第二十一章：数据库编程

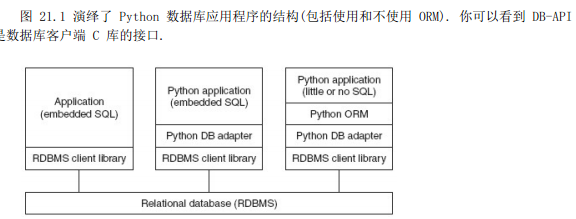
在任何的应用程序中，都需要持久存储。一般说来，有三种基本的存储机制：文件、关系型数据库或其它的一些变种，例如现有系统的 API，ORM、文件管理器、电子表格、配置文件等等。本章的主题是如何在中大型项目中使用关系型数据库.

如果你使用的关系数据库管理系统是基于服务器的，这些数据库就都在同一台服务器上 (一些简单的关系型数据库没有服务器, 如 sqlite). 本章的例子中, MySQL 是一种基于服务器的关系数据库管理系统(只要服务器在运行, 它就一直在等待运行指令)，SQLite 和 Gadfly 则是另一种轻量型的基于文件的关系数据库(它们没有服务器)。

## 21．1数据库 和 Python

下面我们要详细了解 Python 数据库 API. Python 能够直接通过数据库接口, 也可以通过 ORM (不需要自己书写 SQL) 来访问关系数据库.

所以有两种方法，一是直接通过数据库接口,另一种是通过 ORM。



Python提供了多种连接数据库的方式，包括odbc、dao、ado以及python专门模块等方式。Python提供了连接oracle、mysql等数据库的专用模块。Ado和python的专用模块是比较常用的访问方式。

Python的专用模块其实就是第三方的类库，你可以导入或安装才能够使用。

## 21．2通过数据库接口

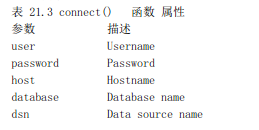
程序员接口(DB-API)DB-API 是一个规范. 它定义了一系列必须的对象和数据库存取方式, 以便为各种各样的底层数据库系统和多种多样的数据库接口程序提供一致的访问接口. 象绝大多数社区成果一样, 这个 API的产生来自于强烈的需求.

一个处理 Python 数据库事务的特殊事物小组（SIG，special interest group） 因此诞生,

最后. DB-API 1.0 问世. DB-API 为不同的数据库提供了一致的访问接口, 在不同的数据库之间移植代码成为一件轻松的事情(一般来说, 只修要修改几行代码). 接下来你会看到这样的例子.

常使用到的方法有：具体可参考书籍p850

connect 方法

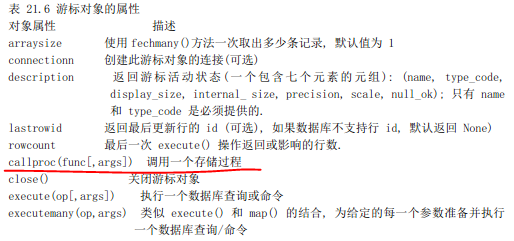


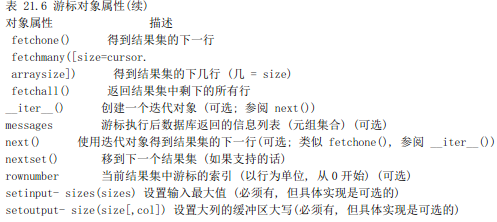
close() 关闭数据库连接

commit() 提交当前事务

cursor() 使用这个连接创建并返回一个游标或类游标的对象

游标对象的属性：





游标对象最重要的属性是 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. 如果你的数据库支持存储过程, 你就可以使用 callproc() 方法.

使用数据库接口程序举例（mysql）。具体参考代码

## 21．3对象-关系管理器(ORMs)

过前一节我们知道, 如今有很多种数据库系统, 他们中的绝大多数都有 Python 接口, 以方便你驾驭他们的能量. 这些系统唯一的缺点是需要你懂得 SQL. 如果你是一个喜欢折腾 Python 对象却讨厌 SQL 查询的家伙, 又想使用关系型数据库做为你的数据存储的后端, 你就完全具备成为一个 ORM 用户的天资.

1）考虑对象，而不是 SQL

这些系统的创建者将绝大多数纯 SQL 层功能抽象为 Python 对象, 这样你就无需编写 SQL 也能够完成同样的任务. 如果你在某些情况下实在需要 SQL, 有些系统也允许你拥有这种灵活性. 但绝大多数情况下, 你应该尽量避免进行直接的 SQL 查询.

数据库的表被转换为 Python 类, 它具有列属性和操作数据库的方法. 让你的应用程序支持

ORM 非常类似使用那些标准的数据库接口程序. 由于大部分工作由 ORM 代为处理, 相比直接使用接口程序来说, 一些事情可能实际需要更多的代码. 另人欣慰的是, 一点点额外的付出会回报你更高的生产率.

2）Python 和 ORM

如今最知名的 Python ORM 模块是 SQLAlchemy 和 SQLObject. 由于二者有着不同的设计哲学,我们会分别给出 SQLAlchemy 和 SQLObject 的例子. 只要你能搞清楚这两种 ORM 的使用, 转到其它的 ORM 将是相当简单的事.

其它的 Python ORM 包括 PyDO/PyDO2, PDO, Dejavu, Durus, QLime 和 ForgetSQL. 一些大型的 Web 开发工具/框架也可以有自己的 ORM 组件, 如 WebWare MiddleKit 和 Django 的数据库 API.需要指出的是, 知名的 ORM 并不意味着就是最适合你的应用程序的 ORM. 那些其它的 ORM 虽然没有纳入我们的讨论范围, 但一样有可能是适合你的应用程序的选择.具体的使用参考代码。

要自己额外的安装SQLObject以及SQLAlchemy

实例收获：

1、怪不得，你自己安装的mysql的port是3308

2、你的mysql-python是安装在python27下的，所以你的编译器应该是python27的，不然你根本无法使用mysqldb这个模块。

3、python数据库编程怎么使用SQLAlchemy。嘿嘿，那个需要安装的，具体参你下的那个pdf。还有使用SQLAlchemy时，前提是你的msqldb也可以使用。

注意：SQLAlchemy这些都是第三方的库，所以呢，你要注意了，这些东西要额外的导入。