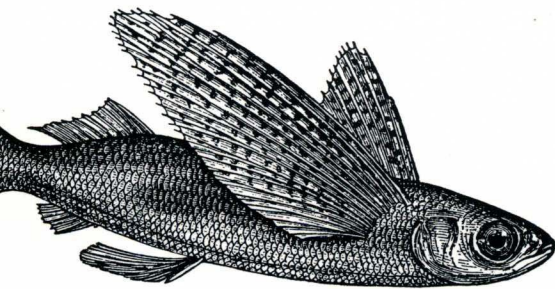


O'REILLY®

TURING

图灵程序设计丛书

第2版



SQLAlchemy

Python数据库实战

Essential SQLAlchemy

通过实例讲解如何在Python程序中使用关系型数据库



[美] 贾森·迈尔斯 里克·科普兰 著

武传海 译



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

SQLAlchemy: Python数据库实战（第2版）

Essential SQLAlchemy, 2nd Edition

[美] 贾森·迈尔斯 里克·科普兰 著
武传海 译

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

O'Reilly Media, Inc.授权人民邮电出版社出版

31522810/610 人民邮电出版社

人民邮电出版社

北京

图书在版编目 (CIP) 数据

SQLAlchemy : Python数据库实战 : 第2版 / (美)
贾森·迈尔斯 (Jason Myers), (美) 里克·科普兰
(Rick Copeland) 著 ; 武传海译. — 北京 : 人民邮电
出版社, 2019.8

(图灵程序设计丛书)

ISBN 978-7-115-51630-5

I. ①S… II. ①贾… ②里… ③武… III. ①软件工
具—程序设计 IV. ①TP311.561

中国版本图书馆CIP数据核字(2019)第140192号

内 容 提 要

本书主要探讨 SQLAlchemy, 这个 Python 库在关系型数据库和传统编程之间架起了一座桥梁, 有助于 Python 程序员将应用程序连接到关系型数据库。本书首先通过对比的方式介绍了 SQLAlchemy 的两种主要使用模式——SQLAlchemy Core 和 SQLAlchemy ORM, 然后探讨了数据库迁移工具 Alembic 的用法, 最后快速讲解了 SQLAlchemy 的高级应用。

本书适合 Python 开发人员阅读。

-
- ◆ 著 [美] 贾森·迈尔斯 里克·科普兰
译 武传海
责任编辑 岳新欣
责任印制 周昇亮
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 10.75
字数: 242千字 2019年8月第1版
印数: 1-3 000册 2019年8月北京第1次印刷
-
- 著作权合同登记号 图字: 01-2018-8086号

定价: 59.00元

读者服务热线: (010)51095183转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

站在巨人的肩上

Standing on Shoulders of Giants



iTuring.cn

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

版权声明

O'Reilly Media 通过图书、杂志、计算机图书、网络图书、音频、视频、电子书、移动设备等多种方式传播知识。自 1978 年开始，O'Reilly 一直致力于提供高质量的图书和数字内容。O'Reilly 一直认为，知识是改变世界的力量。我们关注真正重要的技术趋势，并通过我们的“O'Reilly 系列”来传播它们。我们关注真正重要的技术趋势，并通过我们的“O'Reilly 系列”来传播它们。我们关注真正重要的技术趋势，并通过我们的“O'Reilly 系列”来传播它们。

© 2016 Jason Myers and Rick Copeland.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2019. Authorized translation of the English edition, 2015 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版，2015。

简体中文版由人民邮电出版社出版，2019。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

O'Reilly Media, Inc.介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 *Make* 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过图书出版、在线服务还是面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar 博客有口皆碑。”

——Wired

“O'Reilly 凭借一系列非凡想法（真希望当初我也想到了）建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——CRN

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野，并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去，Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

前言

我们周围到处都是数据，数据的存储、更新以及报表制作是每个应用程序应该具备的关键功能。无论你开发的是 Web 应用程序、桌面应用程序还是其他应用程序，你都需要确保可以快速、安全地获取数据。如今，关系型数据库仍然是存放数据最常用的方式之一。

SQL 是一种用来查询和操作数据库数据的强大语言，但有时我们很难将其与所开发的应用程序集成在一起。使用 Python 做开发时，你可能使用字符串创建过在 ODBC 接口上运行的查询，也可能用过 DB API。虽然这些方法可以有效地处理数据，但安全性令人担忧，而且也很难对数据库做出调整。

本书主要探讨 SQLAlchemy，这是一个非常强大又相当灵活的 Python 库，它在关系型数据库和传统编程之间架起了一座桥梁。虽然 SQLAlchemy 允许我们使用原始 SQL 执行查询，但是它更鼓励我们使用其提供的高级方法来查询和更新数据库。SQLAlchemy 提供的方法用起来很友好，而且具有鲜明的 Python 风格。SQLAlchemy 还提供了许多很棒的工具，通过这些工具，你可以轻松地将应用程序中的类和对象同数据库表映射起来，然后“忘掉它”，或者不断回到模型调优性能上。

SQLAlchemy 功能强大并且十分灵活，但也有点令人望而生畏。SQLAlchemy 教程只讲了这个优秀库中的一小部分内容。另外，尽管在线文档非常多，但大都不适合用来学习 SQLAlchemy，而是更适合作参考资料。本书既是一本 SQLAlchemy 学习指南，也是一本方便的参考手册，每当在工作中碰到问题，你都可以拿来翻一翻，相信你能很快地从中找到答案。

本书重点讲解 SQLAlchemy 1.0 版本，但所讲的大部分内容同样适用于之前的多个版本。对于 0.8 之后的版本，本书示例只需做小幅调整即可运行，并且大部分示例都要求 SQLAlchemy 版本号不低于 0.5。

本书主要分为三个部分：SQLAlchemy Core、SQLAlchemy ORM 和 Alembic。其中，前两

个部分相互对照。我们特意在每个部分使用相同的示例，以方便你比较 SQLAlchemy 的两种主要用法。阅读本书时，你既可以通读 SQLAlchemy Core 和 SQLAlchemy ORM 这两部分内容，也可以只阅读适合你当前需要的那部分。

本书读者

如果你想学习如何在 Python 程序中使用关系型数据库，或者听说过 SQLAlchemy 并且想了解更多信息，那本书正是为你而写的。为了最大限度地利用本书，你的 Python 技能应该达到中等水平，并且对 SQL 数据库有一定的了解。虽然我们努力将本书内容讲得浅显易懂，但如果你刚刚开始学习 Python，建议你先阅读一下 Bill Lubanovic 写的《Python 语言及其应用》¹一书，或者观看 Jessica McKellar 制作的“Introduction to Python”教学视频，这些学习资料都很棒。另外，如果你不熟悉 SQL 和数据库，可以先看看 Alan Beaulieu 编写的《SQL 学习指南》一书。事先掌握这些知识对于学习本书内容很有必要。

如何使用书中示例

本书中的大部分示例都需要在 REPL（“读取 - 求值 - 输出”循环）环境中运行。你可以在命令提示符下输入 `python` 来使用内置的 Python REPL 环境。这些示例也可以在 IPython notebook 中正常运行。本书有几部分会教你创建和使用文件，而非使用 REPL，比如第 4 章。大部分示例代码和各章的 Python 文件都以 IPython notebook 形式提供给大家。更多关于 IPython 的内容，请去 IPython 官网学习。

阅读前提

本书假定你已经掌握了有关 Python 语法和语义的基础知识，特别是 Python 2.7 及之后的版本。你尤其应该熟悉 Python 中的迭代和对象，本书中会经常用到它们。本书第二部分讲解面向对象编程和 SQLAlchemy ORM。你还应该了解基本的 SQL 语法和关系理论，因为本书假设你已经熟悉如何定义模式和表，以及 SELECT、INSERT、UPDATE 和 DELETE 语句的创建。

排版约定

本书使用如下排版约定。

- 黑体字
表示新术语或重点强调的内容。

注 1：此书已由人民邮电出版社出版，详见 <http://www.it-ebooks.info/book/1560>。——编者注

- 等宽字体 (`constant width`)

表示程序片段，以及正文中出现的变量、函数名、数据库、数据类型、环境变量、语句和关键字等。

- 加粗等宽字体 (`constant width bold`)

表示应该由用户输入的命令或其他文本。

- 斜体等宽字体 (`constant width italic`)

表示应该替换成用户输入的值，或根据上下文替换的值。



该图标表示提示或建议。



该图标表示一般性说明。



该图标表示警告或警示。

使用代码示例

补充材料（代码示例、练习等）可以从 <https://github.com/oreillymedia/essential-sqlalchemy-2e> 下载。

本书是要帮你完成工作的。一般来说，如果书中提供了示例代码，你可以把它用在你的程序或文档中。除非你使用了很大一部分代码，否则无须联系我们获得许可。比如，用书中的几个代码片段写一个程序无须获得许可，销售或分发 O'Reilly 图书的示例光盘则需要获得许可；引用书中的示例代码回答问题无须获得许可，将书中大量的代码放到你的产品文档中则需要获得许可。

我们很希望但并不强制要求你在引用本书内容时加上引用说明。引用说明一般包括书名、作者、出版社和 ISBN。比如：“*Essential SQLAlchemy, Second Edition*, by Jason Myers and Rick Copeland (O'Reilly). Copyright 2016 Jason Myers and Rick Copeland, 978-1-4919-1646-9.”

如果你觉得自己对示例代码的用法超出了上述许可的范围，欢迎你通过 permissions@oreilly.com 与我们联系。

O'Reilly Safari



Safari (之前称作 Safari Books Online) 一个针对企业、政府、教育者和个人的会员制培训和参考平台。

会员可以访问来自 250 多家出版商的上千种图书、培训视频、学习路径、交互式教程和精选播放列表, 这些出版商包括 O'Reilly Media、Harvard Business Review、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Adobe、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 等。

要了解更多信息, 可以访问 <http://www.oreilly.com/safari>。

联系我们

请把对本书的评价和问题发给出版社。

美国:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国:

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室 (100035)
奥莱利技术咨询(北京)有限公司

O'Reilly 的每一本书都有专属网页, 你可以在那儿找到书的相关信息, 包括勘误表²、示例代码以及其他信息。本书的网站地址是: <http://oreil.ly/1lwEdiw>。

对于本书的评论和技术性问题, 请发送电子邮件到: bookquestions@oreilly.com。

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息, 请访问以下网站:

<http://www.oreilly.com>。

我们在 Facebook 的地址如下: <http://facebook.com/oreilly>。

请关注我们的 Twitter 动态: <http://twitter.com/oreillymedia>。

我们的 YouTube 视频地址如下: <http://www.youtube.com/oreillymedia>。

注 2: 本书中文版勘误请到 <http://www.ituring.com.cn/book/1986> 查看和提交。——编者注

致谢

首先，非常感谢 Patrick Altman、Eric Floehr 和 Alex Grönholm 在本书出版前提供的重要反馈。如果没有他们的帮助，本书无疑会有许多技术问题，而且会更难读。

其次，感谢 Mike Bayer，正是在他的推荐下我才动笔撰写本书。感谢 Meghan Blanchette 和 Dawn Schanafelt 督促我完成本书的写作，让我成为一个更好的作者，并且包容了我的种种缺点。还要感谢 Brian Dailey 认真审读了本书部分初稿，并提供了很好的反馈，我们相谈甚欢。

再次，感谢纳什维尔开发社区对我的支持，特别是 Cal Evans、Jacques Woodcock、Luke Stokes、William Golden 这几位朋友。

感谢我的雇主——思科系统公司给了我充足的时间并支持我完成本书。还要感谢 Mechanical Keyboards 公司的 Justin，他为我提供了码字所需的一切。

最后，也是最重要的一点：我要感谢我的妻子，感谢她忍受我大声朗读，放任我去写作，并一直给予我支持和鼓励。我爱你，丹妮丝。

电子书

扫描如下二维码，即可购买本书电子版。



SQLAlchemy入门

SQLAlchemy 库用于与各种数据库交互，你可使用一种类似于 Python 类和语句的方式创建数据模型和查询。SQLAlchemy 库是 Mike Bayer 在 2005 年创建的，现在大大小小很多公司都在使用它。事实上，许多公司都把 SQLAlchemy 看作在 Python 中使用关系型数据库的标准方式。

SQLAlchemy 可用于连接大多数常见的数据库，比如 Postgres、MySQL、SQLite、Oracle 等。SQLAlchemy 还提供了一种为其他关系型数据库添加支持的方式。Amazon Redshift（使用 PostgreSQL 自定义方言）就是 SQLAlchemy 社区添加的数据库支持的一个很好的例子。

在本章中，我们要搞清楚为什么要使用 SQLAlchemy，还要学习它的两种主要模式，以及如何连接到数据库。

为什么使用SQLAlchemy

使用 SQLAlchemy 的首要原因是，它将你的代码从底层数据库及其相关的 SQL 特性中抽象出来。SQLAlchemy 使用功能强大的常见语句和类型，确保其 SQL 语句为每种数据库和供应商正确、高效地构建出来，而无须你考虑这些。同时，这使得将逻辑从 Oracle 迁移到 PostgreSQL 或从应用程序数据库迁移到数据仓库变得很容易。它还有助于确保数据在提交到数据库之前得到“净化”并正确地进行了转义。这可以避免一些常见的问题，比如 SQL 注入攻击。

SQLAlchemy 提供了两种主要的使用模式——SQL 表达式语言（通常称为 Core）和 ORM，这为我们使用 SQLAlchemy 提供了很大的灵活性。这两种模式可以单独使用，也可以一起使用，具体用法取决于你的喜好以及应用程序的需求。

SQLAlchemy Core和SQL表达式语言

SQL 表达式语言允许我们以 Python 方式使用常见的 SQL 语句和表达式，它是对标准 SQL

语言的简单抽象。SQL 表达式语言关注的是实际的数据库模式，它在经过标准化之后为大量后端数据库提供了一种一致性的语言。SQL 表达式语言也是 SQLAlchemy ORM 的基础。

ORM

SQLAlchemy ORM 类似于你在其他语言中遇到的对象关系映射（ORM）。它关注应用程序的领域模型，并利用工作单元模式来维护对象状态。它还在 SQL 表达式语言之上做了进一步的抽象，使用户能够以惯用的方式工作。你可以组合或搭配使用 ORM 与 SQL 表达式语言，从而创建出功能更为强大的应用程序。ORM 中用到了一个声明式系统，该系统类似于许多其他 ORM 中使用的 Active-Record 系统，比如 Ruby on Rails 中使用的那个。

虽然 ORM 极其有用，但你必须记住，类的关联方式和底层数据库关系的工作方式是有区别的。第 6 章会更加全面地探讨这些方式如何影响你的实现。

选择SQLAlchemy Core还是SQLAlchemy ORM

在使用 SQLAlchemy 构建应用程序之前，你要决定主要使用 ORM 还是 Core。选择使用 Core 还是 ORM 作为应用程序的主要数据访问层，通常取决于多个因素和个人偏好。

Core 和 ORM 使用的语法略有不同，但它们之间最大的区别是把数据看作模式还是业务对象。SQLAlchemy Core 的视图以模式为中心，与传统 SQL 一样，它关注的是表、键和索引结构。SQLAlchemy Core 在数据仓库、报表、分析和其他场景中（在这些场景中，严格控制对未建模的数据进行查询或操作将非常有用）可以大放异彩。强大的数据库连接池和结果集优化非常适合处理大量数据，甚至适合处理多个数据库的数据。

但是，如果你主要关注的是领域驱动设计，那么选用 ORM 会更好，它会帮你把元数据和业务对象中的底层模式和结构封装起来。这种封装使得和数据库的交互变得更加容易，就像在使用普通的 Python 代码一样。大多数常见的应用程序都可以通过这种方式进行建模。ORM 还可以有效地把领域驱动设计注入到遗留的应用程序或者到处是原始 SQL 语句的应用程序中。微服务也能从对底层数据库的抽象中获益，它使得开发人员可以只关注正在实现的流程。

但是，由于 ORM 构建在 SQLAlchemy Core 之上，所以你也可以借助 ORM 使用 Oracle 数据仓库和 Amazon Redshift 这样的服务，就像和 MySQL 交互一样。因此，ORM 很适合用来合并业务对象和仓库数据。

下面列出了几种应用场景，了解它们有助于你在 Core 和 ORM 中做出最佳选择。

- 虽然你使用的框架中已经内置了 ORM，但你希望添加更强大的报表功能，请选用 Core。
- 如果你想要在一个以模式为中心的视图中查看数据（用法类似于 SQL），请使用 Core。

- 如果你的数据不需要业务对象，请使用 Core。
- 如果你要把数据看作业务对象，请使用 ORM。
- 如果你想快速创建原型，请使用 ORM。
- 如果你需要同时使用业务对象和其他与问题域无关的数据，请组合使用 Core 和 ORM。

你已经了解了 SQLAlchemy 的结构以及 Core 和 ORM 之间的区别，接下来开始安装并使用 SQLAlchemy 来连接数据库。

安装 SQLAlchemy 并连接到数据库

SQLAlchemy 可以与 Python 2.6、Python 3.3 和 PyPy 2.1 或更高版本一起使用。建议使用 pip 来安装 SQLAlchemy（使用命令 `pip install sqlalchemy`）。值得注意的是，还可以使用 `easy_install` 和 `distutils` 来安装它。不过，相比之下，使用 pip 安装更简单。安装过程中，SQLAlchemy 会尝试构建一些 C 扩展，这些扩展可以加快结果集的处理速度，同时提高内存使用效率。如果你的系统中缺少编译器，所以你想禁用这些扩展，那么可以使用 `--global-option=--without-cextensions`。请注意，如果没有 C 扩展，SQLAlchemy 的性能会受到影响。你应该在带有 C 扩展的系统上测试代码，然后再进行优化。

安装数据库驱动程序

默认情况下，SQLAlchemy 直接支持 SQLite3，不需要额外安装驱动程序。不过，在连接其他数据库时需要额外安装一个数据库驱动程序，并且该驱动程序要遵守标准的 Python DBAPI 规范（PEP-249）。这些 DBAPI 为各个数据库服务器所用的方言提供了基础，并且为不同数据库服务器和版本中的独特特性提供了支持。虽然许多数据库都有多种 DBAPI 可用，但我们将只介绍那些最常用的。

- PostgreSQL

Psycopg2 为 PostgreSQL 的各个版本和特性提供了广泛的支持，你可以使用 `pip install psycopg2` 命令安装它。

- MySQL

PyMySQL 是我用来连接 MySQL 数据库服务器的首选 Python 库。可以使用 `pip install pymysql` 命令安装它。SQLAlchemy 支持 MySQL 4.1 及更高版本，这是由 MySQL 的密码工作方式造成的。另外，如果某些语句只在 MySQL 的某个版本中可用，那么，对于那些不支持这些语句的 MySQL 版本，SQLAlchemy 不会为它们提供使用这些语句的方法。如果 SQLAlchemy 中的某个组件或函数在你的环境下不起作用，那你首先要做的是查看一下 MySQL 文档。

- 其他

SQLAlchemy 还可以与 Drizzle、Firebird、Oracle、Sybase 和 Microsoft SQL Server 一起使用。SQLAlchemy 社区还为许多其他的数据库提供了外部方言，如 IBM DB2、Informix、Amazon Redshift、EXASolution、SAP SQL Anywhere、Monet 等。此外，SQLAlchemy 还对创建方言提供了很好的支持，第 7 章将讲解相关内容。

既然我们已经安装好 SQLAlchemy 和 DBAPI 了，接下来创建一个引擎去连接数据库。

连接到数据库

要连接到数据库，需要先创建一个 SQLAlchemy 引擎。SQLAlchemy 引擎为数据库创建一个公共接口来执行 SQL 语句。这是通过包装数据库连接池和方言来实现的，这样它们就可以一起工作，提供对后端数据库的统一访问。如此一来，我们的 Python 代码就不必考虑数据库之间或 DBAPI 之间的差异了。

SQLAlchemy 提供了一个函数来创建引擎。在这个函数中，你可以指定连接字符串，以及其他一些可选的关键字参数。连接字符串是一种特殊格式的字符串，包含如下信息：

- 数据库类型（Postgres、MySQL 等）
- 各数据库类型默认之外的方言（Psycopg2、PyMySQL 等）
- 可选的认证细节（用户名和密码）
- 数据库位置（数据库服务器的文件或主机名）
- 数据库服务器端口（可选）
- 数据库名（可选）

SQLite 数据库连接字符串指定了一个特定的数据库文件或存储位置。示例 P-1 定义了一个存储在当前目录下、名为 cookies.db 的 SQLite 数据库文件，当前目录由第二行代码中的相对路径指定，第三行代码是一个内存数据库，第四行（Unix）和第五行（Windows）中指定了数据库文件的完整路径。在 Windows 平台下，连接字符串如最后一行代码所示；除非你使用原始字符串（`r''`），否则就要使用 `\\` 进行字符串转义。

示例 P-1 为 SQLite 数据库创建引擎

```
from sqlalchemy import create_engine
engine = create_engine('sqlite:///cookies.db')
engine2 = create_engine('sqlite:///memory:')
engine3 = create_engine('sqlite:///home/cookiemonster/cookies.db')
engine4 = create_engine('sqlite:///c:\\Users\\cookiemonster\\cookies.db')
```



`create_engine` 函数会返回一个引擎的实例。但是，在调用需要使用连接的操作（比如查询）之前，它实际上并不会打开连接。

接下来，让我们为名为 mydb 的本地 PostgreSQL 数据库创建一个引擎。为此，我们先从 sqlalchemy 包导入 create_engine 函数。然后，使用 create_engine 函数创建引擎实例。在示例 P-2 中，你会看到我把 postgresql+psycopg2 用作连接字符串的引擎和方言组件，当然只使用 postgres 也可以。相比于隐式方式，我更喜欢显式方式，这也正是“Python 之禅”（Zen of Python）所提倡的。

示例 P-2 为本地 PostgreSQL 数据库创建引擎

```
from sqlalchemy import create_engine
engine = create_engine('postgresql+psycopg2://username:password@localhost:' \
                       '5432/mydb')
```

下面看看位于远程服务器上的 MySQL 数据库。在示例 P-3 中你会看到，在连接字符串之后，我们使用了一个关键字参数 pool_recycle，用来指定回收连接的频率。

示例 P-3 为远程 MySQL 数据库创建引擎

```
from sqlalchemy import create_engine
engine = create_engine('mysql+pymysql://cookiemonster:chocolatechip'
                       '@mysql01.monster.internal/cookies', pool_recycle=3600)
```



默认情况下，超过 8 小时，MySQL 才会关闭空闲连接。为了解决这个问题，可在创建引擎时把 pool_recycle 设置为 3600，如示例 P-3 所示。

create_engine 函数还有一些可选参数，如下所示。

- echo

开启这个参数会记录引擎处理的操作，例如 SQL 语句及其参数。默认值为 false。

- encoding

这个参数用于指定 SQLAlchemy 所使用的字符串编码方式，默认值为 utf-8。大多数 DBAPI 默认支持这种编码。但这个参数并不用来指定后端数据库所使用的编码类型。

- isolation_level

这个参数用来为 SQLAlchemy 指定隔离级别。例如，使用 Psycopg2 的 PostgreSQL 拥有 READ COMMITTED、READ UNCOMMITTED、REPEATABLE READ、SERIALIZABLE 和 AUTOCOMMIT 选项，默认值为 READCOMMITTED。PyMySQL 也有一样的选项，就 InnoDB 数据库来说，默认值为 REPEATABLE READ。



可以使用 isolation_level 这个关键字参数为任何给定的 DBAPI 设置隔离级别。此外，还可以通过方言连接字符串中的键-值对来设置隔离级别，比如支持这个方法的 Psycopg2。