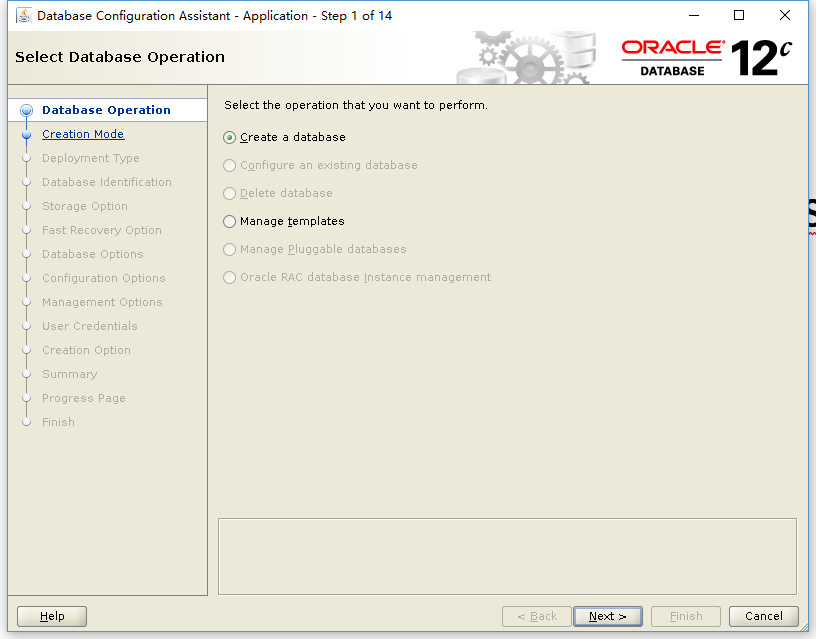
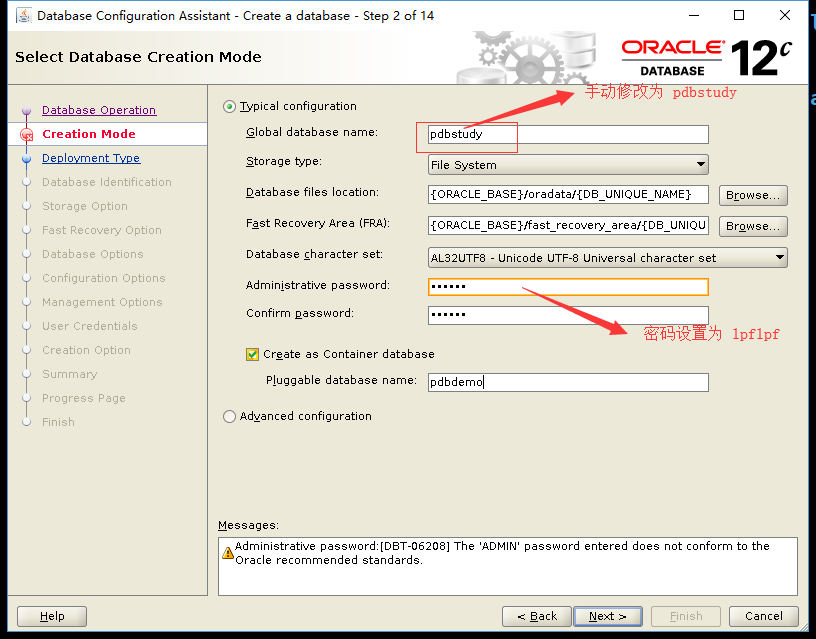
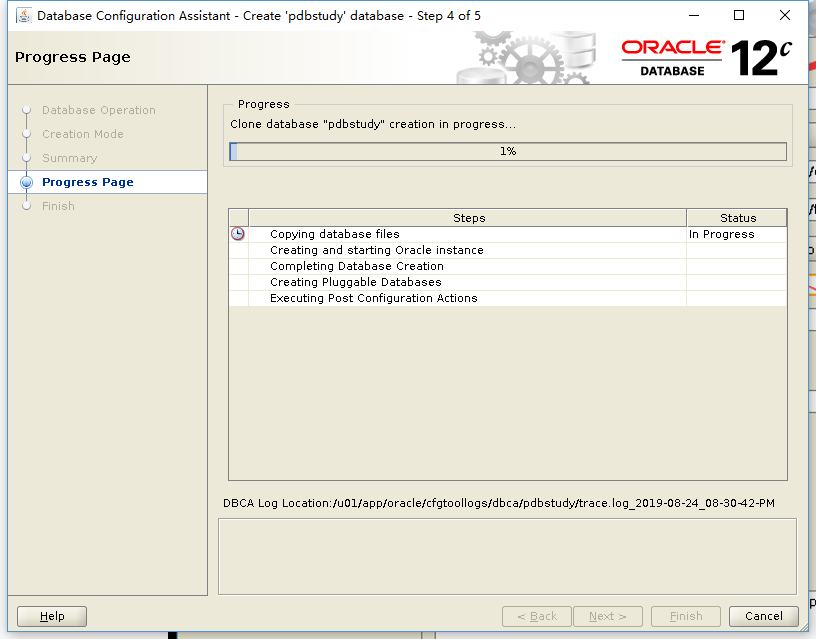
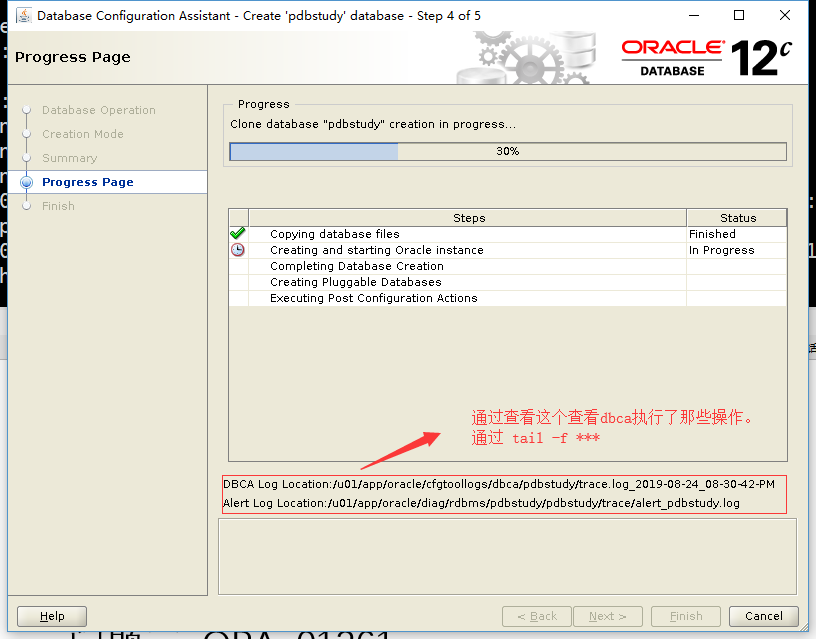
# 使用dbca 创建一个 叫做 pdbstudy 的数据库

1. 修改/home/oracle目录下的.bash\_profile 文件设置 ORACLE\_SID 的取值为 pdbstudy。
2. 执行source /home/oracle/.bash\_profile
3. $ echo $ORACLE\_SID 查看取值是不是 pdbstudy
4. 执行dbca命令。
   1. 



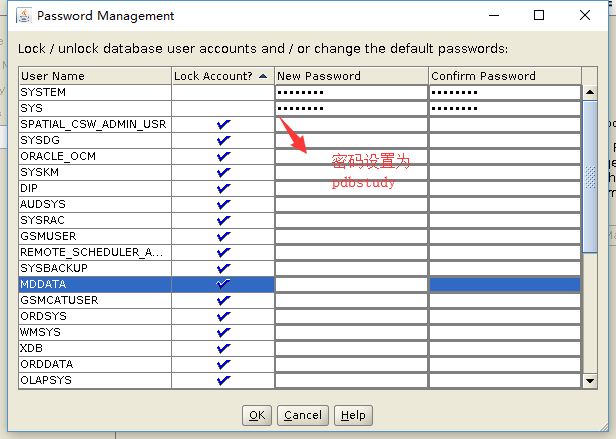
1. 设置密码和pdb的名称





Dbca 执行过程中日志保存在 /u01/app/oracle/diag/rdbms/pdbstudy/pdbstudy/trace/alert\_pdbstudy.log 文件中 。





创建好之后，使用sqlplus 登录

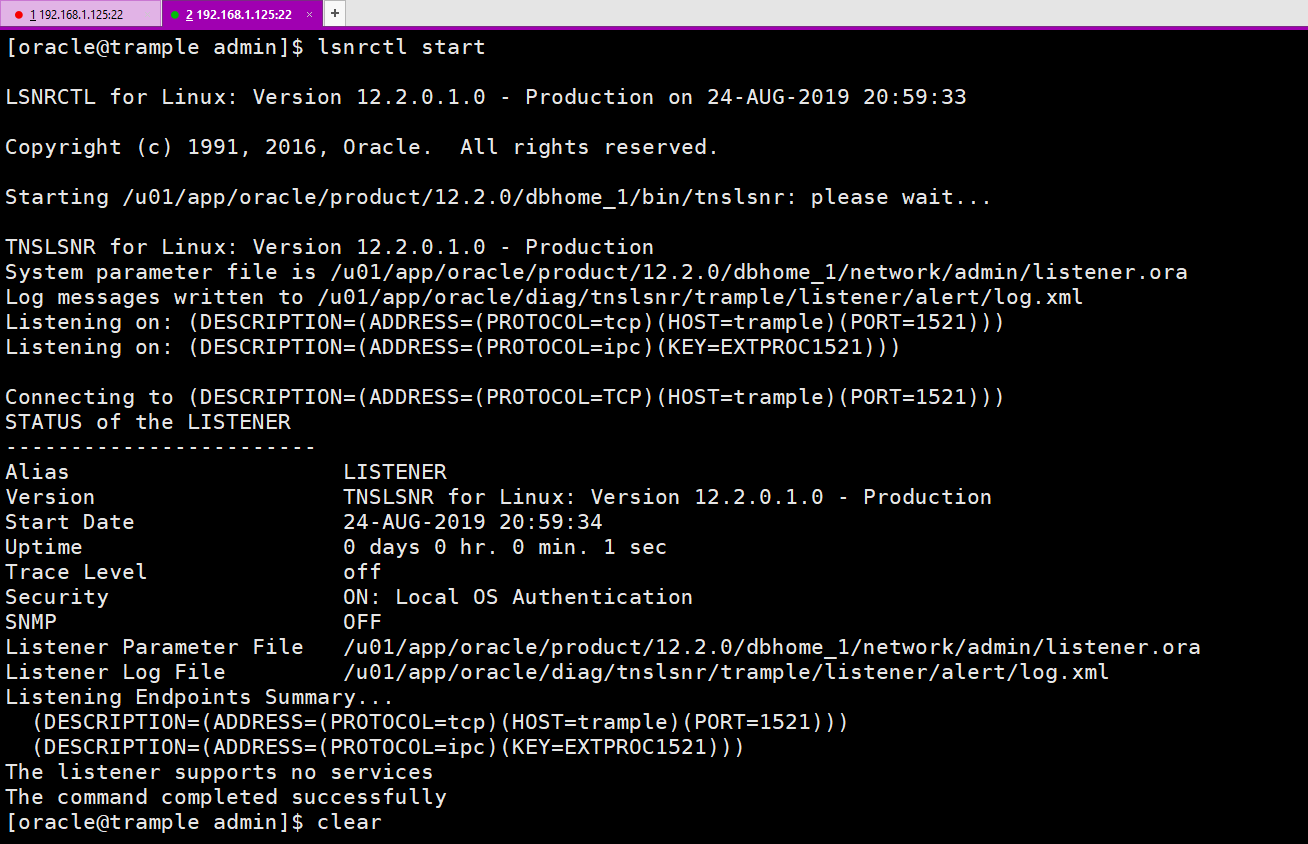
$ > sqlplus /nolog

SQL > conn / as sysdba

SQL> startup

退出sqlplus，启动监听

$ > lsnrctl status

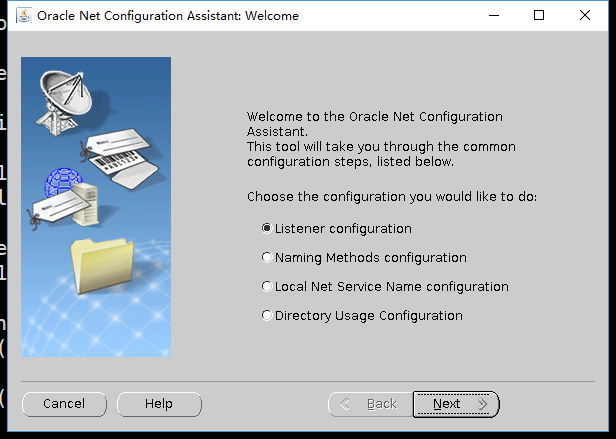


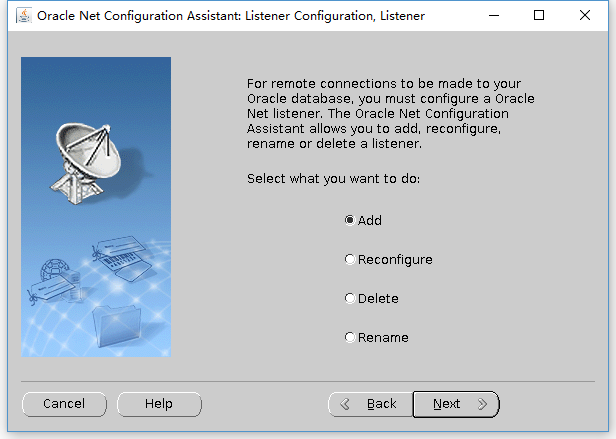
查看相关监听器的配置

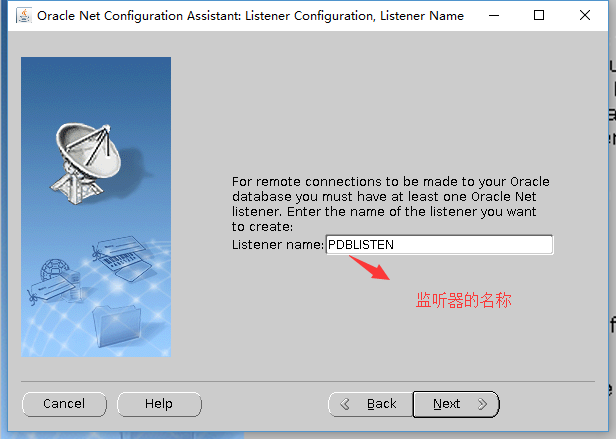
这是因为oracle数据库创建之后，没有创建对于的监听器。

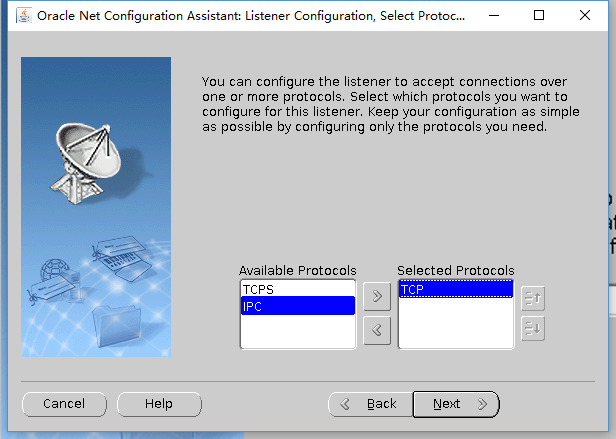
这里我们使用netca命令创建对应的监听器。

$ > netca

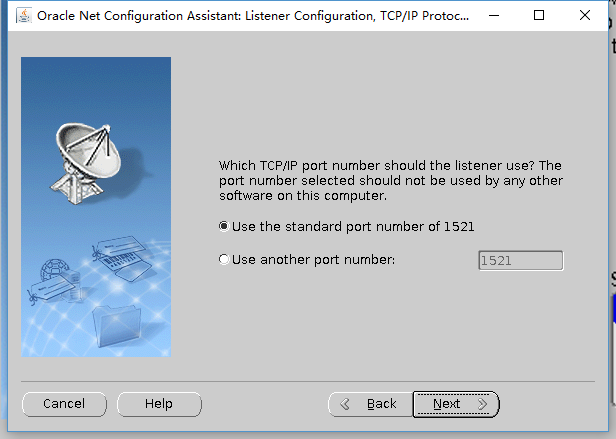




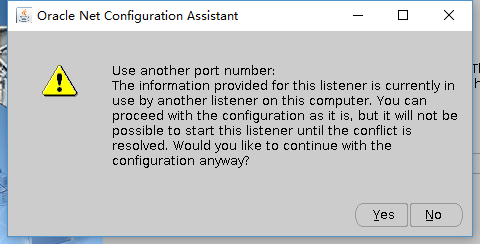




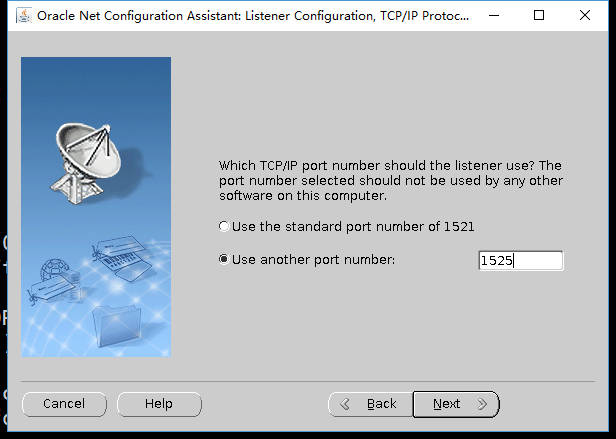
采用默认的网络协议 TCP 进行网络连接

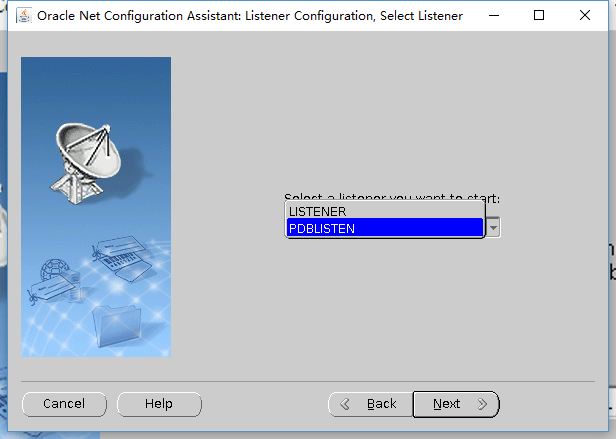


这是使用默认的端口 1521 端口

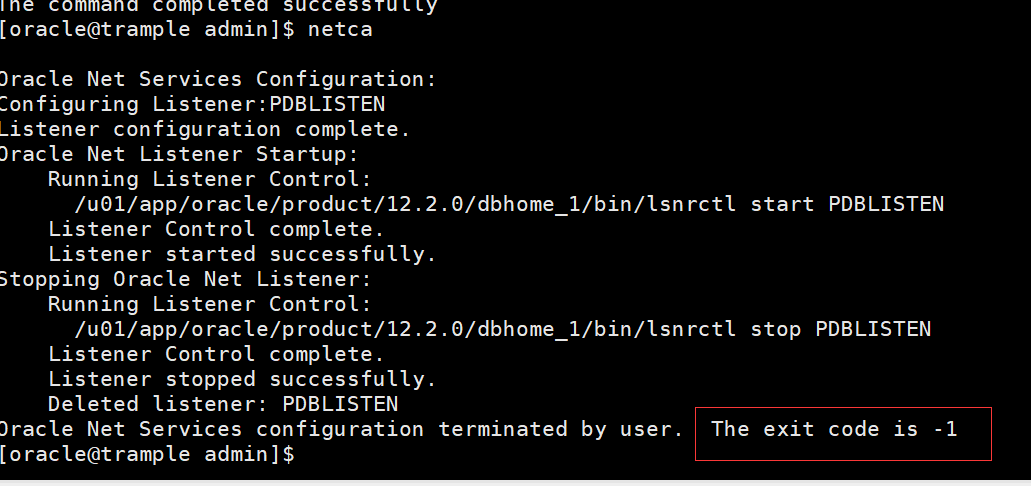


这里出现这个原因是因为我们已经存在其他监听器监听 1521 端口，我们只要 使用一个没有使用的端口就可以了。

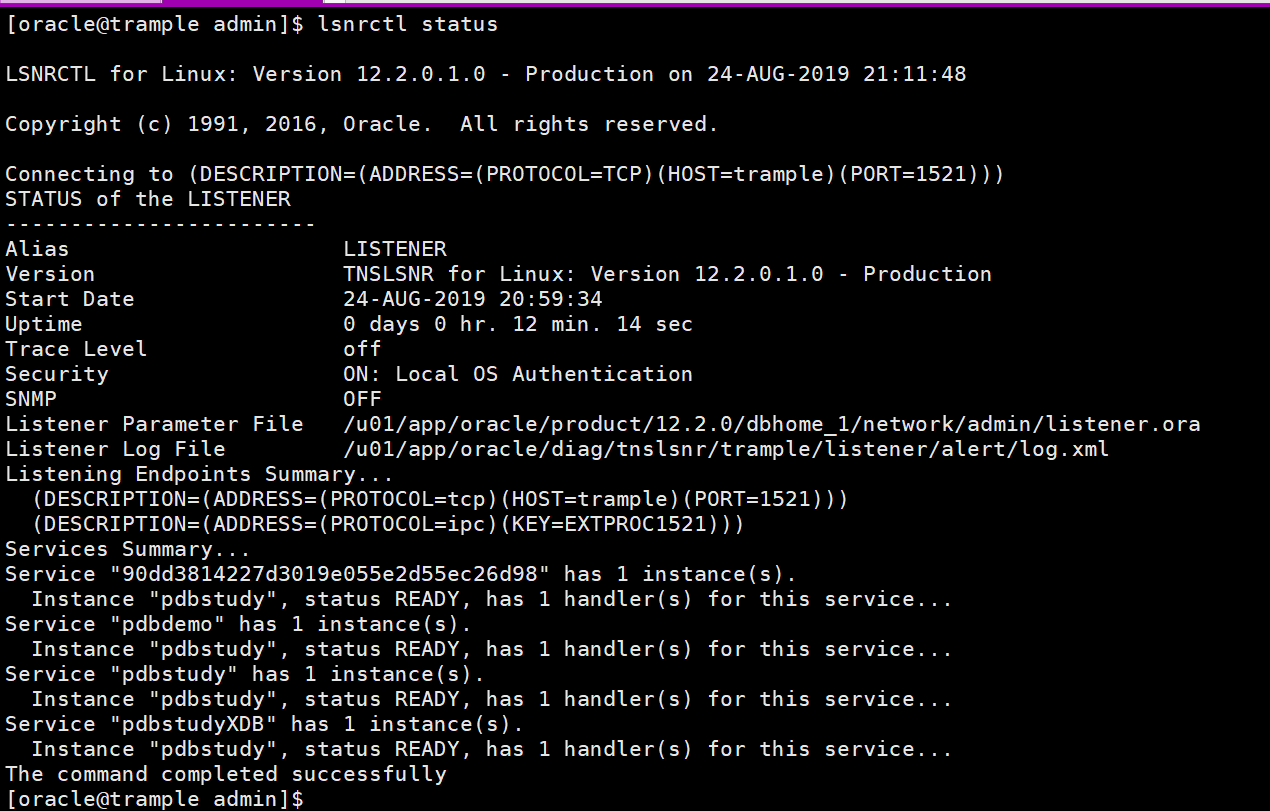




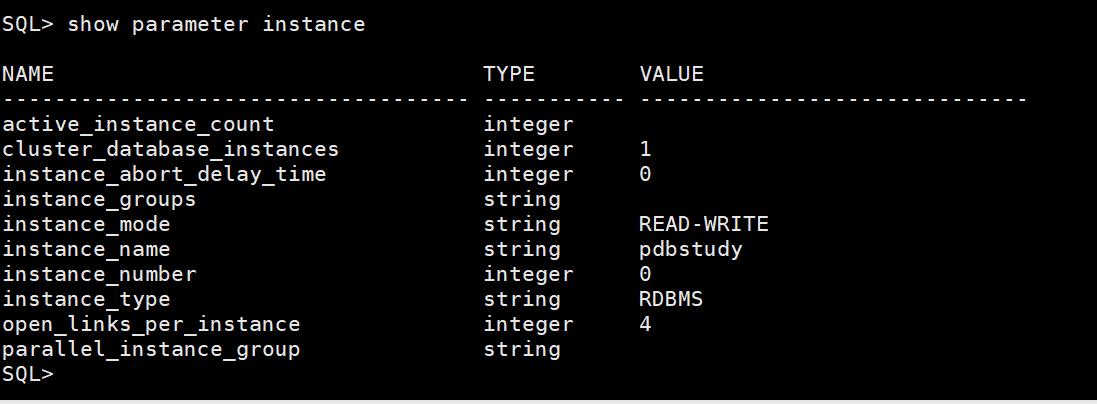
点击Next之后，关闭即可。



接下来，我们执行 lsnrctl status



数据库创建后查询一些信息：

1. 查看数据库有哪些实例
   1. 方法一:
   2. SQL > **select name from v$database;**
   3. 方法二：
   4. **select instance\_name from v$instance;**
   5. **方法三：**
   6. **SQL > show parameter instance;**
   7. 

启动之后，在远程sqlplus中通过

$ > sqlplus [C##lpf\_dev/lpf\_dev@192.168.1.125/pdbstudy](mailto:C##lpf_dev/lpf_dev@192.168.1.125/pdbstudy) 连接oracle数据库

这里C##lpf\_dev 是用户名， lpf\_dev 是密码， 192.168.1.125 是oracle数据库的ip地址。

Pdbstudy是使用dbca创建的oracle数据库。