



قسمت دوازدهم

این مار خوش خط و خال

پایگاه‌های داده (۲)

«احمد شریف پور»

در دومین بخش از مقاله‌های مرتبط با پایتون و پایگاه‌های داده، به تکمیل برنامه قسمت پیشین (در شماره ۱۲۷ ماهنامه) خواهیم پرداخت. در قسمت قبل با مبانی کار با پایگاه‌های داده آشنا شدیم و کد کوچکی را نوشتیم که فایل اولیه پایگاه داده را برای ما به وجود می‌آورد و مقادیر اولیه را در آن می‌نوشت. در این قسمت می‌خواهیم این امکان را برای کاربر فراهم کنیم که محتویات پایگاه داده را ببیند، مواردی به آن بیافزاید یا برخی موارد را از آن حذف کند. برنامه ما در ترمینال اجرا خواهد شد و از یک منوی متنی ساده استفاده خواهد کرد تا کاربر از طریق آن گزینه‌های مورد نظرش را اجرا کند.

همچنین برای نگه‌داری روال‌های مربوط به پایگاه داده یک کلاس تعریف خواهیم کرد. چون این کد بسیار طولانی خواهد بود ما در ابتدا چهار چوب کلی آن را نوشته و سپس قسمت‌های مختلف را به تدریج به آن می‌افزاییم. برای شروع کار این چهار چوب را که در فهرست ۱ آورده شده، وارد کرده و آن را با نام Cookbook.py ذخیره کنید.

همان‌طور که در این کد مشاهده می‌کنید، کلیت یک کلاس و تابعی برای ترسیم یک منو روی صفحه نمایش تعریف شده است. ما به تدریج این چهار چوب را برای رسیدن به برنامه‌نمایی تکمیل خواهیم کرد. اگر در وضعیت فعلی برنامه را اجرا کنید، شکلی همانند شکل ۱ را مشاهده خواهید کرد. در این حالت، تنها کلید 0 برای خروج از برنامه کار خواهد کرد. اکنون به نوشتن روتین راه‌اندازی کلاس و سایر روتین‌های مربوط به اعمال نمایش کل اطلاعات، جست‌وجو، نمایش یک مورد، افزودن و حذف موارد پایگاه داده خواهیم پرداخت. از میان این روتین‌ها، روتین‌های نمایش کل اطلاعات، جست‌وجو و افزودن آیتم به هیچ آرگومان ورودی نیاز ندارند (البته همان‌طور که از بحث‌های کلاس‌ها و اشیاء به یاد دارید، به جزء آرگومان اجباری self). اما روتین‌های نمایش و حذف به پارامتری نیاز دارند که در این کدها آن را which نامیده‌ایم و این پارامتر تعیین می‌کند کدام آیتم باید نمایش داده‌شده یا حذف شود.

راه‌اندازی

در ابتدا باید روتین راه‌اندازی کلاس یا __init__ را کامل کنیم. کدهای مورد نیاز برای این روتین در فهرست ۲ آورده شده است. این کد را جایگزین خطوط ۵ و ۶ فهرست ۱ کنید.

در اینجا ابتدا دو متغیر جهانی تعریف کرده‌ایم. این متغیرها به رغم این‌که در تعریف یک تابع آورده شده‌اند، به واسطه کلمه global، در تمام برنامه قابل استفاده خواهند بود. این کار را به این دلیل انجام می‌دهیم که پایگاه داده و مکان‌نمای مربوط به آن در تمام روتین‌های برنامه قابل دسترس باشند. برای کار با پایگاه داده‌ای که در قسمت قبل ایجاد کردید، باید این برنامه و فایل پایگاه داده را در یک پوشه و در کنار هم قرار دهید و در خط ۵، نام فایل پایگاه داده خود را جایگزین cookbook1.db3 کنید.

نمایش همه موارد

برای نمایش همه موارد از کد فهرست ۳ استفاده خواهیم کرد. این کد را در کلاس Cookbook فهرست ۱ جایگزین خطوط ۷ و ۸ کنید. همین‌طور

```

1 import apsw
2 import string
3
4 class Cookbook:
5     def __init__(self):
6         pass
7     def All(self):
8         pass
9     def Search(self):
10        pass
11    def Show(self, which):
12        pass
13    def Add(self):
14        pass
15    def Delete(self, which):
16        pass
17
18 def Menu():
19     ckb= Cookbook()
20     loop=True
21     while loop:
22         print "*****"
23         print "* Recipe Database *"
24         print "*****"
25         print " 1-Show all"
26         print " 2-Search for a recipe"
27         print " 3-Show a recipe"
28         print " 4-Add a recipe"
29         print " 5-Delete a recipe"
30         print " 0-Exit"
31         print "*****"
32         response= raw_input(" Enter your selection: ")
33         if response == '1':
34             pass
35         elif response == '2':
36             pass
37         elif response == '3':
38             pass
39         elif response == '4':
40             pass
41         elif response == '5':
42             pass
43         elif response == '0':
44             print "\n\tGoodbye"
45             loop=False
46         else:
47             print "\nUnrecognized command. Try again.\n\n"
48
49 Menu()

```

فهرست ۱ چهار چوب کلی برنامه Cookbook

```

1 def __init__(self):
2     global connection
3     global cursor
4     self.totalCount=0
5     connection = apsw.Connection("cookbook1.db3")
6     cursor=connection.cursor()

```

فهرست ۲ کدهای تکمیل تابع راه‌اندازی کلاس Cookbook

```

1 def All(self):
2     print "\n\nNumber \tName \t\tServes \t\tSource"
3     print "-"*55
4     sql = "SELECT * FROM Recipes"
5     cntr=0
6     for x in cursor.execute(sql):
7         cntr += 1
8         print "%s \t%s \t%s \t\t\t%s" %(x[0],x[1],x[2],x[3])
9     print "-"*55
10    self.totalCount = cntr
11    print "Total number of items: %s" %cntr

```

فهرست ۳ کدهای تابع نمایش تمام موارد موجود در پایگاه داده

```

Terminal
File Edit View Terminal Go Help

*****
* Recipe Database *
*****
1-Show all
2-Search for a recipe
3-Show a recipe
4-Add a recipe
5-Delete a recipe
0-Exit
*****
Enter your selection:

```

شکل ۱ منوی دسترسی به پایگاه داده

```

1 def Search(self):
2     print
3     print "="*55
4     print "Searching in recipes by name"
5     term = raw_input("Enter name of the recipe: ")
6     sql = "SELECT pkID , name FROM Recipes WHERE name like\
7     '%s%s%' " %term
8     for x in cursor.execute(sql):
9         self.Show(x[0])

```

فهرست ۴ کدهای تابع جست‌وجو در پایگاه داده براساس نام دستورالعمل

```

1 def Show(self,which):
2     sql = "SELECT * FROM Recipes WHERE pkID = %s" %str(which)
3     print "\n"
4     print "="*55
5     for x in cursor.execute(sql):
6         print "Title: %s \t Serves: %s \t Source: %s\"
7     %(x[1],x[2],x[3])
8
9     sql = "SELECT * FROM Ingredients WHERE pkID = %s" %str(which)
10    print "\nIngredients:"
11    for x in cursor.execute(sql):
12        print x[1]
13
14    sql = "SELECT * FROM Instructions WHERE pkID = %s" %str(which)
15    print "\nInstructions:"
16    for x in cursor.execute(sql):
17        print x[1]
18    print "="*55
19    print "\n"

```

فهرست ۵ کدهای تابع مورد نیاز برای نمایش یک آیتم از پایگاه داده

را به نمایش خواهیم گذاشت. کدهای مربوط به این روتین در فهرست ۵ آورده شده است. این کدها را جایگزین خطوط ۱۱ و ۱۲ فهرست ۱ کنید. در این‌که، در خط ۲ دستور sql مورد نیاز را برای انتخاب آیتمی که شماره ردیف (pkID) آن با آرگومان which به تابع اعلام شده است، تعریف کرده‌ایم. پس از آن با تعریف دو دستور sql دیگر مقادیر متناظر را از جدول‌های Ingredients و

و از او خواهیم خواست که آیتم مورد نظر را انتخاب کند. برای این کار کلمه pass در خط ۳۸ فهرست ۱ را با کدهای زیر جایگزین کنید:

```

ckb.All()
temp = raw_input("What to show:")
ckb.Show(temp)

```

پس از آن با فرستادن آیتم انتخاب شده به عنوان آرگومان which به روتین Show() آن آیتم

باید از خط ۳۴ فهرست ۱ کلمه pass را حذف و عبارت زیر را جایگزین آن کنید:

```

ckb.All()
temp = raw_input("Press any key\
to continue...")

```

در این تابع ما از cursor ساخته شده در روتین __init__ استفاده کرده‌ایم و تمام محتویات جدول Recipes را فراخوانی کرده‌ایم، سپس در یک حلقه تمام مقادیر را با قالب‌بندی‌های استفاده شده در دستور print (که قبلاً با آن آشنا شده‌اید) به صورت منظم چاپ کرده‌ایم. به یاد دارید که به ما باز می‌گرداند، به همین دلیل، ما از x[0] و x[1] و... استفاده کرده‌ایم. در انتهای این روتین و در خط ۱۱ تعداد کل آیتم‌های موجود در جدول Recipes را نیز اعلام کرده‌ایم.

جست‌وجو

برای بخش جست‌وجو، به دلیل کمبود فضا، تنها بخش جست‌وجوی نام دستورالعمل را پیاده‌سازی کرده‌ایم و جست‌وجو بر اساس محتویات و تعداد نفراتی را که با یک دستورالعمل سیر می‌شوند، به عهده شما گذاشته‌ایم. کدهای این روتین در فهرست ۴ آورده شده است. این کدها را جایگزین خطوط ۹ و ۱۰ فهرست کنید.

در این‌که و در خط ۵ عبارت مورد نظر از کاربر پرسیده شده و در خط ۶ با تعریف یک دستور sql عملیات جست‌وجو را روی پایگاه داده انجام داده‌ایم. شاید عبارت %s%s% در این خط‌کده عجیب به نظر برسد. در زبان sql برای تعریف مشابهت از کلمه کلیدی like و فرم %term% استفاده می‌شود. در این حالت کلماتی که مشابه term باشند، برگردانده می‌شوند. اما چون در زبان پایتون از علامت % برای فرمت‌کردن رشته‌ها استفاده می‌شود، یک رشته نمی‌تواند به طور مستقیم شامل علامت % باشد. برای درج این علامت در یک رشته باید دوبار آن را پشت سر هم (%) به کار ببریم. به این ترتیب، در خط ۷ جفت علامت‌های سمت راست و چپ به تک علامت‌های % و قسمت وسط (s%) نیز توسط عبارت term جایگزین می‌شود. در انتهای این روتین و در خط ۹ از تابع Show() که آن را در قسمت بعد تعریف می‌کنیم، برای نمایش آیتم‌های مورد نظر استفاده کرده‌ایم. برای فعال شدن این بخش باید کلمه pass در خط ۳۶ فهرست ۱ با عبارت ckb.Search() جایگزین شود.

نمایش یک مورد

برای عملیاتی کردن این بخش، ابتدا با فراخوانی تابع All() تمام موارد را به کاربر نشان خواهیم داد

وارد کند، ادامه می‌یابد. پس از آن در خطوط ۱۴ تا ۲۱ کل اطلاعات یک بار دیگر برای کاربر نمایش داده می‌شود تا از صحت آن اطمینان حاصل شود. پس از گرفتن تأیید کاربر، در خطوط ۲۷ تا ۲۹ با تعریف دستورالعمل مورد نیاز، موارد ابتدایی را در جدول Recipes ذخیره می‌کنیم. در خطوط ۳۱ و ۳۲ همان‌گونه که در قسمت قبل دیده‌اید، شماره آخرین رکورد افزوده شده به جدول را استخراج کرده‌ایم. از این رکورد برای مرتبط کردن داده‌های جدول‌های Ingredients و Instructions به مقادیر وارد شده در جدول Recipes استفاده کرده‌ایم. ذخیره داده در دو جدول Ingredients و Instructions به ترتیب در خطوط ۳۶ تا ۳۹ و ۴۰ تا ۴۳ انجام شده است. در نهایت، اتمام عملیات ذخیره‌سازی با یک پیغام به کاربر اطلاع داده شده است. فراموش نکنید که برای فعال شدن این بخش باید کلمه pass در خط ۴۰ فهرست را با ckb.Add() جایگزین کنید.

حذف

در نهایت به روتین حذف موارد می‌رسیم. همان‌گونه که حدس می‌زنید، در این روتین نیز ابتدا با فراخوانی ckb.All() همه موارد به کاربر نشان داده شده و در نهایت از او خواسته می‌شود تا آیتم مورد نظر را برای حذف کردن انتخاب کند. این روتین در فهرست ۶ آورده شده است. آن را جایگزین خطوط ۱۵ و ۱۶ فهرست ۱ کنید و برای کار کردن این کد باید کلمه pass در خط ۴۲ فهرست ۱ با عبارت زیر جایگزین شود:

```
ckb.All()
temp = raw_input("What to delete:")
ckb.Delete(temp)
```

در این کد هم با استفاده از آرگومان which شماره انحصاری آیتم را نشان می‌دهد، دستورات SQL مورد نیاز را برای حذف آیتم‌ها از جدول‌های Ingredients, Recipes و Instructions تعریف و اجرا کرده‌ایم. دقت کنید که برای ساده‌تر شدن کار، ما از بسیاری از روتین‌های کنترل ورودی کاربر، مرتب‌سازی خروجی و بسیاری قابلیت‌های دیگر که می‌توانست چنین برنامه‌ای را کامل‌تر کند، صرف‌نظر کرده‌ایم. اگرچه وجود این قابلیت‌ها در غالب نرم‌افزارهای تجاری و حرفه‌ای الزامی است، اما پیاده‌سازی آن‌ها بسیار ساده و بدون مشکل خواهد بود. برای نمونه، در هنگام نمایش یا حذف یک مورد بهتر بود که ما شماره آیتم وارد شده توسط کاربر را بررسی کنیم تا در محدوده صحیحی قرار داشته باشد یا مثلاً کاربر به جای شماره از حروف استفاده نکرده باشد. که چنین مواردی با روال‌های معمول if قابل پیاده‌سازی است.

```
1 def Add(self):
2     ings = ""
3     lastid = 0
4     recipename=raw_input('Enter Recipe Title -> ')
5     recipesource=raw_input("Enter Recipe Source -> ")
6     reciperves=raw_input("Enter number of servings -> ")
7     while True:
8         ing = raw_input('Enter Ingredient ("0" to exit) -> ')
9         if ing != '0':
10            ings = ings + ing + ", "
11        else:
12            break
13        instructions = raw_input('Enter Instructions -> ')
14        print '*55
15        print "Here>s what we have so far"
16        print "Title: %s" % recipename
17        print "Source: %s" % recipesource
18        print "Serves: %s" % reciperves
19        print "Ingredients: %s" % ings
20        print "Instructions: %s" % instructions
21        print '*55
22        resp = raw_input("OK to save? (Y/n) -> ")
23        if resp.upper() != 'N':
24            #connection=apsw.Connection("cookbook1.db3")
25            #cursor=connection.cursor()
26            # Write the Recipe Record
27            sql = 'INSERT INTO Recipes (name,serves,source) VALUES\
28 (" %s", "%s", "%s")' % (recipename,reciperves,recipesource)
29            cursor.execute(sql)
30            # Get the new Recipe pkID
31            sql = "SELECT last_insert_rowid()"
32            cursor.execute(sql)
33            for x in cursor.execute(sql):
34                lastid = x[0]
35                print "last id = %s" % lastid
36            # Write the Instruction Record
37            sql = 'INSERT INTO Ingredients (recipeID,ingredients)\
38 VALUES (%s, "%s")' % (lastid,x)
39            cursor.execute(sql)
40            # Write the Ingredients records
41            sql = 'INSERT INTO Instructions (recipeID,instructions)\
42 VALUES( %s, "%s")' % (lastid,instructions)
43            cursor.execute(sql)
44            # Prompt the user that we are done
45            print 'Done\n'
46        else:
47            print 'Save aborted\n'
```

فهرست ۶ کدهای تابع افزودن آیتم جدید به پایگاه داده

```
1 def Delete(self,which):
2     sql = "DELETE FROM Recipes WHERE pkID = %s" %str(which)
3     cursor.execute(sql)
4     sql = "DELETE FROM Ingredients WHERE pkID = %s" %str(which)
5     cursor.execute(sql)
6     sql = "DELETE FROM Instructions WHERE pkID = %s" %str(which)
7     cursor.execute(sql)
8     print "Recipe compeletly deleted . . .\n"
```

فهرست ۷ کدهای حذف یک آیتم از پایگاه داده

متغیرهای ابتدایی مورد نیاز نظیر نام دستورالعمل، تعداد نفرات و... را تعریف و مقاردهی می‌کند. تنها موردی که شاید به توضیح نیاز داشته باشد، دریافت مواد لازم است که در یک حلقه while انجام شده است و هر ورودی کاربر به صورت رشته‌ای با جداکننده کاما (,) به انتهای متغیر ings افزوده شده است. این روند تا زمانی که کاربر 0 را

Instructions استخراج کرده و با دستور print چاپ کرده‌ایم.

افزودن

روتین مربوط به افزودن موارد، طولانی‌ترین روتین این مجموعه است که آن را در فهرست ۶ مشاهده می‌کنید، آن را جایگزین خطوط ۱۳ و ۱۴ فهرست ۱ کنید. شروع این روتین و خط‌های ۲ تا ۶