



## SQLite چیست؟

ذخیره و بازیابی اطلاعات یکی از نیازهای اساسی در هنگام برنامه‌نویسی سیستم‌های مختلف است. برای انجام این کار، بسته به نیازهای موجود ابزارها و روش‌های مختلفی نیز وجود دارند. از روش‌های ساده مانند نگهداری اطلاعات در یک فایل متنی تا استفاده از نرم‌افزارهای عظیم مانند Oracle.

یکی از ابزارهای ذخیره و بازیابی اطلاعات، SQLite نام دارد. این نرم‌افزار مشهورترین سیستم ذخیره فایلی اطلاعات به شمار می‌رود. شهرت SQLite به دلیل پشتیبانی از انواع مختلف سیستم‌عامل‌ها از جمله ویندوز، لینوکس، آندروئید و مک او اس و همچنین رایگان و قدرتمند بودن آن است.

SQLite با زبان C برنامه‌نویسی شده است و به طور پیوسته در حال بهبود و توسعه است. به همین دلیل سرعت و کارایی بسیار بالایی دارد. در نگارش‌های جدید که در آینده منتشر خواهند شد، بهینه‌سازی‌های گسترده‌ای روی این سیستم به انجام رسیده است که سرعت عملکرد آن را بیش از پیش افزایش داده است.

SQLite نرم‌افزاری با مجوز استفاده Public Domain است. به این معنی که حق مالکیتی ندارد و هر فرد یا سازمانی می‌تواند بدون هیچ محدودیتی از آن به هر شکلی استفاده کند.

## کاربرد SQLite

SQLite یک سیستم ذخیره اطلاعات فایلی است بنابراین استفاده از آن در محیط‌هایی که مشتریان متعددی بخواهند از اطلاعات استفاده کنند ممکن است با مشکلاتی مواجه شود. در عوض در محیط‌هایی مانند نرم‌افزارهای موبایل و وبسایت‌هایی که بازدیدهای سنگین و زیادی نداشته باشند کارایی و عملکرد خوبی به دست خواهد آمد.

برخی از سیستم‌های مدیریت محتوا مانند دروپال، گزینه‌ی کاربرد SQLite برای طراحی سایت را در اختیار برنامه‌نویسان و صاحبان سایت قرار می‌دهند.

بسیاری از نرم‌افزارهای شناخته‌شده در دنیا مانند موزیلا فایرفاکس، گوگل کروم، ویندوز ۱۰، اسکایپ و دراپ‌باکس برای نگهداری بخشی از اطلاعات خود از SQLite استفاده می‌کنند.

یکی دیگر از کاربردهای پایگاه داده‌ی فایل‌ی مثل SQLite، انتقال اطلاعات دارای ساختار و حجیم است. برای مثال ممکن است یک اپلیکیشن موبایل، اطلاعات جمع‌آوری شده از کاربران را در قالب یک پایگاه داده‌ی SQLite به سمت سرور مرکزی ارسال نماید.

## مدیریت آسان

SQLite به دلیل استفاده از سیستم فایل‌ی، نیازی به تنظیمات پیچیده و اساسی مانند بسیاری از DBMS‌های دیگر ندارد. کافی است فایل پایگاه داده را در کنار نرم‌افزار قرار دهید و از آن استفاده کنید.

## امکانات پیشرفته SQLite

SQLite با وجود کم حجم و سریع بودن دارای امکانات پیشرفته‌ای مانند Transactionها، جدول‌های مستقر در حافظه، جستجوی متن Full Text و امکانات گسترده‌ی دیگری است که گاهی به دلیل همین امکانات به صورت موازی در کنار سایر سیستم‌های بانک اطلاعاتی مورد استفاده قرار می‌گیرد.

## ماژول sqlite3

In [1]:

```
1 import sqlite3
```

In [2]:

```
1 new_connection = sqlite3.connect("test_data.db")
```

In [3]:

```
1 print(new_connection)
```

<sqlite3.Connection object at 0x00000231BD907640>

In [4]:

```
1 type(new_connection)
```

Out[4]:

sqlite3.Connection

In [5]:

```
1 new_cursor = new_connection.cursor()
```

In [6]:

```
1 type(new_cursor)
```

Out[6]:

sqlite3.Cursor

## دستورات مقدماتی برای کار با پایگاه داده

In [7]:

```
1 new_sql = "SELECT datetime('now', 'localtime');"
```

In [8]:

```
1 new_cursor.execute(new_sql)
```

Out[8]:

<sqlite3.Cursor at 0x231bf1fc540>

In [9]:

```
1 new_cursor.fetchone()
```

Out[9]:

('2023-07-19 10:12:03',)

In [11]:

```
1 new_dtime = new_cursor.execute("SELECT datetime('now', 'localtime');").fetchone()[0]
```

In [12]:

```
1 print(new_dtime)
```

2023-07-19 10:15:49

In [13]:

```
1 new_connection.close()
```

In [14]:

```
1 ###
2
3 with sqlite3.connect("test_data.db") as new_connection:
4     new_cursor = new_connection.cursor()
5     new_sql = "SELECT datetime('now', 'localtime');"
6     new_dtime = new_cursor.execute(new_sql).fetchone()[0]
7
8 print(new_dtime)
9
```

2023-07-19 10:19:49

## ایجاد جدول و ورود داده

In [38]:

```
1 import sqlite3
2
3 new_connection = sqlite3.connect("test_data.db")
4 new_cursor = new_connection.cursor()
```

In [39]:

```
1 new_cursor.execute("DROP TABLE IF EXISTS clients")
2 new_cursor.execute("CREATE TABLE clients(name TEXT, number INT);")
```

Out[39]:

<sqlite3.Cursor at 0x231c0535a40>

In [22]:

```
1 # new_cursor.execute("DROP TABLE clients")
```

Out[22]:

<sqlite3.Cursor at 0x231c0534440>

In [40]:

```
1 new_cursor.execute("INSERT INTO clients VALUES('Ali', 1998);")
```

Out[40]:

<sqlite3.Cursor at 0x231c0535a40>

In [41]:

```
1 new_connection.commit()
2
3 new_connection.close()
```

In [34]:

```
1 new_connection = sqlite3.connect("test_data.db")
2 new_cursor = new_connection.cursor()
```

In [29]:

```
1 new_cursor.execute("SELECT * FROM clients;")
```

Out[29]:

<sqlite3.Cursor at 0x231c0535fc0>

In [30]:

```
1 new_cursor.fetchone()
```

Out[30]:

('Ali', 1998)

In [31]:

```
1 new_cursor.execute("DROP TABLE clients")
```

Out[31]:

<sqlite3.Cursor at 0x231c0535fc0>

In [35]:

```
1 new_connection.commit()
2 new_connection.close()
```

## بروزرسانی و حذف رکورد

In [42]:

```
1 new_connection = sqlite3.connect("test_data.db")
2 new_cursor = new_connection.cursor()
```

In [43]:

```
1 new_cursor.execute("UPDATE clients set number = 2020 WHERE name = 'Ali' ")
```

Out[43]:

<sqlite3.Cursor at 0x231c05352c0>

In [44]:

```
1 new_connection.commit()
2 new_connection.close()
```

In [49]:

```
1 new_connection = sqlite3.connect("test_data.db")
2 new_cursor = new_connection.cursor()
3
4 new_cursor.execute("SELECT * FROM clients;")
5
6 new_cursor.fetchone()
```

In [50]:

```
1 new_connection.commit()
2 new_connection.close()
```

In [47]:

```
1 ##### DELETE
2
3 new_connection = sqlite3.connect("test_data.db")
4 new_cursor = new_connection.cursor()
5
6 new_cursor.execute("DELETE FROM clients WHERE name = 'Ali'")
```

Out[47]:

<sqlite3.Cursor at 0x231c05bc9c0>

In [48]:

```
1 new_connection.commit()
2 new_connection.close()
```

## خواندن و نوشتن چندین رکورد

In [57]:

```
1 new_values = (  
2     "Ronaldo", 7),  
3     "Messi", 30),  
4     "Salah", 11)  
5 )  
6  
7 with sqlite3.connect("test_data.db") as new_connection:  
8     new_cursor = new_connection.cursor()  
9     new_cursor.execute("DROP TABLE IF EXISTS clients")  
10    new_cursor.execute("CREATE TABLE clients(name TEXT, number INT);")  
11  
12    new_cursor.executemany("INSERT INTO clients VALUES(?, ?)", new_values)  
13  
14    new_cursor.execute("SELECT name, number FROM clients WHERE number < 28")  
15  
16    for record in new_cursor.fetchall():  
17        print(record)
```

```
('Ronaldo', 7)  
( 'Salah', 11)
```