PostgreSQL 安装及操作指南

第一部分 PostgreSQL数据库使用说明

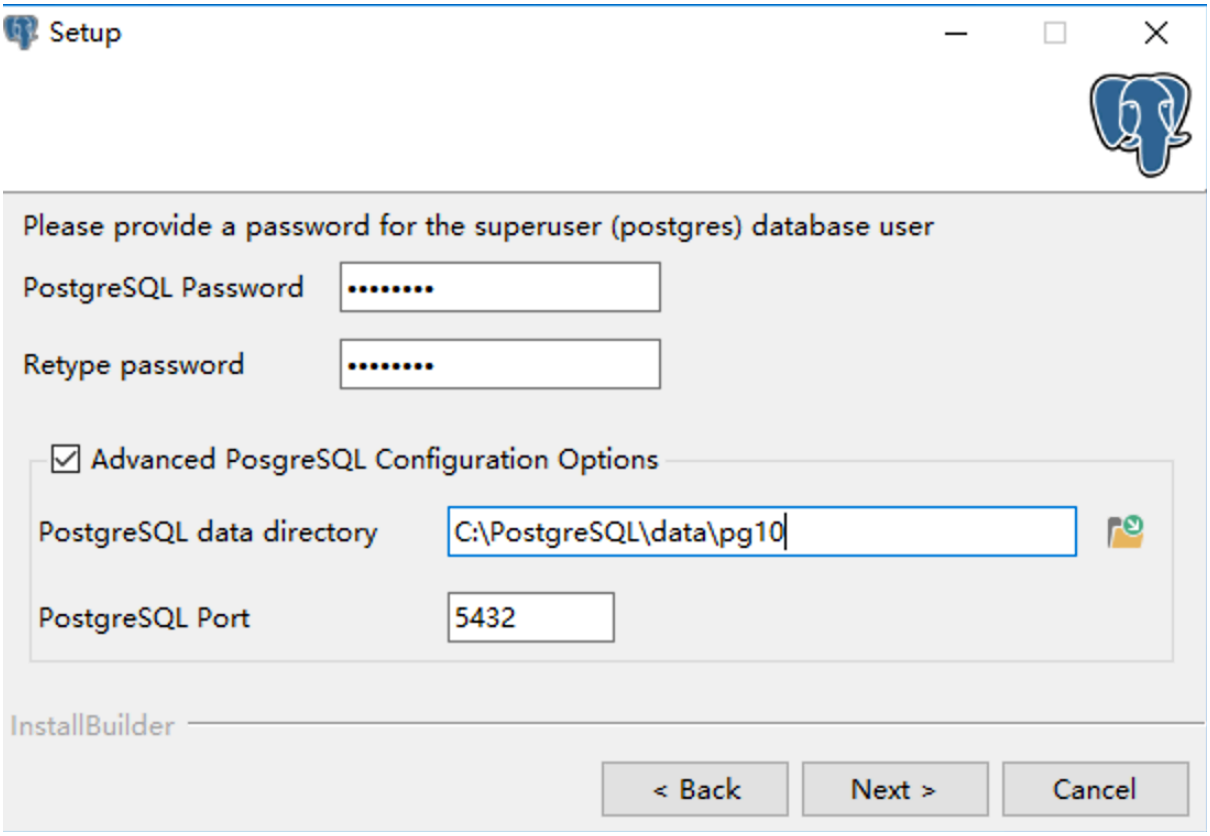
1.1 下载

下载页面：https://www.openscg.com/bigsql/postgresql/installers.jsp/

1.2 安装

直接双击安装包，一路 Next 直到以下界面

在这里，密码请填写 `postgres` ，然后把下面的 Advanced 那个勾给勾上



然后一路 Next 就可以了

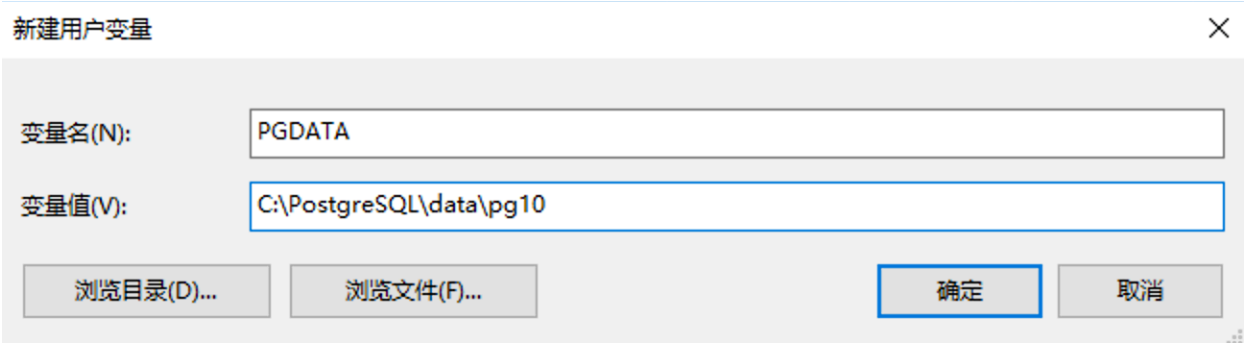
安装完成以后，打开命令行，输入 `pg_ctl` ，然后回车，出现以下输出就是安装成功了

1.3 配置

添加环境变量：

变量名 `PGDATA`

变量值 `C:\PostgreSQL\data\pg10`



1.4 操作指南

启动服务器： `pg_ctl start`

> 如果出现服务器启动失败，请仔细检查其输出信息，根据输出信息来排查。一般是服务器未关闭导致端口被占用，或者是上次关机时没关服务器导致 lock 文件未删除。

关闭服务器： `pg_ctl stop`

创建用户： `createuser --username=name -P -s --username=postgres`

> 此处前面的 `--username` 要创建的用户名，后面的 `--username` 是现有的超级用户。输入这条命令后会让你输入一系列密码，前两次是要创建的用户密码，第三次是超级用户的密码，是在安装时输入的那个密码。下同。

创建数据库，指定用户为 dbuser，数据库名为 exampledb，当前用户为 postgres

```
sudo createdb -O dbuser exampledb --username=postgres
```

登录数据库

```
psql -U username -d dbname -h 127.0.0.1 -p 5432
```

系统用户身份和数据库用户相同时，可以直接登录数据库

```
psql exampledb
```

显示已创建的数据库

```
psql -l
```

1.5 操作流程

以下操作启动了数据库服务器，然后创建了 annotator 用户，以及 annotator 数据库。

> 注：
>
> 1. annotator 用户的密码请指定为 annotator
> 2. 需要以管理员权限运行命令行

```
# 启动服务器
pg_ctl start
# 创建用户
createuser -P -s --username=postgres annotator
# 创建数据库
createdb -O annotator annotator --username=postgres
# 连接数据库
psql -U annotator -d annotator
```

在用户和数据库已经存在的情况下，只需要启动服务器，连接数据库就可以使用数据库的控制台命令了。

```
# 启动服务器
pg_ctl start
# 连接数据库
psql -U annotator -d annotator -h 127.0.0.1 -p 5432
```

1.6 控制台命令

```
\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。
```

也可以直接输入 SQL 语句进行操作。

1.7 SQL 语句

```
# 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;
```

第二部分 数据库接口介绍 DB_Manager

首先需要安装 `psycopg2` ，以允许 python 程序访问 PostgreSQL 数据库。执行以下命令：

```
pip install psycopg2
```

2.1 代码结构介绍

程序中一共有三个类和两个测试函数

类： `DB_Manager, Labeled_DB_Manager, Unlabeled_DB_Manager`

测试函数： `test_unlabeled_db, test_labeled_db`

其中， `DB_Manager` 为后面两个类的基类，提供了访问数据库最基础的操作，比如用户名和密码，数据库的名字，数据库的连接等等

`Labeled_DB_Manager` 和 `Unlabeled_DB_Manager` 均实现了数据库的增删改查，可以在测试函数中查看 demo 程序

2.2 主要函数介绍

```
@abstractmethod
def create(self):
    创建数据表
```

```
@abstractmethod
def insert(self):
    向数据表中插入记录

def delete(self, condition=''):
    删除满足条件的记录
    :param condition: 删除条件

def update(self, ret, condition):
    更新数据库记录
    :param ret: 更新结果
    :param condition: 更新条件

def select(self, condition='', num=-1):
    选择满足条件的记录
    :param condition: 选择条件
    :param num: 返回数量
    :return: 一定数量的选择结果

def drop(self):
    删除数据表
```

第三部分 数据表结构介绍

3.1 已标注数据表 labeled_data

```
labeled_id integer 主键 自动递增
unlabeled_id integer # 数据的 id 号
data_content text
labeled_time timestamp
entity1 text
entity2 text
predicted_relation text
labeled_relation text
```

3.2 未标注数据表 unlabeled_data

```
unlabeled_id integer 主键 自动递增
data_content text
uploaded_time timestamp
```