## python数据库支持

在 python 世界里, 无需怀疑,与数据库协同工作已经是几乎所有应用程序的核心部分。从python中访问数据库需要接口程序。接口程序是一个python模块,它提供数据库客户端库(通常是 C 语言写成的)的接口供你访问。需要提到一点,所有Python接口程序都一定程度上遵守Python DB-API规范。

在python.org找到数据库主题那一节,你会发现所有支持DB-API 2.0的各种数据库模块,文档,SIG 等等。DB-API是一个规范。它定义了一系列必须的对象和数据库存取方式,以便为各种各样的底层数据库系统和多种多样的数据库接口程序提供一致的访问接口。象绝大多数社区成果一样,这个API的产生来自于强烈的需求。

在python的 DB-API规范中要求数据库模块必须提供模块属性，链接对象，游标对象，类型对象和构造器等必须的属性。

对每一种支持的数据库,都有一个或多个Python接口程序允许你连接到目标数据库系统。某些数据库, 比如Sybase, SAP, Oracle 和 SQLServer, 都有两个或更多个接口程序可供选择。你要做的就是挑选一个最能满足你需求的接口程序。你挑选接口程序的标准可以是:性能如何?文档或WEB站点的质量如何?是否有一个活跃的用户或开发社区?接口程序的质量和稳定性如何?等等。

### Mysql的python数据库接口

这里我们以 MySQL 数据库为例,使用python的MySQL接口程序:MySQLdb，来进行python操作数据库的学习。

## 建立连接

要与数据库进行通信, 必须先和数据库建立连接。连接对象处理命令如何送往服务器, 以及如   
何从服务器接收数据等基础功能。连接成功(或一个连接池)后你就能够向数据库服务器发送请求,   
得到响应。

### 常见异常



### 建立连接的常用方法

connect():创建数据库链接

close():关闭数据库链接

commit():提交当前事务

rollback():取消当前事务

cursor():创建游标并返回一个游标对象

connect()调用：注意不是所有的接口程序都是严格按照规范实现的。MySQLdb 就使用了 db 参 数而不是规范推荐的 database 参数来表示要访问的数据库。

一旦执行了close()方法，再试图使用连接对象的方法将会导致异常。对不支持事务的数据库或者虽然支持事务，但设置了自动提交(auto-commit)的数据库系统来说，commit()方法什么也不做。如果你确实需要，可以实现一个自定义方法来关闭自动提交行为。由于 DB-API 要求必须实现此方法，对那些没有事务概念的数据库来说，这个方法只需要有一条pass语句就可以了。类似 commit()，rollback()方法仅对支持事务的数据库有意义。执行完 rollback()，数据库将恢复到提交事务前的状态。根据 PEP249，在提交 commit()之前关闭数据库连接将会自动调用rollback()方法。对不支持游标的数据库来说, cursor()方法仍然会返回一个尽量模仿游标对象的对象。这些是最低要求。特定数据库接口程序的开发者可以任意为他们的接口程序添加额外的属性，只要他们愿意。DB-API 规范建议但不强制接口程序的开发者为所有数据库接口模块编写异常类。如果没有提供异常类，则假定该连接对象会引发一致的模块级异常。一旦你完成了数据库连接，并且关闭了游标对象，你应该执行 commit() 提交你的操作，然后关闭这个连接。

## 创建数据库了连接实例

连接数据库前，请先确认以下事项：

您已经创建了数据库 TESTDB

在TESTDB数据库中您已经创建了表 EMP

EMPLOYEE表字段为 NAME, AGE, SEX 和 NUM

连接数据库TESTDB使用的用户名为 “root” ，密码为 “123”

在你的机子上已经安装了 Python MySQLdb 模块

## 游标对象操作

当你建立连接之后，就可以与数据库进行交互。我们提交一个游标允许用户执行数据库命令和得到查询结果。一个 Python DB-API 游标对象总是扮演游标的角色, 无论数据库是否真正支持游标。从这一点讲, 数据库接口程序必须实现游标对象。只有这样，才能保证无论使用何种后端数据库你的代码都不需要做任何改变。创建游标对象之后，你就可以执行查询或其它命令(或者多个查询和多个命令), 也可以从结果集中取出一条或多条记录。

### 游标属性和方法

在这众多游标对象中最重要的属性是 execute() 和 fetch()方法。所有对数据库服务器的请求都由它们来完成。对 fetchmany()方法来说, 设置一个合理的 arraysize属性会很有用。当然，在不需要时关掉游标对象也是个好主意。