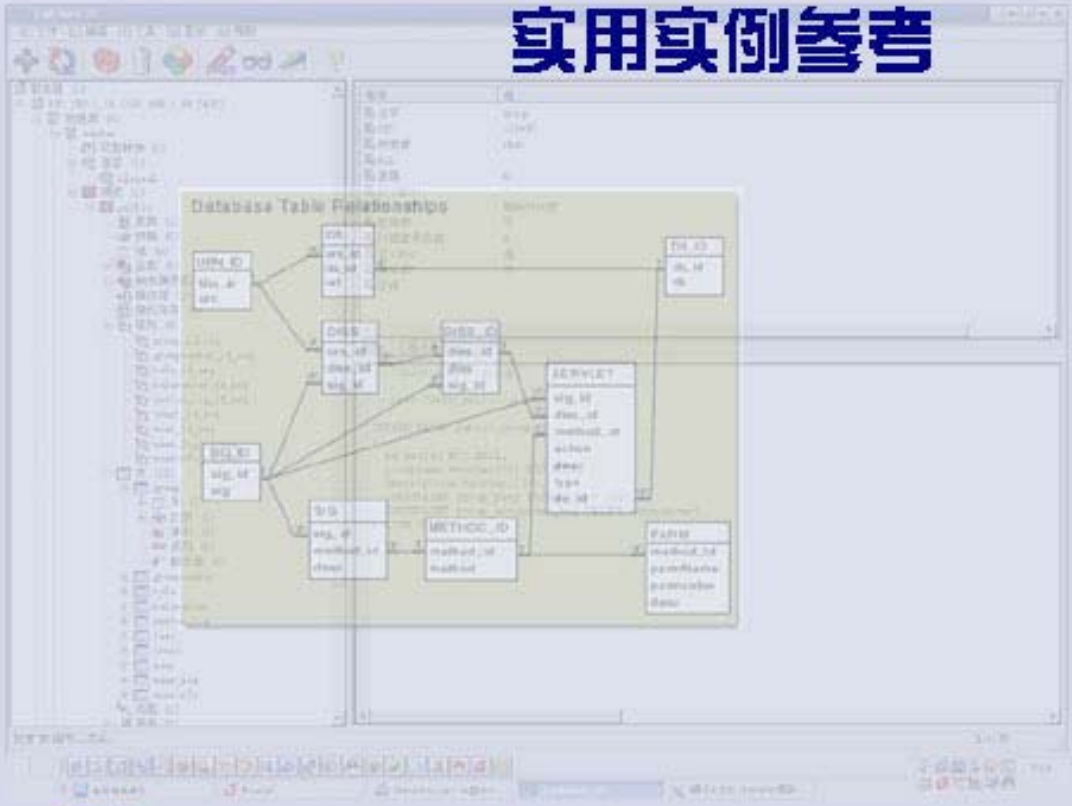
PostgreSQL

实用实例参考



PostgreSQL has won the 2003 Linux Journal Editors' Choice Award for the best DBMS!

深圳市福田区赖林一村

PostgreSQL 实用实例参考

陈景峰(netkiller)

前言

经过三个月的努力《PostgreSQL 实用实例参考》正式版终于推出了。因为最近换了工作，新公司的工作也很忙所以文档进展很慢，从最初几十页写到现在 200 页的文档，每天写文档的时间越来越少，有时一周也就只写 2 页，甚至一周一字未对。

正式版推出了，然后就是不断的修正。可能这段时间《PostgreSQL 实用实例参考》更新会更慢些。因为我还有其它文档要写：《OpenLDAP 文档》、《PHP + Corba + Python 开发分布式组件》、《JBuilder + Weblogic + PostgreSQL 开发 EJB》。。。。。

文档中所有例子，都是在工作总结出来的，如有错误请指正。本人爱写错别字（哈哈）如果你发现了有错字，请发邮件给我 netkiller@9812.net 修正文档，在这里我假设你对数据有一定认识。

目录

1 简介

我接触 PostgreSQL 是 2000 年，但项目中使用 PostgreSQL 是 2003 年，2000 当时应该是 5.x，6.x 版本我并没有深入地研究这个数据库，还是主要使用 MS Sql Server 7/2000 、Oracle 8。

因为很多企业难以支付 MS Sql Server 7/2000 、Oracle 8 这比费用，所以 Free Database 是最佳选择。但大多免费的数据库，功能有限、性能也差，跟本不能满足我们的需求。

1.1 关于性能

有一段时间里我们使用 MySQL，实现在不好用，功能太少，它只实现了 SQL92 中不到 30%的功能。除了 select、insert、update、delete 还有什么功能？一味强调速度快，真的是这样吗？MySQL 数据量增加很大时，速度下划很快。

几万条记录时速度最快，几十万记录时速度不同了，几百万时就开始慢了。PostgreSQL 随着数据量增大时，速度变化差距不象 MySQL 那么大。

有很多朋友在网上大批，特批（批评）触发器、游标，说影响性能。不过我很执著。这里要说明一下 PostgreSQL 的游标与其它数据不同。

1.2 PostgreSQL 对 SQL99 的支持

SQL-2/SQL92	√
SQL-3/SQL99	√
PRIMARY KEY 主键	√
FOREIGN KEY 外键	√
TOAST 大对象	√
View 视图	√
正则表达式	√
Subselect(*subquery*)子查询	√
TRIGGER 触发器	√
RULE 规则	√
FUNCTION 过程/函数	√
CURSOR 游标	√
PLSQL	√（PL/pgSQL,PL/Tcl,PL/Perl,PL/Python,plPHP）
表的锁定	√
事务隔离	√
事务处理	√
Object 对象支持	√（O-RDBMS）

其它：

连接池	进程方式
SSL	√

群集（HA，数据同步复制。。。）	√
ODBC	√
JDBC	√
下面是一些限制：	一行，一个表，一个库的最大尺寸是多少？
一个数据库最大尺寸？	无限制（存在 32TB 的数据库）
一个表的最大尺寸？	32TB
一行的最大尺寸？	1.6TB
一个字段的最大尺寸?	1GB
一个表里最大行数？	无限制
一个表里最大列数？	跟列类型有关,250-1600
一个表里的最大索引数量？	无限制

当然，实际上没有真正的无限制，还是要受可用磁盘空间、可用内存/交换区的制约。 表的最大尺寸 16 TB 不需要操作系统对大文件的支持。大表用多个 1 GB 的文件存储，因此文件系统尺寸的限制是不重要的。 如果缺省的块大小增长到 32K ，最大的表尺寸和最大列数可以增加。

这里引用 http://www.pgsqldb.org/postgres-faq.html4.5 详细请登录网站查看。

2 PostgreSQL 数据库

2.1 PostgreSQL 分区

PostgreSQL 最好自己单独一个分区。

卷	容量	单位
/boot	100	MB
/	1000	MB
/home	120000	MB
/usr	5000	MB
/var	5000	MB
/usr/local/mysql	10000	MB
/var/lib/pgsql	100000	MB
swap	2000	MB

总容量 240000 MB
使用容量 243100 MB

```
[chen@linux chen]$ df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda9              1004024     99892    853128  11% /
/dev/sda1               101089      9498     86372  10% /boot
/dev/sda2            120952116   7648124 107159936   7% /home
none                    515400         0    515400   0% /dev/shm
/dev/sda10             2522048     33260   2360672   2% /tmp
/dev/sda7              5036284   2238244   2542208  47% /usr
```

```
/dev/sda6              5036284   1919140   2861312  41% /var
/dev/sda5             40313964     99444  38166636   1% /var/lib/pgsql
/dev/sda3             60476068    212532  57191508   1% /cvsroot
[chen@linux chen]$

[chen@linux chen]$ df -m
Filesystem           1M-blocks      Used Available Use% Mounted on
/dev/sda9                  980        98       833  11% /
/dev/sda1                   99        10        84  10% /boot
/dev/sda2                  118117     7469    104648   7% /home
none                       503         0       503   0% /dev/shm
/dev/sda10                2463        33      2305   2% /tmp
/dev/sda7                 4918      2186      2482  47% /usr
/dev/sda6                 4918      1875      2794  41% /var
/dev/sda5                39369        98     37272   1% /var/lib/pgsql
/dev/sda3                59059       208     55851   1% /cvsroot
[chen@linux chen]$
```

2.2 RPM 包安装

```
[root@linux software]# ls -1
postgresql-7.3.4-1PGDG.i386.rpm
postgresql-contrib-7.3.4-1PGDG.i386.rpm
postgresql-debuginfo-7.3.4-1PGDG.i386.rpm
postgresql-devel-7.3.4-1PGDG.i386.rpm
postgresql-docs-7.3.4-1PGDG.i386.rpm
postgresql-jdbc-7.3.4-1PGDG.i386.rpm
postgresql-libs-7.3.4-1PGDG.i386.rpm
postgresql-pl-7.3.4-1PGDG.i386.rpm
postgresql-python-7.3.4-1PGDG.i386.rpm
postgresql-server-7.3.4-1PGDG.i386.rpm
postgresql-tcl-7.3.4-1PGDG.i386.rpm
postgresql-test-7.3.4-1PGDG.i386.rpm
[root@linux software]# rpm -Uvh --nodeps `ls -1`
Preparing...                ########################################### [100%]
   1:postgresql-test        ########################################### [  8%]
   2:postgresql             ########################################### [ 17%]
   3:postgresql-contrib     ########################################### [ 25%]
   4:postgresql-debuginfo   ########################################### [ 33%]
   5:postgresql-devel       ########################################### [ 42%]
   6:postgresql-docs        ########################################### [ 50%]
   7:postgresql-jdbc        ########################################### [ 58%]
   8:postgresql-libs        ########################################### [ 67%]
   9:postgresql-pl          ########################################### [ 75%]
```

```
   10:postgresql-python     ########################################### [ 83%]
   11:postgresql-server     ########################################### [ 92%]
   12:postgresql-tcl        ########################################### [100%]
[root@linux software]# rpm -qa|grep postgre
postgresql-devel-7.3.4-1PGDG
postgresql-7.3.4-1PGDG
postgresql-python-7.3.4-1PGDG
postgresql-contrib-7.3.4-1PGDG
postgresql-jdbc-7.3.4-1PGDG
postgresql-server-7.3.4-1PGDG
postgresql-debuginfo-7.3.4-1PGDG
postgresql-libs-7.3.4-1PGDG
postgresql-tcl-7.3.4-1PGDG
postgresql-test-7.3.4-1PGDG
postgresql-pl-7.3.4-1PGDG
postgresql-docs-7.3.4-1PGDG
[root@linux software]#
[root@linux software]# service postgresql start
Starting postgresql service:                               [  OK  ]
[root@linux software]# su postgres
bash-2.05b$ createdb
CREATE DATABASE
bash-2.05b$ psql
Welcome to psql 7.3.4, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

postgres=# \q
bash-2.05b$
bash-2.05b$ vi /var/lib/pgsql/data/postgresql.conf
#==========================================================================



#
#           Connection Parameters
#
#tcpip_socket = false
tcpip_socket = true
#ssl = false
```

```
#max_connections = 32
max_connections = 128
#superuser_reserved_connections = 2

#port = 5432
#hostname_lookup = false
#show_source_port = false

#unix_socket_directory = ''
#unix_socket_group = ''
#unix_socket_permissions = 0777 # octal

#virtual_host = ''

#krb_server_keyfile = ''



#
#            Shared Memory Size
#
#shared_buffers = 64                    # min max_connections*2 or 16, 8KB each
shared_buffers = 256                    # min max_connections*2 or 16, 8KB each
#max_fsm_relations = 1000               # min 10, fsm is free space map, ~40 bytes
#max_fsm_pages = 10000                  # min 1000, fsm is free space map, ~6 bytes
#max_locks_per_transaction = 64 # min 10
#wal_buffers = 8                        # min 4, typically 8KB each

bash-2.05b$ vi /var/lib/pgsql/data/pg_hba.conf
host     all         all               127.0.0.1               255.255.255.255      md5

bash-2.05b$ psql
Welcome to psql 7.3.4, the PostgreSQL interactive terminal.

Type:   \copyright for distribution terms
        \h for help with SQL commands
        \? for help on internal slash commands
        \g or terminate with semicolon to execute query
        \q to quit

postgres=# CREATE USER netkiller WITH PASSWORD 'chen';
CREATE USER
postgres=# CREATE DATABASE netkiller WITH OWNER = netkiller TEMPLATE = template0
ENCODING = 'UNICODE';
CREATE DATABASE
```

```
postgres=# \du
                    List of database users
  User name | User ID |                Attributes
-----------+---------+---------------------------
  netkiller |         100 |
  postgres    |             1 | superuser, create database
(2 rows)

postgres=# \l
               List of databases
     Name        |      Owner      | Encoding
-----------+-----------+-----------
  netkiller | netkiller | UNICODE
  postgres    | postgres    | SQL_ASCII
  template0 | postgres    | SQL_ASCII
  template1 | postgres    | SQL_ASCII
(4 rows)

postgres=# \q
bash-2.05b$
bash-2.05b$ createlang plpgsql netkiller
bash-2.05b$

bash-2.05b$ exit
exit
[root@linux software]#    service postgresql restart
                                                                    [    OK    ]
Starting postgresql service:                                    [    OK    ]
[root@linux software]#
[root@linux software]# psql -h127.0.0.1 -Unetkiller netkiller
Password:
Welcome to psql 7.3.4, the PostgreSQL interactive terminal.

Type:    \copyright for distribution terms
         \h for help with SQL commands
         \? for help on internal slash commands
         \g or terminate with semicolon to execute query
         \q to quit

netkiller=>
```

注意：

1. 程序安装我使用了一个小技巧。（我懒哈哈）rpm -Uvh --nodeps `ls -1`
   安装一定要加—nodeps，ls -1 这里是减号啊拉伯数字 1，不是字母“l” (L)

2. postgres 只能用于 UNIX Domain Socket 方式登陆(/tmp/.s.PGSQL.5432)，不能在 TCP/IP Socket 模式下登陆。

```
[root@linux software]# ls -la /tmp
total 68
drwxrwxrwt    11 root     root         4096 Nov 11 16:29 .
drwxr-xr-x    22 root     root         4096 Nov  5 14:49 ..
srwx------     1 root     nobody          0 Nov  5 11:34 .fam_socket
drwxrwxrwt     2 xfs      xfs          4096 Nov  5 14:49 .font-unix
drwx------     2 root     root         4096 Nov  5 19:06 .gconfd
srw-rw-rw-     1 root     root            0 Nov  5 14:49 .gdm_socket
drwxrwxrwx     2 bin      bin          4096 Nov  5 14:49 .iroha_unix
drwx------     2 root     root         4096 Nov  5 19:14 kde-root
drwx------     2 root     root        16384 Nov  5 18:46 lost+found
drwxr-xr-x     2 root     root         4096 Nov  5 18:55 .mozilla
drwx------     2 root     root         4096 Nov  5 11:38 orbit-root
drwxr-xr-x     2 root     root         4096 Nov  5 19:14 .qt
-rw-------     1 root     root         1024 Nov  5 18:52 .rnd
srwxrwxrwx     1 postgres postgres        0 Nov 11 16:29 .s.PGSQL.5432
-rw-------     1 postgres postgres       26 Nov 11 16:29 .s.PGSQL.5432.lock
-r--r--r--     1 root     root           11 Nov  5 14:49 .X0-lock
drwxrwxrwt     2 root     root         4096 Nov  5 14:49 .X11-unix
[root@linux software]# file /tmp/.s.PGSQL.5432
/tmp/.s.PGSQL.5432: socket
```

使用 file 命令可以查看文件类型，所以/tmp/.s.PGSQL.5432 显示类型为/tmp/.s.PGSQL.5432: socket

[root@linux software]# psql -h127.0.0.1 –Upostgres db 会提示

Password:

psql: FATAL: Password authentication failed for user "postgres"

```
[root@linux software]# psql -h127.0.0.1 -Upostgres netkiller
Password:
Password:
Password:
Password:
Password:
Password:
Password:
psql: FATAL:  Password authentication failed for user "postgres"
```

解决方法是创建一个用户。

3. 登陆提示

[root@linux software]# psql -h127.0.0.1 -Unetkiller netkiller

psql: FATAL: No pg_hba.conf entry for host 127.0.0.1, user netkiller, database netkiller

编辑/var/lib/pgsql/data/pg_hba.conf 文件加入

host all all 127.0.0.1 255.255.255.255 md5

2.3 数据库备份方案

2.3.1 备份数据库脚本

脚本功能是，首先备份数据库、然后打包、压缩为 tar.gz、最后上传到指定位置并删除临时文件。

```
[root@linux root]# cat backup.sh
#!/bin/bash

FTPHOST=ftp.9812.net
USER=netkiller
PASSWD=xxx

echo "Starting Backup PostgreSQL ... "
#big5 gb2312 gb18030 …
export PGCLIENTENCODING=gb18030
su - postgres -c pg_dumpall > pgsql-backup.`date +%Y-%m-%d.%H:%M:%S`.dmp
tar zcvf pgsql-backup.`date +%Y-%m-%d`.tar.gz *.dmp

echo "Upload File ..."
ftp -n ${FTPHOST} <<!
user ${USER} ${PASSWD}
binary
prompt
mkdir backup
cd backup
mput *.tar.gz
close
bye
!
echo "Remove temp file ..."
rm -rf pgsql-backup.*.dmp
rm -rf pgsql-backup.????-??-??.tar.gz
[root@linux root]#
```

如果您没有一台专用于备份数据的机器（有静态 IP 的机器）。上面的备份脚本可更改为：

```
[root@linux root]# cat backup.sh
#!/bin/bash

echo "Starting Backup PostgreSQL ... "
su - postgres -c pg_dumpall > pgsql-backup.`date +%Y-%m-%d.%H:%M:%S`.dmp
tar zcvf pgsql-backup.`date +%Y-%m-%d`.tar.gz *.dmp
```

```
echo "Remove temp file ..."
rm -rf pgsql-backup.*.dmp
[root@linux root]#
```

2.3.2 下载备份脚本

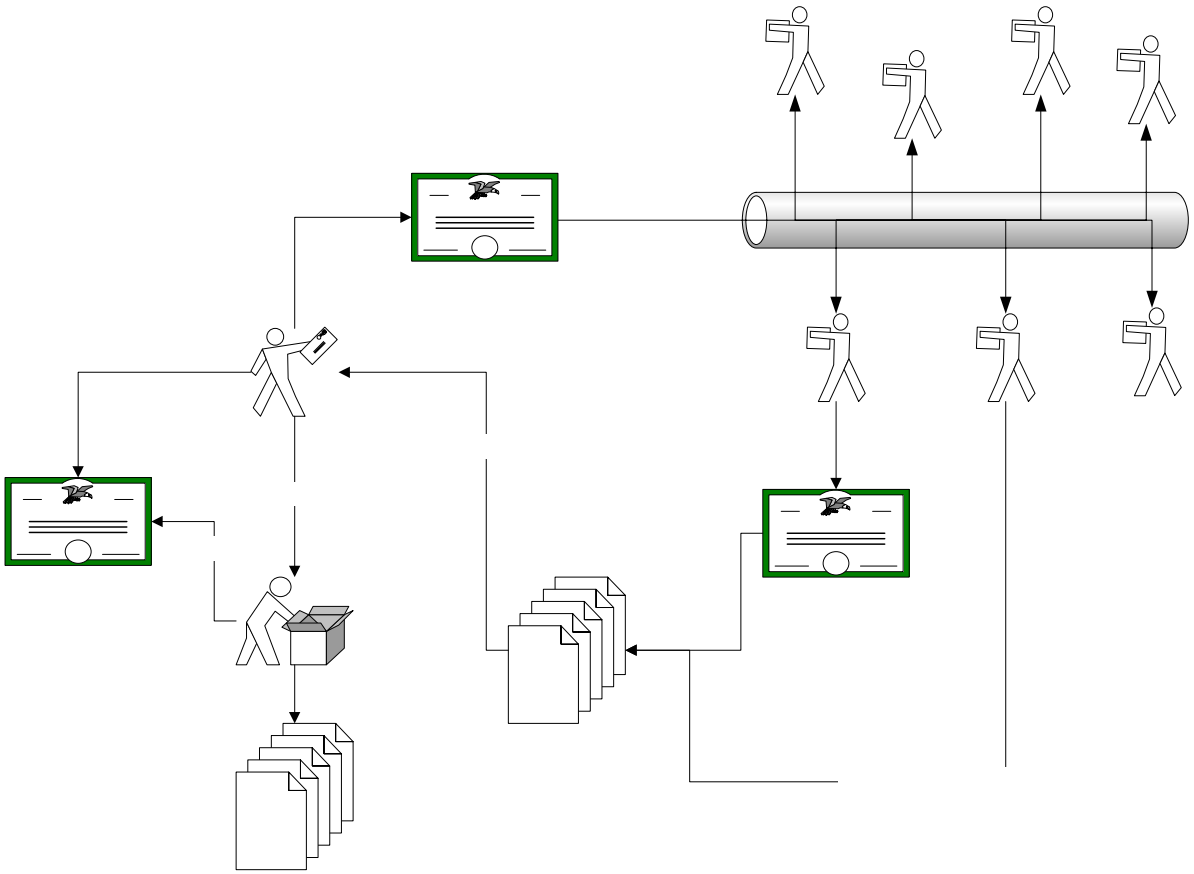
```
[root@linux root]# cat getbackup.sh
#!/bin/bash

FTPHOST=ftp.9812.net
USER=netkiller
PASSWD=xxx

wget ftp://${USER}:${PASSWD}@${FTPHOST}/backup/*

ftp -n ${FTPHOST} <<!
user ${USER} ${PASSWD}
binary
prompt
cd backup
mdelete *
close
bye
!
[root@linux root]#
```

2.3.3 保证备份数据的安全-PGP/GPG 加密

数据库中的内容有些是不能提供给用户的，如其它用户的资料，密码。在数据库中的数据，你可以通过权限来限制用户操作。将数据库备份（导出）到本地 SQL 文本文件中(xxxx.sql 包括 DDL，DML) ，一但备份落入他手，后果不可设想，他很容易得用你的数据，因为你备份的数据是文本文件，没有任何加密措施。

这里介绍 GnuPG 以下简称 GPG，GPG 与 PGP 兼容。由于 PGP 使用了许多专利算法，属于美国加密出口限制之列。而 GnuPG 是 GPL 软件。

GPG 使用非对称加密算法，安全程度很高。所谓非对称加密算法，就是每一个用户都拥有一对密钥：公钥和私钥。其中，私钥由用户保存，公钥提供给 internet 上的用户。



设：

陈景峰的帐号：chen

小明的帐号：ming

以下为 chen 帐号的操作：

1. 查看当前文件夹

```
[chen@linux chen]$ ls -la
total 56
drwx------    4 chen     chen         4096 Dec 12 20:38 .
drwxr-xr-x    7 root     root         4096 Nov 12 11:47 ..
-rw-------    1 chen     chen         4953 Dec 10 14:05 .bash_history
-rw-r--r--    1 chen     chen           24 Feb 11  2003 .bash_logout
-rw-r--r--    1 chen     chen          191 Feb 11  2003 .bash_profile
-rw-r--r--    1 chen     chen          124 Feb 11  2003 .bashrc
-rw-r--r--    1 chen     chen         5531 Feb  4  2003 .canna
-rw-r--r--    1 chen     chen          847 Feb 20  2003 .emacs
-rw-r--r--    1 chen     chen          120 Feb 27  2003 .gtkrc
drwxr-xr-x    3 chen     chen         4096 Aug 12  2002 .kde
-rw-------    1 chen     chen          594 Dec 10 09:38 .viminfo
drwxr-xr-x    2 chen     chen         4096 Nov  5 19:16 .xemacs
[chen@linux chen]$
```

2. 生成密钥（公钥、私钥）

使用 GPG 之前必须生成密钥（公钥、私钥）操作步骤。

gpg --gen-key

```
[chen@linux chen]$ gpg --gen-key
gpg (GnuPG) 1.2.1; Copyright (C) 2002 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information
gpg: /home/chen/.gnupg: directory created
gpg: new configuration file `/home/chen/.gnupg/gpg.conf' created
gpg: keyblock resource `/home/chen/.gnupg/secring.gpg': file open error
gpg: keyring `/home/chen/.gnupg/pubring.gpg' created
Please select what kind of key you want:
   (1) DSA and ElGamal (default)
   (2) DSA (sign only)
   (5) RSA (sign only)
Your selection? 回车
DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair.
                minimum keysize is  768 bits
                default keysize is 1024 bits
     highest suggested keysize is 2048 bits
What keysize do you want? (1024) 回车
Requested keysize is 1024 bits
Please specify how long the key should be valid.
         0 = key does not expire
       <n>  = key expires in n days
       <n>w = key expires in n weeks
       <n>m = key expires in n months
       <n>y = key expires in n years
Key is valid for? (0) 回车
Key does not expire at all
Is this correct (y/n)? y

You need a User-ID to identify your key; the software constructs the user id
from Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: netkiller
Email address: netkiller@9812.net
```

```
Comment: 陈景峰的密钥    （注：输入中文终端要支持 UTF-8）
You are using the `utf-8' character set.
You selected this USER-ID:
    "netkiller (陈景峰的密钥) <netkiller@9812.net>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?o

Enter passphrase:输入密钥口令
Repeat passphrase:输入密钥口令
You need a Passphrase to protect your secret key.

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
++++++++++++++++.++++++++++++++++.++++++++++++++++.++++++++++++++++.++++++++++++++
+.+++++++++++++++++++++++++++++++++++++++++++++++.++++++++++++++++++++.....>+++++.....
....+++++

Not enough random bytes available.    Please do some other work to give
the OS a chance to collect more entropy! (Need 290 more bytes)
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
..+++++.+++++.++++++++++.++++++++++++++++.++++++++++++++++++++++++++..++++++++++++++
++.++++++++++++++++++++.+++++.++++++++++++++++.+++++...++++++++++>+++++.....................
.....................................>+++++.............>.+++++.....<+++++.............+++++^^^
gpg: /home/chen/.gnupg/trustdb.gpg: trustdb created
public and secret key created and signed.
key marked as ultimately trusted.

pub    1024D/B00847C5 2003-12-12 netkiller (陈景峰的密钥) <netkiller@9812.net>
       Key fingerprint = 0058 5847 7598 556F AAFD    81A5 AC07 C873 B008 47C5
sub    1024g/0B70F0CB 2003-12-12

[chen@linux chen]$
```

3. 查看生成密钥

```
[chen@linux chen]$ ls -la
total 52
drwx------    5 chen     chen         4096 Dec 12 20:47 .
drwxr-xr-x    7 root     root         4096 Dec 12 20:44 ..
-rw-r--r--    1 chen     chen           24 Dec 12 20:44 .bash_logout
```

```
-rw-r--r--    1 chen     chen          191 Dec 12 20:44 .bash_profile
-rw-r--r--    1 chen     chen          124 Dec 12 20:44 .bashrc
-rw-r--r--    1 chen     chen         5531 Dec 12 20:44 .canna
-rw-r--r--    1 chen     chen          847 Dec 12 20:44 .emacs
drwx------    2 chen     chen         4096 Dec 12 20:52 .gnupg
-rw-r--r--    1 chen     chen          120 Dec 12 20:44 .gtkrc
drwxr-xr-x    3 chen     chen         4096 Dec 12 20:44 .kde
-rw-------    1 chen     chen           61 Dec 12 20:45 .Xauthority
drwxr-xr-x    2 chen     chen         4096 Dec 12 20:44 .xemacs
[chen@linux chen]$ ls .gnupg/
gpg.conf  pubring.gpg  pubring.gpg~  random_seed  secring.gpg  trustdb.gpg
```

4. 证书的回收

当您的密钥（gpg --gen-key）生成之后，建议您立即做一个公钥回收证书，如果您忘记了您的私钥口令或者您的私钥丢失或者被盗，您可以发布这个证书来声明以前的公钥不再有效。

gpg --output revoke.asc --gen-revoke netkiller （netkiller 你在生成密钥时输入的 Real name:）

gpg --output revoke.asc --gen-revoke netkiller@9812.net （使用邮件地址也可以）

```
[chen@linux chen]$ gpg --output revoke.asc --gen-revoke netkiller
gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information

sec  1024D/B00847C5 2003-12-12      netkiller (陈景峰的密钥) <netkiller@9812.net>

Create a revocation certificate for this key? y
Please select the reason for the revocation:
  0 = No reason specified
  1 = Key has been compromised
  2 = Key is superseded
  3 = Key is no longer used
  Q = Cancel
(Probably you want to select 1 here)
Your decision?
Enter an optional description; end it with an empty line:
> :( cancel
>
Reason for revocation: Key has been compromised
:( cancel
Is this okay?
Please select the reason for the revocation:
  0 = No reason specified
  1 = Key has been compromised
  2 = Key is superseded
  3 = Key is no longer used
```

```
   Q = Cancel
(Probably you want to select 1 here)
Enter an optional description; end it with an empty line:
> :( cancel
>
Is this okay? y

You need a passphrase to unlock the secret key for
user: "netkiller (陈景峰的密钥) <netkiller@9812.net>"
1024-bit DSA key, ID B00847C5, created 2003-12-12

ASCII armored output forced.
Revocation certificate created.

Please move it to a medium which you can hide away; if Mallory gets
access to this certificate he can use it to make your key unusable.
It is smart to print this certificate and store it away, just in case
your media become unreadable.    But have some caution:    The print system of
your machine might store the data and make it available to others!
[chen@linux chen]$ ls
revoke.asc
[chen@linux chen]$ cat revoke.asc
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.2.1 (GNU/Linux)
Comment: A revocation certificate should follow

iFIEIBECABIFAj/Zv08LHQI6KCBjYW5jZWwACgkQrAfIc7AIR8X3agCcDBjqRkFx
QUzcZ/1Gyf1/jjFis04An2rYQz2XrCode08Y78Fj63RVNKD9
=ovDh
-----END PGP PUBLIC KEY BLOCK-----
[chen@linux chen]$
```

5. 密钥列表

   gpg --list-key

```
[chen@linux chen]$ gpg --list-key
gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information
/home/chen/.gnupg/pubring.gpg
------------------------------
pub    1024D/B00847C5 2003-12-12 netkiller (陈景峰的密钥) <netkiller@9812.net>
sub    1024g/0B70F0CB 2003-12-12

[chen@linux chen]$
```

6. 输出公钥

以 ASCII 字符格式输出公钥：gpg --output netkiller.gpg --armor --export netkiller

以二进制格式输出公钥：gpg --output netkiller.gpg --export netkiller

下面是以 ASCII 字符格式输出（其实就是做了一下 BASE64 编码）：

```
[chen@linux chen]$ gpg --output netkiller.gpg --armor --export netkiller
gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information
[chen@linux chen]$ ls
netkiller.gpg    revoke.asc
[chen@linux chen]$ cat netkiller.gpg
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.2.1 (GNU/Linux)

mQGiBD/ZuhgRBACBRuWYRtJ8+8VmnYUgNy7TS/nVl0sHrsGD2kgIWVUuZYgKSUoM
vT4MUHWdd52yesovAV61qsVCfUz+O76ovhQrUzv4jp+bkIOKcc7E07Z2MZmc1BqR
+Gavb3gsJM6DmOLcRiU0m3fqod1KCGFf8K6ZLQUhRJYWDI80KEgJqliG4wCgo2xn
5WS1CIGnvGDFUiGY6VhdamsD/jdiqSIcwFt2x6VMjzeWkHHM5wNYHuBJnp9DPd9g
rn3uEq+tSex8ZXRyzHGj+N4SKezhEYal1D762kDxjGYltk5Xce5dXQBn9fulEDhD
OzOp78GvIvJ/m33D/J6xECbXUz8XsFFhxJ6QnVh/RURY+EvHE1Tmz/fRG69Rc1Uc
JBqCA/0faHEkyDv+FWEsmFKjflDNqN5NHtdWzJZQZKD1Vb64oJ5CK6r2l+vmxbBr
fVpfk5OVXnfMSpLKc7aGA9X+mUMuNrGRNzzzsmVK6urWQovL/BfeukMgDBZXkLd8
fO7aA53XeBhmVC49atFPH8hsOeMdd0mombrzcvKczjMp0ThP9rQzbmV0a2lsbGVy
ICjpmYjmma/ls7DnmoTlr4bpkqUpIDxuZXRraWxsZXJAOTgxMi5uZXQ+iFkEExEC
ABkFAj/ZuhgECwcDAgMVAgMDFgIBAh4BAheAAAoJEKwHyHOwCEfFBqMAn0HoK9Xc
zvzVkFODVZPWUskzwAhqAJ4rbgYEjSN1/CrdUBzTMtecGu9P+7kBDQQ/2boaEAQA
zhoIDY866/GWUUpuarpVKcN1ijn+5M1Pr42vm2Z42ns4PZW3cagHJeIOuJ5R2Aw1
6V4zZwP5PcBScYxQpM0m0bVmTGp/suZmZ6/u3+ADgvJYSxAXdpzP0cL9rVRKqaPa
MKh+HOanAJ9tWcSy6KW83JKG2NS/0U6OSGGDSoNLElMAAwUD/iGBjPfXD5jsepg+
Z9J1RefM5/R1nnBEeOROnWyaczIU1okswlyluAthi+2+ijpEULaqSQ+ZjtuBjcMp
kE5UKKql6yBAk2CqJMVkVLlDbPFqbidkAqGp5riKWKc487jR6iZjIAhHvXL0xPIQ
erBmEpi4UT7RlaCAmYwvZ1nxGP3eiEYEGBECAAYFAj/ZuhoACgkQrAfIc7AIR8U0
xACfT5pZ+0YjSp9z0/9jPwDfhw7J1bcAnjqxP+uKfkuDHnXRyYFErTN+7iHE
=CII0
-----END PGP PUBLIC KEY BLOCK-----
[chen@linux chen]$
```

二进制格式输出：

```
[chen@linux chen]$ gpg --output bin-netkiller.gpg --export netkiller
gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information
[chen@linux chen]$ ls
bin-netkiller.gpg    netkiller.gpg    revoke.asc
[chen@linux chen]$
```

7. 使用 file 命令识别文件

```
[chen@linux chen]$ ls
bin-netkiller.gpg    netkiller.gpg    revoke.asc
[chen@linux chen]$ file bin-netkiller.gpg
bin-netkiller.gpg: data
[chen@linux chen]$ file netkiller.gpg
netkiller.gpg: PGP armored data public key block
[chen@linux chen]$ file revoke.asc
revoke.asc: PGP armored data public key block
[chen@linux chen]$
```

8. 发布公钥

你可以将你的公钥放在主页上下载，也可以 mail 给别人。

```
[chen@linux chen]$ pine
  PINE 4.44      MAIN MENU                                Folder: INBOX    No Messages


                ?       HELP                    -  Get help using Pine

                C       COMPOSE MESSAGE         -  Compose and send a message

                I       MESSAGE INDEX           -  View messages in current folder

                L       FOLDER LIST             -  Select a folder to view

                A       ADDRESS BOOK            -  Update address book

                S       SETUP                   -  Configure Pine Options

                Q       QUIT                    -  Leave the Pine program




     Copyright 1989-2002.  PINE is a trademark of the University of Washington.
                        [Folder "INBOX" opened with 0 messages]
? Help                          P PrevCmd                       R RelNotes
O OTHER CMDS > [ListFldrs] N NextCmd                            K KBLock
```

```
  PINE 4.44      COMPOSE MESSAGE                          Folder: INBOX    No Messages

To          : netkiller@9812.net
Cc          :
Attchmnt:
Subject :  这是我的证书
```

```
----- Message Text -----

Attchmnt




^G Get Help    ^X Send          ^R Rich Hdr    ^Y PrvPg/Top ^K Cut Line    ^O Postpone
^C Cancel      ^D Del Char    ^J Attach       ^V NxtPg/End ^U UnDel Line^T To Files
```

光标至于 Attchmnt:上按^J -> 再按 ^T

```
File to attach:
^G Get Help    ^T    To Files
^C Cancel      TAB Complete
```

```
    PINE 4.44                    BROWSER       Dir: /home/chen

..                 (parent dir)   .gnupg                  (dir)
.kde                       (dir)   mail                     (dir)
.xemacs                    (dir)   .addressbook              0   B
.addressbook.lu          2.3 KB    .bash_logout            24   B
.bash_profile             191   B    .bashrc                124   B
bin-netkiller.gpg         909   B    .canna                5.5 KB
.emacs                    847   B    .gtkrc                 120   B
netkiller.gpg             1.3 KB    .pinerc               14 KB
revoke.asc                275   B    .Xauthority            61   B









                              [ Searched to end of directory ]
? Get Help     E Exit Brwsr                -     Prev Pg    D Delete        C Copy
                 S [Select]     W Where is     Spc Next Pg    R Rename        A Add
```

选择 netkiller.gpg 回车

```
Attachment comment: my netkiller.gpg file
^G Get Help
^C Cancel
```

输入注释信息

```
  PINE 4.44      COMPOSE MESSAGE                          Folder: INBOX    No Messages

To          : netkiller@9812.net
Cc          :
Attchmnt: 1. /home/chen/netkiller.gpg (1.3 KB) "my netkiller.gpg file"
Subject : my netkiller.gpg file
----- Message Text -----

http://linux.9812.net
email:netkiller@9812.net




                [File /home/chen/netkiller.gpg attached as type TEXT/PLAIN]
^G Get Help    ^X Send          ^R Rich Hdr    ^Y PrvPg/Top ^K Cut Line    ^O Postpone
^C Cancel      ^D Del Char    ^J Attach        ^V NxtPg/End ^U UnDel Line^T To Files
```

```
Send message?y
 ? Help          Y [Yes]
^C Cancel        N No
```

选择 y 回车

```
  PINE 4.44      MAIN MENU                                Folder: INBOX    No Messages



  PINE 4.44      COMPOSE MESSAGE                          Folder: INBOX    No Messages

To          : netkiller@9812.net
Cc          :
Attchmnt: 1. /home/chen/netkiller.gpg (1.3 KB) "my netkiller.gpg file"
Subject :  这是我的证书
----- Message Text -----

Attchmnt









                              [Sending mail |          0%      |]
```

发送成功

PINE 4.44 MAIN MENU Folder: INBOX No Messages

? HELP - Get help using Pine

C COMPOSE MESSAGE - Compose and send a message

I MESSAGE INDEX - View messages in current folder

L FOLDER LIST - Select a folder to view

A ADDRESS BOOK - Update address book

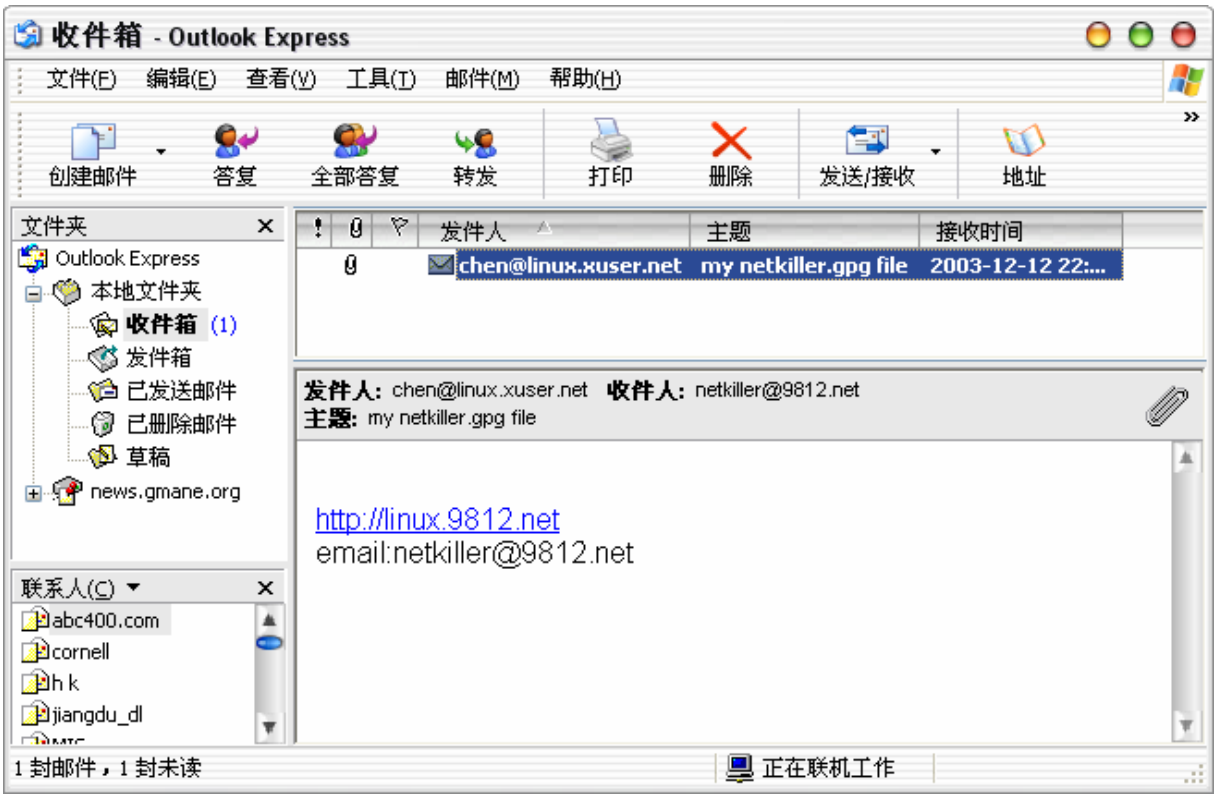
S SETUP - Configure Pine Options

Q QUIT - Leave the Pine program

[Message sent and copied to "sent-mail".]

? Help P PrevCmd R RelNotes

O OTHER CMDS > [ListFldrs] N NextCmd K KBLock



9. 将公钥给其它用户

```
[chen@linux chen]$ cp netkiller.gpg /tmp
```

以下是 ming 帐号的操作：

10. 获得公钥

```
[ming@linux ming]$ cp /tmp/netkiller.gpg .
[ming@linux ming]$ ls
netkiller.gpg
[ming@linux ming]$
```

11. 导入公钥

```
[ming@linux ming]$ gpg --import netkiller.gpg
gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information
gpg: /home/ming/.gnupg: directory created
gpg: new configuration file `/home/ming/.gnupg/gpg.conf' created
gpg: keyblock resource `/home/ming/.gnupg/secring.gpg': file open error
gpg: keyring `/home/ming/.gnupg/pubring.gpg' created
gpg: /home/ming/.gnupg/trustdb.gpg: trustdb created
gpg: key B00847C5: public key "netkiller (▒\x9\x8▒\x9▒峰▒\x9\x8▒\x8▒\x9▒) <netkiller@9812.net>"
imported
gpg: Total number processed: 1
gpg:                imported: 1
[ming@linux ming]$ gpg --list-key
gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information
/home/ming/.gnupg/pubring.gpg
------------------------------
pub   1024D/B00847C5 2003-12-12 netkiller (陈景峰的密钥) <netkiller@9812.net>
sub   1024g/0B70F0CB 2003-12-12

[ming@linux ming]$
```

12. 确认密钥

导入密钥以后，使用数字答名来验证此证书是否合法。

```
[ming@linux ming]$ gpg --fingerprint netkiller
gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information
pub   1024D/B00847C5 2003-12-12 netkiller (陈景峰的密钥) <netkiller@9812.net>
      Key fingerprint = 0058 5847 7598 556F AAFD  81A5 AC07 C873 B008 47C5
sub   1024g/0B70F0CB 2003-12-12
```

```
[ming@linux ming]$
```

13. 密钥签名

导入密钥之后，可以使用(gpg -—sign-key netkiller) 进行签名，签名的主要目的是证明您完全信任这个证书的合法性。

```
[ming@linux ming]$ gpg --gen-key
gpg (GnuPG) 1.2.1; Copyright (C) 2002 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information
Please select what kind of key you want:
   (1) DSA and ElGamal (default)
   (2) DSA (sign only)
   (5) RSA (sign only)
Your selection?
DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair.
                minimum keysize is  768 bits
                default keysize is 1024 bits
    highest suggested keysize is 2048 bits
What keysize do you want? (1024)
Requested keysize is 1024 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct (y/n)? y

You need a User-ID to identify your key; the software constructs the user id
from Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: ming
Name must be at least 5 characters long
Real name: mings
Email address: mings@9812.net
```

```
Comment: I am ming
You selected this USER-ID:
    "mings (I am ming) <mings@9812.net>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
You need a Passphrase to protect your secret key.
Enter passphrase:

passphrase not correctly repeated; try again.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
++++++++++++++.++++++++++++++++++++++++++++++.++++++++++....++++++++++.++++++++
++++++++++++.+++++++++++++++..+++++++++++++++++++++++++++.++++++++++.....>..+++++.....
.....+++++
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
++++++++++++++++++++++++++..+++++++++++++++++++++.++++++++++++++++++++++++++++++++
+++++..+++++++++++++++...++++++++++++++++++++++++++++++++++++>+++++.....................+++++
^^^^^^^^^^^^^^^^
public and secret key created and signed.
key marked as ultimately trusted.

pub    1024D/3D9CE6DF 2003-12-12 mings (I am ming) <mings@9812.net>
       Key fingerprint = 51C5 A223 98B8 A65F 4BF4    B610 4B80 D812 3D9C E6DF
sub    1024g/510C2A18 2003-12-12
[ming@linux ming]$ gpg --sign-key netkiller
gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information

pub    1024D/B00847C5    created: 2003-12-12 expires: never          trust: -/-
sub    1024g/0B70F0CB    created: 2003-12-12 expires: never
(1). netkiller (陈景峰的密钥) <netkiller@9812.net>



pub    1024D/B00847C5    created: 2003-12-12 expires: never          trust: -/-
 Primary key fingerprint: 0058 5847 7598 556F AAFD    81A5 AC07 C873 B008 47C5

       netkiller (陈景峰的密钥) <netkiller@9812.net>

How carefully have you verified the key you are about to sign actually belongs
```

```
to the person named above?  If you don't know what to answer, enter "0".

    (0) I will not answer. (default)
    (1) I have not checked at all.
    (2) I have done casual checking.
    (3) I have done very careful checking.

Your selection? 3
Are you really sure that you want to sign this key
with your key: "mings (I am ming) <mings@9812.net>"

I have checked this key very carefully.

Really sign? y

You need a passphrase to unlock the secret key for
user: "mings (I am ming) <mings@9812.net>"
1024-bit DSA key, ID 3D9CE6DF, created 2003-12-12

Enter passphrase: （注这里输入 mings 的口令）

[ming@linux ming]$
[ming@linux ming]$ gpg --list-key
gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information
/home/ming/.gnupg/pubring.gpg
------------------------------
pub   1024D/B00847C5 2003-12-12 netkiller (陈景峰的密钥) <netkiller@9812.net>
sub   1024g/0B70F0CB 2003-12-12

pub   1024D/3D9CE6DF 2003-12-12 mings (I am ming) <mings@9812.net>
sub   1024g/510C2A18 2003-12-12

[ming@linux ming]$
```

14. 检查签名

```
[chen@linux chen]$ gpg --check-sigs netkiller
gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information
pub   1024D/B00847C5 2003-12-12 netkiller (陈景峰的密钥) <netkiller@9812.net>
sig!3         B00847C5 2003-12-12      netkiller (陈景峰的密钥) <netkiller@9812.net>
sub   1024g/0B70F0CB 2003-12-12
sig!          B00847C5 2003-12-12      netkiller (陈景峰的密钥) <netkiller@9812.net>
[ming@linux ming]$ gpg --check-sigs netkiller
```

```
gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information
pub    1024D/B00847C5 2003-12-12 netkiller (陈景峰的密钥) <netkiller@9812.net>
sig!3          B00847C5 2003-12-12      netkiller (陈景峰的密钥) <netkiller@9812.net>
sig!           79F1102B 2003-12-12      mings (I am mings) <mings@9812.net>
sub    1024g/0B70F0CB 2003-12-12
sig!           B00847C5 2003-12-12      netkiller (陈景峰的密钥) <netkiller@9812.net>

[ming@linux ming]$
[chen@linux chen]$

[ming@linux ming]$ gpg --check-sigs netkiller
gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information
pub    1024D/B00847C5 2003-12-12 netkiller (陈景峰的密钥) <netkiller@9812.net>
sig!3          B00847C5 2003-12-12      netkiller (陈景峰的密钥) <netkiller@9812.net>
sub    1024g/0B70F0CB 2003-12-12
sig!           B00847C5 2003-12-12      netkiller (陈景峰的密钥) <netkiller@9812.net>

[ming@linux ming]$
```

15. 加密和解密

加密：

```
[ming@linux ming]$ pg_dump -Unetkiller -h127.0.0.1 >pgsql-dump.sql
Password:
[ming@linux ming]$
```

加签名：

```
[ming@linux ming]$ gpg -sear netkiller pgsql-dump.sql
gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information

You need a passphrase to unlock the secret key for
user: "mings (I am mings) <mings@9812.net>"
1024-bit DSA key, ID 79F1102B, created 2003-12-12

gpg: checking the trustdb
gpg: checking at depth 0 signed=1 ot(-/q/n/m/f/u)=0/0/0/0/0/1
gpg: checking at depth 1 signed=0 ot(-/q/n/m/f/u)=1/0/0/0/0/0
[ming@linux ming]$
```

不加签名：

```
[ming@linux ming]$ gpg -ear netkiller pgsql-dump.sql
gpg: WARNING: using insecure memory!
```

```
gpg: please see http://www.gnupg.org/faq.html for more information
[ming@linux ming]$ ls
netkiller.gpg    pgsql-dump.sql    pgsql-dump.sql.asc
[ming@linux ming]$
```

加密完成，将文件 pgsql-dump.sql.asc 发给 chen (邮件，WEB/FTP 下载。。。都可以，不用单心被其它人得到对你不利，现在这个文件已经加密了。)
以下为 chen 帐号解密操作：

```
[chen@linux chen]$ gpg -d pgsql-dump.sql.asc > pgsql-dump.sql
gpg: WARNING: using insecure memory!
gpg: please see http://www.gnupg.org/faq.html for more information

You need a passphrase to unlock the secret key for
user: "netkiller (陈景峰的密钥) <netkiller@9812.net>"
1024-bit ELG-E key, ID 0B70F0CB, created 2003-12-12 (main key ID B00847C5)

Enter passphrase:
gpg: encrypted with 1024-bit ELG-E key, ID 0B70F0CB, created 2003-12-12
      "netkiller (陈景峰的密钥) <netkiller@9812.net>"
[chen@linux chen]$
```

2.4 备份计划

2.4.1 服务器端计划

每天零晨 1:00-5:00 这段时间访问的人比较少，我选择服务器端每天零晨 3：00 开始备份，您也可以在其它时间段内备份，根据您的需求而定。

```
[root@linux etc]# cat crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
0 3 * * * root /usr/local/backup/backup.sh
```

2.4.2 客户端计划

客户端每天零晨 4：00 点开始下载备份数据。为什么是 4：00 下载呢？因为服务器备份要一段时间，如果服务器还没有备份完成，这边是不能下载的。所以计划在 3:00 服务器开始备份，4：00 时客户端开始下载已经备份好的数据。

```
[root@linux etc]# cat crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
0 4 * * * root /usr/local/backup/getbackup.sh
```

2.5 数据恢复

```
[root@linux root]# su postgres
bash-2.05b$ psql member -f pgsql-backup.xxxx-xx-xx.xx:xx:xx.dmp
```

2.6 性能提升

2.6.1 共享内存

在 2.2 内核里缺省的共享内存限制（ SHMMAX 和 SHMALL）都是 32 MB，但是你可以在 proc 文件系统里修改这些值（不用重起）． 比如，要允许 128 MB：

方法 1：

echo 134217728 >/proc/sys/kernel/shmall

echo 134217728 >/proc/sys/kernel/shmmax

```
[root@linux root]# cat /proc/sys/kernel/shmall
2097152
[root@linux root]# cat /proc/sys/kernel/shmmax
33554432
[root@linux root]# echo 134217728 >/proc/sys/kernel/shmall
[root@linux root]# echo 134217728 >/proc/sys/kernel/shmmax
[root@linux root]# cat /proc/sys/kernel/shmall
134217728
```

```
[root@linux root]# cat /proc/sys/kernel/shmmax
134217728
```

你可以把这些命令放到一个引导时运行的脚本中． 如 rc.local 文件

```
[root@linux root]# cat /etc/rc.d/rc.local
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
/usr/local/jakarta-tomcat/bin/startup.sh
/usr/local/apache/bin/apachectl start
echo 134217728 >/proc/sys/kernel/shmall
echo 134217728 >/proc/sys/kernel/shmmax
```

方法 2，使用 sysctl 命令来控制这些参数．

```
[root@linux root]# sysctl -w kernel.shmall=134217728
kernel.shmall = 134217728
[root@linux root]# sysctl -w kernel.shmmax=134217728
kernel.shmmax = 134217728
[root@linux root]#
```

方法 3，你可以在一个叫 /etc/sysctl.conf 的文件里面加下面这样的两行：

```
kernel.shmall = 134217728
kernel.shmmax = 134217728
```

```
[root@linux root]# cat /etc/sysctl.conf
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
```

```
kernel.core_uses_pid = 1
kernel.shmall = 134217728
kernel.shmmax = 134217728
```

通常在引导的时候会处理这个文件，但你也可以稍后明确调用 sysctl.

2.6.2 最大连接

跟据你的需要来配置最大连接数，系统默认是 32，配置需要修改两处。

max_connections = 100

shared_buffers = 200

shared_buffers = max_connections*2

```
[root@linux data]# cat postgresql.conf
#
# PostgreSQL configuration file
# -----------------------------
#
# This file consists of lines of the form:
#
#      name = value
#
# (The '=' is optional.) White space may be used. Comments are introduced
# with '#' anywhere on a line. The complete list of option names and
# allowed values can be found in the PostgreSQL documentation. The
# commented-out settings shown in this file represent the default values.
#
# Any option can also be given as a command line switch to the
# postmaster, e.g. 'postmaster -c log_connections=on'. Some options
# can be changed at run-time with the 'SET' SQL command.
#
# This file is read on postmaster startup and when the postmaster
# receives a SIGHUP. If you edit the file on a running system, you have
# to SIGHUP the postmaster for the changes to take effect, or use
# "pg_ctl reload".


#============================================================================


#
#               Connection Parameters
#
#tcpip_socket = false

tcpip_socket = true
```

```
#ssl = false
#ssl = true

#max_connections = 32
max_connections = 100
#superuser_reserved_connections = 2

#port = 5432
#hostname_lookup = false
#show_source_port = false

#unix_socket_directory = ''
#unix_socket_group = ''
#unix_socket_permissions = 0777 # octal

#virtual_host = ''

#krb_server_keyfile = ''


#
#            Shared Memory Size
#
#shared_buffers = 64                    # min max_connections*2 or 16, 8KB each
shared_buffers = 200                    # min max_connections*2 or 16, 8KB each
#max_fsm_relations = 1000               # min 10, fsm is free space map, ~40 bytes
#max_fsm_pages = 10000                  # min 1000, fsm is free space map, ~6 bytes
#max_locks_per_transaction = 64 # min 10
#wal_buffers = 8                        # min 4, typically 8KB each

#
#            Non-shared Memory Sizes
#
#sort_mem = 1024                        # min 64, size in KB
#vacuum_mem = 8192                      # min 1024, size in KB


#
#            Write-ahead log (WAL)
#
#checkpoint_segments = 3                # in logfile segments, min 1, 16MB each
#checkpoint_timeout = 300               # range 30-3600, in seconds
#
```

```
#commit_delay = 0                       # range 0-100000, in microseconds
#commit_siblings = 5                    # range 1-1000
#
#fsync = true
#wal_sync_method = fsync                # the default varies across platforms:
#                                       # fsync, fdatasync, open_sync, or open_datasync
#wal_debug = 0                          # range 0-16


#
#               Optimizer Parameters
#
#enable_seqscan = true
#enable_indexscan = true
#enable_tidscan = true
#enable_sort = true
#enable_nestloop = true
#enable_mergejoin = true
#enable_hashjoin = true

#effective_cache_size = 1000        # typically 8KB each
#random_page_cost = 4                   # units are one sequential page fetch cost
#cpu_tuple_cost = 0.01                  # (same)
#cpu_index_tuple_cost = 0.001       # (same)
#cpu_operator_cost = 0.0025         # (same)

#default_statistics_target = 10 # range 1-1000

#
#               GEQO Optimizer Parameters
#
#geqo = true
#geqo_selection_bias = 2.0          # range 1.5-2.0
#geqo_threshold = 11
#geqo_pool_size = 0                     # default based on tables in statement,
                                        # range 128-1024
#geqo_effort = 1
#geqo_generations = 0
#geqo_random_seed = -1                  # auto-compute seed



#
#               Message display
#
```

```
#server_min_messages = notice      # Values, in order of decreasing detail:
                                        #      debug5, debug4, debug3, debug2, debug1,
                                        #      info, notice, warning, error, log, fatal,
                                        #      panic
#client_min_messages = notice      # Values, in order of decreasing detail:
                                        #      debug5, debug4, debug3, debug2, debug1,
                                        #      log, info, notice, warning, error
#silent_mode = false

#log_connections = false
#log_pid = false
#log_statement = false
#log_duration = false
#log_timestamp = false

#log_min_error_statement = panic # Values in order of increasing severity:
                                        #      debug5, debug4, debug3, debug2, debug1,
                                        #      info, notice, warning, error, panic(off)

#debug_print_parse = false
#debug_print_rewritten = false
#debug_print_plan = false
#debug_pretty_print = false

#explain_pretty_print = true

# requires USE_ASSERT_CHECKING
#debug_assertions = true


#
#            Syslog
#
#syslog = 0                                 # range 0-2
#syslog_facility = 'LOCAL0'
#syslog_ident = 'postgres'


#
#            Statistics
#
#show_parser_stats = false
#show_planner_stats = false
#show_executor_stats = false
```

```
#show_statement_stats = false

# requires BTREE_BUILD_STATS
#show_btree_build_stats = false


#
#             Access statistics collection
#
#stats_start_collector = true
#stats_reset_on_server_start = true
#stats_command_string = false
#stats_row_level = false
#stats_block_level = false


#
#             Lock Tracing
#
#trace_notify = false

# requires LOCK_DEBUG
#trace_locks = false
#trace_userlocks = false
#trace_lwlocks = false
#debug_deadlocks = false
#trace_lock_oidmin = 16384
#trace_lock_table = 0


#
#             Misc
#
#autocommit = true
#dynamic_library_path = '$libdir'
#search_path = '$user,public'
#datestyle = 'iso, us'
#timezone = unknown                     # actually, defaults to TZ environment setting
#australian_timezones = false
#client_encoding = sql_ascii        # actually, defaults to database encoding
#authentication_timeout = 60        # 1-600, in seconds
#deadlock_timeout = 1000                # in milliseconds
#default_transaction_isolation = 'read committed'
#max_expr_depth = 10000                 # min 10
```

```
#max_files_per_process = 1000      # min 25
#password_encryption = true
#sql_inheritance = true
#transform_null_equals = false
#statement_timeout = 0                 # 0 is disabled, in milliseconds
#db_user_namespace = false




#
#            Locale settings
#
# (initialized by initdb -- may be changed)
LC_MESSAGES = 'en_US.UTF-8'
LC_MONETARY = 'en_US.UTF-8'
LC_NUMERIC = 'en_US.UTF-8'
LC_TIME = 'en_US.UTF-8'
```

重新启动数据：

```
[root@linux data]# service postgresql restart
                                                           [    OK    ]
Starting postgresql service:                               [    OK    ]
```

查看配置是否正确：

```
[root@linux root]# psql -Uchen member
Welcome to psql 7.3.3, the PostgreSQL interactive terminal.

Type:    \copyright for distribution terms
         \h for help with SQL commands
         \? for help on internal slash commands
         \g or terminate with semicolon to execute query
         \q to quit

member=> show max_connections;
 max_connections
-----------------
 100
(1 row)
```

2.6.3 vacuumdb

数据库 2：00 优化、3：00 备份、4：00 下载备份数据

```
[root@linux etc]# cat crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
0 2 * * * root /usr/local/pgsql/optimize.sh
```

数据库 vacuumdb 优化脚本

```
[root@linux pgsql]# cat optimize.sh
#!/bin/bash
vacuumdb -hlocalhost -p5432 -Upostgres -a -f -z
[root@linux pgsql]#
```

2.6.4 数据库操作与性能

1. 分组插入数据

   向数据库做大量 Insert 操作时（注：非导入，在某些特殊环境中要做大量的插入操作，而不是导入数据），如你有 10000 条记录要插入到数据库中，建议你将 10000 记录分组插入

第一组	begin; insert into …… insert into …… insert into …… ……. 1000 条 insert into ……. insert into …… commit;
第二组	begin; insert into …… ……. 1000 条 insert into ……. insert into …… commit;
第十组	begin; insert into …… …….

	1000 条 insert into ……. insert into …… commit;

2. 通过 copy from 插入数据

pg_copy_from
(PHP 4 >= 4.2.0)

pg_copy_from -- 根据数组将记录插入表中
说明
bool pg_copy_from (resource connection, string table_name, array rows [, string delimiter [, string null_as]])
pg_copy_from() 将数组 rows 的内容作为记录插入表中。它在内部使用了 COPY FROM SQL 命令来插入记录。如果成功则返回 TRUE，失败则返回 FALSE。

参见 pg_copy_to()。

3. 操作之后使用重建索引
   vacuumdb -hlocalhost -p5432 -Upostgres -a -f -z

2.6.5 硬件方面

1. 一般服务器
   PC 服务器有条件建议使用 SATA（串行）硬盘。
   没有条件可以买时下最快的 ATA 硬盘（也不是越快越好，还要稳定）
   正常情况下几块 ATA 66 (5400rpm)硬盘做 RAID 0，要比一块 ATA100(7200rpm)还要快。
   RAID 是解决服务器硬盘瓶颈最佳方案。建议使用 RAID 0，RAID 速度最快，就是不安全、稳ＰＣ机ＡＴＡ硬件更差。但速度诱人，只要做好备份，是没有问题的。
   目前ＰＣ上ＡＴＡ只能做ＲＡＩＤ０（条带）和ＲＡＩＤ１（镜象）。ＳＡＴＡ可以做ＲＡＩＤ０，１，５，１０，５０。不过我还没见过ＳＴＡＴ 的ＲＡＩＤ卡，深圳赛格也没得卖。
2. 高档服务器
   高档服务器中主流使用ＳＣＳＩ硬盘，公司也出得起ＹＹＹ买。所以干脆一次就配置４－５ＣＳＩ硬盘。４块盘做ＲＡＩＤ５，剩余１块做热交换 hotswap。
   因为ＳＣＳＩ性能稳定，所以满足 RAID5 速度，可以做ＲＡＩＤ０。

附表 1

RAID 级别	RAID 0	RAID 1	RAID 3	RAID 5
名称	条带	镜像	专用校验条带	分散校验条带
允许故障	否	是	是	是
冗余类型	无	副本	校验	校验
热备用操作	不可	可以	可以	可以
硬盘数量	一个以上	两个	三个以上	三个以上

可用容量	最大	最小	中间	中间
减少容量	无	50%	一个磁盘	一个磁盘
读性能	高（盘的数量决定）	中间	高	高
随机写性能	最高	中间	最低	低
连续写性能	最高	中间	低	最低
典型应用	无故障的迅速读写	允许故障的小檔、随机数据写入	允许故障的大檔、连续数据传输	允许故障的小檔、随机数据传输
可用容量	总的磁盘的容量	只能用磁盘容量的 50%	（n-1）/n 的磁盘容量。其中 n 为磁盘数	（n-1）/n 的总磁盘容量。其中 n 为磁盘数

附表 2

RAID 级别	RAID 10	RAID 30	RAID 50
名称	跨越镜像数组	跨越专用校验数组	跨越分散校验数组
允许故障	是	是	是
冗余类型	副本	校验	校验
热备用操作	可以	可以	可以
磁盘数量			
跨越2个数组	4	6,8,10,12,14 或 16	6,8,10,12,14 或 16
跨越3个数组	6	9,12 或 15	9,12 或 15
跨越4个数组	8	12 或 16	12 或 16
可用容量	最小	中间	中间
减少容量	50%	每个数组中一个磁盘	每个数组中一个磁盘
读性能	中间	高	高
随机写性能	中间	最低	低
连续写性能	中间	低	最低
典型应用	允许故障高速度小文件、随机数据写入	允许故障高速度大文件、连续数据传输	允许故障高速度小文件、随机数据传输
可用容量	磁盘容量的 50%	n-2/2 的磁盘容量。其中 n 为磁盘数目	n-2/n 的磁盘容量。其中 n 为磁盘数

3. 网络，光纤存储我没使用过，这里也不谈了。

2.7 使用 SSL 进行安全的 TCP/IP 联接

2.7.1 设置用户信息：

```
[root@linux8 root]# su - postgres
-bash-2.05b$ ls
data    initdb.i18n
-bash-2.05b$ cd data/
-bash-2.05b$ ls
```

```
base       pg_clog          pg_ident.conf   pg_xlog                postmaster.opts
global     pg_hba.conf      PG_VERSION      postgresql.conf        postmaster.pid
-bash-2.05b$ openssl req -new -text -out server.req
Using configuration from /usr/share/ssl/openssl.cnf
Generating a 1024 bit RSA private key
....++++++
.....................................................++++++
writing new private key to 'privkey.pem'
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:CN
State or Province Name (full name) [Berkshire]:Guang Zhou
Locality Name (eg, city) [Newbury]:Shen Zhen
Organization Name (eg, company) [My Company Ltd]:Open Source Organization
Organizational Unit Name (eg, section) []:technical
Common Name (eg, your name or your server's hostname) []:www.9812.net
Email Address []:netkiller@9812.net

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:chen
An optional company name []:netkiller
-bash-2.05b$ ls
base       pg_clog          pg_ident.conf   pg_xlog                postmaster.opts   privkey.pem
global     pg_hba.conf      PG_VERSION      postgresql.conf        postmaster.pid    server.req
-bash-2.05b$
```

注意上面的 server.req 文件，我们来看看它的内容：

```
-bash-2.05b$ cat server.req
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=CN, ST=Guang Zhou, L=Shen Zhen, O=Open Source Organization, OU=technical,
CN=www.9812.net/Email=netkiller@9812.net
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
```

```
                RSA Public Key: (1024 bit)
                    Modulus (1024 bit):
                        00:a5:30:9a:ef:75:9f:40:40:ee:90:4e:06:f7:f7:
                        0b:de:97:d0:1a:2e:48:ef:4c:7b:c2:cd:f2:f4:30:
                        1b:f4:c7:9d:65:7a:53:d7:d7:7c:ea:25:8f:be:b0:
                        57:f5:89:91:2e:80:4c:ff:f1:96:1e:42:06:01:64:
                        9f:98:69:24:c1:7f:e6:0c:a5:ae:b9:9c:4c:29:db:
                        a3:a3:3d:76:da:89:c0:33:29:c5:a5:8b:7a:e1:e5:
                        f4:3b:f3:7d:54:d4:65:fa:c8:c0:1c:11:07:1c:24:
                        03:8e:f0:61:d9:70:cf:fa:dd:e2:04:4a:31:c2:63:
                        2a:5f:44:ec:48:68:30:44:8d
                    Exponent: 65537 (0x10001)
            Attributes:
                challengePassword           :chen
                unstructuredName            :netkiller
        Signature Algorithm: md5WithRSAEncryption
            09:4a:1c:e5:87:7a:9c:6f:69:ed:cd:11:8d:b6:bc:da:e0:4a:
            f5:7a:33:70:0d:5f:28:63:82:79:39:6b:a5:ae:02:7b:87:cb:
            86:74:2e:2b:eb:ec:23:3b:dc:02:25:29:02:74:e7:92:76:ed:
            34:e1:63:e9:ef:dc:12:33:31:84:31:ce:b3:d4:f2:49:92:a5:
            2c:5e:0a:3d:73:f8:1f:95:8f:71:f9:2d:ee:eb:4a:9c:8c:13:
            a5:26:a2:d2:49:c3:7e:69:c7:1b:73:bb:59:8d:9c:bf:dd:ac:
            4b:c4:41:02:b1:3c:a6:c9:c9:eb:00:b3:75:2d:e2:ab:29:b3:
            85:75
-----BEGIN CERTIFICATE REQUEST-----
MIICFzCCAYACAQAwgacxCzAJBgNVBAYTAkNOMRMwEQYDVQQIEwpHdWFuZyBaaG91
MRIwEAYDVQQHEwlTaGVuIFpoZW4xITAfBgNVBAoTGE9wZW4gU291cmNlIE9yZ2Fu
aXphdGlvbjESMBAGA1UECxMJdGVjaG5pY2FsMRUwEwYDVQQDEwx3d3cuOTgxMi5u
ZXQxITAfBgkqhkiG9w0BCQEWEm5ldGtpbGxlckA5ODEyLm5ldDCBnzANBgkqhkiG
9w0BAQEFAAOBjQAwgYkCgYEApTCa73WfQEDukE4G9/cL3pfQGi5I70x7ws3y9DAb
9MedZXpT19d86iWPvrBX9YmRLoBM//GWHkIGAWSfmGkkwX/mDKWuuZxMKdujoz12
2onAMynFpYt64eX0O/N9VNRl+sjAHBEHHCQDjvBh2XDP+t3iBEoxwmMqX0TsSGgw
RI0CAwEAAaAvMBMGCSqGSIb3DQEJBzEGEwRjaGVuMBgGCSqGSIb3DQEJAjELEwlu
ZXRraWxsZXIwDQYJKoZIhvcNAQEEBQADgYEACUoc5Yd6nG9p7c0Rjba82uBK9Xoz
cA1fKGOCeTlrpa4Ce4fLhnQuK+vsIzvcAiUpAnTnknbtNOFj6e/cEjMxhDHOs9Ty
SZKlLF4KPXP4H5WPcfkt7utKnIwTpSai0knDfmnHG3O7WY2cv92sS8RBArE8psnJ
6wCzdS3iqymzhXU=
-----END CERTIFICATE REQUEST-----
```

2.7.2 生产秘钥文件：

```
-bash-2.05b$ openssl rsa -in privkey.pem -out server.key
read RSA key
```

```
Enter PEM pass phrase:
writing RSA key
-bash-2.05b$ ls
base      pg_clog       pg_ident.conf  pg_xlog          postmaster.opts  privkey.pem  server.req
global    pg_hba.conf   PG_VERSION     postgresql.conf  postmaster.pid   server.key
-bash-2.05b$
```

注意上面的 privkey.pem ，server.key 文件，我们来看看它们的内容：

```
-bash-2.05b$ cat privkey.pem
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,EE59B06E786A2FCA

C8RnlMX5tF7CRdx/jxHk/2D4SUu+PVNfphwDbsytmUJIx5qMQAHxCy+NdIDZX9L/
AWIwaShdwFOaP6CMwrzBav54DW1/IlbF688X3DA6xUY1+ZvV4RU4t1O6EhEPINth
1KBqgtSw8lnu6HQa6aIFvZ4f/Wqluk04ylGe4CLLW1xPQ36ntw3tFXPm3eIFl3wQ
lNxYjNTxSjA9x5IBzyJpaJJk27f+/WJARDkFKOwUn9J71lPC5yYybv6IG65xFg6/
kpLqfzx/wAaJxReB/EP95jLVkEmzyi6rqzsBLLgAl6mxGGN5kT34lfK4v9xuRWRz
J2AlBJnloq8NTE48N2g7N1UqHl0r3nNkLdfYEeq6th7d12hiSAcGECvSfhlWirsx
sFYcrAhBGCK+4OXjn717AYeAYw+/JPrX0ZuDVFogVKNB9x/S15+y8yh2AgIUjpJ/
BOZ3LCxXyFznu4yBvxNoTOJT2xWuAXVk5AI3UftOfBAvRZdayAwh6LdoNG77ead1
hNwIAvS5LUiLG8KeAbQHlJuh51YCpmEBCsTqrZybMNoEAiCg0Gn/5tE5cfVmH3Ei
LjhCTtRJ6oGx6dsYaY4A1Jt1+B8DMNnRTez8NN5D2+4wasr4dTYwsRXRyqMCPZJH
+z8m6zautVoHlhGQhRxO4ZcBunyJrdW5XQBGfAcUbp1xORCvqP+SW8Z4wDyu2Sk+
MlxPL1T4P3xEANG7hOlsabBiQ2kyCq1iiJCHBlfXxIm86c1ffRYTrdB+PoFyyaII
ErS68kMbv+Y5Tr+X3Ml1AMNEEU5YAn/O1wSoL5Cz0nIpKeknKAl/vA==
-----END RSA PRIVATE KEY-----

-bash-2.05b$ cat server.key
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQClMJrvdZ9AQO6QTgb39wvel9AaLkjvTHvCzfL0MBv0x51lelPX
13zqJY++sFf1iZEugEz/8ZYeQgYBZJ+YaSTBf+YMpa65nEwp26OjPXbaicAzKcWl
i3rh5fQ7831U1GX6yMAcEQccJAOO8GHZcM/63eIESjHCYypfROxIaDBEjQIDAQAB
AoGABMbGBByLkUkPXN7UtsDO+A29t7QU6c51Wamo18S4WjiXZYLG/9u8Qez6HhJt
SK1EpGqTT2dF5vQTxmCJeNe5d078YIFCbIQckgG2hLSsRyV8QclSguJLC5Tgvzua
tTFdVH50UbyAtkifiR3wt5qBuIjtxz/v0ePJ2EdhcdCAqQUCQQDUarpjOof/hTKb
wwOyJIVDycQs27dF+LiGD6YxD97WC6iZR5u7YukqzJk+GXi9EbjdQzybkp1oxDuF
LQAFXJoDAkEAxxVCo1MgYiKtc2lqSr/q2j1R//sPQq5ajv7pvU1WGhx3xS2iZt9l
/jzNx6ZUG7hxd5gi6G6I3UFAFoOLq06qLwJAPT7InvOxYqs0/FQuLJ77DaCPP5/a
KAKesYixklPRHEYgRpGvBUhvkjeLt6wAdAM4GhPY1cJgQGTUBIIFD4azoQJAVjap
xgrwoi78SFelVTupW9tkUGOL50eUJgrUdEsyd1pOr9AkXUJva9svDj/EesC0OqNi
sp9zm8VvGJDdAlGttwJBAMEnnl9ZGglIBRbS7srVLHhXFYs+xkQgTW6bvcQ+aW+G
MW/vpVcsFzSuaAtlBVoZ1ltCRGPSbVgQkp14yqGITQg=
-----END RSA PRIVATE KEY-----
```

上面的 privkey.pem ，server.key 文件内容一看就知道是 BASE64 编码的，我对它的内容也很好奇，将它解码看看内容是什么：

```
0?\┐┌┐三?氾 u 烜@類 N− 鹼  造?.H 風{峦螋 0← 羟滬 zS 鬃|?從癢鮇?€L  駷-B−┌d 煒 i$??ギ算 L)郝?v 趪?)钮媧徨?齦 T 訣  ?◀
$屏 a 賞销葩┘J1 耩*_D 霩 h0D?└┌┌┐亏┘破┘  婗 I⇳\拊独硒椒?槲 uY---啄竄 8 椶偲  奂 A 禚-↕ mH 堪  揙 gE 骠‼芙堿坠 wN 黜丁l??秭船 G%|A 萁傣 K  斷?毡 1]T~tQ 紑禜煠  鸱殃笀砬?镅闵谿 aq 裙?┐A 詳篤:??浢└?匍赡,鄯 E  ??拗  G 泯 b?蔬>├x?篙 C<洇漢??|\?┐A?B  b"瓂 ijJ 筷?Q  ?B 甖廅榻 MV→  w?  遝?頹  蛁 w?鑞堺 A@┬儖玁?┐@=>蒔蟊  b?歙.,威瓟?熩(┐灡堷扮?F  F  懷|Ho?嫹  t└8→‼卣耟@d??⇳ 喔?@V6 ├ 稷.麃 W  ;  踕 Pc 嫮 G? 詔 K2wZN  $]Bok??膠来:  玃 s 浥 o↑ 愄┐Q  A?濞 Y→ H|┬翌收,xW┴?艱 Mn 涕?io?o 铳 W,┤4 卣 e|Z├諿 BDc 襪 X┼挐 x 省土
```

看来是二制的，哈哈。

2.7.3 产生证书文件：

```
-bash-2.05b$ openssl req -x509 -in server.req -text -key server.key -out server. crt
Using configuration from /usr/share/ssl/openssl.cnf
-bash-2.05b$
-bash-2.05b$ cat server.crt
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 0 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=CN, ST=Guang Zhou, L=Shen Zhen, O=Open Source Organization, OU=technical, CN=www.9812.net/Email=netkiller@9812.net
        Validity
            Not Before: Oct 25 01:13:05 2003 GMT
            Not After : Nov 24 01:13:05 2003 GMT
        Subject: C=CN, ST=Guang Zhou, L=Shen Zhen, O=Open Source Organization, OU=technical, CN=www.9812.net/Email=netkiller@9812.net
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:a5:30:9a:ef:75:9f:40:40:ee:90:4e:06:f7:f7:
                    0b:de:97:d0:1a:2e:48:ef:4c:7b:c2:cd:f2:f4:30:
                    1b:f4:c7:9d:65:7a:53:d7:d7:7c:ea:25:8f:be:b0:
                    57:f5:89:91:2e:80:4c:ff:f1:96:1e:42:06:01:64:
                    9f:98:69:24:c1:7f:e6:0c:a5:ae:b9:9c:4c:29:db:
                    a3:a3:3d:76:da:89:c0:33:29:c5:a5:8b:7a:e1:e5:
                    f4:3b:f3:7d:54:d4:65:fa:c8:c0:1c:11:07:1c:24:
                    03:8e:f0:61:d9:70:cf:fa:dd:e2:04:4a:31:c2:63:
                    2a:5f:44:ec:48:68:30:44:8d
```

```
                    Exponent: 65537 (0x10001)
            X509v3 extensions:
                X509v3 Subject Key Identifier:
                    93:4F:D5:41:4A:CA:A2:83:19:C3:5D:BE:58:E6:45:70:7E:95:A5:0A
                X509v3 Authority Key Identifier:
                    keyid:93:4F:D5:41:4A:CA:A2:83:19:C3:5D:BE:58:E6:45:70:7E:95:A5:0A
                    DirName:/C=CN/ST=Guang Zhou/L=Shen Zhen/O=Open Source
Organization/OU=technical/CN=www.9812.net/Email=netkiller@9812.net
                    serial:00

                X509v3 Basic Constraints:
                    CA:TRUE
        Signature Algorithm: md5WithRSAEncryption
            3f:f1:99:89:37:ec:1b:80:e2:c6:3a:8e:ed:e8:94:b8:70:10:
            34:1c:9a:ef:f7:be:b7:05:51:f4:a2:cb:03:4e:f4:dd:6f:73:
            51:49:d2:91:fc:eb:40:3c:30:54:b6:f0:aa:a1:e8:d4:33:b2:
            9b:d0:0e:0d:b4:4b:65:c5:ae:bf:ed:fa:ff:1c:e6:1d:aa:41:
            4f:da:76:7a:57:7d:8d:f5:1b:17:65:fc:63:4f:db:dd:45:33:
            3d:e7:c9:dd:e8:d6:8d:6f:a5:d7:97:da:7f:cf:09:15:ab:2f:
            0a:f3:70:e0:d0:d3:50:90:05:78:92:ac:8a:17:78:23:b7:66:
            c6:55
-----BEGIN CERTIFICATE-----
MIID0DCCAzmgAwIBAgIBADANBgkqhkiG9w0BAQQFADCBpzELMAkGA1UEBhMCQ04x
EzARBgNVBAgTCkd1YW5nIFpob3UxEjAQBgNVBAcTCVNoZW4gWmhlbjEhMB8GA1UE
ChMYT3BlbiBTb3VyY2UgT3JnYW5pemF0aW9uMRIwEAYDVQQLEwl0ZWNobmljYWwx
FTATBgNVBAMTDHd3dy45ODEyLm5ldDEhMB8GCSqGSIb3DQEJARYSbmV0a2lsbGVy
QDk4MTIubmV0MB4XDTAzMTAyNTAxMTMwNVoXDTAzMTEyNDAxMTMwNVowgacxCzAJ
BgNVBAYTAkNOMRMwEQYDVQQIEwpHdWFuZyBaaG91MRIwEAYDVQQHEwlTaGVuIFpo
ZW4xITAfBgNVBAoTGE9wZW4gU291cmNlIE9yZ2FuaXphdGlvbjESMBAGA1UECxMJ
dGVjaG5pY2FsMRUwEwYDVQQDEwx3d3cuOTgxMi5uZXQxITAfBgkqhkiG9w0BCQEW
Em5ldGtpbGxlckA5ODEyLm5ldDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
pTCa73WfQEDukE4G9/cL3pfQGi5I70x7ws3y9DAb9MedZXpT19d86iWPvrBX9YmR
LoBM//GWHkIGAWSfmGkkwX/mDKWuuZxMKdujoz122onAMynFpYt64eX0O/N9VNRl
+sjAHBEHHCQDjvBh2XDP+t3iBEoxwmMqX0TsSGgwRI0CAwEAAaOCAQgwggEEMB0G
A1UdDgQWBBSTT9VBSsqigxnDXb5Y5kVwfpWlCjCB1AYDVR0jBIHMMIHJgBSTT9VB
SsqigxnDXb5Y5kVwfpWlCqGBraSBqjCBpzELMAkGA1UEBhMCQ04xEzARBgNVBAgT
Ckd1YW5nIFpob3UxEjAQBgNVBAcTCVNoZW4gWmhlbjEhMB8GA1UEChMYT3BlbiBT
b3VyY2UgT3JnYW5pemF0aW9uMRIwEAYDVQQLEwl0ZWNobmljYWwxFTATBgNVBAMT
DHd3dy45ODEyLm5ldDEhMB8GCSqGSIb3DQEJARYSbmV0a2lsbGVyQDk4MTIubmV0
ggEAMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEEBQADgYEAP/GZiTfsG4DixjqO
7eiUuHAQNBya7/e+twVR9KLLA0703W9zUUnSkfzrQDwwVLbwqqHo1DOym9AODbRL
ZcWuv+36/xzmHapBT9p2eld9jfUbF2X8Y0/b3UUzPefJ3ejWjW+l15faf88JFasv
CvNw4NDTUJAFeJKsihd4I7dmxlU=
-----END CERTIFICATE-----
```

```
-bash-2.05b$
```

2.7.4 权限方面：

删除 rm privkey.pem 文件，server.key 权限设为 600

```
-bash-2.05b$ rm privkey.pem
-bash-2.05b$ chmod og-rwx server.key
-bash-2.05b$ ls -l
total 56
drwx------  10 postgres postgres  4096 Oct 23 12:03 base
drwx------   2 postgres postgres  4096 Oct 25 08:35 global
drwx------   2 postgres postgres  4096 Jul  8 17:01 pg_clog
-rw-------   1 postgres postgres  2714 Jul  8 17:57 pg_hba.conf
-rw-------   1 postgres postgres  1441 Jul  8 17:01 pg_ident.conf
-rw-------   1 postgres postgres     4 Jul  8 17:01 PG_VERSION
drwx------   2 postgres postgres  4096 Oct 15 01:03 pg_xlog
-rw-------   1 postgres postgres  5336 Oct 24 17:01 postgresql.conf
-rw-------   1 postgres postgres    32 Oct 25 08:35 postmaster.opts
-rw-------   1 postgres postgres    44 Oct 25 08:35 postmaster.pid
-rw-r--r--   1 postgres postgres  3670 Oct 25 09:13 server.crt
-rw-------   1 postgres postgres   887 Oct 25 09:04 server.key
-rw-r--r--   1 postgres postgres  2377 Oct 25 08:59 server.req
-bash-2.05b$
```

2.7.5 配置 postgresql.conf 文件：

开启 SSL。将#ssl = false 改为 ssl = true

```
-bash-2.05b$ vi postgresql.conf
ssl = true
```

我的 postgresql.conf 文件：

```
-bash-2.05b$ cat postgresql.conf
#
# PostgreSQL configuration file
# -----------------------------
#
# This file consists of lines of the form:
#
#      name = value
#
# (The '=' is optional.) White space may be used. Comments are introduced
```

```
# with '#' anywhere on a line. The complete list of option names and
# allowed values can be found in the PostgreSQL documentation. The
# commented-out settings shown in this file represent the default values.
#
# Any option can also be given as a command line switch to the
# postmaster, e.g. 'postmaster -c log_connections=on'. Some options
# can be changed at run-time with the 'SET' SQL command.
#
# This file is read on postmaster startup and when the postmaster
# receives a SIGHUP. If you edit the file on a running system, you have
# to SIGHUP the postmaster for the changes to take effect, or use
# "pg_ctl reload".


#==========================================================================


#
#               Connection Parameters
#
#tcpip_socket = false

tcpip_socket = true

#ssl = false
ssl = true

#max_connections = 32
max_connections = 100
#superuser_reserved_connections = 2

#port = 5432
#hostname_lookup = false
#show_source_port = false

#unix_socket_directory = ''
#unix_socket_group = ''
#unix_socket_permissions = 0777 # octal

#virtual_host = ''

#krb_server_keyfile = ''
```

```
#
#               Shared Memory Size
#
#shared_buffers = 64                    # min max_connections*2 or 16, 8KB each
shared_buffers = 200                    # min max_connections*2 or 16, 8KB each
#max_fsm_relations = 1000               # min 10, fsm is free space map, ~40 bytes
#max_fsm_pages = 10000                  # min 1000, fsm is free space map, ~6 bytes
#max_locks_per_transaction = 64 # min 10
#wal_buffers = 8                        # min 4, typically 8KB each

#
#               Non-shared Memory Sizes
#
#sort_mem = 1024                        # min 64, size in KB
#vacuum_mem = 8192                      # min 1024, size in KB



#
#               Write-ahead log (WAL)
#
#checkpoint_segments = 3                # in logfile segments, min 1, 16MB each
#checkpoint_timeout = 300               # range 30-3600, in seconds
#
#commit_delay = 0                       # range 0-100000, in microseconds
#commit_siblings = 5                    # range 1-1000
#
#fsync = true
#wal_sync_method = fsync                # the default varies across platforms:
#                                       # fsync, fdatasync, open_sync, or open_datasync
#wal_debug = 0                          # range 0-16



#
#               Optimizer Parameters
#
#enable_seqscan = true
#enable_indexscan = true
#enable_tidscan = true
#enable_sort = true
#enable_nestloop = true
#enable_mergejoin = true
#enable_hashjoin = true

#effective_cache_size = 1000        # typically 8KB each
```

```
#random_page_cost = 4                  # units are one sequential page fetch cost
#cpu_tuple_cost = 0.01                 # (same)
#cpu_index_tuple_cost = 0.001      # (same)
#cpu_operator_cost = 0.0025        # (same)

#default_statistics_target = 10 # range 1-1000

#
#            GEQO Optimizer Parameters
#
#geqo = true
#geqo_selection_bias = 2.0          # range 1.5-2.0
#geqo_threshold = 11
#geqo_pool_size = 0                    # default based on tables in statement,
                                            # range 128-1024
#geqo_effort = 1
#geqo_generations = 0
#geqo_random_seed = -1                # auto-compute seed



#
#            Message display
#
#server_min_messages = notice      # Values, in order of decreasing detail:
                                            #      debug5, debug4, debug3, debug2, debug1,
                                            #      info, notice, warning, error, log, fatal,
                                            #      panic
#client_min_messages = notice      # Values, in order of decreasing detail:
                                            #      debug5, debug4, debug3, debug2, debug1,
                                            #      log, info, notice, warning, error
#silent_mode = false

#log_connections = false
#log_pid = false
#log_statement = false
#log_duration = false
#log_timestamp = false

#log_min_error_statement = panic # Values in order of increasing severity:
                                                  #      debug5, debug4, debug3, debug2, debug1,
                                                  #      info, notice, warning, error, panic(off)

#debug_print_parse = false
#debug_print_rewritten = false
```

```
#debug_print_plan = false
#debug_pretty_print = false

#explain_pretty_print = true

# requires USE_ASSERT_CHECKING
#debug_assertions = true


#
#	Syslog
#
#syslog = 0				# range 0-2
#syslog_facility = 'LOCAL0'
#syslog_ident = 'postgres'


#
#	Statistics
#
#show_parser_stats = false
#show_planner_stats = false
#show_executor_stats = false
#show_statement_stats = false

# requires BTREE_BUILD_STATS
#show_btree_build_stats = false


#
#	Access statistics collection
#
#stats_start_collector = true
#stats_reset_on_server_start = true
#stats_command_string = false
#stats_row_level = false
#stats_block_level = false


#
#	Lock Tracing
#
#trace_notify = false
```

```
# requires LOCK_DEBUG
#trace_locks = false
#trace_userlocks = false
#trace_lwlocks = false
#debug_deadlocks = false
#trace_lock_oidmin = 16384
#trace_lock_table = 0



#
#               Misc
#
#autocommit = true
#dynamic_library_path = '$libdir'
#search_path = '$user,public'
#datestyle = 'iso, us'
#timezone = unknown                     # actually, defaults to TZ environment setting
#australian_timezones = false
#client_encoding = sql_ascii        # actually, defaults to database encoding
#authentication_timeout = 60        # 1-600, in seconds
#deadlock_timeout = 1000                # in milliseconds
#default_transaction_isolation = 'read committed'
#max_expr_depth = 10000                 # min 10
#max_files_per_process = 1000       # min 25
#password_encryption = true
#sql_inheritance = true
#transform_null_equals = false
#statement_timeout = 0                  # 0 is disabled, in milliseconds
#db_user_namespace = false




#
#               Locale settings
#
# (initialized by initdb -- may be changed)
LC_MESSAGES = 'en_US.UTF-8'
LC_MONETARY = 'en_US.UTF-8'
LC_NUMERIC = 'en_US.UTF-8'
LC_TIME = 'en_US.UTF-8'
```

2.7.6 测试 SSL

```
[root@linux root]# psql -h 127.0.0.1 -Uchen member
Welcome to psql 7.3.3, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

member=> \q
[root@linux root]# service postgresql restart
                                                           [  OK  ]
Starting postgresql service:                               [  OK  ]
[root@linux root]
[root@linux root]# psql -h 127.0.0.1 -Uchen member
Welcome to psql 7.3.3, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

SSL connection (cipher: EDH-RSA-DES-CBC3-SHA, bits: 168)

member=>
```

登陆后下方显示 SSL connection (cipher: EDH-RSA-DES-CBC3-SHA, bits: 168)恭喜你成功了！

服务器将在同一个 TCP 端口上同时监听标准的和 SSL 的联接，并且将与任何正在联接的客户端进行协商，协商是否使用 SSL．参阅 Chapter 19 获取如何强制服务器端只使用 SSL 进行某些联接的信息．这段引用 http://www.pgsqldb.org/pgsqldoc-cvs/ssl-tcp.html

2.7.7 配置 pg_hba.conf 强制使用 SSL 联接：

```
-bash-2.05b$ vi pg_hba.conf
hostssl     all         all         127.0.0.1       255.255.255.255   md5
```

2.7.8 连接测试：

```
[root@linux8 root]# service postgresql restart
                                                           [  OK  ]
Starting postgresql service:                               [  OK  ]
[root@linux8 root]# psql -h 127.0.0.1 -Uchen member
Welcome to psql 7.3.3, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

SSL connection (cipher: EDH-RSA-DES-CBC3-SHA, bits: 168)

member=>
```

我的 pg_hba.conf 文件：

```
-bash-2.05b$ cat pg_hba.conf
# PostgreSQL Client Authentication Configuration File
# ===================================================
#
# Refer to the PostgreSQL Administrator's Guide, chapter "Client
# Authentication" for a complete description.  A short synopsis
# follows.
#
# This file controls: which hosts are allowed to connect, how clients
# are authenticated, which PostgreSQL user names they can use, which
# databases they can access.  Records take one of three forms:
#
# local      DATABASE  USER  METHOD  [OPTION]
# host       DATABASE  USER  IP-ADDRESS  IP-MASK  METHOD  [OPTION]
# hostssl    DATABASE  USER  IP-ADDRESS  IP-MASK  METHOD  [OPTION]
#
# (The uppercase quantities should be replaced by actual values.)
# DATABASE can be "all", "sameuser", "samegroup", a database name (or
# a comma-separated list thereof), or a file name prefixed with "@".
# USER can be "all", an actual user name or a group name prefixed with
# "+" or a list containing either.  IP-ADDRESS and IP-MASK specify the
# set of hosts the record matches.  METHOD can be "trust", "reject",
# "md5", "crypt", "password", "krb4", "krb5", "ident", or "pam".  Note
# that "password" uses clear-text passwords; "md5" is preferred for
```

```
# encrypted passwords.    OPTION is the ident map or the name of the PAM
# service.
#
# This file is read on server startup and when the postmaster receives
# a SIGHUP signal.    If you edit the file on a running system, you have
# to SIGHUP the postmaster for the changes to take effect, or use
# "pg_ctl reload".

# Put your actual configuration here
# ----------------------------------
#
# CAUTION: The default configuration allows any local user to connect
# using any PostgreSQL user name, including the superuser, over either
# Unix-domain sockets or TCP/IP.    If you are on a multiple-user
# machine, the default configuration is probably too liberal for you.
# Change it to use something other than "trust" authentication.
#
# If you want to allow non-local connections, you need to add more
# "host" records.    Also, remember TCP/IP connections are only enabled
# if you enable "tcpip_socket" in postgresql.conf.

# TYPE    DATABASE       USER            IP-ADDRESS            IP-MASK                 METHOD

local     all              all                                                          trust
host      all              all             127.0.0.1             255.255.255.255     trust

# Using sockets credentials for improved security. Not available everywhere,
# but works on Linux, *BSD (and probably some others)

#local    all        all                 ident      sameuser
#host      all              all             127.0.0.1             255.255.255.255     md5
#local     all              all                                                          trust
#host      all              all             0.0.0.0               0.0.0.0               md5
hostssl     all              all             127.0.0.1             255.255.255.255     md5
```

2.7.9 注意事项：

1. 秘钥和证书（server.key，server.crt）必须放在 data 目前中，即与 postgresql.conf 同在一个目录中
2. server.key 权限必须设为 600，所有者为 postgres，否则会提示你权限或用户、组不正确。
3. 删除 privkey.pem 文件

2.8 使用 SSH 进行安全 TCP/IP 联接

ＳＳＨ帮助信息，注意-L、-R 两个参数：

```
[chen@linux chen]$ ssh --help
Usage: ssh [options] host [command]
Options:
  -l user         Log in using this user name.
  -n              Redirect input from /dev/null.
  -F config       Config file (default: ~/.ssh/config).
  -A              Enable authentication agent forwarding.
  -a              Disable authentication agent forwarding (default).
  -X              Enable X11 connection forwarding.
  -x              Disable X11 connection forwarding (default).
  -i file         Identity for public key authentication (default: ~/.ssh/identity)
  -t              Tty; allocate a tty even if command is given.
  -T              Do not allocate a tty.
  -v              Verbose; display verbose debugging messages.
                  Multiple -v increases verbosity.
  -V              Display version number only.
  -P              Don't allocate a privileged port.
  -q              Quiet; don't display any warning messages.
  -f              Fork into background after authentication.
  -e char         Set escape character; ``none'' = disable (default: ~).
  -c cipher       Select encryption algorithm
  -m macs         Specify MAC algorithms for protocol version 2.
  -p port         Connect to this port.    Server must be on the same port.
  -L listen-port:host:port      Forward local port to remote address
  -R listen-port:host:port      Forward remote port to local address
                  These cause ssh to listen for connections on a port, and
                  forward them to the other side by connecting to host:port.
  -D port         Enable dynamic application-level port forwarding.
  -C              Enable compression.
  -N              Do not execute a shell or command.
  -g              Allow remote hosts to connect to forwarded ports.
  -1              Force protocol version 1.
  -2              Force protocol version 2.
  -4              Use IPv4 only.
  -6              Use IPv6 only.
  -o 'option' Process the option as if it was read from a configuration file.
  -s              Invoke command (mandatory) as SSH2 subsystem.
  -b addr         Local IP address.
```

说明：

-L listen-port:host:port　　转发本地端口到远程地址

-R listen-port:host:port　　转发远程端口到本地地址

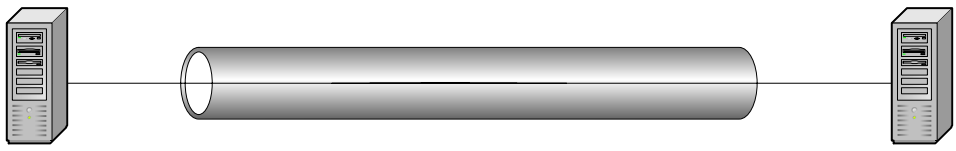
使用方法：

ssh –L 本地端口:连接 PostgreSQL 的 host:5432 登录用户@要转发的远程主机

```
[root@linux8 root]# ssh –L 3333:localhost:5432 root@127.0.0.1
root@127.0.0.1's password:
Last login: Sat Oct 25 09:27:30 2003 from 192.168.1.2
[root@linux8 root]# psql –p3333 –Uchen
psql: could not connect to server: No such file or directory
        Is the server running locally and accepting
        Connections on Unix domain socket "/tmp/.s.PGSQL.3333"?
[root@linux8 root]# psql –p3333 –Uchen member
psql: could not connect to server: No such file or directory
        Is the server running locally and accepting
        Connections on Unix domain socket "/tmp/.s.PGSQL.3333"?
[root@linux8 root]# psql –h 127.0.0.1 –p3333 –Uchen member
Welcome to psql 7.3.3, the PostgreSQL interactive terminal.

Type:   \copyright for distribution terms
        \h for help with SQL commands
        \? For help on internal slash commands
        \g or terminate with semicolon to execute query
        \q to quit

SSL connection (cipher: EDH-RSA-DES-CBC3-SHA, bits: 168)

Member=>
```

注意：我上面用了 SSH+SSL。服务器→SSH→SSL-------Fast Ethernet------ SSL→SSH→客户端

2.8.1 实例 1



请看上面图片，现在假设 server1 应用服务器，server2 是数据库数据服务器。现在要从 server1 连接通过 SSH 连接 server2。

server1 环境：

IP：192.168.0.1

域名：client.9812.net

server2 环境：

IP：192.168.0.2

域名：server.9812.net

SSH 命令：

登陆 server1 输入命令

server1$ ssh -f –n –L 8000:server.9812.net:5432 server.9812.net

server1$ psql –h 127.0.0.1 –p 8000 –Unetkiller mydb

3 数据定义（DDL）

3.1 日期时间常量

3.1.1 当前日期

current_date

```
netkiller=> select current_date;
      date
------------
 2003-11-28
(1 row)

netkiller=>
```

3.1.2 当前时间

current_time

```
netkiller=> select current_time;
           timetz
--------------------
 19:38:47.270235+08
(1 row)

netkiller=>
```

3.1.3 当前日期时间

current_timestamp

```
netkiller=> select current_timestamp;
                timestamptz
```

```
-------------------------------
 2003-11-28 19:39:25.548505+08
(1 row)

netkiller=>
```

3.1.4 除去时区

1. current_timestamp::timestamp (0)
2. current_timestamp::timestamp (0) without time zone;

```
netkiller=> select current_timestamp::timestamp (0);
          timestamp
---------------------
 2003-11-28 19:44:33
(1 row)

netkiller=>
netkiller=> select current_timestamp::timestamp (0) without time zone;
          timestamp
---------------------
 2003-11-28 19:40:10
(1 row)

netkiller=>
```

3.1.5 计算时间差

```
netkiller=> select to_date('2003-12-2','YYYY-MM-DD')-to_date('2003-12-1','YYYY-MM-DD');
?column?
----------
1
(1 row)

netkiller=>
netkiller=> select to_date('2003-12-2','YYYY-MM-DD')-to_date('2003-10-2','YYYY-MM-DD');
?column?
----------
61
(1 row)
```

3.1.6 计算时间和

```
netkiller=> select to_date('2003-12-6','yyyy-mm-dd')+12 ;
?column?
------------
2003-12-18
(1 row)

netkiller=> select to_date('2003-12-6','yyyy-mm-dd')+20 ;
?column?
------------
2003-12-26
(1 row)
```

3.1.7 date_part

```
netkiller=> select date_part('epoch', '2003-12-3 10:20:30' - timestamp '2003-12-1 02:00:00') ;
 date_part
-----------
      202830
(1 row)

netkiller=> select date_part('day', '2003-12-3 10:20:30' - timestamp '2003-12-1 02:00:00') ;
 date_part
-----------
              2
(1 row)

netkiller=> select date_part('hour', '2003-12-3 10:20:30' - timestamp '2003-12-1 02:00:00') ;
date_part
-----------
              8
(1 row)

netkiller=>
```

详细使用方法请参考 http://www.pgsqldb.org 上的文档。

3.2 汉字做字段名

PostgreSQL 是支持“区域”，“字符集支持”的，允许你使用本区域的字符集做为字段名。但要注意，你的终端要支持该字符集支持。我这里使用 UNICODE，EUC_CN 也适用。

```
Create table "组"(
        "序号" Serial NOT NULL UNIQUE,
        "组名" Varchar(20) NOT NULL,
        "描述" Varchar(255),
        UNIQUE   ("组名"),
        PRIMARY KEY ("序号")
);
```

创建表：

```
member=> Create table "组"(
member(> "序号" Serial NOT NULL UNIQUE,
member(> "组名" Varchar(20) NOT NULL,
member(> "描述" Varchar(255),
member(> UNIQUE ("组名"),
member(>    PRIMARY KEY ("序号")
member(> );
NOTICE:    CREATE TABLE will create implicit sequence '组_序号_seq' for SERIAL column '组.序号'
NOTICE:    CREATE TABLE / PRIMARY KEY will create implicit index '组_pkey' for table '组'
NOTICE:    CREATE TABLE / UNIQUE will create implicit index '组_组名_key' for table '组'
CREATE TABLE
member=> \d
                        List of relations
 Schema |              Name              |      Type      | Owner
--------+--------------------------------+----------------+-------
 public | group                          | table          | chen
 public | group_id_seq                   | sequence       | chen
 public | groupmember                    | table          | chen
 public | groupmember_id_seq             | sequence       | chen
 public | role                           | table          | chen
 public | role_id_seq                    | sequence       | chen
 public | rolemember                     | table          | chen
 public | rolemember_id_seq              | sequence       | chen
 public | system_log                     | table          | chen
 public | system_log_id_seq              | sequence       | chen
 public | trust                          | table          | chen
 public | trust_id_seq                   | sequence       | chen
 public | user                           | table          | chen
 public | user_id_seq                    | sequence       | chen
 public | user_log                       | table          | chen
 public | user_log_id_seq                | sequence       | chen
 public | userinfo                       | table          | chen
 public | userinfo_id_seq                | sequence       | chen
 public | vgroup                         | view           | chen
 public | vgroupmember                   | view           | chen
```

```
 public | vsystem_log          | view     | chen
 public | vuser                | view     | chen
 public | 组                   | table    | chen
 public | 组_序号_seq          | sequence | chen
(24 rows)
```

查看表结构：

```
member=> \d 组
                                    Table "public.组"
 Column |          Type          |                          Modifiers
--------+------------------------+-------------------------------------------------------------
 序号   | integer                | not null default nextval('public."组_序号_seq"'::text)
 组名   | character varying(20)  | not null
 描述   | character varying(255) |
Indexes:  组_pkey primary key btree ("序号"),
          组_组名_key unique btree ("组名")
```

插入数据：

```
member=> insert into 组(组名,描述) values('域用户','9812.net 域内用户');
INSERT 110971 1
member=> insert into "组"("组名","描述") values('域用户','9812.net 域内用户');
ERROR:  Cannot insert a duplicate key into unique index 组_组名_key
member=> insert into "组"("组名","描述") values('计算机维护组','维护计算机的用户用户');
INSERT 110973 1
```

查看数据：

```
member=> select * from 组;
 序号 |      组名      |              描述
------+----------------+--------------------------------
    1 | 域用户         | 9812.net 域内用户
    3 | 计算机维护组   | 维护计算机的用户用户
(2 rows)
member=> select * from "组";
 序号 |      组名      |              描述
------+----------------+--------------------------------
    1 | 域用户         | 9812.net 域内用户
    3 | 计算机维护组   | 维护计算机的用户用户
(2 rows)
```

注：在操作非英文字段的表时。建议最好前，后加上“"”，“"”符号。并非所有 API 都支持非英文的编码。

3.3 “::”数据转换

PostgreSQL 数据之间的转换可以使用“::”操作符。

3.3.1 text to varchar

vperson 表 gender 字段为布尔型（boolean）在视图中要显示为 true 显示为“先生”，false 显示为“女士”CASE WHEN 表达式应该是：
CASE WHEN p.gender = true THEN '先生' ELSE '女士' END as gender,
直接使用'先生', '女士' PostgreSQL 认为' '中间的字符为 text 类型，请看下面：

```
postgres=# CREATE OR REPLACE VIEW vperson AS
postgres-#         SELECT p.uid,p.name,
postgres-#         CASE WHEN p.gender = true THEN '先生' ELSE '女士' END as gender,
postgres-#         p.nickname,p.mobile,p.tel,p.fax,p.email,p.province,p.city,p.addre
ss,p.postalcode
postgres-#         FROM "person" p
postgres-#         Order By p.uid;
CREATE VIEW
postgres=# \dv vperson
              List of relations
 Schema |    Name     | Type |    Owner
--------+---------+------+----------
 public | vperson | view | postgres
(1 row)

postgres=# \d person
                              Table "public.person"
    Column       |                Type                 |          Modifiers
-------------+-----------------------+---------------------
 uid             | integer                          | not null default 0
 name            | character varying(20)    | not null
 gender          | boolean                          | not null default 'F'
 nickname        | character varying(20)    |
 mobile          | character varying(13)    |
 tel             | character varying(20)    | not null
 fax             | character varying(20)    |
 email           | character varying(60)    |
 province        | character varying(10)    | not null
 city            | character varying(10)    | not null
 address         | character varying(255) | not null
 postalcode    | character varying(6)      | not null
 rate            | character varying(20)    | default '0'
```

```
 bank        | character varying(20)   | not null default ''
 bankaccount | character varying(20)   | not null default ''
Indexes: person_pkey primary key btree (uid)
Check constraints: "person_rate" ((((((rate = '0'::character varying) OR (rate = '1'::character varying)) OR (rate = '2'::character varying)) OR (rate = '3'::character varying)) OR (rate = '4'::character varying)) OR (rate = '5'::character varying))

postgres=#
postgres=# \d vperson
                    View "public.vperson"
   Column   |                Type                | Modifiers
------------+------------------------+-----------
 uid        | integer                            |
 name       | character varying(20)  |
 gender     | text                               |
 nickname   | character varying(20)  |
 mobile     | character varying(13)  |
 tel        | character varying(20)  |
 fax        | character varying(20)  |
 email      | character varying(60)  |
 province   | character varying(10)  |
 city       | character varying(10)  |
 address    | character varying(255) |
 postalcode | character varying(6)   |
View definition: SELECT p.uid, p.name, CASE WHEN (p.gender = true) THEN '先生'::
text ELSE '女士'::text END AS gender, p.nickname, p.mobile, p.tel, p.fax, p.emai
l, p.province, p.city, p.address, p.postalcode FROM person p ORDER BY p.uid;
```

使用“::”将 test 转为 varchar：

```
CASE WHEN p.gender = true THEN '先生'::varchar(2) ELSE '女士'::varchar(2) END as gender,
```

例：

```
CREATE OR REPLACE VIEW vperson AS
    SELECT p.uid,p.name,
        CASE WHEN p.gender = true THEN '先生'::varchar(2) ELSE '女士'::varchar(2) END as gender,
        p.nickname,p.mobile,p.tel,p.fax,p.email,p.province,p.city,p.address,p.postalcode
    FROM "person" p
    Order By p.uid;
```

```
postgres=# drop view vperson ;
DROP VIEW
postgres=# CREATE OR REPLACE VIEW vperson AS
postgres-#     SELECT p.uid,p.name,
postgres-#     CASE WHEN p.gender = true THEN '先生'::varchar(2) ELSE '女士'::varchar(2) END as
gender,
```

```
postgres-#         p.nickname,p.mobile,p.tel,p.fax,p.email,p.province,p.city,p.address,p.postalcode
postgres-#         FROM "person" p
postgres-#         Order By p.uid;
CREATE VIEW
postgres=# \d vperson
                      View "public.vperson"
     Column     |                Type                | Modifiers
------------+------------------------+-----------
  uid           | integer                          |
  name          | character varying(20)    |
  gender        | character varying(2)     |
  nickname      | character varying(20)    |
  mobile        | character varying(13)    |
  tel           | character varying(20)    |
  fax           | character varying(20)    |
  email         | character varying(60)    |
  province      | character varying(10)    |
  city          | character varying(10)    |
  address       | character varying(255) |
  postalcode | character varying(6)     |
View definition: SELECT p.uid, p.name, CASE WHEN (p.gender = true) THEN ('先生'::character varying)::character varying(2) ELSE ('女士'::character varying)::character varying(2) END AS gender, p.nickname, p.mobile, p.tel, p.fax, p.email, p.province, p.city, p.address, p.postalcode FROM person p ORDER BY p.uid;

postgres=#
```

3.4 序列

3.4.1 等差列

```
-- ------------------------------------------------------
--    'Region'
-- ------------------------------------------------------
DROP TABLE region;
DROP SEQUENCE       region_id_seq;
DROP INDEX          region_id_index;
DROP VIEW vregion;

CREATE TABLE region (
       id                integer DEFAULT nextval('region_id_seq') NOT NULL,
       region        varchar(20) DEFAULT '' NOT NULL,
       description    text ,
       note         text ,
```

```
        remark              text ,
        create_date    timestamp DEFAULT now() ,
        modify_date    timestamp DEFAULT now() ,
        PRIMARY KEY (id),
        UNIQUE (id,region)
);
CREATE SEQUENCE   region_id_seq;
CREATE INDEX          region_id_index ON region (id);

CREATE VIEW vregion AS
        SELECT       pv.id,pv.region,pv.description,pv.note,pv.remark,to_char(pv.create_date,'YYYY-MM-DD
HH:MI:SS') as date
                FROM region pv
                ORDER BY pv.id;
```

3.4.2 “1，2，3，4，5，6，7，8，9…”

```
DROP SEQUENCE       region_id_seq;
CREATE SEQUENCE   region_id_seq;
```

```
member=> insert into region(region) values('广西');
INSERT 111264 1
member=>
member=> insert into region(region) values('贵州');
INSERT 111265 1
member=>
member=> insert into region(region) values('海南');
INSERT 111266 1
member=>
member=> insert into region(region) values('河北');
INSERT 111267 1
member=>
member=> insert into region(region) values('河南');
INSERT 111268 1
member=>
member=> insert into region(region) values('黑龙江');
INSERT 111269 1
member=> select * from vregion ;
 id | region | description | note | remark |             date
----+--------+-------------+------+--------+---------------------
  1 | 安徽   |             |      |        | 2003-11-01 10:44:26
  2 | 北京   |             |      |        | 2003-11-01 10:44:26
  3 | 重庆   |             |      |        | 2003-11-01 10:44:26
  4 | 福建   |             |      |        | 2003-11-01 10:44:26
  5 | 甘肃   |             |      |        | 2003-11-01 10:44:26
  6 | 广东   |             |      |        | 2003-11-01 10:44:26
  7 | 广西   |             |      |        | 2003-11-01 10:44:26
```

```
  8 | 贵州   |             |        |            | 2003-11-01 10:44:26
  9 | 海南   |             |        |            | 2003-11-01 10:44:26
 10 | 河北   |             |        |            | 2003-11-01 10:44:26
 11 | 河南   |             |        |            | 2003-11-01 10:44:26
 12 | 黑龙江 |             |        |            | 2003-11-01 10:44:26
(12 rows)
```

3.4.3 “1，3，5，7，9…”

```
DROP SEQUENCE       region_id_seq;
Delete from region;
CREATE SEQUENCE region_id_seq INCREMENT 2 START 1;
```

```
member=> DROP SEQUENCE region_id_seq;
DROP SEQUENCE
member=> Delete from region;
DELETE 15
member=>
member=> CREATE SEQUENCE region_id_seq INCREMENT 2 START 1;
CREATE SEQUENCE
member=> insert into region(region) values('广东');
INSERT 111282 1
member=>
member=> insert into region(region) values('广西');
INSERT 111283 1
member=>
member=> insert into region(region) values('贵州');
INSERT 111284 1
member=>
member=> insert into region(region) values('海南');
INSERT 111285 1
member=>
member=> insert into region(region) values('河北');
INSERT 111286 1
member=>
member=> insert into region(region) values('河南');
INSERT 111287 1
member=>
member=> insert into region(region) values('黑龙江');
INSERT 111288 1
member=> select * from region ;
 id | region | description | note | remark |            create_date            |            modify_date
----+--------+-------------+------+--------+-----------------------------------+-----------------------------------
```

```
  1 | 安徽     |                |        |            | 2003-11-01 11:49:58.004475 | 2003-11-01
11:49:58.004475
  3 | 北京     |                |        |            | 2003-11-01 11:49:58.093188 | 2003-11-01
11:49:58.093188
  5 | 重庆     |                |        |            | 2003-11-01 11:49:58.138582 | 2003-11-01
11:49:58.138582
  7 | 福建     |                |        |            | 2003-11-01 11:49:58.166903 | 2003-11-01
11:49:58.166903
  9 | 甘肃     |                |        |            | 2003-11-01 11:49:58.195132 | 2003-11-01
11:49:58.195132
 11 | 广东     |                |        |            | 2003-11-01 11:49:58.239133 | 2003-11-01
11:49:58.239133
 13 | 广西     |                |        |            | 2003-11-01 11:49:58.267372 | 2003-11-01
11:49:58.267372
 15 | 贵州     |                |        |            | 2003-11-01 11:49:58.295643 | 2003-11-01
11:49:58.295643
 17 | 海南     |                |        |            | 2003-11-01 11:49:58.324202 | 2003-11-01
11:49:58.324202
 19 | 河北     |                |        |            | 2003-11-01 11:49:58.352543 | 2003-11-01
11:49:58.352543
 21 | 河南     |                |        |            | 2003-11-01 11:49:58.381273 | 2003-11-01
11:49:58.381273
 23 | 黑龙江   |                |        |            | 2003-11-01 11:49:58.415112 | 2003-11-01
11:49:58.415112
(12 rows)
```

3.4.4 “2，4，6，8，10…”

```
DROP SEQUENCE       region_id_seq;
Delete from region;
CREATE SEQUENCE region_id_seq INCREMENT 2 START 2;
```

```
member=> DROP SEQUENCE region_id_seq;
ERROR:    sequence "region_id_seq" does not exist
member=> Delete from region;
DELETE 0
member=> CREATE SEQUENCE region_id_seq INCREMENT 2 START 2;
CREATE SEQUENCE
member=> insert into region(region) values('安徽');
INSERT 111303 1
member=> insert into region(region) values('北京');
INSERT 111304 1
……
……
```

```
member=> insert into region(region) values('海南');
INSERT 111311 1
member=> insert into region(region) values('河北');
INSERT 111312 1
member=> select * from vregion;
 id | region | description | note | remark |             date
----+--------+-------------+------+--------+---------------------
  2 | 安徽   |             |      |        | 2003-11-01 12:00:28
  4 | 北京   |             |      |        | 2003-11-01 12:00:28
  6 | 重庆   |             |      |        | 2003-11-01 12:00:28
  8 | 福建   |             |      |        | 2003-11-01 12:00:28
 10 | 甘肃   |             |      |        | 2003-11-01 12:00:28
 12 | 广东   |             |      |        | 2003-11-01 12:00:28
 14 | 广西   |             |      |        | 2003-11-01 12:00:28
 16 | 贵州   |             |      |        | 2003-11-01 12:00:28
 18 | 海南   |             |      |        | 2003-11-01 12:00:28
 20 | 河北   |             |      |        | 2003-11-01 12:00:28
(10 rows)
```

3.4.5 n_1+n_2

CREATE SEQUENCE region_id_seq INCREMENT n_2 START n_1;

3.5 约束

3.6 检查约束

有这样一个需求，在很多电子商务网站上都要对用户进行诚信评估，诚信分为五级（五个星），这样就要求某字段插入的数据 0，1，2，3，4，5。“0”表示该用户没用评估。

```
-- ============================================================
--   'trust'
-- ============================================================
Create table "trust"
(
        "id" Serial NOT NULL UNIQUE,
        "uid" integer NOT NULL Default 0,
        "rate" Varchar(20) Default '0' Check (rate in ('0','1','2','3','4','5')),
       primary key ("id")
);
Alter table "trust" add    foreign key ("uid") references "user" ("id") on update restrict on delete restrict;
```

```
member=> Insert into trust (uid) values((select id from "user" where userid='netkiller'));
```

```
INSERT 111237 1
member=> Insert into trust (uid,rate) values((select id from "user" where userid='netkiller'),5);
INSERT 111220 1
member=> Insert into trust (uid,rate) values((select id from "user" where userid='netkiller'),2);
INSERT 111236 1
member=> Insert into trust (uid,rate) values((select id from "user" where userid='netkiller'),6);
ERROR:    ExecInsert: rejected due to CHECK constraint "trust_rate" on "trust"
member=> Insert into trust (uid,rate) values((select id from "user" where userid='netkiller'),10);
ERROR:    ExecInsert: rejected due to CHECK constraint "trust_rate" on "trust"
member=> select * from trust;
  id | uid | rate
----+-----+------
   1 | 257 | 2
   4 | 257 | 0
   5 | 257 | 5
(3 rows)
```

当插入数据不在枚举的范围内，提示 ERROR: ExecInsert: rejected due to CHECK constraint "trust_rate" on "trust"。

3.7 非空约束

显示的有 note 字段为空的记录：

```
member=> select * from vregion where note is null;
```

3.8 唯一约束

3.8.1 单字段约束

这个例子对 groupname 字段做唯一操作。

```
-- ============================================================
--   'group'
-- ============================================================
Create table "group"
(
      "id" Serial NOT NULL UNIQUE,
      "groupname" Varchar(20) NOT NULL,
      "description" Varchar(255),
      UNIQUE   (groupname),
      PRIMARY KEY ("id")
);
```

测试：

```
member=> insert into "group"(groupname,description) values('Admin','xxxxxxxxxxxxxxxxx');
INSERT 110497 1
member=> insert into "group"(groupname,description) values('Admin','xxxxxxxxxxxxxxxxx');
ERROR:  Cannot insert a duplicate key into unique index group_groupname_key
member=> insert into "group"(groupname,description) values('Admin','xxxxxxxxxxxxxxxxx');
ERROR:  Cannot insert a duplicate key into unique index group_groupname_key
```

Psql 命令行返回 ERROR: Cannot insert a duplicate key into unique index group_groupname_key 唯一约束成功。

3.8.2 多个字段组合约束

UNIQUE (rid,uid)中有多个参数，是对 rid,uid 组合约束。
例如：
1，1
1，2
是正确的

1，1
2，1
也是正确的

2，1
1，1
2，2
1，2
1，1
不正确的不允许插入数据“1，1”，数据“1，1”出现了两次，所以要同时满足 rid,uid 两个条件。

三个字段以上组合：
1,1,1
1,1,2
1,2,1
2,1,2
2,1,1
2,2,2
正确可以插入数据

1，2，1
2，1，2
2，2，1
1，1，2
2，2，1
“2，2，1”,“2，2，1”出现两次，违反约束条件，所以不能再次插入数据“2，2，1”。

```
-- ============================================================
--    'rolemember'
-- ============================================================
-- drop table rolemember CASCADE ;
Create table "rolemember"
(
      "id" Serial NOT NULL UNIQUE,
      "rid" integer NOT NULL Default 0,
      "uid" integer NOT NULL Default 0,
      UNIQUE   (rid,uid),
  primary key ("id")
);
```

```
member=> insert into rolemember(rid,uid) values((select id from role where rolename ='System'),(select id from vuser where userid='sysop'));
INSERT 110954 1
member=> insert into rolemember(rid,uid) values((select id from role where rolename ='System'),(select id from vuser where userid='sysop'));
ERROR:    Cannot insert a duplicate key into unique index rolemember_rid_key
member=> insert into rolemember(rid,uid) values((select id from role where rolename ='System'),(select id from vuser where userid='admin'));
ERROR:    More than one tuple returned by a subselect used as an expression.
member=> insert into rolemember(rid,uid) values((select id from role where rolename ='System'),(select id from vuser where userid='test'));
INSERT 110956 1
member=> insert into rolemember(rid,uid) values((select id from role where rolename ='System'),(select id from vuser where userid='test'));
ERROR:    Cannot insert a duplicate key into unique index rolemember_rid_key
```

3.8.3 唯一约束的注意事项

这个例子对 groupname 字段做唯一操作。

```
-- ============================================================
--    'group'
-- ============================================================
Create table "group"
(
      "id" Serial NOT NULL UNIQUE,
      "groupname" Varchar(20) NOT NULL,
      "description" Varchar(255),
      UNIQUE   (id,groupname),
      PRIMARY KEY ("id")
);
```

仔细看这个例子没有错。

运行结果：

```
postgres=# Create table "group"
postgres-# (
postgres(# "id" Serial NOT NULL UNIQUE,
postgres(# "groupname" Varchar(20) NOT NULL,
postgres(# "description" Varchar(255),
postgres(# UNIQUE (id,groupname),
postgres(#    PRIMARY KEY ("id")
postgres(# );
NOTICE:  CREATE TABLE will create implicit sequence 'group_id_seq' for SERIAL column 'group.id'
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index 'group_pkey' for table 'group'
NOTICE:  CREATE TABLE / UNIQUE will create implicit index 'group_id_key' for table 'group'
CREATE TABLE
```

运行结果也没有错，现在插入数据。

```
insert into "group"(groupname,description) values('Admin','xxxxxxxxxxxxxxxxxx');
insert into "group"(groupname,description) values('Guest','xxxxxxxxxxxxxxxxxx');
insert into "group"(groupname,description) values('Domain','xxxxxxxxxxxxxxxxxx');
```

```
postgres=# insert into "group"(groupname,description) values('Admin','xxxxxxxxxxxxxxxxxx');
INSERT 110466 1
postgres=# insert into "group"(groupname,description) values('Guest','xxxxxxxxxxxxxxxxxx');
INSERT 110467 1
postgres=# insert into "group"(groupname,description) values('Domain','xxxxxxxxxxxxxxxxxx');
INSERT 110468 1
postgres=#
postgres=# insert into "group"(groupname,description) values('Admin','xxxxxxxxxxxxxxxxxx');
INSERT 110469 1
postgres=# insert into "group"(groupname,description) values('Guest','xxxxxxxxxxxxxxxxxx');
INSERT 110470 1
postgres=# insert into "group"(groupname,description) values('Domain','xxxxxxxxxxxxxxxxxx');
INSERT 110471 1
postgres=# insert into "group"(groupname,description) values('Admin','xxxxxxxxxxxxxxxxxx');
INSERT 110472 1
postgres=# insert into "group"(groupname,description) values('Guest','xxxxxxxxxxxxxxxxxx');
INSERT 110473 1
postgres=# insert into "group"(groupname,description) values('Domain','xxxxxxxxxxxxxxxxxx');
INSERT 110474 1
postgres=# insert into "group"(groupname,description) values('Admin','xxxxxxxxxxxxxxxxxx');
INSERT 110475 1
postgres=# insert into "group"(groupname,description) values('Guest','xxxxxxxxxxxxxxxxxx');
INSERT 110476 1
postgres=# insert into "group"(groupname,description) values('Domain','xxxxxxxxxxxxxxxxxx');
```

```
INSERT 110477 1
postgres=# insert into "group"(groupname,description) values('Admin','xxxxxxxxxxxxxxxxx');
INSERT 110478 1
postgres=# insert into "group"(groupname,description) values('Guest','xxxxxxxxxxxxxxxxx');
INSERT 110479 1
postgres=# insert into "group"(groupname,description) values('Domain','xxxxxxxxxxxxxxxxx');
INSERT 110480 1
postgres=# select * from "group";
 id | groupname |        description
----+-----------+--------------------
  1 | Admin     | xxxxxxxxxxxxxxxxx
  2 | Guest     | xxxxxxxxxxxxxxxxx
  3 | Domain    | xxxxxxxxxxxxxxxxx
  4 | Admin     | xxxxxxxxxxxxxxxxx
  5 | Guest     | xxxxxxxxxxxxxxxxx
  6 | Domain    | xxxxxxxxxxxxxxxxx
  7 | Admin     | xxxxxxxxxxxxxxxxx
  8 | Guest     | xxxxxxxxxxxxxxxxx
  9 | Domain    | xxxxxxxxxxxxxxxxx
 10 | Admin     | xxxxxxxxxxxxxxxxx
 11 | Guest     | xxxxxxxxxxxxxxxxx
 12 | Domain    | xxxxxxxxxxxxxxxxx
 13 | Admin     | xxxxxxxxxxxxxxxxx
 14 | Guest     | xxxxxxxxxxxxxxxxx
 15 | Domain    | xxxxxxxxxxxxxxxxx
(15 rows)
```

但你会发现对 groupname 字段的唯一约束不起使用。失效原因：
"id" Serial NOT NULL UNIQUE, (唯一约束)
UNIQUE (id,groupname), (id 字段又做了一次唯一约束)
这就是它失效的原因。正确的脚本写法是：

```
Create table "group"
(
    "id" Serial NOT NULL UNIQUE,
    "groupname" Varchar(20) NOT NULL,
    "description" Varchar(255),
    UNIQUE (groupname),
    PRIMARY KEY ("id")
);
```

```
member=> insert into "group"(groupname,description) values('Admin','xxxxxxxxxxxxxxxxx');
INSERT 110497 1
member=> insert into "group"(groupname,description) values('Admin','xxxxxxxxxxxxxxxxx');
ERROR:  Cannot insert a duplicate key into unique index group_groupname_key
member=> insert into "group"(groupname,description) values('Admin','xxxxxxxxxxxxxxxxx');
```

ERROR: Cannot insert a duplicate key into unique index group_groupname_key

Psql 命令行反回 ERROR: Cannot insert a duplicate key into unique index group_groupname_key
唯一约束成功。

3.9 主键/外键

3.9.1 主键

下面书写方式，推荐第二种，比较清晰。

```
CREATE TABLE products (
    product_no integer PRIMARY KEY,
    name text,
    price numeric
);

CREATE TABLE example (
    a integer,
    b integer,
    c integer,
    PRIMARY KEY (a, c)
);
```

3.9.2 外键约束

下面第前两种写法不推荐。第三、四种写法较清晰。

1. 第一种书写方式

```
CREATE TABLE orders (
    order_id integer PRIMARY KEY,
    product_no integer REFERENCES products,
    quantity integer
);
```

2. 第二种书写方式

```
CREATE TABLE orders (
    order_id integer PRIMARY KEY,
    product_no integer REFERENCES products (product_no),
    quantity integer
);
```

3. 第三种书写方式

```
CREATE TABLE table1 (
  a integer PRIMARY KEY,
  b integer,
  c integer,
```

FOREIGN KEY (b, c) REFERENCES other_table (c1, c2)
);

4. 第四种书写方式，在 SQL 脚本最后面添加外键约束

Alter table "groupmember" add foreign key ("uid") references "user" ("id") on update restrict on delete restrict;

Alter table "groupmember" add foreign key ("gid") references "group" ("id") on update restrict on delete restrict;

Alter table "rolemember" add foreign key ("uid") references "user" ("id") on update restrict on delete restrict;

Alter table "rolemember" add foreign key ("rid") references "role" ("id") on update restrict on delete restrict;

3.9.3 PostgreSQL 7.3.x 新增功能

```
CREATE TABLE order_items (
    product_no integer REFERENCES products ON DELETE RESTRICT,
    order_id integer REFERENCES orders ON DELETE CASCADE,
    quantity integer,
    PRIMARY KEY (product_no, order_id)
);
```

类似 ON DELETE，还有 ON UPDATE 选项，它是在主键被修改（更新）的时候调用的。

以前我们删除其它表中受外键约束的记录，使用规则或触发器来完成。现可以用 CASCADE

3.9.4 例子-分类目录

实现一个无限向下分类的目录，例如：

计算机与互联网

- 免费资源
  - 软件下载(3431)
  - 壁纸/屏保/桌面(109)
  - 免费电子贺卡(197)
  - 代理服务器(33)
  - 免费电子邮箱(73)
  - 免费主页空间(75)
  - 免费聊天室(11)
  - 免费论坛(36)
- 软件
  - XXXXXXXX
  - XXXXXXXX
  - XXXXXXXX
  - XXXXXXXX
- 硬件
- 互联网

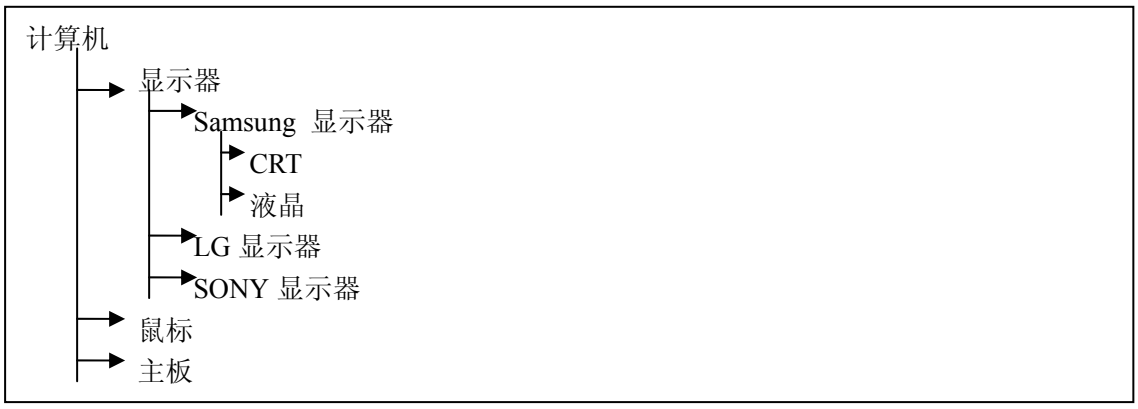
编程

数据结构定义：

```
Drop table "directory" CASCADE;

Create table "directory"
(
    "id" Serial NOT NULL,
    "root_id" Integer NOT NULL Default 0,
    "name"    Varchar(20)NOT NULL ,
    "status"  boolean Default 'true',
    "created" Timestamp Default current_timestamp,
    "modified" Timestamp Default current_timestamp,
    UNIQUE  (id,root_id),
    PRIMARY KEY ("id")
--     FOREIGN KEY (root_id) REFERENCES directory (id) ON DELETE CASCADE
);
INSERT INTO directory (id,root_id,name) VALUES (0,0,'/');
Alter table "directory" add  FOREIGN KEY (root_id) REFERENCES directory (id) ON DELETE CASCADE;
Create index "directory_index" on "directory" using btree ("id","root_id","name");
```

数据存储状态：

Id	Root_id	Name
0	0	/
1	0	计算机
2	1	显示器
3	1	鼠标
4	1	主板
5	2	Samsung 显示器
6	2	LG 显示器
7	2	SONY 显示器



上图是一个分类目录，当删除子目录时如果子目录中有目录或数据，将删除这些数据和目录

说明：

id 目录根
root_id REFERENCES id ON DELETE CASCADE 当 pk 删除时关联的 fk 自动删除
name 目录名
status 状态 true 可用，false 不可用
created 创建时间
modified 修改时间

注意：

因为使用了关联字段，所以不能在 create table 中使用
FOREIGN KEY (root_id) REFERENCES directory (id) ON DELETE CASCADE
因为插入记录做参考表中的“id”字段，创建表的中没有数据，所以无法插入数据。
先创建表，不定义 FOREIGN KEY，然后初始化插入第一条数据：
INSERT INTO directory (id,root_id,name) VALUES (0,0,'/');
再定义外建：
Alter table "directory" add FOREIGN KEY (root_id) REFERENCES directory (id) ON DELETE CASCADE;

```
postgres=# Create table "directory"
postgres-# (
postgres(#        "id" Serial NOT NULL,
postgres(#        "root_id" Integer NOT NULL Default 0,
postgres(#        "name" Varchar(20)NOT NULL ,
postgres(#        "status"boolean Default 'true',
postgres(#        "created" Timestamp Default current_timestamp,
postgres(#        "modified" Timestamp Default current_timestamp,
postgres(#        UNIQUE (id,root_id),
postgres(#        PRIMARY KEY ("id")
postgres(# --       FOREIGN KEY (root_id) REFERENCES directory (id) ON DELETE CASCADE
postgres(# );
NOTICE:  CREATE TABLE will create implicit sequence 'directory_id_seq' for SERIAL column 'directory.id'
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index 'directory_pkey' for table 'directory'
NOTICE:  CREATE TABLE / UNIQUE will create implicit index 'directory_id_key' for table 'directory'
CREATE TABLE
postgres=# INSERT INTO directory (id,root_id,name) VALUES (0,0,'/');
INSERT 17110 1
postgres=# Alter table "directory" add  FOREIGN KEY (root_id) REFERENCES directory (id) ON DELETE CASCADE;
NOTICE:  ALTER TABLE will create implicit trigger(s) for FOREIGN KEY check(s)
ALTER TABLE
```

```
postgres=# Create index "directory_index" on "directory" using btree ("id","root_id","name");
CREATE INDEX
postgres=# INSERT INTO directory (root_id,name) VALUES (0,'计算机');
INSERT 17116 1
postgres=# SELECT * from directory ;
 id | root_id |      name       | status |            created            |           modified
----+---------+-----------------+--------+-------------------------------+-------------------------------
  0 |       0 | /               | t      | 2003-11-12 16:55:39.727365 | 2003-11-12 16:55:39.727365
  1 |       0 | 计算机          | t      | 2003-11-12 16:56:39.663584 | 2003-11-12 16:56:39.663584
(2 rows)

postgres=# INSERT INTO directory (root_id,name) VALUES (0,'金融');
INSERT 17117 1
postgres=# SELECT * from directory ;
 id | root_id |      name       | status |            created            |           modified
----+---------+-----------------+--------+-------------------------------+-------------------------------
  0 |       0 | /               | t      | 2003-11-12 16:55:39.727365 | 2003-11-12 16:55:39.727365
  1 |       0 | 计算机          | t      | 2003-11-12 16:56:39.663584 | 2003-11-12 16:56:39.663584
  2 |       0 | 金融            | t      | 2003-11-12 16:57:50.509436 | 2003-11-12 16:57:50.509436
(3 rows)

postgres=# INSERT INTO directory (root_id,name) VALUES (1,'显示器');
INSERT 17118 1
postgres=# INSERT INTO directory (root_id,name) VALUES (1,'鼠标');
INSERT 17119 1
postgres=# INSERT INTO directory (root_id,name) VALUES (1,'主板');
INSERT 17120 1
postgres=# SELECT * from directory ;
 id | root_id |      name       | status |            created            |           modified
----+---------+-----------------+--------+-------------------------------+-------------------------------
  0 |       0 | /               | t      | 2003-11-12 16:55:39.727365 | 2003-11-12 16:55:39.727365
  1 |       0 | 计算机          | t      | 2003-11-12 16:56:39.663584 | 2003-11-12 16:56:39.663584
  2 |       0 | 金融            | t      | 2003-11-12 16:57:50.509436 | 2003-11-12 16:57:50.509436
  3 |       1 | 显示器          | t      | 2003-11-12 16:59:15.911196 | 2003-11-12 16:59:15.911196
  4 |       1 | 鼠标            | t      | 2003-11-12 16:59:30.646916 | 2003-11-12 16:59:30.646916
  5 |       1 | 主板            | t      | 2003-11-12 16:59:44.400317 | 2003-11-12 16:59:44.400317
(6 rows)

postgres=# INSERT INTO directory (root_id,name) VALUES (3,'Samsung  显示器');
INSERT 17121 1
postgres=# INSERT INTO directory (root_id,name) VALUES (3,'LG 显示器');
INSERT 17122 1
postgres=# INSERT INTO directory (root_id,name) VALUES (3,'SONY 显示器');
INSERT 17123 1
```

```
postgres=# SELECT * from directory ;
 id | root_id |          name          | status |          created           |          modified
----+---------+------------------------+--------+----------------------------+----------------------------
  0 |       0 | /                      | t      | 2003-11-12 16:55:39.727365 | 2003-11-12 16:55:39.727365
  1 |       0 | 计算机                 | t      | 2003-11-12 16:56:39.663584 | 2003-11-12 16:56:39.663584
  2 |       0 | 金融                   | t      | 2003-11-12 16:57:50.509436 | 2003-11-12 16:57:50.509436
  3 |       1 | 显示器                 | t      | 2003-11-12 16:59:15.911196 | 2003-11-12 16:59:15.911196
  4 |       1 | 鼠标                   | t      | 2003-11-12 16:59:30.646916 | 2003-11-12 16:59:30.646916
  5 |       1 | 主板                   | t      | 2003-11-12 16:59:44.400317 | 2003-11-12 16:59:44.400317
  6 |       3 | Samsung 显示器         | t      | 2003-11-12 17:00:45.964053 | 2003-11-12 17:00:45.964053
  7 |       3 | LG 显示器              | t      | 2003-11-12 17:01:03.736121 | 2003-11-12 17:01:03.736121
  8|        3 | SONY 显示器            | t      | 2003-11-12 17:01:18.257337 | 2003-11-12 17:01:18.257337
(9 rows)

postgres=# INSERT INTO directory (root_id,name) VALUES (7,'CRT 显示器');
INSERT 17124 1
postgres=# INSERT INTO directory (root_id,name) VALUES (7,'液晶显示器');
INSERT 17125 1
postgres=# INSERT INTO directory (root_id,name) VALUES (8,'液晶显示器');
INSERT 17126 1
postgres=# INSERT INTO directory (root_id,name) VALUES (8,'特利隆显示器');
INSERT 17127 1
postgres=# INSERT INTO directory (root_id,name) VALUES (7,'钻石隆显示器');
INSERT 17128 1
postgres=# SELECT * from directory ;
 id | root_id |          name          | status |          created           |          modified
----+---------+------------------------+--------+----------------------------+----------------------------
  0 |       0 | /                      | t      | 2003-11-12 16:55:39.727365 | 2003-11-12 16:55:39.727365
  1 |       0 | 计算机                 | t      | 2003-11-12 16:56:39.663584 | 2003-11-12 16:56:39.663584
  2 |       0 | 金融                   | t      | 2003-11-12 16:57:50.509436 | 2003-11-12 16:57:50.509436
  3 |       1 | 显示器                 | t      | 2003-11-12 16:59:15.911196 | 2003-11-12 16:59:15.911196
  4 |       1 | 鼠标                   | t      | 2003-11-12 16:59:30.646916 | 2003-11-12 16:59:30.646916
  5 |       1 | 主板                   | t      | 2003-11-12 16:59:44.400317 | 2003-11-12 16:59:44.400317
  6 |       3 | Samsung 显示器         | t      | 2003-11-12 17:00:45.964053 | 2003-11-12 17:00:45.964053
  7 |       3 | LG 显示器              | t      | 2003-11-12 17:01:03.736121 | 2003-11-12 17:01:03.736121
  8 |       3 | SONY 显示器            | t      | 2003-11-12 17:01:18.257337 | 2003-11-12 17:01:18.257337
  9 |       7 | CRT 显示器             | t      | 2003-11-12 17:03:05.594891 | 2003-11-12
```

```
17:03:05.594891
 10 |       7 | 液晶显示器     | t      | 2003-11-12 17:03:21.793674 | 2003-11-12 17:03:21.793674
 11 |       8 | 液晶显示器     | t      | 2003-11-12 17:03:30.688531 | 2003-11-12 17:03:30.688531
 12 |       8 | 特利隆显示器 | t      | 2003-11-12 17:03:57.697321 | 2003-11-12 17:03:57.697321
 13 |       7 | 钻石隆显示器 | t      | 2003-11-12 17:04:28.61153  | 2003-11-12 17:04:28.61153
(14 rows)
```

测试：

1. 删除子目录：计算机/显示器/ LG 显示器/ CRT 显示器
   CRT 显示器的 id 是 9
   SQL：DELETE FROM directory WHERE id=9;

```
postgres=# DELETE FROM directory WHERE id=9;
DELETE 1
postgres=# SELECT * from directory ;
 id | root_id |          name          | status |            created            |           modified
----+---------+------------------------+--------+-------------------------------+-------------------------------
  0 |       0 | /                      | t      | 2003-11-12 16:55:39.727365 | 2003-11-12
16:55:39.727365
  1 |       0 | 计算机                 | t      | 2003-11-12 16:56:39.663584 | 2003-11-12 16:56:39.663584
  2 |       0 | 金融                   | t      | 2003-11-12 16:57:50.509436 | 2003-11-12
16:57:50.509436
  3 |       1 | 显示器                 | t      | 2003-11-12 16:59:15.911196 | 2003-11-12 16:59:15.911196
  4 |       1 | 鼠标                   | t      | 2003-11-12 16:59:30.646916 | 2003-11-12
16:59:30.646916
  5 |       1 | 主板                   | t      | 2003-11-12 16:59:44.400317 | 2003-11-12
16:59:44.400317
  6 |       3 | Samsung 显示器   | t      | 2003-11-12 17:00:45.964053 | 2003-11-12 17:00:45.964053
  7 |       3 | LG 显示器              | t      | 2003-11-12 17:01:03.736121 | 2003-11-12
17:01:03.736121
  8 |       3 | SONY 显示器            | t      | 2003-11-12 17:01:18.257337 | 2003-11-12
17:01:18.257337
 10 |       7 | 液晶显示器     | t      | 2003-11-12 17:03:21.793674 | 2003-11-12 17:03:21.793674
 11 |       8 | 液晶显示器     | t      | 2003-11-12 17:03:30.688531 | 2003-11-12 17:03:30.688531
 12 |       8 | 特利隆显示器 | t      | 2003-11-12 17:03:57.697321 | 2003-11-12 17:03:57.697321
 13 |       7 | 钻石隆显示器 | t      | 2003-11-12 17:04:28.61153  | 2003-11-12 17:04:28.61153
(13 rows)

postgres=#
```

2. 删除子目录：计算机/显示器/ LG 显示器
   LG 显示器目录下的子目录：液晶显示器、钻石隆显示器也将被删除

```
postgres=# DELETE FROM directory WHERE id=7;
DELETE 1
postgres=# SELECT * from directory ;
 id | root_id |          name          | status |            created            |           modified
```

```
----+---------+-------------------+--------+---------------------------+----------------------------
  0 |       0 | /                 | t      | 2003-11-12 16:55:39.727365 | 2003-11-12
16:55:39.727365
  1 |       0 | 计算机            | t      | 2003-11-12 16:56:39.663584 | 2003-11-12 16:56:39.663584
  2 |       0 | 金融              | t      | 2003-11-12 16:57:50.509436 | 2003-11-12
16:57:50.509436
  3 |       1 | 显示器            | t      | 2003-11-12 16:59:15.911196 | 2003-11-12 16:59:15.911196
  4 |       1 | 鼠标              | t      | 2003-11-12 16:59:30.646916 | 2003-11-12
16:59:30.646916
  5 |       1 | 主板              | t      | 2003-11-12 16:59:44.400317 | 2003-11-12
16:59:44.400317
  6 |       3 | Samsung 显示器    | t      | 2003-11-12 17:00:45.964053 | 2003-11-12 17:00:45.964053
  8 |       3 | SONY 显示器       | t      | 2003-11-12 17:01:18.257337 | 2003-11-12
17:01:18.257337
 11 |       8 | 液晶显示器        | t      | 2003-11-12 17:03:30.688531 | 2003-11-12 17:03:30.688531
 12 |       8 | 特利隆显示器      | t      | 2003-11-12 17:03:57.697321 | 2003-11-12 17:03:57.697321
(10 rows)
```

3. 再删除：计算机/显示器/ SONY 显示器

```
postgres=# DELETE FROM directory WHERE id=8;
DELETE 1
postgres=# SELECT * from directory ;
 id | root_id |       name        | status |          created           |          modified
----+---------+-------------------+--------+----------------------------+----------------------------
  0 |       0 | /                 | t      | 2003-11-12 16:55:39.727365 | 2003-11-12
16:55:39.727365
  1 |       0 | 计算机            | t      | 2003-11-12 16:56:39.663584 | 2003-11-12 16:56:39.663584
  2 |       0 | 金融              | t      | 2003-11-12 16:57:50.509436 | 2003-11-12 16:57:50.509436
  3 |       1 | 显示器            | t      | 2003-11-12 16:59:15.911196 | 2003-11-12 16:59:15.911196
  4 |       1 | 鼠标              | t      | 2003-11-12 16:59:30.646916 | 2003-11-12 16:59:30.646916
  5 |       1 | 主板              | t      | 2003-11-12 16:59:44.400317 | 2003-11-12 16:59:44.400317
  6 |       3 | Samsung 显示器    | t      | 2003-11-12 17:00:45.964053 | 2003-11-12 17:00:45.964053
(7 rows)
```

4. 删除子目录：计算机/显示器

   显示器目录下的子目录：

   下有目录 LG 显示器/ CRT 显示器、SONY 显示器/……、LG 显示器/……

   删除显示器目录后，下的所有子目录将被删除。

```
postgres=# INSERT INTO directory (root_id,name) VALUES (3,'LG 显示器');
INSERT 17129 1
postgres=# INSERT INTO directory (root_id,name) VALUES (3,'SONY 显示器');
INSERT 17130 1
postgres=# INSERT INTO directory (root_id,name) VALUES (6,'CRT 显示器');
```

```
INSERT 17131 1
postgres=# INSERT INTO directory (root_id,name) VALUES (14,'CRT 显示器');
INSERT 17132 1
postgres=# INSERT INTO directory (root_id,name) VALUES (15,'CRT 显示器');
INSERT 17133 1
postgres=# INSERT INTO directory (root_id,name) VALUES (15,'特利隆显示器');
INSERT 17134 1
postgres=# INSERT INTO directory (root_id,name) VALUES (15,'钻石隆显示器');
INSERT 17135 1
postgres=# INSERT INTO directory (root_id,name) VALUES (6,'液晶显示器');
INSERT 17136 1
postgres=# INSERT INTO directory (root_id,name) VALUES (14,'液晶显示器');
INSERT 17137 1
postgres=# INSERT INTO directory (root_id,name) VALUES (15,'液晶显示器');
INSERT 17138 1
postgres=# SELECT * from directory ;
 id | root_id |            name            | status |            created            |           modified
----+---------+----------------------------+--------+-------------------------------+-------------------------------
  0 |       0 | /                          | t      | 2003-11-12 16:55:39.727365    | 2003-11-12 16:55:39.727365
  1 |       0 | 计算机                     | t      | 2003-11-12 16:56:39.663584    | 2003-11-12 16:56:39.663584
  2 |       0 | 金融                       | t      | 2003-11-12 16:57:50.509436    | 2003-11-12 16:57:50.509436
  3 |       1 | 显示器                     | t      | 2003-11-12 16:59:15.911196    | 2003-11-12 16:59:15.911196
  4 |       1 | 鼠标                       | t      | 2003-11-12 16:59:30.646916    | 2003-11-12 16:59:30.646916
  5 |       1 | 主板                       | t      | 2003-11-12 16:59:44.400317    | 2003-11-12 16:59:44.400317
  6 |       3 | Samsung 显示器             | t      | 2003-11-12 17:00:45.964053    | 2003-11-12 17:00:45.964053
 14 |       3 | LG 显示器                  | t      | 2003-11-12 17:28:03.927651    | 2003-11-12 17:28:03.927651
 15 |       3 | SONY 显示器                | t      | 2003-11-12 17:28:15.235316    | 2003-11-12 17:28:15.235316
 16 |       6 | CRT 显示器                 | t      | 2003-11-12 17:28:49.586084    | 2003-11-12 17:28:49.586084
 17 |      14 | CRT 显示器                 | t      | 2003-11-12 17:28:55.290861    | 2003-11-12 17:28:55.290861
 18 |      15 | CRT 显示器                 | t      | 2003-11-12 17:28:59.731191    | 2003-11-12 17:28:59.731191
 19 |      15 | 特利隆显示器               | t      | 2003-11-12 17:29:10.747115    | 2003-11-12 17:29:10.747115
 20 |      15 | 钻石隆显示器               | t      | 2003-11-12 17:29:30.770079    | 2003-11-12 17:29:30.770079
 21 |       6 | 液晶显示器                 | t      | 2003-11-12 17:29:47.006177    | 2003-11-12 17:29:47.006177
 22 |      14 | 液晶显示器                 | t      | 2003-11-12 17:29:51.904914    | 2003-11-12 17:29:51.904914
 23 |      15 | 液晶显示器                 | t      | 2003-11-12 17:29:57.355213    | 2003-11-12 17:29:57.355213
```

```
(17 rows)

postgres=# DELETE FROM directory WHERE id=3;
DELETE 1
postgres=# SELECT * from directory ;
 id | root_id |      name       | status |            created            |           modified
----+---------+-----------------+--------+-------------------------------+-------------------------------
  0 |       0 | /               | t      | 2003-11-12 16:55:39.727365 | 2003-11-12 16:55:39.727365
  1 |       0 | 计算机          | t      | 2003-11-12 16:56:39.663584 | 2003-11-12 16:56:39.663584
  2 |       0 | 金融            | t      | 2003-11-12 16:57:50.509436 | 2003-11-12 16:57:50.509436
  4 |       1 | 鼠标            | t      | 2003-11-12 16:59:30.646916 | 2003-11-12 16:59:30.646916
  5 |       1 | 主板            | t      | 2003-11-12 16:59:44.400317 | 2003-11-12 16:59:44.400317
(5 rows)
```

不再举例了，删除 id=0 将删除计算机包括下面的所有目录被删除。
注意，千万不要删除 id=0。

3.9.5 总结

分类目录的例子中使用了 ON DELETE CASCADE，方便了操作，但也有危险。如果不用 ON DELETE CASCADE 而用程序来实现，需要使用递归算法，非常麻烦。

3.10 模式

一些用户为了使某些模块的表看起来清晰，一般他们采用“模块名_表名”：
Auth_user
Auth_group
Bbs_topic
Bbs_message

PostgreSQL 不必这样命名，可以使用 Schema（模式）如：
Auth.user
Auth.group
Bbs.topic
Bbs.message

3.10.1 创建模式

CREATE SCHEMA your_schema;
例：
CREATE SCHEMA btob;
CREATE SCHEMA auction;

3.10.2 删除模式

DROP SCHEMA your_schema;
删除模式，并且同时删除模式下的（表，视图，触发器，过程……）
DROP SCHEMA your_schema CASCADE;
例：
DROP SCHEMA btob CASCADE;
DROP SCHEMA btob CASCADE;

3.10.3 模式搜索路径

查看当前模式 SHOW search_path ;

```
netkiller=> SHOW search_path ;
 search_path
--------------
 $user,public
(1 row)
netkiller=> \dt
                List of relations
 Schema |       Name         | Type   |     Owner
--------+--------------+-------+-----------
 public | company          | table | netkiller
 public | group            | table | netkiller
 public | groupmember | table | netkiller
 public | guestbook      | table | netkiller
 public | prodorder      | table | netkiller
 public | role               | table | netkiller
 public | rolemember    | table | netkiller
 public | system_log    | table | netkiller
 public | templates      | table | netkiller
 public | trust              | table | netkiller
 public | user               | table | netkiller
 public | user_log        | table | netkiller
 public | userinfo        | table | netkiller
(13 rows)
```

如果不设置模式搜索路径，“\dt”只显示 public 模式下的表。
设置模式 SET search_path TO public,btob,auction;

```
netkiller=> SET search_path TO public,btob,auction;
SET
netkiller=> \dt
                List of relations
 Schema   |       Name         | Type   |     Owner
---------+--------------+-------+-----------
```

```
 auction | messages          | table | netkiller
 auction | product           | table | netkiller
 auction | product_order | table | netkiller
 btob    | directory         | table | netkiller
 btob    | trade             | table | netkiller
 btob    | trade_message | table | netkiller
 public  | company           | table | netkiller
 public  | group             | table | netkiller
 public  | groupmember       | table | netkiller
 public  | guestbook         | table | netkiller
 public  | prodorder         | table | netkiller
 public  | role              | table | netkiller
 public  | rolemember        | table | netkiller
 public  | system_log        | table | netkiller
 public  | templates         | table | netkiller
 public  | trust             | table | netkiller
 public  | user              | table | netkiller
 public  | user_log          | table | netkiller
 public  | userinfo          | table | netkiller
(19 rows)

netkiller=>
```

```
-- ================================================================================
-- 'btob.directory'
-- ================================================================================
Drop table btob.directory CASCADE;

Create table btob.directory
(
        "id" Serial NOT NULL,
        "root_id" Integer NOT NULL Default 0,
        "name"      Varchar(20)NOT NULL ,
        "status"    boolean Default 'true',
        "created" Timestamp Default current_timestamp,
        "modified" Timestamp Default current_timestamp,
        UNIQUE   (id,root_id,name),
        PRIMARY KEY ("id")
--        FOREIGN KEY (root_id) REFERENCES directory (id) ON DELETE CASCADE
);
INSERT INTO btob.directory (id,root_id,name) VALUES (0,0,'/');
Alter table btob.directory add    FOREIGN KEY (root_id) REFERENCES btob.directory (id) ON DELETE
CASCADE;
Create index "directory_index" on btob.directory using btree ("id","root_id","name");
```

4 实体关系（Entity-Relation）

在关系数据库中，关系能防止冗余的数据。例如，如果正在设计一个数据库来跟踪有关书的信息，而每本书的信息（如书名、出版日期和出版商）都保存在一个名为 titles 的表中。同时还有一些想保存的有关出版商的信息，例如出版商的电话号码、地址和邮政编码。如果将所有这些信息都保存在 titles 表中，则对于某个出版商出版的每本书，出版商的电话号码将是重复的。

一个更好的解决方案是，单独在一个名为 publishers 的表中只保存一次出版商信息。然后在 titles 表中设置指针，以引用 publishers 表中的项。

若要确保数据同步，可以在 titles 表和 publishers 表之间强制引用完整性。引用完整性关系能确保某个表中的信息与另一个表中的信息相匹配。例如，titles 表中的每个书名必须和 publishers 表的特定出版商相关联。不能在数据库中添加数据库中不存在的出版商的书名。

为更好地理解表关系，请参见：

- 表关系类型
  表与表之间存在三种类型的关系。所创建的关系类型取决于相关联的列是如何定义的。
  - 一对多关系
  - 多对多关系
  - 一对一关系
- 引用完整性概述
- 表关系类型

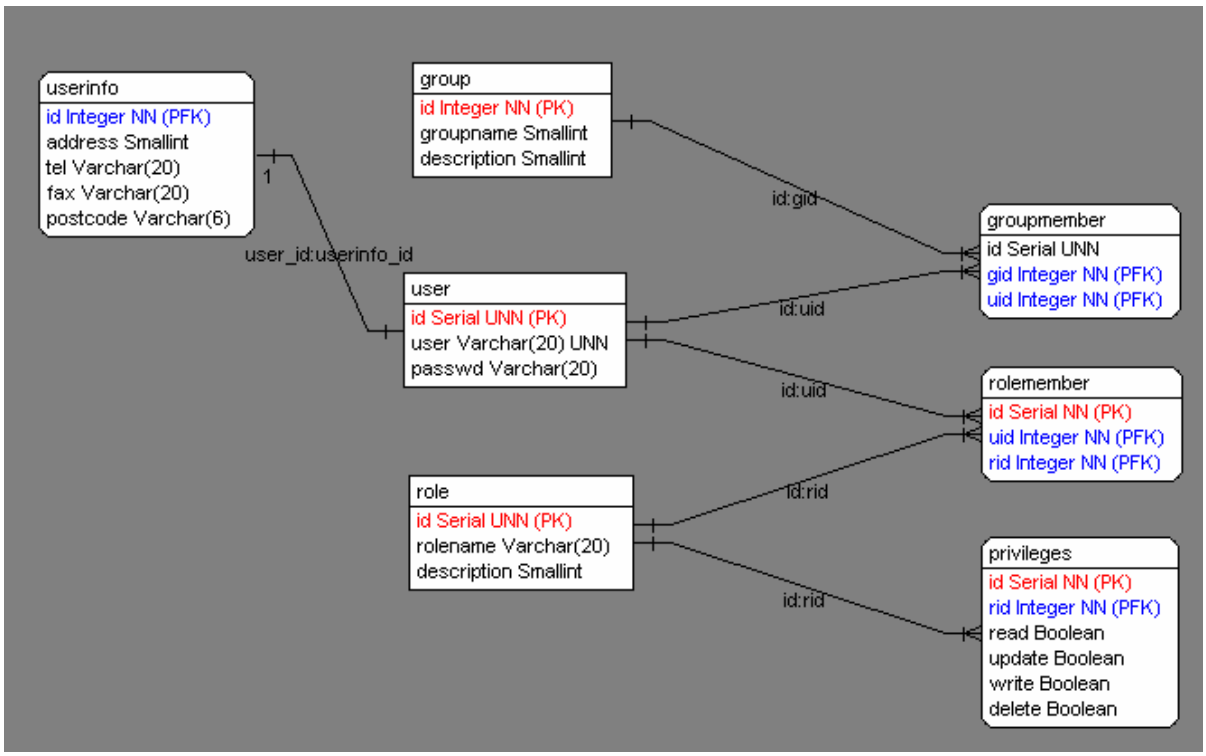
关系是通过匹配键列中的数据而工作的，而键列通常是两个表中具有相同名称的列。在大多数情况下，关系将一个表中为每个行提供唯一标识符的主键与另一个表中外键内的项相匹配。例如，通过在 titles 表的 title_id 列（主键）和 sales 表的 title_id 列（外键）之间创建一个关系，可以使销售额与特定的销售书名相关联。

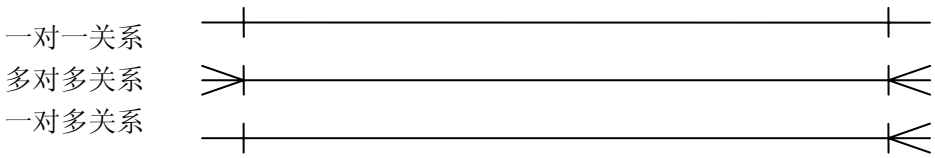
4.1 E-R 图（Entity-Relation）

PostgreSQL 本身没有 GUI（图形用户界面）管理工具，其它 GUI 工具也没有 E-R 图功能，如 pgAdmin III。这里我给在各位介绍一个很好的数据库设计工具“CASE Studio 2”，

1. 实体关系设计
2. 能过 E-R 图产生 SQL（DDL）脚本
3. 已经存在数据库的逆向工程
4. 产生非常详细的 HTML 和 RTF 报告
5. 用户定义 Add-ins 和模板
6. 版本管理，Galery,To-Do 列表

7. 数据流程设计
8. 界面见附录。





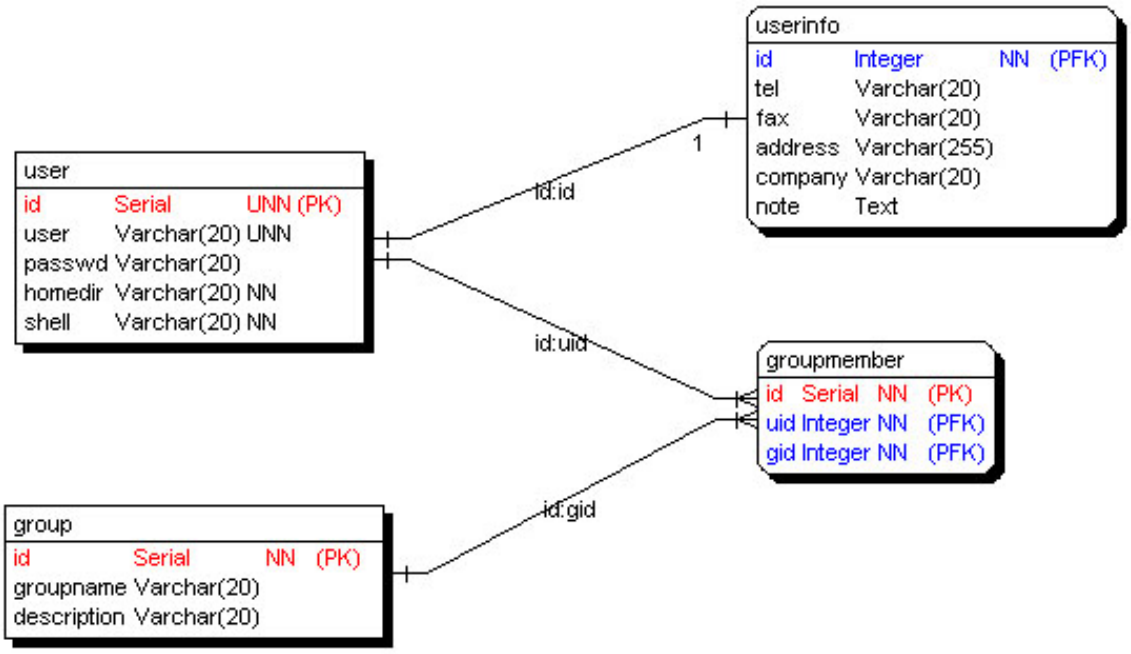
4.2 一对多关系

一对多关系是最常见的关系类型。在这种关系类型中，表 A 中的行可以在表 B 中有许多匹配行，但是表 B 中的行只能在表 A 中有一个匹配行。

例如，publishers 表和 titles 表是一对多的关系：每一个出版商可出版许多书，但每一本书只能有一个出版商。

如果在相关列中只有一列是主键或具有唯一约束，则创建的是一对多关系。

一对多关系中的主键方由一个键 符号表示。关系中的外键方由一个无穷大 符号表示。




```
Drop table "groupmember" Restrict;
Drop table "userinfo" Restrict;
Drop table "group" Restrict;
Drop table "user" Restrict;

Create table "user"
(
	"id" Serial NOT NULL Default 0 UNIQUE ,
	"user" Varchar(20) NOT NULL UNIQUE ,
	"passwd" Varchar(20),
	"homedir" Varchar(20) NOT NULL Default /home/,
	"shell" Varchar(20) NOT NULL Default /bin/bash,
  primary key ("id")
);

Create table "group"
(
	"id" Serial NOT NULL Default 0,
	"groupname" Varchar(20),
	"description" Varchar(20),
  primary key ("id")
);

Create table "userinfo"
(
	"id" integer NOT NULL Default 0,
	"tel" Varchar(20),
```

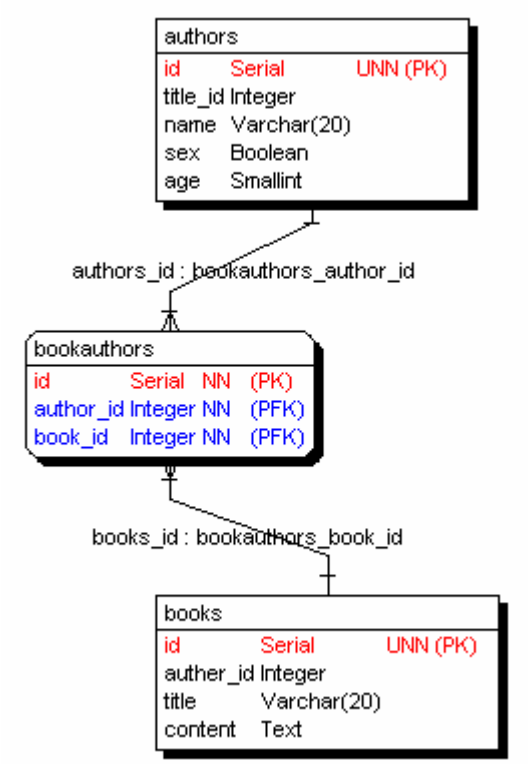
```
        "fax" Varchar(20),
        "address" Varchar(255),
        "company" Varchar(20),
        "note" Text,
  primary key ("id")
);

Create table "groupmember"
(
        "id" Serial NOT NULL,
        "uid" integer NOT NULL,
        "gid" integer NOT NULL,
  primary key ("id","uid","gid")
);

Alter table "userinfo" add    foreign key ("id") references "user" ("id") on update restrict on delete restrict;
Alter table "groupmember" add    foreign key ("uid") references "user" ("id") on update restrict on delete restrict;
Alter table "groupmember" add    foreign key ("gid") references "group" ("id") on update restrict on delete restrict;
```

4.3 多对多关系

在多对多关系中，表 A 中的一行可与表 B 中的多行相匹配，反之亦然。通过定义称为连接表的第三方表创建这样的关系，该连接表的主键包括表 A 和表 B 中的外键。

例如，authors 表和 books 表是多对多关系，该关系通过从这些表中的每个表与 bookauthors 表的一对多关系定义。bookauthors 表的主键由 author_id 列（authors 表的主键）和 book_id 列（books 表的主键）组成。

注意：Case Studio 2 产生的 SQL（DDL）脚本并不完全正确。所以还要加以修改才可以使用。




```
Drop table "bookauthors" Restrict;
Drop table "books" Restrict;
Drop table "authors" Restrict;

Create table "authors"
(
	"id" Serial NOT NULL UNIQUE ,
	"title_id" integer,
	"name" Varchar(20),
	"sex" Boolean,
	"age" Smallint,
  primary key ("id")
);

Create table "books"
(
	"id" Serial NOT NULL UNIQUE ,
	"auther_id" integer,
	"title" Varchar(20),
	"content" Text,
  primary key ("id")
);

Create table "bookauthors"
(
```

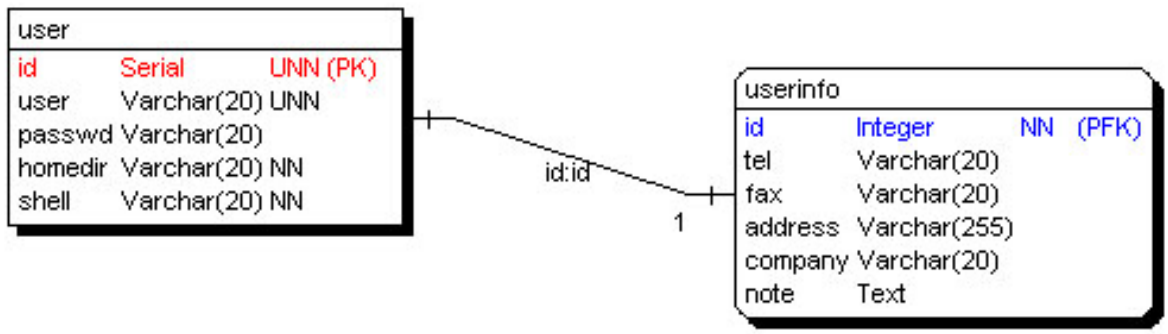
```
    "id" Serial NOT NULL,
    "author_id" integer NOT NULL,
    "book_id" integer NOT NULL,
  primary key ("id","author_id","book_id")
改为
primary key ("id")

);


Alter table "bookauthors" add   foreign key ("author_id") references "authors" ("id") on update restrict on delete restrict;
Alter table "bookauthors" add   foreign key ("book_id") references "books" ("id") on update restrict on delete restrict;
```

4.4 一对一关系

在一对一关系中，表 A 中的一行最多只能与表 B 中的一行相匹配，反之亦然。如果两个相关列都是主键或具有唯一约束，则创建的是一对一关系。

这种关系不常见，因为这种方式的大部分相关信息都在一个表中。使用一对一关系可以是为了：

- 分割一个含有许多列的表。
- 出于安全考虑而隔离表的某一部分。
- 存储可以很容易删除的临时数据，只需删除表即可删除这些数据。
- 存储只应用于主表子集的信息。
- 一对一关系的主键方由键 符号表示。外键方也由键 符号表示。




```
Drop table "userinfo" Restrict;
Drop table "user" Restrict;

Create table "user"
(
    "id" Serial NOT NULL Default 0 UNIQUE ,
    "user" Varchar(20) NOT NULL UNIQUE ,
    "passwd" Varchar(20),
```

```
    "homedir" Varchar(20) NOT NULL Default /home/,
    "shell" Varchar(20) NOT NULL Default /bin/bash,
  primary key ("id")
);


Create table "userinfo"
(
    "id" integer NOT NULL Default 0,
    "tel" Varchar(20),
    "fax" Varchar(20),
    "address" Varchar(255),
    "company" Varchar(20),
    "note" Text,
  primary key ("id")
);

Alter table "userinfo" add    foreign key ("id") references "user" ("id") on update restrict on delete restrict;
```

4.5 引用完整性

引用完整性是一种规则系统，这些规则可确保相关表中各行间关系的有效性，并确保不会意外删除或更改相关的数据。

在强制引用完整性时必须遵循以下规则：

如果在相关表的主键中不存在某个值，则不能在相关表的外键列中输入该值。但是，可以在外键列中输入空值。例如，在 employee 表中没有包括某职员，则不能指明分配给该职员的工作，但是可在 employee 表的 job_id 列输入空值来指明没有给该职员分配工作。

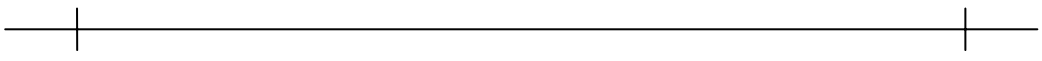
如果在相关表中存在与某行匹配的行，则不能从主表中删除该行。例如，如果在 employee 表中给多个职员分配了由 jobs 表中某行所代表的工作时，则不能删除该行。

当主表的某行有相关行时，则不能更改主键值。例如，如果将 jobs 表中的一项工作分配给某职员，则不能从 employee 表中删除该职员。

当满足下述所有条件时，可以设置引用完整性：

1. 主表中相匹配的列是主键或具有唯一约束 。
2. 相关列具有相同的数据类型和长度。
3. 两个表属于同一个数据库。
4. 数据库关系图中的已强制关系和未强制关系

在数据库关系图中创建关系线将在相关表上创建外键约束，从而自动强制引用完整性。在数据库关系图中，已强制关系用实线表示。例如：



在关系图中，未强制关系用虚线表示，这种关系的外键约束被禁用。例如：

-----┼---┼----

5 视图

我发现很多程序员开发数据应该程序。SQL 语句写了五六行，每行最少 100 个字符。好比如下：
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA
这么一大堆 !@%%#@%@^#$&$&%*&^*%^(&)(*) faint 写这么长主要的操作是“，”号 join 操作。

我的习惯是先使用 VIEW 做第一次的过滤，然后在做第二次查询。最后的操作基本上都是 Select * from table where xxx = xxxx order by xxxx ASC limit x offset x 操作读出想要的记录。

5.1 VIEW 基本使用实例

这是一个最基本用法。这里主要用于格式化 Timestamp 输出 2003-09-24 11:23:29 格式。CASE WHEN u.active=true THEN 'Y' ELSE 'N' END as "active" ，SQL CASE 表达式是一种通用的 条件表达式，类似于其它语言中的 if/else 语句。

```
-- ============================================================
--   'user'
-- ============================================================
Create table "user"
(
    "id" Serial NOT NULL,
    "userid" Varchar(50) NOT NULL,
    "passwd" Varchar(50),
    "name"     Varchar(20)NOT NULL ,
    "nickname" Varchar(20)NOT NULL ,
    "active" Boolean Default 'F',
    "email"     Varchar(50) NOT NULL,
    "question" Varchar(255) NOT NULL,
    "answer" Varchar(255) NOT NULL,
    "begin_date" Timestamp Default now(),
    "end_date" Timestamp Default now(),
    UNIQUE  (userid,email),
```

```
  primary key ("id")
);
Create index "user_index" on "user" using btree ("id","userid");

-- ----------------------------------------------------
--    'vuser'
-- ----------------------------------------------------
drop view vuser;
CREATE VIEW vuser AS
       SELECT u.id,u.userid,u."name",u.nickname,
              CASE WHEN u.active=true THEN 'Y' ELSE 'N' END as "active",
              u.email,u.question,u.answer,
              to_char(u.begin_date,'YYYY-MM-DD HH:MI:SS') as begin_date,
              to_char(u.end_date,'YYYY-MM-DD HH:MI:SS') as end_date
       FROM "user" u
       Order By u.id;
```

5.2 使用 HTML 格式化 VIEW 的实例

```
-- ======================================================
--    'user'
-- ======================================================
Create table "user"
(
       "id" Serial NOT NULL,
       "userid" Varchar(50) NOT NULL,
       "passwd" Varchar(50),
       "name"      Varchar(20)NOT NULL ,
       "nickname" Varchar(20)NOT NULL ,
       "active" Boolean Default 'F',
       "email"      Varchar(50) NOT NULL,
       "question" Varchar(255) NOT NULL,
       "answer" Varchar(255) NOT NULL,
       "begin_date" Timestamp Default now(),
       "end_date" Timestamp Default now(),
       UNIQUE   (userid,email),
  primary key ("id")
);
Create index "user_index" on "user" using btree ("id","userid");

-- ----------------------------------------------------
--    'vuser'
-- ----------------------------------------------------
drop view vuser1;
```

```
CREATE OR REPLACE VIEW vuser1 AS
    SELECT '<center>'||u.id||'</center>' as ID,'<font color=red>'||u.userid||'</font>' as "用户名",u."name"
as "姓名",u.nickname as "昵称",
        CASE WHEN u.active=true THEN '启用' ELSE '禁用' END as "active",
        u.email,u.question,u.answer,
        to_char(u.begin_date,'YYYY 年 MM 月 DD 日  HH:MI:SS') as begin_date,
        to_char(u.end_date,'YYYY 年 MM 月 DD HH 时 MI 分 SS 秒') as end_date
    FROM "user" u
    Order By u.id;

member=> select * from vuser1;
          id          |          用户名          | 性名 | 昵称 | active |       email       |
question | answer |          begin_date          |          end_date
--------------------+----------------------------+--------+--------+--------+---------------+----------+--------+---------
-------------------+--------------------------------
 <center>1</center> | <font color=red>sysop</font> | chen    | chen    | 启用 | chen@chen.com |
xxxxxxx  | xxxx    | 2003 年 09 月 24 日  11:23:29 | 2003 年 09 月 24 11 时 23 分 29 秒
 <center>2</center> | <font color=red>admin</font> | chen    | chen    | 禁用 | chen@chen.com |
xxxxxxx  | xxxx    | 2003 年 09 月 24 日  11:23:29 | 2003 年 09 月 30 11 时 23 分 29 秒

下面有一下更大胆的用法。
-- ========================================================
--   'group'
-- ========================================================
Create table "group"
(
    "id" Serial NOT NULL UNIQUE,
    "groupname" Varchar(20) NOT NULL,
    "description" Varchar(255),
    UNIQUE  (groupname),
    PRIMARY KEY ("id")
);
Create index "group_index" on "group" using btree ("id","groupname");
DROP VIEW vgroup;
CREATE VIEW vgroup AS
    SELECT '<tr>'||g.id||'</td>' as id,'<tr>'||g.groupname||'</td>' as groupname,'<tr>'||g.description||'</td>'
as desc
        FROM "group" g
    ORDER BY g.id;
```

```
postgres=# DROP VIEW vgroup;
ERROR:    view "vgroup" does not exist
postgres=# CREATE VIEW vgroup AS
```

```
postgres-#                    SELECT    '<tr>'||g.id||'</td>'    as    id,'<tr>'||g.groupname||'</td>'    as
groupname,'<tr>'||g.description||'</td>' as desc
postgres-#          FROM "group" g
postgres-#          ORDER BY g.id;
CREATE VIEW
postgres=# select * from vgroup ;
         id          |       groupname       |                    desc
-------------+-----------------+---------------------------
  <tr>1</td>    | <tr>Admin</td>    | <tr>xxxxxxxxxxxxxxxxx</td>
  <tr>2</td>    | <tr>Guest</td>    | <tr>xxxxxxxxxxxxxxxxx</td>
  <tr>3</td>    | <tr>Domain</td> | <tr>xxxxxxxxxxxxxxxxx</td>
  <tr>4</td>    | <tr>Admin</td>    | <tr>xxxxxxxxxxxxxxxxx</td>
  <tr>5</td>    | <tr>Guest</td>    | <tr>xxxxxxxxxxxxxxxxx</td>
  <tr>6</td>    | <tr>Domain</td> | <tr>xxxxxxxxxxxxxxxxx</td>
  <tr>7</td>    | <tr>Admin</td>    | <tr>xxxxxxxxxxxxxxxxx</td>
  <tr>8</td>    | <tr>Guest</td>    | <tr>xxxxxxxxxxxxxxxxx</td>
  <tr>9</td>    | <tr>Domain</td> | <tr>xxxxxxxxxxxxxxxxx</td>
  <tr>10</td> | <tr>Admin</td>    | <tr>xxxxxxxxxxxxxxxxx</td>
  <tr>11</td> | <tr>Guest</td>    | <tr>xxxxxxxxxxxxxxxxx</td>
  <tr>12</td> | <tr>Domain</td> | <tr>xxxxxxxxxxxxxxxxx</td>
  <tr>13</td> | <tr>Admin</td>    | <tr>xxxxxxxxxxxxxxxxx</td>
  <tr>14</td> | <tr>Guest</td>    | <tr>xxxxxxxxxxxxxxxxx</td>
  <tr>15</td> | <tr>Domain</td> | <tr>xxxxxxxxxxxxxxxxx</td>
(15 rows)
```

进一步修改上面的 VIEW，使它更方便输出。

```
DROP VIEW vgroup;
CREATE VIEW vgroup AS
        SELECT '<tr align=center>'||g.id||'</td>' as id,
        '<tr align=left><font color=red><b>'||g.groupname||'</b></font></td>' as groupname,
        '<tr><i>'||g.description||'</i></td>' as desc
        FROM "group" g
        ORDER BY g.id;
```

```
postgres=# select * from vgroup ;
                  id                  |                                                groupname
|                        desc
--------------------------+-----------------------------------------------------------+---------------------------------
  <tr   align=center>1</td>      |  <tr   align=left><font   color=red><b>Admin</b></font></td>      |
<tr><i>xxxxxxxxxxxxxxxxx</i></td>
  <tr   align=center>2</td>      |  <tr   align=left><font   color=red><b>Guest</b></font></td>      |
<tr><i>xxxxxxxxxxxxxxxxx</i></td>
  <tr   align=center>3</td>      |  <tr   align=left><font   color=red><b>Domain</b></font></td>   |
<tr><i>xxxxxxxxxxxxxxxxx</i></td>
  <tr   align=center>4</td>      |  <tr   align=left><font   color=red><b>Admin</b></font></td>      |
```

```
<tr><i>xxxxxxxxxxxxxxxxx</i></td>
 <tr  align=center>5</td>    |  <tr  align=left><font  color=red><b>Guest</b></font></td>    |
<tr><i>xxxxxxxxxxxxxxxxx</i></td>
 <tr  align=center>6</td>    |  <tr  align=left><font  color=red><b>Domain</b></font></td>  |
<tr><i>xxxxxxxxxxxxxxxxx</i></td>
 <tr  align=center>7</td>    |  <tr  align=left><font  color=red><b>Admin</b></font></td>    |
<tr><i>xxxxxxxxxxxxxxxxx</i></td>
 <tr  align=center>8</td>    |  <tr  align=left><font  color=red><b>Guest</b></font></td>    |
<tr><i>xxxxxxxxxxxxxxxxx</i></td>
 <tr  align=center>9</td>    |  <tr  align=left><font  color=red><b>Domain</b></font></td>  |
<tr><i>xxxxxxxxxxxxxxxxx</i></td>
 <tr  align=center>10</td>  |  <tr  align=left><font  color=red><b>Admin</b></font></td>    |
<tr><i>xxxxxxxxxxxxxxxxx</i></td>
 <tr  align=center>11</td>  |  <tr  align=left><font  color=red><b>Guest</b></font></td>    |
<tr><i>xxxxxxxxxxxxxxxxx</i></td>
 <tr  align=center>12</td>  |  <tr  align=left><font  color=red><b>Domain</b></font></td>  |
<tr><i>xxxxxxxxxxxxxxxxx</i></td>
 <tr  align=center>13</td>  |  <tr  align=left><font  color=red><b>Admin</b></font></td>    |
<tr><i>xxxxxxxxxxxxxxxxx</i></td>
 <tr  align=center>14</td>  |  <tr  align=left><font  color=red><b>Guest</b></font></td>    |
<tr><i>xxxxxxxxxxxxxxxxx</i></td>
 <tr  align=center>15</td>  |  <tr  align=left><font  color=red><b>Domain</b></font></td>  |
<tr><i>xxxxxxxxxxxxxxxxx</i></td>
(15 rows)
```

上面的例子输出数据中代有 HTML 标记，在 B/S 结构程序的开发中很方便。当输出数据要改变风格时，只要 CREATE OR REPLACE VIEW your_view AS 就可以，面不必关心页面与程序。

5.3 view 中使用汉字做字段名

下面是一个使用汉字做字段名的例子，输出类似表格：

ID	组名	描述
3	Backup Admin	系统管理员
2	Power Admin	系统管理员
1	System Admin	系统管理员

```
ID |      组名      |        描述
----+--------------+-----------------
   3 | Backup Admin |  系统管理员
   2 | Power Admin   |  系统管理员
   1 | System Admin |  系统管理员
```

```
DROP VIEW vgroup;
```

```
CREATE VIEW vgroup AS
    SELECT g.id as "ID",g.groupname as "组名",g.description as "描述"
    FROM "group" g
    ORDER BY g.groupname;
```

```
postgres=# CREATE VIEW vgroup AS
postgres-#     SELECT g.id as "ID",g.groupname as "组名",g.description as "描述"
postgres-#     FROM "group" g
postgres-#     ORDER BY g.groupname;
CREATE VIEW

postgres=# \d vgroup
                View "public.vgroup"

 Column |                Type                | Modifiers
--------+------------------------+-----------
 ID     | integer                            |
 组名   | character varying(20)  |
 描述   | character varying(255) |
View definition: SELECT g.id AS "ID", g.groupname AS "组名", g.description AS "描述" FROM "group"
g ORDER BY g.groupname;

postgres=# insert into "group"(groupname,description) values('System Admin','系统管理员');
INSERT 35031 1
postgres=# insert into "group"(groupname,description) values('Power Admin','系统管理员');
INSERT 35032 1
postgres=# insert into "group"(groupname,description) values('Backup Admin','系统管理员');
INSERT 35033 1
postgres=# select * from vgroup ;
 ID |      组名      |       描述
----+--------------+-----------------
  3 | Backup Admin |  系统管理员
  2 | Power Admin  |  系统管理员
  1 | System Admin |  系统管理员

(3 rows)

postgres=#
```

5.4 取出字符如果超过 20 个在后尾加“…”

实现方法：

1. 首先在取字符长度，可以使用 character_length()或 char_length()函数，它们的功能是一样的。
2. 然后判断字符是否大于 20 个。

3. 大于 20 个字符，使用 substring()函数截取前 20 个字符。并在后尾加上“…”
4. 如果小于 20 个字符，直接取出
5. 要注意数据转换。使用::varchar(长度)

实例：

```
DROP SCHEMA oa CASCADE;
CREATE SCHEMA oa;
SET search_path TO public,btob,btoc,ctoc,oa;
-- ============================================================================
-- 'oa.meeting'
-- ============================================================================
Drop table oa.meeting CASCADE;

Create table oa.meeting
(
    "id"    Serial NOT NULL UNIQUE,
    "subject"    varchar(100) NOT NULL,
    "caller"     Varchar(10) NOT NULL ,
    "begin_time" Timestamp Default current_timestamp::timestamp (0) without time zone,
    "end_time"       Timestamp Default current_timestamp::timestamp (0) without time zone,
    "place"      Varchar(100) NOT NULL ,
    "prolocutor" Varchar(10) NOT NULL ,
    "conferee"       Varchar(255) NOT NULL ,
    "recorder"       Varchar(10) NOT NULL ,
    "leitmotiv"    Varchar(255) NOT NULL ,
    "details"    text,
    UNIQUE   (subject),
    PRIMARY KEY ("id")
);

DROP VIEW oa.vmeeting;
CREATE OR REPLACE VIEW oa.vmeeting    AS
    SELECT    id,CASE    WHEN    character_length(subject)>20    THEN
substring(subject,1,20)||'...'::varchar(20) ELSE subject::varchar(20) END,
    caller,to_char(begin_time,'YYYY-MM-DD HH:MI') as begin_time,
    CASE WHEN character_length(place)>10 THEN substring(place,1,10)||'...'::varchar(15) ELSE
place::varchar(20) END as place,
    prolocutor
    FROM oa.meeting
    Order By id DESC;
select * from oa.vmeeting;
```

6 查询 SQL（DML）

6.1 子查询

下面是一个子查询的例子表 groupmember、rolemember 是存储组成员、角色成员的表。

```
Create table "user"
(
	"id" Serial NOT NULL,
	"userid" Varchar(50) NOT NULL,
	"passwd" Varchar(50),
	"name"	Varchar(20)NOT NULL ,
	"nickname" Varchar(20)NOT NULL ,
	"active" Boolean Default 'F',
	"email"	Varchar(50) NOT NULL,
	"question" Varchar(255) NOT NULL,
	"answer" Varchar(255) NOT NULL,
	"begin_date" Timestamp Default now(),
	"end_date" Timestamp Default now(),
	UNIQUE	(userid,email),
  primary key ("id")
);
Create index "user_index" on "user" using btree ("id","userid");

Create table "group"
(
	"id" Serial NOT NULL UNIQUE,
	"groupname" Varchar(20) NOT NULL,
	"description" Varchar(255),
	UNIQUE	(groupname),
	PRIMARY KEY ("id")
);
Create index "group_index" on "group" using btree ("id","groupname");

Create table "groupmember"
(
	"id" Serial NOT NULL UNIQUE,
	"gid" integer NOT NULL Default 0,
	"uid" integer NOT NULL Default 0,
	primary key ("id")
);

-- ============================================================
```

```
--    'Foreign Key'
-- ======================================================
Alter table "groupmember" add    foreign key ("uid") references "user" ("id") on update restrict on delete restrict;
Alter table "groupmember" add    foreign key ("gid") references "group" ("id") on update restrict on delete restrict;

-- ------------------------------------------------------
--    'vgroupmember'
-- ------------------------------------------------------
-- DROP VIEW vgroupmember;
CREATE VIEW vgroupmember AS
        SELECT gm.id,gm.gid,g.groupname,gm.uid,u.userid,u.name
                FROM "group" g,"user" u,groupmember gm
                Where u.id = gm.uid and g.id = gm.gid
        ORDER BY gm.id;
```

```
member=> select * from groupmember ;
  id | gid | uid
----+-----+-----
   1 |     1 | 245
  10 |     7 | 200
  11 |     7 | 201
  12 |     7 | 202
  13 |     3 | 200
  14 |     3 | 201
  15 |     3 | 202
  16 |     3 | 203
  17 |     3 | 204
  18 |     3 | 205
  19 |     6 | 247
  20 |     6 | 201
  21 |     6 | 203
  22 |     6 | 204
  23 |     6 | 205
  24 |     3 | 249
  25 |     3 | 250
  26 |     7 | 251
  27 |     3 | 252
(19 rows)

（组 ID 为 3 的用户列表）
member=> select * from groupmember where gid = 3;
  id | gid | uid
```

```
----+-----+-----
 13 |     3 | 200
 14 |     3 | 201
 15 |     3 | 202
 16 |     3 | 203
 17 |     3 | 204
 18 |     3 | 205
 24 |     3 | 249
 25 |     3 | 250
 27 |     3 | 252
(9 rows)
```

添加组成员步骤是：

1. 取用户 id，select id from vuser where userid='sysop'
2. 取组名 id，select id from "group" where groupname ='System'
3. 向 groupmember 表插入数据，insert into groupmember(gid,uid) values（组 ID，用户 ID）

子查询，SQL 语句应该写成：

insert into groupmember(gid,uid) values((select id from "group" where groupname ='System'),(select id from vuser where userid='sysop'));

插放数据测试：

```
member=> insert into groupmember(gid,uid) values((select id from "group" where groupname ='System'),(select id from vuser where userid='sysop'));
INSERT 110940 1

member=> select * from vgroupmember where gid = 1;
 id | gid | groupname | uid | userid | name
----+-----+-----------+-----+--------+------
  1 |   1 | System    | 245 | sysop  | chen
(2 rows)
```

测试成功，因为数据 id 号难懂，不易查看，这里使用了一个视图。

6.2 substring()函数截取部分汉字

要截取汉字的首要条件是数据库编码必须是 UNICODE。UNICODE 中英文一个字母=中文一个汉字长度。

如果使用非 UNICODE 编码的数据，EUC_CN 会输出提示你输入加“'”。

```
pureftpd=> select substring('数据库的编码是用系统表' from 1 for 4);
pureftpd'>
pureftpd'>'
pureftpd(> )
pureftpd-> ;
ERROR:  parser: unterminated quoted identifier at or near ""
```

```
)" at character 69
```

SQL_ASCII 还是输出半个汉字。

```
help=> select substring('数据库的编码是用系统表' from 1 for 4);
 substring
-----------
 数▒
(1 row)
```

查看数据库编码：

```
member=> \l
                List of databases
     Name      |   Owner    | Encoding
---------------+------------+-----------
 help          | postgres | SQL_ASCII
 member        | chen       | UNICODE
 mydatabase | postgres | UNICODE
 postgres      | postgres | SQL_ASCII
 pureftpd      | pureftpd | EUC_CN
 site          | postgres | EUC_CN
 template0     | postgres | SQL_ASCII
 template1     | postgres | SQL_ASCII
(8 rows)
```

测试：

```
member=> select substring('数据库的编码是用系统表' from 1 for 4);
 substring
-----------
 数据库的
(1 row)

member=> select substring('数据库的编码是用系统表' from 1 for 2);
 substring
-----------
 数据
(1 row)

member=> select * from  组;
序号 |       组名        |              描述
------+--------------+----------------------
     1 | 域用户            | 9812.net 域内用户
     3 | 计算机维护组 | 维护计算机的用户用户
(2 rows)
```

```
member=> select 组名,substring(描述 from 1 for 5) as 描述 from 组;
      组名       |      描述
--------------+------------
 域用户          | 9812.
 计算机维护组 | 维护计算机
(2 rows)
```

7 过程与函数

在其它数据中分别存在过程与函数，它们的功能没有区别，为什么要分为成过程与函数呢。因为过程是没有反回值的，而函数是要通过 RETURN 反回值的。返回值可以是任意一个符合 SQL 数据类型的值或结果集，在 PostgreSQL 中过程与函数都是使用 CREATE FUNCTION 语句来创建。

7.1 基本使用实例

我们创建一个名为 system_log 的表。通过 add_system_log()这个过程追加记录，有些朋友可能认为这是多此一举，我们过程来完成当然有自己的想法，好处是用户不必关心数据库结构和 SQL 语句，可以方便地在任何语言中调用。

```
DROP TABLE system_log CASCADE;
Create table system_log
(
    id    Serial NOT NULL UNIQUE,
    uid integer NOT NULL Default 0,
    ip    inet ,
    status varchar(255),
    description varchar(255),
    login_date Timestamp Default now(),
    PRIMARY KEY("id"),
    FOREIGN KEY (uid) REFERENCES "user" (id)
);
-- ----------------------------------------------------------
-- 'Function'
-- ----------------------------------------------------------
-- DROP FUNCTION add_system_log(integer,inet,varchar);
CREATE OR REPLACE FUNCTION add_system_log(integer,inet,varchar,varchar) RETURNS boolean AS '
    DECLARE
        vUID        ALIAS FOR $1;
        vIP         ALIAS FOR $2;
        vSTATUS     ALIAS FOR $3;
        vDESC       ALIAS FOR $4;
```

```
    BEGIN
        insert into system_log(uid,ip,status,description) values(vUID,vIP,vSTATUS,vDESC);
        RETURN true;
    END;
' LANGUAGE 'plpgsql';
select add_system_log(1,'127.0.0.1','Create Database','Initialization Database');
```

```
member=> select add_system_log(1,'127.0.0.1','Create Database','Initialization Database');
 add_system_log
----------------
 t
(1 row)
member=> select * from system_log ;
id | uid |     ip          | status   |       description       |         login_date
--+-----+---------------+---------+-----------------------+---------------------------

1 |   2 | 192.168.0.5   | 上线 | 密码                    | 2003-09-26 15:33:18.6732
2 |   2 | 192.168.0.2   | 登录 | 用户 admin              | 2003-09-26 16:19:21.824283
3 |   2 | 192.168.1.31 | 登录 | 用户 admin              | 2003-09-26 17:10:47.269064
(3 rows)

member=> select add_system_log(1,'127.0.0.1','Create Database','Initialization Database') as log;
 log
-----
 t
(1 row)
```

7.2 过程中使用 Select Into

这里有两个例子 adduser(varchar,varchar)、deluser(integer)。

函数：adduser(用户名,密码)

返回：布尔值 ture 成功，false 失败

功能：添加用户，首先查看用户是否存在，如果用户存在就反回 false，如果用户不存在就插入记录并返回 true。

函数：deluser(用户 ID)

返回：布尔值 ture 成功，false 失败

功能：删除用户，如果用户存在就删除用户并返回 true，如果用户不存在就返回 false。

```
-- ----------------------------------------------------
--   'siteuser'
-- ----------------------------------------------------
--DROP TABLE IF EXISTS siteuser;
DROP TABLE siteuser CASCADE;
DROP SEQUENCE siteuser_id_seq;
DROP INDEX siteuser_id_index;
```

```
CREATE TABLE siteuser (
	id integer DEFAULT nextval('siteuser_id_seq') NOT NULL,
	username   varchar(20) DEFAULT '0' NOT NULL,
	Password   varchar(50) DEFAULT '0' NOT NULL,
	realname   varchar(10) DEFAULT '0' NOT NULL,
	email              varchar(50) DEFAULT '0' NOT NULL,
	create_date    timestamp DEFAULT now(),
	modify_date    timestamp DEFAULT now(),
	UNIQUE             (id,username),
	PRIMARY     KEY (id)
);
CREATE SEQUENCE siteuser_id_seq;
CREATE INDEX siteuser_id_index ON siteuser (id);

DROP FUNCTION adduser(varchar,varchar);
CREATE OR REPLACE FUNCTION adduser(varchar,varchar) RETURNS boolean AS '
	DECLARE
		bool boolean := false;
		name text;
		uid    integer;
		user ALIAS FOR $1;
		pass ALIAS FOR $2;
		su     siteuser%ROWTYPE;
	BEGIN
		SELECT INTO name username FROM siteuser WHERE username = user;
		IF NOT FOUND then
			insert into siteuser(username,password) values(user,pass);
			SELECT INTO uid id FROM siteuser WHERE username = user;
			bool := true;
		ELSE
			bool := false;
			RAISE NOTICE "Calling adduser() return %",bool;
		END IF;
	RETURN bool;
	END;
' LANGUAGE 'plpgsql';
```

```
select adduser('ccscc','eeee');
select * from siteuser ;
```

```
DROP FUNCTION deluser(integer);
CREATE OR REPLACE FUNCTION deluser(integer) RETURNS boolean AS '
	DECLARE
```

```
        bool boolean := false;
        userid text;
    BEGIN
        SELECT INTO userid id FROM siteuser WHERE id = $1;
        IF FOUND then
            delete from siteuser where id = $1;
            bool := true;
        ELSE
        END IF;
        RETURN bool;
    END;
' LANGUAGE 'plpgsql';
```

```
select deluser(1);
```

7.3 返回 integer

```
CREATE OR REPLACE FUNCTION get_user_id(varchar) RETURNS integer AS '
    DECLARE
        vUser       ALIAS FOR $1;
        uid integer;
    BEGIN
        SELECT INTO uid id FROM "users" WHERE userid = vUser;
        RETURN uid;
    END;
' LANGUAGE 'plpgsql';
```

7.4 返回 void

```
CREATE OR REPLACE FUNCTION enable_user(varchar) RETURNS void AS '
    DECLARE
        vUser       ALIAS FOR $1;
    BEGIN
        Update "users" set status = "true" where userid = vUser;
        RETURN;
    END;
' LANGUAGE 'plpgsql';

CREATE OR REPLACE FUNCTION disable_user(varchar) RETURNS void AS '
    DECLARE
        vUser       ALIAS FOR $1;
```

```
    BEGIN
        Update "users" set status = "false" where userid = vUser;
        RETURN;
    END;
' LANGUAGE 'plpgsql';
```

8 规则

这里有两上规则例子 role_rule、group_rule。它们功能都是删除其它表中受外键约束的记录。用触发器也可以实现同样功能，触发器繁琐，要写一个过程。而规则灵巧得多。

如果是单一的 Select、Update、Insert、 Delect 推荐使用规则。如果要同时 Update,Insert, Delect 可以使用触发器

```
        IF TG_OP = "UPDATE" THEN
            xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
        END IF;
        IF TG_OP = "INSERT" THEN
            xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
            xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
        END IF;
        IF TG_OP = "DELETE" THEN
            xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
            xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
            xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
        END IF;
```

8.1 规则实例

```
-- ============================================================
--   'group'
-- ============================================================
Create table "group"
(
    "id" Serial NOT NULL UNIQUE,
    "groupname" Varchar(20) NOT NULL,
    "description" Varchar(255),
    UNIQUE  (groupname),
    PRIMARY KEY ("id")
);
Create index "group_index" on "group" using btree ("id","groupname");
-- ============================================================
--   'groupmember'
```

```
-- ============================================================
Create table "groupmember"
(
	"id" Serial NOT NULL UNIQUE,
	"gid" integer NOT NULL Default 0,
	"uid" integer NOT NULL Default 0,
	primary key ("id")
);
-- ------------------------------------------------------------
--   'vgroupmember'
-- ------------------------------------------------------------
-- DROP VIEW vgroupmember;
CREATE VIEW vgroupmember AS
	SELECT gm.id,gm.gid,g.groupname,gm.uid,u.userid,u.name
		FROM "group" g,"user" u,groupmember gm
		Where u.id = gm.uid and g.id = gm.gid
	ORDER BY gm.id;
-- ------------------------------------------------------------
--   'RULE'
-- ------------------------------------------------------------
CREATE RULE group_rule AS ON Delete TO "group"
	DO Delete From groupmember where gid = OLD.id;

-- ------------------------------------------------------------
--   'Insert Data'
-- ------------------------------------------------------------
Insert into "group"(groupname,description) values('System','系统管理员');
Insert into "group"(groupname,description) values('Administrator','站点管理员');
Insert into "group"(groupname,description) values('gold','金');
Insert into "group"(groupname,description) values('silver','银');
Insert into "group"(groupname,description) values('copper','铜');
Insert into "group"(groupname,description) values('advance','高级会员');
Insert into "group"(groupname,description) values('free','免费会员');

-- ============================================================
--   'role'
-- ============================================================
-- drop table role CASCADE;
Create table "role"
(
	"id" Serial NOT NULL UNIQUE,
	"rolename" Varchar(20) NOT NULL,
	"description" Varchar(255),
	UNIQUE   (rolename),
```

```
        PRIMARY KEY ("id")
);
Create index "role_index" on "role" using btree ("id","rolename");

-- ============================================================
--    'rolemember'
-- ============================================================
-- drop table rolemember CASCADE ;
Create table "rolemember"
(
        "id" Serial NOT NULL UNIQUE,
        "rid" integer NOT NULL Default 0,
        "uid" integer NOT NULL Default 0,
  primary key ("id")
);
-- ------------------------------------------------------------
--    'vrolemember'
-- ------------------------------------------------------------
CREATE VIEW vrolemember AS
        SELECT rm.id,rm.rid,r.rolename,rm.uid,u.userid,u.name
                FROM "role" r,"user" u,rolemember rm
                Where u.id = rm.uid and r.id = rm.rid
                ORDER BY rm.id;
-- ------------------------------------------------------------
--    'RULE'
-- ------------------------------------------------------------
CREATE RULE role_rule AS ON Delete TO role
        DO Delete From rolemember where rid = OLD.id;

-- ------------------------------------------------------------
--    'Insert Data'
-- ------------------------------------------------------------
Insert into role(rolename,description) values('System','系统管理员');
Insert into role(rolename,description) values('Administrator','站点管理员');
Insert into role(rolename,description) values('gold','金');
Insert into role(rolename,description) values('silver','银');
Insert into role(rolename,description) values('copper','铜');
Insert into role(rolename,description) values('advance','高级会员');
Insert into role(rolename,description) values('free','免费会员');

-- ============================================================
--    'Foreign Key'
-- ============================================================
Alter table "groupmember" add    foreign key ("uid") references "user" ("id") on update restrict on delete
```

restrict;
Alter table "groupmember" add foreign key ("gid") references "group" ("id") on update restrict on delete restrict;
Alter table "rolemember" add foreign key ("uid") references "user" ("id") on update restrict on delete restrict;
Alter table "rolemember" add foreign key ("rid") references "role" ("id") on update restrict on delete restrict;

9 触发器

9.1 一般用法

下面 clientinfo_tri 触发器与 clientinfo_rule 规则功能相同。

```
CREATE OR REPLACE FUNCTION clientinfo_tri_func () RETURNS opaque AS '
    BEGIN
        Delete from prodorder where clientinfo_id = OLD.id;
    RETURN OLD;
    END;
' LANGUAGE 'plpgsql';

DROP TRIGGER clientinfo_tri on clientinfo;
CREATE TRIGGER clientinfo_tri
        BEFORE Delete ON clientinfo FOR EACH ROW
        EXECUTE PROCEDURE clientinfo_tri_func ();

CREATE RULE clientinfo_rule AS ON Delete TO prodorder
    DO  Delete from prodorder where clientinfo_id = OLD.id;
```

9.2 多个触发器使用同一个过程

这是一个多个触发器使用同一个函数例子。

触发器 siteuser_delete_tri 功能是：

当删除 siteuser 表中记录时，首先删除其它表中受外键约束的记录，如果不使用触发器或规则来完成，就很麻烦了。

如下：

```
Begin;
Delete from company where uid = id;
Delete from link where uid = id;
Delete from product_sort where uid = id;
Delete from news where uid = id;
```

```
Delete from count where uid = id;
Delete from guestbook where uid = id;
Delete from clientinfo where uid = id;
Delete from column_bar where uid = id;
Delete from drumbeating where uid = id;
Commit;
```

在程序中完成上面的操作,

```
String sql1="Delete from company where uid = ?";
String sql2="Delete from link where uid = ?";
String sql3="Delete from product_sort where uid = ?";
String sql4="Delete from news where uid = ?";
String sql5="Delete from count where uid = ?";
String sql6="Delete from guestbook where uid = ?";
String sql7="Delete from clientinfo where uid = ?";
String sql8="Delete from column_bar where uid = ?";
String sql9="Delete from drumbeating where uid = ?";
String id  = <your-id>;
DBConnect odb = null;
try{
    odb = new DBConnect(oDatabase);
    odb.Begin();
    odb.prepareStatement(sql1);
    odb.setString(1,id);
    rs = odb. executeUpdate ();
    odb.prepareStatement(sql2);
    odb.setString(1,id);
    rs = odb. executeUpdate ();
    odb.prepareStatement(sql3);
    odb.setString(1,id);
    rs = odb. executeUpdate ();
    odb.prepareStatement(sql4);
    odb.setString(1,id);
    rs = odb. executeUpdate ();
    odb.prepareStatement(sql5);
    odb.setString(1,id);
    rs = odb. executeUpdate ();
    odb.prepareStatement(sql6);
    odb.setString(1,id);
    rs = odb. executeUpdate ();
    odb.prepareStatement(sql7);
    odb.setString(1,id);
    rs = odb. executeUpdate ();
    odb.prepareStatement(sql8);
    odb.setString(1,id);
```

```
                    rs = odb. executeUpdate ();
                    odb.prepareStatement(sql9);
                    odb.setString(1,id);
                    rs = odb. executeUpdate ();

                    odb.Commit();
                }
                catch(Exception e){
                    e.printStackTrace();
                }
                finally{
                    try{
                        odb.close();
                    }catch(Exception e){
                        e.printStackTrace();
                    }
                }
```

触发器 siteuser_insert_tri 功能是：

当向 siteuser 表中插入记录时同时向 company、count、drumbeating……表中增加一条记录。

```
-- ----------------------------------------------------
--  'siteuser'
-- ----------------------------------------------------
--DROP TABLE IF EXISTS siteuser;
DROP TABLE siteuser CASCADE;
DROP SEQUENCE siteuser_id_seq;
DROP INDEX siteuser_id_index;

CREATE TABLE siteuser (
    id integer DEFAULT nextval('siteuser_id_seq') NOT NULL,
    username   varchar(20) DEFAULT '0' NOT NULL,
    Password   varchar(50) DEFAULT '0' NOT NULL,
    realname   varchar(10) DEFAULT '0' NOT NULL,
    email          varchar(50) DEFAULT '0' NOT NULL,
    create_date    timestamp DEFAULT now(),
    modify_date    timestamp DEFAULT now(),
    UNIQUE          (id,username),
    PRIMARY     KEY (id)
);
CREATE SEQUENCE siteuser_id_seq;
CREATE INDEX siteuser_id_index ON siteuser (id);

DROP FUNCTION siteuser_tri_func() CASCADE ;
CREATE OR REPLACE FUNCTION siteuser_tri_func () RETURNS opaque AS '
--      DECLARE
```

```
--          user_id CONSTANT INTEGER := OLD.id;
    BEGIN
        IF TG_OP = "DELETE" THEN
            Delete from company where uid = OLD.id;
            Delete from link where uid = OLD.id;
            Delete from product_sort where uid = OLD.id;
            Delete from news where uid = OLD.id;
            Delete from count where uid = OLD.id;
            Delete from guestbook where uid = OLD.id;
            Delete from clientinfo where uid = OLD.id;
            Delete from column_bar where uid = OLD.id;
            Delete from drumbeating where uid = OLD.id;
        END IF;
        IF TG_OP = "INSERT" THEN
            INSERT INTO company(uid) values(NEW.id);
            INSERT INTO count(uid,number,fontcolor,backgroundcolor)
values(NEW.id,0,"ffffff","000000");
            INSERT INTO drumbeating(uid,logourl,bannerurl)
values(NEW.id,"default_logo.jpg","default_banner.jpg");
        END IF;
    RETURN OLD;
    END;
' LANGUAGE 'plpgsql';


DROP TRIGGER siteuser_delete_tri on siteuser;
CREATE TRIGGER siteuser_delete_tri
        BEFORE Delete ON siteuser FOR EACH ROW
        EXECUTE PROCEDURE siteuser_tri_func ();

DROP TRIGGER siteuser_insert_tri on siteuser;
CREATE TRIGGER siteuser_insert_tri
        AFTER INSERT ON siteuser FOR EACH ROW
        EXECUTE PROCEDURE siteuser_tri_func ();
```

9.3 时间调度触发器

PostgreSQL 本身关没有提供这种功能。但是我们可以用其它方式来代替同样的功能。

```
[root@linux pgsql]# cat function.sh
#!/bin/bash

psql -Uchen member -c "select add_system_log(1,'127.0.0.1','Create Database','Initialization Database');"
# or
# psql -Uchen member -f your_sql_script.sql
```

9.3.1 定时触发器

定时触发器与周期触发器。的不同是定时，只运行一次，而周期是有规律的运行。

- 明天的这时候运行 function.sh

```
[root@linux8 pgsql]# at -f function.sh tomorrow
warning: commands will be executed using (in order) a) $SHELL b) login shell c) /bin/sh
job 15 at 2003-10-14 18:05
```

- 今晚 8：00 运行 optimize.sh Shell

```
[root@linux8 pgsql]# at -f optimize.sh 20:00 + 1 days
warning: commands will be executed using (in order) a) $SHELL b) login shell c) /bin/sh
job 18 at 2003-10-14 20:00
```

4 天后的零晨 1：00 运行 function.sh

```
[root@linux8 pgsql]# at -f function.sh + 5 days
warning: commands will be executed using (in order) a) $SHELL b) login shell c) /bin/sh
job 20 at 2003-10-18 01:00
```

- 查看 at queue

```
[root@linux8 pgsql]# atq
11      2003-10-14 01:00 a root
12      2003-10-14 18:00 a root
13      2003-10-14 18:00 b root
14      2003-10-14 18:03 a root
15      2003-10-14 18:05 a root
16      2003-10-14 01:00 a root
17      2003-10-13 20:00 a root
18      2003-10-14 20:00 a root
19      2003-10-15 01:00 a root
20      2003-10-18 01:00 a root
或使用
[root@linux8 pgsql]# at -l
11      2003-10-14 01:00 a root
12      2003-10-14 18:00 a root
13      2003-10-14 18:00 b root
14      2003-10-14 18:03 a root
15      2003-10-14 18:05 a root
16      2003-10-14 01:00 a root
17      2003-10-13 20:00 a root
18      2003-10-14 20:00 a root
19      2003-10-15 01:00 a root
20      2003-10-18 01:00 a root
```

9.3.2 周期触发器

- 每天的零晨 3：00 运行一次 function.sh

```
[root@linux etc]# cat crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
0 3 * * * root /usr/local/pgsql/function.sh
```

- 每天，每两个小时运行一次 function.sh

```
0 */2 * * * root /usr/local/pgsql/function.sh
```

- 每月 1 日早上零晨 2 点运行 function.sh

```
0 2 1 * * root /usr/local/pgsql/function.sh
```

- 每个月的星期一 17：00 点运行 function.sh

```
0 17 * * mon root /usr/local/pgsql/function.sh
```

例子不再举了，请参考操作系统手册。

10 游标

```
CREATE FUNCTION reffunc(refcursor) RETURNS refcursor AS '
BEGIN
        OPEN $1 FOR SELECT * FROM system_log;
        RETURN $1;
END;
' LANGUAGE 'plpgsql';

BEGIN;
SELECT reffunc('funccursor');
FETCH ALL IN funccursor;
COMMIT;
```

```
member=>
member=> BEGIN;
BEGIN
member=> SELECT reffunc('funccursor');
   reffunc
------------
 funccursor
(1 row)

member=> FETCH ALL IN funccursor;
 id | uid |          ip          |        status        |                    description                    |
login_date
----+-----+--------------+----------------+-----------------------------------+---------------------------
  8 |   1 | 192.168.0.1  | eeee                 | ddddddddddddddddddddddddddddddddddd | 2003-09-25
14:55:19.62117
53 |   1 | 127.0.0.1       | Create Database | Initialization Database                  | 2003-09-25
15:23:34.941765
 54 |   2 | 192.168.0.5  | 上线                 | 密码                                | 2003-09-26
15:33:18.6732
 55 |   2 | 192.168.0.2  | 登录                 | 用户 admin                          | 2003-09-26
16:19:21.824283
 56 |   2 | 192.168.1.31 | 登录                 | 用户 admin                          | 2003-09-26
17:10:47.269064
```

10.1 游标结果集

这是一个在 Java 中读取游标反回的结果集例子：

```
public void Cursors(){
    String sql                  = "SELECT reffunc('funccursor')";
    String sql2                 = "FETCH ALL IN funccursor";

    DBConnect odb = null;
    try{
        odb = new DBConnect(oDatabase);
        odb.Begin();
        odb.prepareStatement(sql);
        rs = odb.executeQuery();
        odb.prepareStatement(sql2);

        //odb.setString(1,user);
        rs = odb.executeQuery();
        if(rs!=null) {
            while(rs.next()){
```

```
                    System.out.print(rs.getString(1)+"|");
                    System.out.print(rs.getString(2)+"|");
                    System.out.print(rs.getString(3)+"|");
                    System.out.println(rs.getString(4));
                }
            }
            odb.Commit();
        }
        catch(Exception e){
            e.printStackTrace();
        }
        finally{
            try{
                odb.close();
            }catch(Exception e){
                e.printStackTrace();
            }
        }
}
```

```
53|1|127.0.0.1|Create Database
54|2|192.168.0.5|上线
55|2|192.168.0.2|登录
56|2|192.168.1.31|登录
57|2|127.0.0.1|登录
58|1|127.0.0.1|登录
```

10.2 例子 2

```
CREATE FUNCTION reffunc(refcursor,integer) RETURNS refcursor AS '
BEGIN
        OPEN $1 FOR SELECT id,pname,num as amount,price,newprice,(newprice*num) as subtotal,((price-newprice)*num) as cost_saving FROM product WHERE id = $2;
        RETURN $1;
END;
' LANGUAGE 'plpgsql';
```

11 事务处理

11.1 批量插入、更新、删除

对数据进行批量的插入、更新、删除操作，建议使用事务处理。

1. 降低对物理磁盘的读写操作，以便利高数据性能
2. 保持数据完整、一至

请看下面的例子

11.1.1 批量插入操作-例 1

```
insert into "group"(groupname,description) values('System Admin','系统管理员');
insert into "group"(groupname,description) values('Power Admin','系统管理员');
insert into "group"(groupname,description) values('Backup Admin','系统管理员');
……  ……
……  ……
……  ……
insert into "group"(groupname,description) values('xxxxxxxxxxxx','xxxxxxxxxxx');
```

这个例子要完成大量的数据插入操作。在程序在实现这个操作，基本都是使用一个循环操作如：

```
for(条件){
    运行 SQL 语句：insert into "group"(groupname,description) values('变量','变量');
}
```

如果插入数据足够多，你会发现你的磁盘在嚓嚓做响，并且硬盘指示灯疯狂闪烁。

优化后更改为：

```
begin
for(条件){
    运行 SQL 语句：insert into "group"(groupname,description) values('变量','变量');
}
commit
```

插入数据被保存到内存中，只有 commit 时你的磁盘才会嚓嚓响。除非你的存内不够用，系统使用交换分区。当前主流的服务器配置基本都是 1G-2G 内存再加 4G 交换分区，一次插入几千行的事务小 CASE。

这里给一个 Java 的例子：

```
public void delGroupMember(ArrayList id) {
    String sql = "delete from groupmember where id=?";
    try{
        odb = new DBConnect(oDatabase);
        odb.Begin();
        for(int i=0;i<id.size();i++){
```

```
                odb.prepareStatement(sql);
                odb.setString(1,(String)id.get(i));
                odb.executeUpdate();
            }
            odb.Commit();
            odb.AutoCommit();
        }
        catch(Exception e){
            odb.Rollback();
            e.printStackTrace();
        }
        finally{
            try{
                odb.close();
            }catch(Exception e){
                e.printStackTrace();
            }
        }
    }
```

11.2 保持数据完整-例 2

当删除 A 表中一条记录，同时删除 B，C，D，E，F……等表中的相关数据，当然你可以使用 RULE、TRIGGER 来完成，我这是只是举个例子。

由程序完成这样的删除作操，如果不使用事务处理，中途中断操作数据就会不一至。

```
Begin;
    Delete from company where uid = id;
    Delete from link where uid = id;
    Delete from product_sort where uid = id;
    Delete from news where uid = id;
    Delete from count where uid = id;
    Delete from guestbook where uid = id;
    Delete from clientinfo where uid = id;
    Delete from column_bar where uid = id;
    Delete from drumbeating where uid = id;
Commit;
```

12 用户权限

需求例子

某公司经营虚拟主机，数据库等业务，需求如下

1. 每个虚拟主机用户对应一个数据（PostgreSQL）。
2. 数据库与数据库用户之间独立，互不干扰。
3. 有一些用户虚拟主机在其它公司，要买我们的 PostgreSQL 数据库服务。
4. 用户分为企业，个人。
5. 每户有时限，过期后自动失效（停用）。

分析：
1. 每个主机对应一个数据库，并且用户之间独立，我们可以配置 pg_hba.conf 文件来完成需求（1），（2），（3）。
2. 创建企业组和个人组来实现需求中的（4）
3. 过期策略，使用 VALID UNTIL 选项来完成

建议：

数据库与用户名命名规则，建议你使用该用户虚拟主机的域名。如：
www.9812.net 数据库为:9812_net 用户名为:9812_net
数据库名使用“_”下划线需要使用双引号"9812_net", "xuser_net"
如果不想使用双引号，给用户添麻烦。也可以命名为:
9812net，xusernet，kdeopencom
或不用后缀直接
9812，xuser，kdeopen

12.1.1 组

12.1.1.1 创建组

CREATE GROUP groupname;

要向组中增加用户或删除用户：
ALTER GROUP groupname ADD USER user1, user2... ;
ALTER GROUP groupname DROP USER user1, user2... ;

```
postgres=# CREATE GROUP company;
CREATE GROUP
postgres=# CREATE GROUP person;
CREATE GROUP
postgres=#
```

12.1.1.2 删除组

DROP GROUP groupname;

```
postgres=# DROP GROUP vhost;
DROP GROUP
```

12.1.2 用户

12.1.2.1 创建用户

1. 创建一个名为 netkiller 的用户，指定密码 chen。
   CREATE USER netkiller WITH PASSWORD 'chen';
2. 创建一个用户：dbuser1，密码：chen，用户在 2003-12-1 过期。
   CREATE USER dbuser1 WITH PASSWORD 'chen' VALID UNTIL '2003-12-1';

```
postgres=# CREATE USER dbuser1 WITH PASSWORD 'chen' VALID UNTIL '2003-12-1';
CREATE USER
postgres=# CREATE USER "kdeopen_comt" WITH PASSWORD 'chen' VALID UNTIL '2003-12-30';
CREATE USER
postgres=# CREATE USER "9812_net" WITH PASSWORD 'chen' VALID UNTIL '2004-2-1';
CREATE USER
```

3. 创建用户一个隶属于 vhost 组的用户。
   CREATE USER vhostdbuser WITH PASSWORD 'chen' IN GROUP vhost;

```
postgres=# CREATE GROUP vhost;
CREATE GROUP
postgres=# CREATE USER vhostdbuser WITH PASSWORD 'chen' IN GROUP vhost;
CREATE USER
postgres=# CREATE USER vhostdbuser1 WITH PASSWORD 'chen' VALID UNTIL '2003-12-1' IN
GROUP vhost;
CREATE USER
postgres=#
postgres=# \du
                          List of database users
   User name      | User ID |                Attributes
--------------+---------+---------------------------
 9812_net        |     102 |
 dbuser1         |     101 |
 kdeopen_com |     103 |
 netkiller       |     100 |
 postgres        |       1 | superuser, create database
 vhostdbuser   |     104 |
 vhostdbuser1 |     105 |
(7 rows)

postgres=#
```

12.1.2.2 删除用户

DROP USER username;

```
postgres=# DROP USER dbuser1;
DROP USER
postgres=#
```

12.1.2.3 修改密码

ALTER USER "用户" PASSWORD '密码';

```
postgres=# ALTER USER "9812_net" PASSWORD '123456';
ALTER USER
postgres=#
```

测试：

```
bash-2.05b$ psql -h 127.0.0.1 -U9812
Password: 输入新密码
psql: FATAL: Password authentication failed for user "9812"

bash-2.05b$ psql -h 127.0.0.1 -U9812_net
Password:
Welcome to psql 7.3.4, the PostgreSQL interactive terminal.

Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help on internal slash commands
      \g or terminate with semicolon to execute query
      \q to quit

9812_net=>
```

12.1.3 创建数据

创建数据库，并指定所有者：

CREATE DATABASE kdeopen WITH OWNER = 9812_net TEMPLATE = template0 ENCODING = 'UNICODE';

```
postgres=# CREATE DATABASE "9812_net" WITH OWNER = "9812_net" TEMPLATE = template0
ENCODING = 'UNICODE';
CREATE DATABASE
postgres=# \l
              List of databases
   Name      |    Owner    | Encoding
```

```
-----------+-----------+-----------
 9812_net   | 9812_net  | UNICODE
 netkiller | netkiller | UNICODE
 postgres   | postgres  | SQL_ASCII
 template0 | postgres  | SQL_ASCII
 template1 | postgres  | SQL_ASCII
(5 rows)

postgres=#
```

12.1.4 用户认证

12.1.4.1 本地连接

```
# local      DATABASE   USER   METHOD   [OPTION]
# host       DATABASE   USER   IP-ADDRESS   IP-MASK   METHOD   [OPTION]
# hostssl    DATABASE   USER   IP-ADDRESS   IP-MASK   METHOD   [OPTION]
```

pg_hba.conf 文件配置例子

```
[root@linux data]# cat pg_hba.conf
host      all          all          127.0.0.1          255.255.255.255     md5
local     all          all                                                 trust
```

测试，从 127.0.0.1 连接成功

```
[root@linux data]# psql -h127.0.0.1 -u member
psql: Warning: The -u option is deprecated. Use -U.
User name: chen
Password:
Welcome to psql 7.3.3, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

member=>
```

从外地测试，从 LAN 连接失败

```
[root@linux data]# psql -hwww.9812.net -u member
psql: Warning: The -u option is deprecated. Use -U.
User name: chen
Password:
psql: FATAL:  No pg_hba.conf entry for host 61.145.139.113, user chen, database member
```

```
[root@linux2 data]#
```

注意：是本地连接也要加-h 127.0.0.1。否则系统认为你用 UNIX Domain Socket 连接 /tmp/ .s.PGSQL.5432

12.1.4.2 允许任何 IP 连接主机

```
[root@linux data]# cat pg_hba.conf
host    all         all         0.0.0.0         0.0.0.0         md5
```

12.1.5 脚本例子

9812 是本地用户（应用程序和数据库在同一台服务器上）
kdeopen 是外部用户来自 202.103.190.130
xuser 是动态 IP 拔号用户，可以在任何地方连接他的 xuser 数据库

pg_hba.conf：

```
[root@linux data]# cat pg_hba.conf
host    9812        9812        127.0.0.1       255.255.255.255   md5
host    kdeopen     kdeopen     202.103.190.130 255.255.255.0     md5
host    xuser       xuser       0.0.0.0         0.0.0.0           md5
```

SQL 脚本：

```
CREATE GROUP company;
CREATE GROUP person;
CREATE USER 9812 WITH PASSWORD 'chen' VALID UNTIL '2003-12-1' IN GROUP person;
CREATE DATABASE 9812 WITH OWNER = 9812 TEMPLATE = template0 ENCODING = 'UNICODE';
CREATE USER kdeopen WITH PASSWORD 'chen' VALID UNTIL '2003-12-1' IN GROUP company;
CREATE DATABASE kdeopen WITH OWNER = kdeopen TEMPLATE = template0 ENCODING = 'UNICODE';
CREATE USER xuser WITH PASSWORD 'chen' VALID UNTIL '2003-12-1' IN GROUP person;
CREATE DATABASE xuser WITH OWNER = xuser TEMPLATE = template0 ENCODING = 'UNICODE';
```

12.1.6 权限

过几天写
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

13 FAQ

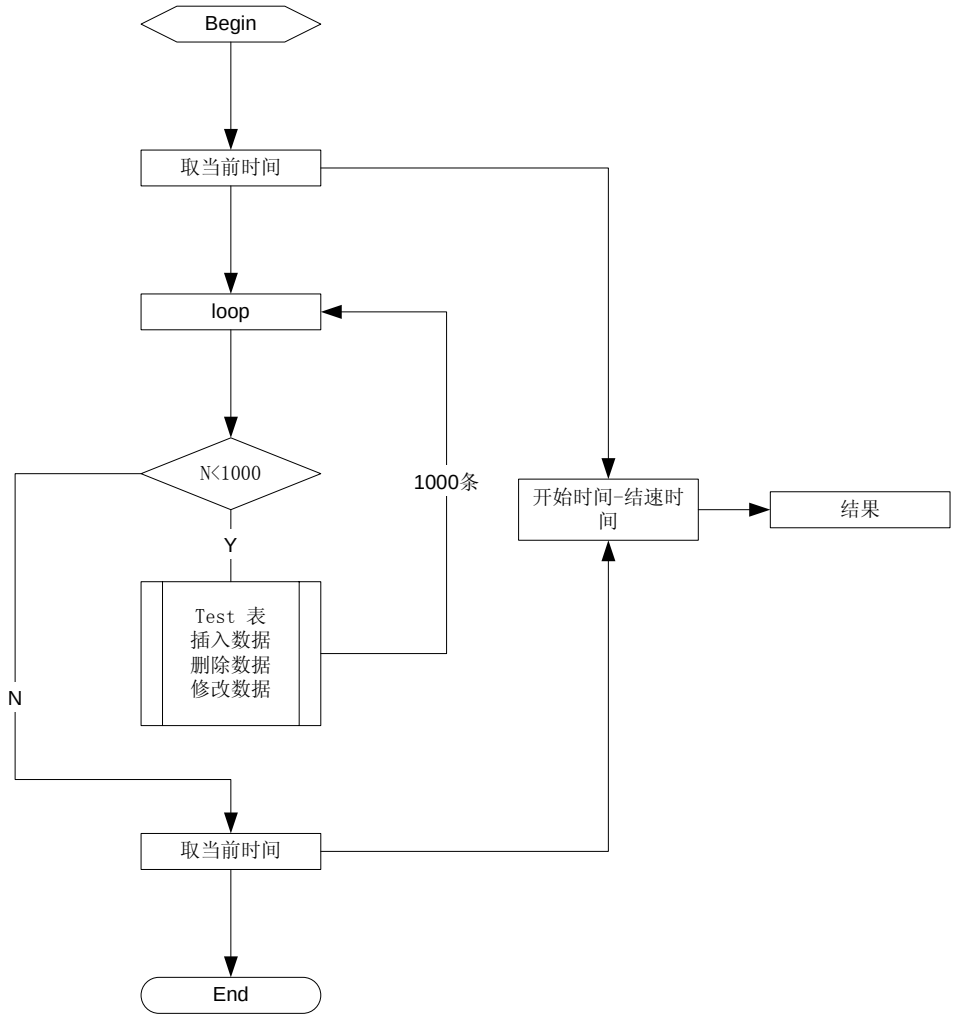
13.1 Postgresql 与其它数据库

网上有很多文章评论 PostgreSQL ， MySQL ， Oracle , Sybase , MS Sql Server……而且还做了测试。结果当然是 MySQL 快了。还有一些无聊人丑化 PostgreSQL，说它不稳定，爱宕机……

我要问测试过程得出的结果是否公平？

- 做测试的人是做什么的。是数据方面高手吗？或 DBA 吗？
- 他对数据优化了吗？
- 记录多少？
- 存储大小？
- 测试结果是应用于什么领域？
- 使用什么 API（jdbc,odbc,c++ api……）

如果你看过这类文章你会发现他们基本上是如下套路：



1. 创建数据
2. 创建表
3. 取当前时间，开始时间
4. 使用程序循环对 test 表分别插入、删除、更新数据
5. 取当前时间，结束时间
6. 两次时间差，得出结果
7. 这样可以得出三个结果，插入、删除、更新

这样的测试结果一点也不公平。

我这里有几种测试数据库性能方法。只供参考，不一定正确、准确。

测试环境和件条：

- 相同配置的计算机
- 相同操作系统
- 使用同一种语言和数据库接口（最公平的是使用 C++ API 连接数据库）
- 注意索引
- 在 win 平测试结果不准

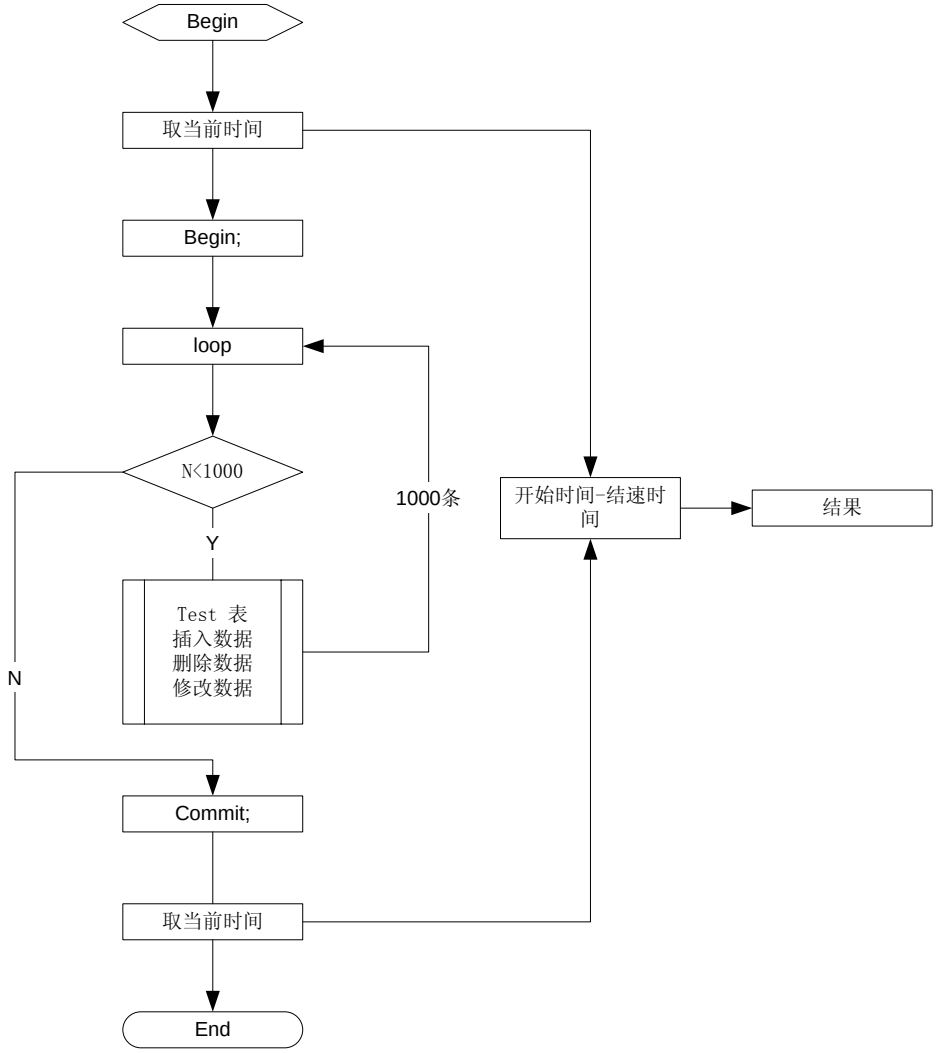
测试实例：

1. 首先进行单表的插入、删除、更新测试。得出结果（流程同上）
2. 进行事务处理单表的插入、删除、更新测试，每 100，1000，5000 等为一个组做事务，得

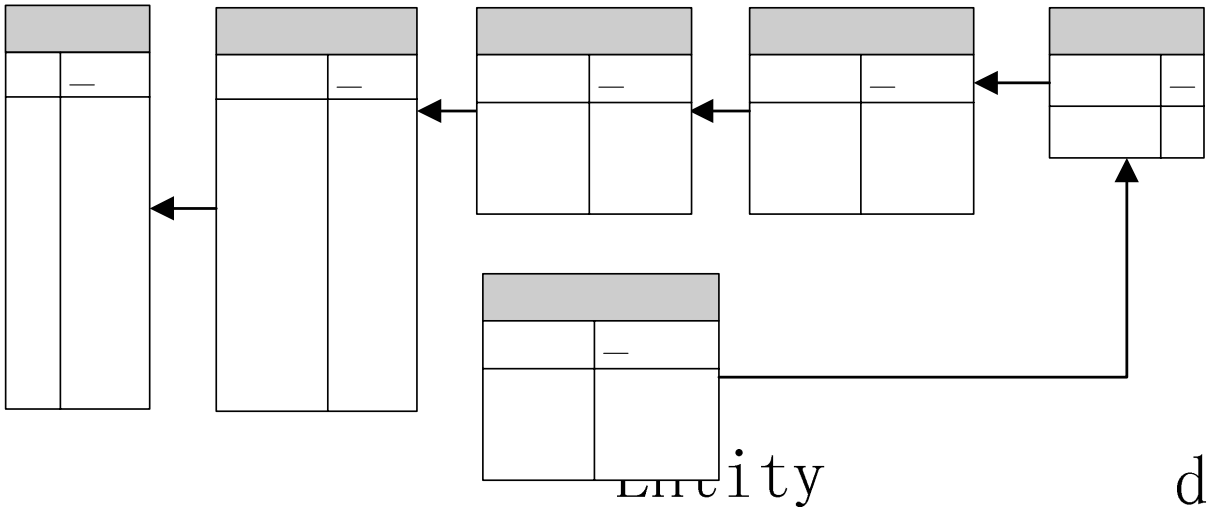
出结果 图 （一）

3. 多表测试
   创建一个实表，有 11 字段，id 字段为 PK 主键，在表中插入 10 条记录
   然后创建维表，有 11 字段，id 字段为 FK 外建，参考实表 id 字段（一对多），插入 10 条记录，
   以此类推。见 图（二）使用 join 测试分别在不同维度时的结果，更高级测试请参看《数据仓库》
4. 对于大型数据库是不能光看记录多少，还要看数据库容量。从 5G 起步，10G，50G，100G，200G…… 使用上面的多表测试方法将其中某字段改为大对象。得出结果
5. 子查询性能测试
6. 压力测试 要自己写一个小程序，多线程连接数据。
   然后几台机上运行这个程序 5 几进程，每个进程中有 10 个线程

图 （一）



图（二）

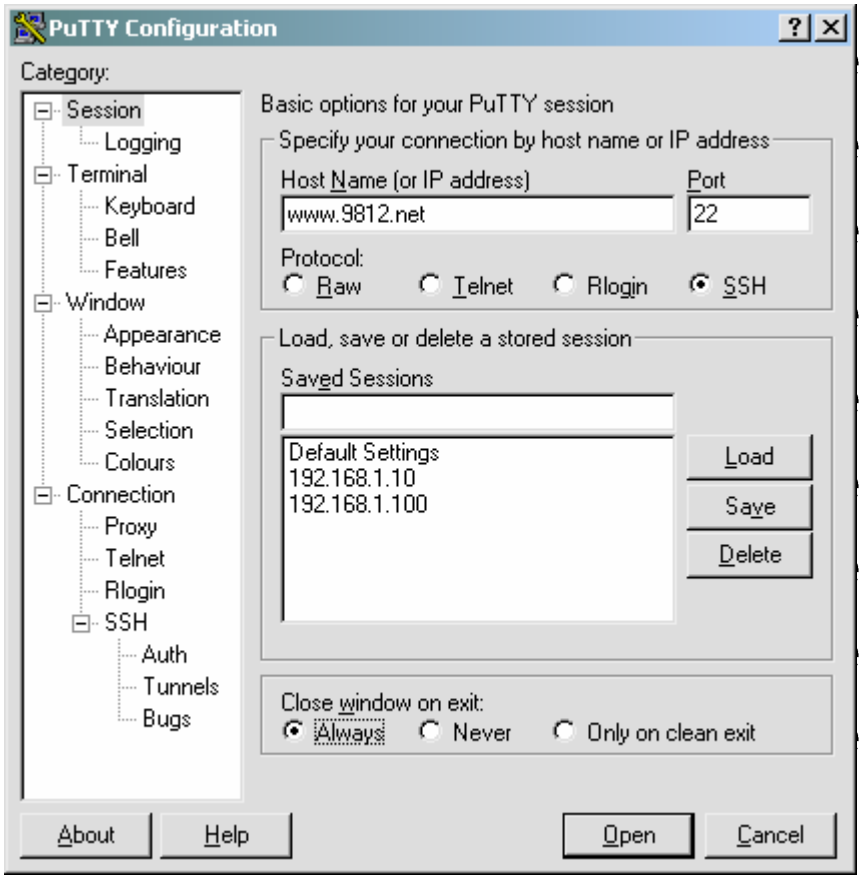


Entity dimen

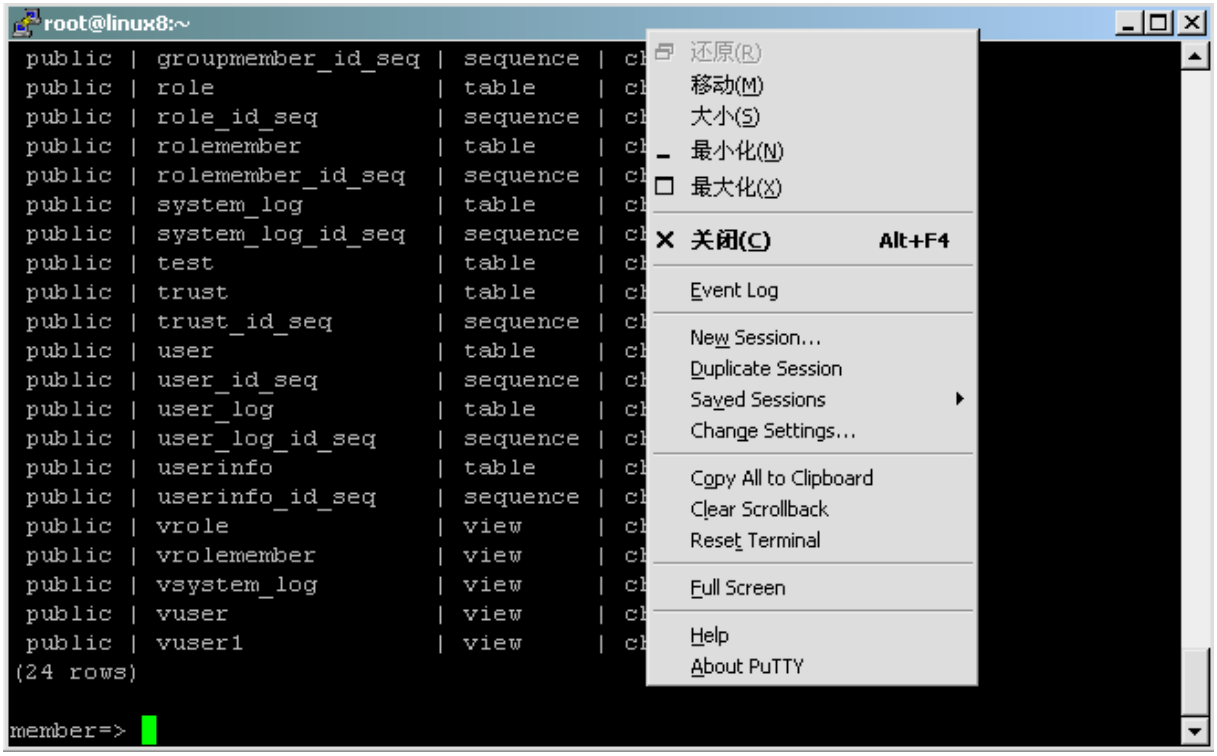
PK id FK1

13.2 Putty 中输入汉字的问题

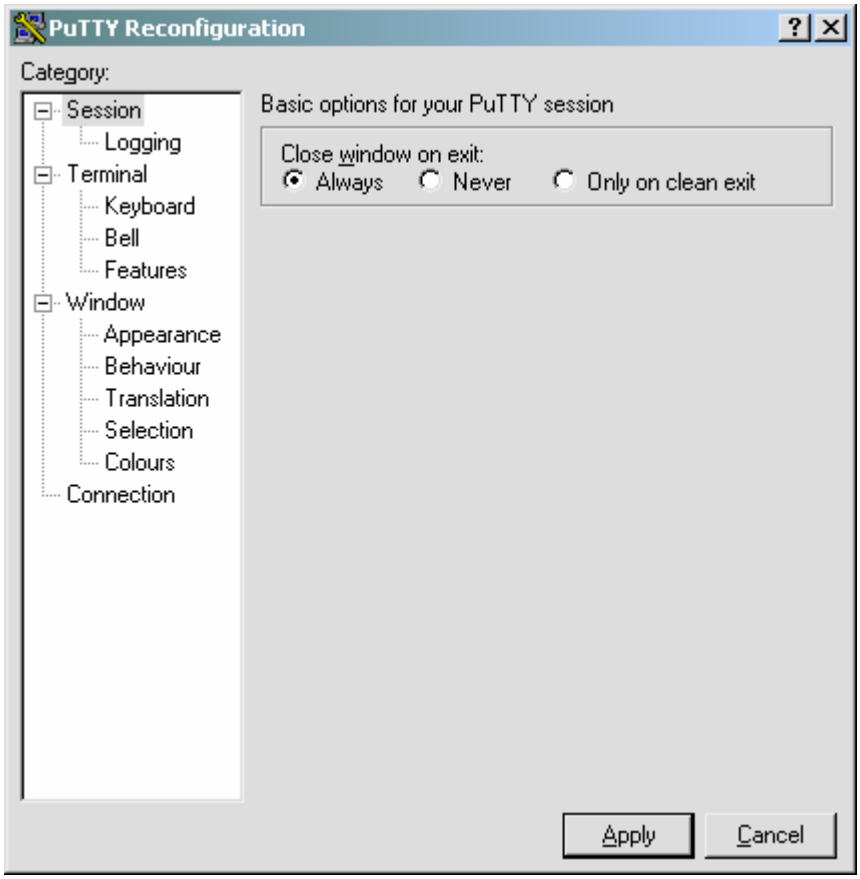
1. 运行 Putty 登陆到服务器



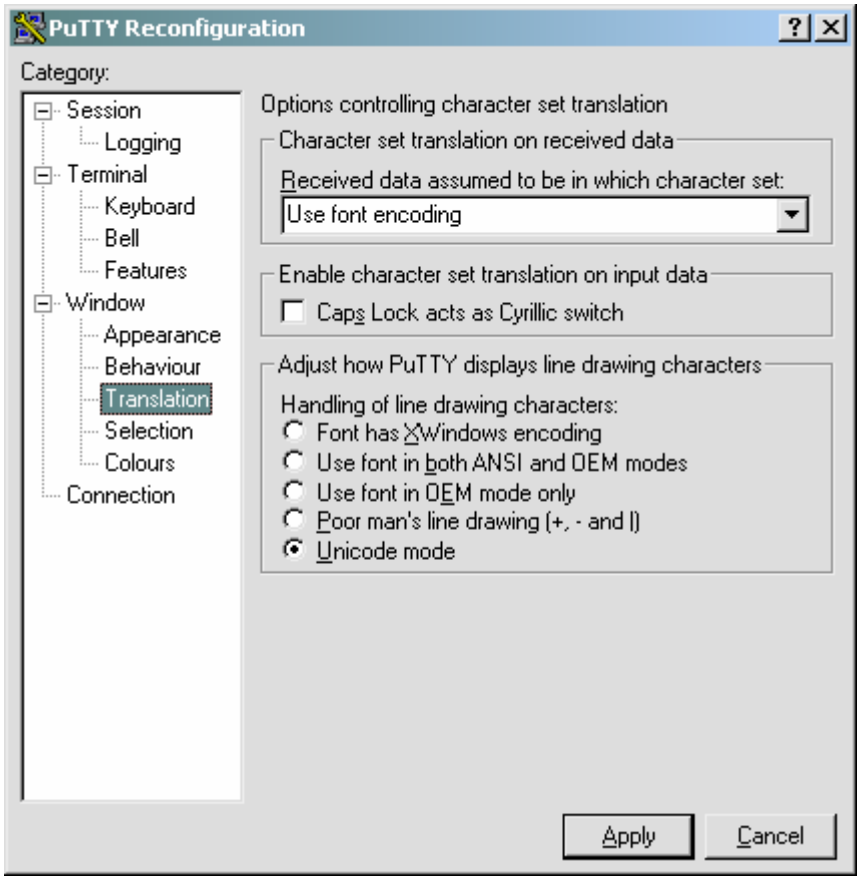
2. 在标题栏上单击右键



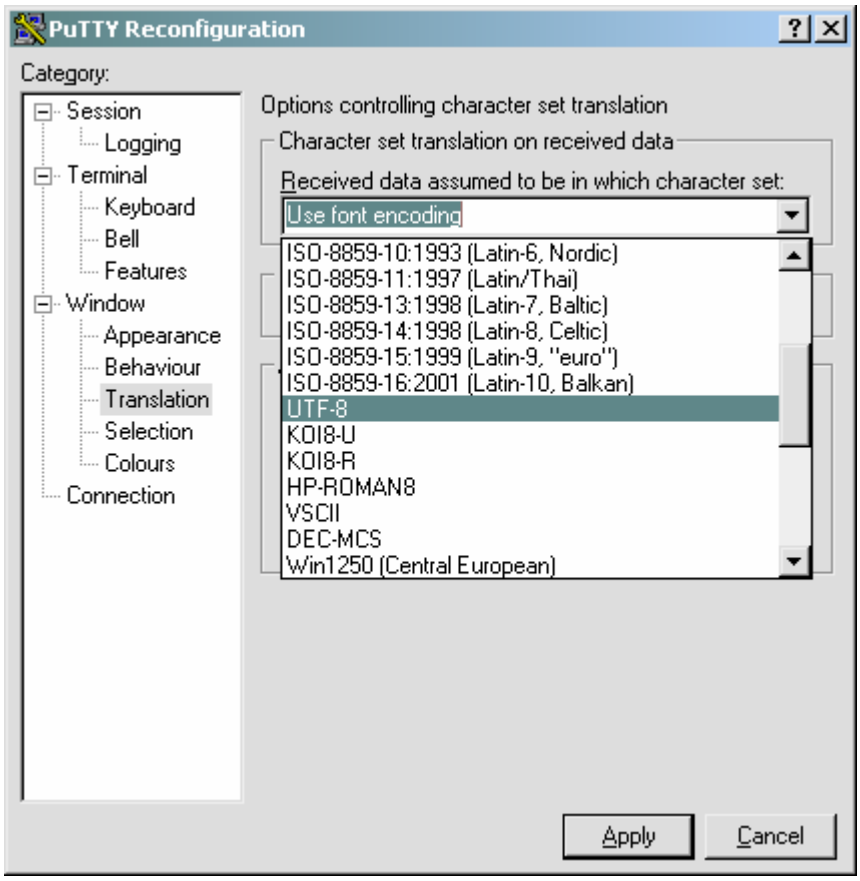
3. 选择“Change Settings…”菜单
4. 弹出下面对话框



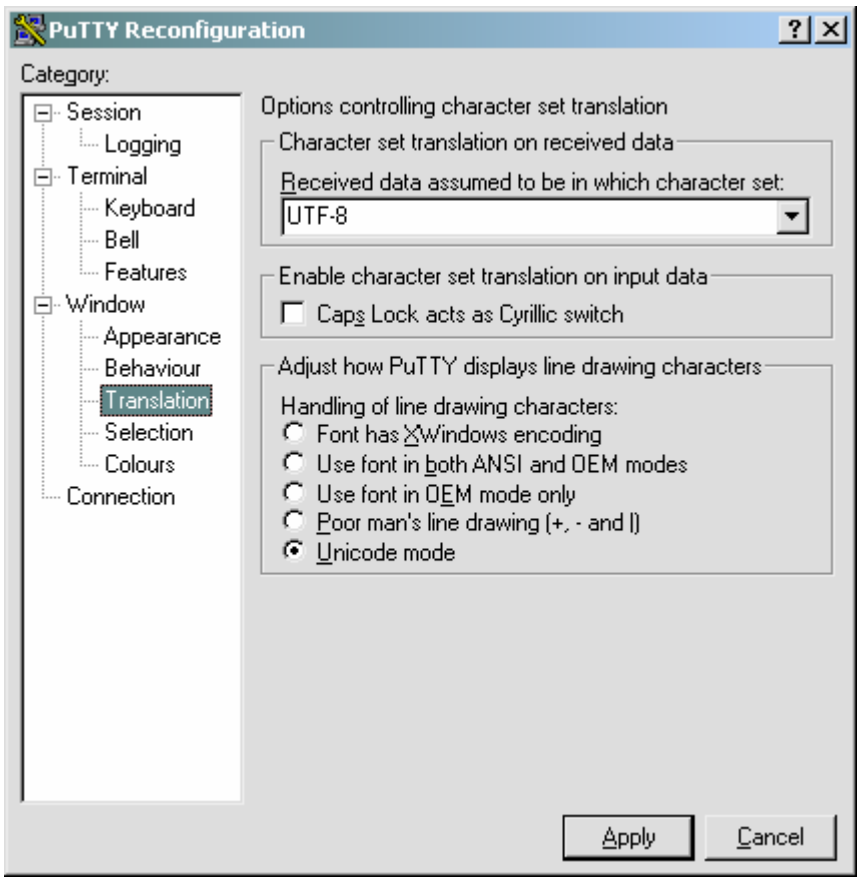
5. 选择 Category:列表下 Window 项中的 Translation
6. 弹出下面对话框



7. 在 Character set translation on received data 区域 Received data assumed tobe in which character set: 下拉列表中选择“UTF-8”编码

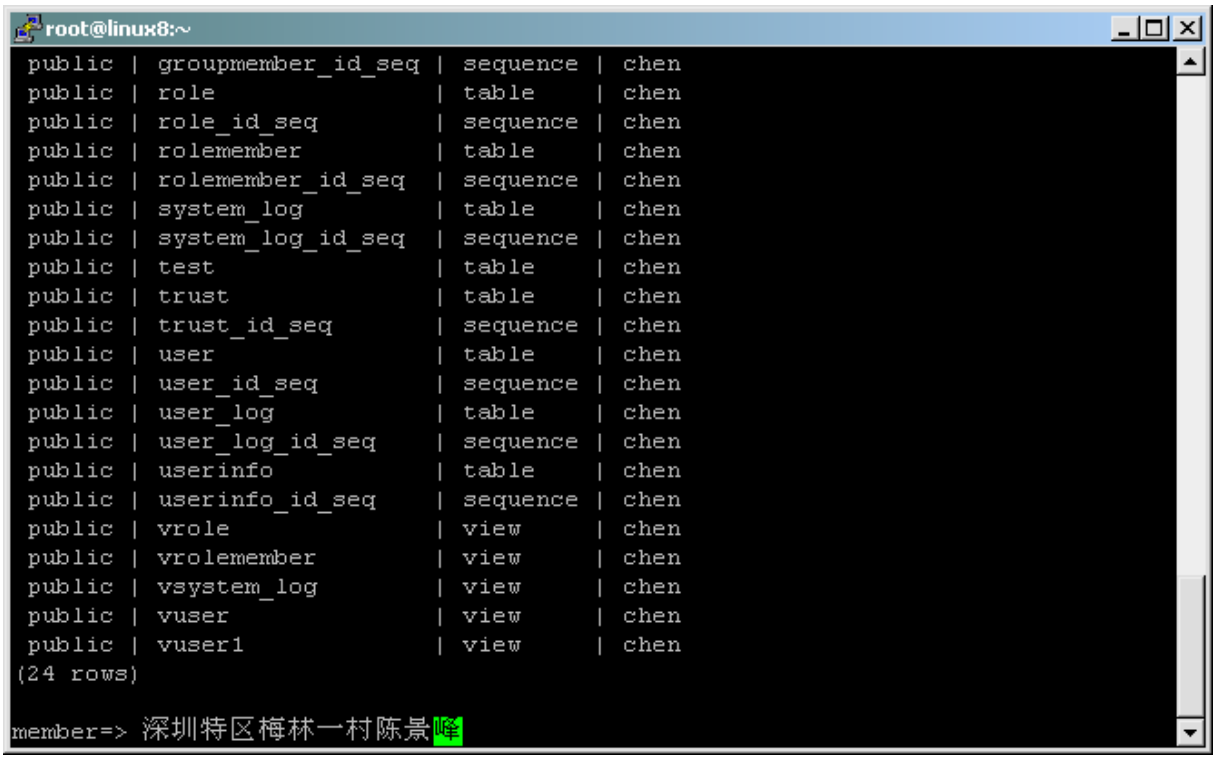


8. 弹出下面对话框



9. 单击应用按钮“Apply”

10. 在 putty 中输入中文。任何输入法都可以。



13.3 控制台下输入汉字

zhcon 是工作在 Linux 控制台下的高效双字节中/日/韩(CJK)虚拟终端，就像 DOS 环境中的 UCDOS 一样，为控制台(console)环境提供完整的双字节语言环境。

主页：http://zhcon.sourceforge.net/

项目网页：http://sourceforge.net/projects/zhcon/

下载 zhcon：

wget http://keihanna.dl.sourceforge.net/sourceforge/zhcon/zhcon-0.2.3.tar.gz

截图：

http://zhcon.sourceforge.net/images/scr_vim61.png

http://zhcon.sourceforge.net/images/scr_mc.gif

http://zhcon.sourceforge.net/images/scr_lynx.gif

http://zhcon.sourceforge.net/images/scr_emac.gif

http://zhcon.sourceforge.net/images/scr_bbs.gif

http://zhcon.sourceforge.net/images/scr_vi.gif

http://zhcon.sourceforge.net/images/scr_overspot.gif

安装：

```
[root@linux zhcon-0.2.3]# ./configure
[root@linux zhcon-0.2.3]# make
[root@linux zhcon-0.2.3]# make install
```

运行：

```
[root@linux root]#zhcon
```

13.4 PostgreSQL RPM 包安装后，为何没有 5432 端口

安装：

rpm -Uvh --nodeps postgresql-libs-?.?.?-1PGDG.i386.rpm

rpm -Uvh --nodeps postgresql-?.?.?-1PGDG.i386.rpm

rpm -Uvh --nodeps postgresql-devel-?.?.?-1PGDG.i386.rpm

rpm -Uvh --nodeps postgresql-server-?.?.?-1PGDG.i386.rpm

rpm -Uvh --nodeps postgresql-contrib-?.?.?-1PGDG.i386.rpm

rpm -Uvh --nodeps postgresql-docs-?.?.?-1PGDG.i386.rpm

rpm -Uvh --nodeps postgresql-jdbc-?.?.?-1PGDG.i386.rpm

rpm -Uvh --nodeps postgresql-pl-?.?.?-1PGDG.i386.rpm

rpm -Uvh --nodeps postgresql-python-?.?.?-1PGDG.i386.rpm

rpm -Uvh --nodeps postgresql-tcl-?.?.?-1PGDG.i386.rpm

rpm -Uvh --nodeps postgresql-test-?.?.?-1PGDG.i386.rpm

rpm -qa|grep post

查看：

```
[root@linux script]# rpm -qa|grep post
postgresql-contrib-7.3.3-1PGDG
postgresql-7.3.3-1PGDG
```

```
postgresql-tcl-7.3.3-1PGDG
postgresql-devel-7.3.3-1PGDG
postgresql-jdbc-7.3.3-1PGDG
postgresql-test-7.3.3-1PGDG
postgresql-server-7.3.3-1PGDG
postgresql-pl-7.3.3-1PGDG
postgresql-libs-7.3.3-1PGDG
postgresql-python-7.3.3-1PGDG
postfix-1.1.11-5
postgresql-docs-7.3.3-1PGDG
```

启动、初始化：

```
[root@linux root]# service postgresql start
[root@linux root]# su postgres
bash-2.05b$ createdb
bash-2.05b$ psql
Welcome to psql 7.3.3, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

postgres=#
```

使用“\q”退出数据库。

配置：

PostgreSQL 配置默认没有启用 TCP/IP 连接，开启 tcpip_socket 编辑 postgresql.conf 文件

```
[root@linux root]# cd /var/lib/pgsql/data/
[root@linux data]# vi postgresql.conf
#
#           Connection Parameters
#

tcpip_socket = true

#ssl = false

#max_connections = 32
#superuser_reserved_connections = 2

#port = 5432
#hostname_lookup = false
#show_source_port = false
:wq (保存)
```

tcpip_socket = false 改为 tcpip_socket = true 即可，端口默认 5432。如果你想使用其它端口配置 port = 5432 即可。

运行：

```
[root@linux root]# service postgresql restart
                                                          [  OK  ]
Starting postgresql service:                              [  OK  ]
```

查看：

使用端口扫描工具 NAMP 查看 PostgreSQL tcpip_socket 功能是否启用，

输出 5432/tcp open postgre 配置成功。

```
[root@linux root]# nmap localhost

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on localhost.localdomain (127.0.0.1):
(The 1592 ports scanned but not shown below are in state: closed)
Port        State       Service
22/tcp      open        ssh
25/tcp      open        smtp
80/tcp      open        http
111/tcp     open        sunrpc
139/tcp     open        netbios-ssn
389/tcp     open        ldap
2401/tcp    open        cvspserver
3306/tcp    open        mysql
5432/tcp    open        postgres

Nmap run completed -- 1 IP address (1 host up) scanned in 2 seconds
[root@linux root]#
```

测试：

```
[root@linux root]# psql -h127.0.0.1 -Upostgres
Welcome to psql 7.3.3, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

postgres=#
```

13.5 PHP 连接 PostgreSQL

编译 PHP 时请加 --with-pgsql 选项

13.6 汉字编码问题

传统汉字由两个字节组成（Hz，Hzk 国际 GB2312,GBK），最新版的 GB18030 兼容部分（GB2312，GBK）是 2 字节，其余的是 4 字节。其它一些过度产品还有 GB 2311、GB 11383、GB 12345、GB 13000.1、GB17000。采用 UTF-8 编码回多占用一些空间（英文 1 个字节，一个汉字需 3 个字节），但解决了国际化问题，UTF-8 兼容 GB2312、BIG5、EUC_CN、EUC_TW、GB18030 等多种国家的语言编码。

解决了国际化字符问题条件是创建数据库的编码必须是 UNICODE。UNICODE 中英文一个字母=中文一个汉字长度。

13.6.1 Jsp/Java

13.6.2 toChinese() 方法

我常在 BBS 上看到很多用户提问，Jsp/Java 读出表单、数据库中的汉字显示为“??????????”，多数用户都是使用类似下面的方法：

```
public static String toChinese(String strvalue){
    try{
        if(strvalue==null)
            return null;
        else{
            strvalue = new String(strvalue.getBytes("ISO8859_1"), "GBK");
            return strvalue;
        }
    }catch(Exception e){
            return null;
    }
}
```

13.6.3 Unicode (UTF-8) 完全解决方案

笔者不赞成使用这种方法，因为无论 GBK，还是 GB2312 只限于英文字母、汉字存储。我推荐使用 Unicode(UTF-8) 编码，可以存储任何数据。创建数据库编码采用 EUC_CN 或 Unicode 编码都可以（建议使用 Unicode）。Jsp 页面中要加入下面几行代码。加入代码之后不需要对字符（包括汉字）做认任转换处理。全部采用 UTF-8 编码存储数据。

13.6.3.1 setCharacterEncoding() 方案

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*" errorPage="error.jsp" %>
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset= gb2312">
<link href="../includes/style.css" rel="stylesheet" type="text/css">
</head>
```

改为：

```
<%@ page contentType="text/html; charset=utf-8" language="java" import="java.sql.*" errorPage="" %>
<jsp:include page="../includes/unicode.jsp" flush="true" />
<%@ page import="java.util.*"%>
<jsp:useBean id="sd" scope="page" class="ebusiness.btob.supplydemand"/>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<link href="../includes/style.css" rel="stylesheet" type="text/css">
</head>
<%
    String CharacterEncoding = "UTF-8";
    response.setContentType("text/html;charset=UTF-8");
    request.setCharacterEncoding(CharacterEncoding);
%>
```

```
String CharacterEncoding = "UTF-8";
request.setCharacterEncoding(CharacterEncoding);
```

设置提交出去的数据就是 utf-8 的..

创建数据库：

```
bash-2.05b$ createdb -E Unicode mydatabase
CREATE DATABASE
bash-2.05b$ psql -l
              List of databases
    Name       |   Owner    | Encoding
------------+----------+-----------
 member        | postgres | SQL_ASCII
```

```
  mydatabase | postgres | UNICODE
  postgres   | postgres | SQL_ASCII
  pureftpd   | pureftpd | EUC_CN
  site       | postgres | EUC_CN
  template0  | postgres | SQL_ASCII
  template1  | postgres | SQL_ASCII
(7 rows)

bash-2.05b$
```

或参考“用户权限”中的“脚本例子”小节。

```
CREATE GROUP person;
CREATE USER xuser WITH PASSWORD 'chen' VALID UNTIL '2003-12-1' IN GROUP person;
CREATE DATABASE xuser WITH OWNER = xuser TEMPLATE = template0 ENCODING = 'UNICODE';
```

Inc.jsp

```
<%
	response.addHeader("Expires","-1");
	response.addHeader("Pragma", "no-cache");
	response.addHeader("cache-control", "no-store");

	String CharacterEncoding = "UTF-8";
	response.setContentType("text/html;charset=UTF-8");
	request.setCharacterEncoding(CharacterEncoding);
%>
```

jsp 页面文件：

```
<%@ page contentType="text/html; charset=utf-8" language="java" import="java.sql.*" errorPage="error.jsp" %>
<%@ page import="java.util.*"%>
<%@ page import="net.xuser.util.*"%>
<%@ include file="include/inc.jsp" %>
<jsp:include page="include/privileges.jsp" flush="true" />
<jsp:useBean id="oGroup" scope="page" class="netkiller.member.group" />
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset= utf-8">
<title> </title>
<link href="include/style.css" rel="stylesheet" type="text/css">
</head>

<body>

</body>
```

</html>
上面的方法在跨页面提交时有问题，汉字还是出乱码。但提交给自身是没有问题的。

13.6.3.2 Web.xml Filter 过滤方案：

SetCharacterEncodingFilter.java

```
/*
 *                                                                 $Header:
/home/cvs/jakarta-tomcat-4.0/webapps/examples/WEB-INF/classes/filters/SetCharacterEncodingFilter.java
,v 1.1.2.1 2001/10/17 22:52:17 craigmcc Exp $
 * $Revision: 1.1.2.1 $
 * $Date: 2001/10/17 22:52:17 $
 *
 * ====================================================================
 *
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 1999-2001 The Apache Software Foundation.  All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. The end-user documentation included with the redistribution, if
 *    any, must include the following acknowlegement:
 *        "This product includes software developed by the
 *         Apache Software Foundation (http://www.apache.org/)."
 *    Alternately, this acknowlegement may appear in the software itself,
 *    if and wherever such third-party acknowlegements normally appear.
 *
 * 4. The names "The Jakarta Project", "Tomcat", and "Apache Software
 *    Foundation" must not be used to endorse or promote products derived
 *    from this software without prior written permission. For written
```

```
 *        permission, please contact apache@apache.org.
 *
 * 5. Products derived from this software may not be called "Apache"
 *        nor may "Apache" appear in their names without prior written
 *        permission of the Apache Group.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED.    IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * ====================================================================
 *
 * This software consists of voluntary contributions made by many
 * individuals on behalf of the Apache Software Foundation.    For more
 * information on the Apache Software Foundation, please see
 * <http://www.apache.org/>.
 *
 * [Additional notices, if required by prior licensing conditions]
 *
 */


package filters;


import java.io.IOException;
import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.UnavailableException;


/**
```

```
 * <p>Example filter that sets the character encoding to be used in parsing the
 * incoming request, either unconditionally or only if the client did not
 * specify a character encoding.  Configuration of this filter is based on
 * the following initialization parameters:</p>
 * <ul>
 * <li><strong>encoding</strong> - The character encoding to be configured
 *       for this request, either conditionally or unconditionally based on
 *       the <code>ignore</code> initialization parameter.  This parameter
 *       is required, so there is no default.</li>
 * <li><strong>ignore</strong> - If set to "true", any character encoding
 *       specified by the client is ignored, and the value returned by the
 *       <code>selectEncoding()</code> method is set.  If set to "false,
 *       <code>selectEncoding()</code> is called <strong>only</strong> if the
 *       client has not already specified an encoding.  By default, this
 *       parameter is set to "true".</li>
 * </ul>
 *
 * <p>Although this filter can be used unchanged, it is also easy to
 * subclass it and make the <code>selectEncoding()</code> method more
 * intelligent about what encoding to choose, based on characteristics of
 * the incoming request (such as the values of the <code>Accept-Language</code>
 * and <code>User-Agent</code> headers, or a value stashed in the current
 * user's session.</p>
 *
 * @author Craig McClanahan
 * @version $Revision: 1.1.2.1 $ $Date: 2001/10/17 22:52:17 $
 */

public class SetCharacterEncodingFilter implements Filter {


    // ----------------------------------------------------- Instance Variables


    /**
     * The default character encoding to set for requests that pass through
     * this filter.
     */
    protected String encoding = null;


    /**
     * The filter configuration object we are associated with.  If this value
     * is null, this filter instance is not currently configured.
```

```
 */
protected FilterConfig filterConfig = null;


/**
 * Should a character encoding specified by the client be ignored?
 */
protected boolean ignore = true;


// ---------------------------------------------------------- Public Methods


/**
 * Take this filter out of service.
 */
public void destroy() {

    this.encoding = null;
    this.filterConfig = null;

}


/**
 * Select and set (if specified) the character encoding to be used to
 * interpret request parameters for this request.
 *
 * @param request The servlet request we are processing
 * @param result The servlet response we are creating
 * @param chain The filter chain we are processing
 *
 * @exception IOException if an input/output error occurs
 * @exception ServletException if a servlet error occurs
 */
public void doFilter(ServletRequest request, ServletResponse response,
                     FilterChain chain)
throws IOException, ServletException {

    // Conditionally select and set the character encoding to be used
    if (ignore || (request.getCharacterEncoding() == null)) {
        String encoding = selectEncoding(request);
        if (encoding != null)
            request.setCharacterEncoding(encoding);
```

```
        }

    // Pass control on to the next filter
        chain.doFilter(request, response);

    }


    /**
     * Place this filter into service.
     *
     * @param filterConfig The filter configuration object
     */
    public void init(FilterConfig filterConfig) throws ServletException {

    this.filterConfig = filterConfig;
        this.encoding = filterConfig.getInitParameter("encoding");
        String value = filterConfig.getInitParameter("ignore");
        if (value == null)
            this.ignore = true;
        else if (value.equalsIgnoreCase("true"))
            this.ignore = true;
        else if (value.equalsIgnoreCase("yes"))
            this.ignore = true;
        else
            this.ignore = false;

    }


    // ---------------------------------------------------- Protected Methods


    /**
     * Select an appropriate character encoding to be used, based on the
     * characteristics of the current request and/or filter initialization
     * parameters.  If no character encoding should be set, return
     * <code>null</code>.
     * <p>
     * The default implementation unconditionally returns the value configured
     * by the <strong>encoding</strong> initialization parameter for this
     * filter.
     *
     * @param request The servlet request we are processing
```

```
     */
    protected String selectEncoding(ServletRequest request) {

        return (this.encoding);

    }


}
```

编译该文件

将 filters 包(目录)复制到 WEB-INF/classes 目录下。

编辑 WEB-INF 下的 web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>

<welcome-file-list>
<welcome-file>index.jsp</welcome-file>
</welcome-file-list>

<filter>
<filter-name>Set Character Encoding</filter-name>
<filter-class>filters.SetCharacterEncodingFilter</filter-class>
<init-param>
<param-name>encoding</param-name>
<param-value>utf-8</param-value>
</init-param>
<init-param>
<param-name>ignore</param-name>
<param-value>true</param-value>
</init-param>
</filter>

<filter-mapping>
<filter-name>Set Character Encoding</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>

</web-app>
```

汉字编码的解决方案完成。

13.6.3.3 Jdbc url charSet 方案

jdbc:postgresql://localhost:5432/database?charSet=UTF-8

13.6.4 PHP

13.6.4.1 set CLIENT_ENCODING TO 'GB18030';方案

PHP 连接 PostgreSQL 数据库，我是使用 PHPBB 提供的 Class，只要稍加改造就可很好支持 UNICODE。注意下面的红色代码！

```
<?php
   /***************************************************************************
    *                                  postgres7.php
    *                                -------------------
    *     begin                    : Saturday, Feb 13, 2001
    *     copyright                : (C) 2001 The phpBB Group
    *     email                    : supportphpbb.com
    *
    *     $Id: postgres7.php,v 1.19 2002/03/05 02:19:38 psotfx Exp $
    *
    ***************************************************************************/

/***************************************************************************
 *
 *     This program is free software; you can redistribute it and/or modify
 *     it under the terms of the GNU General Public License as published by
 *     the Free Software Foundation; either version 2 of the License, or
 *     (at your option) any later version.
 *
 ***************************************************************************/

if(!defined("SQL_LAYER"))
{

define("SQL_LAYER","postgresql");

class sql_db
{

      var $db_connect_id;
      var $query_result;
      var $in_transaction = 0;
```

```
	var $row = array();
	var $rowset = array();
	var $rownum = array();
	var $num_queries = 0;

	//
	// Constructor
	//
	function sql_db($sqlserver, $sqluser, $sqlpassword, $database, $persistency = true)
	{
		$this->connect_string = "";

		if( $sqluser )
		{
			$this->connect_string .= "user=$sqluser ";
		}

		if( $sqlpassword )
		{
			$this->connect_string .= "password=$sqlpassword ";
		}

		if( $sqlserver )
		{
			if( ereg(":", $sqlserver) )
			{
				list($sqlserver, $sqlport) = split(":", $sqlserver);
				$this->connect_string .= "host=$sqlserver port=$sqlport ";
			}
			else
			{
				if( $sqlserver != "localhost" )
				{
					$this->connect_string .= "host=$sqlserver ";
				}
			}
		}

		if( $database )
		{
			$this->dbname = $database;
			$this->connect_string .= "dbname=$database";
		}
```

```
        $this->persistency = $persistency;

        $this->db_connect_id = ( $this->persistency ) ? pg_pconnect($this->connect_string) : pg_connect($this->connect_string);

        return ( $this->db_connect_id ) ? $this->db_connect_id : false;
    }

    //
    // Other base methods
    //
    function sql_close()
    {
        if( $this->db_connect_id )
        {
            //
            // Commit any remaining transactions
            //
            if( $this->in_transaction )
            {
                @pg_exec($this->db_connect_id, "COMMIT");
            }

            if( $this->query_result )
            {
                @pg_freeresult($this->query_result);
            }

            return @pg_close($this->db_connect_id);
        }
        else
        {
            return false;
        }
    }

    //
    // Query method
    //
    function sql_query($query = "", $transaction = false)
    {
        //
        // Remove any pre-existing queries
        //
```

```
unset($this->query_result);
if( $query != "" )
{
    $this->num_queries++;

    $query = preg_replace("/LIMIT ([0-9]+),([ 0-9]+)/", "LIMIT \\2 OFFSET \\1", $query);
    $encoding    = " set CLIENT_ENCODING TO 'GB18030';";
    //$encoding    = "SET CLIENT_ENCODING TO 'GBK'";  也可以
    $this->query_result = @pg_exec($this->db_connect_id, $encoding);

    if( $transaction == BEGIN_TRANSACTION && !$this->in_transaction )
    {
        $this->in_transaction = TRUE;

        if( !@pg_exec($this->db_connect_id, "BEGIN") )
        {
            return false;
        }
    }

    $this->query_result = @pg_exec($this->db_connect_id, $query);
    if( $this->query_result )
    {
        if( $transaction == END_TRANSACTION )
        {
            $this->in_transaction = FALSE;

            if( !@pg_exec($this->db_connect_id, "COMMIT") )
            {
                @pg_exec($this->db_connect_id, "ROLLBACK");
                return false;
            }
        }

        $this->last_query_text[$this->query_result] = $query;
        $this->rownum[$this->query_result] = 0;

        unset($this->row[$this->query_result]);
        unset($this->rowset[$this->query_result]);

        return $this->query_result;
    }
    else
    {
```

```
                if( $this->in_transaction )
                {
                        @pg_exec($this->db_connect_id, "ROLLBACK");
                }
                $this->in_transaction = FALSE;

                return false;
        }
    }
    else
    {
        if( $transaction == END_TRANSACTION && $this->in_transaction )
        {
            $this->in_transaction = FALSE;

            if( !@pg_exec($this->db_connect_id, "COMMIT") )
            {
                @pg_exec($this->db_connect_id, "ROLLBACK");
                return false;
            }
        }

        return true;
    }
}

//
// Other query methods
//
function sql_numrows($query_id = 0)
{
    if( !$query_id )
    {
        $query_id = $this->query_result;
    }

    return ( $query_id ) ? @pg_numrows($query_id) : false;
}

function sql_numfields($query_id = 0)
{
    if( !$query_id )
    {
        $query_id = $this->query_result;
```

```
        }

        return ( $query_id ) ? @pg_numfields($query_id) : false;
    }

    function sql_fieldname($offset, $query_id = 0)
    {
        if( !$query_id )
        {
            $query_id = $this->query_result;
        }

        return ( $query_id ) ? @pg_fieldname($query_id, $offset) : false;
    }

    function sql_fieldtype($offset, $query_id = 0)
    {
        if( !$query_id )
        {
            $query_id = $this->query_result;
        }

        return ( $query_id ) ? @pg_fieldtype($query_id, $offset) : false;
    }

    function sql_fetchrow($query_id = 0)
    {
        if( !$query_id )
        {
            $query_id = $this->query_result;
        }

        if($query_id)
        {
            $this->row = @pg_fetch_array($query_id, $this->rownum[$query_id]);

            if( $this->row )
            {
                $this->rownum[$query_id]++;
                return $this->row;
            }
        }

        return false;
```

```
	}

	function sql_fetchrowset($query_id = 0)
	{
		if( !$query_id )
		{
			$query_id = $this->query_result;
		}

		if( $query_id )
		{
			unset($this->rowset[$query_id]);
			unset($this->row[$query_id]);
			$this->rownum[$query_id] = 0;

			while( $this->rowset = @pg_fetch_array($query_id, $this->rownum[$query_id], PGSQL_ASSOC) )
			{
				$result[] = $this->rowset;
				$this->rownum[$query_id]++;
			}

			return $result;
		}

		return false;
	}

	function sql_fetchfield($field, $row_offset=-1, $query_id = 0)
	{
		if( !$query_id )
		{
			$query_id = $this->query_result;
		}

		if( $query_id )
		{
			if( $row_offset != -1 )
			{
				$this->row = @pg_fetch_array($query_id, $row_offset, PGSQL_ASSOC);
			}
			else
			{
				if( $this->rownum[$query_id] )
```

```
                {
                    $this->row = @pg_fetch_array($query_id, $this->rownum[$query_id]-1, PGSQL_ASSOC);
                }
                else
                {
                    $this->row = @pg_fetch_array($query_id, $this->rownum[$query_id], PGSQL_ASSOC);

                    if( $this->row )
                    {
                        $this->rownum[$query_id]++;
                    }
                }
            }

            return $this->row[$field];
        }

        return false;
    }

    function sql_rowseek($offset, $query_id = 0)
    {

        if(!$query_id)
        {
            $query_id = $this->query_result;
        }

        if( $query_id )
        {
            if( $offset > -1 )
            {
                $this->rownum[$query_id] = $offset;
                return true;
            }
            else
            {
                return false;
            }
        }

        return false;
```

```
	}

	function sql_nextid()
	{
		$query_id = $this->query_result;

		if($query_id && $this->last_query_text[$query_id] != "")
		{
			if( preg_match("/^INSERT[\t\n ]+INTO[\t\n ]+([a-z0-9\_\-]+)/is", $this->last_query_text[$query_id], $tablename) )
			{
				$query = "SELECT currval('" . $tablename[1] . "_id_seq') AS last_value";
				$temp_q_id = @pg_exec($this->db_connect_id, $query);
				if( !$temp_q_id )
				{
					return false;
				}

				$temp_result = @pg_fetch_array($temp_q_id, 0, PGSQL_ASSOC);

				return ( $temp_result ) ? $temp_result['last_value'] : false;
			}
		}

		return false;
	}

	function sql_affectedrows($query_id = 0)
	{
		if( !$query_id )
		{
			$query_id = $this->query_result;
		}

		return ( $query_id ) ? @pg_cmdtuples($query_id) : false;
	}

	function sql_freeresult($query_id = 0)
	{
		if( !$query_id )
		{
			$query_id = $this->query_result;
		}
```

```
        return ( $query_id ) ? @pg_freeresult($query_id) : false;
    }

    function sql_error($query_id = 0)
    {
        if( !$query_id )
        {
            $query_id = $this->query_result;
        }

        $result['message'] = @pg_errormessage($this->db_connect_id);
        $result['code'] = -1;

        return $result;
    }

} // class ... db_sql

} // if ... defined

?>
```

注：SET CLIENT_ENCODING TO 'value'; ， value 是：

EUC_JP, SJIS, EUC_KR, UHC, JOHAB, EUC_CN, GBK, EUC_TW, BIG5, LATIN1 to LATIN10, ISO_8859_5, ISO_8859_6, ISO_8859_7, ISO_8859_8, WIN, ALT, KOI8, WIN1256, TCVN, WIN874, GB18030, WIN1250

例表中没有 GB2312，SET CLIENT_ENCODING TO 'GB2312'这样操作不会返回错误：

```
member=> SET CLIENT_ENCODING TO 'GB2312';
ERROR:  invalid value for option 'client_encoding': 'GB2312'
member=>
```

$encoding = " set CLIENT_ENCODING TO 'GB18030';"; ， set CLIENT_ENCODING TO 'GB18030' 尾部的“;”符号加不加都可以。

13.6.4.2 convert()方案

另一种方法是使用 convert() 函数：

select convert(描述,'UNICODE','GBK')as desc from 组;

select convert(组名 using utf_8_to_gb18030) from 组;

13.6.4.3 PHP iconv() 函数方案

在数据外部转码，使用 PHP 提供的 iconv()函数来完成。

注！安装编译 PHP 时要加入--with-iconv 模块。

```
function addUser($user,$passwd,$name,$nickname,$active="false",$email){
```

```
	global $db;
	$name = iconv( 'GB2312','UTF-8', $name );
	$sql = "insert into \"user\"(userid,passwd,name,nickname,active,email,question,answer)
values('$user','$passwd','$name','$nickname','$active','$email','question','answer')";
	if ( !($result = $db->sql_query($sql)) ){
		message_die(GENERAL_ERROR, 'Error in obtaining userdata', '', __LINE__, __FILE__, $sql);
	}
};

function getUserInfo($uid){
	global $db;
	$sql = "select * from \"user\" where userid='$uid'";

//	$codesql = "set CLIENT_ENCODING TO 'EUC_CN';";
//	$codesql = "set CLIENT_ENCODING TO 'BIG5';";
//	$codesql = "set CLIENT_ENCODING TO 'GB18030';";
	$db->sql_query($codesql);

	if ( !($result = $db->sql_query($sql)) ){
		message_die(GENERAL_ERROR, 'Could not \'select\' userdata', '', __LINE__, __FILE__, $sql);
	}
	$count=$db->sql_numrows();

	for ($i=0;$i<$count;$i++){
		if( $row = $db->sql_fetchrow($result) ){
			$tmp[0] = $row[0];
			$tmp[1] = $row[1];
			$tmp[2] = $row[2];
			$tmp[3] = iconv( 'UTF-8', 'GBK', $row[3] );
//			$tmp[3] = $row[3];
			$tmp[4] = $row[4];
			$tmp[5] = $row[5];
			$tmp[6] = $row[6];
			$tmp[7] = $row[7];
			$tmp[8] = $row[8];
			$tmp[9] = $row[9];
			$tmp[10] = $row[10];
		}
	}
	return $tmp;
};
```

13.6.4.4 在标准 I/O 上使用 Linux iconv 命令方案

Iconv 也是一个 Linux 提供的命令：

```
[root@linux script]# iconv --help
Usage: iconv [OPTION...] [FILE...]
Convert encoding of given files from one encoding to another.

 Input/Output format specification:
  -f, --from-code=NAME          encoding of original text
  -t, --to-code=NAME            encoding for output

 Information:
  -l, --list                    list all known coded character sets

 Output control:
  -c                            omit invalid characters from output
  -o, --output=FILE             output file
  -s, --silent                  suppress warnings
      --verbose                 print progress information

  -?, --help                    Give this help list
      --usage                   Give a short usage message
  -V, --version                 Print program version

Mandatory or optional arguments to long options are also mandatory or optional
for any corresponding short options.

Report bugs using the `glibcbug' script to <bugs@gnu.org>.
```

使用方法

```
[root@linux script]# iconv -f gb2312 -t big5 /tmp/gb.txt -o /tmp/big5.txt
```

如果你的 PHP 编译安装时没有加入--with-iconv 模块。你又不想重新编译它，可以通过管道来调用 LINUX iconv 命令，从标准 I/O 上（输入/输出）的反回数据。

```
function iconvs($from,$to,$gb){
    $fp = popen( "echo \"$gb\" | iconv -f $from -t $to", "r" );
    while (!feof($fp)) {
        $buffer .= fgets($fp, 4096);
    }
    pclose($fp);
    return $buffer;
}
```

```
    function gb18030_utf8($gb){
        $encode = $this->iconvs("GB18030","UTF-8",$gb);
        return $encode;
    }

    function utf8_gb18030($gb){
        $encode = $this->iconvs("UTF-8","GB18030",$gb);
        return $encode;
    }
```

iconv 支持字符集列表：

```
[root@linux script]# iconv -l
The following list contain all the coded character sets known.  This does
not necessarily mean that all combinations of these names can be used for
the FROM and TO command line parameters.  One coded character set can be
listed with several different names (aliases).

  437, 500, 500V1, 850, 851, 852, 855, 856, 857, 860, 861, 862, 863, 864, 865,
  866, 869, 874, 904, 1026, 1046, 1047, 8859_1, 8859_2, 8859_3, 8859_4, 8859_5,
  8859_6, 8859_7, 8859_8, 8859_9, 10646-1:1993, 10646-1:1993/UCS4,
  ANSI_X3.4-1968, ANSI_X3.4-1986, ANSI_X3.4, ANSI_X3.110-1983, ANSI_X3.110,
  ARABIC, ARABIC7, ARMSCII-8, ASCII, ASMO-708, ASMO_449, BALTIC, BIG-5,
  BIG-FIVE, BIG5-HKSCS, BIG5, BIG5HKSCS, BIGFIVE, BS_4730, CA, CN-BIG5, CN-GB,
  CN, CP-AR, CP-GR, CP-HU, CP037, CP038, CP273, CP274, CP275, CP278, CP280,
  CP281, CP282, CP284, CP285, CP290, CP297, CP367, CP420, CP423, CP424, CP437,
  CP500, CP737, CP775, CP813, CP819, CP850, CP851, CP852, CP855, CP856, CP857,
  CP860, CP861, CP862, CP863, CP864, CP865, CP866, CP868, CP869, CP870, CP871,
  CP874, CP875, CP880, CP891, CP903, CP904, CP905, CP912, CP915, CP916, CP918,
  CP920, CP922, CP930, CP932, CP933, CP935, CP936, CP937, CP939, CP949, CP950,
  CP1004, CP1026, CP1046, CP1047, CP1070, CP1079, CP1081, CP1084, CP1089,
  CP1124, CP1129, CP1132, CP1133, CP1160, CP1161, CP1162, CP1163, CP1164,
  CP1250, CP1251, CP1252, CP1253, CP1254, CP1255, CP1256, CP1257, CP1258,
  CP1361, CP10007, CPIBM861, CSA7-1, CSA7-2, CSASCII, CSA_T500-1983, CSA_T500,
  CSA_Z243.4-1985-1, CSA_Z243.4-1985-2, CSA_Z243.419851, CSA_Z243.419852,
  CSDECMCS, CSEBCDICATDE, CSEBCDICATDEA, CSEBCDICCAFR, CSEBCDICDKNO,
  CSEBCDICDKNOA, CSEBCDICES, CSEBCDICESA, CSEBCDICESS, CSEBCDICFISE,
  CSEBCDICFISEA, CSEBCDICFR, CSEBCDICIT, CSEBCDICPT, CSEBCDICUK, CSEBCDICUS,
  CSEUCKR, CSEUCPKDFMTJAPANESE, CSGB2312, CSHPROMAN8, CSIBM037, CSIBM038,
  CSIBM273, CSIBM274, CSIBM275, CSIBM277, CSIBM278, CSIBM280, CSIBM281,
  CSIBM284, CSIBM285, CSIBM290, CSIBM297, CSIBM420, CSIBM423, CSIBM424,
  CSIBM500, CSIBM851, CSIBM855, CSIBM856, CSIBM857, CSIBM860, CSIBM863,
  CSIBM864, CSIBM865, CSIBM866, CSIBM868, CSIBM869, CSIBM870, CSIBM871,
  CSIBM880, CSIBM891, CSIBM903, CSIBM904, CSIBM905, CSIBM918, CSIBM922,
  CSIBM930, CSIBM932, CSIBM933, CSIBM935, CSIBM937, CSIBM939, CSIBM943,
```

CSIBM1026, CSIBM1124, CSIBM1129, CSIBM1132, CSIBM1133, CSIBM1160, CSIBM1161, CSIBM1163, CSIBM1164, CSIBM11621162, CSISO4UNITEDKINGDOM, CSISO10SWEDISH, CSISO11SWEDISHFORNAMES, CSISO14JISC6220RO, CSISO15ITALIAN, CSISO16PORTUGESE, CSISO17SPANISH, CSISO18GREEK7OLD, CSISO19LATINGREEK, CSISO21GERMAN, CSISO25FRENCH, CSISO27LATINGREEK1, CSISO49INIS, CSISO50INIS8, CSISO51INISCYRILLIC, CSISO58GB1988, CSISO60DANISHNORWEGIAN, CSISO60NORWEGIAN1, CSISO61NORWEGIAN2, CSISO69FRENCH, CSISO84PORTUGUESE2, CSISO85SPANISH2, CSISO86HUNGARIAN, CSISO88GREEK7, CSISO89ASMO449, CSISO90, CSISO92JISC62991984B, CSISO99NAPLPS, CSISO103T618BIT, CSISO111ECMACYRILLIC, CSISO121CANADIAN1, CSISO122CANADIAN2, CSISO139CSN369103, CSISO141JUSIB1002, CSISO143IECP271, CSISO150, CSISO150GREEKCCITT, CSISO151CUBA, CSISO153GOST1976874, CSISO646DANISH, CSISO2022CN, CSISO2022JP, CSISO2022JP2, CSISO2022KR, CSISO2033, CSISO5427CYRILLIC, CSISO5427CYRILLIC1981, CSISO5428GREEK, CSISO10367BOX, CSISOLATIN1, CSISOLATIN2, CSISOLATIN3, CSISOLATIN4, CSISOLATIN5, CSISOLATIN6, CSISOLATINARABIC, CSISOLATINCYRILLIC, CSISOLATINGREEK, CSISOLATINHEBREW, CSKOI8R, CSKSC5636, CSMACINTOSH, CSNATSDANO, CSNATSSEFI, CSN_369103, CSPC8CODEPAGE437, CSPC775BALTIC, CSPC850MULTILINGUAL, CSPC862LATINHEBREW, CSPCP852, CSSHIFTJIS, CSUCS4, CSUNICODE, CUBA, CWI-2, CWI, CYRILLIC, DE, DEC-MCS, DEC, DECMCS, DIN_66003, DK, DS2089, DS_2089, E13B, EBCDIC-AT-DE-A, EBCDIC-AT-DE, EBCDIC-BE, EBCDIC-BR, EBCDIC-CA-FR, EBCDIC-CP-AR1, EBCDIC-CP-AR2, EBCDIC-CP-BE, EBCDIC-CP-CA, EBCDIC-CP-CH, EBCDIC-CP-DK, EBCDIC-CP-ES, EBCDIC-CP-FI, EBCDIC-CP-FR, EBCDIC-CP-GB, EBCDIC-CP-GR, EBCDIC-CP-HE, EBCDIC-CP-IS, EBCDIC-CP-IT, EBCDIC-CP-NL, EBCDIC-CP-NO, EBCDIC-CP-ROECE, EBCDIC-CP-SE, EBCDIC-CP-TR, EBCDIC-CP-US, EBCDIC-CP-WT, EBCDIC-CP-YU, EBCDIC-CYRILLIC, EBCDIC-DK-NO-A, EBCDIC-DK-NO, EBCDIC-ES-A, EBCDIC-ES-S, EBCDIC-ES, EBCDIC-FI-SE-A, EBCDIC-FI-SE, EBCDIC-FR, EBCDIC-GREEK, EBCDIC-INT, EBCDIC-INT1, EBCDIC-IS-FRISS, EBCDIC-IT, EBCDIC-JP-E, EBCDIC-JP-KANA, EBCDIC-PT, EBCDIC-UK, EBCDIC-US, EBCDICATDE, EBCDICATDEA, EBCDICCAFR, EBCDICDKNO, EBCDICDKNOA, EBCDICES, EBCDICESA, EBCDICESS, EBCDICFISE, EBCDICFISEA, EBCDICFR, EBCDICISFRISS, EBCDICIT, EBCDICPT, EBCDICUK, EBCDICUS, ECMA-114, ECMA-118, ECMA-128, ECMA-CYRILLIC, ECMACYRILLIC, ELOT_928, ES, ES2, EUC-CN, EUC-JISX0213, EUC-JP, EUC-KR, EUC-TW, EUCCN, EUCJP, EUCKR, EUCTW, FI, FR, GB, GB2312, GB13000, GB18030, GBK, GB_1988-80, GB_198880, GEORGIAN-ACADEMY, GEORGIAN-PS, GOST_19768-74, GOST_19768, GOST_1976874, GREEK-CCITT, GREEK, GREEK7-OLD, GREEK7, GREEK7OLD, GREEK8, GREEKCCITT, HEBREW, HP-ROMAN8, HPROMAN8, HU, IBM-856, IBM-922, IBM-930, IBM-932, IBM-933, IBM-935, IBM-937, IBM-939, IBM-943, IBM-1046, IBM-1124, IBM-1129, IBM-1132, IBM-1133, IBM-1160, IBM-1161, IBM-1162, IBM-1163, IBM-1164, IBM037, IBM038, IBM256, IBM273, IBM274, IBM275, IBM277, IBM278, IBM280, IBM281, IBM284, IBM285, IBM290, IBM297, IBM367, IBM420, IBM423, IBM424, IBM437, IBM500, IBM775, IBM813, IBM819, IBM850, IBM851, IBM852, IBM855, IBM856, IBM857, IBM860, IBM861, IBM862, IBM863, IBM864, IBM865, IBM866, IBM868, IBM869, IBM870, IBM871, IBM874, IBM875, IBM880, IBM891, IBM903, IBM904, IBM905,

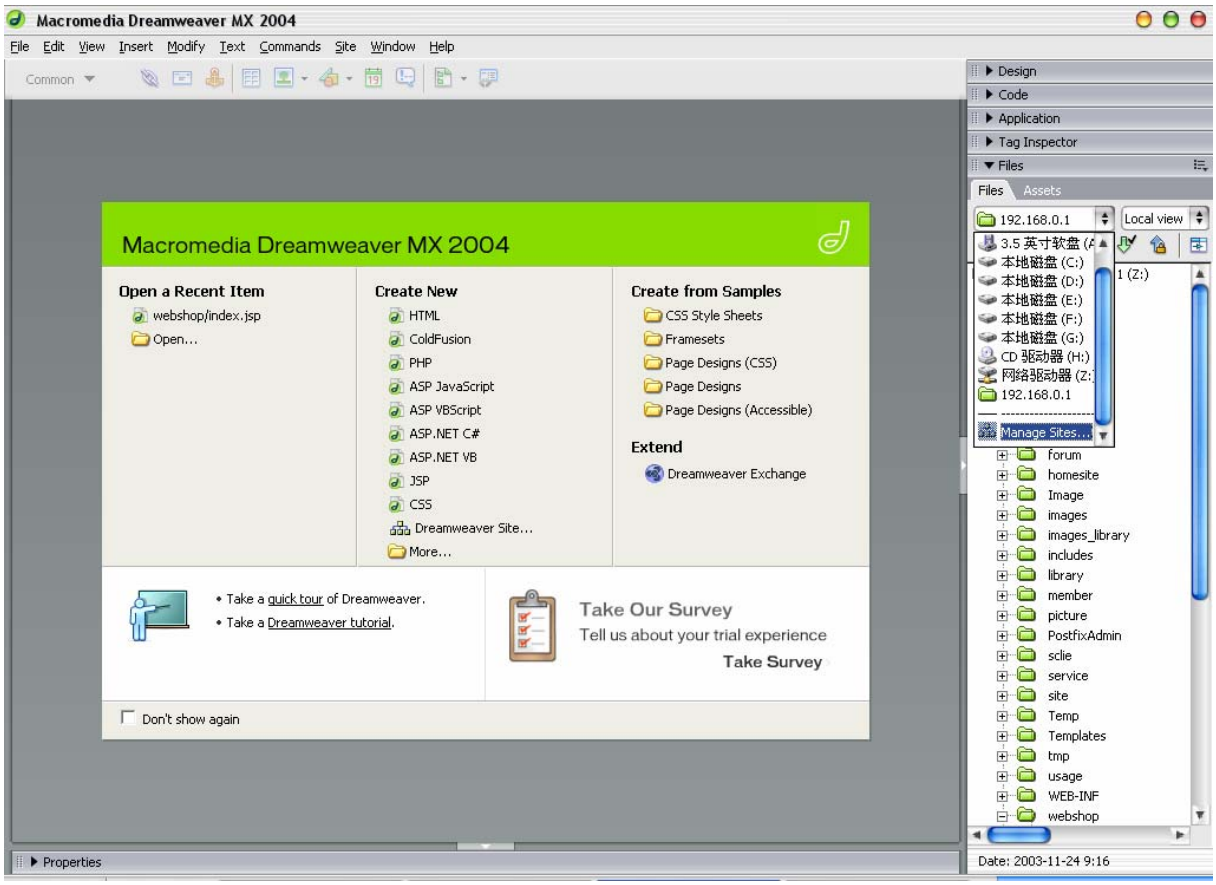
IBM912, IBM915, IBM916, IBM918, IBM920, IBM922, IBM930, IBM932, IBM933,
IBM935, IBM937, IBM939, IBM943, IBM1004, IBM1026, IBM1046, IBM1047, IBM1089,
IBM1124, IBM1129, IBM1132, IBM1133, IBM1160, IBM1161, IBM1162, IBM1163,
IBM1164, IEC_P27-1, IEC_P271, INIS-8, INIS-CYRILLIC, INIS, INIS8,
INISCYRILLIC, ISIRI-3342, ISIRI3342, ISO-2022-CN-EXT, ISO-2022-CN,
ISO-2022-JP-2, ISO-2022-JP-3, ISO-2022-JP, ISO-2022-KR, ISO-8859-1,
ISO-8859-2, ISO-8859-3, ISO-8859-4, ISO-8859-5, ISO-8859-6, ISO-8859-7,
ISO-8859-8, ISO-8859-9, ISO-8859-10, ISO-8859-11, ISO-8859-13, ISO-8859-14,
ISO-8859-15, ISO-8859-16, ISO-10646, ISO-10646/UCS2, ISO-10646/UCS4,
ISO-10646/UTF-8, ISO-10646/UTF8, ISO-CELTIC, ISO-IR-4, ISO-IR-6, ISO-IR-8-1,
ISO-IR-9-1, ISO-IR-10, ISO-IR-11, ISO-IR-14, ISO-IR-15, ISO-IR-16, ISO-IR-17,
ISO-IR-18, ISO-IR-19, ISO-IR-21, ISO-IR-25, ISO-IR-27, ISO-IR-37, ISO-IR-49,
ISO-IR-50, ISO-IR-51, ISO-IR-54, ISO-IR-55, ISO-IR-57, ISO-IR-60, ISO-IR-61,
ISO-IR-69, ISO-IR-84, ISO-IR-85, ISO-IR-86, ISO-IR-88, ISO-IR-89, ISO-IR-90,
ISO-IR-92, ISO-IR-98, ISO-IR-99, ISO-IR-100, ISO-IR-101, ISO-IR-103,
ISO-IR-109, ISO-IR-110, ISO-IR-111, ISO-IR-121, ISO-IR-122, ISO-IR-126,
ISO-IR-127, ISO-IR-138, ISO-IR-139, ISO-IR-141, ISO-IR-143, ISO-IR-144,
ISO-IR-148, ISO-IR-150, ISO-IR-151, ISO-IR-153, ISO-IR-155, ISO-IR-156,
ISO-IR-157, ISO-IR-166, ISO-IR-179, ISO-IR-193, ISO-IR-197, ISO-IR-199,
ISO-IR-203, ISO-IR-209, ISO-IR-226, ISO646-CA, ISO646-CA2, ISO646-CN,
ISO646-CU, ISO646-DE, ISO646-DK, ISO646-ES, ISO646-ES2, ISO646-FI, ISO646-FR,
ISO646-FR1, ISO646-GB, ISO646-HU, ISO646-IT, ISO646-JP-OCR-B, ISO646-JP,
ISO646-KR, ISO646-NO, ISO646-NO2, ISO646-PT, ISO646-PT2, ISO646-SE,
ISO646-SE2, ISO646-US, ISO646-YU, ISO2022CN, ISO2022CNEXT, ISO2022JP,
ISO2022JP2, ISO2022KR, ISO6937, ISO8859-1, ISO8859-2, ISO8859-3, ISO8859-4,
ISO8859-5, ISO8859-6, ISO8859-7, ISO8859-8, ISO8859-9, ISO8859-10,
ISO8859-11, ISO8859-13, ISO8859-14, ISO8859-15, ISO8859-16, ISO88591,
ISO88592, ISO88593, ISO88594, ISO88595, ISO88596, ISO88597, ISO88598,
ISO88599, ISO885910, ISO885911, ISO885913, ISO885914, ISO885915, ISO885916,
ISO_646.IRV:1991, ISO_2033-1983, ISO_2033, ISO_5427-EXT, ISO_5427,
ISO_5427:1981, ISO_5427EXT, ISO_5428, ISO_5428:1980, ISO_6937-2,
ISO_6937-2:1983, ISO_6937, ISO_6937:1992, ISO_8859-1, ISO_8859-1:1987,
ISO_8859-2, ISO_8859-2:1987, ISO_8859-3, ISO_8859-3:1988, ISO_8859-4,
ISO_8859-4:1988, ISO_8859-5, ISO_8859-5:1988, ISO_8859-6, ISO_8859-6:1987,
ISO_8859-7, ISO_8859-7:1987, ISO_8859-8, ISO_8859-8:1988, ISO_8859-9,
ISO_8859-9:1989, ISO_8859-10, ISO_8859-10:1992, ISO_8859-14,
ISO_8859-14:1998, ISO_8859-15:1998, ISO_9036, ISO_10367-BOX, ISO_10367BOX,
ISO_69372, IT, JIS_C6220-1969-RO, JIS_C6229-1984-B, JIS_C62201969RO,
JIS_C62291984B, JOHAB, JP-OCR-B, JP, JS, JUS_I.B1.002, KOI-7, KOI-8, KOI8-R,
KOI8-T, KOI8-U, KOI8, KOI8R, KOI8U, KSC5636, L1, L2, L3, L4, L5, L6, L7, L8,
L10, LATIN-GREEK-1, LATIN-GREEK, LATIN1, LATIN2, LATIN3, LATIN4, LATIN5,
LATIN6, LATIN7, LATIN8, LATIN10, LATINGREEK, LATINGREEK1, MAC-CYRILLIC,
MAC-IS, MAC-SAMI, MAC-UK, MAC, MACCYRILLIC, MACINTOSH, MACIS, MACUK,
MACUKRAINIAN, MS-ANSI, MS-ARAB, MS-CYRL, MS-EE, MS-GREEK, MS-HEBR,

```
MS-MAC-CYRILLIC, MS-TURK, MSCP949, MSCP1361, MSMACCYRILLIC, MSZ_7795.3,
MS_KANJI, NAPLPS, NATS-DANO, NATS-SEFI, NATSDANO, NATSSEFI, NC_NC0010,
NC_NC00-10, NC_NC00-10:81, NF_Z_62-010, NF_Z_62-010_(1973), NF_Z_62-010_1973,
NF_Z_62010, NF_Z_62010_1973, NO, NO2, NS_4551-1, NS_4551-2, NS_45511,
NS_45512, OS2LATIN1, OSF00010001, OSF00010002, OSF00010003, OSF00010004,
OSF00010005, OSF00010006, OSF00010007, OSF00010008, OSF00010009, OSF0001000A,
OSF00010020, OSF00010100, OSF00010101, OSF00010102, OSF00010104, OSF00010105,
OSF00010106, OSF00030010, OSF0004000A, OSF0005000A, OSF05010001, OSF100201A4,
OSF100201A8, OSF100201B5, OSF100201F4, OSF100203B5, OSF1002011C, OSF1002011D,
OSF1002035D, OSF1002035E, OSF1002035F, OSF1002036B, OSF1002037B, OSF10010001,
OSF10020025, OSF10020111, OSF10020115, OSF10020116, OSF10020118, OSF10020122,
OSF10020129, OSF10020352, OSF10020354, OSF10020357, OSF10020359, OSF10020360,
OSF10020364, OSF10020365, OSF10020366, OSF10020367, OSF10020370, OSF10020387,
OSF10020388, OSF10020396, OSF10020402, OSF10020417, PT, PT2, R8, ROMAN8, SE,
SE2, SEN_850200_B, SEN_850200_C, SHIFT-JIS, SHIFT_JIS, SHIFT_JISX0213, SJIS,
SS636127, ST_SEV_358-88, T.61-8BIT, T.61, T.618BIT, TCVN-5712, TCVN,
TCVN5712-1, TCVN5712-1:1993, TIS-620, TIS620-0, TIS620.2529-1, TIS620.2533-0,
TIS620, TS-5881, UCS-2, UCS-2BE, UCS-2LE, UCS-4, UCS-4BE, UCS-4LE, UCS2,
UCS4, UHC, UJIS, UK, UNICODE, UNICODEBIG, UNICODELITTLE, US-ASCII, US, UTF-7,
UTF-8, UTF-16, UTF-16BE, UTF-16LE, UTF-32, UTF-32BE, UTF-32LE, UTF7, UTF8,
UTF16, UTF16BE, UTF16LE, UTF32, UTF32BE, UTF32LE, VISCII, WCHAR_T,
WIN-SAMI-2, WINBALTRIM, WINDOWS-1250, WINDOWS-1251, WINDOWS-1252,
WINDOWS-1253, WINDOWS-1254, WINDOWS-1255, WINDOWS-1256, WINDOWS-1257,
WINDOWS-1258, WINSAMI2, WS2, YU
[root@linux script]#
```

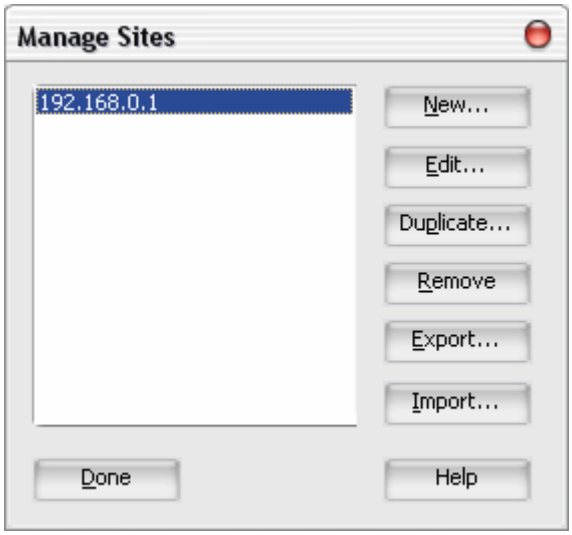
强大的 iconv 是由 C 语言中的 iconv()函数实现的

13.7 Macromedia Dreamweaver MX 2004 JSP 开发环境的配置

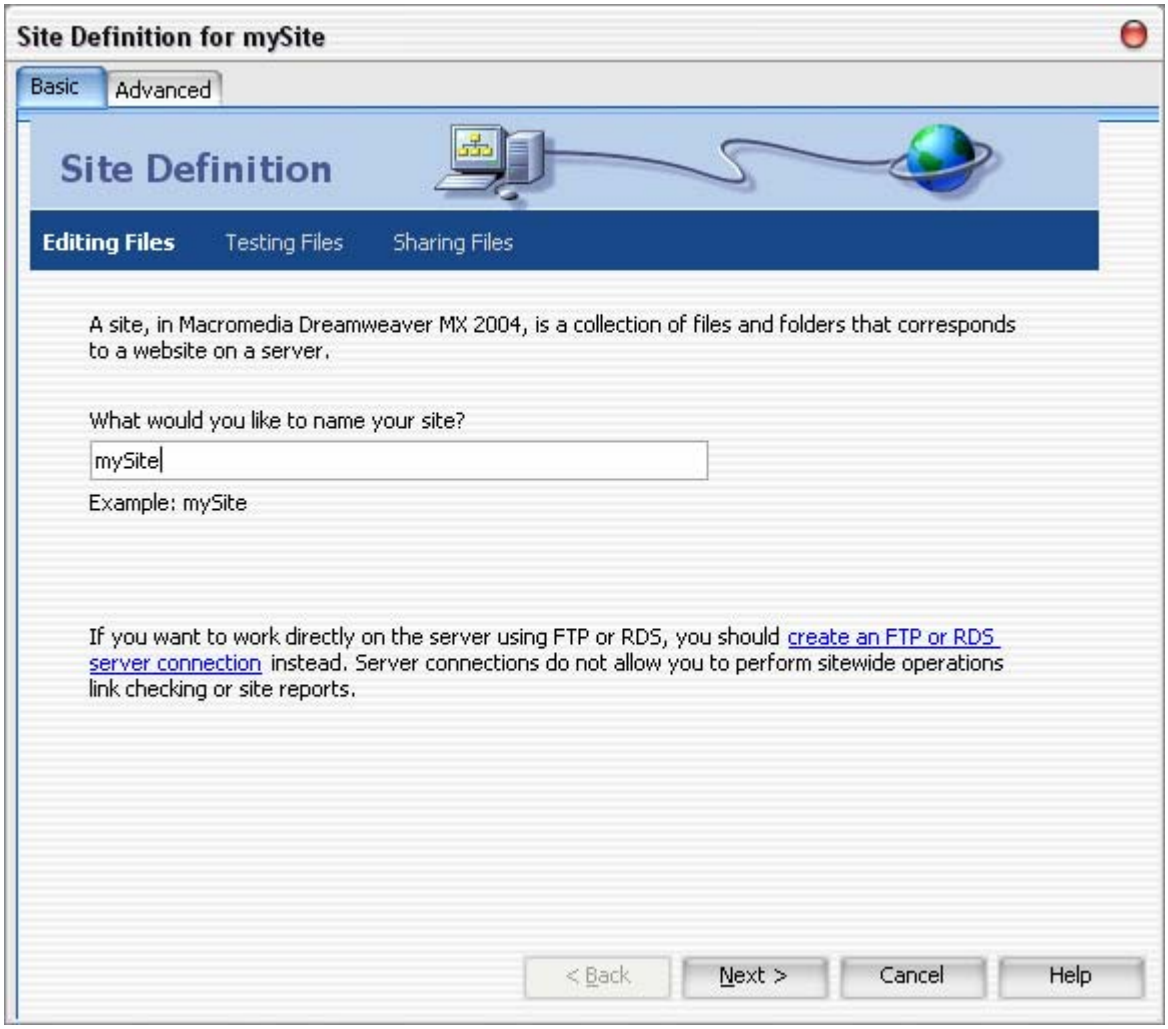
新建站点：



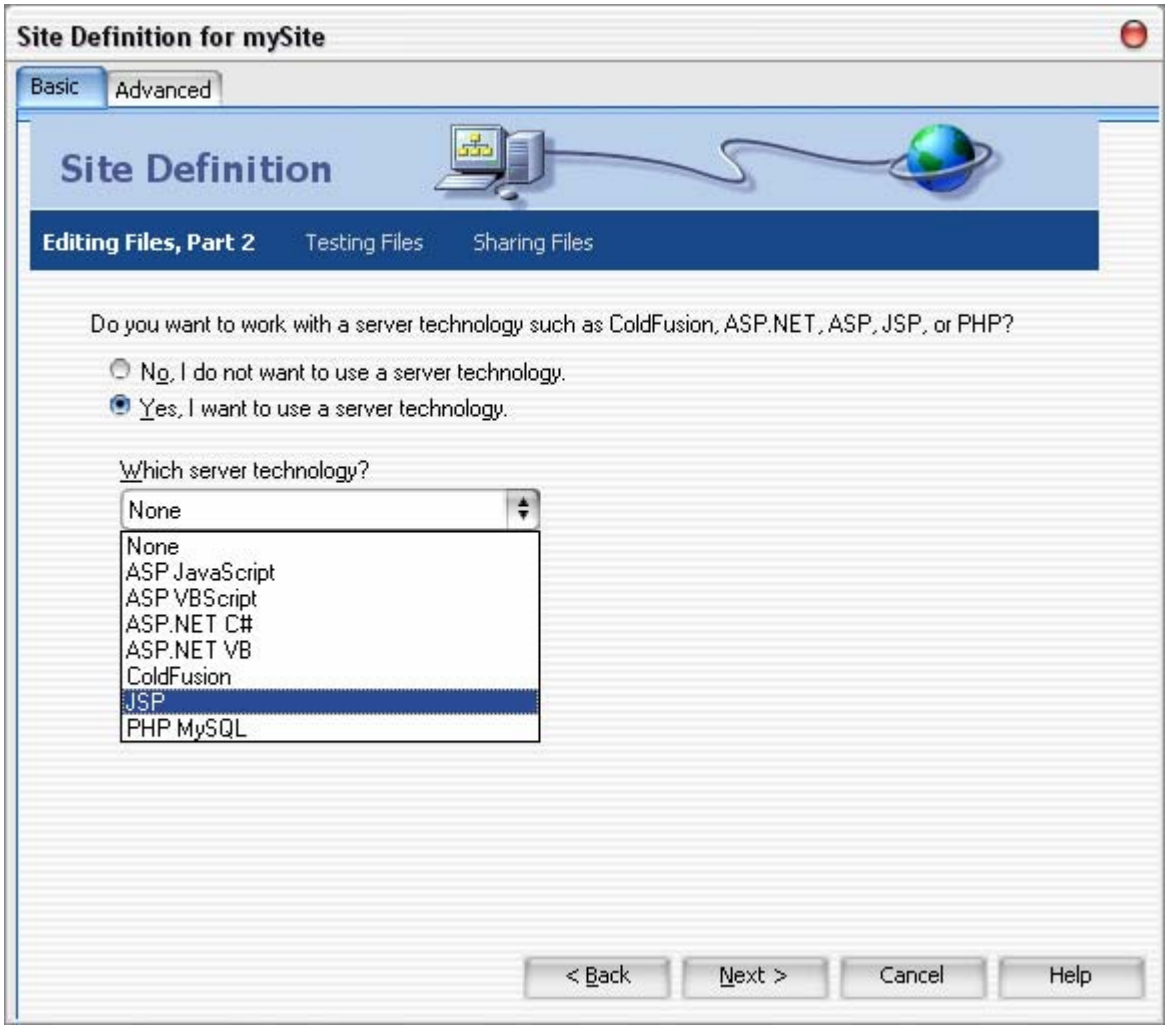
1　选择 Manage Sites…



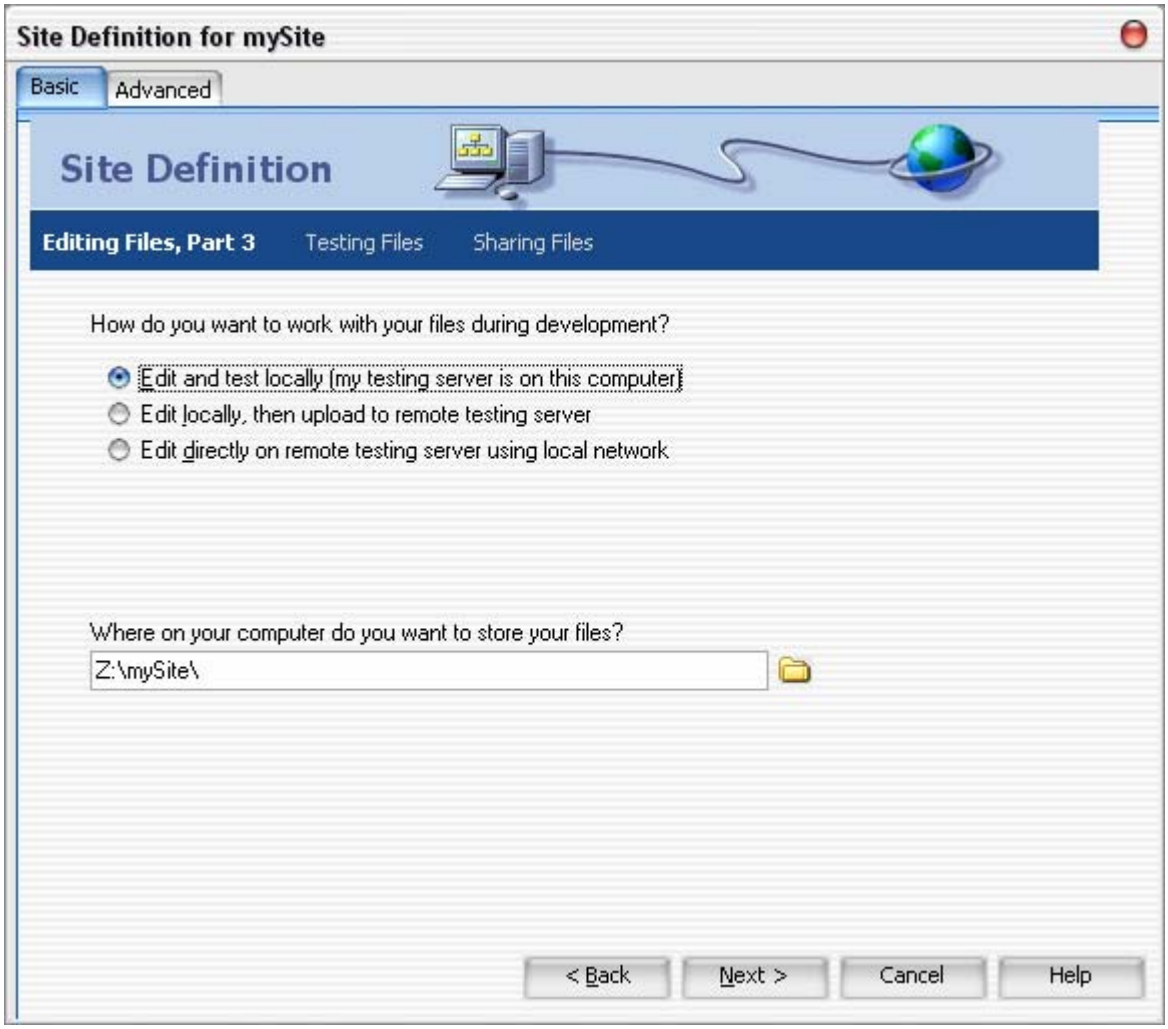
2　单击 New…，然后选择 Site



3　输入站点名称：mySite，下一步



4　选择我想使用服务器技术，在列表中选择“JSP”，下一步

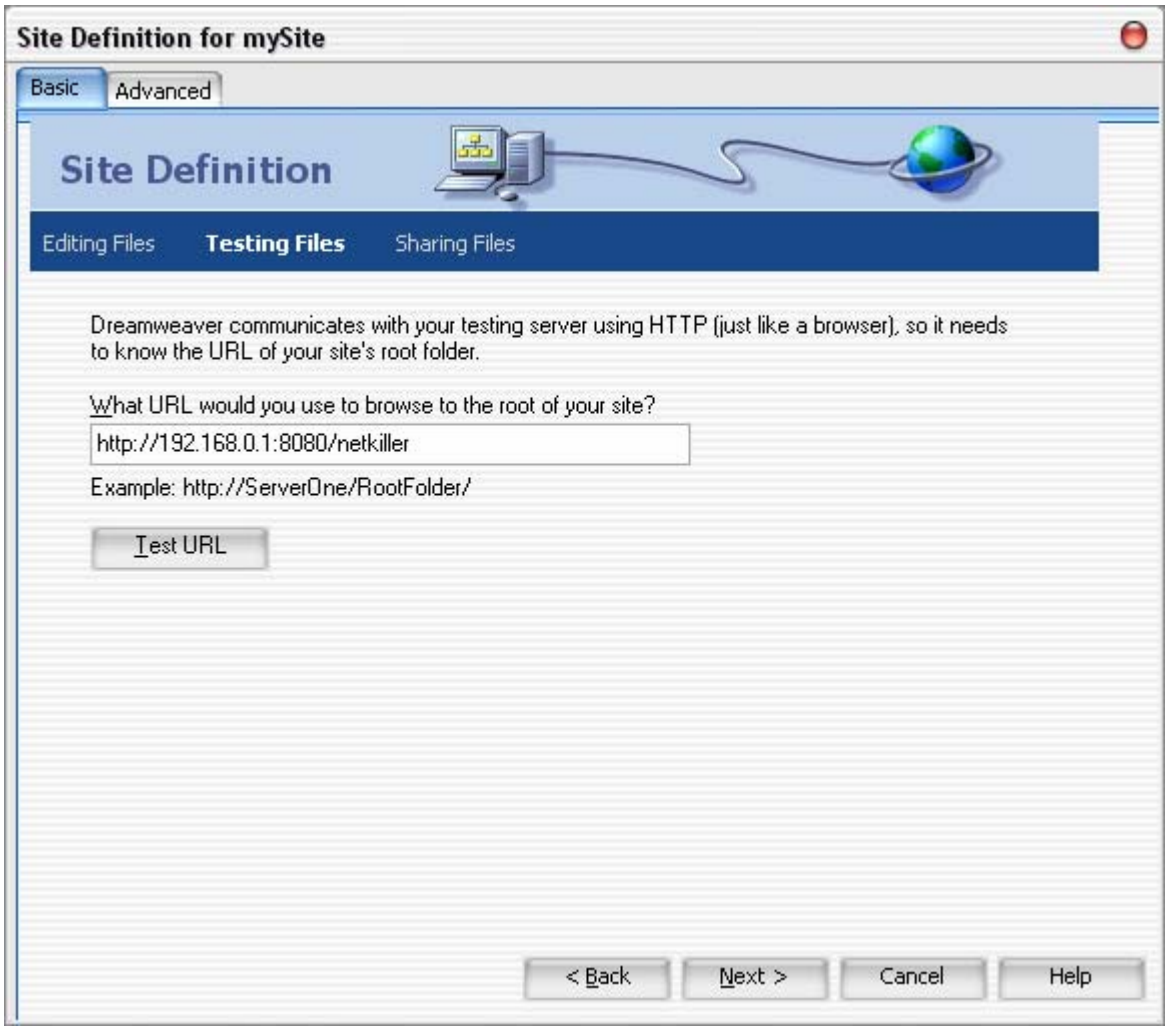


5　选择，编辑、测试一台计算机上。选择一个站点目录。下一步

注：我的 z:是使用 net use z: \\192.168.0.1\web 映射的。\\192.168.0.1\web 是 linux 服务器，web 是 Samba 共享/var/www/html/webapps， 而/var/www/html/webapps 是 netkiller.xml <Context path="/netkiller" docBase="/var/www/html/webapps" reloadable="true" debug="0" privileged="true">

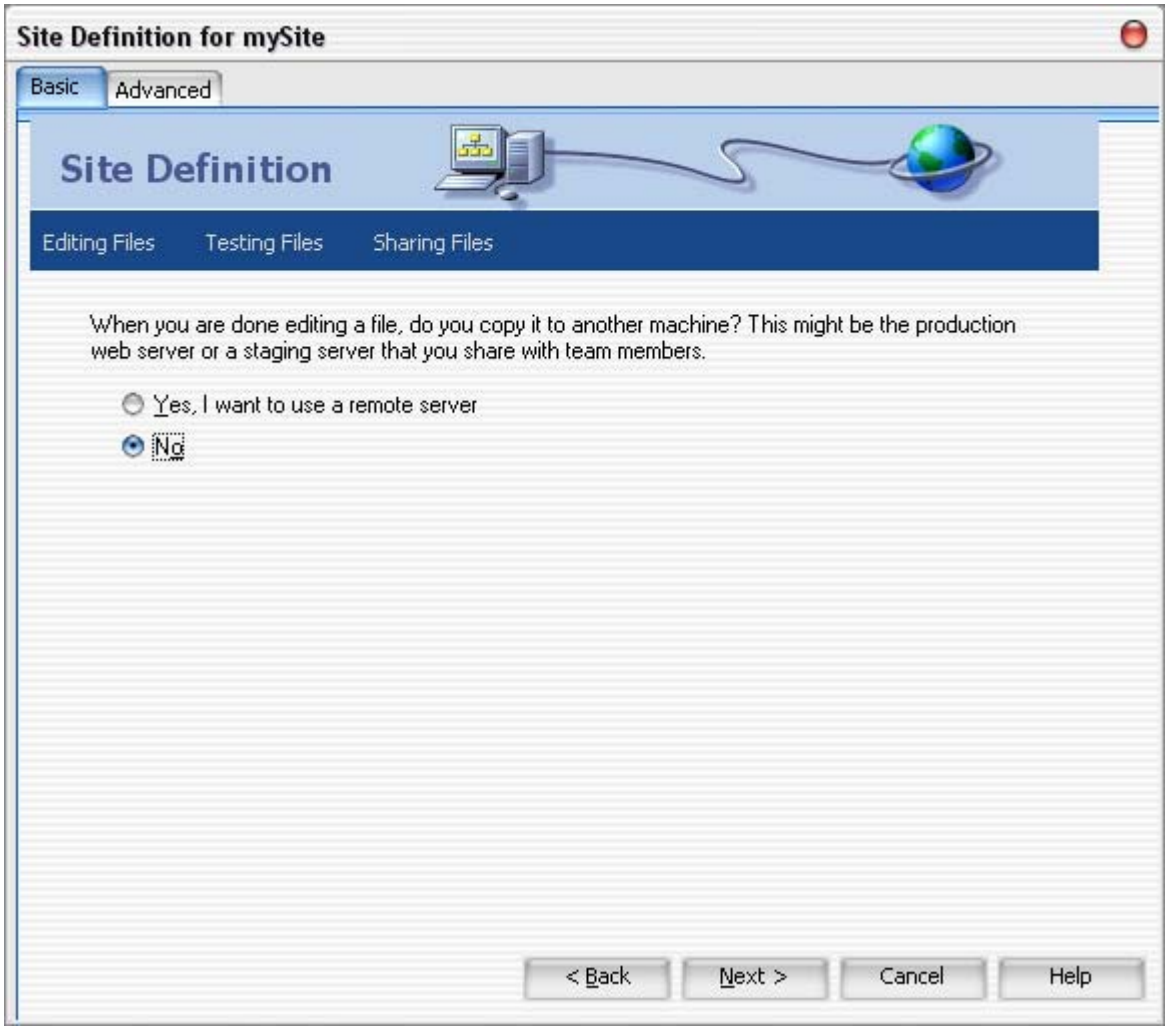
```
[root@linux root]# cat /etc/samba/smb.conf
[web]
   comment = PC Directories
   path = /var/www/html/webapps
   public = yes
   writable = yes

[root@linux root]# cat /usr/local/jakarta-tomcat/webapps/netkiller.xml
<!--

      Context configuration file for the Tomcat Administration Web App

      $Id: admin.xml,v 1.3 2002/07/23 12:12:15 remm Exp $
```

```
-->


<Context path="/netkiller" docBase="/var/www/html/webapps" reloadable="true"
        debug="0" privileged="true">

  <!-- Uncomment this Valve to limit access to the Admin app to localhost
    for obvious security reasons. Allow may be a comma-separated list of
    hosts (or even regular expressions).
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
     allow="127.0.0.1"/>
  -->

  <Logger className="org.apache.catalina.logger.FileLogger"
             prefix="localhost_netkiller_log." suffix=".txt"
          timestamp="true"/>

</Context>
```

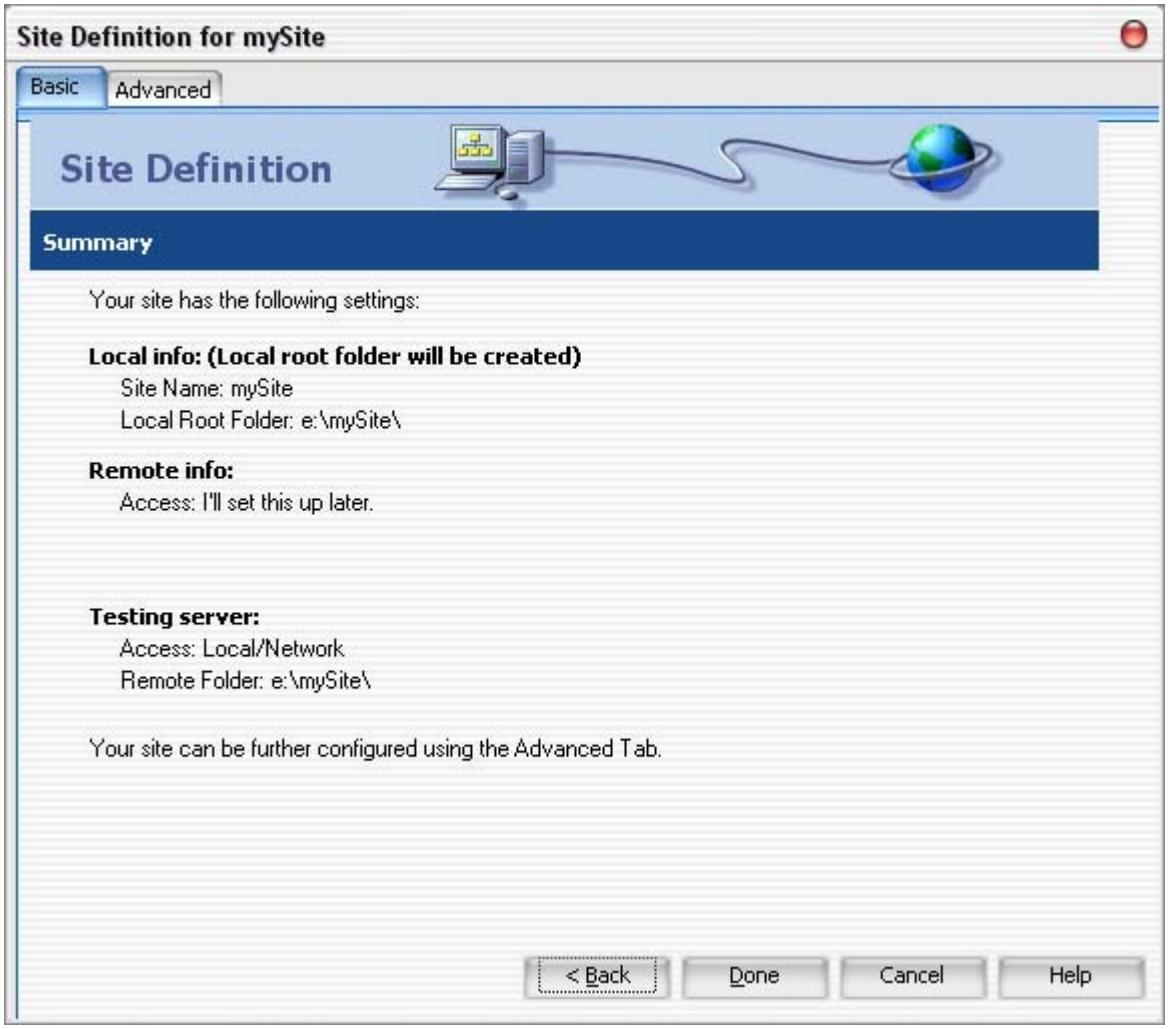


6 URL 输入: http://192.168.0.1:8080/netkiller，单击“Test User”

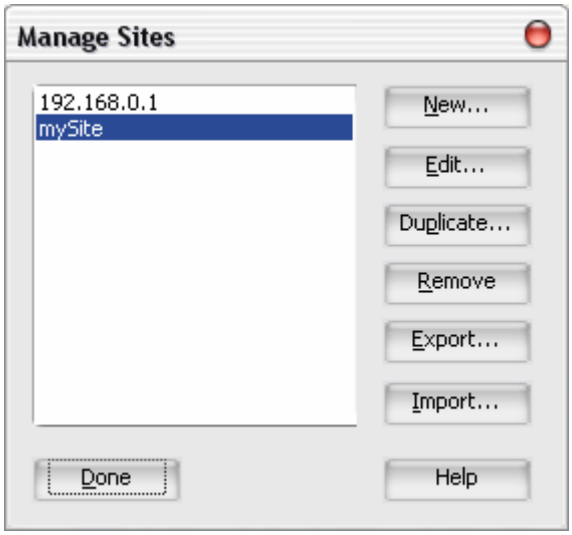
7　下一步



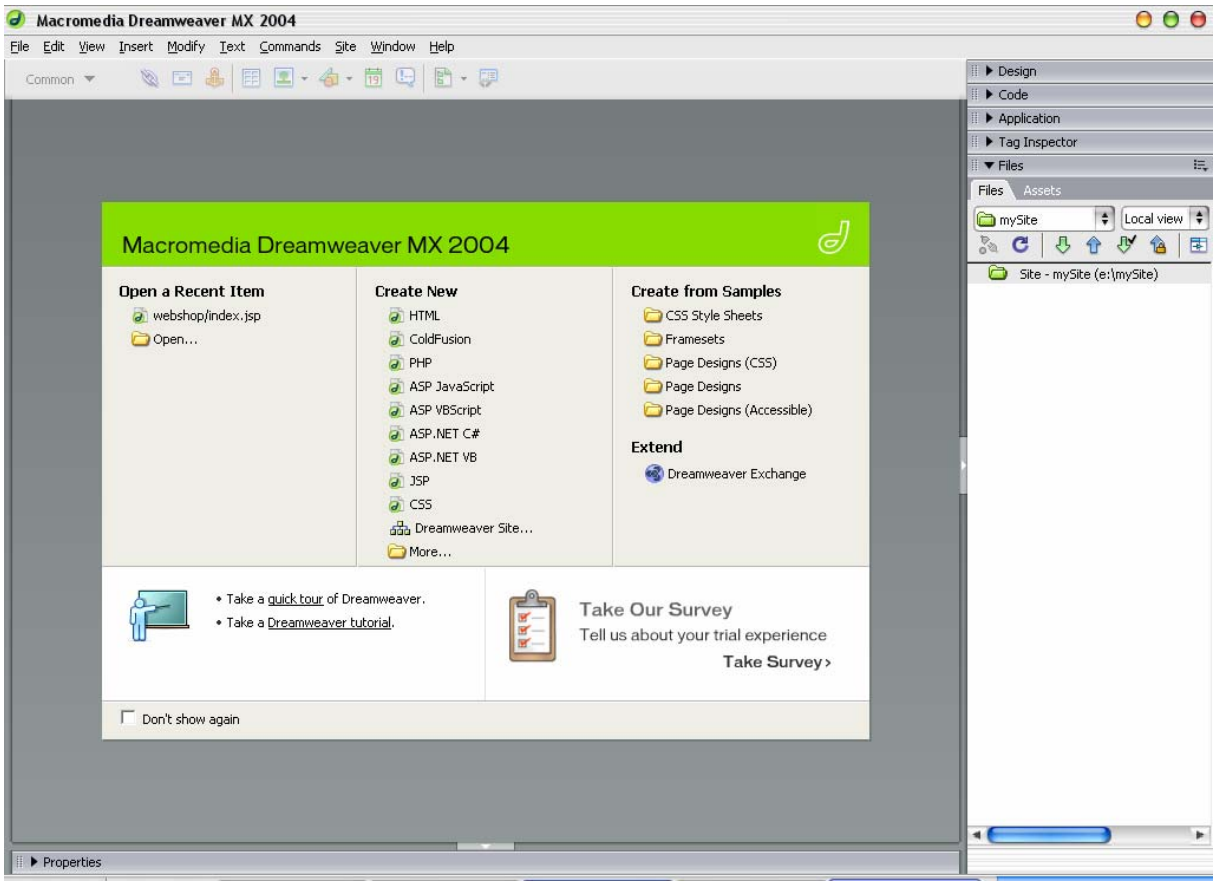
8　选择：No ，下一步



9　单击“Done”



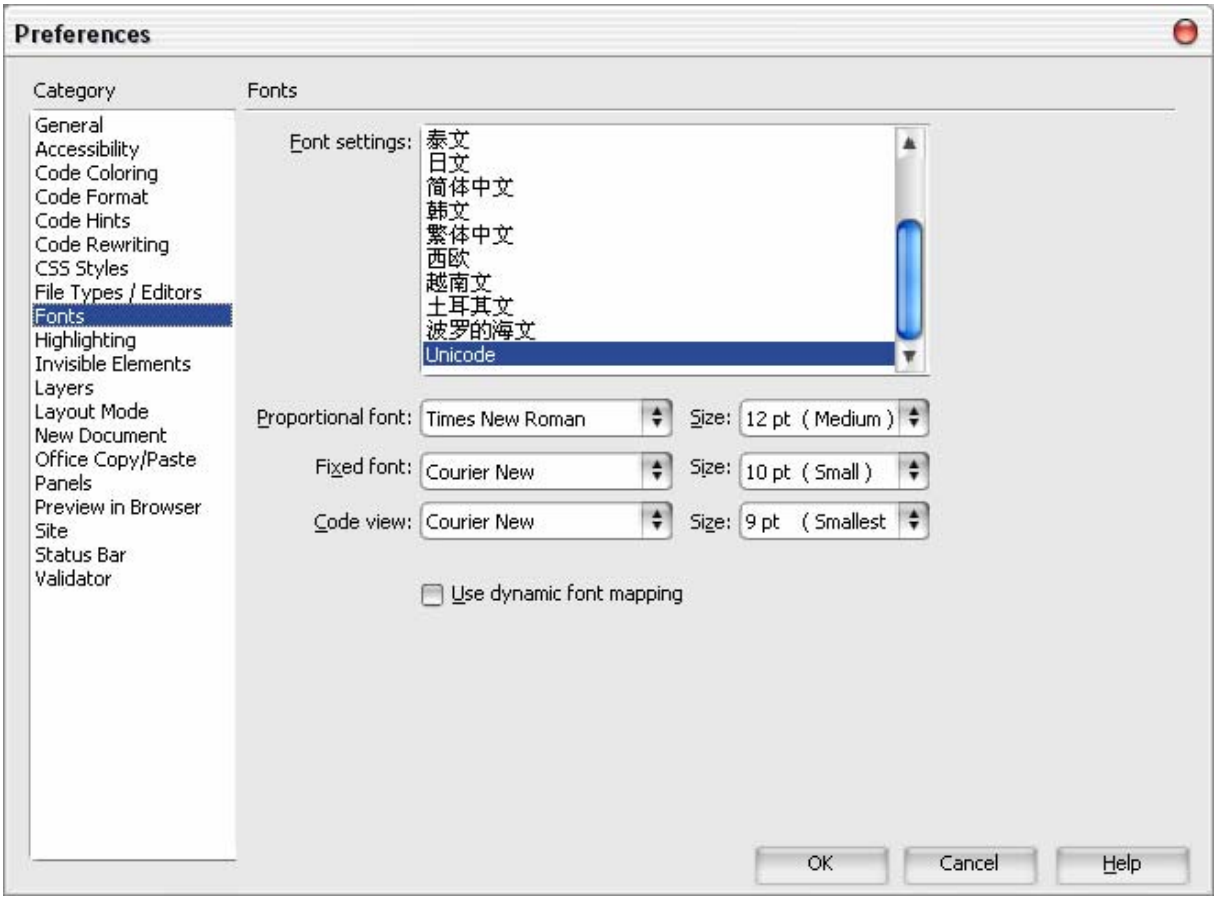
10　单击“Done”



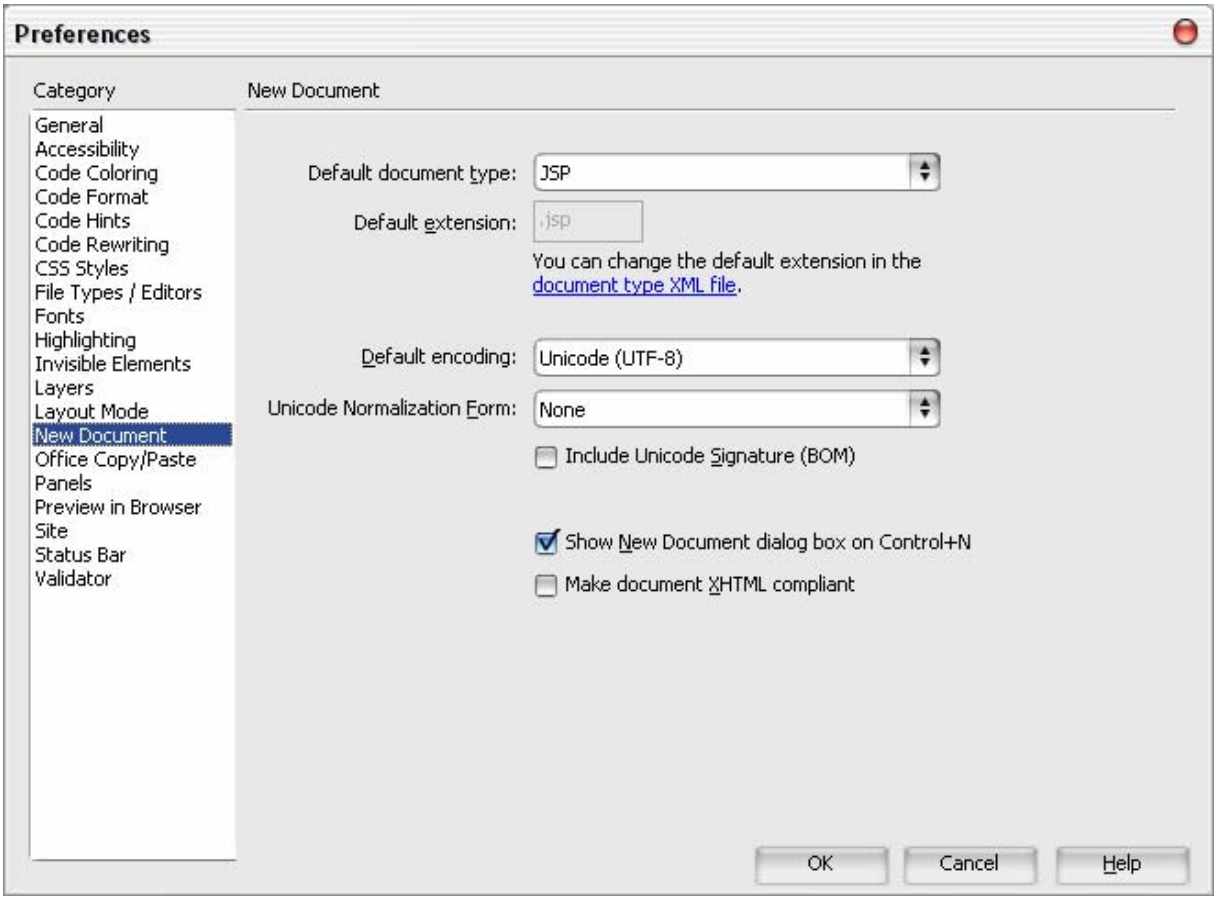
Dreamweaver 站点如何操作这里就不讲了，请看相关教材

11　Ctrl+u 调出 Dreamweaver 控制面版，做如下设置。

11.1　Fonts 字体设置：



11.2 New Document 新建文档设置：



11.3　单击 OK 按钮。

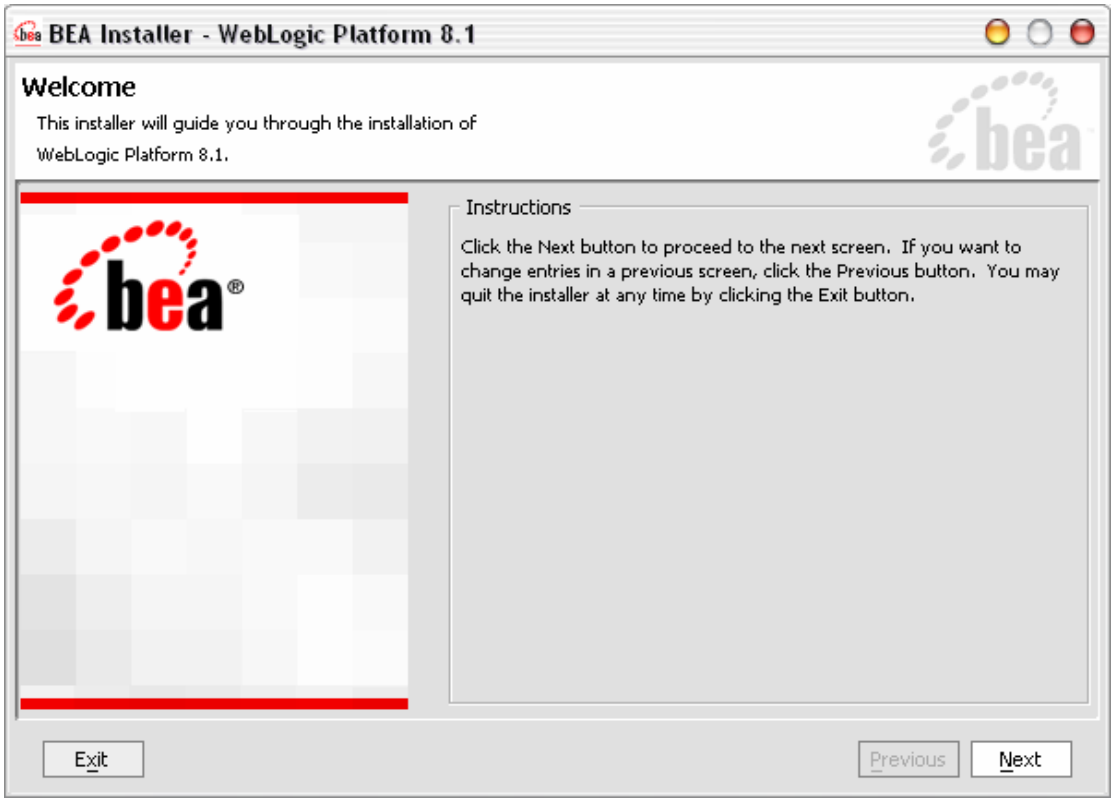
13.8 JBuilder + Weblogic + PostgreSQL 开发环境

JBuilder9 Weblogic8 + PostgreSQL7 How to

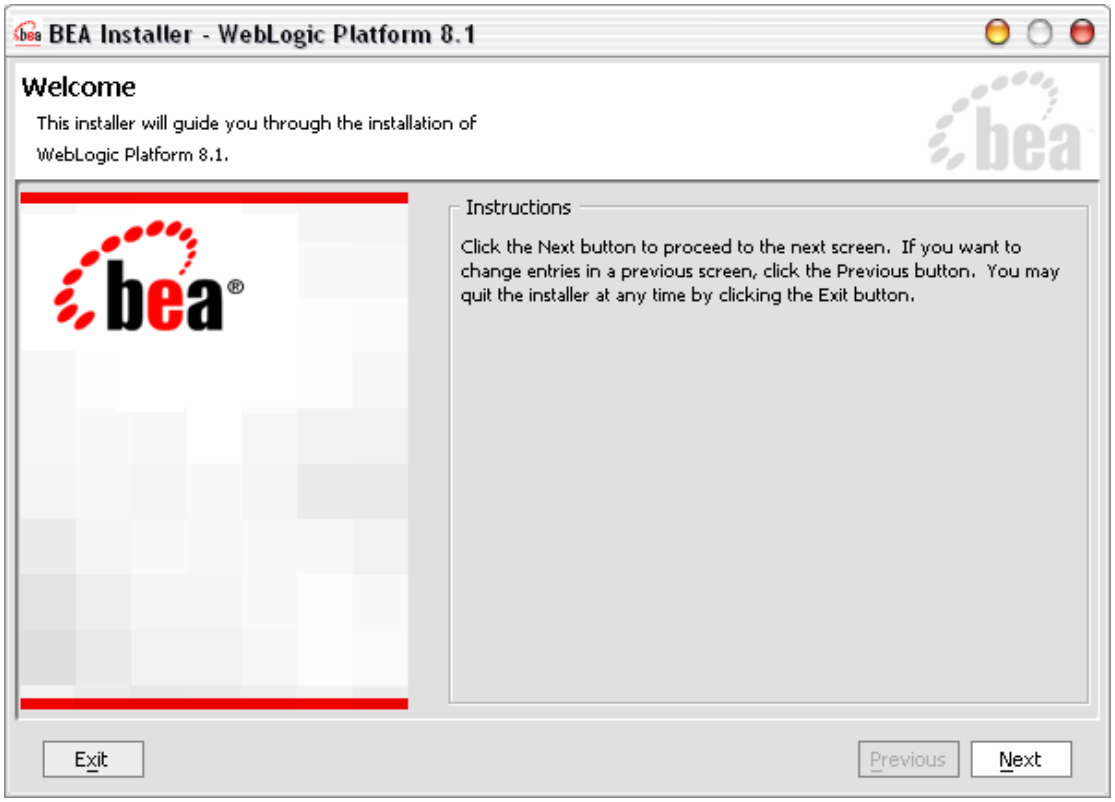
BEA Weblogic platform811 安装

1. 双击下载的 platform811_win32.exe 文件

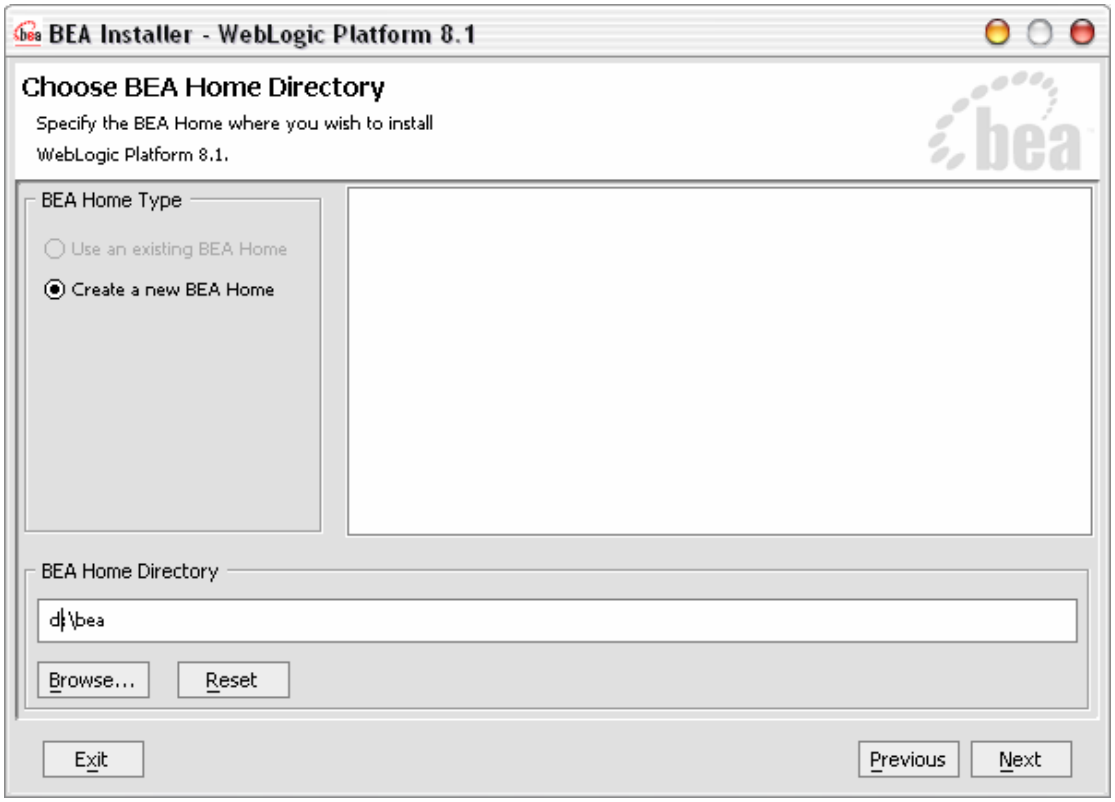
2. 开始解压



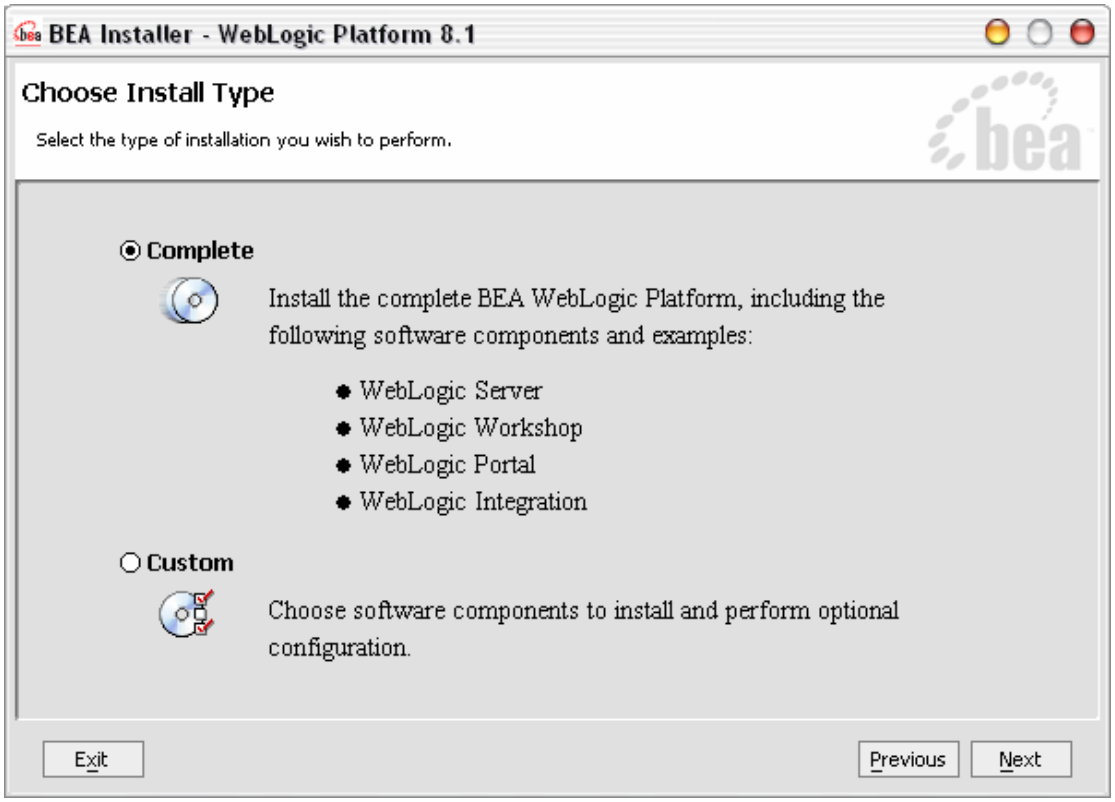
3. 单击 Next



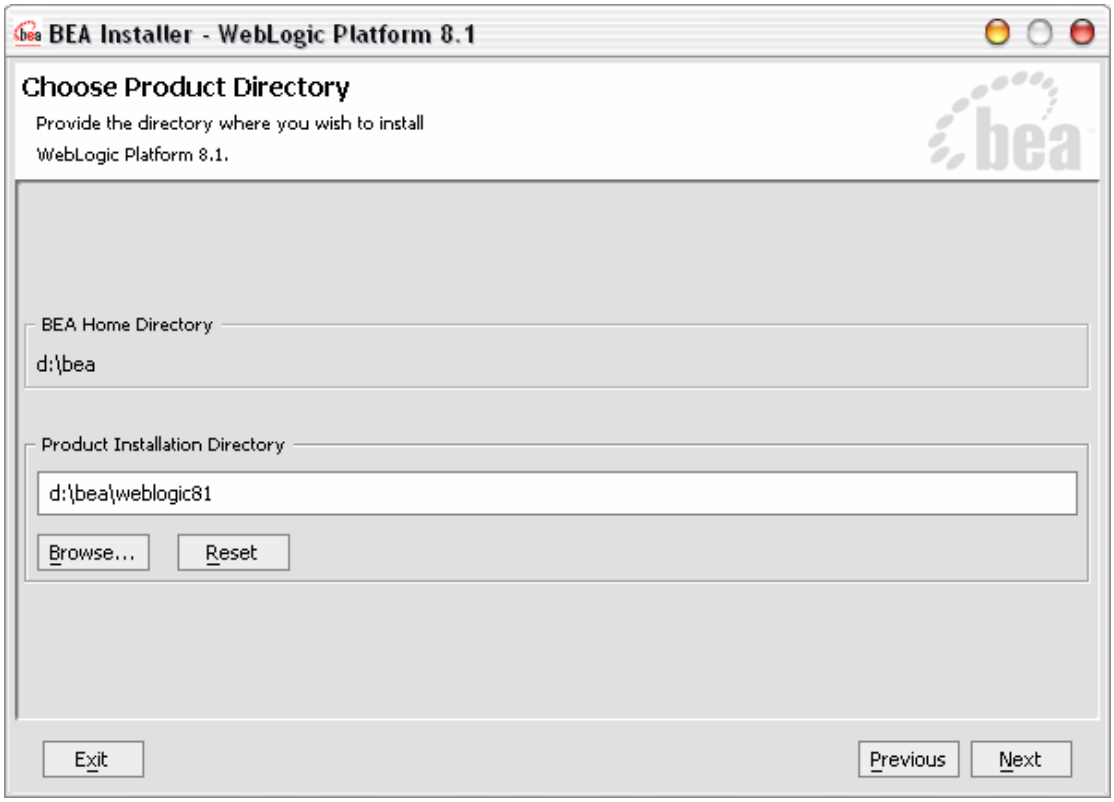
4. 单击 Next



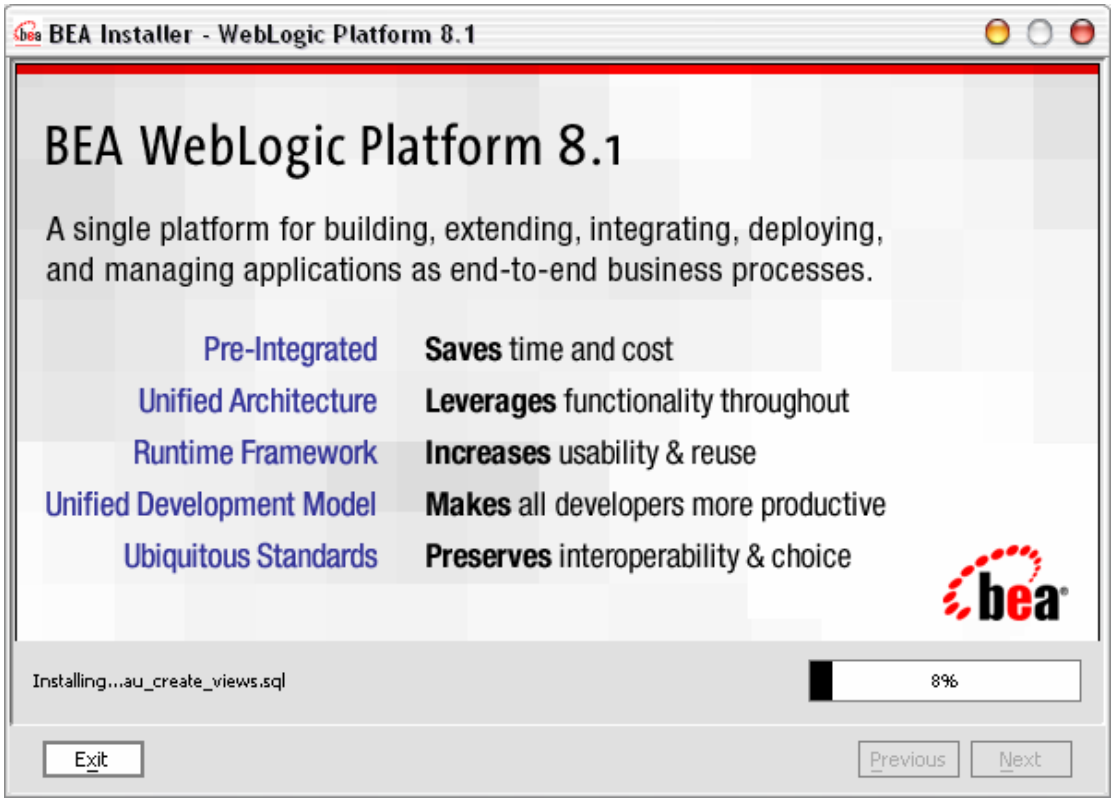
5. 选择安装 BEA 目录



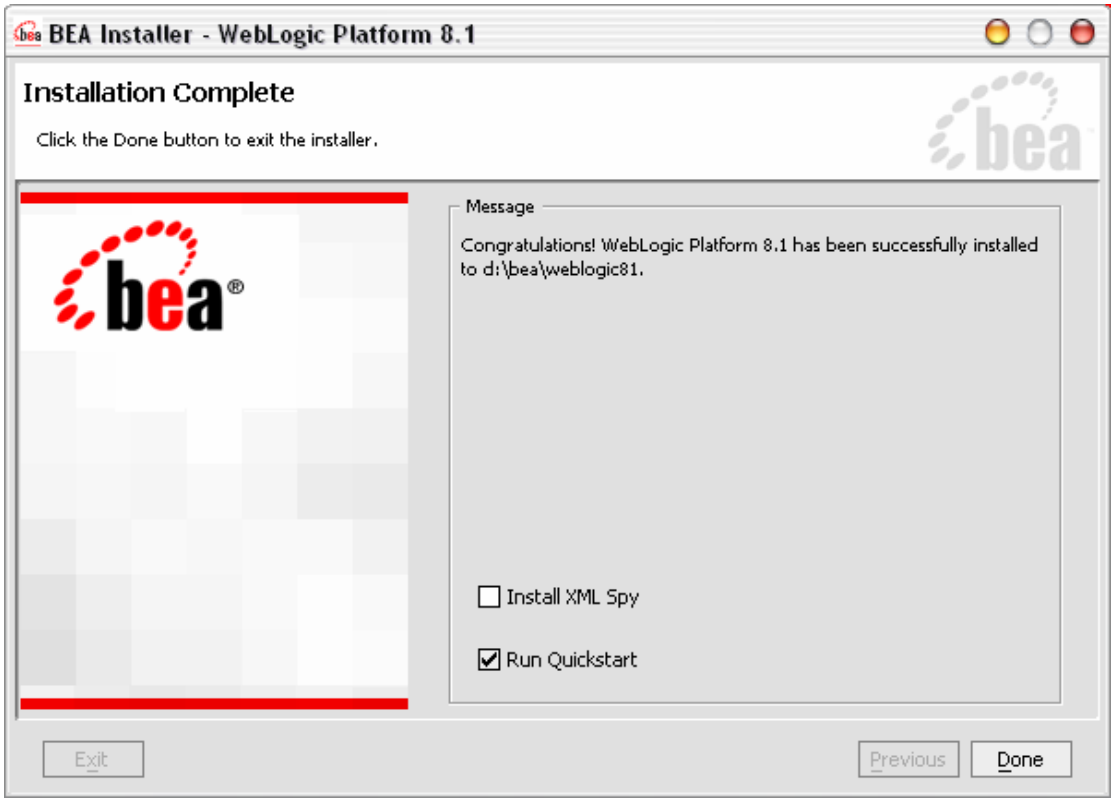
6. 默认完全安装，单击下一步 Next



7. 选择安装 bea weblogic8.1 产品目录，下一步 Next



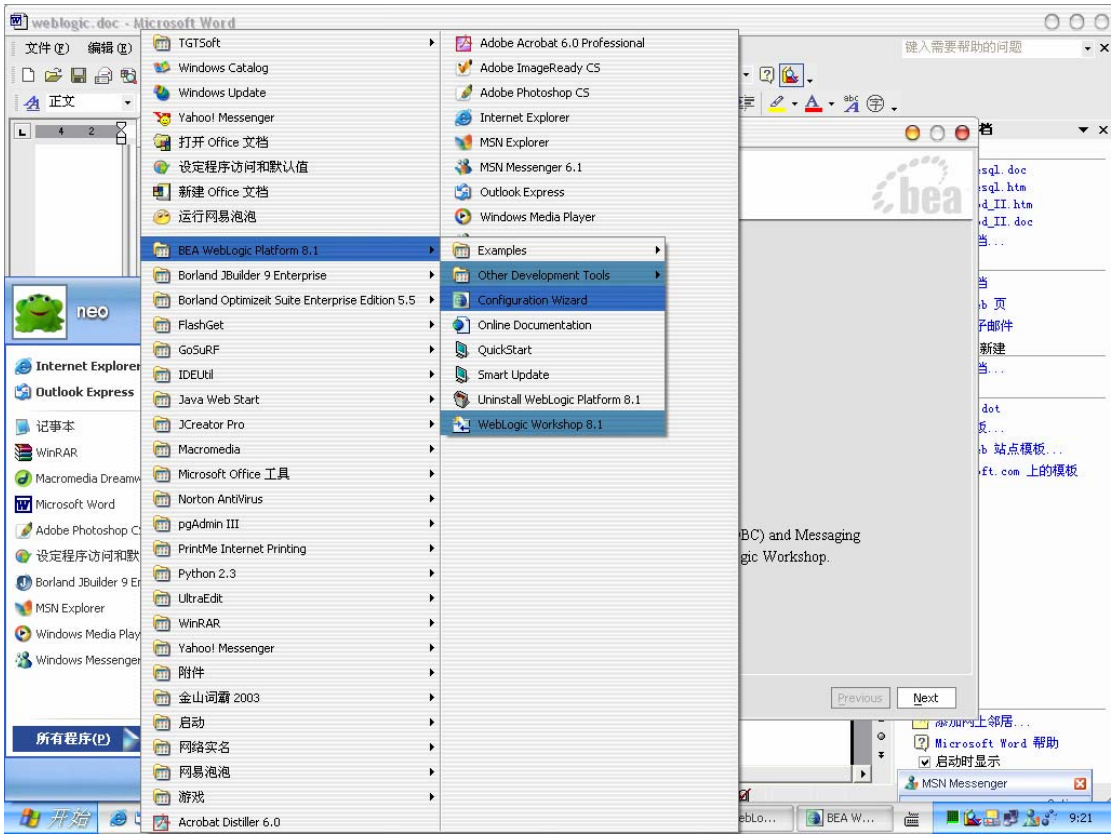
8. 开始安装。需要几分钟



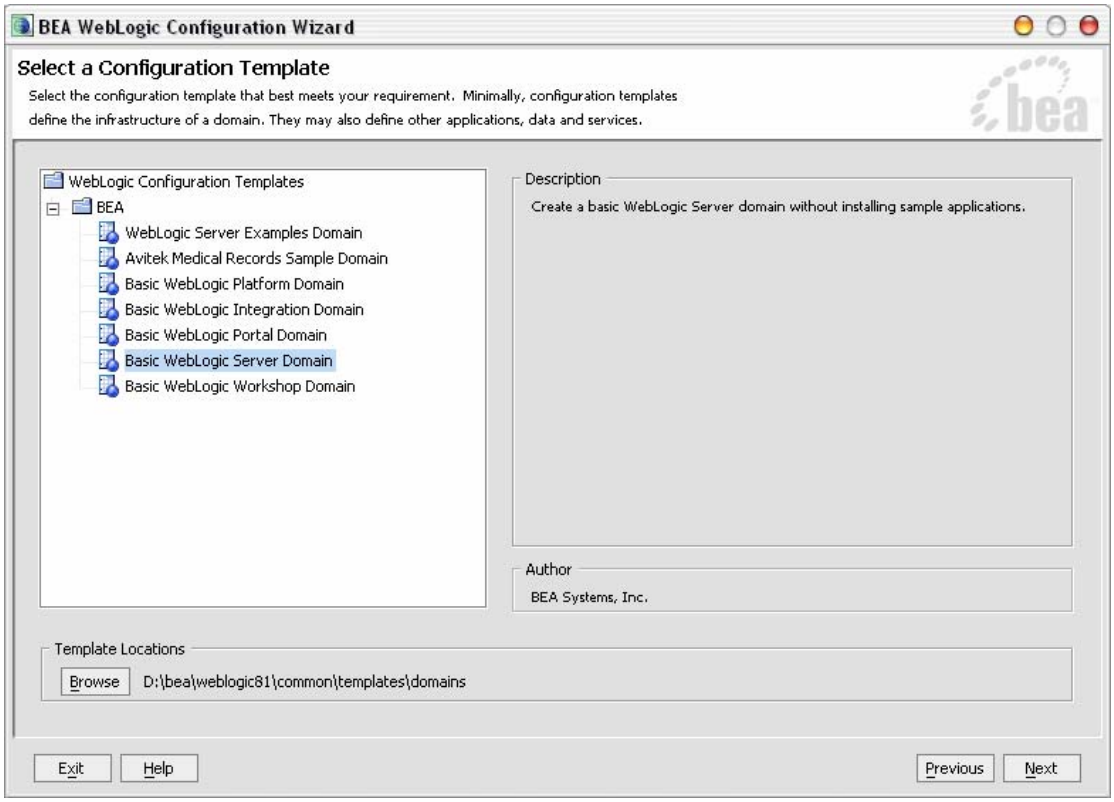
9. 单击 Done 完成安装

配置向导

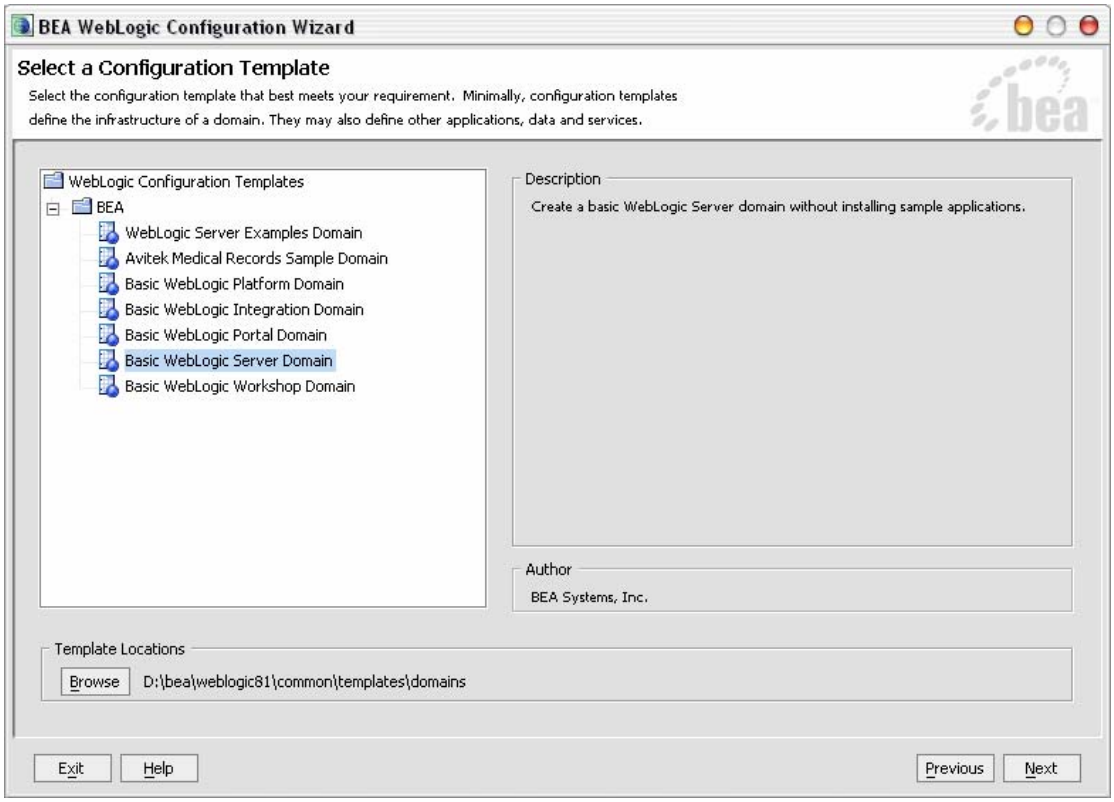
1. 启动 bea weblogic 的 Configuration Wizard 工具



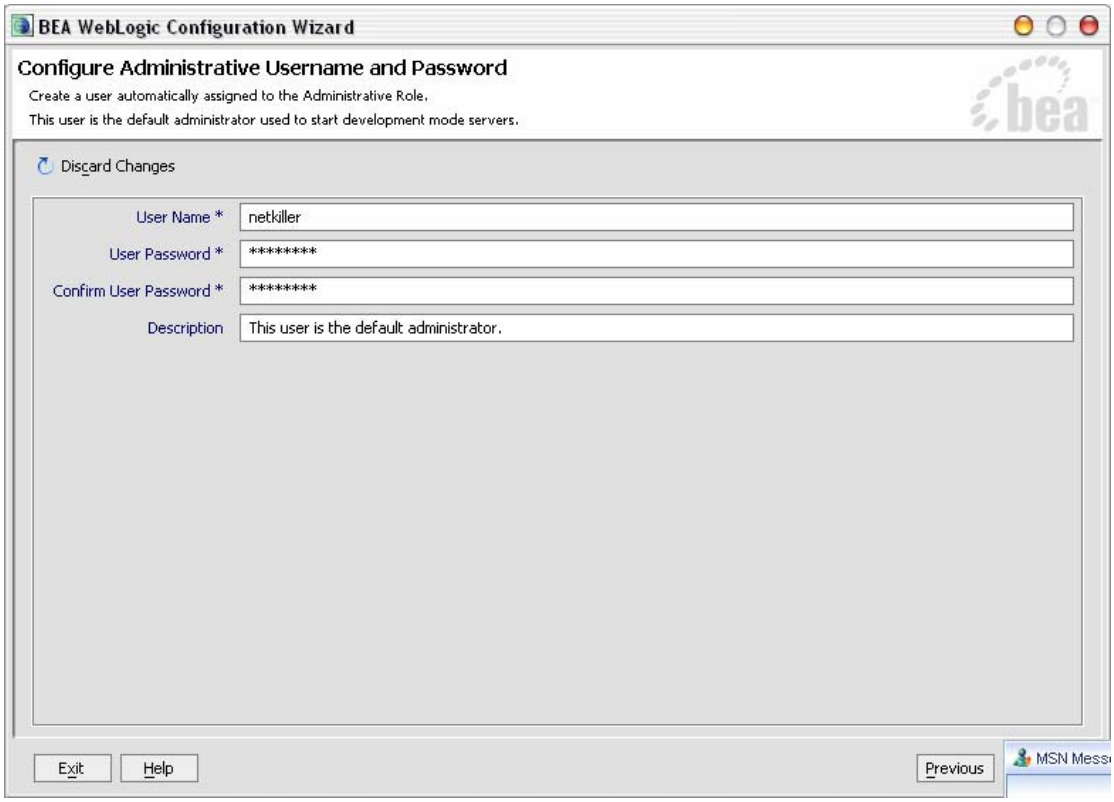
1. 显示 Configuration Wizard 界面



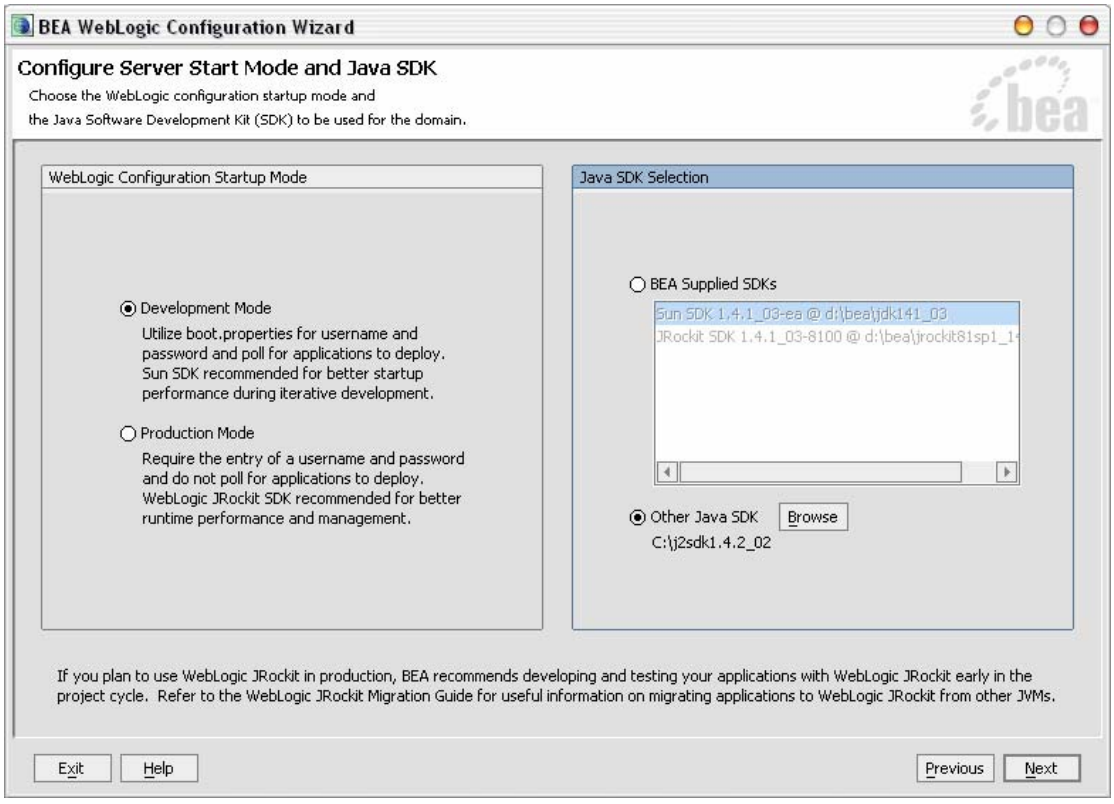
2. 单击 Next



3. 单击 Next

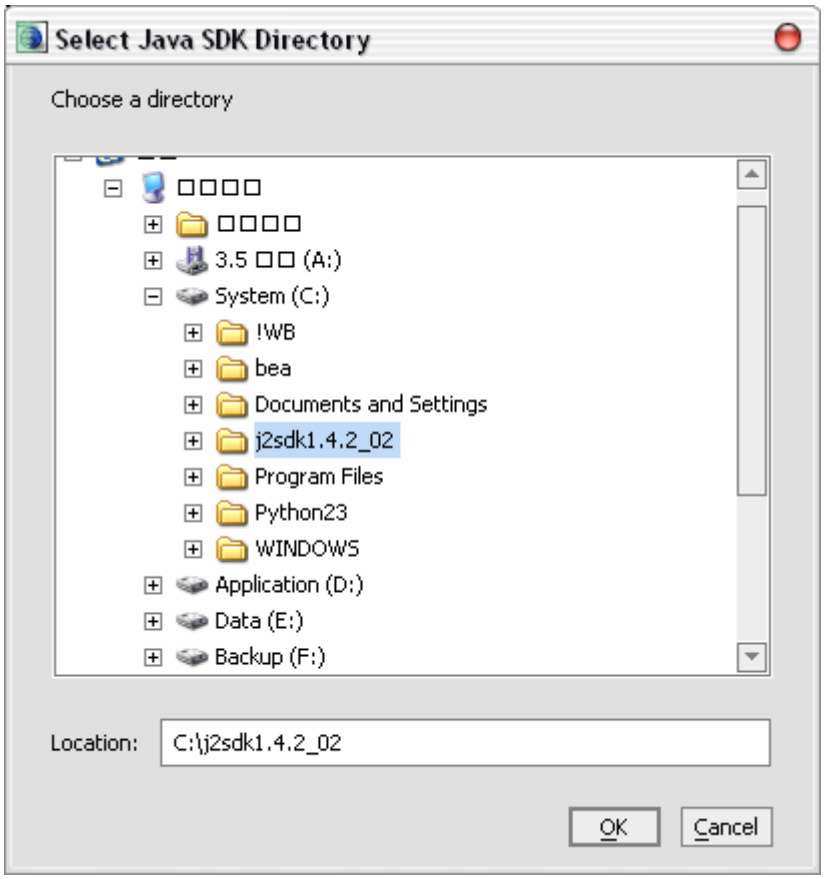


4. 输入用户、密码（密码需要 8 位）

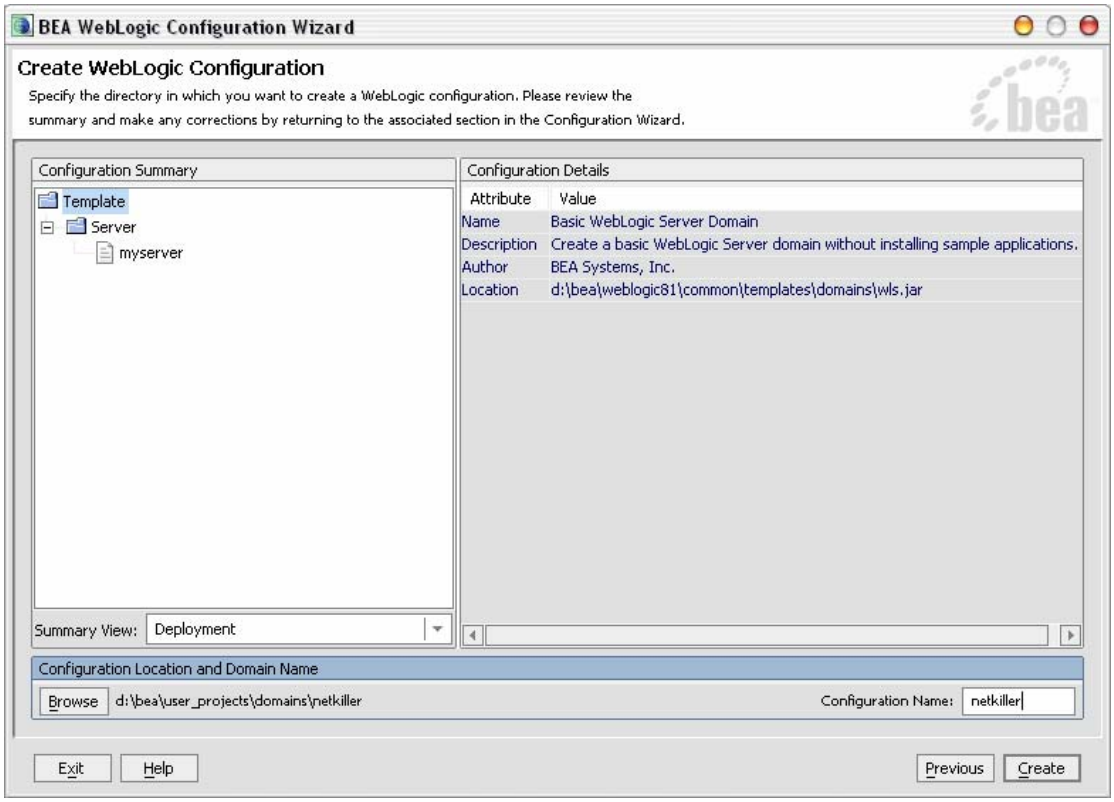


5. bae weblogic 默认安装了 JDK1.4.1，我要使用 JDK 1.4.2

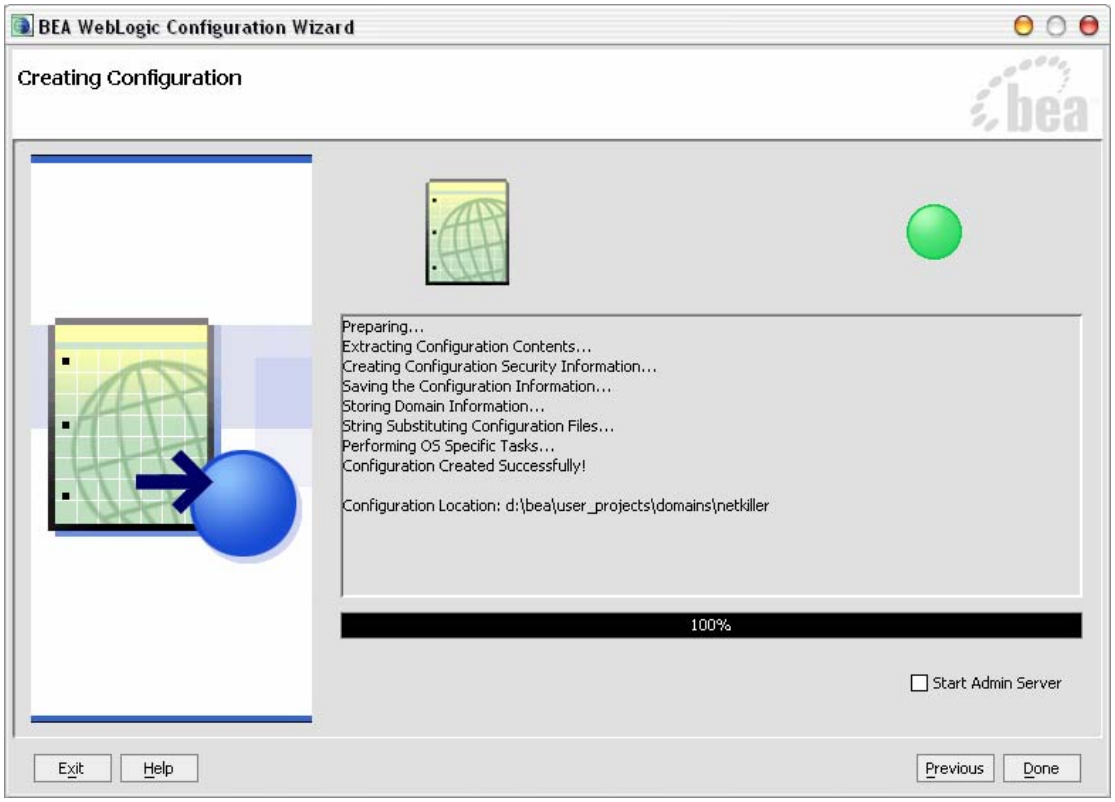
选择 Other Java SDK



6. 单击 Next



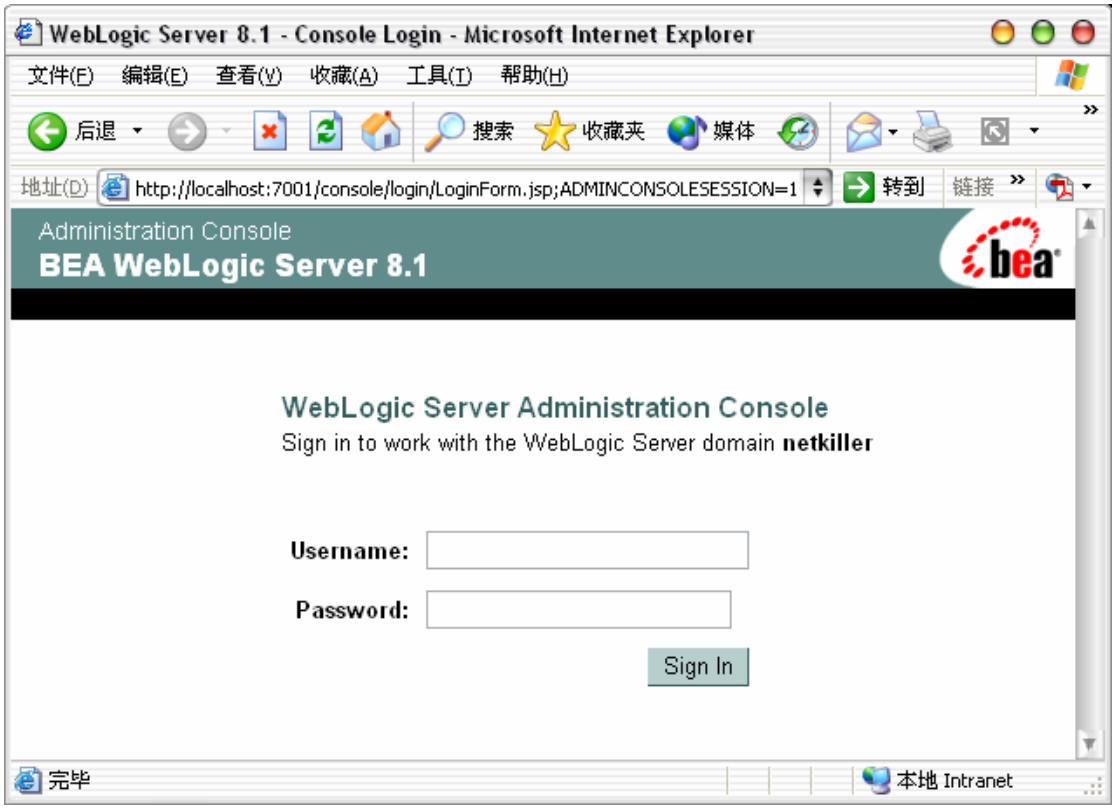
7. Configuration Name : netkiller



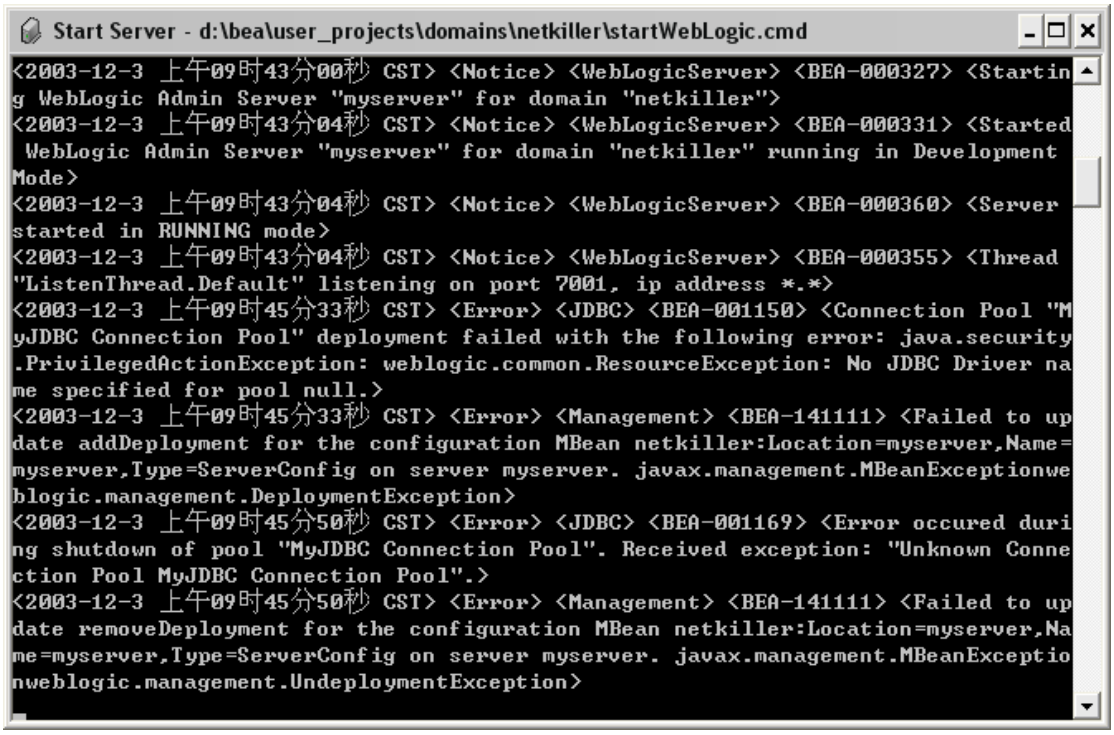
8. 选择 Start Admin Server ，单击 Done

数据库连接池

9. 在 IE 地址栏中输入：http://localhost:7001/console



10. 关闭 weblogic



11. 配置 PostgreSQL JDBC 驱动

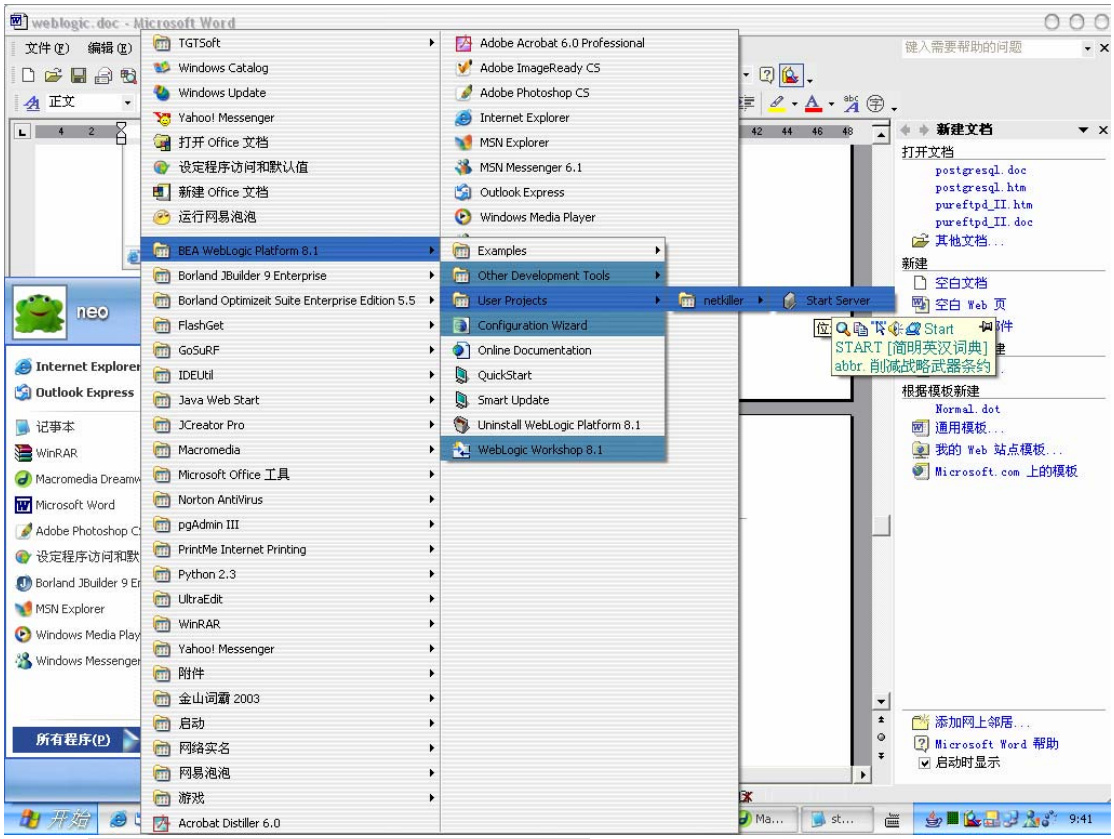
将 PostgreSQL JDBC 驱动 pg73jdbc3.jar 复制到 D:\bea\weblogic81\server\lib 目录下

编辑文件 D:\bea\user_projects\domains\netkiller\startWebLogic.cmd

在 set CLASSPATH 下面加入

set CLASSPATH=%CLASSPATH%;%WL_HOME%\server\lib\pg73jdbc3.jar

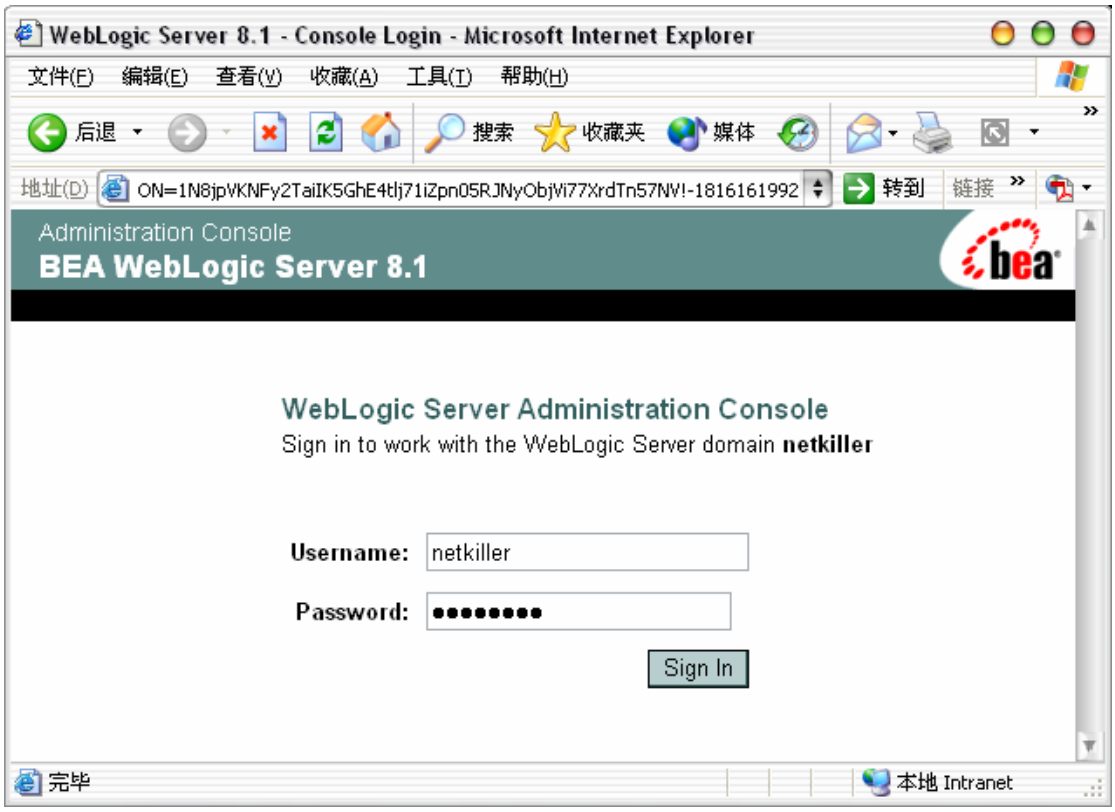
12. 然后启动 weblogic →netkiller



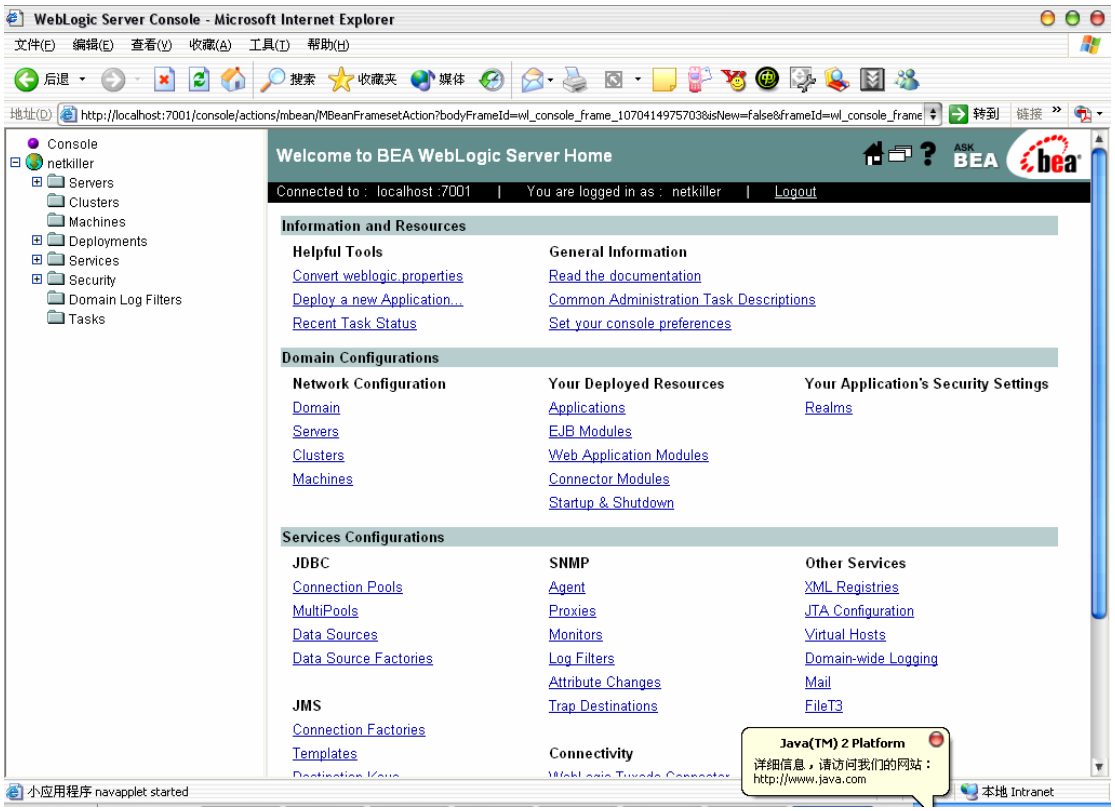
注意：d:\bea\WEBLOG~1\server\lib\pg73jdbc3.jar

```
d:\bea\user_projects\domains\netkiller>ECHO OFF
.
CLASSPATH=C:\J2SDK1~1.2_0\lib\tools.jar;d:\bea\WEBLOG~1\server\lib\weblogic_sp.j
ar;d:\bea\WEBLOG~1\server\lib\weblogic.jar;d:\bea\WEBLOG~1\server\lib\ojdbc14.ja
r;d:\bea\WEBLOG~1\common\eval\pointbase\lib\pbserver44.jar;d:\bea\WEBLOG~1\commo
n\eval\pointbase\lib\pbclient44.jar;C:\J2SDK1~1.2_0\jre\lib\rt.jar;d:\bea\WEBLOG
~1\server\lib\webservices.jar;;d:\bea\WEBLOG~1\server\lib\pg73jdbc3.jar
.
PATH=d:\bea\WEBLOG~1\server\bin;C:\J2SDK1~1.2_0\jre\bin;C:\J2SDK1~1.2_0\bin;C:\W
INDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;d:\bea\WEBLOG~1\server\bin\o
ci920_8
.
*************************************************
*    To start WebLogic Server, use a username and      *
*    password assigned to an admin-level user.    For *
*    server administration, use the WebLogic Server *
*    console at http:\\[hostname]:[port]\console        *
*************************************************
```

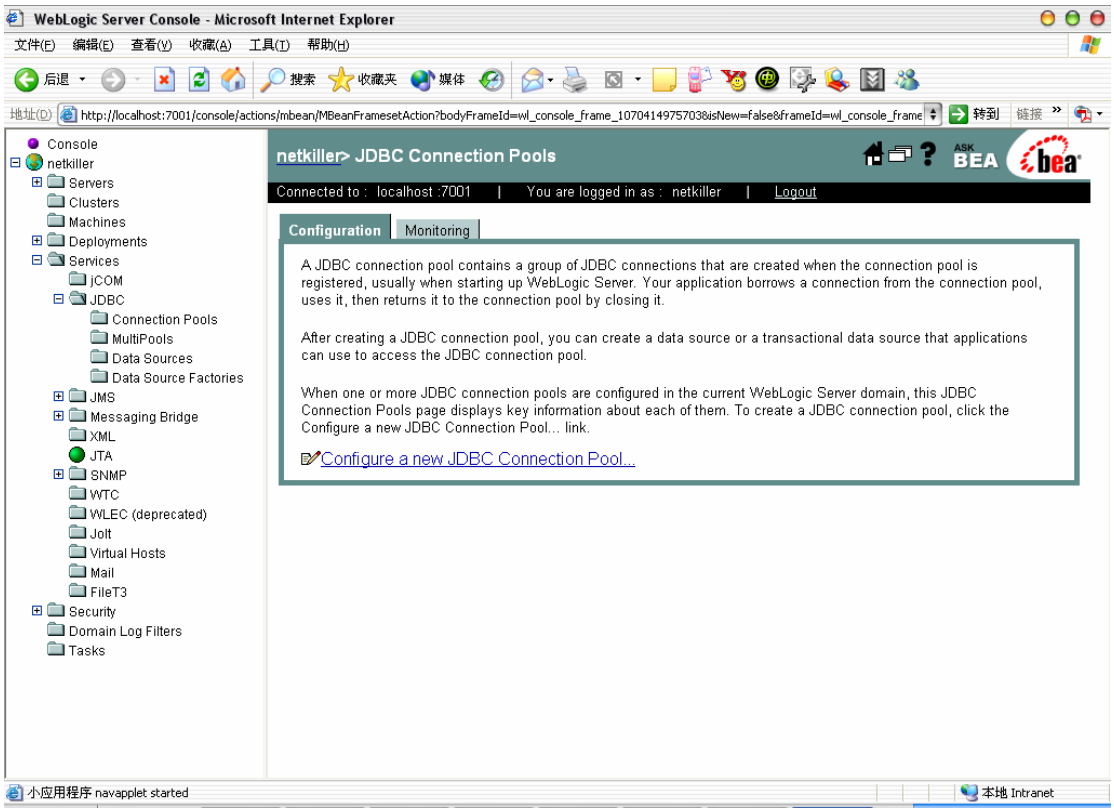
13. 登录



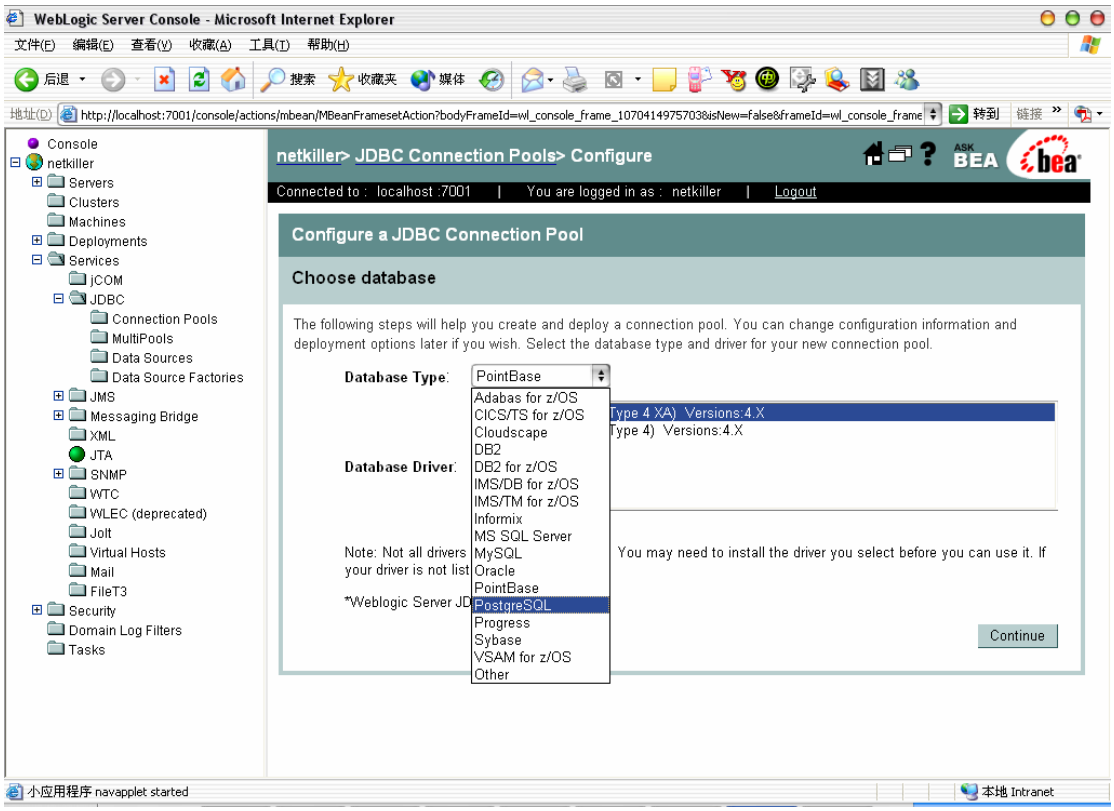
14. 登陆后显示



15. 选择 netkiller-->Service-->JDBC-->Connection Pools
16. 单击 Configure a new JDBC Connection Pool...



17. 选择 Database Type: 选择 PostgreSQL，然后单击 Continue

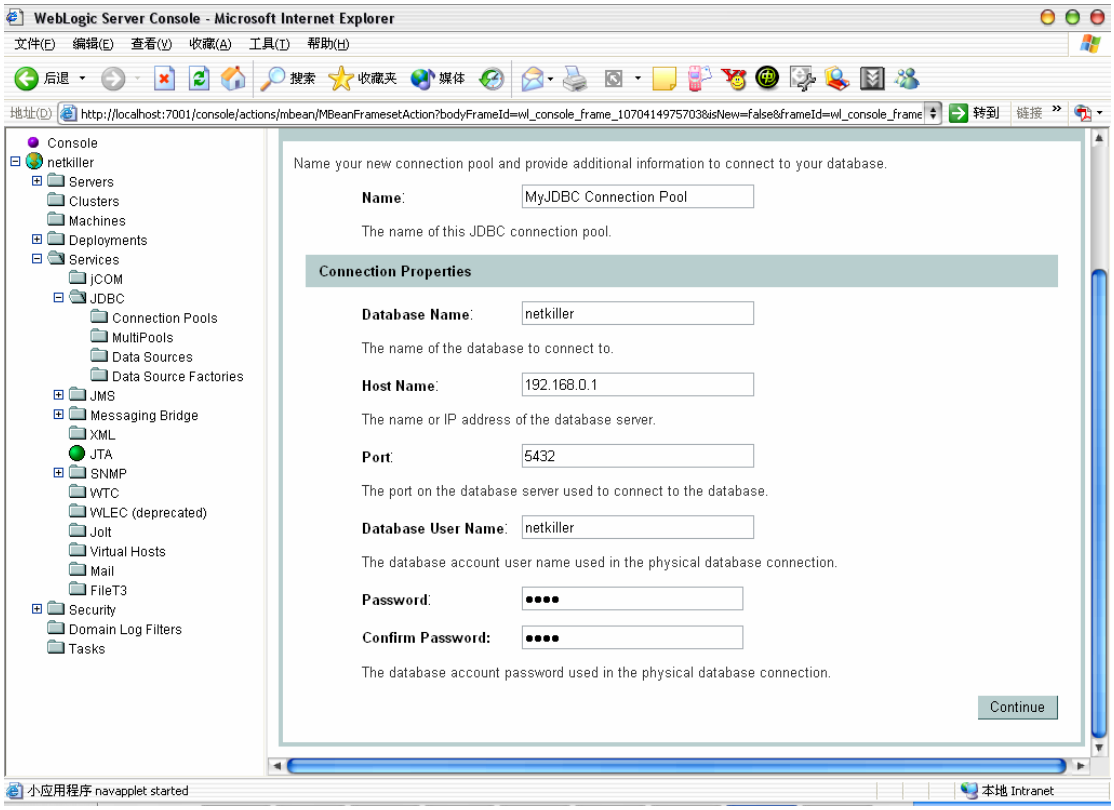


18. 输入数据

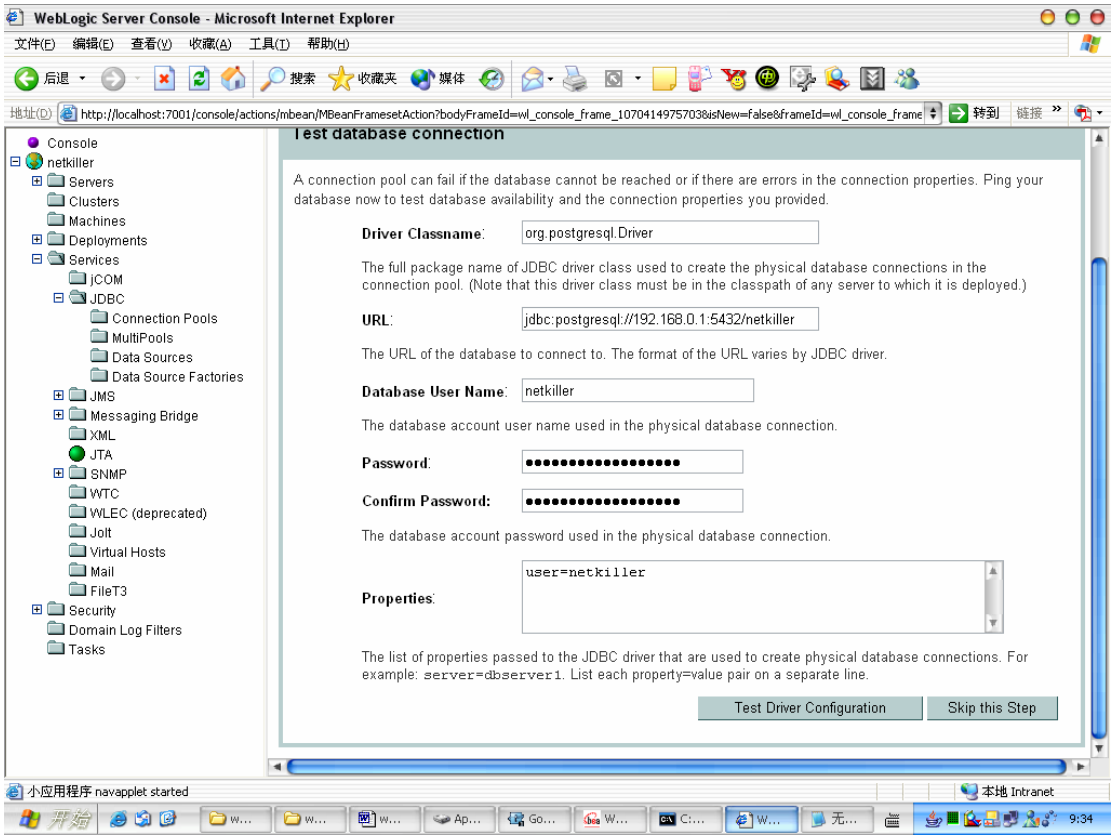
Name：连接池名

Database Name：数据库名

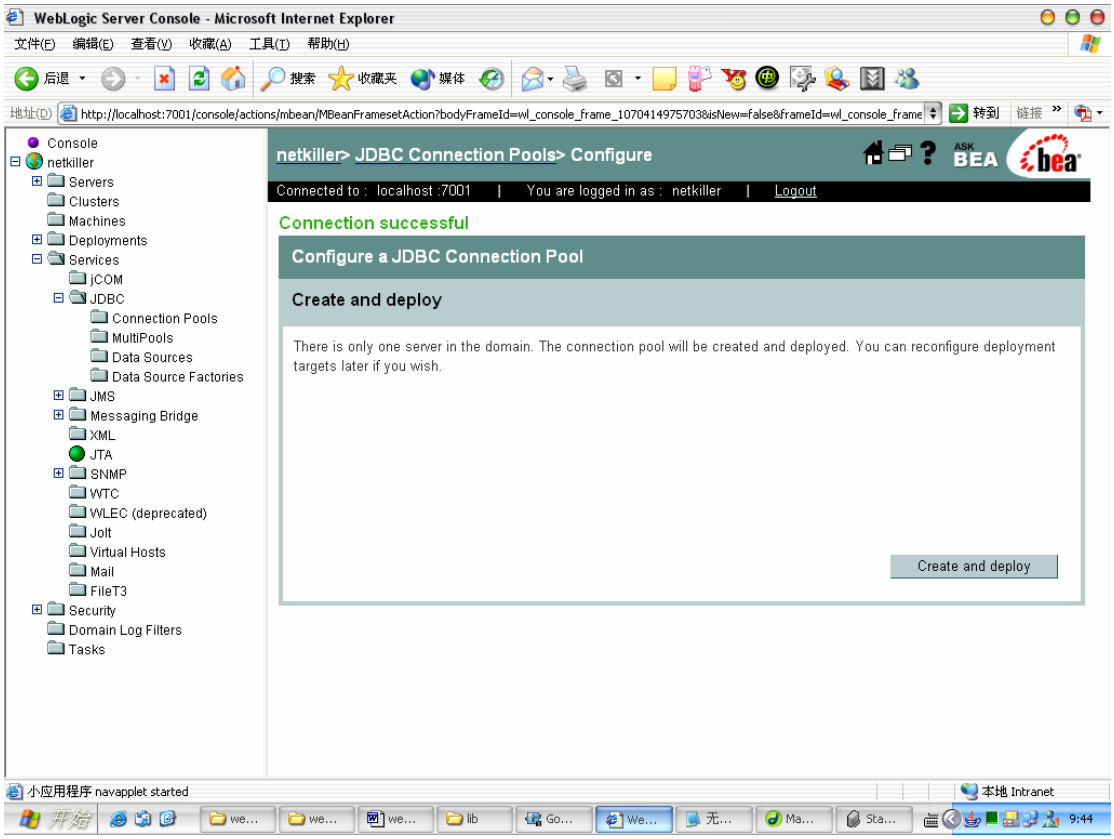
Host Name：主机名
Port：端口号
Database User Name：数据库用户名
Password：数据库密码
Confirm Password：确认密码
然后单击 Continue



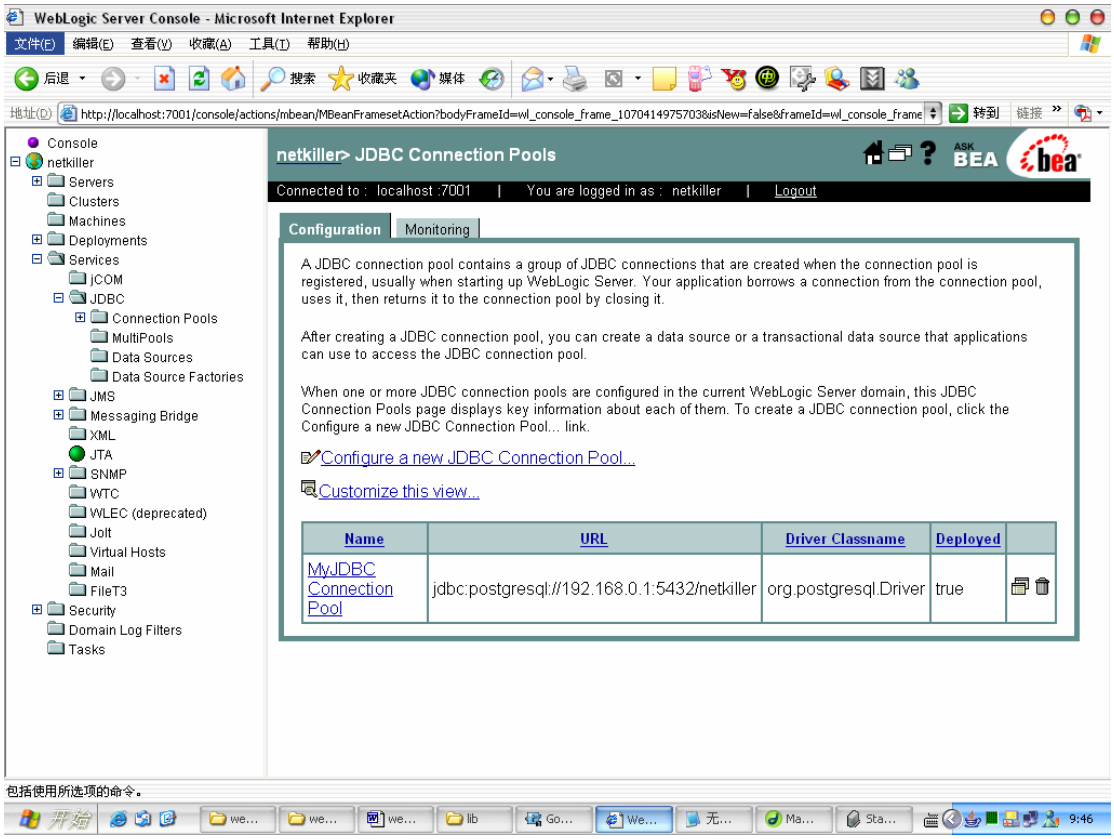
19. 测试数据库连接



1. 单击 Test Driver Configuation，输入绿色 Connection successful 测试成功。如果是红色表示出错。



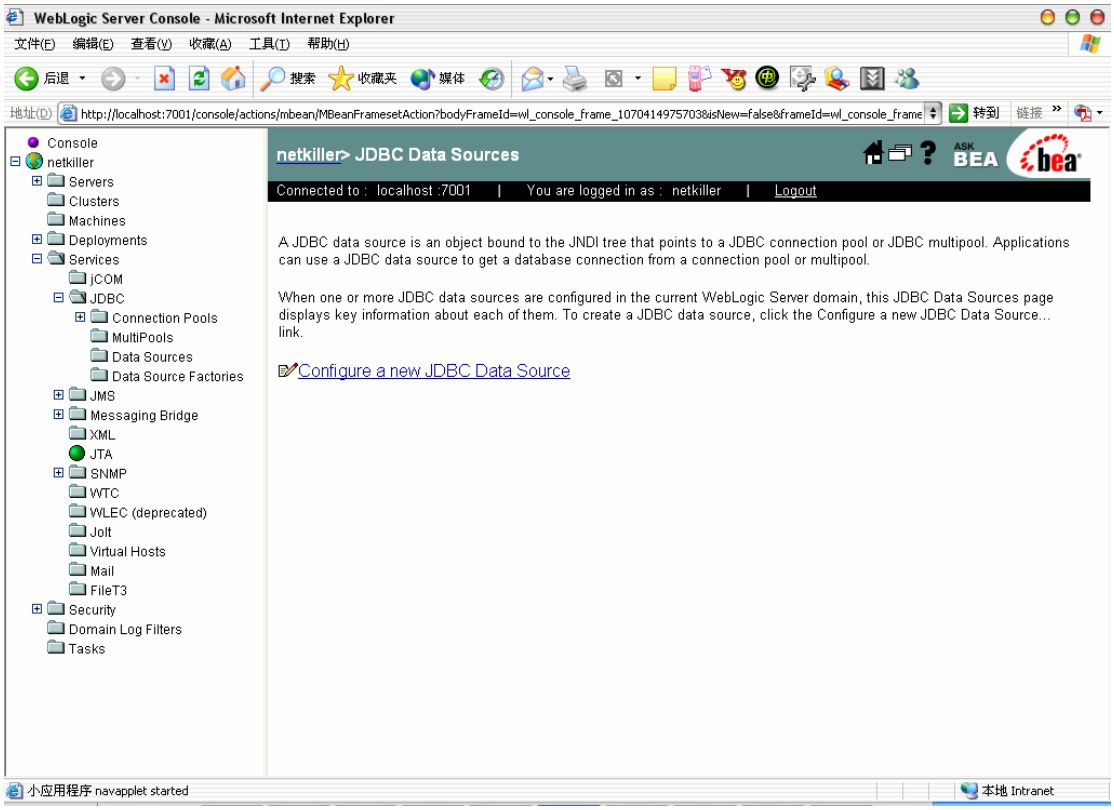
2. 单击 Create and deploy



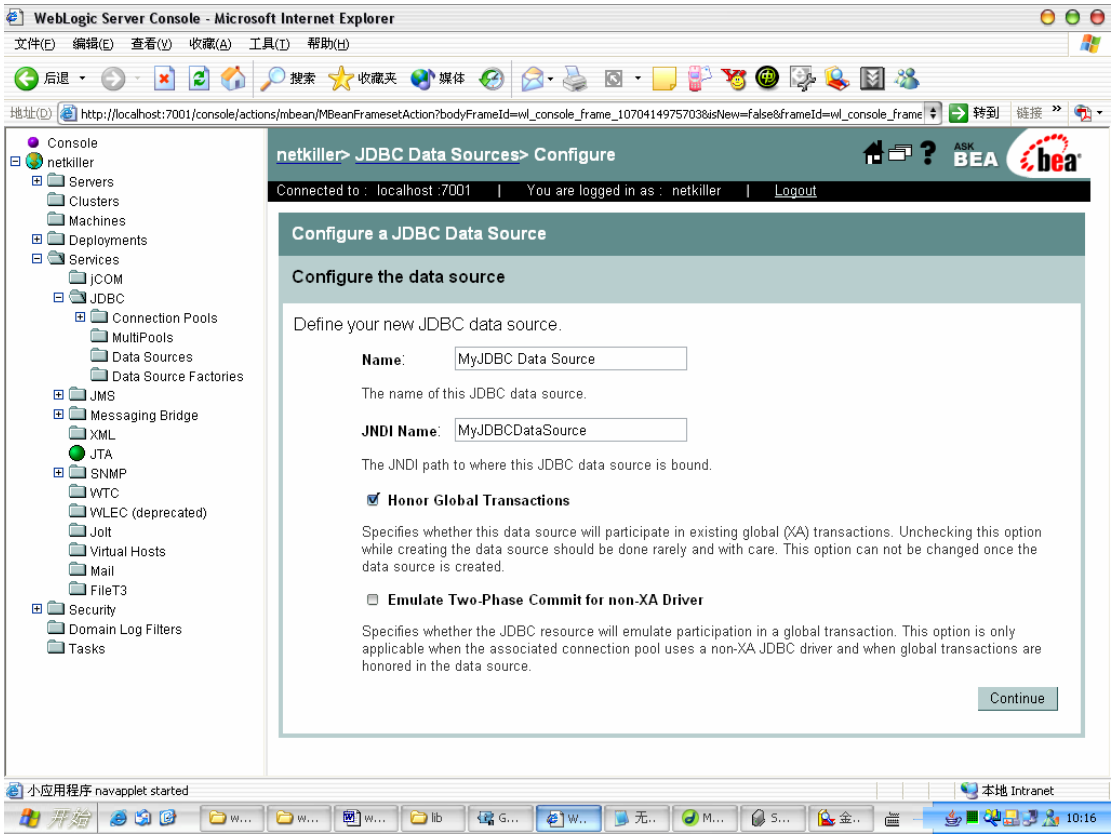
3. 连接池配置完成

数据源配置

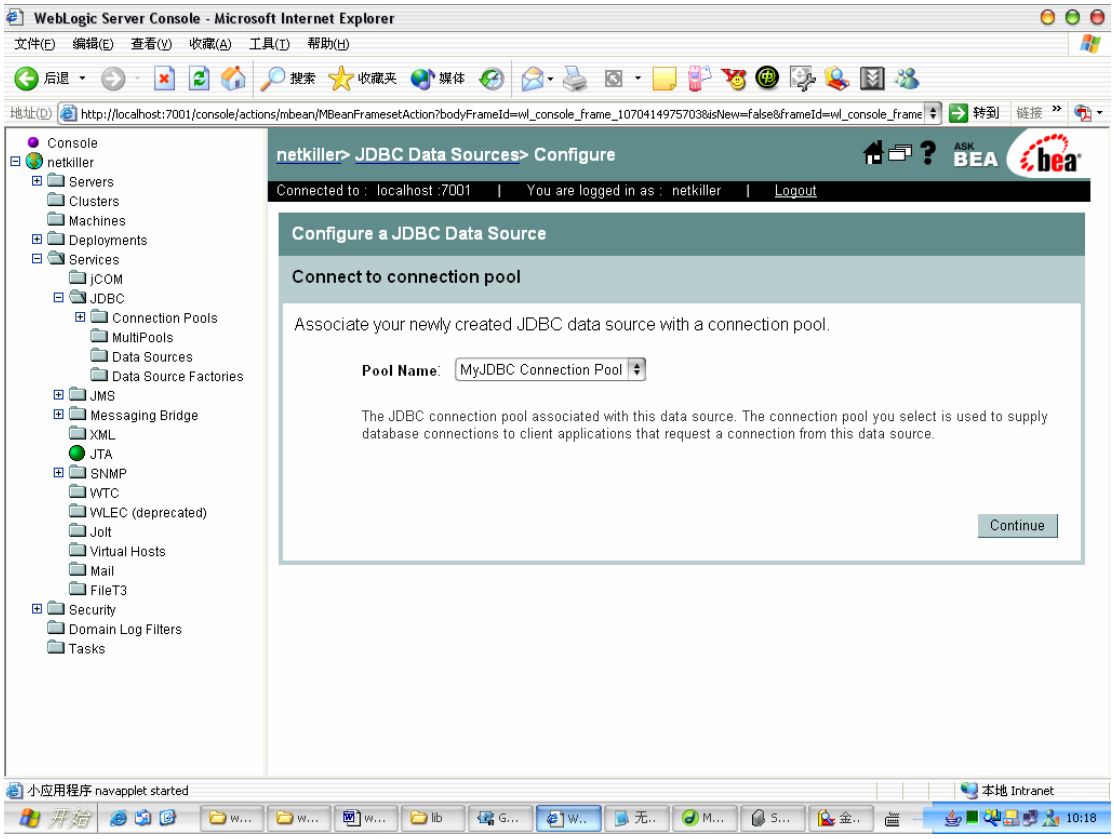
1. 选择 netkiller-->Service-->JDBC--> Data Source



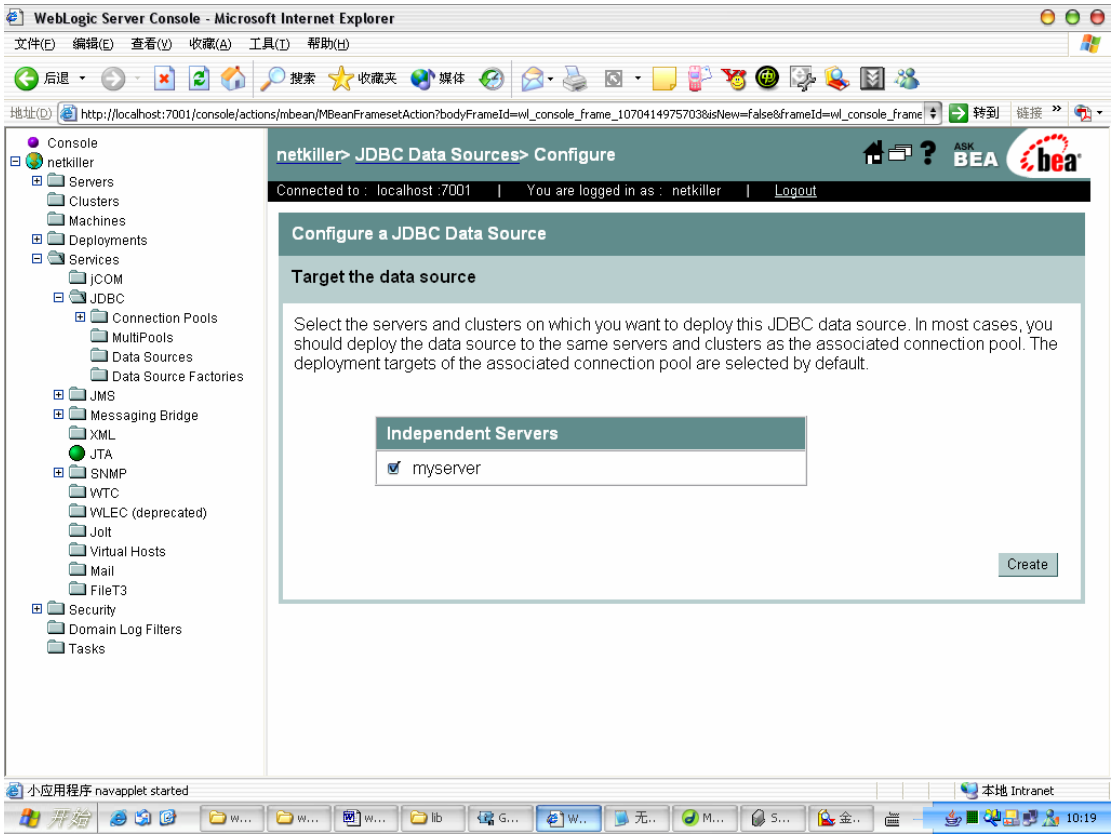
2. 单击 Configure a new JDBC Data Source
3. JNDI Name：输入 MyJDBCDataSource



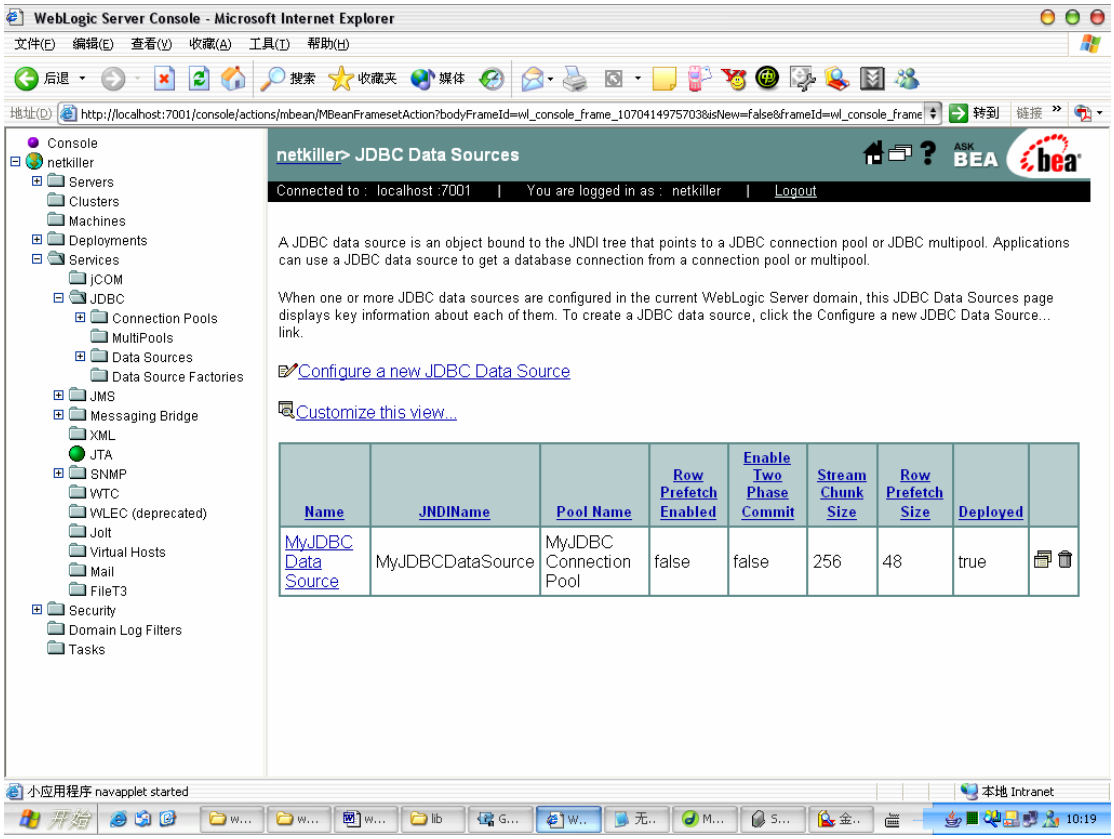
4. 单击 Continue



5. 选择一个连接池，然后单击 Continue



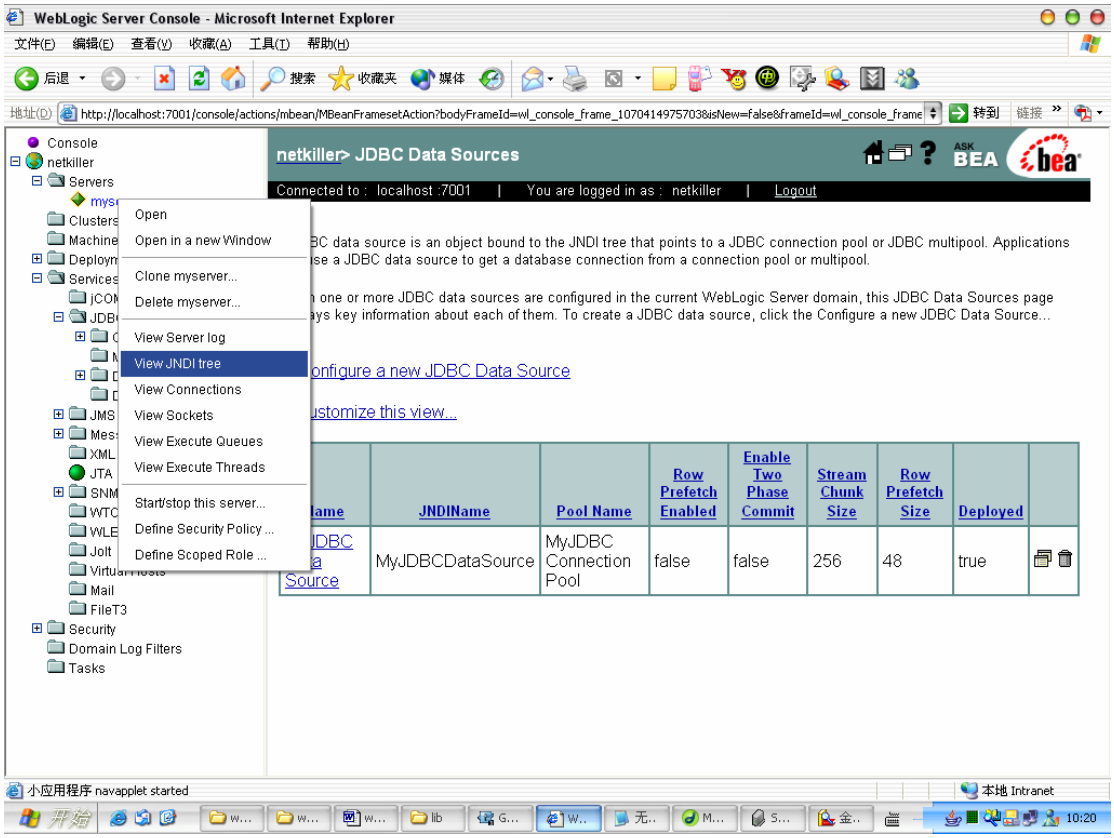
6. 单击 Create



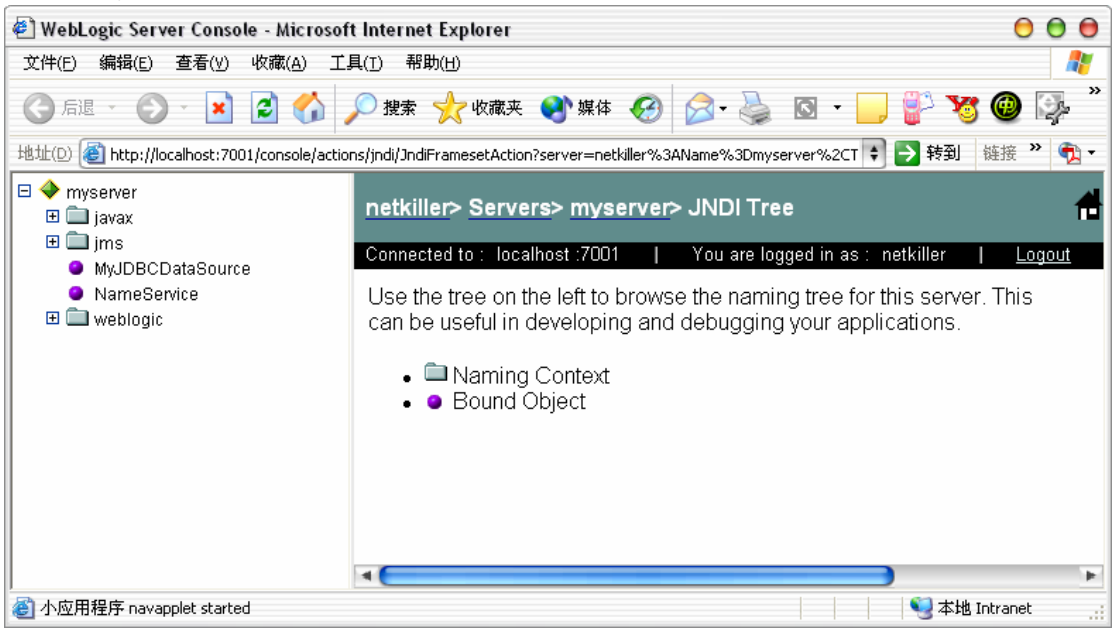
7. 数据源配置完成

查看ＪＮＤＩ树

1. 展开 netkiller→Servers→myserver



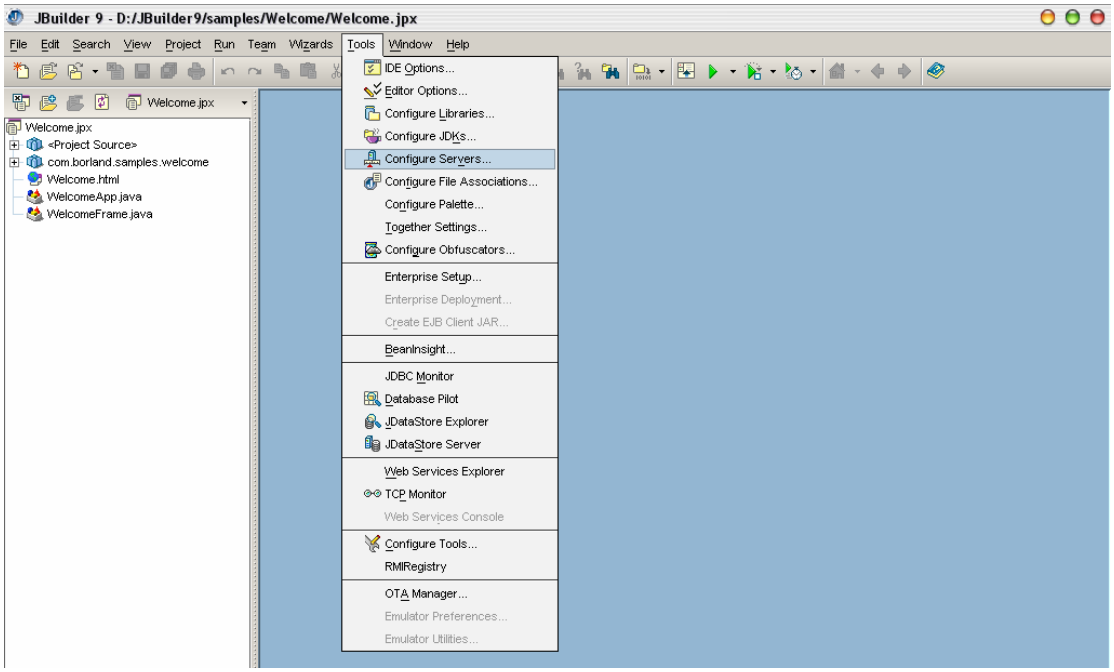
2. 在 myserver 上单击鼠标右键，选择 View JNDI tree



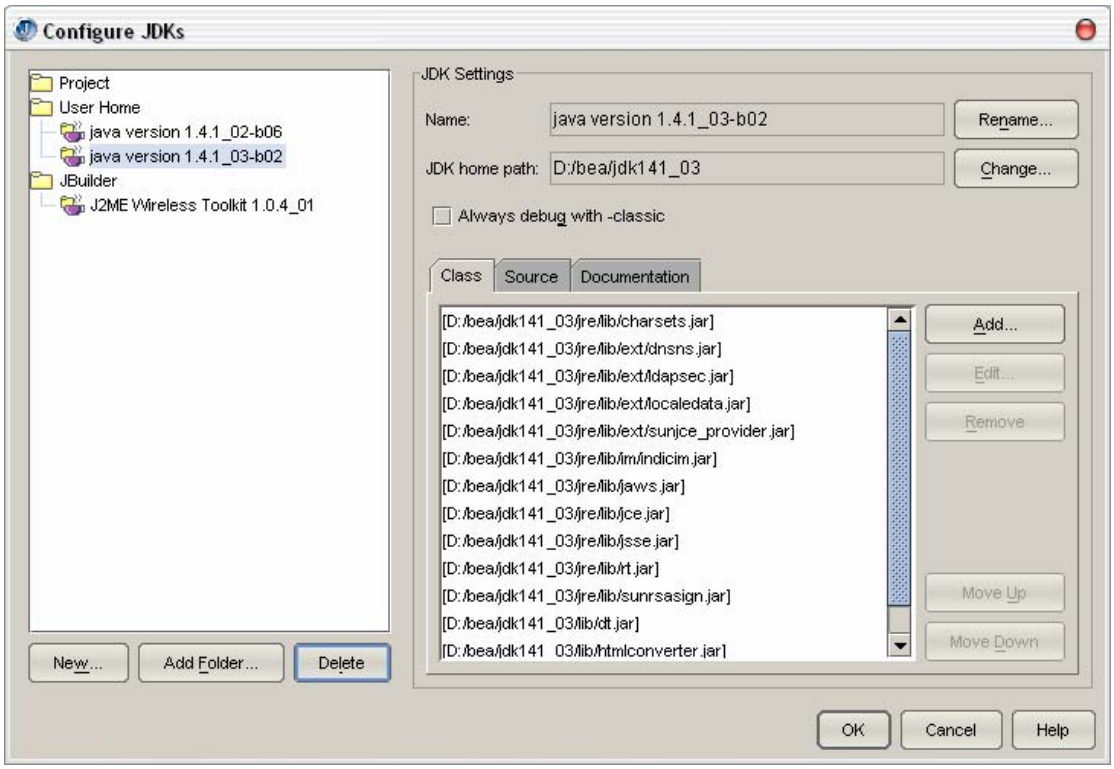
myserver 下可以看到 myJDBCDataSource

3. 数据源配置完成

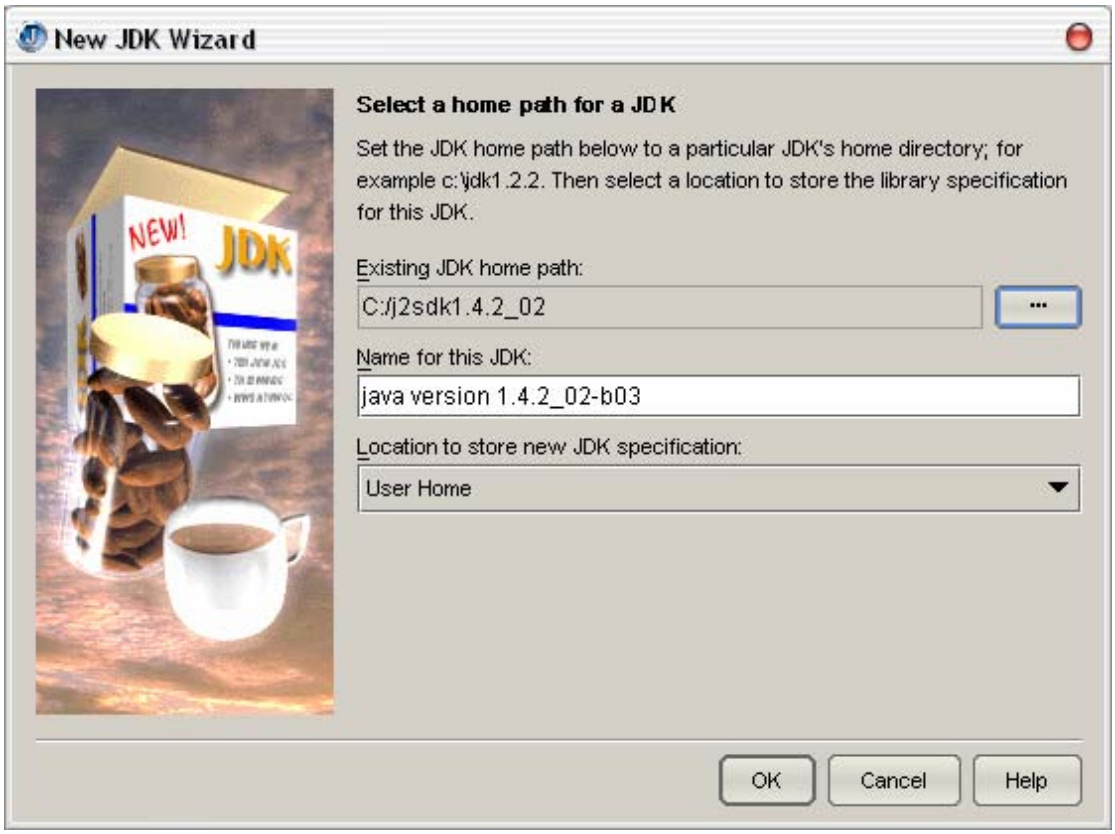
JBuilder 9.0 Weblogic 配置



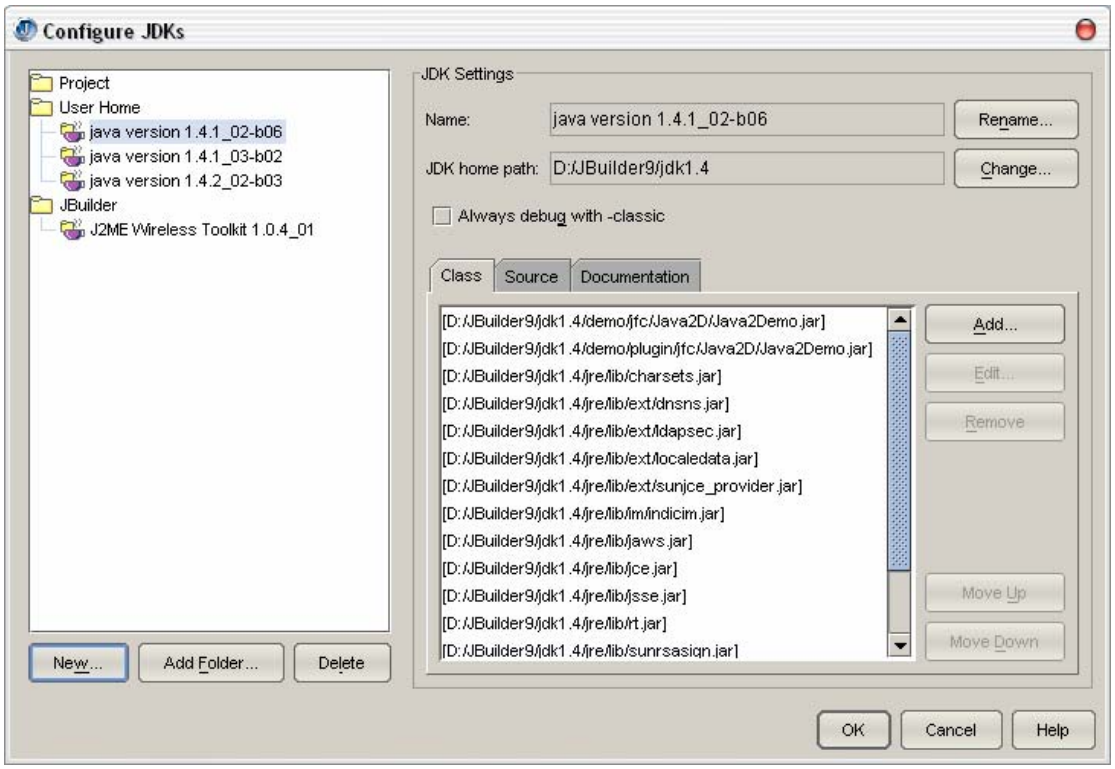
1. Tools→Configure JDKs



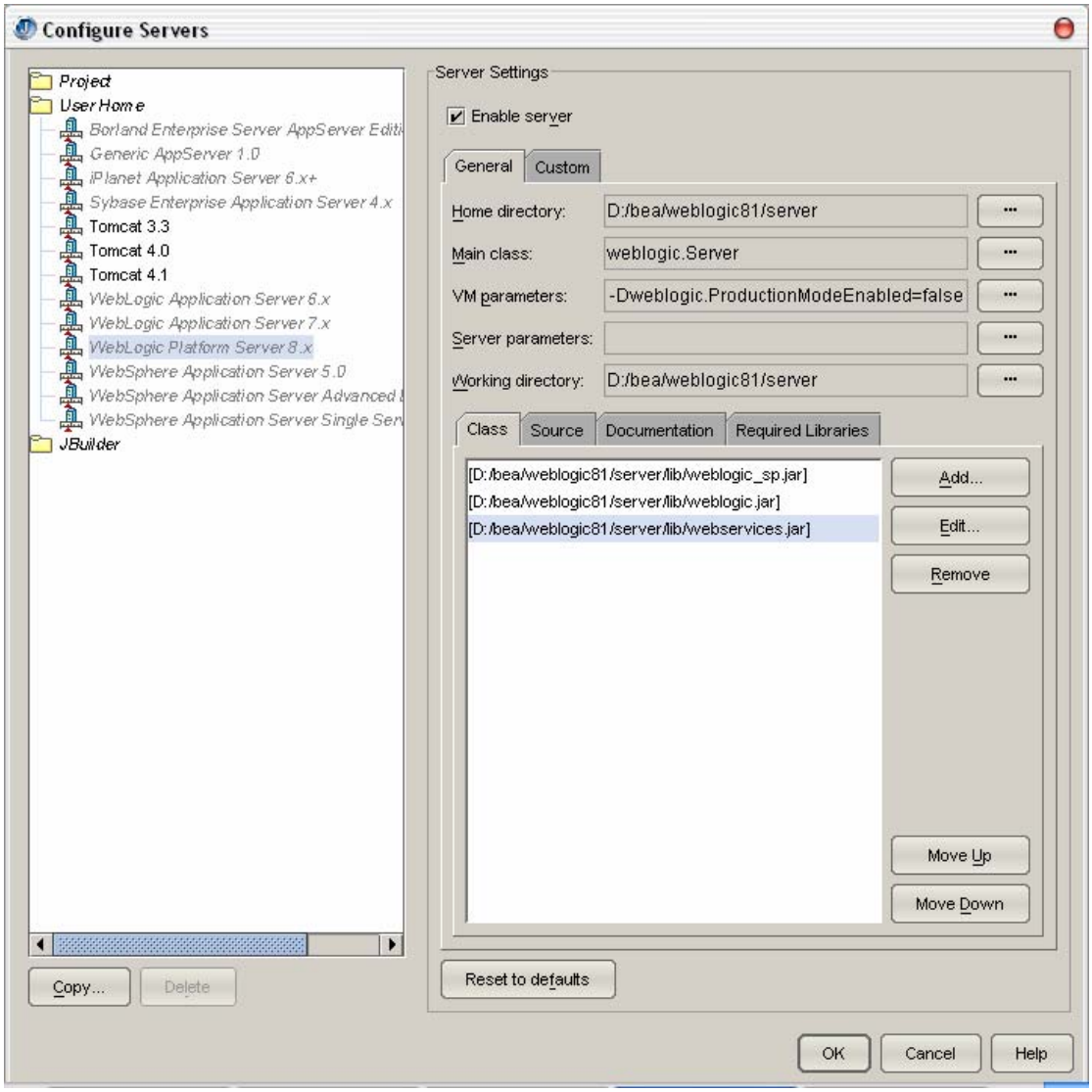
2. 单击 New



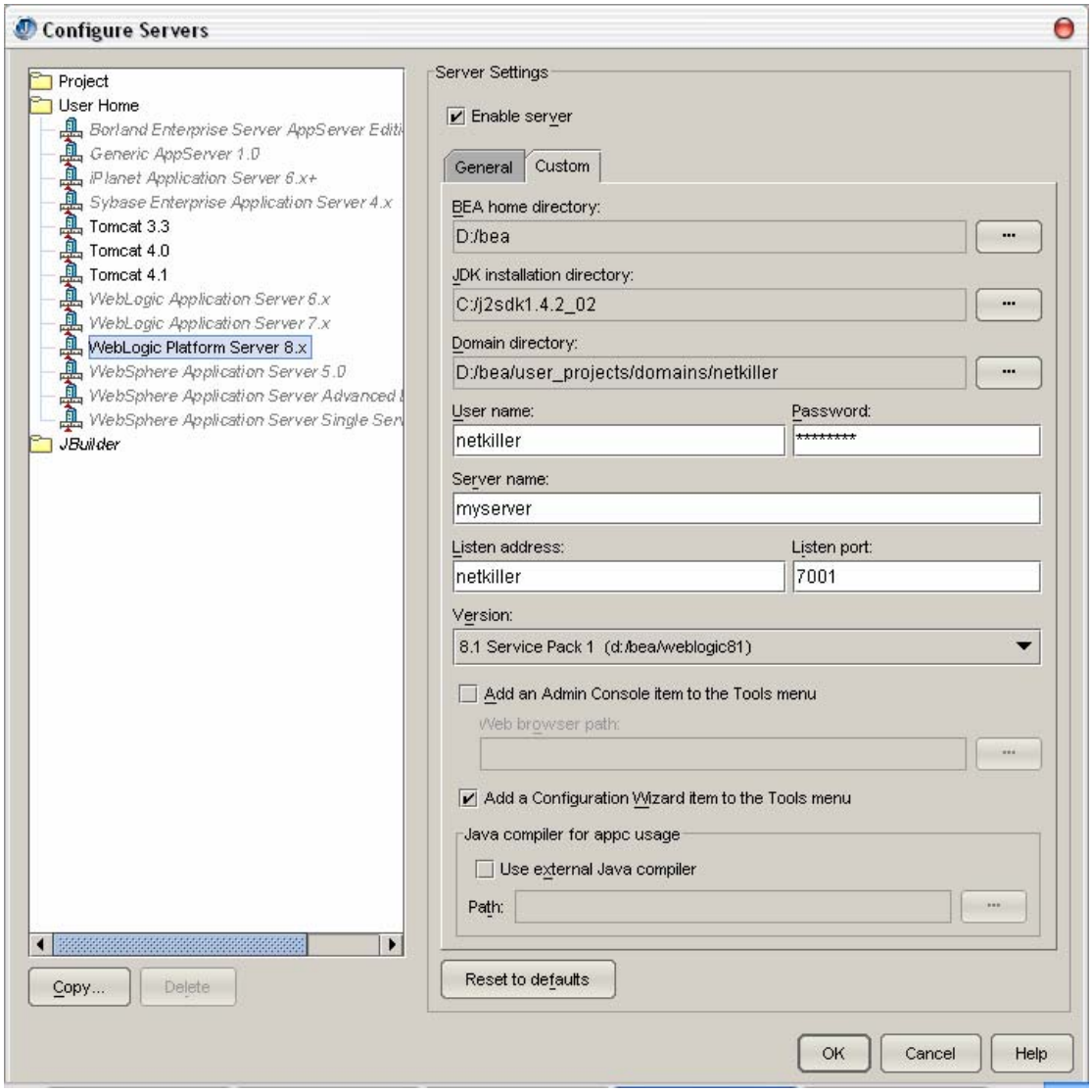
3. 选择 JDK 安装目录



4. 单击 OK，完成
5. Tools → Configure Servers



6. 选择 Weblogic Platform Server 8.x
7. ☑Enable Server
8. Home directory：选择 D:/bea/weblogic81/server
9. 切换到 Custom



10. 设置

BEA home directory：d:/bea

JDK Installation directory：你的 JDK 安装目录

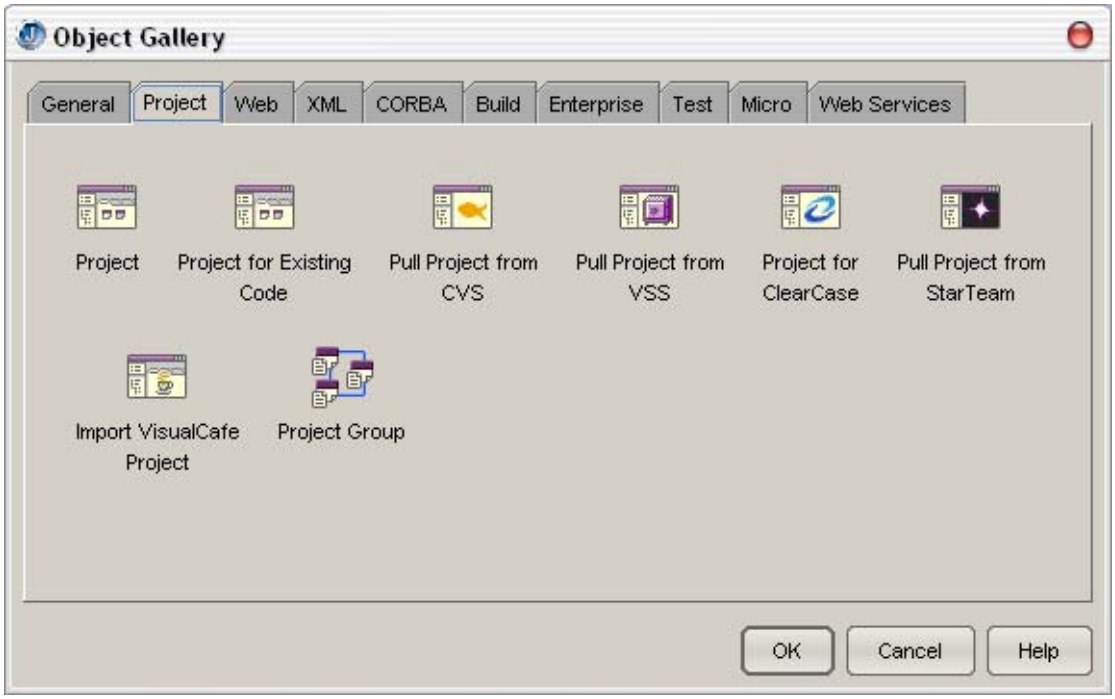
Domain directory：D:/bea/user_projects/domains/netkiller

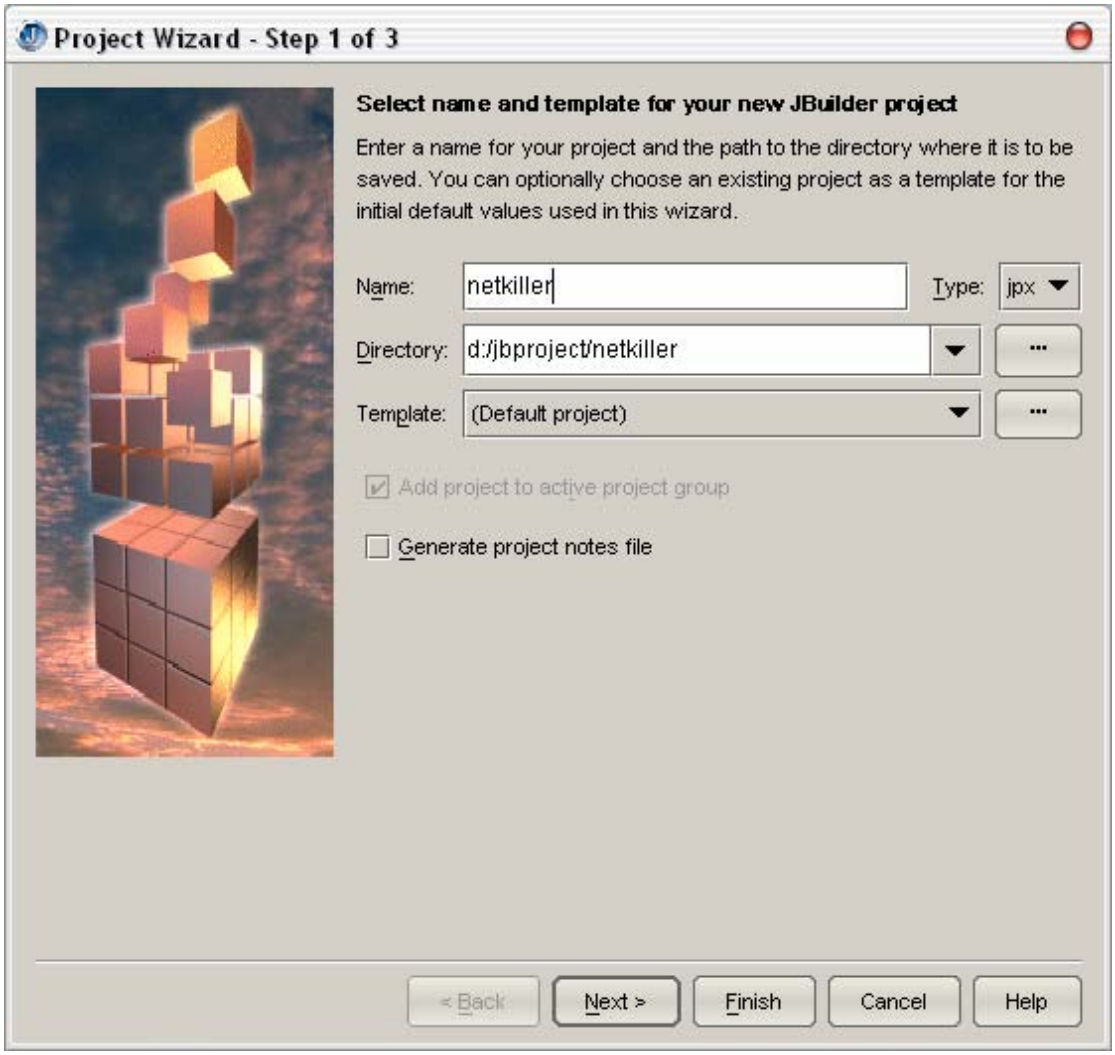
输入密码：********

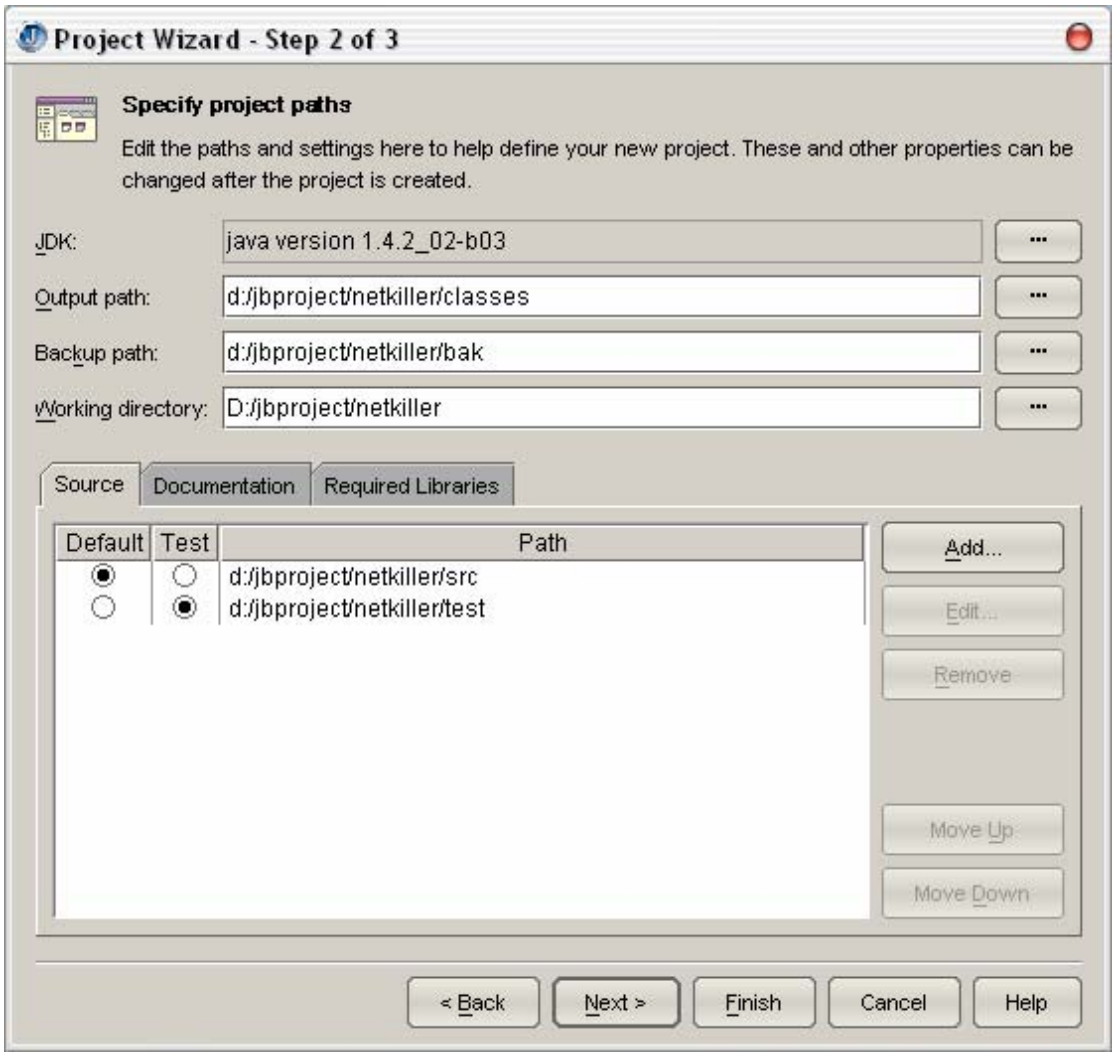
勾去 Add an Console item to the Tools menu

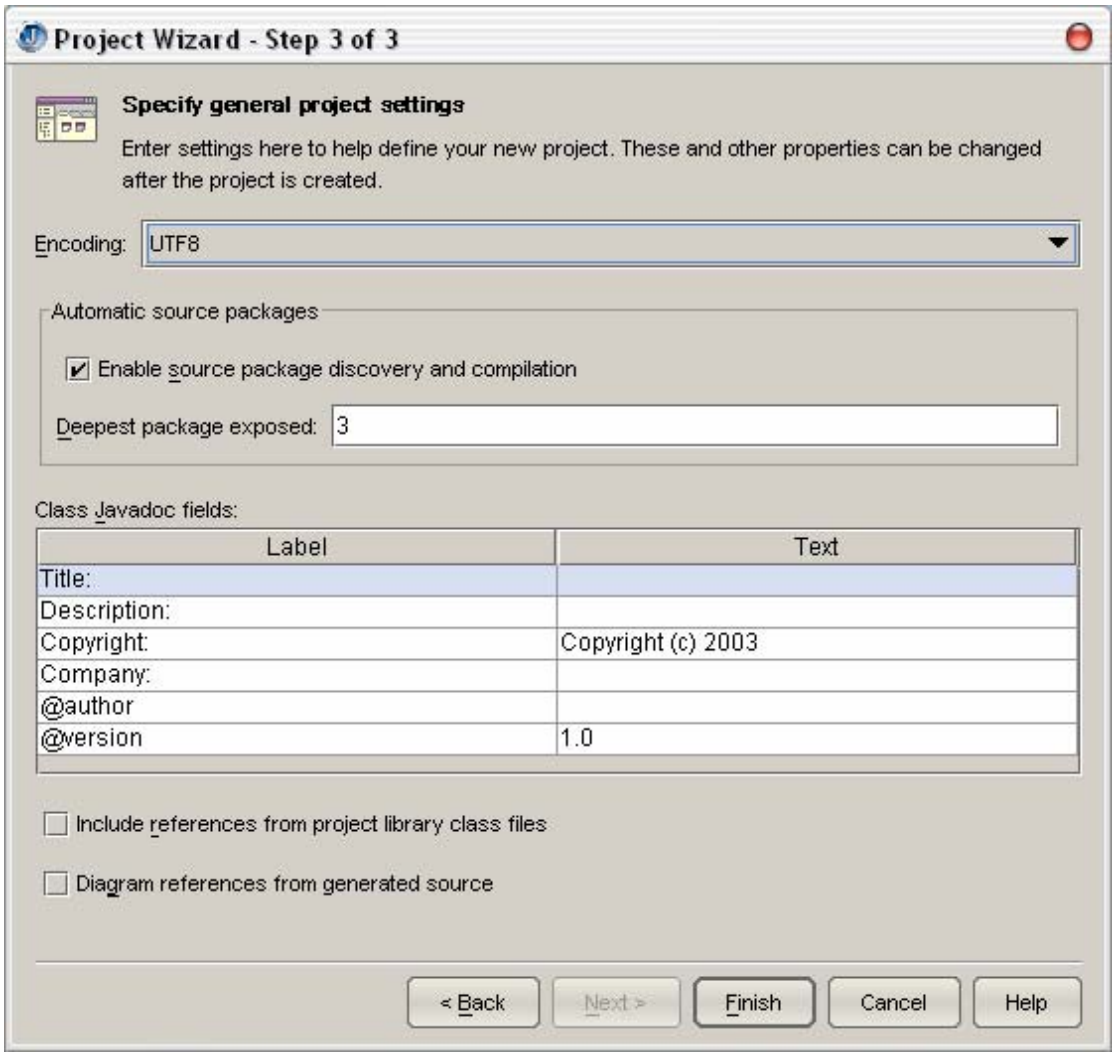
单击 OK 完成

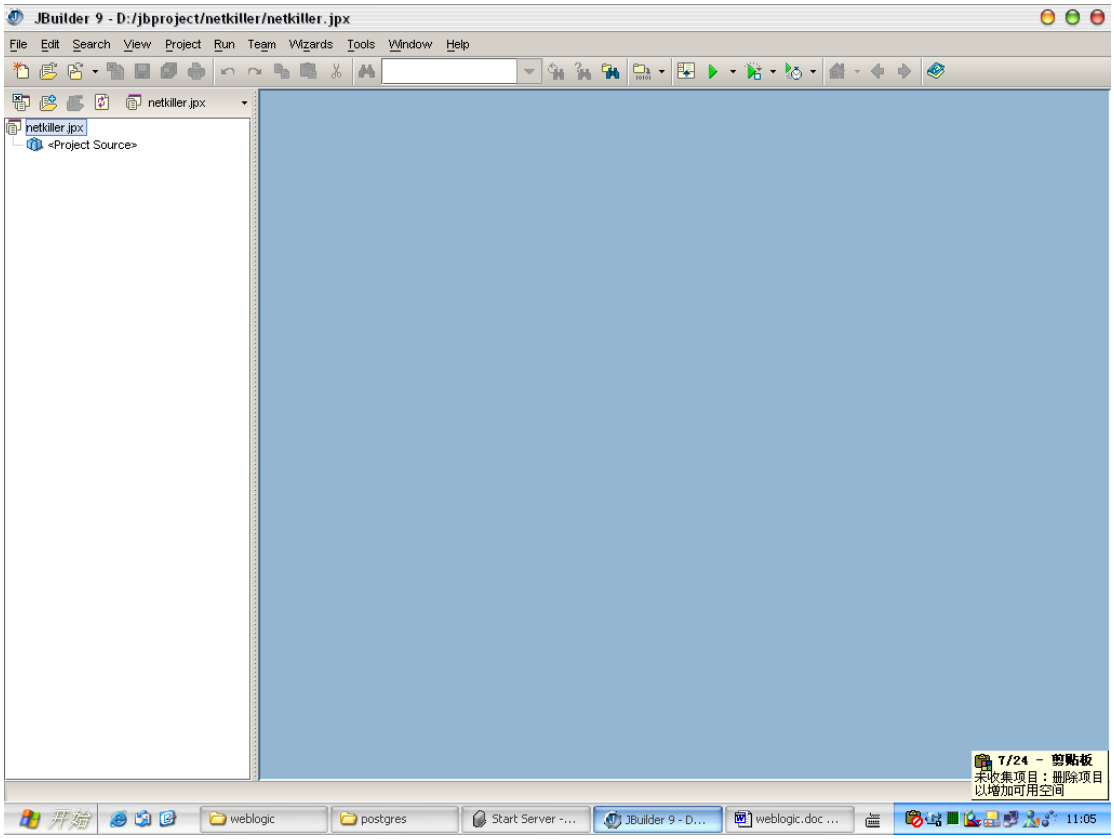
新建项目

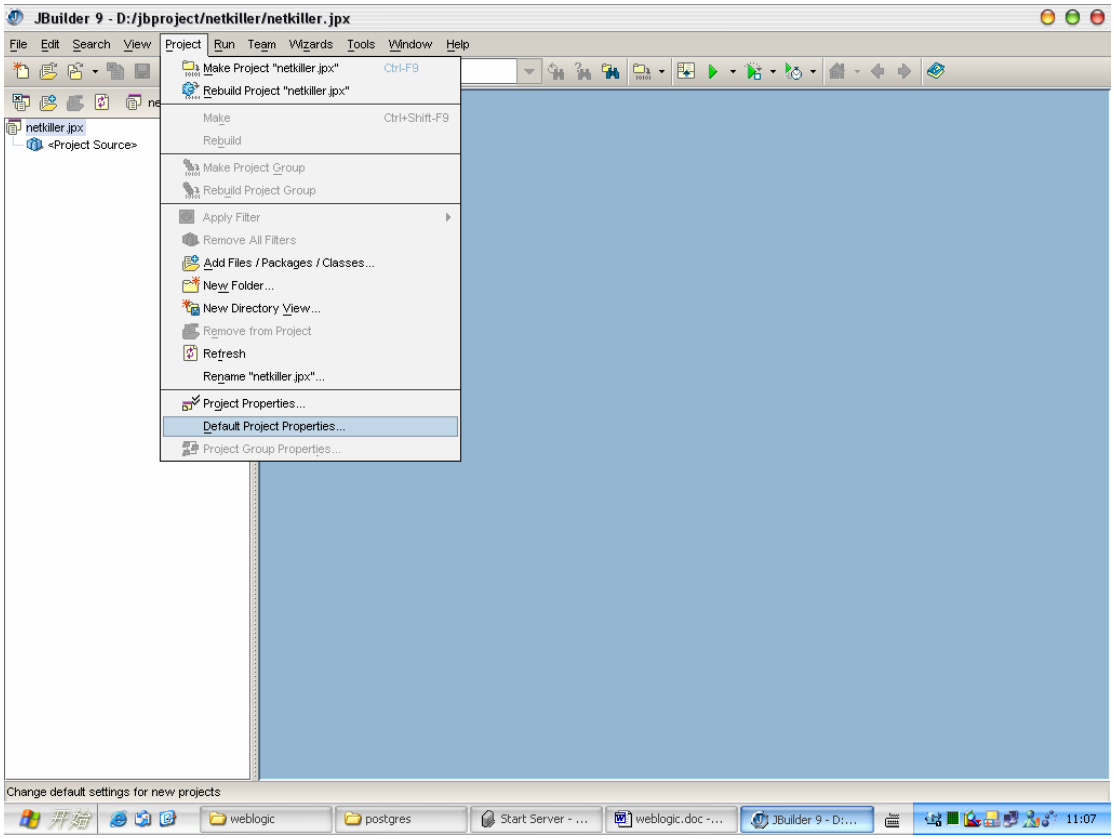

Object Gallery
General
Project
Web
XML
CORBA
Build
Enterprise
Test
Micro
Web Services
Project
Project for Existing Code
Pull Project from CVS
Pull Project from VSS
Project for ClearCase
Pull Project from StarTeam
Import VisualCafe Project
Project Group
OK
Cancel
Help

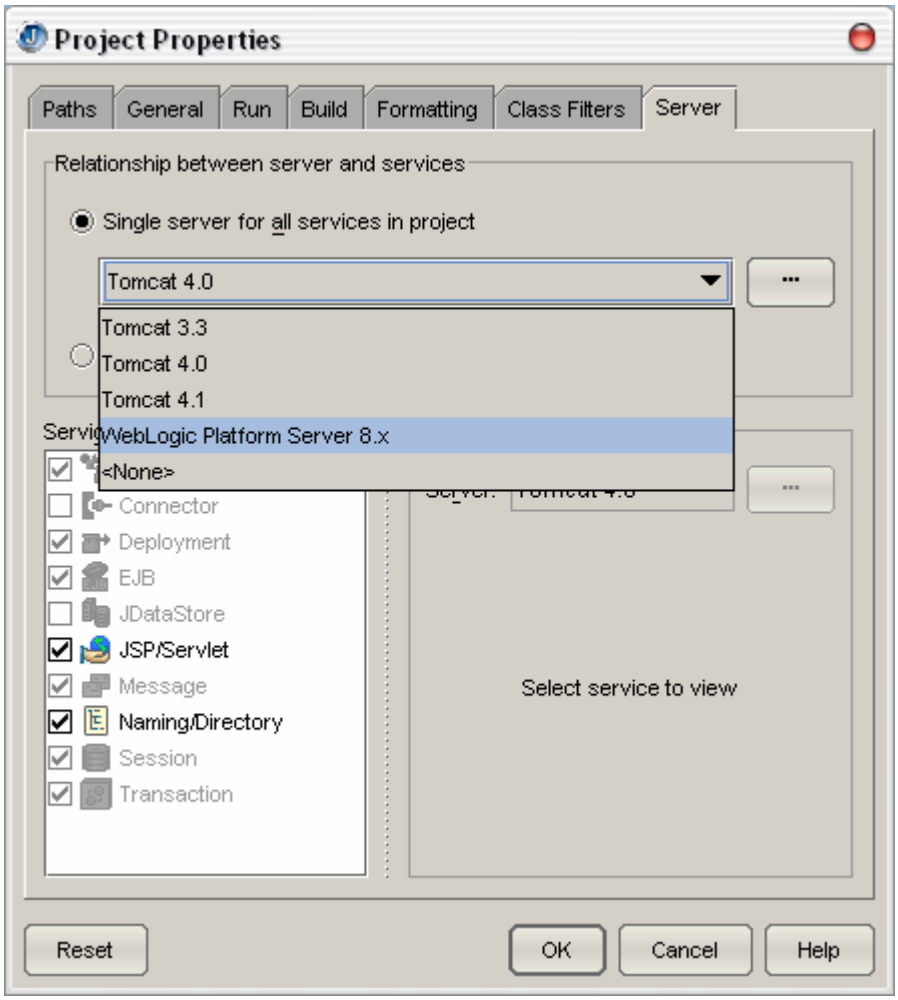

Project Wizard - Step 1 of 3
Select name and template for your new JBuilder project
Enter a name for your project and the path to the directory where it is to be saved. You can optionally choose an existing project as a template for the initial default values used in this wizard.
Name:
netkiller
Type:
jpx
Directory:
d:/jbproject/netkiller
Template:
(Default project)
Add project to active project group
Generate project notes file
< Back
Next >
Finish
Cancel
Help


Project Wizard - Step 2 of 3
Specify project paths
Edit the paths and settings here to help define your new project. These and other properties can be changed after the project is created.
JDK:
java version 1.4.2_02-b03
Output path:
d:/jbproject/netkiller/classes
Backup path:
d:/jbproject/netkiller/bak
Working directory:
D:/jbproject/netkiller
Source
Documentation
Required Libraries
Default
Test
Path
d:/jbproject/netkiller/src
d:/jbproject/netkiller/test
Add...
Edit...
Remove
Move Up
Move Down
< Back
Next >
Finish
Cancel
Help


Project Wizard - Step 3 of 3
Specify general project settings
Enter settings here to help define your new project. These and other properties can be changed after the project is created.
Encoding: UTF8
Automatic source packages
Enable source package discovery and compilation
Deepest package exposed: 3
Class Javadoc fields:
Label
Text
Title:
Description:
Copyright:
Copyright (c) 2003
Company:
@author
@version
1.0
Include references from project library class files
Diagram references from generated source
< Back
Next >
Finish
Cancel
Help


JBuilder 9 - D:/jbproject/netkiller/netkiller.jpx
File Edit Search View Project Run Team Wizards Tools Window Help
netkiller.jpx
<Project Source>
7/24 - 剪貼板
未收集項目：刪除項目以增加可用空間
weblogic
postgres
Start Server -...
JBuilder 9 - D...
weblogic.doc ...
11:05

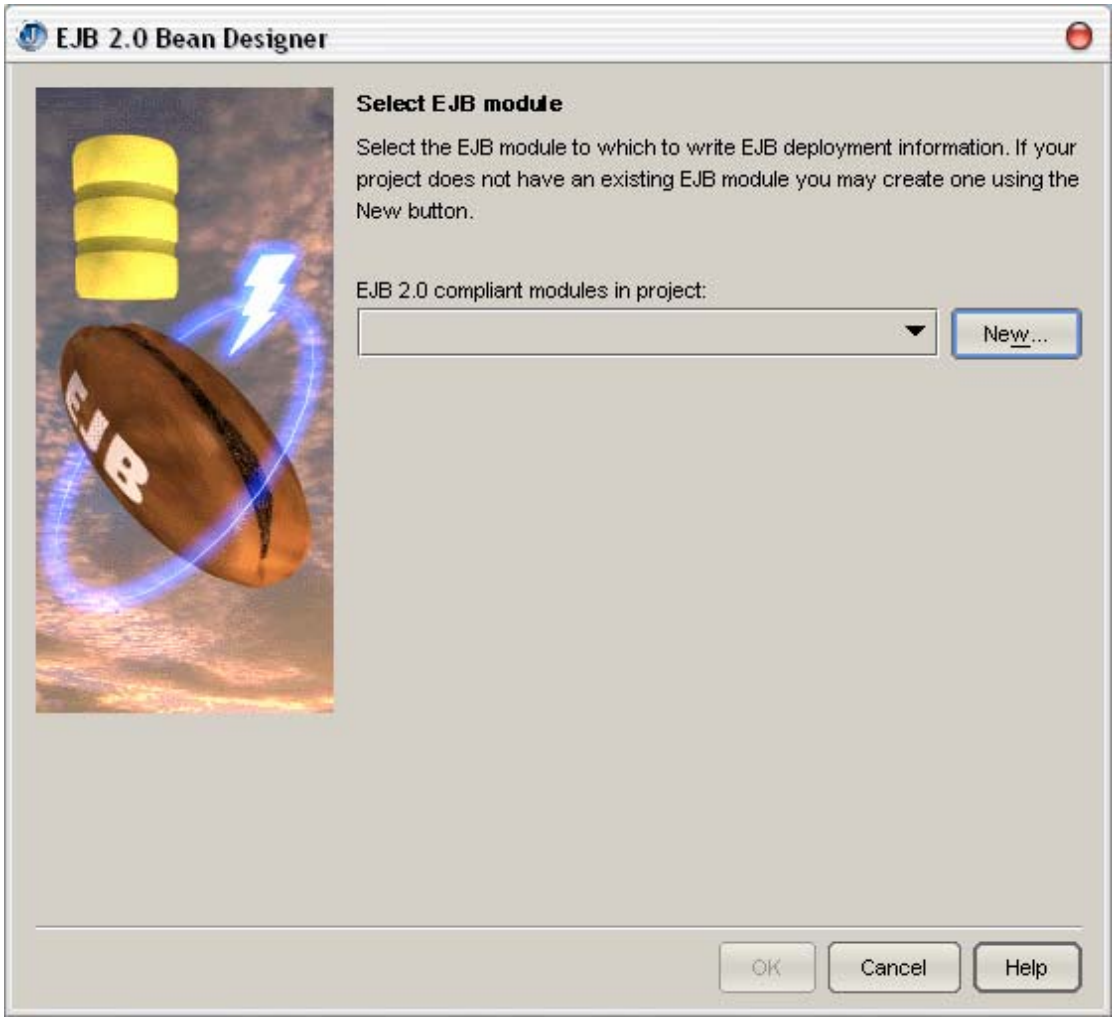

JBuilder 9 - D:/jbproject/netkiller/netkiller.jpx
File Edit Search View Project Run Team Wizards Tools Window Help
Make Project "netkiller.jpx" Ctrl-F9
Rebuild Project "netkiller.jpx"
Make Ctrl+Shift-F9
Rebuild
Make Project Group
Rebuild Project Group
Apply Filter
Remove All Filters
Add Files / Packages / Classes...
New Folder...
New Directory View...
Remove from Project
Refresh
Rename "netkiller.jpx"...
Project Properties...
Default Project Properties...
Project Group Properties...
netkiller.jpx
<Project Source>
Change default settings for new projects
weblogic
postgres
Start Server - ...
weblogic.doc -...
JBuilder 9 - D:...
11:07


Project Properties
Paths
General
Run
Build
Formatting
Class Filters
Server
Relationship between server and services
Single server for all services in project
Tomcat 4.0
Tomcat 3.3
Tomcat 4.0
Tomcat 4.1
WebLogic Platform Server 8.x
<None>
Connector
Deployment
EJB
JDataStore
JSP/Servlet
Message
Naming/Directory
Session
Transaction
Select service to view
Reset
OK
Cancel
Help

单击 OK

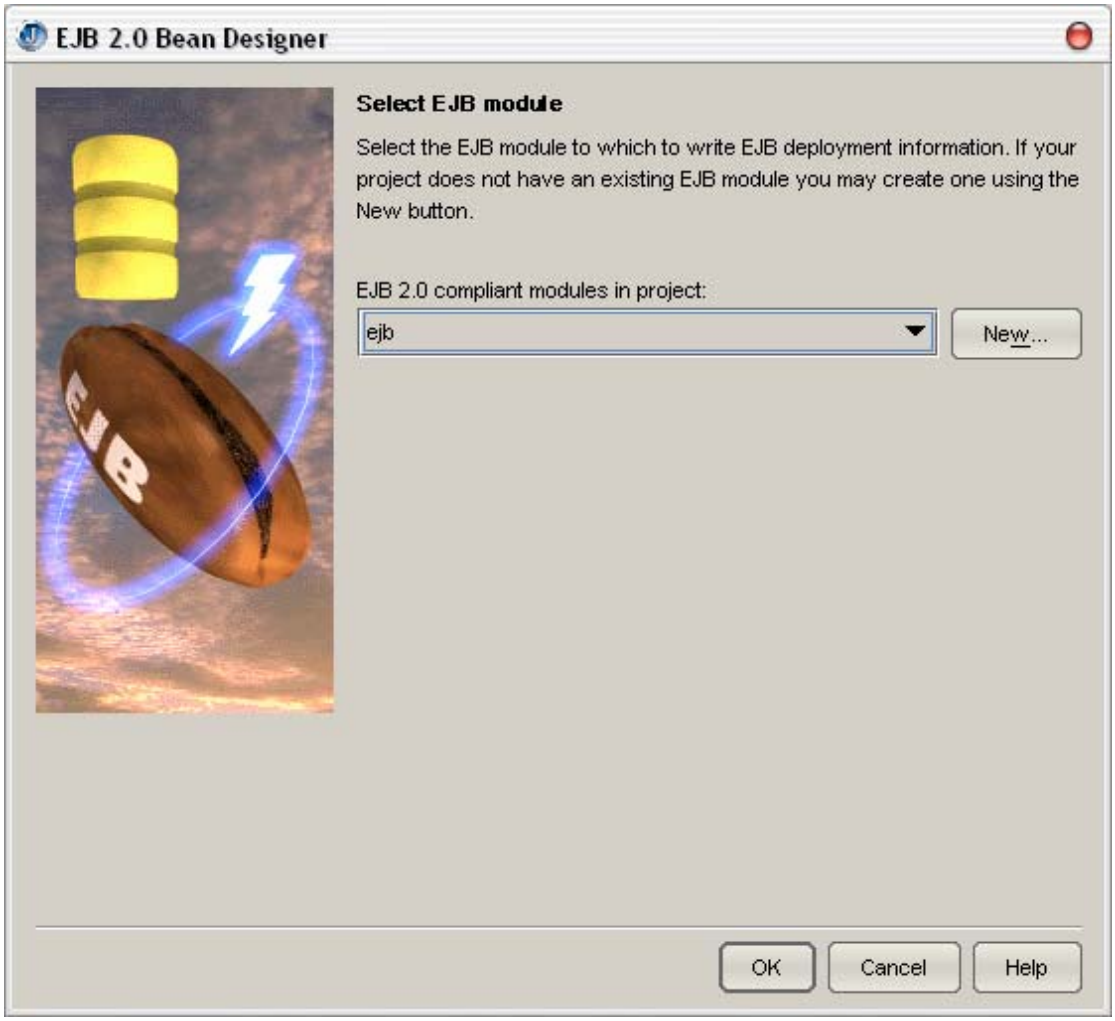
新建


Object Gallery
General
Project
Web
XML
CORBA
Build
Enterprise
Test
Micro
Web Services
EJB Module
EJB Module from Descriptors
Enterprise JavaBean 1.x
EJB 1.x Entity Bean Modeler
EJB 2.0 Bean Designer
EJB Test Client
EAR
JMS
OK
Cancel
Help

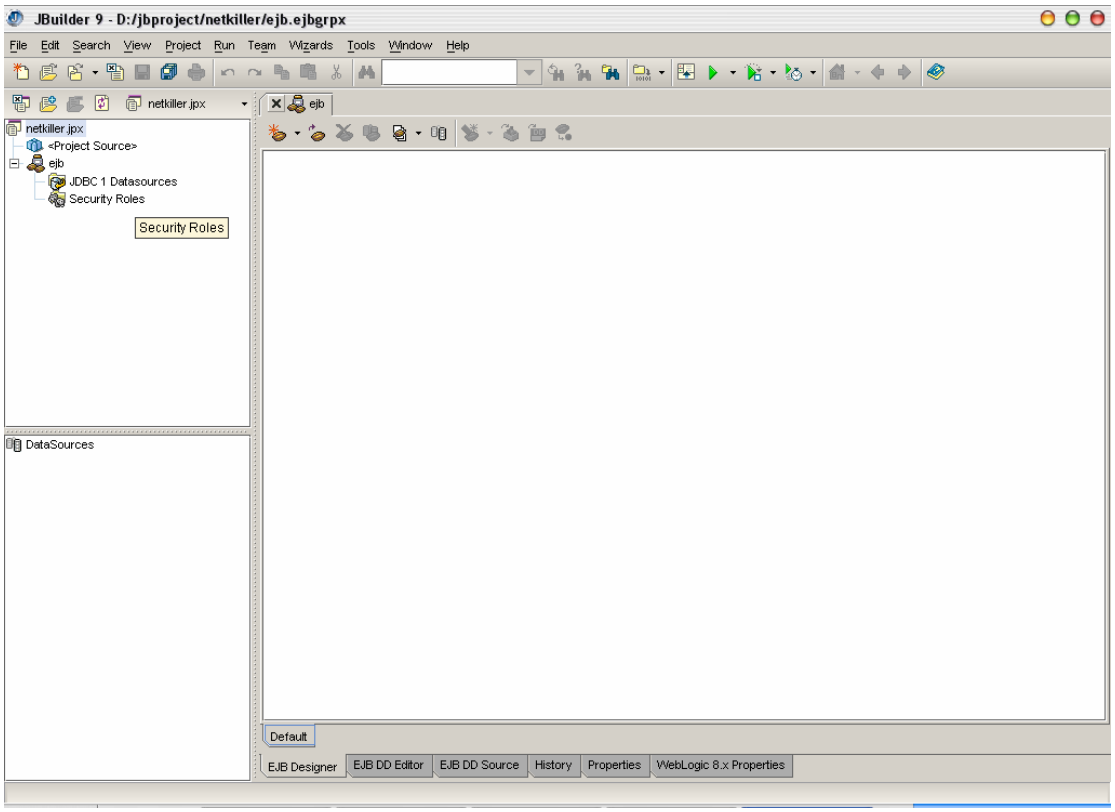

EJB 2.0 Bean Designer
Select EJB module
Select the EJB module to which to write EJB deployment information. If your project does not have an existing EJB module you may create one using the New button.
EJB 2.0 compliant modules in project:
New...
OK
Cancel
Help


EJB Module Wizard
Enter new EJB module details
Fill in the following fields to quickly define and create a new EJB module. Each module defines a deployable EJB jar automatically produced by the build system. The module type selected determines if the content will be recorded in a binary or XML file format.
Name:
ejb
Format:
XML
Version:
EJB 2.0 compliant
Output JAR file
Name:
ejb.jar
Path:
D:/jbproject/netkiller
...
OK
Cancel
Help

OK


EJB 2.0 Bean Designer
Select EJB module
Select the EJB module to which to write EJB deployment information. If your project does not have an existing EJB module you may create one using the New button.
EJB 2.0 compliant modules in project:
ejb
New...
OK
Cancel
Help

OK



14 附录

14.1 实例

```
/*
#########################################################################
Created         2003-7-21
Modified 2003-7-22
Project         广东服装网二期
Model           会员系统
Company         广东服装网
Author          陈景峰
Version         1.0
Database PostgreSQL 7.3.3
#########################################################################
*/

-- ============================================================
```

```
--    'CREATE DATABASE'
-- ========================================================
-- --------------------------------------------------------
-- CREATE USER netkiller WITH PASSWORD 'chen';
-- CREATE DATABASE netkiller WITH OWNER = netkiller TEMPLATE = template0 ENCODING = 'EUC_CN';
-- postgres=# CREATE DATABASE member WITH OWNER = netkiller TEMPLATE = template0 ENCODING = 'UNICODE';
-- createlang plpgsql netkiller
-- createlang plpgsql member

-- --------------------------------------------------------

Drop index "group_index";
Drop table "group" CASCADE;
-- --------------------------------------------------------
Drop index "role_index";
Drop table "role" CASCADE;
-- --------------------------------------------------------
Drop table "rolemember" Restrict;
-- --------------------------------------------------------
Drop table "groupmember" CASCADE;
-- --------------------------------------------------------
Drop table "trust" Restrict;
-- --------------------------------------------------------
Drop table userinfo CASCADE;
DROP TABLE system_log CASCADE;
DROP TABLE user_log CASCADE;
-- --------------------------------------------------------
Drop table "user" CASCADE;

-- ========================================================
--    'user'
-- ========================================================
Create table "user"
(
        "id" Serial NOT NULL,
        "userid" Varchar(50) NOT NULL,
        "passwd" Varchar(50),
        "name"      Varchar(20)NOT NULL ,
        "nickname" Varchar(20)NOT NULL ,
        "active" Boolean Default 'F',
        "email"      Varchar(50) NOT NULL,
        "question" Varchar(255) NOT NULL,
```

```
        "answer" Varchar(255) NOT NULL,
        "begin_date" Timestamp Default now(),
        "end_date" Timestamp Default now(),
        UNIQUE   (userid,email),
  primary key ("id")
);
Create index "user_index" on "user" using btree ("id","userid");

-- ----------------------------------------------------
--    'vuser'
-- ----------------------------------------------------
drop view vuser;
CREATE VIEW vuser AS
        SELECT u.id,u.userid,u."name",u.nickname,
                CASE WHEN u.active=true THEN 'Y' ELSE 'N' END as "active",
                u.email,u.question,u.answer,
                to_char(u.begin_date,'YYYY-MM-DD HH:MI:SS') as begin_date,
                to_char(u.end_date,'YYYY-MM-DD HH:MI:SS') as end_date
        FROM "user" u
        Order By u.id;

-- DROP FUNCTION user_tri_func() CASCADE ;
CREATE OR REPLACE FUNCTION user_tri_func () RETURNS TRIGGER AS '
        DECLARE
                -- old_id ALIAS FOR OLD.id;
                -- new_id CONSTANT INTEGER := New.id;
        BEGIN
                IF TG_OP = "DELETE" THEN
                        Delete from groupmember where uid = OLD.id;
                        Delete from rolemember where uid = OLD.id;
                        Delete from userinfo where uid = OLD.id;
                        Delete from trust where uid = OLD.id;
                        Delete from system_log where uid = OLD.id;
                        Delete from user_log where uid = OLD.id;
        END IF;
                IF TG_OP = "INSERT" THEN
                        -- INSERT INTO company(uid) values(NEW.id);
                END IF;
        RETURN OLD;
        END;
' LANGUAGE 'plpgsql';
-- DROP TRIGGER user_delete_tri on user;
-- DROP TRIGGER user_insert_tri on user;
CREATE TRIGGER users_delete_tri
```

```
        BEFORE Delete ON "user" FOR EACH ROW
        EXECUTE PROCEDURE user_tri_func ();

CREATE TRIGGER users_insert_tri
        AFTER INSERT ON "user" FOR EACH ROW
        EXECUTE PROCEDURE user_tri_func ();
/*
-- ========================================================
--    'privileges'
-- ========================================================
--Drop table "privileges" Restrict;
Create table "privileges"
(
      "id" integer NOT NULL UNIQUE ,
      "read" integer,
      "update" integer,
      "write" integer,
  primary key ("id")
);
*/
-- ========================================================
--    'group'
-- ========================================================
Create table "group"
(
      "id" Serial NOT NULL UNIQUE,
      "groupname" Varchar(20) NOT NULL,
      "description" Varchar(255),
      UNIQUE   (groupname),
      PRIMARY KEY ("id")
);
Create index "group_index" on "group" using btree ("id","groupname");
-- ========================================================
--    'groupmember'
-- ========================================================
Create table "groupmember"
(
      "id" Serial NOT NULL UNIQUE,
      "gid" integer NOT NULL Default 0,
      "uid" integer NOT NULL Default 0,
      UNIQUE   (gid,uid),
      primary key ("id")
);
-- --------------------------------------------------
```

```
--    'vgroupmember'
-- ----------------------------------------------------
-- DROP VIEW vgroupmember;
CREATE VIEW vgroupmember AS
       SELECT gm.id,gm.gid,g.groupname,gm.uid,u.userid,u.name
              FROM "group" g,"user" u,groupmember gm
              Where u.id = gm.uid and g.id = gm.gid
       ORDER BY gm.id;
/*
-- ----------------------------------------------------
--    'vgroup'
-- ----------------------------------------------------
-- PHP Interface View
CREATE OR REPLACE VIEW vgroup AS
       SELECT gm.id,gm.uid,gm.gid,g.groupname
       FROM "group" g,groupmember gm
       Where g.id = gm.gid
       ORDER BY gm.id;
*/
-- ----------------------------------------------------
--    'RULE'
-- ----------------------------------------------------
CREATE RULE group_rule AS ON Delete TO "group"
       DO Delete From groupmember where gid = OLD.id;

-- ======================================================
--    'role'
-- ======================================================
-- drop table role CASCADE;
Create table "role"
(
       "id" Serial NOT NULL UNIQUE,
       "rolename" Varchar(20) NOT NULL,
       "description" Varchar(255),
       UNIQUE   (rolename),
       PRIMARY KEY ("id")
);
Create index "role_index" on "role" using btree ("id","rolename");

-- ======================================================
--    'rolemember'
-- ======================================================
-- drop table rolemember CASCADE ;
Create table "rolemember"
```

```
(
        "id" Serial NOT NULL UNIQUE,
        "rid" integer NOT NULL Default 0,
        "uid" integer NOT NULL Default 0,
        UNIQUE   (rid,uid),
  primary key ("id")
);
-- ----------------------------------------------------
--    'vrolemember'
-- ----------------------------------------------------
CREATE VIEW vrolemember AS
        SELECT rm.id,rm.rid,r.rolename,rm.uid,u.userid,u.name
                FROM "role" r,"user" u,rolemember rm
                Where u.id = rm.uid and r.id = rm.rid
                ORDER BY rm.id;

/*
-- ----------------------------------------------------
--    'vrole'
-- ----------------------------------------------------
CREATE OR REPLACE VIEW vrole AS
        SELECT rm.id,rm.uid,rm.rid,r.rolename
        FROM "user" u,role r,rolemember rm
        Where u.id = rm.uid and r.id = rm.rid
        ORDER BY rm.id;
*/
-- ----------------------------------------------------
--    'RULE'
-- ----------------------------------------------------
CREATE RULE role_rule AS ON Delete TO role
        DO Delete From rolemember where rid = OLD.id;

-- ====================================================
--    'Foreign Key'
-- ====================================================
Alter table "groupmember" add    foreign key ("uid") references "user" ("id") on update restrict on delete restrict;
Alter table "groupmember" add    foreign key ("gid") references "group" ("id") on update restrict on delete restrict;
Alter table "rolemember" add    foreign key ("uid") references "user" ("id") on update restrict on delete restrict;
Alter table "rolemember" add    foreign key ("rid") references "role" ("id") on update restrict on delete restrict;
```

```
-- ======================================================
--    'trust'
-- ======================================================
Create table "trust"
(
      "id" Serial NOT NULL UNIQUE,
      "uid" integer NOT NULL Default 0,
      "rate" Varchar(20) Default '0' Check (rate in ('0','1','2','3','4','5')),
      primary key ("uid")
);
Alter table "trust" add    foreign key ("uid") references "user" ("id") on update restrict on delete restrict;
-- ======================================================
--    'userinfo'
-- ======================================================
-- Drop table userinfo CASCADE;
Create table "userinfo"
(
      id      Serial NOT NULL UNIQUE,
      uid   integer NOT NULL Default 0,
      tel    Varchar (20) NOT NULL,
      fax varchar(20),
      email varchar(60),
      province varchar(8),
      city varchar(10),
      address varchar(255) DEFAULT '' NOT NULL,
      postalcode        varchar(6)      DEFAULT '' NOT NULL,       -- post office code
      bank          varchar(20)      DEFAULT '' NOT NULL,     -- bank
      bankaccount      varchar(20)      DEFAULT '' NOT NULL,     -- Back Account
      "rate" Varchar(20) Default '0' Check (rate in ('0','1','2','3','4','5')),
      PRIMARY KEY(id),
      FOREIGN KEY (uid) REFERENCES "user" (id)
);


-- ======================================================
--    'log system'
-- ======================================================
Create table user_log
(
      id     Serial NOT NULL UNIQUE,
      uid integer NOT NULL Default 0,
      ip     inet,
      status varchar(255),
      PRIMARY KEY("id"),
```

```
	FOREIGN KEY (uid) REFERENCES "user" (id)
);
-- drop table system_log CASCADE;
Create table system_log
(
	id	Serial NOT NULL UNIQUE,
	uid integer NOT NULL Default 0,
	ip	inet ,
	status varchar(255),
	description varchar(255),
	login_date Timestamp Default now(),
	PRIMARY KEY("id"),
	FOREIGN KEY (uid) REFERENCES "user" (id)
);
-- ------------------------------------------------------
--	'system_log'
-- ------------------------------------------------------
drop view vsystem_log ;
CREATE OR REPLACE VIEW vsystem_log AS
	SELECT u.userid,u.name,log.ip,log.status,log.description,
		to_char(log.login_date,'YYYY-MM-DD HH:MI:SS') as login_date
	FROM "user" u,system_log log
	Where log.uid = u.id
	ORDER BY log.id;
-- ------------------------------------------------------
-- 'Function'
-- ------------------------------------------------------
-- DROP FUNCTION add_system_log(integer,inet,varchar);
CREATE OR REPLACE FUNCTION add_system_log(integer,inet,varchar,varchar) RETURNS boolean
AS '
	DECLARE
		vUID		ALIAS FOR $1;
		vIP	ALIAS FOR $2;
		vSTATUS		ALIAS FOR $3;
		vDESC	ALIAS FOR $4;
	BEGIN
		insert into system_log(uid,ip,status,description) values(vUID,vIP,vSTATUS,vDESC);
		RETURN true;
	END;
' LANGUAGE 'plpgsql';
select add_system_log(1,'127.0.0.1','Create Database','Initialization Database');

-- ------------------------------------------------------
--	'Insert Data'
```

```
-- ----------------------------------------------------
insert into "user"(userid,passwd,name,nickname,email,question,answer) values('sysop','chen','chen','chen','chen@chen.com','xxxxxxx','xxxx');
insert into "user"(userid,passwd,name,nickname,email,question,answer) values('admin','chen','chen','chen','chen@chen.com','xxxxxxx','xxxx');
insert into "user"(userid,passwd,name,nickname,email,question,answer) values('netkiller','chen','陈景峰','Netkiller','chen@chen.com','xxxxxxx','xxxx');
update "user" set active='true' where userid='sysop';
-- ----------------------------------------------------
--  'Insert Data'
-- ----------------------------------------------------
Insert into "group"(groupname,description) values('System','系统管理员');
Insert into "group"(groupname,description) values('Administrator','站点管理员');
Insert into "group"(groupname,description) values('gold','gold diamond');
Insert into "group"(groupname,description) values('silver','silver diamond');
Insert into "group"(groupname,description) values('advance','高级会员');
Insert into "group"(groupname,description) values('free','免费会员');

-- ----------------------------------------------------
--  'Insert Data'
-- ----------------------------------------------------
Insert into "role"(rolename,description) values('System','系统管理员');
Insert into "role"(rolename,description) values('Administrator','站点管理员');
Insert into "role"(rolename,description) values('gold','gold diamond');
Insert into "role"(rolename,description) values('silver','silver diamond');
Insert into "role"(rolename,description) values('advance','高级会员');
Insert into "role"(rolename,description) values('free','免费会员');

insert into groupmember(gid,uid) values((select id from "group" where groupname ='System'),(select id from vuser where userid='sysop'));
insert into rolemember(rid,uid) values((select id from role where rolename ='System'),(select id from vuser where userid='sysop'));
```

14.2 实例

```
--  主机: localhost  数据库  : XXXX
-- Create Database XXXX;
-- DROP DATABASE XXXX;
-- createdb -E EUC_CN XXXX
-- createlang plpgsql XXXX
--    CREATE USER XXXX WITH PASSWORD "XXXXXX";

-- ----------------------------------------------------
```

```
--    'siteuser'
-- ------------------------------------------------------
--DROP TABLE IF EXISTS siteuser;
DROP TABLE siteuser CASCADE;
DROP SEQUENCE siteuser_id_seq;
DROP INDEX siteuser_id_index;

CREATE TABLE siteuser (
      id integer DEFAULT nextval('siteuser_id_seq') NOT NULL,
      username   varchar(20) DEFAULT '0' NOT NULL,
      Password   varchar(50) DEFAULT '0' NOT NULL,
      realname    varchar(10) DEFAULT '0' NOT NULL,
      email              varchar(50) DEFAULT '0' NOT NULL,
      create_date    timestamp DEFAULT now(),
      modify_date    timestamp DEFAULT now(),
      UNIQUE              (id,username),
      PRIMARY     KEY (id)
);
CREATE SEQUENCE siteuser_id_seq;
CREATE INDEX siteuser_id_index ON siteuser (id);

DROP FUNCTION siteuser_tri_func() CASCADE ;
CREATE OR REPLACE FUNCTION siteuser_tri_func () RETURNS opaque AS '
--       DECLARE
--           user_id CONSTANT INTEGER := OLD.id;
      BEGIN
           IF TG_OP = "DELETE" THEN
                Delete from company where uid = OLD.id;
                Delete from link where uid = OLD.id;
                Delete from product_sort where uid = OLD.id;
                Delete from news where uid = OLD.id;
                Delete from count where uid = OLD.id;
                Delete from guestbook where uid = OLD.id;
                Delete from clientinfo where uid = OLD.id;
                Delete from column_bar where uid = OLD.id;
                Delete from drumbeating where uid = OLD.id;
           END IF;
           IF TG_OP = "INSERT" THEN
                INSERT INTO company(uid) values(NEW.id);
                INSERT                INTO                count(uid,number,fontcolor,backgroundcolor)
values(NEW.id,0,"ffffff","000000");
                INSERT                INTO                drumbeating(uid,logourl,bannerurl)
values(NEW.id,"default_logo.jpg","default_banner.jpg");
           END IF;
```

```
        RETURN OLD;
        END;
' LANGUAGE 'plpgsql';


DROP TRIGGER siteuser_delete_tri on siteuser;
CREATE TRIGGER siteuser_delete_tri
            BEFORE Delete ON siteuser FOR EACH ROW
            EXECUTE PROCEDURE siteuser_tri_func ();

DROP TRIGGER siteuser_insert_tri on siteuser;
CREATE TRIGGER siteuser_insert_tri
            AFTER INSERT ON siteuser FOR EACH ROW
            EXECUTE PROCEDURE siteuser_tri_func ();

DROP FUNCTION adduser(varchar,varchar);
CREATE OR REPLACE FUNCTION adduser(varchar,varchar) RETURNS boolean AS '
        DECLARE
                bool boolean := false;
                name text;
                uid     integer;
                user ALIAS FOR $1;
                pass ALIAS FOR $2;
                su      siteuser%ROWTYPE;
        BEGIN
                SELECT INTO name username FROM siteuser WHERE username = user;
                IF NOT FOUND then
                        insert into siteuser(username,password) values(user,pass);
                        SELECT INTO uid id FROM siteuser WHERE username = user;
                        bool := true;
                ELSE
                        bool := false;
                        RAISE NOTICE "Calling adduser() return %",bool;
                END IF;
        RETURN bool;
        END;
' LANGUAGE 'plpgsql';
--select adduser('ccscc','eeee');
--select * from siteuser ;
--      record
DROP FUNCTION deluser(integer);
CREATE OR REPLACE FUNCTION deluser(integer) RETURNS boolean AS '
        DECLARE
                bool boolean := false;
```

```
        userid text;
    BEGIN
        SELECT INTO userid id FROM siteuser WHERE id = $1;
        IF FOUND then
            delete from siteuser where id = $1;
            bool := true;
        ELSE
        END IF;
        RETURN bool;
    END;
' LANGUAGE 'plpgsql';
--select deluser(28);


DROP VIEW vsiteuser;
CREATE VIEW vsiteuser AS
    SELECT su.id,su.username,
        to_char(su.create_date,'YYYY-MM-DD HH:MI:SS') as date
        FROM siteuser su
        ORDER BY su.id;

-- ----------------------------------------------------
--   'style'
-- ----------------------------------------------------
DROP TABLE style;
DROP SEQUENCE     style_id_seq;
DROP INDEX        style_id_index;
DROP VIEW vstyle;

CREATE TABLE style (
    id              integer DEFAULT nextval('style_id_seq') NOT NULL,
--      uid         integer    DEFAULT '1' NOT NULL,    -- with foreign key
    number          varchar(20) DEFAULT '' NOT NULL,   -- sytle number
    stylename       varchar(20) DEFAULT '' NOT NULL,
    miniature   varchar(50) DEFAULT '' NOT NULL,   -- mini image
    url             varchar(50) DEFAULT '' NOT NULL,  --          link          url                    eg.
/usr/local/apache/htdocs/autosite/style/n0001
    description     text        DEFAULT '' NOT NULL,
    create_date   timestamp DEFAULT now() ,
    modify_date   timestamp DEFAULT now() ,
    PRIMARY KEY (id),
    UNIQUE (id)
--   FOREIGN KEY (uid) REFERENCES siteuser (id)
);
```

```
CREATE SEQUENCE   style_id_seq;
CREATE INDEX          style_id_index ON style (id);

CREATE VIEW vstyle AS
      SELECT st.id,st.number,st.stylename,st.miniature,st.url,st.description,
            to_char(st.create_date,'YYYY-MM-DD HH:MI:SS') as date
            FROM style st
            ORDER BY st.id;

insert into style(number,stylename,miniature,url) values('0001','pink','../style/images/style0001.png','');
insert into style(number,stylename,miniature,url) values('0002','green','../style/images/style0002.png','');
insert into style(number,stylename,miniature,url) values('0003','cyan','../style/images/style0003.png','');
insert into style(number,stylename,miniature,url) values('0004','orange','../style/images/style0004.png','');
insert into style(number,stylename,miniature,url) values('0005','offwhite','../style/images/style0005.png','');

-- ----------------------------------------------------
--    'company'
-- ----------------------------------------------------
DROP TABLE company;
DROP SEQUENCE company_id_seq;
DROP INDEX company_id_index;
DROP VIEW vcompany;

CREATE TABLE company (
    id                  integer DEFAULT nextval('company_id_seq') NOT NULL,
    uid                 integer      DEFAULT '1' NOT NULL,  -- with foreign key
    linkman             varchar(20)    DEFAULT '' NOT NULL,
    cncname             varchar(50)    DEFAULT '' NOT NULL,       -- Chinese Companyname
    encname             varchar(50)    DEFAULT '' NOT NULL,       -- English Companyname
    email         varchar(50)    DEFAULT '' NOT NULL,
    telephone     varchar(20)    DEFAULT '' NOT NULL,       -- telephone number
    fax                 varchar(20)    DEFAULT '' NOT NULL,       -- company fax
    Province_id         integer      DEFAULT '1' NOT NULL,  -- with foreign key
    state         varchar(20)    DEFAULT '' NOT NULL,       -- company state
    city                varchar(20)    DEFAULT '' NOT NULL,       -- company city
    address             varchar(50)    DEFAULT '' NOT NULL,       -- company address
    postalcode    varchar(6)     DEFAULT '' NOT NULL,       -- post office code
    range         text DEFAULT '' NOT NULL,
    details             text DEFAULT '' NOT NULL,       -- company details
    other         text ,                                  -- company other
    bank                varchar(20)     DEFAULT '' NOT NULL,     -- bank
    bankaccount         varchar(20)     DEFAULT '' NOT NULL,     -- Back Account
    Category_id         integer      DEFAULT '1' NOT NULL,
    style_id            integer    DEFAULT '1' NOT NULL,
```

```
    create_date    timestamp DEFAULT now() ,
    modify_date    timestamp DEFAULT now() ,
    UNIQUE (id,uid),
    PRIMARY KEY (id),
    FOREIGN KEY (uid) REFERENCES siteuser (id),
    FOREIGN KEY (style_id) REFERENCES style (id),
    FOREIGN KEY (Province_id) REFERENCES Region (id),
    FOREIGN KEY (Category_id) REFERENCES Category (id)
);

CREATE SEQUENCE company_id_seq;
CREATE INDEX company_id_index ON company (id);

CREATE VIEW vcompany AS
      SELECT cp.id,cp.uid,su.username,cp.linkman,cp.cncname,cp.encname,cp.email,cp.telephone,cp.fax,
            vpv.province,cp.city,cp.address,cp.postalcode,cp.bank,cp.bankaccount,cg.category,
            to_char(cp.create_date,'YYYY-MM-DD HH:MI:SS') as date
            FROM siteuser su,company cp,vProvince vpv,category cg
            where su.id = cp.uid and cp.province_id = vpv.id and cp.category_id = cg.id
            ORDER BY cp.id;

-- ------------------------------------------------------
--    'link'
-- ------------------------------------------------------
DROP TABLE link;
DROP SEQUENCE       link_id_seq;
DROP INDEX          link_id_index;
DROP VIEW vlink;

CREATE TABLE link (
      id              integer DEFAULT nextval('link_id_seq') NOT NULL,
      uid             integer     DEFAULT '1' NOT NULL,    -- with foreign key
      linkname        varchar(30) DEFAULT '' NOT NULL,   -- link name
      url             varchar(100) DEFAULT '' NOT NULL,        -- link url
      image           varchar(100) DEFAULT '' NOT NULL,        -- image(jpg,png,gif....)
      create_date    timestamp DEFAULT now() ,
      modify_date    timestamp DEFAULT now() ,
      UNIQUE (id,uid),
      PRIMARY KEY (id),
      FOREIGN KEY (uid) REFERENCES siteuser (id)
);
CREATE SEQUENCE   link_id_seq;
CREATE INDEX        link_id_index ON company (id);
```

```
CREATE VIEW vlink AS
	SELECT		l.id,l.uid,su.username,l.linkname,l.url,l.image,to_char(l.create_date,'YYYY/MM/DD
HH:MI:SS') as date
		FROM siteuser su, link l
		WHERE su.id = l.uid
		ORDER BY l.id;
-- ORDER BY commit_log.commit_date	LIMIT 100;
-- ---------------------------------------------------
--	'product_sort'
-- ---------------------------------------------------
DROP TABLE product_sort;
DROP SEQUENCE	product_sort_id_seq;
DROP INDEX		product_sort_id_index;
DROP VIEW vproduct_sort;

CREATE TABLE product_sort (
	id			integer DEFAULT nextval('product_sort_id_seq') NOT NULL,
	uid			integer	DEFAULT '1' NOT NULL,	-- with foreign key
	plist		varchar(20) DEFAULT '' NOT NULL,	--
	create_date	timestamp DEFAULT now() ,
	modify_date	timestamp DEFAULT now() ,
	PRIMARY KEY (id),
	UNIQUE (id,uid),
	FOREIGN KEY (uid) REFERENCES siteuser (id)
);
CREATE SEQUENCE	product_sort_id_seq;
CREATE INDEX		product_sort_id_index ON style (id);

CREATE VIEW vproduct_sort AS
	SELECT	ps.id,ps.uid,su.username,ps.plist,to_char(ps.create_date,'YYYY-MM-DD	HH:MI:SS')	as
date
		FROM siteuser su, product_sort ps
		WHERE su.id = ps.uid
		ORDER BY ps.id;

CREATE OR REPLACE FUNCTION product_sort_id_del_func () RETURNS opaque AS '
--	DECLARE
	BEGIN
		Delete from product where pid = OLD.id;
	RETURN OLD;
	END;
' LANGUAGE 'plpgsql';

DROP TRIGGER product_sort_tri on product_sort;
```

```
CREATE TRIGGER product_sort_tri
        BEFORE Delete
        --AFTER Delete
        ON product_sort FOR EACH ROW
        EXECUTE PROCEDURE product_sort_id_del_func ();

-- ------------------------------------------------------
--    'product'
-- ------------------------------------------------------
DROP TABLE product;
DROP SEQUENCE       product_id_seq;
DROP INDEX          product_id_index;
DROP VIEW vproduct;

CREATE TABLE product (
      id            integer DEFAULT nextval('product_id_seq') NOT NULL,
      pid           integer     DEFAULT '1' NOT NULL,    -- with foreign key
      pname         varchar(50) DEFAULT '' NOT NULL,   -- product name
      price       numeric(8,2) DEFAULT '0.00' NOT NULL,
      newprice    numeric(8,2) DEFAULT '0.00' NOT NULL,
      amount        integer     DEFAULT '0' NOT NULL,
      image         varchar(100) DEFAULT 'default_product.jpg' NOT NULL,    -- image(jpg,png,gif....)
      height        integer DEFAULT '320' NOT NULL,
      width         integer DEFAULT '320' NOT NULL,
      description   text ,
      isonline    boolean DEFAULT 'false' NOT NULL,
      create_date   timestamp DEFAULT now() ,
      modify_date   timestamp DEFAULT now() ,
      PRIMARY KEY (id),
      UNIQUE (id,pid),
      FOREIGN KEY (pid) REFERENCES product_sort (id)
);
CREATE SEQUENCE   product_id_seq;
CREATE INDEX      product_id_index ON style (id);

CREATE VIEW vproduct AS
      SELECT      p.id,ps.id      as      pid,ps.plist,p.pname,to_char(p.price,'999G999D99')      as
price,to_char(p.newprice,'999G999D99') as newprice,p.amount,p.image,p.height,p.width,p.description,
          CASE WHEN p.isonline=true THEN 'Y' ELSE 'N'     END      as isonline,
          to_char(p.create_date,'YYYY-MM-DD') as date
          FROM product_sort ps, product p
          WHERE ps.id = p.pid
          order by p.id;
```

```
-- ------------------------------------------------------
--    'clientinfo'
-- ------------------------------------------------------
DROP TABLE clientinfo;
DROP SEQUENCE       clientinfo_id_seq;
DROP INDEX          clientinfo_id_index;
DROP VIEW vclientinfo;

CREATE TABLE clientinfo (
      id              integer DEFAULT nextval('clientinfo_id_seq') NOT NULL,
      uid             integer     DEFAULT '1' NOT NULL,    -- with foreign key
      cname           varchar(20) DEFAULT '' NOT NULL,   -- link of friends
      telephone   varchar(20) DEFAULT '' NOT NULL,
      fax             varchar(20) DEFAULT '' NOT NULL,
      email           varchar(50) DEFAULT '' NOT NULL,
      address         varchar(50) DEFAULT '320' NOT NULL,
      postalcode      varchar(50) DEFAULT '320' NOT NULL,
      note        text            DEFAULT '' NOT NULL,
      create_date    timestamp DEFAULT now() ,
      modify_date    timestamp DEFAULT now() ,
      PRIMARY KEY (id),
      UNIQUE (id),
      FOREIGN KEY (uid) REFERENCES siteuser (id)
);
CREATE SEQUENCE   clientinfo_id_seq;
CREATE INDEX        clientinfo_id_index ON style (id);

CREATE VIEW vclientinfo AS
      SELECT                        ci.id,ci.uid,su.username,ci.cname,ci.telephone                  as
tel,ci.fax,ci.email,ci.address,ci.postalcode,ci.note,
      to_char(ci.create_date,'YYYY-MM-DD HH:MI:SS') as date
            FROM siteuser su, clientinfo ci
            WHERE su.id = ci.uid;

CREATE OR REPLACE FUNCTION clientinfo_tri_func () RETURNS opaque AS '
--       DECLARE
--            user_id CONSTANT INTEGER := OLD.id;
      BEGIN
            IF TG_OP = "DELETE" THEN
                  Delete from prodorder where clientinfo_id = OLD.id;
            END IF;
      RETURN OLD;
      END;
' LANGUAGE 'plpgsql';
```

```
DROP TRIGGER clientinfo_tri on clientinfo;
CREATE TRIGGER clientinfo_tri
        BEFORE Delete ON clientinfo FOR EACH ROW
        EXECUTE PROCEDURE clientinfo_tri_func ();
-- ------------------------------------------------------
--   'production order'
-- ------------------------------------------------------
DROP TABLE prodorder;
DROP SEQUENCE       prodorder_id_seq;
DROP INDEX          prodorder_id_index;

CREATE TABLE prodorder (
      id              integer DEFAULT nextval('prodorder_id_seq') NOT NULL,
      pid             integer    DEFAULT '1' NOT NULL,    -- foreign key
      amount          integer    DEFAULT '1' NOT NULL,
      clientinfo_id integer      DEFAULT '1' NOT NULL,    -- foreign key
      create_date   timestamp DEFAULT now() ,
      modify_date   timestamp DEFAULT now() ,
      PRIMARY KEY (id),
      UNIQUE (id),
      FOREIGN KEY (pid) REFERENCES product (id),
      FOREIGN KEY (clientinfo_id) REFERENCES clientinfo (id)
);
CREATE SEQUENCE   prodorder_id_seq;
CREATE INDEX        prodorder_id_index ON style (id);

-- ------------------------------------------------------
--   'drumbeating'
-- ------------------------------------------------------
DROP TABLE drumbeating;
DROP SEQUENCE       drumbeating_id_seq;
DROP INDEX          drumbeating_id_index;
DROP VIEW vdrumbeating;

CREATE TABLE drumbeating (
      id              integer DEFAULT nextval('drumbeating_id_seq') NOT NULL,
      uid             integer    DEFAULT '1' NOT NULL,    -- with foreign key
      logourl         varchar(100) DEFAULT '' NOT NULL,       -- image(jpg,png,gif....)
      bannerurl   varchar(100) DEFAULT '' NOT NULL,
      lheight         integer DEFAULT '60' NOT NULL,
      lwidth          integer DEFAULT '120' NOT NULL,
      bheight         integer DEFAULT '60' NOT NULL,
```

```
	bwidth		integer DEFAULT '468' NOT NULL,
	description	text ,
	create_date	timestamp DEFAULT now() ,
	modify_date	timestamp DEFAULT now() ,
	PRIMARY KEY (id),
	UNIQUE (id,uid),
	FOREIGN KEY (uid) REFERENCES siteuser (id)
);
CREATE SEQUENCE	drumbeating_id_seq;
CREATE INDEX		drumbeating_id_index ON style (id);

CREATE VIEW vdrumbeating AS
	SELECT dr.id,su.username,dr.logourl,dr.bannerurl
		FROM siteuser su, drumbeating dr
		WHERE su.id = dr.uid;


-- ----------------------------------------------------
--	'news'
-- ----------------------------------------------------
DROP TABLE news;
DROP SEQUENCE	news_id_seq;
DROP INDEX		news_id_index;
DROP VIEW vnews;

CREATE TABLE news (
	id		integer DEFAULT nextval('news_id_seq') NOT NULL,
	uid		integer	DEFAULT '1' NOT NULL,	-- with foreign key
	title		varchar(100) DEFAULT '' NOT NULL,	-- link of friends
	image		varchar(100) DEFAULT '' NOT NULL,
	content		text	DEFAULT '' NOT NULL,
	create_date	timestamp DEFAULT now() ,
	modify_date	timestamp DEFAULT now() ,
	PRIMARY KEY (id),
	UNIQUE (id),
	FOREIGN KEY (uid) REFERENCES siteuser (id)
);
CREATE SEQUENCE	news_id_seq;
CREATE INDEX		news_id_index ON news (id);

CREATE VIEW vnews AS
	SELECT ns.id,ns.uid,su.username,ns.title,ns.image,ns.content,to_char(ns.create_date,'YYYY/MM/DD
HH:MI:SS') as date
		FROM siteuser su, news ns
		WHERE su.id = ns.uid
```

```
        order by ns.id;

-- ------------------------------------------------------
--    'count'
-- ------------------------------------------------------
DROP TABLE count;
DROP SEQUENCE       count_id_seq;
DROP INDEX          count_id_index;
DROP VIEW vcount;

CREATE TABLE count (
    id                  integer DEFAULT nextval('count_id_seq') NOT NULL,
    uid                   integer     DEFAULT '1' NOT NULL,    -- with foreign key
    number              varchar(20) DEFAULT '' NOT NULL,
    fontcolor           varchar(10) DEFAULT 'FFFFFF' NOT NULL,
    backgroundcolor     varchar(10) DEFAULT '000000' NOT NULL,
    create_date         timestamp DEFAULT now() ,
    modify_date    timestamp DEFAULT now() ,
    PRIMARY KEY (id),
    UNIQUE (id,uid),
    FOREIGN KEY (uid) REFERENCES siteuser (id)
);

CREATE SEQUENCE   count_id_seq;
CREATE INDEX        count_id_index ON count (id);

CREATE VIEW vcount AS
    SELECT c.id,su.username,c.number,c.fontcolor,c.backgroundcolor
        FROM siteuser su, count c
        WHERE su.id = c.uid;

-- ------------------------------------------------------
--    'column_bar'
-- ------------------------------------------------------
DROP TABLE column_bar;
DROP SEQUENCE       column_bar_id_seq;
DROP INDEX          column_bar_id_index;
DROP VIEW vcolumn_bar;

CREATE TABLE column_bar (
    id              integer DEFAULT nextval('column_bar_id_seq') NOT NULL,
    uid             integer     DEFAULT '1' NOT NULL,    -- with foreign key
    colname         varchar(10) DEFAULT '' NOT NULL,   -- link of friends
    url             varchar(50) DEFAULT '' NOT NULL,
```

```
	title		varchar(50) DEFAULT '' NOT NULL,
	image		varchar(50) DEFAULT '' NOT NULL,
	height		varchar(50) DEFAULT '' NOT NULL,
	width		varchar(50) DEFAULT '' NOT NULL,
	content		text DEFAULT '' NOT NULL,
	create_date	timestamp DEFAULT now() ,
	modify_date	timestamp DEFAULT now() ,
	PRIMARY KEY (id),
	UNIQUE (id),
	FOREIGN KEY (uid) REFERENCES siteuser (id)
);
CREATE SEQUENCE	column_bar_id_seq;
CREATE INDEX	column_bar_id_index ON column_bar (id);

CREATE VIEW vcolumn_bar AS
	SELECT
cb.id,cb.uid,su.username,cb.colname,cb.url,cb.title,cb.image,cb.content,to_char(cb.create_date,'YYYY-M
M-DD HH:MI:SS') as date
		FROM siteuser su, column_bar cb
		WHERE su.id = cb.uid
		ORDER BY cb.id;

-- ----------------------------------------------------
--	'guestbook'
-- ----------------------------------------------------
DROP TABLE guestbook;
DROP SEQUENCE	guestbook_id_seq;
DROP INDEX	guestbook_id_index;
DROP VIEW vguestbook;

CREATE TABLE guestbook (
	id		integer DEFAULT nextval('guestbook_id_seq') NOT NULL,
	uid		integer	DEFAULT '1' NOT NULL,	-- with foreign key
	name		varchar(20) DEFAULT '' NOT NULL,	-- link of friends
	email		varchar(50) DEFAULT '' NOT NULL,
	telephone	varchar(20) DEFAULT '' NOT NULL,
	fax		varchar(20) ,
	homepage	varchar(50) ,
	title		varchar(50) DEFAULT '' NOT NULL,
	content		text ,
	style_id	integer	DEFAULT '1' NOT NULL,
	create_date	timestamp DEFAULT now() ,
	modify_date	timestamp DEFAULT now() ,
	PRIMARY KEY (id),
```

```
    UNIQUE (id),
    FOREIGN KEY (uid) REFERENCES siteuser (id),
    FOREIGN KEY (style_id) REFERENCES style (id)
);
CREATE SEQUENCE   guestbook_id_seq;
CREATE INDEX      guestbook_id_index ON guestbook (id);

CREATE VIEW vguestbook AS
    SELECT gb.id,gb.uid,su.username,gb.name,gb.email,gb.telephone as
tel,gb.fax,gb.homepage,gb.title,gb.content,to_char(gb.create_date,'YYYY/MM/DD HH:MI:SS') as date
        FROM siteuser su, guestbook gb
        WHERE su.id = gb.uid
        ORDER BY gb.id;

-- ----------------------------------------------------
--  'Category'
-- ----------------------------------------------------
DROP TABLE Category;
DROP SEQUENCE    Category_id_seq;
DROP INDEX       Category_id_index;
DROP VIEW vCategory;

CREATE TABLE Category (
    id              integer DEFAULT nextval('Category_id_seq') NOT NULL,
    Category        varchar(20) DEFAULT '' NOT NULL,
    description     text ,
    note        text ,
    remark          text ,
    create_date     timestamp DEFAULT now() ,
    modify_date     timestamp DEFAULT now() ,
    PRIMARY KEY (id),
    UNIQUE (id,Category)
);
CREATE SEQUENCE   Category_id_seq;
CREATE INDEX      Category_id_index ON Category (id);

CREATE VIEW vCategory AS
    SELECT          c.id,c.Category,c.description,c.note,c.remark,to_char(c.create_date,'YYYY-MM-DD
HH:MI:SS') as date
        FROM Category c
        ORDER BY c.id;

-- ----------------------------------------------------
--  'Region'
```

```
-- -----------------------------------------------------
DROP TABLE region;
DROP SEQUENCE      region_id_seq;
DROP INDEX         region_id_index;
DROP VIEW vregion;

CREATE TABLE region (
    id              integer DEFAULT nextval('region_id_seq') NOT NULL,
    region      varchar(20) DEFAULT '' NOT NULL,
    description    text ,
    note          text ,
    remark          text ,
    create_date    timestamp DEFAULT now() ,
    modify_date    timestamp DEFAULT now() ,
    PRIMARY KEY (id),
    UNIQUE (id,region)
);
CREATE SEQUENCE   region_id_seq;
CREATE INDEX         region_id_index ON region (id);

CREATE VIEW vregion AS
    SELECT       pv.id,pv.region,pv.description,pv.note,pv.remark,to_char(pv.create_date,'YYYY-MM-DD
HH:MI:SS') as date
        FROM region pv
        ORDER BY pv.id;

-- -----------------------------------------------------
--   'province'    VIEW
-- -----------------------------------------------------
DROP VIEW vprovince;
CREATE VIEW vprovince AS
    SELECT                                    pv.id,pv.region                              as
province,pv.description,pv.note,pv.remark,to_char(pv.create_date,'YYYY-MM-DD HH:MI:SS') as date
        FROM region pv
        ORDER BY pv.id;

-- -----------------------------------------------------
--   'Country'
-- -----------------------------------------------------
DROP TABLE country;
DROP SEQUENCE      country_id_seq;
DROP INDEX         country_id_index;
DROP VIEW vcountry;
```

```
CREATE TABLE country (
	id			integer DEFAULT nextval('country_id_seq') NOT NULL,
	country		varchar(20) DEFAULT '' NOT NULL,
	domain		varchar(2)	DEFAULT '' NOT NULL,
	Language	varchar(20)	DEFAULT '' NOT NULL,
	description	text ,
	note		text ,
	remark		text ,
	create_date	timestamp DEFAULT now() ,
	modify_date	timestamp DEFAULT now() ,
	PRIMARY KEY (id),
	UNIQUE (id,country)
);
CREATE SEQUENCE	country_id_seq;
CREATE INDEX		country_id_index ON country (id);

CREATE VIEW vcountry AS
	SELECT
pv.id,pv.country,pv.domain,pv.description,pv.note,pv.remark,to_char(pv.create_date,'YYYY-MM-DD
HH:MI:SS') as date
		FROM country pv
		ORDER BY pv.id;

begin;

insert into category(category) values('农业');
insert into category(category) values('交通运输');
insert into category(category) values('包装、印刷');
insert into category(category) values('食品');
insert into category(category) values('建筑与装饰');
insert into category(category) values('广告、策划');
insert into category(category) values('服装');
insert into category(category) values('工业设备');
insert into category(category) values('纺织');
insert into category(category) values('家居用品');
insert into category(category) values('信息咨询');
insert into category(category) values('电子');
insert into category(category) values('医药保健');
insert into category(category) values('商业代理');
insert into category(category) values('家用电器');
insert into category(category) values('艺术、工艺');
insert into category(category) values('服务业');
insert into category(category) values('电脑、软件');
insert into category(category) values('娱乐、休闲');
```

```
insert into category(category) values('经济技术合作');
insert into category(category) values('化工');
insert into category(category) values('摄影摄像');
insert into category(category) values('安全、保安');
insert into category(category) values('冶金矿产');
insert into category(category) values('体育用品');
insert into category(category) values('不动产');
insert into category(category) values('能源');
insert into category(category) values('办公、教育');
insert into category(category) values('库存积压');
insert into category(category) values('环保');
insert into category(category) values('媒体、传播');
insert into category(category) values('综合');
insert into category(category) values('域名注册');


insert into region(region) values('安徽');
insert into region(region) values('北京');
insert into region(region) values('重庆');
insert into region(region) values('福建');
insert into region(region) values('甘肃');
insert into region(region) values('广东');
insert into region(region) values('广西');
insert into region(region) values('贵州');
insert into region(region) values('海南');
insert into region(region) values('河北');
insert into region(region) values('河南');
insert into region(region) values('黑龙江');
insert into region(region) values('湖北');
insert into region(region) values('湖南');
insert into region(region) values('江苏');
insert into region(region) values('江西');
insert into region(region) values('吉林');
insert into region(region) values('辽宁');
insert into region(region) values('内蒙古');
insert into region(region) values('宁夏');
insert into region(region) values('青海');
insert into region(region) values('山东');
insert into region(region) values('山西');
insert into region(region) values('陕西');
insert into region(region) values('上海');
insert into region(region) values('四川');
insert into region(region) values('天津');
insert into region(region) values('西藏');
```

```
insert into region(region) values('新疆');
insert into region(region) values('云南');
insert into region(region) values('浙江');


--insert into country(domain,country) values('','');
insert into country(domain,country) values('AL','阿尔巴尼亚');
insert into country(domain,country) values('DZ','阿尔及利亚');
insert into country(domain,country) values('AF','阿富汗');
insert into country(domain,country) values('AR','阿根廷');
insert into country(domain,country) values('AE','阿拉伯联合酋长国');
insert into country(domain,country) values('AW','阿鲁巴');
insert into country(domain,country) values('OM','阿曼');
insert into country(domain,country) values('AZ','阿塞拜疆');
insert into country(domain,country) values('EG','埃及');
insert into country(domain,country) values('ET','埃塞俄比亚');
insert into country(domain,country) values('IE','爱尔兰');
insert into country(domain,country) values('EE','爱沙尼亚');
insert into country(domain,country) values('AD','安道尔');
insert into country(domain,country) values('AO','安哥拉');
insert into country(domain,country) values('AI','安圭拉岛');
insert into country(domain,country) values('AG','安提瓜和巴布达');
insert into country(domain,country) values('AT','奥地利');
insert into country(domain,country) values('AU','澳大利亚');
insert into country(domain,country) values('MO','澳门特别行政区');
insert into country(domain,country) values('BB','巴巴多斯');
insert into country(domain,country) values('PG','巴布亚新几内亚');
insert into country(domain,country) values('BS','巴哈马');
insert into country(domain,country) values('PK','巴基斯坦');
insert into country(domain,country) values('PY','巴拉圭');
insert into country(domain,country) values('BH','巴林');
insert into country(domain,country) values('PA','巴拿马');
insert into country(domain,country) values('BR','巴西');
insert into country(domain,country) values('BY','白俄罗斯');
insert into country(domain,country) values('BM','百慕大群岛');
insert into country(domain,country) values('BG','保加利亚');
insert into country(domain,country) values('MP','北马里亚纳群岛');
insert into country(domain,country) values('BJ','贝宁');
insert into country(domain,country) values('BE','比利时');
insert into country(domain,country) values('IS','冰岛');
insert into country(domain,country) values('PR','波多黎各');
insert into country(domain,country) values('PL','波兰');
insert into country(domain,country) values('BA','波斯尼亚和黑塞哥维那');
insert into country(domain,country) values('BO','玻利维亚');
```

```
insert into country(domain,country) values('BZ','伯利兹');
insert into country(domain,country) values('BW','博茨瓦纳');
insert into country(domain,country) values('BT','不丹');
insert into country(domain,country) values('IO','不列颠印度洋属土');
insert into country(domain,country) values('BF','布基纳法索');
insert into country(domain,country) values('BI','布隆迪');
insert into country(domain,country) values('BV','布韦岛');
insert into country(domain,country) values('KP','朝鲜');
insert into country(domain,country) values('GQ','赤道几内亚');
insert into country(domain,country) values('DK','丹麦');
insert into country(domain,country) values('DE','德国');
insert into country(domain,country) values('TP','东帝汶');
insert into country(domain,country) values('TG','多哥');
insert into country(domain,country) values('DM','多米尼克');
insert into country(domain,country) values('DO','多米尼克共和国');
insert into country(domain,country) values('RU','俄罗斯');
insert into country(domain,country) values('EC','厄瓜多尔');
insert into country(domain,country) values('ER','厄立特里亚');
insert into country(domain,country) values('FR','法国');
insert into country(domain,country) values('TF','法国南部和南极州');
insert into country(domain,country) values('FO','法罗群岛');
insert into country(domain,country) values('PF','法属波利尼西亚');
insert into country(domain,country) values('GF','法属圭亚那');
insert into country(domain,country) values('VA','梵蒂冈');
insert into country(domain,country) values('PH','菲律宾');
insert into country(domain,country) values('FJ','斐济群岛');
insert into country(domain,country) values('FI','芬兰');
insert into country(domain,country) values('CV','佛得角群岛');
insert into country(domain,country) values('FK','福克兰群岛（马尔维纳斯群岛）');
insert into country(domain,country) values('GM','冈比亚');
insert into country(domain,country) values('CG','刚果');
insert into country(domain,country) values('CD','刚果民主共和国');
insert into country(domain,country) values('CO','哥伦比亚');
insert into country(domain,country) values('CR','哥斯达黎加');
insert into country(domain,country) values('GD','格林纳达');
insert into country(domain,country) values('GL','格陵兰');
insert into country(domain,country) values('GE','格鲁吉亚');
insert into country(domain,country) values('CU','古巴');
insert into country(domain,country) values('GP','瓜德罗普岛（法属）');
insert into country(domain,country) values('GU','关岛');
insert into country(domain,country) values('GY','圭亚那');
insert into country(domain,country) values('KZ','哈萨克斯坦');
insert into country(domain,country) values('HT','海地');
insert into country(domain,country) values('KR','韩国');
```

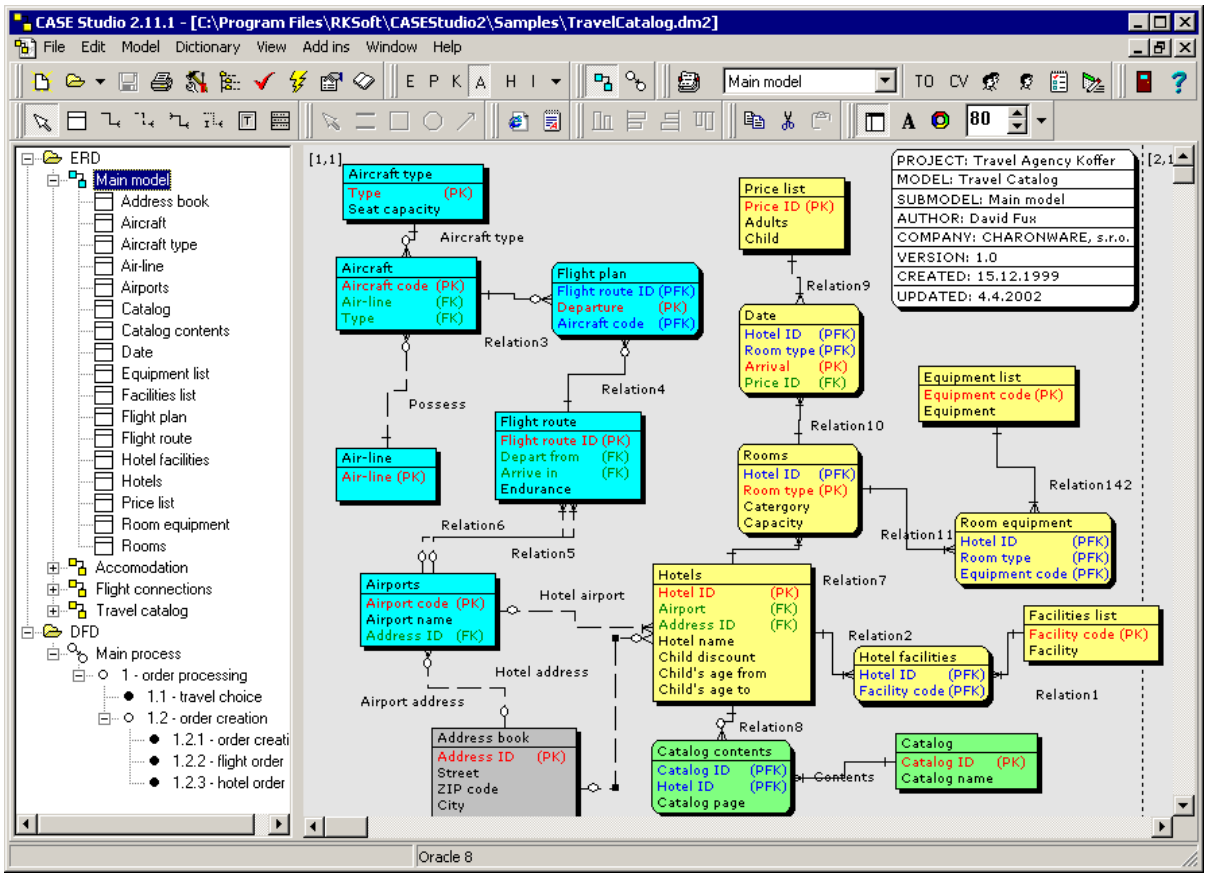
```
insert into country(domain,country) values('NL','荷兰');
insert into country(domain,country) values('AN','荷属安的列斯群岛');
insert into country(domain,country) values('HM','赫德和麦克唐纳群岛');
insert into country(domain,country) values('HN','洪都拉斯');
insert into country(domain,country) values('KI','基里巴斯');
insert into country(domain,country) values('DJ','吉布提');
insert into country(domain,country) values('KG','吉尔吉斯斯坦');
insert into country(domain,country) values('GN','几内亚');
insert into country(domain,country) values('GW','几内亚比绍');
insert into country(domain,country) values('CA','加拿大');
insert into country(domain,country) values('GH','加纳');
insert into country(domain,country) values('GA','加蓬');
insert into country(domain,country) values('KH','柬埔寨');
insert into country(domain,country) values('CZ','捷克共和国');
insert into country(domain,country) values('ZW','津巴布韦');
insert into country(domain,country) values('CM','喀麦隆');
insert into country(domain,country) values('QA','卡塔尔');
insert into country(domain,country) values('KY','开曼群岛');
insert into country(domain,country) values('CC','科科斯群岛');
insert into country(domain,country) values('KM','科摩罗');
insert into country(domain,country) values('CI','科特迪瓦');
insert into country(domain,country) values('KW','科威特');
insert into country(domain,country) values('HR','克罗地亚（赫尔瓦次卡）');
insert into country(domain,country) values('KE','肯尼亚');
insert into country(domain,country) values('CK','库克群岛');
insert into country(domain,country) values('LV','拉脱维亚');
insert into country(domain,country) values('LS','莱索托');
insert into country(domain,country) values('LA','老挝');
insert into country(domain,country) values('LB','黎巴嫩');
insert into country(domain,country) values('LT','立陶宛');
insert into country(domain,country) values('LR','利比里亚');
insert into country(domain,country) values('LY','利比亚');
insert into country(domain,country) values('LI','列支敦士登');
insert into country(domain,country) values('RE','留尼汪岛');
insert into country(domain,country) values('LU','卢森堡');
insert into country(domain,country) values('RW','卢旺达');
insert into country(domain,country) values('RO','罗马尼亚');
insert into country(domain,country) values('MG','马达加斯加岛');
insert into country(domain,country) values('MV','马尔代夫');
insert into country(domain,country) values('MT','马耳他');
insert into country(domain,country) values('MW','马拉维');
insert into country(domain,country) values('MY','马来西亚');
insert into country(domain,country) values('ML','马里');
insert into country(domain,country) values('MK','马其顿共和国');
```

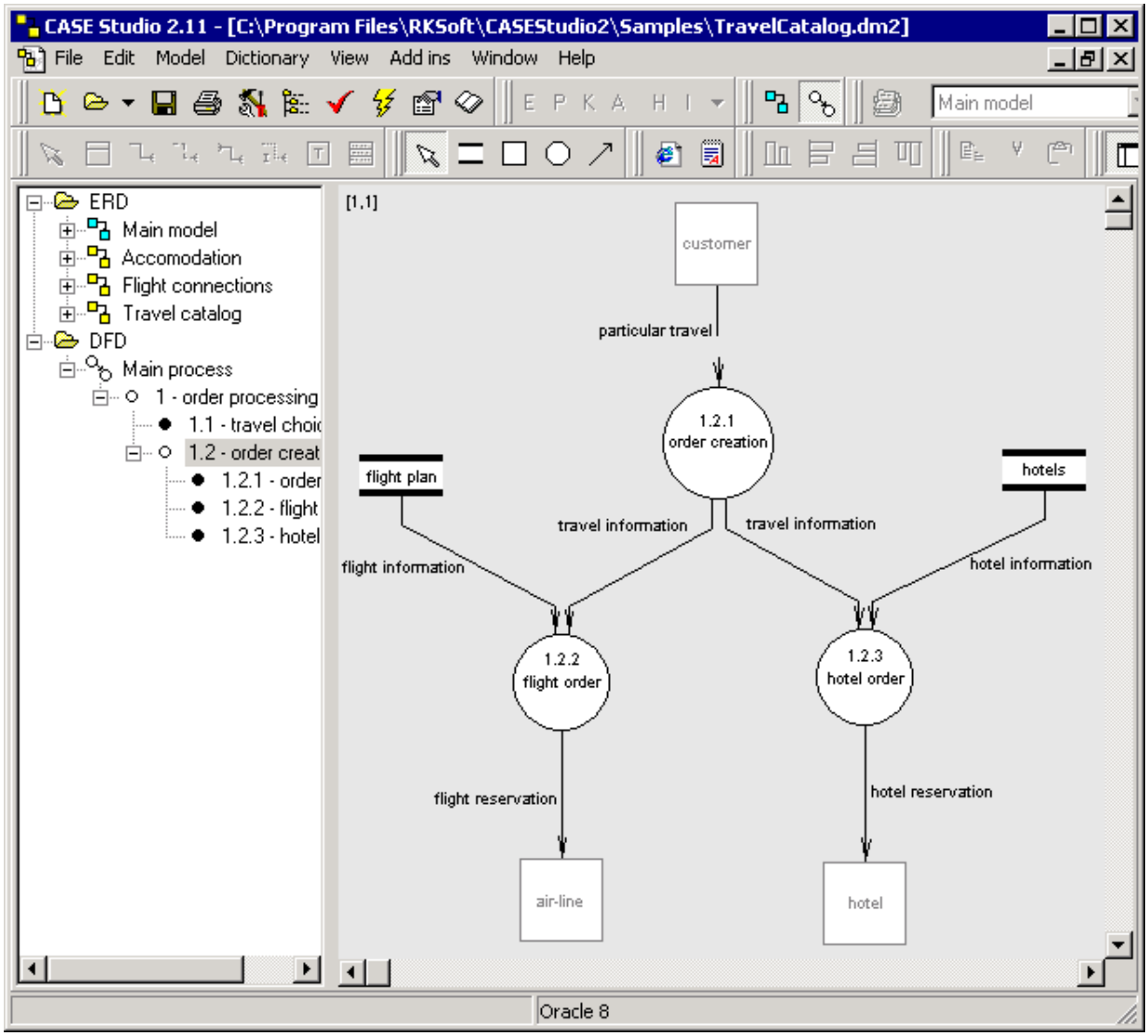
```
insert into country(domain,country) values('MH','马绍尔群岛');
insert into country(domain,country) values('MQ','马提尼克岛');
insert into country(domain,country) values('YT','马约特岛');
insert into country(domain,country) values('MU','毛里求斯');
insert into country(domain,country) values('MR','毛里塔尼亚');
insert into country(domain,country) values('US','美国');
insert into country(domain,country) values('AS','美属萨摩亚');
insert into country(domain,country) values('VI','美属维尔京群岛');
insert into country(domain,country) values('UM','美属小奥特兰群岛');
insert into country(domain,country) values('MN','蒙古');
insert into country(domain,country) values('MS','蒙特塞拉特(英)');
insert into country(domain,country) values('BD','孟加拉国');
insert into country(domain,country) values('PE','秘鲁');
insert into country(domain,country) values('FM','密克罗尼西亚');
insert into country(domain,country) values('MM','缅甸');
insert into country(domain,country) values('MD','摩尔多瓦');
insert into country(domain,country) values('MA','摩洛哥');
insert into country(domain,country) values('MC','摩纳哥');
insert into country(domain,country) values('MZ','莫桑比克');
insert into country(domain,country) values('MX','墨西哥');
insert into country(domain,country) values('NA','纳米比亚');
insert into country(domain,country) values('ZA','南非');
insert into country(domain,country) values('AQ','南极洲');
insert into country(domain,country) values('GS','南乔治亚和南桑德威奇群岛');
insert into country(domain,country) values('YU','南斯拉夫');
insert into country(domain,country) values('NR','瑙鲁');
insert into country(domain,country) values('NP','尼泊尔');
insert into country(domain,country) values('NI','尼加拉瓜');
insert into country(domain,country) values('NE','尼日尔');
insert into country(domain,country) values('NG','尼日利亚');
insert into country(domain,country) values('NU','纽埃');
insert into country(domain,country) values('NO','挪威');
insert into country(domain,country) values('NF','诺福克岛');
insert into country(domain,country) values('PW','帕劳');
insert into country(domain,country) values('PN','皮特克恩群岛');
insert into country(domain,country) values('PT','葡萄牙');
insert into country(domain,country) values('JP','日本');
insert into country(domain,country) values('SE','瑞典');
insert into country(domain,country) values('CH','瑞士');
insert into country(domain,country) values('SV','萨尔瓦多');
insert into country(domain,country) values('WS','萨摩亚');
insert into country(domain,country) values('SL','塞拉利昂');
insert into country(domain,country) values('SN','塞内加尔');
insert into country(domain,country) values('CY','塞浦路斯');
```

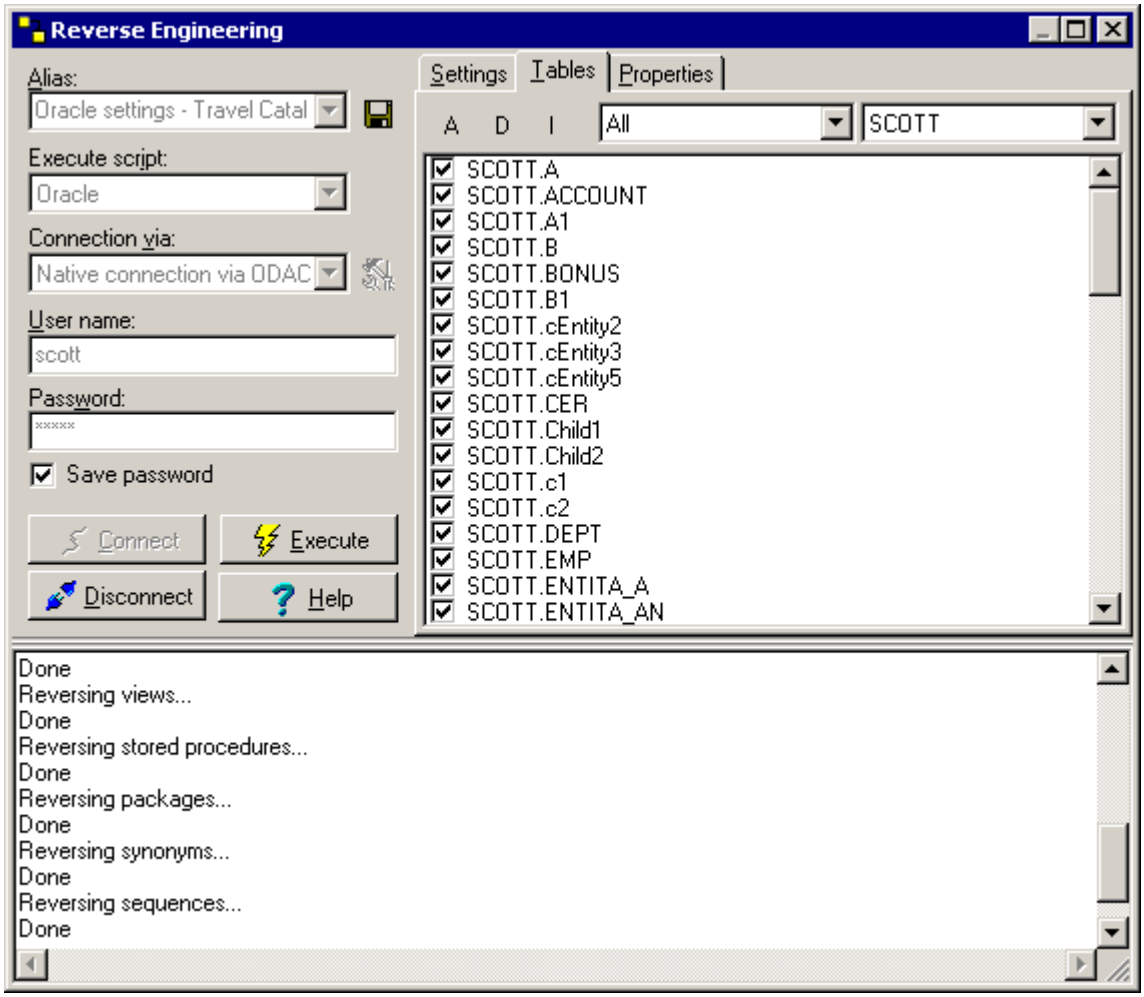
```
insert into country(domain,country) values('SC','塞舌尔群岛');
insert into country(domain,country) values('SA','沙特阿拉伯');
insert into country(domain,country) values('CX','圣诞岛');
insert into country(domain,country) values('ST','圣多美和普林西比');
insert into country(domain,country) values('SH','圣赫勒拿岛');
insert into country(domain,country) values('KN','圣基茨和尼维斯');
insert into country(domain,country) values('LC','圣卢西亚');
insert into country(domain,country) values('SM','圣马力诺');
insert into country(domain,country) values('PM','圣皮埃尔岛和密克隆岛');
insert into country(domain,country) values('VC','圣文森特和格林纳丁斯');
insert into country(domain,country) values('LK','斯里兰卡');
insert into country(domain,country) values('SK','斯洛伐克');
insert into country(domain,country) values('SI','斯洛文尼亚');
insert into country(domain,country) values('SJ','斯瓦尔巴群岛和扬马延');
insert into country(domain,country) values('SZ','斯威士兰');
insert into country(domain,country) values('SD','苏丹');
insert into country(domain,country) values('SR','苏里南');
insert into country(domain,country) values('SB','所罗门群岛');
insert into country(domain,country) values('SO','索马里');
insert into country(domain,country) values('TJ','塔吉克斯坦');
insert into country(domain,country) values('TH','泰国');
insert into country(domain,country) values('TZ','坦桑尼亚');
insert into country(domain,country) values('TO','汤加');
insert into country(domain,country) values('TC','特克斯群岛和凯科斯群岛');
insert into country(domain,country) values('TT','特立尼达和多巴哥');
insert into country(domain,country) values('TN','突尼斯');
insert into country(domain,country) values('TV','图瓦卢');
insert into country(domain,country) values('TR','土耳其');
insert into country(domain,country) values('TM','土库曼斯坦');
insert into country(domain,country) values('TK','托克劳');
insert into country(domain,country) values('WF','瓦利斯群岛和富图纳群岛');
insert into country(domain,country) values('VU','瓦努阿图');
insert into country(domain,country) values('GT','危地马拉');
insert into country(domain,country) values('VE','委内瑞拉');
insert into country(domain,country) values('BN','文莱');
insert into country(domain,country) values('UG','乌干达');
insert into country(domain,country) values('UA','乌克兰');
insert into country(domain,country) values('UY','乌拉圭');
insert into country(domain,country) values('UZ','乌兹别克斯坦');
insert into country(domain,country) values('ES','西班牙');
insert into country(domain,country) values('GR','希腊');
insert into country(domain,country) values('HK','香港特别行政区');
insert into country(domain,country) values('SG','新加坡');
insert into country(domain,country) values('NC','新喀里多尼亚');
```

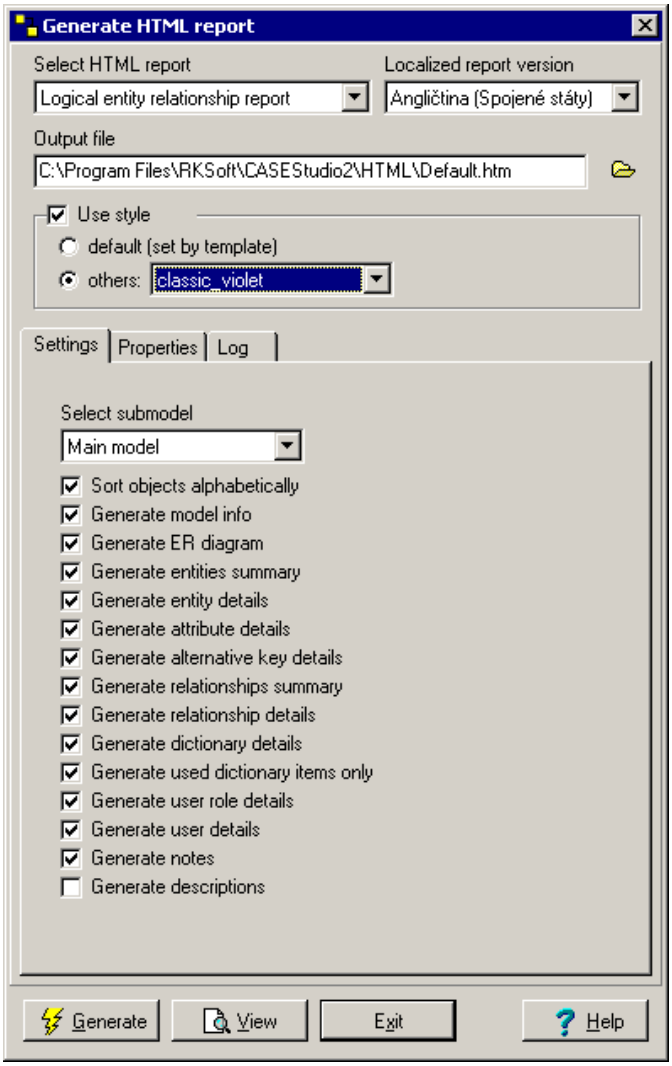
```
insert into country(domain,country) values('NZ','新西兰');
insert into country(domain,country) values('HU','匈牙利');
insert into country(domain,country) values('SY','叙利亚');
insert into country(domain,country) values('JM','牙买加');
insert into country(domain,country) values('AM','亚美尼亚');
insert into country(domain,country) values('YE','也门');
insert into country(domain,country) values('IQ','伊拉克');
insert into country(domain,country) values('IR','伊朗');
insert into country(domain,country) values('IL','以色列');
insert into country(domain,country) values('IT','意大利');
insert into country(domain,country) values('IN','印度');
insert into country(domain,country) values('ID','印度尼西亚');
insert into country(domain,country) values('UK','英国');
insert into country(domain,country) values('VG','英属维尔京群岛');
insert into country(domain,country) values('JO','约旦');
insert into country(domain,country) values('VN','越南');
insert into country(domain,country) values('ZM','赞比亚');
insert into country(domain,country) values('TD','乍得');
insert into country(domain,country) values('GI','直布罗陀');
insert into country(domain,country) values('CL','智利');
insert into country(domain,country) values('CF','中非共和国');
insert into country(domain,country) values('CN','中国');
insert into country(domain,country) values('TW','中国台湾');

commit;
ROLLBACK;
```

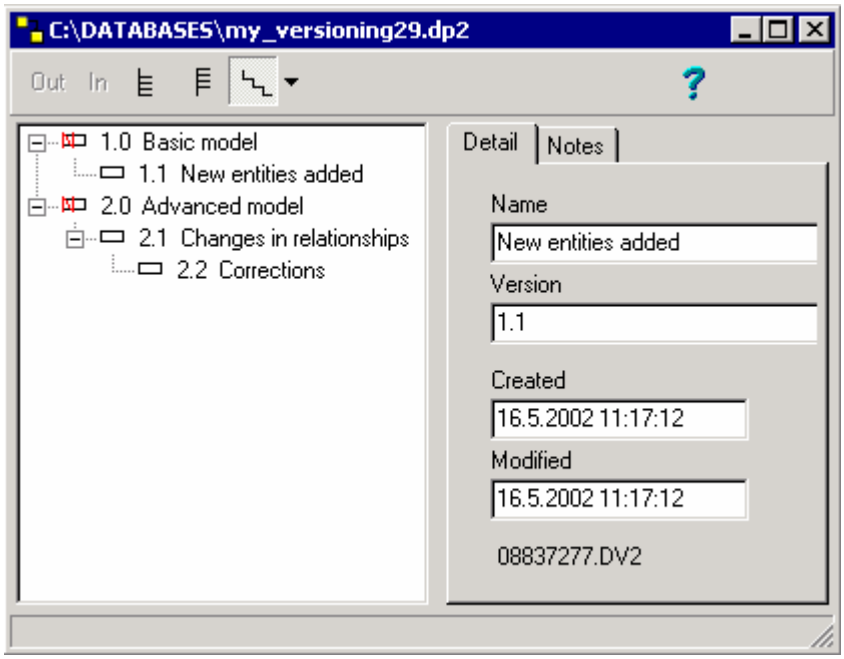
14.3 Case Studio 2

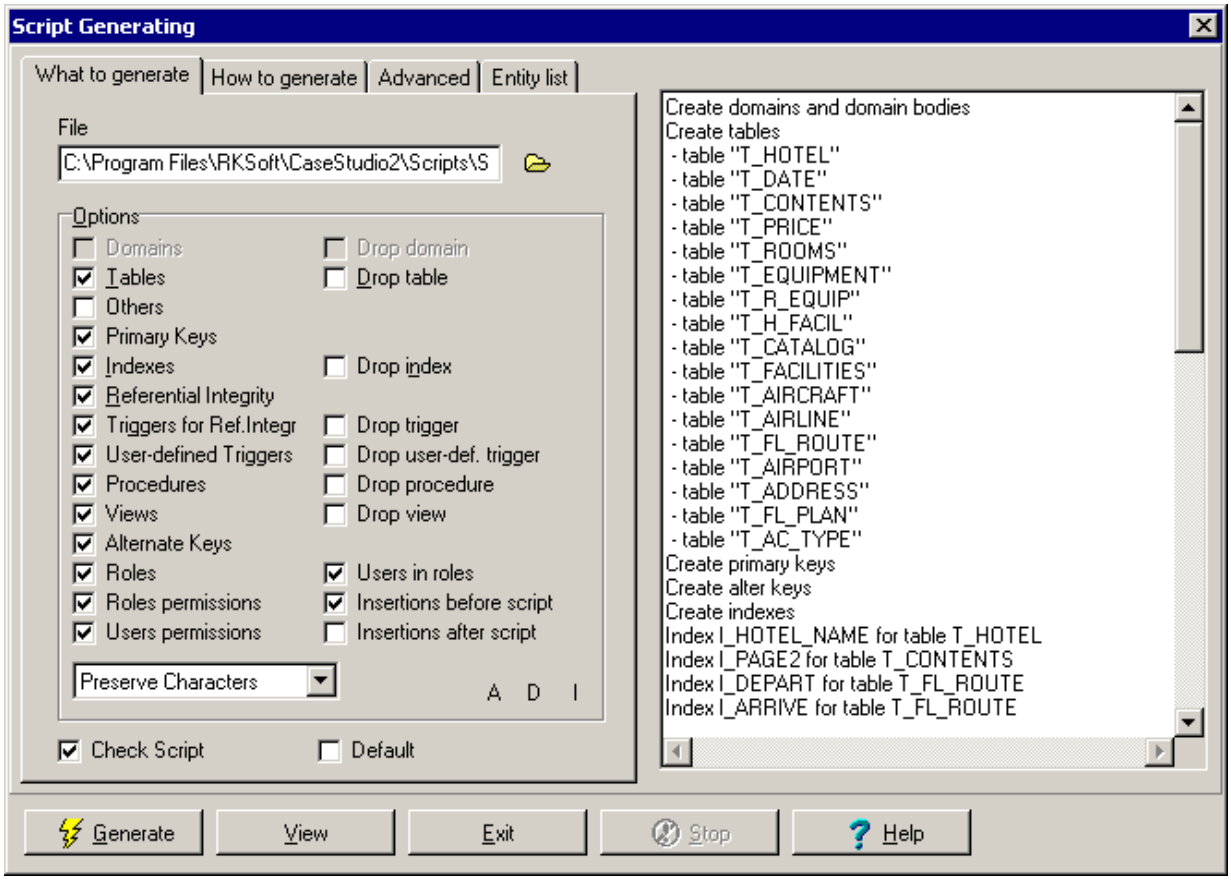



CASE Studio 2.11 - [C:\Program Files\RKSoft\CASEStudio2\Samples\TravelCatalog.dm2]
File Edit Model Dictionary View Add ins Window Help
Main model
ERD
Main model
Accomodation
Flight connections
Travel catalog
DFD
Main process
1 - order processing
1.1 - travel choi
1.2 - order creat
1.2.1 - order
1.2.2 - flight
1.2.3 - hotel
[1,1]
customer
particular travel
1.2.1
order creation
flight plan
hotels
travel information
travel information
flight information
hotel information
1.2.2
flight order
1.2.3
hotel order
flight reservation
hotel reservation
air-line
hotel
Oracle 8


Reverse Engineering
Alias:
Oracle settings - Travel Catal
Execute script:
Oracle
Connection via:
Native connection via ODAC
User name:
scott
Password:
*****
Save password
Connect
Execute
Disconnect
Help
Settings
Tables
Properties
A
D
I
All
SCOTT
SCOTT.A
SCOTT.ACCOUNT
SCOTT.A1
SCOTT.B
SCOTT.BONUS
SCOTT.B1
SCOTT.cEntity2
SCOTT.cEntity3
SCOTT.cEntity5
SCOTT.CER
SCOTT.Child1
SCOTT.Child2
SCOTT.c1
SCOTT.c2
SCOTT.DEPT
SCOTT.EMP
SCOTT.ENTITA_A
SCOTT.ENTITA_AN
Done
Reversing views...
Done
Reversing stored procedures...
Done
Reversing packages...
Done
Reversing synonyms...
Done
Reversing sequences...
Done


Generate HTML report
Select HTML report
Logical entity relationship report
Localized report version
Angličtina (Spojené státy)
Output file
C:\Program Files\RKSoft\CASEStudio2\HTML\Default.htm
Use style
default (set by template)
others:
classic_violet
Settings
Properties
Log
Select submodel
Main model
Sort objects alphabetically
Generate model info
Generate ER diagram
Generate entities summary
Generate entity details
Generate attribute details
Generate alternative key details
Generate relationships summary
Generate relationship details
Generate dictionary details
Generate used dictionary items only
Generate user role details
Generate user details
Generate notes
Generate descriptions
Generate
View
Exit
Help


C:\DATABASES\my_versioning29.dp2
Out
In
1.0 Basic model
1.1 New entities added
2.0 Advanced model
2.1 Changes in relationships
2.2 Corrections
Detail
Notes
Name
New entities added
Version
1.1
Created
16.5.2002 11:17:12
Modified
16.5.2002 11:17:12
08837277.DV2



CASE Studio 2 - Feature	LITE	FULL
Entity relationship diagrams		
Clipper (older version)	☑	☑
DBISAM 3	☑	☑
IBM DB2 UDB ver. 8.1	☑	☑
IBM DB2 UDB ver. 7.1	☑	☑
Informix (older version)	☑	☑
Informix 9	☑	☑
Ingres (older version)	☑	☑
InterBase 7	☑	☑
InterBase 6 SQL 3	☑	☑
InterBase 6 SQL 1	☑	☑
InterBase 5	☑	☑
InterBase 4	☑	☑
MS Access 2000	☑	☑
MS Access 97	☑	☑
MS SQL 2000	☑	☑
MS SQL 7	☑	☑
MS SQL 6.5	☑	☑
MySQL 4 (incl. innoDB)	☑	☑
MySQL 3.23	☑	☑
Oracle 9	☑	☑

Oracle 8	☑	☑
Oracle 7	☑	☑
Paradox (older version)	☑	☑
Pervasive	☑	☑
PostgreSQL 7.3 BETA	☑	☑
PostgreSQL 7	☑	☑
Sybase 12.5	☑	☑

Data flow diagram

Data flows	-	☑
Data stores	-	☑
Processes	-	☑
Terminators	-	☑

Reverse engineering

Clipper (older version)	-	-
DBISAM 3	-	☑
IBM DB2 UDB ver. 7	-	☑
IBM DB2 UDB ver. 8	-	☑
Informix (older version)	-	-
Informix 9	☑	-
Ingres (older version)	-	-
InterBase 7	-	☑
InterBase 6 SQL 3, InterBase 6 SQL 1	-	☑
InterBase 5	-	☑
InterBase 4	-	☑
MS Access 2000	-	☑
MS Access 97	-	☑
MS SQL 2000	-	☑
MS SQL 7	-	☑
MS SQL 6.5	-	☑
MySQL 4 (incl. innoDB)	-	☑
MySQL 3.23	-	☑
Oracle 9	-	☑
Oracle 8	-	☑
Oracle 7	-	☑
Paradox (older version)	-	☑
Pervasive	☑	-
PostgreSQL 7.3 BETA	-	☑
PostgreSQL 7	-	☑
Sybase 12.5	-	☑

Reports & export possibilities

RTF reports	☑	☑

HTML reports	☑	☑
Export into JPG, BMP or PNG	☑	☑
Export into XML	☑	☑

Features

After script items	☑	☑
Alternate keys	☑	☑
Alter scripts	-	-
Autolayout	-	-
Before script items	☑	☑
Cardinality	☑	☑
Data dictionary (domains)	☑	☑
Descriptions	☑	☑
Drops generation	☑	☑
Foreign keys definition	☑	☑
Functions	☑	☑
Indexes	☑	☑
Informative relationships	☑	☑
Internal clipboard	☑	☑
M:N relationships	☑	☑
Network features	-	-
New user defined data types	☑	☑
Non-identifying relationships	☑	☑
Notes	☑	☑
Object type bodies*	☑	☑
Object types*	☑	☑
Package bodies*	☑	☑
Packages*	☑	☑
Patterns for views, procedures and triggers	☑	☑
Primary keys definition	☑	☑
Referential integrity	☑	☑
Rough uppercase/lowercase selection	☑	☑
Self relationships	☑	☑
Sequences*	☑	☑
SQL script generation	☑	☑
Storages	☑	☑
Stored procedures	☑	☑
Support of compound foreign keys	☑	☑
Synonyms*	☑	☑
Triggers	☑	☑
User permissions (in models)	-	☑
Users (in models)	-	☑
Users group (in models)	-	☑
Views	☑	☑

Model maintenance		
Database/Model conversion**	☑	☑
Gallery	☑	☑
Model check	☑	☑
Submodels	☑	☑
To-Do list	☑	☑
Version comparsion	-	☑
Version manager	-	☑
Scripting and user defined templates		
OLE Automation objects	-	☑
Large COM interface	-	☑
MS Scripting	☑	☑
Jscript support	☑	☑
VB script support	☑	☑
Templates editor	-	☑
Templates export	-	☑
Templates import	-	☑
User defined variables editor	-	☑
Displaying		
Visual creation of DFD	-	☑
Visual creation of ERD	☑	☑
Alignment in columns	☑	☑
Attributes syntax highlighting	☑	☑
Entity background color selection	☑	☑
Font selection	☑	☑
Logical model	-	-
Physical model	☑	☑
Logical view / physical view	☑	☑
Model background color selection	☑	☑
Shadows	☑	☑
Stamp	☑	☑
Straight relationship lines	☑	☑
Text descriptions	☑	☑
Variable display level	☑	☑
Support		
Free email support	☑	☑
Free updates of CASE Studio 2	☑	☑

14.4 安装脚本

14.4.1 setenv.sh

```
#=========================================================
# Author:Netkiller
# Script: setenv.sh
#=========================================================

SRCHOME=../src
BINARYHOME=../binary
RPMSHOME=../rpms
CONFHOME=../conf
# Install Portable Threads
PThreads=pth-2.0.0.tar.gz

# Shared Memory Allocation
SMemory=mm-1.3.0.tar.gz

APACHE_HOME=/usr/local/apache
APACHE=apache_1.3.27.tar.gz
MOD_SSL=mod_ssl-2.8.14-1.3.27.tar.gz
MOD_PERL=mod_perl-1.0-current.tar.tar
MOD_FASTCGI=mod_fastcgi-2.4.0.tar.gz
PHP=php-4.3.2.tar.gz
```

14.4.2 install.sh

```
#=========================================================
# Author:Netkiller
# Script:install.sh
#=========================================================
if [ -f setenv.sh ]; then
    . setenv.sh
fi

#rpms_location="$base_dir/"

POSTGRESQL=${RPMSHOME}
Install_PostgreSQL(){
      cd $POSTGRESQL
```

```
        rpm -Uvh --nodeps postgresql-libs-?.?.?-1PGDG.i386.rpm
        rpm -Uvh --nodeps postgresql-?.?.?-1PGDG.i386.rpm
        rpm -Uvh --nodeps postgresql-devel-?.?.?-1PGDG.i386.rpm
        rpm -Uvh --nodeps postgresql-server-?.?.?-1PGDG.i386.rpm
        rpm -Uvh --nodeps postgresql-contrib-?.?.?-1PGDG.i386.rpm
        rpm -Uvh --nodeps postgresql-docs-?.?.?-1PGDG.i386.rpm
        rpm -Uvh --nodeps postgresql-jdbc-?.?.?-1PGDG.i386.rpm
        rpm -Uvh --nodeps postgresql-pl-?.?.?-1PGDG.i386.rpm
        rpm -Uvh --nodeps postgresql-python-?.?.?-1PGDG.i386.rpm
        rpm -Uvh --nodeps postgresql-tcl-?.?.?-1PGDG.i386.rpm
        rpm -Uvh --nodeps postgresql-test-?.?.?-1PGDG.i386.rpm
        cd ..
        rpm -qa|grep post
}

Uninstall_PostgreSQL(){
        rpm -e --nodeps `rpm -qa |grep postgresql`
}

Status_PostgreSQL(){
        rpm -qa|grep postgresql
}

Install_Apache(){
        cd ${SRCHOME}
        tar zxf apache_?.?.??.tar.gz >/dev/null
        cd    apache_?.?.??
        ./configure --prefix=/usr/local/apache --enable-module=so
        make
        make install
        cd ..
        rm -rf apache_?.?.??
        echo "/usr/local/apache/bin/apachectl start" >> /etc/rc.d/rc.local
}

Install_PHP(){
        cd ${SRCHOME}
        tar zxf php-?.?.?.tar.gz
        cd php-?.?.?
        ./configure --prefix=/usr/local/php --with-apxs=/usr/local/apache/bin/apxs \
                --with-config-file-path=/usr/local/etc --enable-track-vars --with-xml \
                --with-pgsql --with-mysql \
        --with-ldap --enable-ftp --with-openssl --with-iconv --with-kerberos \
        --with-java=/usr/local/java
```

```
        make
        make install
        cp php.ini-dist /usr/local/etc/php.ini
        #LoadModule php4_module modules/libphp4.so
        #AddType application/x-httpd-php .php .php4 .php3 .phtml
}

Install_MySQL(){
        if ! rpm -qa|grep -q "mysql" ; then
        tar zxvf mysql-standard-4.0.12-pc-linux-i686.tar.gz    >/dev/null
        mv mysql-standard-4.0.12-pc-linux-i686 /usr/local/mysql
        cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld
        chkconfig mysqld reset
        chown mysql.mysql -R /usr/local/mysql
        else
                echo "mysql package is already installed"
        fi
#       echo "su mysql -c"cd /usr/local/mysql/ ;./configure">/dev/null">> /etc/rc.d/rc.local
}

Install_MySQL(){
cd ${RPMSHOME}
rpm -Uvh MySQL-server-4.0.13-0.i386.rpm
rpm -Uvh MySQL-client-4.0.13-0.i386.rpm
rpm -Uvh MySQL-devel-4.0.13-0.i386.rpm
rpm -Uvh MySQL-shared-4.0.13-0.i386.rpm
rpm -Uvh MySQL-shared-compat-4.0.13-0.i386.rpm

}


Install_Sun_J2sdk(){
        cd ${BINHOME}
        chmod 700 j2sdk-1_4_1_02-linux-i586.bin
        ./j2sdk-1_4_1_02-linux-i586.bin
        mv j2sdk1.4.1_02 /usr/local/java
        cp ../conf/profile.sh /etc/profile.d/java.sh
        chmod 755 /etc/profile.d/java.sh
}

Install_Tomcat(){
        cd ${BINARYHOME}
        tar zxvf tomcat-4.1.24.tar.gz >/dev/null
        mv jakarta-tomcat-4.1.24 /usr/local/jakarta-tomcat
```

```
#       echo "/usr/local/jakarta-tomcat/bin/startup.sh" >> /etc/rc.d/rc.local
}

Install_Ant(){
        cd ${BINARYHOME}
        tar zxvf apache-ant-1.5.3-1-bin.tar.gz >/dev/null
        mv apache-ant-1.5.3-1 /usr/local/apache-ant
}

Install_Zhcon(){

        tar zxvf zhcon-?.?.?.tar.gz
        ./configure
        make
        make install
}

usage()
{
        echo "Usage: $0 {install|uninstall|status|reinstall|config|} [base_dir]"
        echo "Usage: $0 {apache|php|mysql} [base_dir]"
        echo "Usage: $0 {j2sdk|tomcat|ant|jdbc|postgresql} [base_dir]"
#    echo "base_dir               the location of install disk, optional"
}

# --- main process starts ---

case "$1" in
   install)
                banner
                install
                ;;
   uninstall)
                banner
#                uninstall
#Uninstall_PostgreSQL
                ;;
   status)
                banner
                status
                ;;
   reinstall)
                banner
                uninstall
```

```
        install
        ;;
    config)
        banner
        config
        ;;
    apache)
      Install_Apache
      ;;
    php)
      Install_PHP
      ;;
    mysql)
        Install_MySQL
        ;;
    j2sdk)
      Install_Sun_J2sdk
      ;;
    tomcat)
      Install_Tomcat
      ;;
    ant)
      Install_Ant
      ;;
    postgresql)
      Install_PostgreSQL
      ;;
    *)
        usage
        exit 1
esac

# --- main process ends ---
```

14.5 附件

文档中用到的一些相关的文件可以去主页上下载

http://www.kdeopen.com

http://linux.9812.net/

14.6 PostgreSQL 成功案例与解决方案

http://www.isc.org/products/OpenReg/

OpenReg is an implementation of a domain registry, such as might be used by top-level domain operators to manage the delegation of domains in a "shared registry" environment. OpenReg:

- supports the Extensible Provisioning Protocol (EPP), the IETF standards-track protocol for interaction between registries and registrars;
- is designed and debugged as a distributed multi-process system;
- supports PostgreSQL and is designed to accommodate to very large registries;
- publishes zone files to be served using BIND;
- gathers comprehensive profiling and load statistics;
- is published as free software, under a BSD-style licence.

15 参考资料

http://www.postgresql.org
http://www.pgsqldb.org/pgsqldoc-cvs/index.html

16 关于

作者信息：

陈景峰，昵称：netkiller, UNIX like 爱好者,研究方向园区网、群集系统、网络安全、TB 级数据库、LDAP、J2EE，Corba，企业整体解决方案。

主页地址：
http://www.kdeopen.com
http://linux.9812.net/
我的简历：http://home.9812.net/linux/resume/resume.htm

OICQ:13721218
ICQ:101888222
Yahoo:snetkiller
AIM:xnetkiller
MSN:snetkiller@passpost.com
网易泡泡：openunix@163.com
E-Mail: netkiller@9812.net

有问题最好给我发 Email 或去下面的 BBS 里讨论`

http://www.pgsqldb.org

http://www.chinaunix.com

http://www.linuxforum.net

国外网站：

http://www.linuxfocus.org/

http://www.opendocspublishing.com/

http://www.casestudio.com/ 一个数据设计工具

http://www.datanamic.com/一个数据设计工具

http://www.commandprompt.com/

17 版本、声明

文件状态	文件标识	《PostgreSQL 实用实例参考》
□草稿	当前版本	2.0
☑正式发布	作　　者	Netkiller(陈景峰)
☑正在修改	开始日期	2003 年 10 月 7 日星期二 2003-12-5

版本历史

版本/状态	作者	参与者	起止日期	备注
1.0.0/草稿	陈景峰	符乃晴 尚丽娜	2003-10-15	校对
1.1.0/修改	陈景峰		2003-10-17	PostgreSQL RPM 包安装 附件中安装脚本 Jsp/Java 汉字编码问题
1.1.1	陈景峰		2003-10-22	查询, 多个字段组合约束
1.1.2	陈景峰		2003-10-23	substring()函数截取部分汉字 PHP 字符编码问题
1.1.3	陈景峰		2003-10-24	共享内存 最大连接
1.2.3	陈景峰		2003-10-25	SSL,SSH
1.2.3	陈景峰	朱陪江	2003-10-29	校对
1.3.3	陈景峰		2003-11-1	序列 约束
1.4.3	陈景峰		2003-11-11	RPM 安装
1.4.4	陈景峰		2003-11-12	例子-分类目录

1.5.0	陈景峰		2003-11-25	函数，过程实例
1.6.0	陈景峰		2003-11-28	日期时间 “::”转换数据 性能提升-硬件
1.6.1	陈景峰		2003-11-29	用户，组，认证
1.6.2	陈景峰		2003-12-1	DW 的 JSP 开发环境 Java 汉字编码问题 web.xml 方案
正式/修正 2.0.0	陈景峰		2003-12-5	Jbuilder+weblogic+postgresql
2.1.0	陈景峰		2003/12/12	取出字符如果超过 20 个在后尾加“…” 保证备份数据的安全-PGP/GPG 加密
	陈景峰		2003-12-27	数据库性能测试