

تا اینجا ما پایگاه داده و اطلاعات اولیه آن را وارد کرده‌ایم. در قسمت بعدی، به معرفی نحوه کار با این پایگاه داده خواهیم پرداخت.



یا به عنوان نمونه‌ای دیگر اگر بخواهیم از تمام دستورالعمل‌های موجود فقط نام غذا و مواد لازم را استخراج کنیم، باید از دستوری به فرم زیر استفاده کنیم:

```
SELECT name , instructions FROM Recipes
```

برای درج یا افزودن رکوردی تازه در یک جدول باید از دستور INSERT INTO استفاده کنیم. قالب کلی این دستور مانند زیر است:

```
INSERT INTO [table name] (field list) VALUES (values to insert)
```

به عنوان نمونه برای وارد کردن یک دستورالعمل جدید در جدول باید دستوری به فرم زیر را اجرا کرد:

```
INSERT INTO Recipes (name,servings,source) VALUES ("Tacos" ,
4, "Greg")
```

برای پاک کردن یک رکورد خاص هم می‌توان از دستور DELETE فرم زیر استفاده کرد:

```
DELETE FROM [table] WHERE [constrains]
```

کدنویسی

همان‌گونه که پیش‌تر هم گفته شد، ما در پایگاه داده خودمان سه جدول داریم که می‌توانند از طریق استفاده از فیلد recipeID در هر رکورد که خود به فیلد pkID در جدول اصلی دستورالعمل‌ها ارجاع دارد به هم مرتبط شوند و اکنون زمان آن رسیده تا کدهایی بنویسیم که پایگاه داده را ایجاد کرده، جدول‌ها را بسازند و آن‌ها را با داده‌های نمونه پر کنند. اگر چه می‌توانیم این کدها را در برنامه اصلی درج کنیم، اما ترجیح می‌دهیم آن‌ها را به صورت جداگانه بنویسیم تا بعدها نیز قابل استفاده باشند. فهرست ۱ را در ویرایشگر دلخواه‌تان وارد کرده و آن را با نام db1.py ذخیره و اجرا کنید.

در خط ۱ ما ما جاول apsw را import کرده‌ایم تا بتوانیم از توابع فراهم شده توسط آن استفاده کنیم. در خط ۳ یک اتصال یا کانکشن به پایگاه داده ساخته‌ایم. این کار باعث می‌شود، اگر پایگاه داده موجود باشد، برنامه آن را برای اجرای عملیات بعدی باز کند و در صورتی که چنین پایگاه داده‌ای موجود نباشد، apsw آن را تولید خواهد کرد.

در مثال ما، چون این پایگاه داده وجود ندارد، apsw آن را در همان پوشه برنامه ایجاد خواهد کرد. پس از ایجاد پایگاه داده، ما به یک مکان‌نما (cursor) نیاز داریم. مکان‌نما شرایط و ابزار لازم را برای تعامل با پایگاه داده فراهم خواهد آورد. این مکان‌نما در خط ۴ ایجاد شده است. پس از آن نوبت به ایجاد جدول‌ها می‌رسد.

برای ایجاد جدول دستورالعمل‌ها، در خط ۷ متغیری حاوی یک دستور SQL ایجاد کرده‌ایم و پس از آن در خط بعدی این دستور را اجرا کرده‌ایم. در خطوط بعدی سایر جدول‌ها را نیز به همین طریق ایجاد کرده‌ایم. پس از آن و در خطوط ۲۱ تا ۲۴ با دستور INSERT INTO تعدادی داده اولیه را در این جدول وارد کرده‌ایم.

فقط توجه داشته باشید که فیلد pkID به صورت خودکار و توسط موتور SQLite پر خواهد شد. در پایگاه‌های داده، چون ما از نام فیلدها استفاده می‌کنیم، ترتیب وارد کردن آن‌ها اهمیتی نخواهد داشت.