



图灵程序设计丛书

PostgreSQL即学即用（第2版）

PostgreSQL: Up and Running Second Edition

[美] Regina Obe Leo Hsu 著
丁奇鹏 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权人民邮电出版社出版

人民邮电出版社
北 京

图书在版编目 (C I P) 数据

PostgreSQL即学即用 : 第2版 / (美) 奥贝
(Obe, R.), (美) 徐 (Hsu, L.) 著 ; 丁奇鹏译. — 北京:
人民邮电出版社, 2016. 1
(图灵程序设计丛书)
ISBN 978-7-115-41128-0

I. ①P… II. ①奥… ②徐… ③丁… III. ①关系数
据库系统 IV. ①TP311. 132. 3

中国版本图书馆CIP数据核字(2015)第285900号

内 容 提 要

本书将帮助你理解和使用 PostgreSQL 这一开源数据库系统。你不仅会学到版本 9.2、9.3 和 9.4 中的企业级特性, 还会发现 PostgreSQL 不只是个数据库系统, 也是一个出色的应用平台。本书通过示例展示了如何实现在其他数据库中难以或无法完成的任务。这一版内容覆盖了 LATERAL 查询、增强的 JSON 支持、物化视图和其他关键话题。

本书适合数据库管理员、后端开发人员以及其他对 PostgreSQL 感兴趣的读者。

-
- ◆ 著 [美] Regina Obe Leo Hsu
译 丁奇鹏
责任编辑 岳新欣
执行编辑 赵瑞琳
责任印制 杨林杰
 - ◆ 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 印刷
 - ◆ 开本: 800×1000 1/16
印张: 14
字数: 330千字 2016年1月第1版
印数: 1—3 500册 2016年1月北京第1次印刷
著作权合同登记号 图字: 01-2015-6365号
-

定价: 59.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

版权声明

© 2015 by Regina Obe and Leo Hsu.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2016. 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，2016。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。无论是通过书籍出版、在线服务或者面授课程，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（真希望当初我也想到了）非凡想法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*

目录

前言	xi
第 1 章 基础知识	1
1.1 如何获得 PostgreSQL	1
1.2 管理工具	1
1.2.1 psql	2
1.2.2 pgAdmin	2
1.2.3 phpPgAdmin	3
1.2.4 Adminer	3
1.3 PostgreSQL 数据库对象	4
1.4 最新版本的 PostgreSQL 中引入的新特性	9
1.4.1 为什么要升级	10
1.4.2 PostgreSQL 9.4 版中引入的新特性	10
1.4.3 PostgreSQL 9.3 版新特性列表	12
1.4.4 PostgreSQL 9.2 版新特性列表	13
1.4.5 PostgreSQL 9.1 版新特性列表	14
1.5 数据库驱动程序	14
1.6 如何获得帮助	15
1.7 PostgreSQL 的主要衍生版本	15
第 2 章 数据库管理	17
2.1 配置文件	17
2.1.1 postgresql.conf	18
2.1.2 pg_hba.conf	21
2.1.3 配置文件的重新加载	23

2.2	连接管理	24
2.3	角色	25
2.3.1	创建可登录角色	26
2.3.2	创建组角色	26
2.4	创建 database	28
2.4.1	模板数据库	28
2.4.2	schema 的使用	29
2.5	权限管理	31
2.5.1	权限的类型	31
2.5.2	入门介绍	31
2.5.3	GRANT	32
2.5.4	默认权限	33
2.5.5	PostgreSQL 权限体系中一些与众不同的特点	34
2.6	扩展包机制	34
2.6.1	扩展包的安装	36
2.6.2	通用扩展包	38
2.7	备份与恢复	40
2.7.1	使用 pg_dump 进行有选择性的备份	41
2.7.2	使用 pg_dumpall 进行全库备份	42
2.7.3	数据恢复	43
2.8	基于表空间机制进行存储管理	45
2.8.1	表空间的创建	45
2.8.2	在表空间之间迁移对象	45
2.9	禁止的行为	46
2.9.1	切记不要删除 PostgreSQL 系统文件	46
2.9.2	不要把操作系统管理员权限授予 PostgreSQL 的系统账号 (postgres)	47
2.9.3	不要把 shared_buffers 缓存区设置得过大	47
2.9.4	不要将 PostgreSQL 服务器的侦听端口设为一个已被其他程序占用的端口	47
第 3 章	psql 工具	49
3.1	环境变量	49
3.2	psql 的两种操作模式：交互模式与非交互模式	50
3.3	定制 psql 操作环境	51
3.3.1	自定义 psql 界面提示符	52
3.3.2	语句执行时间统计	53
3.3.3	事务自动提交	53
3.3.4	命令别名	54
3.3.5	取出前面执行过的命令行	54
3.4	psql 使用技巧	55
3.4.1	执行 shell 命令	55
3.4.2	用 watch 命令重复执行语句	55

3.4.3	显示对象信息	55
3.5	使用 psql 实现数据的导入和导出	56
3.5.1	使用 psql 进行数据导入	56
3.5.2	使用 psql 进行数据导出	58
3.5.3	从外部程序复制数据以及将数据复制到外部程序	58
3.6	使用 psql 制作简单的报表	58
第 4 章	pgAdmin 的使用	61
4.1	pgAdmin 入门	61
4.1.1	功能概览	61
4.1.2	如何连接到 PostgreSQL 服务器	63
4.1.3	pgAdmin 界面导航	63
4.2	pgAdmin 功能特性介绍	64
4.2.1	在 pgAdmin 中调用 psql	65
4.2.2	在 pgAdmin 中编辑 postgresql.conf 和 pg_hba.conf 文件	65
4.2.3	创建数据库资产并设置权限	66
4.2.4	数据导入和导出	68
4.2.5	备份与恢复	69
4.3	pgScript 脚本机制	72
4.4	以图形化方式解释执行计划	75
4.5	使用 pgAgent 执行定时任务	75
4.5.1	安装 pgAgent	76
4.5.2	规划定时任务	76
4.5.3	一些有用的 pgAgent 相关查询语句	78
第 5 章	数据类型	80
5.1	数值类型	80
5.1.1	serial 类型	81
5.1.2	生成数组序列的函数	81
5.2	字符和字符串	82
5.2.1	字符串函数	83
5.2.2	将字符串拆分为数组、表或者子字符串	83
5.2.3	正则表达式和模式匹配	84
5.3	时间类型	85
5.3.1	时区详解	87
5.3.2	日期时间类型的运算符和函数	89
5.4	数组类型	91
5.4.1	数组构造函数	92
5.4.2	引用数组中的元素	93
5.4.3	数组的拆分与连接	93
5.4.4	将数组元素展开为记录行	93

5.5	区间类型	94
5.5.1	离散区间和连续区间	95
5.5.2	原生支持的区间类型	95
5.5.3	定义区间的方法	96
5.5.4	定义含区间类型字段的表	97
5.5.5	适用于区间类型的运算符	97
5.6	JSON 数据类型	98
5.6.1	插入 JSON 数据	98
5.6.2	查询 JSON 数据	99
5.6.3	输出 JSON 数据	101
5.6.4	JSON 类型的二进制版本: jsonb	101
5.7	XML 数据类型	103
5.7.1	插入 XML 数据	103
5.7.2	查询 XML 数据	104
5.8	自定义数据类型和复合数据类型	105
5.8.1	所有表都有一个对应的自定义数据类型	105
5.8.2	构建自定义数据类型	107
5.8.3	为自定义数据类型构建运算符和函数	107
第 6 章 表、约束和索引		109
6.1	表	109
6.1.1	基本的建表操作	109
6.1.2	继承表	110
6.1.3	无日志表	110
6.1.4	TYPE OF	111
6.2	约束机制	112
6.2.1	外键约束	112
6.2.2	唯一性约束	113
6.2.3	check 约束	113
6.2.4	排他性约束	113
6.3	索引	114
6.3.1	PostgreSQL 原生支持的索引类型	115
6.3.2	运算符类	116
6.3.3	函数索引	118
6.3.4	基于部分记录的索引	118
6.3.5	多列索引	119
第 7 章 PostgreSQL 的特色 SQL 语法		121
7.1	视图	121
7.1.1	单表视图	122
7.1.2	使用触发器来更新视图	123

7.1.3	物化视图	125
7.2	灵活易用的 PostgreSQL 专有 SQL 语法	127
7.2.1	DISTINCT ON	127
7.2.2	LIMIT 和 OFFSET 关键字	128
7.2.3	简化的类型转换语法	128
7.2.4	一次性插入多条记录	128
7.2.5	使用 ILIKE 实现不区分大小写的查询	129
7.2.6	可以返回结果集的函数	129
7.2.7	限制对继承表的 DELETE、UPDATE、INSERT 操作的影响范围	130
7.2.8	DELETE USING 语法	130
7.2.9	将修改影响到的记录行返回给用户	130
7.2.10	在查询中使用复合数据类型	131
7.2.11	DO	132
7.3	适用于聚合操作的 FILTER 子句	133
7.4	窗口函数	135
7.4.1	PARTITION BY 子句	136
7.4.2	ORDER BY 子句	136
7.5	CTE 表达式	138
7.5.1	基本 CTE 用法介绍	139
7.5.2	可写 CTE 用法介绍	140
7.5.3	递归 CTE 用法介绍	140
7.6	LATERAL 横向关联语法	141
第 8 章	函数编写	144
8.1	PostgreSQL 函数功能剖析	145
8.1.1	函数功能基础知识介绍	145
8.1.2	触发器和触发器函数	146
8.1.3	聚合操作	147
8.1.4	受信与非受信语言	149
8.2	使用 SQL 语言来编写函数	149
8.2.1	编写基本的 SQL 函数	150
8.2.2	使用 SQL 语言编写聚合函数	151
8.3	使用 PL/pgSQL 语言编写函数	153
8.3.1	编写基础的 PL/pgSQL 函数	153
8.3.2	使用 PL/pgSQL 编写触发器函数	154
8.4	使用 PL/Python 语言编写函数	155
8.5	使用 PL/V8、PL/CoffeeScript 以及 PL/LiveScript 语言来编写函数	157
8.5.1	编写基本的函数	159
8.5.2	使用 PL/V8 来编写聚合函数	160

第 9 章 查询性能调优	162
9.1 通过 EXPLAIN 命令查看语句执行计划	162
9.1.1 EXPLAIN 选项	162
9.1.2 运行示例以及输出内容解释	163
9.1.3 图形化展示执行计划	166
9.2 搜集语句的执行统计信息	167
9.3 人工干预规划器生成执行计划的过程	168
9.3.1 策略设置	168
9.3.2 你的索引被用到了吗	169
9.3.3 表的统计信息	170
9.3.4 磁盘页的随机访问成本以及磁盘驱动器的性能	171
9.4 数据缓存机制	172
9.5 编写更好的 SQL 语句	173
9.5.1 在 SELECT 语句中滥用子查询	174
9.5.2 尽量避免使用 SELECT * 语法	176
9.5.3 善用 CASE 语法	177
9.5.4 使用 Filter 语法替代 CASE 语法	178
第 10 章 复制与外部数据	180
10.1 复制功能概览	180
10.1.1 复制功能涉及的术语	181
10.1.2 复制机制的演进	182
10.1.3 第三方复制解决方案	182
10.2 复制环境的搭建	183
10.2.1 主服务器的配置	183
10.2.2 从属服务器的配置	184
10.2.3 启动复制进程	185
10.3 外部数据封装器	186
10.3.1 查询平面文件	186
10.3.2 以不规则数组的形式查询不规范的平面文件	187
10.3.3 查询其他 PostgreSQL 服务实例上的数据	188
10.3.4 查询非传统数据源	190
附录 A PostgreSQL 的安装	192
附录 B PostgreSQL 自带的命令行工具	196
作者简介	204
封面介绍	204

前言

PostgreSQL (<http://www.postgresql.org/>) 是一个开源的关系型数据库管理系统，最初源于加州大学伯克利分校的一个研究项目。该系统最早是基于 BSD 许可证发布的，但目前已改为使用 PostgreSQL 许可证（简称 TPL）发布。事实上这两种许可证无论从哪方面看都没有区别。PostgreSQL 的悠久历史可追溯到 1985 年。

PostgreSQL 拥有诸多企业级特性，比如支持窗口函数（用户可以自定义聚合函数并当作窗口函数使用）、普通 CTE 表达式、递归 CTE 表达式以及流式复制等。这些特性在 Oracle、SQL Server、DB2 等较新版本的商用数据库中很常见，但在开源数据库界却几乎没有。另外，PostgreSQL 有一点与众不同，它可以在不用重编译任何代码的情况下轻松实现系统功能的扩展。PostgreSQL 不但支持众多高级特性，而且性能也很好，在很多应用场景下其性能甚至可以超越包括商用数据库在内的大多数数据库。

本书将介绍 PostgreSQL 的诸多高级特性，其中有的特性是 ANSI SQL 标准中所规定的，而有的特性是 PostgreSQL 自己独创的。如果你当前正在使用 PostgreSQL，又或者以前曾用过但了解程度一般，那么通过本书可以学到之前可能错过的一些功能“遗珠”，还可以了解到最新几个版本中引入的新特性。本书适合对关系型数据库有一定使用经验的读者，但不要求使用过 PostgreSQL。书中将对比 PostgreSQL 与其他数据库处理同一任务的机制，同时也将展示只有 PostgreSQL 才支持的一些“高大上”功能，这些功能在别的数据库中要么实现起来很困难，要么根本不可能实现。如果你完全未使用过数据库，通过本书也可以学到 PostgreSQL 的功能和使用方法。不过，鉴于本书的定位，书中不会过多介绍关于 SQL 或者关系型数据库理论方面的基础知识，我们建议你阅读其他相关书籍来了解这些内容。

本书主要介绍 PostgreSQL 9.2、9.3 和 9.4 版，但也会覆盖一些在更早版本中已支持的高级特性。

本书读者

我们希望本书对数据库行业现有从业人员以及刚刚开始从事数据库领域工作的读者能够有所帮助。具体来说，本书面向的读者群如下。

- 如果你正在学习关系型数据库，那么我们希望本书能够对你有所帮助，并希望你将在将来的职业生涯中青睐于 PostgreSQL。为了降低难度，我们在本书第 2 版中对许多专题进行了拓展介绍，并尽可能提供了入门级的例子。
- 如果你当前已经是 PostgreSQL 的用户或者 DBA，那么我们希望此书能让你工作更加得心应手。书中会有你已经熟悉的内容，但也一定会有你不熟悉的一些技巧，以及新版本中引入的新特性，如果善加利用，必定会提高你的工作效率。好吧，如果你真的很出色并且对书中内容均已了解，那么本书对你来说依然有价值，为什么呢？因为它比官方的 PostgreSQL 手册要轻上 20 倍，最起码是便于携带了。
- 如果你还没接触过 PostgreSQL，那么本书除了能够向你介绍 PostgreSQL 知识外，还将扮演你身边的 PostgreSQL “布道师”的角色，这位“布道师”将向你证明：多绑在商业数据库上一天，你就会被那些厂商掏走更多的钱；多用那些“烂”数据库一天，你的系统就不得不多做一天功能上的妥协。

如果你的工作与数据库领域甚至是 IT 界毫无关系，又或者你刚刚幼儿园毕业，那么能否购买本书呢？答案依然是可以的！因为封面上可爱的象鼩鼠图片就已经让本书物有所值了。

PostgreSQL 的特别之处以及选用理由

PostgreSQL 之所以特别，是因为它不仅仅是一个数据库，还是一个功能强大的应用开发平台。

PostgreSQL 支持用多种编程语言编写存储过程和函数。除了系统自带的编程语言外，还可以通过安装语言扩展包来支持新的语言。内置的基础语言包括 SQL 和 PL/pgSQL，通过安装扩展包还可以支持 PL/Perl、PL/Python、PL/V8（又称为 PL/JavaScript）以及 PL/R 等。前述语言的安装包在 PostgreSQL 发行版中大多都已自带，需要时安装一下扩展包即可使用。这种支持多语言的能力是非常有价值的，因为每种语言的特点不同，有的语言适合解决特定领域内的问题，有的语言过程化特性更丰富，有的语言语法特性更强大，那么开发人员就可以根据待解决问题的特点来选择最合适的语言。比如可以通过使用 R 语言的统计和图形函数以及 R 语言中简洁的专业表达式来解决统计领域的问题，可以通过 Python 来调用 Web 服务，也可以编写 map reduce 函数并在 SQL 语句中调用。

用户甚至可以写一个聚合函数并嵌入到上述这些语言中，这样就可以把 SQL 语言的聚合运算能力与上述语言本身的能力相结合，从而拓展了这些语言的能力范围。此外，PostgreSQL 还支持调用 C 语言写的函数，与调用上述其他语言写的函数没有区别。可以在

同一个 SQL 语句中调用分别由不同语言编写的多个函数。甚至可以仅使用 SQL 语言（即无过程化能力的纯 SQL）来创建一个用户自定义聚合函数。在 MySQL 和 SQL Server 中，用户自定义聚合函数是需要对数据库软件本身进行重编译才能做到的，但在 PostgreSQL 中却不需要。简而言之，PostgreSQL 对多语言的支持是极其灵活和强大的，开发人员不但能够为不同任务选择不同的编程语言，也能够为同一任务的不同子任务选用不同的语言。另外，在其他绝大多数数据库都不允许使用 SQL 的场景下，PostgreSQL 也允许你使用 SQL。PostgreSQL 中不用编译任何代码就可以创建非常复杂的函数。

PostgreSQL 支持非常强大的用户自定义数据类型功能，不但使用方法简单，而且性能也超越了绝大多数其他关系型数据库。在用户自定义数据类型方面，与 PostgreSQL 实力最接近的对手只有 Oracle。用户可以在 PostgreSQL 中定义新的数据类型，然后就可以将该数据类型用作表列类型。每种数据类型都有伴随的数组类型，这样可以将某个类型的数组存储到某个数据列中，或者在 SQL 语句中使用该数组。另外还可以为新增的数据类型定义相应的运算符、函数和索引绑定来与其协同工作。很多 PostgreSQL 的第三方扩展包就利用该自定义数据类型能力来优化性能，或者通过添加支持某个领域专用的特殊 SQL 语法来让业务代码更简洁和易于维护，或者实现一些在别的数据库中完全不可能实现的功能。

如果不需要自定义数据类型和相应的函数，那么系统自带的数据类型也是种类繁多的，比如 json（在 9.2 版中引入），另外还有很多数据类型扩展包可供选择。很多这类扩展包是 PostgreSQL 发行版中自带的。从 PostgreSQL 9.1 开始引入了一种新的语法：CREATE EXTENSION。该语法仅通过一条 SQL 语句就实现了扩展包的安装。如果需要在某个数据库中使用某个扩展包的功能，则在该数据库中安装该扩展包即可。通过 CREATE EXTENSION 语法，可以安装前述任何一种过程式语言（简称 PL），以及使用比较广泛的数据类型及其相应的函数和运算符，比如 hstore 键-值存储、ltree 层次化存储、PostGIS 地理空间扩展以及数不胜数的其他扩展。举个例子，如果想加载 hstore，只需执行下面这个命令即可：

```
CREATE EXTENSION hstore;
```

此外，有一条 SQL 命令可以列出所有可用的以及已安装的扩展包（详见 2.6 节）。

前面提到了 PostgreSQL 支持各种扩展，也提到了它对多种编程语言的支持，但你可能对这些都不感兴趣。你可能觉得：“哦，支持 Python，还支持 Perl……可那又怎么样？能不能说点我不知道的？”别着急，当我们继续往后深入学习的时候，你一定能够不时地体会到“哇，这个功能太牛了！”这样的惊喜，而这样的惊喜在我们多年使用 PostgreSQL 的过程中已经体会过太多次。PostgreSQL 的每次升级都会为用户提供新的特性，这些特性包括易用性的升级、性能的提升，以及对于关系型数据库功能极限的不断超越。到最后你会发现：我为什么还要使用别家的数据库？PostgreSQL 已经提供了我所需要的一切功能了啊，而且还是免费的！你不再需要去阅读那些商业数据库附带的密密麻麻的授权条款，来了解

在一个 8 核服务器上支持甲特性、乙特性和丙特性所需要的费用是多少，也不需要了解如果服务器 CPU 从 8 核升级到 16 核后要再为许可证加多少钱。

此外，PostgreSQL 在其支持的所有平台上的功能表现都很一致。因此你根本不需要担心你的客户要求支持哪种操作系统，Unix、Linux、Mac OS X、Windows，PostgreSQL 全都支持。PostgreSQL 官方站点提供各种操作系统下的二进制安装包下载，当然你也可以自行编译安装。

不适用PostgreSQL的场景

PostgreSQL 从一开始就被设计为一个通用的事务型数据库。很多人把它用在小型的桌面应用程序中，就类似一个 SQL Server Express 免费版或者是 Oracle Express 免费版。但这种用法存在的问题是：作为一个独立的数据库系统，PostgreSQL 自身会进行全面的安全管理（比如用户登录机制，这个机制在嵌入式场景下是不需要的），这些都是要耗费性能的，但 PostgreSQL 又无法取消安全机制并将其交由上层应用去管理，因此单用户应用场景下 PostgreSQL 可能不是最好的选择。此种场景下 SQLite 或者 Firebird 是更合适的选择，因为用户权限管理、安全检查和 DB 操作日志功能都是由上层应用自己完成的。

令人遗憾的一个事实是，很多共享主机（多个租户共享同一个操作系统）供应商并不支持预安装 PostgreSQL，或者只支持安装一个很陈旧的版本。因此，如果在使用共享主机服务，你可能就不得不使用 MySQL。本书第 1 版出版以后，这个情况有了很大改善。目前虚拟专用主机和云服务器（每个租户独享一个操作系统，多租户之间互不干扰）的租用价格已经趋于合理，而且会越来越便宜。与共享主机相比，其价格不会高出很多，而且可以在上面安装任何你希望安装的软件。因此选择专用的云主机服务会是比较明智的做法，因为你可以安装 PostgreSQL 的最新稳定版，而无需受制于与其他租户共享主机时的种种限制。此外，主流的 PaaS 平台均已支持 PostgreSQL，而且一般都支持最新的发行版。这些主流平台包括：SalesForce Heroku PostgreSQL、Engine Yard、Red Hat OpenShift，以及 Amazon RDS for PostgreSQL。

PostgreSQL 的功能极其强大，强大到“令人生畏”。它绝对不是一套仅仅能做些数据存储的小软件，它是如同一只聪慧的大象般智能又强大的大型系统。如果你所需要的仅仅是一个键-值存储，或者一个只要能装数据就行的小软件，那么 PostgreSQL 必定会远比你的期望强大百倍。

如何获取本书使用的数据和示例代码

可从本书的官方链接 (http://www.postgresql.org/downloads/postgresql_book_2e.zip) 获取。如果发现所提供的数据有误，请将相关信息发布到本书的勘误页面 (<http://www.oreilly.com/catalog/errata.csp?isbn=0636920032144>)。

关于PostgreSQL的更多信息

本书致力于展示 PostgreSQL 区别于其他数据库的独特功能，以及如何使用这些功能来解决现实世界的问题。你将学到如何使用一些你在数据库领域前所未见的方法来解决问题。除了那些“高大上”的内容，我们也会向你展示如何处理一些基本的任务，比如数据库管理、权限设置、性能问题定位、性能调优、使用不同的桌面工具连接到数据库、命令行操作以及开发工具的使用，等等。

PostgreSQL 有一套内容丰富的在线文档。本书不会重复这些文档中已提供的信息，而是会鼓励你去探索博大精深的 PostgreSQL 中的未知领域。官方手册 (<http://www.postgresql.org/docs/manuals>) 目前包含 2250 多页的内容，同时提供 HTML 和 PDF 两种格式。此外，如果你需要的话，最近几个版本的官方手册还提供纸质印刷版。由于其规模庞大，内容丰富，纸质版通常是分成 3 到 4 册提供的。

其他可用的 PostgreSQL 资源还包括：

- Planet PostgreSQL (<http://planet.postgresql.org>) 是 PostgreSQL 技术博客文章的汇聚站点，其中包含从 PostgreSQL 核心开发人员到普通用户编写的各类文章，包括新特性演示以及对现有功能的使用说明；
- PostgreSQL Wiki (<https://wiki.postgresql.org>) 提供对 PostgreSQL 各个方面的使用技巧说明，以及从其他数据库移植到 PostgreSQL 的方法；
- PostgreSQL Books (<http://www.postgresql.org/docs/books/>) 提供有关 PostgreSQL 的书籍列表信息；
- PostGIS in Action Books (<http://www.postgis.us/>) 是我们已出版的关于 PostGIS 的书籍的官方站点。

代码与输出格式

对于括号中的内容，我们一般会将左括号与之前的内容放置于同一行，右括号单独放置一行，以便于纵栏印刷。格式如下：

```
function ( Welcome to PostgreSQL
);
```

为节省版面，我们还移除了命令行执行输出结果中无意义的空格，因此如果发现实际输出结果的格式与书中提供的不一致，请不用担心，这是正常的。

请注意，虽然我们建议在编码时逗号后要加一个空格，但本书中的确有些地方因版面宽度的关系而去掉了这种空格。

PostgreSQL 的 SQL 解释器会将语句中的制表符、换行符和回车符当作空格处理。在我们

提供的示例代码中，一般会使用空格而不是制表符作为缩进符。请确保使用的代码编辑器不会自动将制表符、换行符和回车符删除，或者把它们转换为空格以外的什么字符，否则会导致问题。

如果在执行示例代码时遇到了问题，请检查确认你复制过来的代码与我们提供的原始代码是否一致。

注意有些示例适用于 Linux，而有些适用于 Windows。例如某些关于外部数据封装器的例子中要求使用带完整路径的文件名，你会看到示例代码中有类似于 `/postgresql_book/somefile.csv` 这样的路径，这指的是 Linux 服务器根目录下的路径。如果使用的是 Windows 环境，那么需要加上驱动器符，因此路径要改为：`C:/postgresql_book/somefile.csv`。请注意：即使是在 Windows 上，路径中也应该使用 Linux 的路径分隔符 `/`，而不是 Windows 传统的 `\`。

排版约定

本书使用了下列排版约定。

- 楷体
表示新术语。
- 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 bold`*)
表示应替换成用户提供的值或由上下文决定的值。



该图标表示提示或建议。



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 http://www.postgresql.org/docs/9.4/book_2e.zip 下载。

本书是要帮你完成工作的。一般来说，如果本书提供了示例代码，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，否则无需联系我们获得许可。比如，用本书的几个代码片段写一个程序就无需获得许可，销售或分发 O'Reilly 图书的示例光盘则需要获得许可；引用本书中的示例代码回答问题无需获得许可，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：“*PostgreSQL: Up and Running, Second Edition* by Regina Obe and Leo Hsu (O'Reilly). Copyright 2015 Regina Obe and Leo Hsu, 978-1-4493-7319-1.”

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。

Safari[®] Books Online



Safari Books Online (<http://www.safaribooksonline.com>) 是应运而生的数字图书馆。它同时以图书和视频的形式出版世界顶级技术和商务作家的专业作品。技术专家、软件开发人员、Web 设计师、商务人士和创意专家等，在开展调研、解决问题、学习和认证培训时，都将 Safari Books Online 视作获取资料的首选渠道。

对于组织团体、政府机构和个人，Safari Books Online 提供各种产品组合和灵活的定价策略。用户可通过一个功能完备的数据库检索系统访问 O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、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 以及其他几十家出版社的上千种图书、培训视频和正式出版之前的书稿。要了解 Safari Books Online 的更多信息，我们网上见。

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，你可以在那儿找到本书的相关信息，包括勘误表、示例代码以及其他信息。本书的网站地址是：

<http://shop.oreilly.com/product/0636920032144.do>

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>

本章将带你开始 PostgreSQL 的探索之旅。首先将介绍如何下载和安装 PostgreSQL，然后会讲到一些必备的管理工具和 PostgreSQL 术语。本书写作之时，PostgreSQL 9.4 正在等待发布，我们将重点介绍该版本的一些新特性。在本章的末尾，我们将给出一些帮助资源列表，在遇到问题时你可以从中获得帮助。

1.1 如何获得PostgreSQL

若干年前，你只能通过手动编译源码的方式来安装 PostgreSQL。还好那种痛苦的时代已经一去不复返了。当然，现在依然可以通过编译源码来安装，但大多数用户会使用制作好的安装包来安装，只需敲击几下键盘和鼠标就可以了。

如果你是首次安装 PostgreSQL，那么应该选择适用于你的操作系统平台的最新稳定版发行包。PostgreSQL 官方站点的核心发布页面上维护了一个列表 (<http://www.postgresql.org/download>)，记录了适用于各操作系统的二进制包的下载地址。在附录 A 中，你将会看到安装指导和一些特殊定制过的版本的下载链接地址。

1.2 管理工具

PostgreSQL 常用的管理工具有四种：psql、pgAdmin、phpPgAdmin 和 Adminer。PostgreSQL 的核心开发团队维护着前三种，因此它们一般会随着 PostgreSQL 的版本发布而同步更新。Adminer 并非 PostgreSQL 的专用管理工具，它支持管理多种类型的关系型数据库，包括 SQLite、MySQL、SQL Server 和 Oracle 等。除了我们刚刚提到的这四种以外还有大量优秀

的管理工具，开源的和商业的都有。

1.2.1 psql

psql 是一种用于执行查询的命令行工具，每个 PostgreSQL 发行版中都自带 psql。它有一些独特的功能，比如导入或者导出基于分隔符（分隔符可以是逗号或者制表符等字符）格式的平面数据文件，以及生成简易的 HTML 格式报表等。psql 是 PostgreSQL 从诞生之初就一直附带的命令行工具，它是很多资深 PostgreSQL 专家日常操作工具的不二之选，非常适用于只有控制台字符界面而无图形用户界面的使用场景。另外在通过 shell 脚本执行数据库操作时，psql 也是必备工具。不过新用户一般更喜欢使用图形界面工具，而且也无法理解为什么“老”一代人会对命令行方式那么执着。

1.2.2 pgAdmin

pgAdmin (www.pgadmin.org) 是一种广泛使用的开源 PostgreSQL 图形界面管理工具。如果你的 PostgreSQL 安装包里没有附带此工具，请从其官网单独下载安装。

pgAdmin 运行于图形化桌面环境下，可以同时连接到多个 PostgreSQL 服务器上，这些服务器可以是安装在任意操作系统平台上的任意 PostgreSQL 版本。

即使你的数据库安装在只有字符控制台界面的 Linux 服务器上，只要你在本地工作站上安装了 pgAdmin，也可以用这种强大的图形化工具对其进行管理。

图 1-1 是 pgAdmin 的界面示意图。

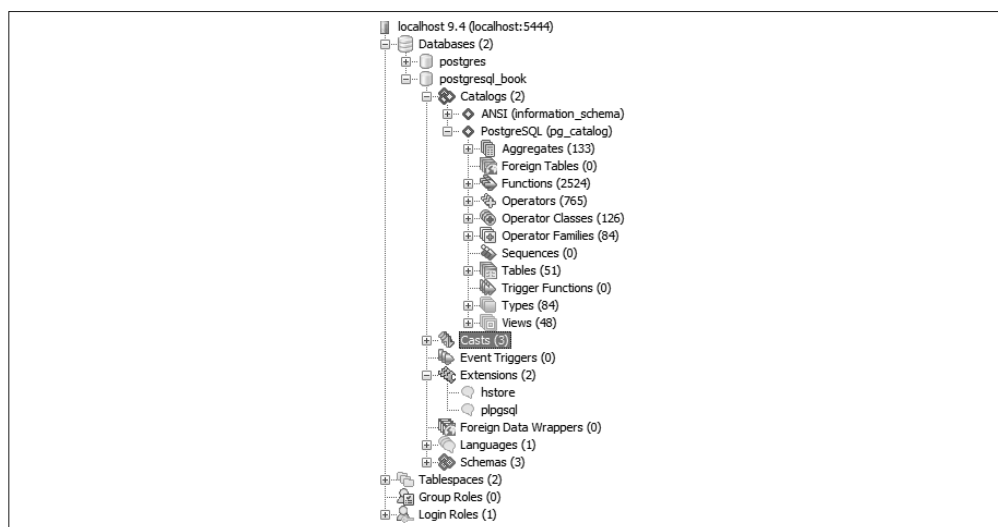


图 1-1: pgAdmin

如果你对 PostgreSQL 还不太熟悉，那么 pgAdmin 毫无疑问将是你开始 PostgreSQL 学习之旅的最佳入口。只需要在主界面上摸索一下，你就可以对 PostgreSQL 的丰富功能一览无遗。如果你正打算逃离 SQL Server 阵营，并且习惯于 SQL Server 的 Management Studio，那么很快就能适应 pgAdmin。

1.2.3 phpPgAdmin

phpPgAdmin (<https://github.com/phpPgAdmin/phpPgAdmin>) 是一种免费的基于 Web 页面的管理工具，其界面如图 1-2 所示。它是从流行的 MySQL 管理工具 phpMyAdmin 移植而来的，二者的差别主要在于 phpPgAdmin 新增了对 schema、过程化语言、类型转换器、运算符等对象的管理功能。如果你对 phpMyAdmin 很熟悉，会发现 phpPgAdmin 的界面风格与其完全类似。

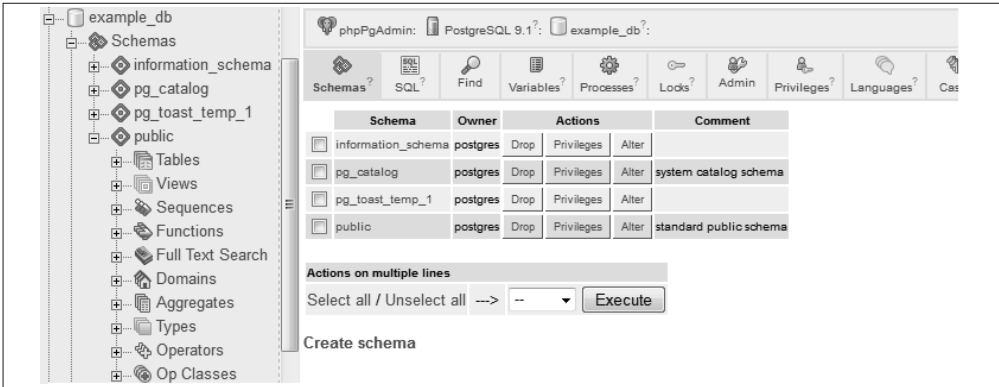


图 1-2: phpPgAdmin

1.2.4 Adminer

如果你正在寻找一款除了能够管理 PostgreSQL，还能管理别的数据库的整合型工具，那么 Adminer (<http://www.adminer.org/>) 将是你合适的选择。Adminer 是一款轻量级的开源 PHP 应用程序，可以在同一套图形界面上管理 PostgreSQL、MySQL、SQLite、SQL Server 以及 Oracle 等多种数据库。

Adminer 有一种独特的功能让我们印象深刻：它能够以图形化方式展示数据库中的对象，并将外键约束关系以连接线的方式展示出来。另外，整个 Adminer 程序的本体仅包含一个 PHP 文件，非常简洁，这可以大大减少你安装部署时的麻烦。

图 1-3 中，左侧是登录屏幕的截图，右侧是表间关系图形化后呈现的效果。很多用户会因为登录屏幕上没有填写端口号的地方而感到困惑。如果 PostgreSQL 使用标准的 5432 侦听端口，那么登录时不填也没问题；但如果不是，那么就需要在服务器名称后面加上端口

号，注意用冒号分隔主机名和端口号，如图 1-3 所示。

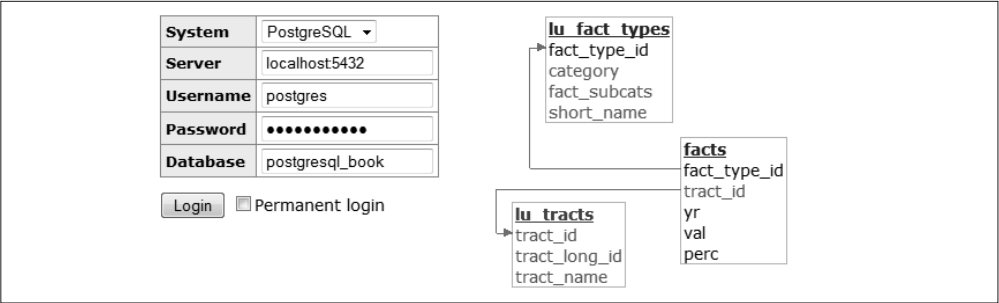


图 1-3: Adminer

对于简单的查询和修改操作来说，Adminer 的功能是足够的。但为了支持多种数据库，Adminer 的功能体系已经被裁剪成了各数据库均支持的最小公共集合，因此你无法实现 PostgreSQL 所特有的一些操作，比如创建新用户、授予权限或者是查询当前权限列表等。如果你是 DBA，那么建议使用 pgAdmin，当然也可以安装一套 Adminer 以备不时之需。

1.3 PostgreSQL数据库对象

假设你现在已经安装好了 PostgreSQL，请启动并连接好 pgAdmin，然后点开左侧的目录树，此时展现在你面前的是一堆令人眼花缭乱的数据对象，有些你可能很熟悉，有些则可能闻所未闻。PostgreSQL 对象类型的数量超过了绝大多数关系型数据库（这还是在未安装任何扩展包的情况下）。这些对象中，有许多你可能永远都不会用到，但如果你发现业务上需要实现一种新的对象类型，那么一般来说你要实现的东西在那一堆眼花缭乱的对象中已经有前人实现过了，所以只需要正确选用即可。本书不会介绍 PostgreSQL 以标准方式安装完毕后所提供的所有对象类型，因为 PostgreSQL 引入新特性的速度惊人，任何一本书都不可能全面覆盖所有对象类型。因此我们仅讨论你有必要了解的那些对象类型。

- 服务
在大多数操作系统上，PostgreSQL 是作为一种服务（或者叫守护进程）安装的。多个 PostgreSQL 服务可以运行于同一台物理服务器上，但它们的侦听端口不能重复，也不能共享同一个数据存储目录。本书中，server（服务器）和 service（服务）这两个术语可互换使用，因为大多数人在一台物理服务器上只会安装一个服务。
- database¹
每个 PostgreSQL 服务可以包含多个独立的 database。

注 1：database 一词含义宽泛，既可表示广义的数据库系统，又可以表示某些特定数据库系统中的某一级数据存储单位，如表述不当极易给读者造成混淆。因此本书中会区别使用，表示广义的数据库系统时，用中文“数据库”；表示狭义的数据存储单位时，用英文“database”。——译者注

- schema²

ANSI SQL 标准中对 schema 有着明确的定义，database 的下一层逻辑结构就是 schema。如果把 database 比作一个国家，那么 schema 就是一些独立的州（或者是省、府、辖区等，具体取决于各国的实际情况）。大多数对象是隶属于某个 schema 的，然后 schema 又隶属于某个 database。在创建一个新的 database 时，PostgreSQL 会自动为其创建一个名为 public 的 schema。如果未设置 search_path 变量（后续会介绍该变量的含义），那么 PostgreSQL 会将你创建的所有对象默认放入 public schema 中。如果表的数量较少，这是没问题的，但如果你有几千张表，那么我们还是建议你将它们分门别类放入不同的 schema 中。

- catalog³

catalog 是系统级的 schema，用于存储系统函数和系统元数据。每个 database 创建好以后默认都会含有两个 catalog：一个名为 pg_catalog，用于存储 PostgreSQL 系统自带的函数、表、系统视图、数据类型转换器以及数据类型定义等元数据；另一个是 information_schema，用于存储 ANSI 标准中所要求提供的元数据查询视图，这些视图遵从 ANSI SQL 标准的要求，以指定的格式向外界提供 PostgreSQL 元数据信息。

一直以来，PostgreSQL 数据库的发展都严格地遵循着其“自由与开放”的核心理念。如果你足够了解这款数据库，会发现它几乎是一种可以“自我生长”的数据库。比如，它所有的核心设置都保存在系统表中，用户可以不受限制地查看和修改这些数据，这为 PostgreSQL 提供了远超任何一种商业数据库的巨大灵活性（不过从另一个角度看，将这种灵活性称为“可破坏性”也未尝不可）。只要仔细地研究一下 pg_catalog，你就可以了解到 PostgreSQL 这样一个庞大的系统是如何基于各种部件构建起来的。如果你有超级用户权限，那么可以直接修改 pg_catalog 的内容（当然，如果改得不对，那你的行为就跟搞破坏没什么两样了）。

Information_schema catalog 在 MySQL 和 SQL Server 中也有。PostgreSQL Information_schema 中最常用的视图一般有几个：columns 视图，列出了数据库中的所有表列；tables 视图，列出了数据库中的所有表（包括视图）；views 视图，列出了所有视图以及用于构建或重新构建该视图的关联 SQL。同样，在 MySQL 和 SQL Server 中也有这些视图，不过它们所含的列没有 PostgreSQL 那么多。PostgreSQL 另外添加了几个用于描述自定义数据类型列的列，比如 columns.udt_name。

注 2：数据库业界对于 schema 有多种译法：纲要、模式、方案，等等。但各种译法都不能准确直观地表达出其原本的含义，即位于一个独立命名空间内的一组相关数据库对象的集合，因此前述译法从来没有一种成为主流。一般业界人员都直接使用英文 schema。考虑到这个情况，为防止初级用户理解困难，我们也按照业界习惯直接使用英文原名。——译者注

注 3：catalog 的译法与 schema 存在相同的问题，翻译为“目录”后并不能让读者准确地理解其原意，反而容易造成混淆，因此还是沿用英文原名。——译者注

尽管 `columns`、`tables`、`views` 这三个元数据视图本身也是标准的 PostgreSQL 视图，但由于其身份的特殊性，pgAdmin 界面中还是把它们放在了 `information_schema` → `Catalog Objects` 分支下。

- 变量

变量是 PostgreSQL 统一配置机制（GUC）的一部分，是可以在多个级别进行设置的各种选项，这些级别包括服务级、`database` 级以及其他级别。很多人容易在 `search_path` 这个选项上犯错，该选项的工作机制是：如果在 `search_path` 中指定了 `schema` 的名称，那么该 `schema` 资产在使用时就无需显式指定其 `schema` 名，系统会按照 `search_path` 中登记的 `schema` 名按顺序逐个搜索。2.4.2 节会对 `search_path` 进行详细讨论。

- 扩展包

PostgreSQL 9.1 版中引入了对于扩展包机制的支持。开发人员可以通过该机制将一组相关的函数、数据类型、数据类型转换器、用户自定义索引、表以及 GUC 等对象打包成一个功能扩展包，该扩展包可以整体安装或者整体删除。扩展包在概念上与 Oracle 的 `package` 类似，一般推荐使用该机制来为数据库提供功能扩展。关于扩展包的具体安装步骤，请参考开发手册。一般来说需要先将扩展包的二进制安装包和脚本复制到服务器，然后再为需要该扩展包功能的 `database` 单独安装。

假设你需要用到某个扩展包的功能，那么仅需将其安装到对应的 `database` 中即可，而不必为当前数据库系统中的每个 `database` 都安装一遍。比如需要对某个 `database` 中的数据进行高级文本搜索，那么单独为该 `database` 安装 `fuzzystrmatch` 扩展包即可。安装扩展包时可以指定装到哪个 `schema`，若不指定则默认会装到 `public` `schema` 中。我们不建议这么做，因为这会导致 `public` `schema` 变得庞大复杂且难以管理，尤其是如果你将自己的数据库对象也都存入 `public` `schema` 中，那么情况会变得更糟糕。我们建议你创建一个独立的 `schema` 用于存放所有扩展包的对象，甚至为规模较大的扩展包单独创建一个 `schema`。为避免出现找不到新增扩展包对象的问题，请将这些新增的 `schema` 名称加入 `search_path` 变量中，这样就可以直接使用扩展包的功能而无需关注它到底装到了哪个 `schema` 中。也有一些扩展包明确要求必须安装到某个 `schema` 下，这种情况下你就不能自行指定了。例如有很多语言扩展包，比如 `plv8`，就要求必须安装到 `pg_catalog` 中。

- 表

任何一个数据库中，表都是最核心和最“忙碌”的对象类型。在 PostgreSQL 中，表首先属于某个 `schema`，而 `schema` 又属于某个 `database`，这样就构成了一种三级存储结构。

PostgreSQL 的表支持两种很强大的功能。第一种是表继承，即一张表可以有父表和子表。这种层次化的结构可以极大地简化数据库设计，还可以为你省掉大量的重复查询代码。我们将在本书后面的示例 6-2 中介绍表继承机制。

第二种是创建一张表的同时，系统会自动为此表创建一种对应的自定义数据类型。换句话说，你可以将某个完整的数据结构定义为一个表，然后将该表用作另一个表中的一个列。关于这种复合数据类型的更多细节，请参见 5.8 节。

- 外部表和外部数据封装器

外部表的首次亮相是在 9.1 版中。它们是一些虚拟表，通过它们可以直接在本地数据库中访问来自外部数据源的数据。只要数据映射关系配置正确，那么外部表的用法就与普通表没有任何区别。外部表支持映射到以下类型的数据源：CSV 文件、另一个服务器上的 PostgreSQL 表、SQL Server 或 Oracle 这些异构数据库中的表、Redis 这样的 NoSQL 数据库或者甚至像 Twitter 或 Salesforce 这样的 Web 服务。外部表映射关系的建立是通过配置外部数据封装器（Foreign Data Wrapper, FDW）实现的。FDW 是 PostgreSQL 和外部数据源之间的一架“魔法桥”，可实现两边数据的互联互通。其内部实现机制遵循 SQL 标准中的 MED（Management of External Data）规范，更多细节请参考维基百科上关于 MED 的描述（<http://en.wikipedia.org/wiki/SQL/MED>）。

许多编程人员已经为当下绝大部分流行的数据源开发了 FDW 并已免费共享出来。你也可以通过创建自己的 FDW 来练习。（我们建议你一旦成功了也公布出来，这样整个社区都可以分享你的劳动成果。）FDW 是通过扩展包机制实现的，装好以后在 pgAdmin 界面上名为 Foreign Data Wrapper 的目录节点下能看到它。

- 表空间

表空间是用于存储数据的物理空间。PostgreSQL 将用于物理存储的表空间和用于逻辑存储的 schema 分开管理，二者之间并无耦合关系。这样就很容易在不影响业务应用逻辑的情况下，将 database 甚至是单张表和索引在不同的物理驱动器之间进行移动。

- 视图

大多数关系型数据库都支持视图，通过视图可以大大简化复杂的查询逻辑，另外也可以通过对视图执行更新操作来修改其基表数据。PostgreSQL 也不例外，从 9.3 版开始支持对基于单表的视图直接使用 SQL 进行更新操作，这样就不需要再写额外的规则或者触发器来实现对此类简单视图的更新。但对于包含更复杂逻辑的视图，或者是基于多张表的视图，对其进行数据更新操作时仍需编写规则或者触发器。9.3 版还引入了对物化视图的支持，该机制通过对视图数据进行缓存来实现对常用查询的加速。更多细节请参见 7.1.3 节。

- 函数

PostgreSQL 中函数执行后的返回结果可以是一个标量值或几个记录集。可以在函数中对表数据进行修改，其他数据库对于这种会修改表记录的函数一般称为存储过程。

- 内置编程语言

函数是以过程化语言（Procedural Language, PL）编写的。PostgreSQL 默认支持三

种内置编程语言：SQL、PL/pgSQL 以及 C 语言。可以通过 `CREATE EXTENSION` 或者 `CREATE PROCEDURAL LANGUAGE` 命令来安装其他语言包。目前较常用的语言有 Python、JavaScript、Perl 以及 R。第 8 章中有大量的相关示例。

- 运算符

运算符本质上是符号化的已命名函数（例如 `=`、`&&` 等），它需要一个或者两个实参（argument），底层有一个相应的函数来实现其运算逻辑。PostgreSQL 支持自定义运算符。如果你定义了自己的数据类型，那么可定义一些运算符来与之配合工作。比如可以为自定义数据类型定义 `=` 运算符。你甚至可以为两个完全不同的数据类型定义一个运算符，来对其进行比较运算。

- 数据类型（或者仅仅类型）

每种数据库产品都会支持一系列的数据类型，比如整型、字符型、数组，等等。除前述常见类型外，PostgreSQL 还支持复合数据类型，这种类型可以是多种数据类型的一个组合，比如虚数、极坐标、张量都是复合数据类型。如果你定义了自己的复合数据类型，那么可以创建一组函数和运算符来配合它工作，可以做得很专业，很复杂，很“高端”，比如自定义实现 `div`（散度运算）、`grad`（梯度运算）和 `curls`（旋度运算）等。哪位读者若有兴趣，可以试一下。

- 数据类型转换器

`cast` 是数据类型转换器，就是将一种数据类型转换为另一种类型的工具。转换器在其底层其实是通过调用转换函数来实现真正的转换逻辑的。PostgreSQL 的独到之处在于支持用户自定义转换器，这样就可以改变系统默认的转换行为。例如，如果你需要把邮政编码（美国的邮政编码是一个 5 位的整数）从 `integer` 转换为 `character`，那么可以自定义一个支持“数字不足 5 位则前面自动补 0”规则的转换器。转换器可以被隐式调用也可以被显式调用。隐式转换是系统自动执行的，一般来说，将一种特定数据类型转为更通用的数据类型（比如数字转换为字符串）时就会发生隐式类型转换。如果进行隐式转换时系统找不到合适的转换器，你就必须显式执行转换动作。

- 序列

序列控制 `serial` 数据类型的自动递增。在 PostgreSQL 中定义 `serial` 列时，PostgreSQL 会自动创建序列，但你很容易更改初始值、增量和下一个值。因为序列是独立对象，所以多个表可以使用同一个序列对象。这样你可以创建跨越多个表的独特键值。SQL Server 和 Oracle 也都有序列对象，但必须手动创建。

- 行或记录

本书中，“行”和“记录”这两个术语可互换使用。在 PostgreSQL 中，“记录”这个概念可以脱离表而独立存在。当你在函数或者在 SQL 语句中使用记录构造函数语法（语法类似于：`SELECT ROW(1,2.5,'this is a test')`）时，会对这一点有深刻体会。

- 触发器

绝大多数企业级数据库都支持触发器机制，该机制可以实现对数据修改事件的捕获，并在之后触发用户自定义的操作行为。触发器的触发时机是可设置的，可以是语句级触发或者记录级触发，可以是修改前触发也可以是修改后触发。

PostgreSQL 的触发器技术正在快速的演进之中。9.0 版引入了对 WITH 子句的支持，通过它可以实现带条件的记录级触发，即只有当某条记录符合指定的 WHEN 条件时，触发器才会被调用。9.0 版还引入了 UPDATE OF 子句，通过它可以指定要监控哪些列的更改。当列更改时，就会触发触发器，详情请参见第 8 章中的示例 8-11。在 9.1 版中，视图中的数据更改可以触发触发器。在 9.3 版中，数据定义语言（DDL）事件可以触发触发器。目前支持触发器的 DDL 命令列表请参见官方手册中“触发器触发时机一览表”（<http://www.postgresql.org/docs/9.4/interactive/event-trigger-matrix.html>）。在 9.4 版中，针对外部表的触发器也获得了支持。请参考官方文档中“创建触发器”（<http://www.postgresql.org/docs/current/interactive/sql-createtrigger.html>）这一节的内容以获取更多信息。

- 规则

规则是一种能够将一种动作替换为另一种动作的机制。PostgreSQL 内部就是使用此机制来实现视图的。比如我们定义了这样一个视图：

```
CREATE VIEW vw_pupils AS SELECT * FROM pupils WHERE active;
```

实际上 PostgreSQL 在后台自动创建了一个 INSTEAD OF SELECT 类型的规则，其内容是：当查询名为 vw_pupils 的表时，自动重定向为查询 pupils 表中 active 字段值为 true 的记录。

规则还可以用于替代一些特定的简单触发器。通常情况下，在对记录行进行更新 / 插入 / 删除操作时会调用触发器。规则却不一样，它的运作机理是修改用户原本的行为逻辑（也就是你执行的 SQL 语句本身），或者是在用户原有逻辑的基础上额外附加一些 SQL 逻辑。相比触发器而言，这种整体取而代之的方式避免了针对每条记录都调用一次触发器所造成的负担。一般来说，如果你希望在数据修改时加载自定义逻辑，我们还是推荐使用触发器而不是规则。很多 PostgreSQL 用户认为规则是过时的技术，因为出问题时很难诊断，而且规则只支持用 SQL 语法来编写，PostgreSQL 所支持的其他编程语言则无用武之地。

1.4 最新版本 PostgreSQL 中引入的新特性

PostgreSQL 的版本发布是很有规律的，每年的 9 月份会发布一个大版本。每个新版本都会带来易用性、稳定性、安全性、性能等方面的提升，以及一些试验性质的功能。而且版本升级过程也变得越来越简单。那么显而易见，请尽量把你的数据库及时升级到最新的稳定

版。关于每个版本引入的关键特性列表，请参见官方提供的“PostgreSQL 各版本功能特性一览表” (<http://www.postgresql.org/about/featurematrix>)。

1.4.1 为什么要升级

如果你正在使用 PostgreSQL 8.4 或者更早期的版本，请立即升级！因为 8.4 版在 2014 年 7 月已进入生命周期终结（End of Life, EOL）状态，此后不再提供官方的升级支持。请参考 PostgreSQL 官方的发行版支持策略 (<http://www.postgresql.org/support/versioning/>) 以获取更多关于 PostgreSQL EOL 政策的细节。请务必不要使用已过了 EOL 期限的版本，因为开发组不会再为其提供新的安全更新和功能补丁。一旦这种老版本出了问题，你只能花钱去请 PostgreSQL 专家级顾问来解决故障或寻找临时解决方案，这种服务一般都是很昂贵的，而且你不一定能找得到这种专家。

不管当前使用的是哪个大版本，你都应该尽快跟进小版本号的更新。比如从 8.4.17 升级到 8.4.21，只需要替换二进制文件并重启一下即可。小版本仅修改 bug 而不会涉及功能变化，因此这种升级是很安全的，也会为你降低出问题的几率。

1.4.2 PostgreSQL 9.4版中引入的新特性

本书写作期间，PostgreSQL 9.3 是最新的稳定发行版，9.4 还只是测试版，但其安装包已可下载用于测试。9.4 测试版包含以下特性。

- 物化视图特性的改进。在 9.3 版中，刷新物化视图期间会对其加锁并禁止访问。但一般来说刷新物化视图会需要一定的时间，因此在生产环境中该刷新动作会导致物化视图可用性显著降低。9.4 版中取消了刷新时的加锁动作，因此即使是正在被刷新的物化视图也可被访问。但请注意：利用此特性的前提是物化视图必须要拥有一个唯一索引。
- 新增了对 SQL:2008 标准中规定的 `percentile_disc`（不连续百分比）和 `percentile_cont`（连续百分比）这两个分析函数的支持，须配合 `WITHIN GROUP (ORDER BY...)` 子句使用。详细例子可参见 [depsz 博客网站](http://www.depesz.com/2014/01/11/waiting-for-9-4-support-ordered-set-within-group-aggregates/) 的一篇关于 `ORDERED SET WITHIN GROUP` 聚合运算的介绍文章 (<http://www.depsz.com/2014/01/11/waiting-for-9-4-support-ordered-set-within-group-aggregates/>)。这些函数为你提供了系统原生的快速取中间值功能。例如，如果我们希望从一批考试成绩中取中间点到 3/4 处部分的值，可执行以下查询：

```
SELECT subject, percentile_cont(ARRAY[0.5, 0.75])
       WITHIN GROUP (ORDER BY score) As med_75_score
FROM test_scores GROUP BY subject;
```

在 PostgreSQL 中，要实现 `percentile_cont` 和 `percentile_disc`，可以取一个数组或 0 到 1 之间的单个值（此值代表所希望查询的百分比范围），并且此实现会相应地返回一个值数组或单个值。语句中的 `ORDER BY score` 表示希望根据 `score` 字段的值来进行百分比计算。

- 创建视图时支持 `WITH CHECK OPTION` 子句，其作用是确保在视图上执行更新或者插入操作时，修改后或者新插入的记录仍然是在本视图可见范围内。详见第 7 章中的示例 7-2。
- 新增对 `jsonb` 数据类型的支持，该数据类型是 JSON (JavaScript Object Notation) 类型的二进制存储版本，并且支持索引。通过 `jsonb` 类型可以对 JSON 格式的文档数据建立索引，并可加快对其内部元素的访问速度。详细信息请参考 5.6 节，同时可参考这两篇博客文章：“引入新的 `jsonb` 数据类型：JSON 类型的结构化存储格式” (<http://www.depesz.com/2014/03/25/waiting-for-9-4-introduce-jsonb-a-structured-format-for-storing-json/>) 以及“`jsonb`：通配符查询” (<http://obartunov.livejournal.com/177977.html>)。
- GIN 索引的查询速度提升，同时占用空间减少。GIN 索引的使用范围日益广泛，并且非常适用于全文搜索、三连词处理、`hstore` 键值数据库以及 `jsonb` 类型支持等场景。在很多情况下你甚至可以把它当作 B- 树索引的一个替代品，而且一般来说 GIN 索引占用的空间会更少。详情请参见“使用 GIN 索引来代替位图索引”这篇文章的介绍 (<http://hlinnaka.iki.fi/2014/03/28/gin-as-a-substitute-for-bitmap-indexes>)。
- 支持更多 JSON 函数。请参见 Depesz 博客站的文章“9.4 版中的新 JSON 函数介绍” (<http://www.depesz.com/2014/01/30/waiting-for-9-4-new-json-functions/>)。
- 支持使用以下语法轻松地将所有资产从一个表空间移动到另一个表空间中：`ALTER TABLESPACE old_space MOVE ALL TO new_space;`。
- 支持对返回的结果集中的记录加上数字编号。当我们从数组、`hstore`、复合类型等格式数据源中取出非格式化数据时，由于缺少可用于唯一标识记录的主键，因此一般需要为每条记录加一个数字型的行号。现在可以将系统列 `ordinality`（该列是在 ANSI SQL 标准中定义的）添加到输出。以下是一个使用 `hstore` 对象以及返回一个键值对的 `each` 函数的例子：

```
SELECT ordinality, key, value
FROM each('breed=>pug,cuteness=>high'::hstore) WITH ordinality;
```

- 支持通过执行 SQL 命令来更改系统配置设置。`ALTER system SET ...` 语法可实现对全局系统设置的动态修改，这一功能在之前版本中只能通过修改 `postgresql.conf` 文件才能实现。关于 `postgresql.conf` 文件用法的详情请参见 2.1.1 节。
- 支持对外部表建触发器。通过该功能，即便外部数据源与你相隔万里之遥，只要对方一修改数据，你立即就可以得到通知。不过目前我们尚不确定该功能的实际使用效果到底如何，因为在数据源极其遥远的情况下，由于存在网络延迟，其效果就很难说了。
- 新增 `unnest` 函数，该函数以可预见的方式将不同大小的数组分配到各个列中。
- 新增 `ROWS FROM` 语法，该语法可以将多个函数返回的结果集逐行拼接起来，最后作为一个完整的结果集返回，因此即使这些结果集之间的元素个数不一致也没关系，如下例所示：

```
SELECT * FROM ROWS FROM (
  jsonb_each('{ "a": "foo1", "b": "bar" }'::jsonb),
```

```
jsonb_each('{\"c\":\"foo2\"}':::jsonb)) x  
(a1,a1_val,a2_val);
```

- 支持使用 C 对动态后台工作线程进行编码以按需完成工作。contrib/worker_spi 目录下的 9.4 版源码中实现了一个小型的示例可供参考。

1.4.3 PostgreSQL 9.3版新特性列表

9.3 版（发布于 2013 年）中引入的主要特性如下。

- 增加了对 ANSI SQL 的标准 LATERAL 子句的支持。LATERAL 语法允许将 FROM 子句与关联一起使用以引用关联的另一方上的变量。在一个关联查询语句中，如果参与关联的一方是一个临时结果集，比如一个子查询或者一个能够返回结果集的函数，那么在该子查询或者函数中可以引用关联关系另一方结果集中的列。如果没有 LATERAL 语法，这种横向关联是不可能的，只有 WHERE 关联条件部分才能进行被关联方之间列的交叉引用。如果你需要使用 unnest、generate_series 或者以正则表达式作为查询条件的查询语句来构造关联一方的结果集，那么 LATERAL 语法几乎是不可或缺的，因为这种情况下势必要参考关联关系另一方的数据才能生成自己一方的数据。请参考 7.6 节以了解更多细节。
- 支持并行 pg_dump。从 8.4 版开始支持并行恢复，在该版本中实现了对并行备份的支持，这可以大大加快大型数据库的备份速度。
- 支持物化视图（详细信息请参见 7.1.3 节）。物化视图可以将经常使用的视图中的数据进行持久化，从而避免反复执行相同的查询。
- 支持自动可更新视图。对视图执行更新操作不再需要创建触发器或者规则，现在可以直接针对视图执行 UPDATE 操作，该操作在系统内部会自动映射到此视图的基表上。
- 支持定义基于递归 CTE 表达式的视图。
- 支持更多针对 JSON 类型的构造函数和解析函数。详见 7.6 节。
- 支持在基于正则表达式条件的查询中使用索引。
- 引入了支持 64 位大对象操作的 API，该 API 可操作 TB 级大小的对象。之前最多支持到 2 GB。
- postgres_fdw 驱动程序（10.3.3 节中会进行介绍）支持对其他 PostgreSQL 数据库（即使在使用较低版本 PostgreSQL 的远程服务器上）进行读写操作。伴随此更改的是对实施可写功能的 FDW API 的升级。
- 复制功能做了大量改进，其中最主要的两点是：实现了复制功能的架构无关性，即复制的架构不依赖于特定的操作系统或者硬件；另外支持了基于流式复制的重新选主过程。
- 支持使用 C 语言创建后台工作线程，可用于执行一些定时任务。
- 支持对 DDL 操作定义触发器。
- 支持新的 psql 命令：watch。详细信息参见 3.4.2 节。
- 支持新的 COPY DATA 命令，可用于 PostgreSQL 与外部程序之间导入或者导出数据。3.5.3 节有详细介绍。

1.4.4 PostgreSQL 9.2版新特性列表

9.2 版（2012 年 9 月发布）中引入的主要特性如下。

- 支持索引内查询。如果查询的列都位于索引内，那么 PostgreSQL 会省略不必要的表访问动作，仅依据索引自身数据就可以完成全部查询动作。该优化会给键值查询以及类似 COUNT(*) 这种聚合查询带来巨大的性能提升。
- 内存排序操作性能提升可达 20%。
- 在预处理语句中做了一些改进。现在会对预处理语句进行解析、分析和重写，但你可以跳过该计划来避免被绑定到特定实参输入上。你现在还可以保存依赖实参的某个预处理语句的计划。这样会降低预处理语句比等效的即席查询执行得更差的几率。⁴
- 支持级联流式复制，即支持从一个 slave 节点到另一个 slave 节点的流式复制。
- 支持基于空间分区树算法的 SP-GiST 索引，这种索引是 GiST 索引的强化版本，对于很多依赖 GiST 索引的扩展包来说，使用这种索引可以得到巨大的性能提升。
- 新增了对 ALTER TABLE IF EXISTS 语法的支持，修改表结构之前可以不用再检查此表是否存在。
- ALTER TABLE 和 ALTER TYPE 这两个命令新增了大量功能选项，本来要实现这些功能的话需要删除表并重建。更多细节请参见 depesz 博客网站的“More Alter Table Alter Types”这篇博文 (<http://www.depesz.com/2012/02/14/waiting-for-9-2-more-rewrite-less-alter-table-alter-types/>)。
- 增加了更多的 pg_dump 和 pg_restore 选项。细节请参见“9.2 版中 pg_dump 工具的增强”(http://www.postgresqlonline.com/journal/archives/252-PostgreSQL-9.2-pg_dump-enhancements.html)。
- 新增了对 PL/V8 语法扩展包的支持，以后可以使用目前非常流行的 JavaScript 语言来编写函数。
- JSON 正式成为系统内置数据类型，同时转正的还有 row_to_json 和 array_to_json 函数。使用 Ajax 的 Web 开发人员应该会很欢迎该特性。详见 7.6 节和示例 7-16。
- 新增了对区间数据类型的支持，不再需要通过编写函数来实现类似功能。区间类型虽然

注 4：语句的 PREPARE 预解析机制有改进。在之前版本中进行 PREPARE 操作时，规划器会对预解析的语句生成执行计划，但由于此时的语句中不携带具体的选项值（预提交的语法形式如 PREPARE my_stmt AS select a from tab where b=\$1 and c>\$2），因此规划器会基于表的统计信息来进行推测，并生成一个通用的、比较保守的执行计划，这往往会导致得到的执行计划并非最优，从而出现语句经过预解析后再执行反而比直接执行更慢的情况。为解决此问题，当前版本中的 PREPARE 操作有所变化，仍然会执行语句解析、语句分析、语句改写这几个过程，但会跳过生成执行计划这一步骤，留到语句 EXECUTE 阶段根据真正的选项再生成执行计划，这样就保证了执行计划是最优的。但这带来的另一个问题就是 EXECUTE 效率降低，因为每次都要生成执行计划。为此系统做了另一个优化：如果 EXECUTE 阶段生成的执行计划与原来在 PREPARE 阶段生成的通用执行计划效率类似，那么系统还是会将通用执行计划存储下来并复用。通过以上策略，避免了有时执行预提交的语句反而更慢的情况。——译者注

是首次引入，但系统为其提供了丰富的运算符和配套函数。Exclusion 类型的约束可以很好地保证该类型数据的合法性。

- 支持在 SQL 函数中通过实参名称引用实参，之前版本仅支持通过实参编号引用实参。有多个实参时，通过实参名称引用会更方便。

1.4.5 PostgreSQL 9.1版新特性列表

在 9.1 版中，PostgreSQL 推出了诸多企业级特性，开始与 SQL Server 和 Oracle 等重量级对手展开正面竞争。

- 内置的复制特性全面增强，支持同步复制。
- 引入新的 CREATE EXTENSION 和 ALTER EXTENSION 命令来管理扩展包，安装和卸载扩展包变得轻而易举。
- 引入兼容 ANSI 标准的外部数据封装器（FDW）机制，用于查询外部数据源。
- 支持可写 CTE 表达式，此后 UPDATE 和 INSERT 语句也可以享受到 CTE 语法带来的便利。
- 支持建立无日志的表，以加快写操作速度。有的场景下，记录预写日志确无必要。
- 支持针对视图建立触发器。在之前的版本中，如果希望对视图进行修改操作，只能通过建立 DO INSTEAD 类型的规则（详情请参见官网链接：<http://www.postgresql.org/docs/current/interactive/rules-update.html>）来实现，且这种规则只能用 SQL 语言编写，但支持视图触发器后，就可以用上 PostgreSQL 支持的所有语言，这样就可以针对视图实现更复杂更抽象的业务逻辑。
- 引入 KNN GiST 索引类型，该类索引能对常用的很多扩展包带来性能提升，比如全文检索、三连词（用于模糊查找以及区分大小写查找操作）以及 PostGIS 等。

1.5 数据库驱动程序

任何情况下，你都不可能脱离具体的业务系统而仅仅使用 PostgreSQL 数据库本身，那显然是无意义的。为了实现 PostgreSQL 与业务系统之间的交互，就需要借助数据库驱动程序。PostgreSQL 拥有大量免费驱动，支持各种编程语言和开发工具。此外，很多商业公司也以很低廉的价格提供了各有特色的驱动。目前比较流行的几种开源驱动如下。

- PHP 驱动：PHP 语言广泛应用于 Web 开发领域，大多数 PHP 发行包都自带了较老的 pgsql 驱动或者是较新的 pdo_pgsql 驱动。一般来说这两种驱动默认都会安装，不过可能需要修改 php.ini 来决定启用哪一种。
- JDBC 驱动：JAVA 开发所使用的 JDBC 驱动一直是与最新版 PostgreSQL 同步更新的，可以从 PostgreSQL 官方站点下载 (<https://jdbc.postgresql.org>)。
- .NET 驱动：.NET 框架(含微软的官方版和 Mono 社区的开源版)可使用 Npgsql 驱动(<http://npgsql.projects.pgfoundry.org>)。目前该驱动支持微软 .NET 框架 V3.5 及之后的版本，包括微软 Entity Framework 开发框架以及 Mono 开源 .NET 框架。

- ODBC 驱动：如果需要从微软 Access、Office 系列工具或者其他支持 ODBC 的产品连接到 PostgreSQL，可从 PostgreSQL 官网下载 ODBC 驱动 (<http://www.postgresql.org/ftp/odbc/versions/msi>)，支持 32 位和 64 位两个版本。
- LibreOffice/OpenOffice 驱动：LibreOffice 3.5 及之后的版本中自带了 PostgreSQL 驱动，但 3.5 之前的版本以及 OpenOffice 是不带的，可以使用 JDBC 或者 SDBC 驱动。更多细节请参见“OO Base 与 PostgreSQL”这篇博文 (<http://www.postgresqlonline.com/journal/categories/23-oobase>)。
- Python 驱动：Python 可通过多种驱动 (<https://wiki.postgresql.org/wiki/Python>) 访问 PostgreSQL，目前 psycopg (<http://initd.org/psycopg/>) 是最流行的一种。Python 的 Django 开发框架 (<https://docs.djangoproject.com/en/dev/ref/databases/#postgresql-notes>) 对 PostgreSQL 也有着良好的支持。
- Ruby 驱动：对 Ruby 开发人员来说，请使用 rubygems pg 驱动 (<https://rubygems.org/gems/pg>)。
- Perl 驱动：Perl 可以使用 DBI 和 DBD::Pg 驱动。也可以使用由 CPAN 网站 (http://search.cpan.org/modlist/Database_Interfaces) 提供的 DBD::PgPP 驱动。
- Node.js 驱动：Node.js 是一个基于 Google V8 JavaScript 引擎的运行平台，可用于构建高可扩展性的网络应用。该平台目前支持三种 PostgreSQL 驱动：Node Postgres (<https://github.com/brianc/node-postgres>)、Node Postgres Pure（与前者的区别在于不需要编译，<https://github.com/brianc/node-postgres-pure>），以及 Node-DBI (<https://github.com/DrBenton/Node-DBI>)。

1.6 如何获得帮助

在使用 PostgreSQL 过程中，你迟早会需要寻求帮助，我们希望你能尽早了解到求助的途径。我们最为推荐的途径是邮件列表，不管你是 PostgreSQL 的新用户还是老用户，邮件列表都能为你解答技术问题。可以先打开 PostgreSQL 邮件列表页面 (<http://www.postgresql.org/community/lists/>)，该页面上有各种邮件列表的信息以及订阅方法。如果你是新手，那么订阅 PGSQL-General 这个邮件列表（该邮件列表的历史信息归档地址：<http://archives.postgresql.org/pgsql-general>）是最合适的。如果你发现了 PostgreSQL 的 bug，那么打开“PostgreSQL 故障报告”这个页面 (<http://www.postgresql.org/docs/current/interactive/bug-reporting.html>)，上面会告诉你具体如何操作。

1.7 PostgreSQL 的主要衍生版本

PostgreSQL 使用了 MIT/BSD 风格的许可证，任何人都可以合法地对其修改并二次传播，因此对于那些想创建自己数据库分支的人来说，PostgreSQL 是绝佳的选择。在过去的很多年间，有很多团队创建了自己的 PostgreSQL 衍生版本，其中的部分修改也已经回馈到

PostgreSQL 的主干代码中。

目前数据仓库领域使用很广泛的 Netezza (<http://www.netezza.com>) 就是源自 PostgreSQL。亚马逊公司的 Redshift 数据仓库 (<http://aws.amazon.com/redshift/>) 事实上是 PostgreSQL 的一个分支的分支。支持 PB 级数据分析的著名数据仓库 GreenPlum 最初的源头是 Bizgres, 而 Bizgres 是一款基于 PostgreSQL 的面向大数据的数据仓库和智能分析软件。EnterpriseDB 公司 (<http://enterprisedb.com>) 推出的 PostgreSQL Advanced Plus 也是以 PostgreSQL 为基础, 另外增加了对于 Oracle 语法和特性的兼容支持, 以吸引原 Oracle 用户。EnterpriseDB 公司向 PostgreSQL 社区提供了资金和开发力量的支持, 对此我们表示感谢。他们的 Postgres Plus Advanced Server 产品在版本更新节奏上也一直是密切跟进最新的 PostgreSQL 稳定版的。

前述衍生产品都是商业化的闭源软件。tPostgres (<http://www.tpostgres.org>)、Postgres-XC (<http://postgres-xc.sourceforge.net>) 和 Big SQL (<http://www.bigsql.org>) 是三款还处于发展初期但已经崭露头角的开源衍生产品, 它们都得到了 OpenSCG 公司 (<http://www.openscg.com/>) 的资金支持。tPostgreSQL 的最新版本基于 PostgreSQL 9.3, 其目标是取代 Microsoft SQL Server。tPostgreSQL 中内嵌了 pgtsql 语言扩展包, 可以用 T-SQL 语法来编写函数。pgtsql 语言包是标准的 PostgreSQL 扩展包, 因此它其实可以安装到任何一台 PostgreSQL 9.3 数据库上。Postgres-XC 是一套集群服务器系统, 它能够提供可扩展的写能力并支持同步多主复制, 其分布式处理和多主复制能力使它在所有类似系统中脱颖而出。目前 Postgres-XC 还只是 1.0 版本。最后介绍一下 BigSQL, 它实现了 PostgreSQL 和 Hadoop with Hive 这两款重量级产品的融合。BigSQL 自带了 `hadoop_fdw` 这款扩展包, 可以查询和更新外部 Hadoop 数据源的数据。

此外, 最近还发布了一款 PostgreSQL 开源分支产品 Postgres-XL (XL 代表 eXtensible Lattice, 即可扩展的晶格。<http://www.postgres-xl.org/>), 该产品面向大规模并行处理 (MPP) 领域, 支持节点间数据的分片存储能力。