

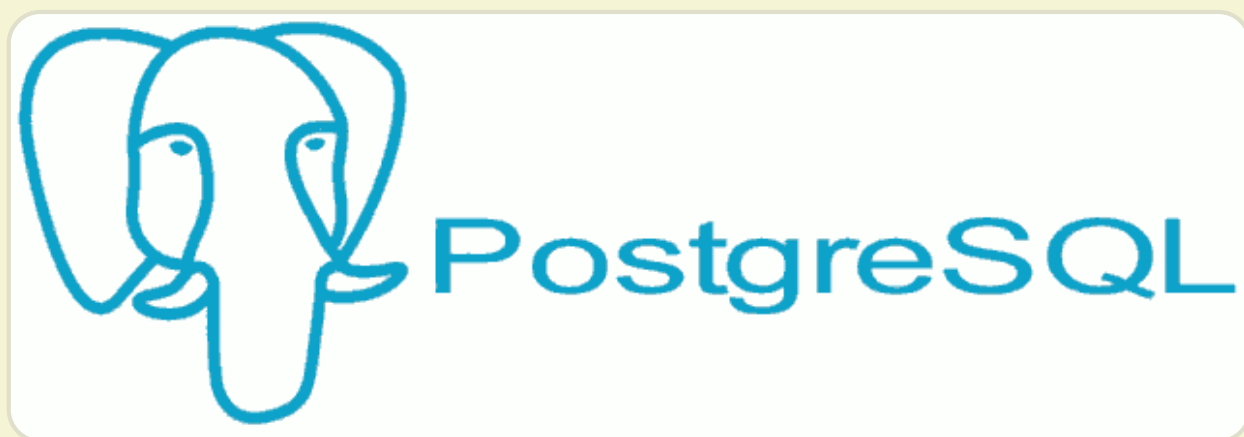
PostgreSQL新手入门

作者： 阮一峰

日期： 2013年12月22日

自从MySQL被Oracle收购以后，[PostgreSQL](#)逐渐成为开源关系型数据库的首选。

本文介绍PostgreSQL的安装和基本用法，供初次使用者上手。以下内容基于Debian操作系统，其他操作系统实在没有精力兼顾，但是大部分内容应该普遍适用。



一、安装

首先，安装PostgreSQL客户端。

```
sudo apt-get install postgresql-client
```

然后，安装PostgreSQL服务器。

```
sudo apt-get install postgresql
```

正常情况下，安装完成后，PostgreSQL服务器会自动在本机的5432端口开启。

如果还想安装图形管理界面，可以运行下面命令，但是本文不涉及这方面内容。

```
sudo apt-get install pgadmin3
```

二、添加新用户和新数据库

初次安装后，默认生成一个名为**postgres**的数据库和一个名为**postgres**的数据库用户。这里需要注意的是，同时还生成了一个名为**postgres**的Linux系统用户。

下面，我们使用**postgres**用户，来生成其他用户和新数据库。好几种方法可以达到这个目的，这里介绍两种。

第一种方法，使用**PostgreSQL**控制台。

首先，新建一个Linux新用户，可以取你想要的名字，这里为**dbuser**。

```
sudo adduser dbuser
```

然后，切换到**postgres**用户。

```
sudo su - postgres
```

下一步，使用**psql**命令登录**PostgreSQL**控制台。

```
psql
```

这时相当于系统用户**postgres**以同名数据库用户的身份，登录数据库，这是不用输入密码的。如果一切正常，系统提示符会变为"**postgres=#**"，表示这时已经进入了数据库控制台。以下的命令都在控制台内完成。

第一件事是使用**\password**命令，为**postgres**用户设置一个密码。

```
\password postgres
```

第二件事是创建数据库用户**dbuser**（刚才创建的是Linux系统用户），并设置密码。

```
CREATE USER dbuser WITH PASSWORD 'password';
```

第三件事是创建用户数据库，这里为**exampledb**，并指定所有者为**dbuser**。

```
CREATE DATABASE exampledb OWNER dbuser;
```

第四件事是将exampledb数据库的所有权限都赋予dbuser，否则dbuser只能登录控制台，没有任何数据库操作权限。

```
GRANT ALL PRIVILEGES ON DATABASE exampledb to dbuser;
```

最后，使用\q命令退出控制台（也可以直接按ctrl+D）。

```
\q
```

第二种方法，使用**shell**命令行。

添加新用户和新数据库，除了在PostgreSQL控制台内，还可以在shell命令行下完成。这是因为PostgreSQL提供了命令程序createuser和createdb。还是以新建用户dbuser和数据库exampledb为例。

首先，创建数据库用户dbuser，并指定其为超级用户。

```
sudo -u postgres createuser --superuser dbuser
```

然后，登录数据库控制台，设置dbuser用户的密码，完成后退出控制台。

```
sudo -u postgres psql
```

```
\password dbuser
```

```
\q
```

接着，在shell命令行下，创建数据库exampledb，并指定所有者为dbuser。

```
sudo -u postgres createdb -O dbuser exampledb
```

三、登录数据库

添加新用户和新数据库以后，就要以新用户的名义登录数据库，这时使用的是psql命

令。

```
psql -U dbuser -d exampledb -h 127.0.0.1 -p 5432
```

上面命令的参数含义如下：**-U**指定用户，**-d**指定数据库，**-h**指定服务器，**-p**指定端口。

输入上面命令以后，系统会提示输入dbuser用户的密码。输入正确，就可以登录控制台了。

psql命令存在简写形式。如果当前Linux系统用户，同时也是PostgreSQL用户，则可以省略用户名（**-U**参数的部分）。举例来说，我的Linux系统用户名为ruanyf，且PostgreSQL数据库存在同名用户，则我以ruanyf身份登录Linux系统后，可以直接使用下面的命令登录数据库，且不需要密码。

```
psql exampledb
```

此时，如果PostgreSQL内部还存在与当前系统用户同名的数据库，则连数据库名都可以省略。比如，假定存在一个叫做ruanyf的数据库，则直接键入psql就可以登录该数据库。

```
psql
```

另外，如果要恢复外部数据，可以使用下面的命令。

```
psql exampledb < exampledb.sql
```

四、控制台命令

除了前面已经用到的\password命令（设置密码）和\q命令（退出）以外，控制台还提供一系列其他命令。

- \h: 查看SQL命令的解释，比如\h select。
- \?: 查看psql命令列表。
- \l: 列出所有数据库。
- \c [database_name]: 连接其他数据库。

- \d: 列出当前数据库的所有表格。
- \d [table_name]: 列出某一张表格的结构。
- \du: 列出所有用户。
- \e: 打开文本编辑器。
- \conninfo: 列出当前数据库和连接的信息。

五、数据库操作

基本的数据库操作，就是使用一般的SQL语言。

创建新表

```
CREATE TABLE user_tbl(name VARCHAR(20), signup_date DATE);
```

插入数据

```
INSERT INTO user_tbl(name, signup_date) VALUES('张三', '2013-12-22');
```

选择记录

```
SELECT * FROM user_tbl;
```

更新数据

```
UPDATE user_tbl set name = '李四' WHERE name = '张三';
```

删除记录

```
DELETE FROM user_tbl WHERE name = '李四' ;
```

添加栏位

```
ALTER TABLE user_tbl ADD email VARCHAR(40);
```

更新结构

```
ALTER TABLE user_tbl ALTER COLUMN signup_date SET NOT NULL;
```

更名栏位

```
ALTER TABLE user_tbl RENAME COLUMN signup_date TO signup;
```

删除栏位

```
ALTER TABLE user_tbl DROP COLUMN email;
```

表格更名





```
ALTER TABLE user_tbl RENAME TO backup_tbl;
```

删除表格

```
DROP TABLE IF EXISTS backup_tbl;
```

(完)

文档信息

- 版权声明：自由转载-非商用-非衍生-保持署名（[创意共享3.0许可证](#)）
- 发表日期：2013年12月22日
- 更多内容： [档案](#) » [开发者手册](#)
- 购买文集： 《如何变得有思想》
- 社交媒体： twitter,  weibo
- Feed订阅：

相关文章

- **2016.03.08:** [Systemd 入门教程：实战篇](#)

上一篇文章，我介绍了 Systemd 的主要命令，今天介绍如何使用它完成一些基本的任务。

- **2016.03.07:** [Systemd 入门教程：命令篇](#)

Systemd 是 Linux 系统工具，用来启动守护进程，已成为大多数发行版的标准配置。

- **2016.02.28:** [Linux 守护进程的启动方法](#)

"守护进程"（daemon）就是一一直在后台运行的进程（daemon）。

- **2016.01.06:** [Commit message 和 Change log 编写指南](#)

Git 每次提交代码，都要写 Commit message（提交说明），否则就不允许提交。

