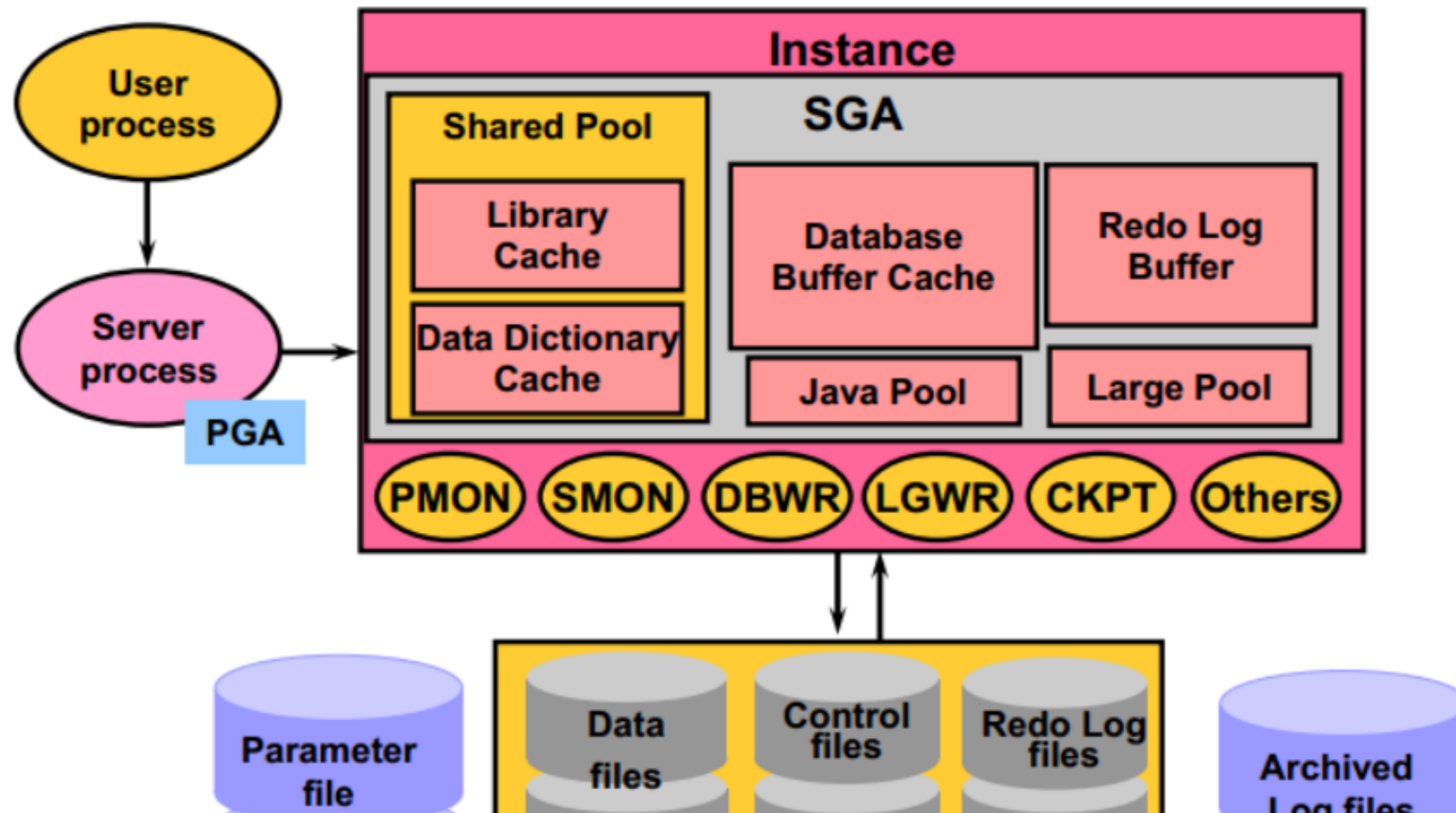


# 一. Oracle体系结构概述

## 1. Oracle组件架构

### Oracle组件架构

## Overview of Primary Components





3-38-image.png)

## 2. Oracle服务器

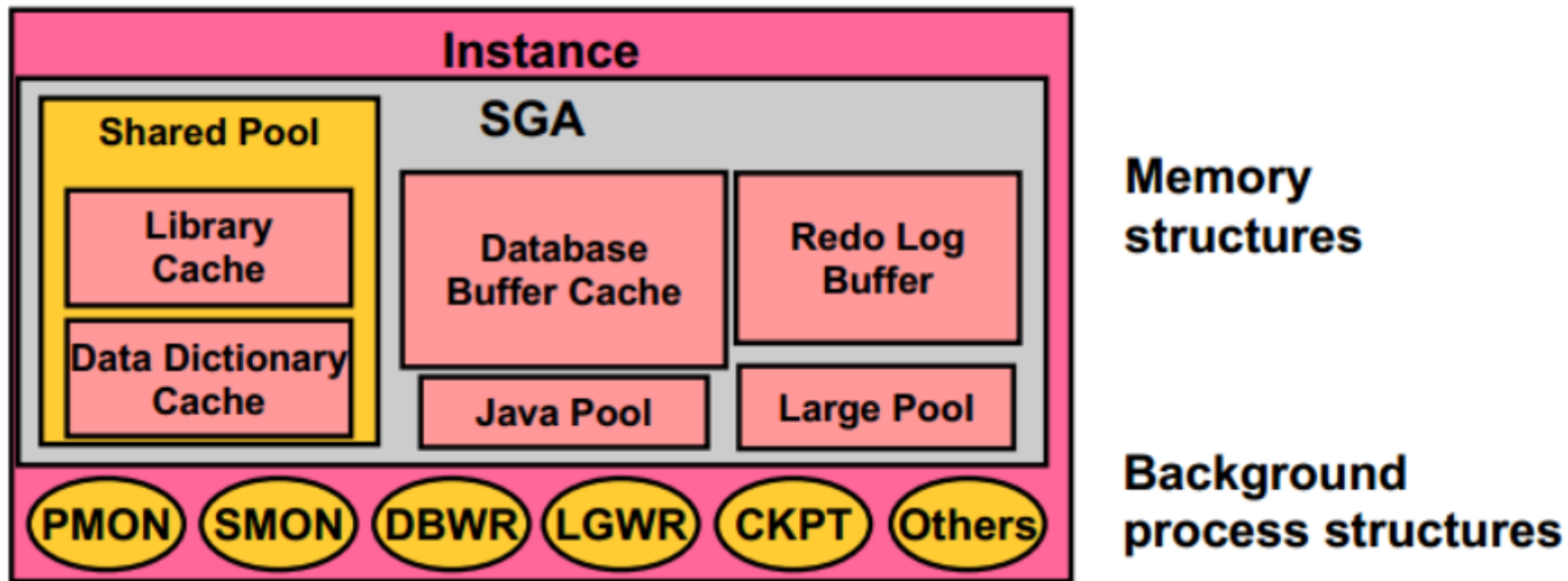
---

- Oracle服务器是一个数据库管理系统，它提供开放、全面、集成的方法进行信息管理
- 由Oracle实例和Oracle数据库组成

## 3. Oracle实例

---

# Oracle实例



## 实例

概述: 在数据库服务器上启动数据库后, Oracle软件会分配一个称为系统全局区(SGA)的共享内存区, 还会启动若干个Oracle后台进程. 两者的组合就称为一个Oracle实例.

- 访问数据库的一种方法
- 一个实例只能打开一个数据库
- 由内存和后台进程组成

在同一台计算机上可以并发执行多个实例,每个实例只访问它自己的物理数据库

RAC: 多个实例关联访问同一个数据库

## 内存

概述: 此处的内存主要有两个区域, SGA和PGA

- 系统全局区SGA : 在实例启动时分配, 是Oracle实例的基本组件
- 程序全局区PGA : 在服务器进程启动时分配

内存存储的信息有:

- 程序代码
- 连接会话的信息
- 程序执行过程需要的其他信息, 例如查询语句的结果集返回前的状态
- Oracle进程间的通信信息, 例如, 锁信息
- 缓存数据, 例如数据块和重做日志条目信息

参数

MEMORY\_TARGET

设定为0, 让SGA和PGA分别管理

## 进程

在操作系统上运行一系列步骤的一种机制

一个具有一定独立功能的程序

关于某个数据集的一次运行活动

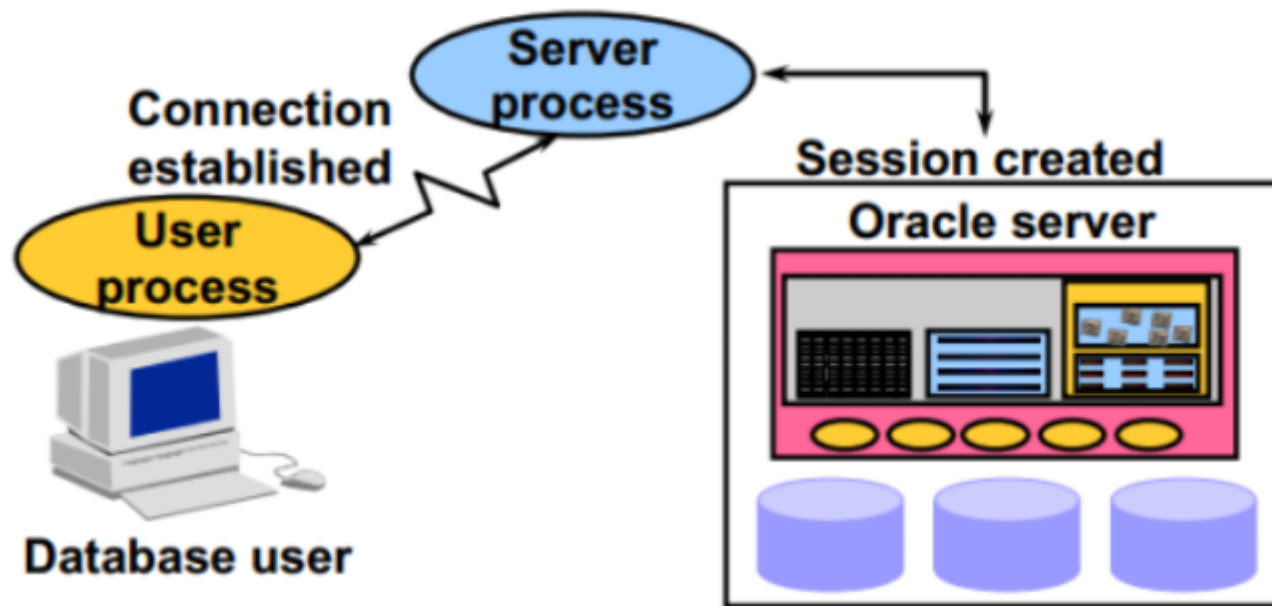
操作系统动态执行的基本单元

在传统的操作系统中, 进程既是基本的分配单元, 也是基本的执行单元

在Linux下, Oracle后台进程是Linux进程; 在NT中, Oracle后台进程是一个进程中执行线程

## 进程

### Oracle进程类型



用户进程 User process = 客户端进程 Client process

- 用户请求连接Oracle服务器时在客户端启动的进程
- 在数据库用户请求连接到Oracle服务器时启动, 请求与Oracle服务器交互的程序, 必须先建立连接, 不与Oracle服务器直接交互

#### 服务器进程 Server process

- 在用户连接上Oracle实例创建会话后, 在服务端启动的进程
- 直接与Oracle服务器交互的程序
- 执行生成的调用并返回相关结果
- 一个服务进程专门用来响应用户进程的请求
- 用户进程断开数据库, 与之相对应的服务器进程就终止了

#### 后台进程 Background process : **后台进程是实例和数据库的纽带**

- 在Oracle例程启动后创建的进程
- 直接与Oracle交互的程序
- 执行维护物理结构与内存结构之间的关系
- 必备的后台进程
  - DBWn
    - 数据库编写器
    - 进程从磁盘读取数据并将其写回到磁盘
    - 一个Oracle实例有许多数据库编写器, 如DBW0, DBW1, DBW2等
  - PMON

- ■ Oracle数据库中最活跃的一个进程
  - 调节所有其他进程的进程监视器
  - 能够清理异常连接的数据库连接, 并自动向侦听器进程注册数据库实例
- CKPT
  - 检查点过程
  - **在Oracle中, 磁盘上的数据称为块, 内存中的数据称为缓冲区。当该块写入缓冲区并更改时, 缓冲区变脏, 需要将其写入磁盘。CKPT进程使用检查点信息更新控制和数据文件头, 并向脏盘写入脏缓冲区的信号。**
- LGWR
  - 可恢复架构的关键
  - 在数据库中发生的每个变化都被写入到一个名为redo的日志文件中用于恢复; 而这些变化就是有LGWR进程编写和记录的
  - LGWR进程首先将更改写入内存, 然后将磁盘写入重做日志, 然后将其用于恢复
- SMON
  - 执行系统级清理操作的系统监视进程
  - 在发生故障的情况下自动恢复实例, 例如断电和清理临时文件
- 可选的后台进程
  - ARCn
    - 归档进程, 将重做日志的内容复制到归档重做日志文件
    - 可以有多个进程

- 允许写入多个目标

- LMDn、RECO、CJQ0、LMON、Snnn、Dnnn、Pnnn、LCKn、QMNn

附加

专有服务器模式下, 一个用户进程与服务器进程一一对应

共享服务器模式下, 多个用户进程可能对应一个服务器进程

## 4. Oracle数据库

---

Oracle数据库, 是操作系统上数据文件的集合, 数据文件是真正存放用户数据的地方

### 3个必须且重要的文件 + 3个非必须但重要的文件

- data files + control files + redo log files
  - 数据文件 : 包含数据库中实际数据的数据文件
  - 控制文件 : 包含维护和核实数据库完整性所必须的信息
  - 重做日志文件 : 记录数据库的更改, 以便在发生故障时能够恢复数据
- parameter files + password files + archived redo log files
  - 参数文件 : 定义了Oracle实例的特征. 例如它包含的参数大小决定了SGA中某些内存结构的大小
  - 密码文件 : 对有权启动和关闭Oracle实例的用户进行身份验证
  - 归档重做日志文件 : 重做日志文件的脱机副本

## 5. 连接与会话

---



## 连接 Connection

客户端进程与Oracle实例之间的一个物理通信通道. 物理通信通道可以通过进程间通信机制或网络协议确认.

## 会话 Session

会话是一个逻辑概念, 表示用户登录数据库的一种状态. 当用户通过密码登陆数据库时, 就建立了一个用户对应的会话. 用户断开数据库连接, 会话即结束.

对于一个用户来说, 可能同时存在多个会话, 因为oracle数据库允许用户同时多次并发连接, 例如sqlplus工具, 或应用程序中配置连接数据库

## 进程与会话的查询

- 查看会话的信息, 包括用户连接的程序、所在的客户端

```
select sid, serial#, paddr, username, program, machine
  from v$session
  where username='SYSTEM';
```

- 查看会话对应的Oracle服务进程

```
select username, program, terminal, spid
  from v$process
  where addr in ('000000008FF3E1C0', '000000008FF40240');
```

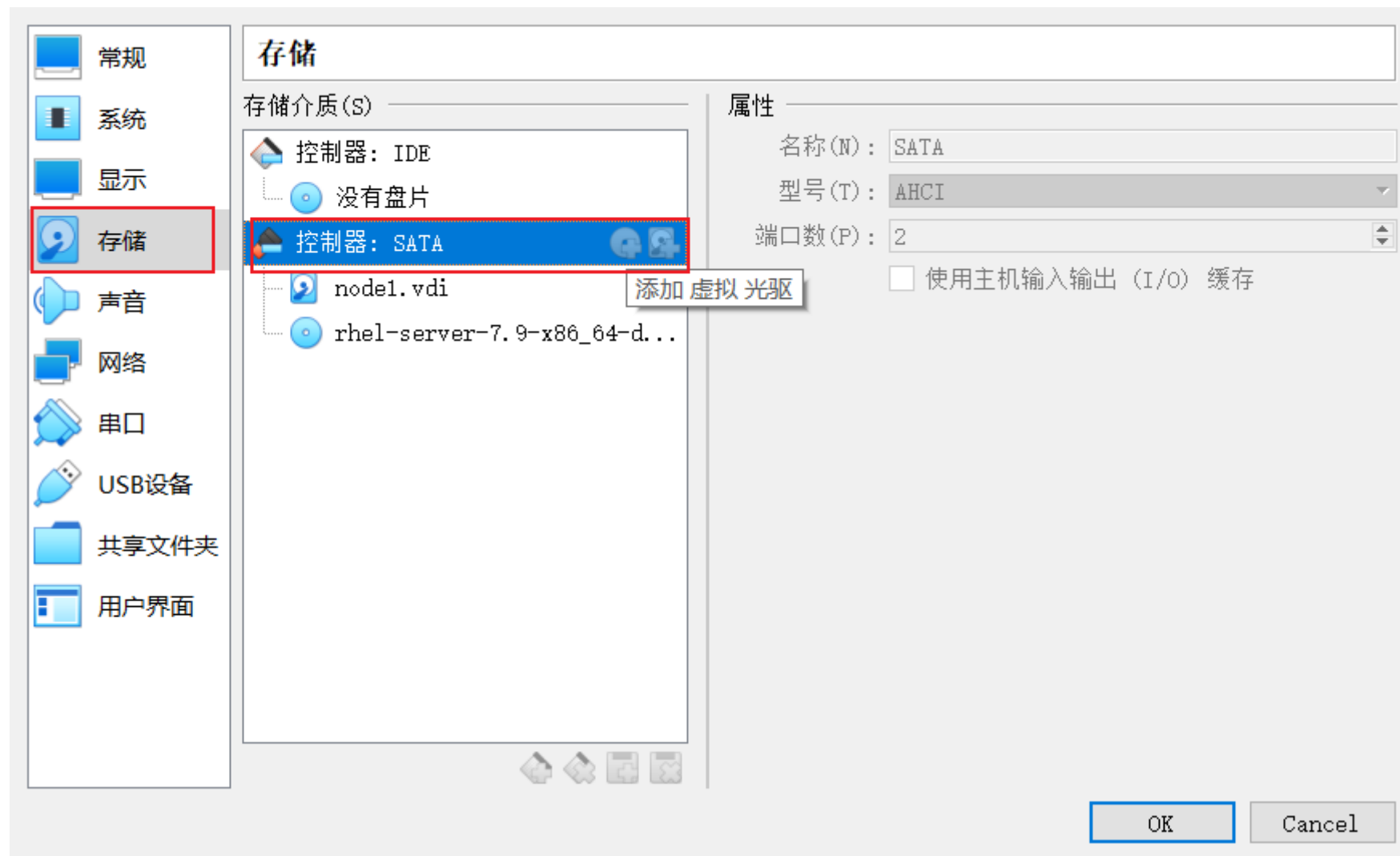
- 杀会话

```
alter system kill session 'xxx,xxx'
```

## 二. 配置本地yum源

### 1. 在虚拟机(服务器)中添加iso镜像

[VirtualBox中配置linux OS的本地磁盘镜像作为其软件源\\_kexianmiao的专栏-CSDN博客](#)



## 2. 建立挂载目录

```
mkdir /mnt/cdrom
```

## 3. 挂载镜像

```
mount /dev/cdrom /mnt/cdrom
```

## 4. 备份原本的yum源文件

```
vim /etc/yum.repos.d/redhat.repo

#[163]
#name=163
#baseurl=http://mirrors.163.com/centos/7/os/x86_64
#gpgcheck=0
#enabled=1
```

## 5. 建立本地源文件

```
vim /etc/yum.repos.d/local.repo

# 配置内容
[local]
name=local
baseurl=file:///mnt/cdrom
gpgcheck=1
enabled=1
gpgkey=file:///mnt/cdrom/RPM-GPG-KEY-redhat-release
```

## 6. 加载本地源

```
# 清除yum源
yum clean all

# 缓存本地源
yum makecache

# 测试
yum repolist all
```

```
[root@node1 yum.repos.d]# yum makecache
已加载插件: langpacks, product-id, search-disabled-repos, subscription-manager
'utf8' codec can't decode byte 0xd3 in position 12: invalid continuation byte
local | 2.8 kB 00:00:00
(1/6): local/group | 628 kB 00:00:00
(2/6): local/primary | 2.1 MB 00:00:00
(3/6): local/filelists | 3.1 MB 00:00:00
(4/6): local/group_xz | 95 kB 00:00:00
(5/6): local/other | 1.1 MB 00:00:00
(6/6): local/productid | 1.6 kB 00:00:00
local 5230/5230
local 5230/5230
local 5230/5230
元数据缓存已建立

[root@node1 yum.repos.d]# yum repolist all
已加载插件: langpacks, product-id, search-disabled-repos, subscription-manager
'utf8' codec can't decode byte 0xd3 in position 12: invalid continuation byte
源标识 源名称 状态
local local 启用: 5,230
repolist: 5,230

[root@node1 yum.repos.d]#
```

## 7. 开机自动挂载

```
vim /etc/fstab  
/dev/sr1 /mnt/cdrom iso9660 defaults 0 1
```

## 三. 安装软件及创建数据库

---

### 1. 安装软件的两种方式

---

- OUI方式
  - 通过Oracle Universal Installer选择Install database software only进行安装软件
- 静默安装
  1. 修改安装配置文件: vi ./database/response/db\_install.rsp
  2. 执行安装 ./runInstaller -silent -responseFile /oracle/db\_install.rsp
  3. 安装软件后, root执行以下脚本: 1) /oracle/oraInventory/orainstRoot.sh 2) /oracle/product/11.2.0/db\_1/root.sh

### 2. 创建数据库的四种方式

---

#### 1. OUI方式

安装Oracle Server的过程中, Oracle Universal Installer 会启动DBCA, 接着DBCA将自动创建一个初始数据库。

#### 2. DBCA

Database Configuration Assistant是一个用来简化数据库创建操作的图形用户界面

### 3. 静默创建数据库

1. 以oracle用户登录

2. 设置环境变量

```
export ORACLE_BASE=/oracle
export ORACLE_HOME=$ORACLE_BASE/product/11.2.0/db_1
export ORACLE_SID=mynewdb
export ORACLE_HOSTNAME=node1
export ORACLE_UNQNAME=mynewdb
export PATH=$ORACLE_HOME/bin:$ORACLE_HOME/jdk/bin:$PATH
```

3. 修改静默建库使用的文件

```
cp $ORACLE_HOME/assistants/dbca/dbca.rsp /oracle/dbca.rsp
# 修改 /oracle/dbca.rsp

# 生成建库使用的dbca.rsp文件，使用如下配置即可
cat > /oracle/dbca.rsp <<!
[GENERAL]
RESPONSEFILE_VERSION = "11.2.0"
OPERATION_TYPE = "createDatabase"

[CREATEDATABASE]
GDBNAME = "mynewdb"
SID = "mynewdb"
TEMPLATENAME = "General_Purpose.dbc"
SYSPASSWORD = "123456"
SYSTEMPASSWORD = "123456"
EMCONFIGURATION = "NONE"
SYSMANPASSWORD = "123456"
DBSNMPPASSWORD = "123456"
DATAFILEDESTINATION = /oradata
#STORAGETYPE=FS
CHARACTERSET = "ZHS16GBK"
NATIONALCHARACTERSET= "AL16UTF16"
LISTENERS = "listener"
MEMORYPERCENTAGE = "40"
DATABASETYPE = "MULTIPURPOSE"
AUTOMATICMEMORYMANAGEMENT = "TRUE"
#TOTALMEMORY = "800"
!
```

#### 4. 执行静默建库语句

```
dbca -silent -createdatabase -responseFile /oracle/dbca.rsp
```

## 4. 通过CREATE DATABASE命令

1. 以oracle用户登录
2. 设置环境变量

```
export ORACLE_BASE=/oracle
export ORACLE_HOME=$ORACLE_BASE/product/11.2.0/db_1
export ORACLE_SID=mynewdb2

export ORACLE_HOSTNAME=node1
export ORACLE_UNQNAME=mynewdb2
export PATH=$ORACLE_HOME/bin:$ORACLE_HOME/jdk/bin:$PATH

mkdir -p /oracle/admin/mynewdb2/adump
mkdir -p /oracle/admin/mynewdb2/dpdump
mkdir -p /oracle/admin/mynewdb2/pfile
mkdir -p /oracle/cfgtoollogs/dbca/mynewdb2

mkdir -p /oradata/mynewdb2/
```

3. 建立参数文件



```
cat > $ORACLE_HOME/dbs/initmynewdb2.ora <<!  
*.audit_file_dest='/oracle/admin/mynewdb2/adump'  
*.audit_trail='db'  
*.compatible='11.2.0.0.0'  
*.control_files='/oradata/mynewdb2/control01.ctl'  
*.db_block_size=8192  
*.db_domain=''  
*.db_name='mynewdb2'  
*.db_writer_processes=1  
*.diagnostic_dest='/oracle'  
*.dispatchers='(PROTOCOL=TCP) (SERVICE=mynewdb2XDB)'  
*.memory_target=838860800  
*.open_cursors=300  
*.processes=1500  
*.remote_login_passwordfile='EXCLUSIVE'  
*.sessions=1655  
*.undo_tablespace='UNDOTBS'  
!
```

#### 4. 建立密码文件

```
orapwd file=$ORACLE_HOME/dbs/orapwmynewdb2 password=123456 entries=5 force=y
```

#### 5. 启动到nomount

```
sqlplus / as sysdba <<!  
startup nomount  
!
```

#### 6. 执行创建数据库语句

```
CREATE DATABASE mynewdb2
USER SYS IDENTIFIED BY oracle
USER SYSTEM IDENTIFIED BY oracle
LOGFILE GROUP 1 ('/oradata/mynewdb2/redo01a.log', '/oradata/mynewdb2/redo01b.log') SIZE 8M BLOCKSIZE 512,
GROUP 2 ('/oradata/mynewdb2/redo02a.log', '/oradata/mynewdb2/redo02b.log') SIZE 8M BLOCKSIZE 512,
GROUP 3 ('/oradata/mynewdb2/redo03a.log', '/oradata/mynewdb2/redo03b.log') SIZE 8M BLOCKSIZE 512
MAXLOGFILES 16
MAXLOGMEMBERS 3
MAXLOGHISTORY 1
MAXDATAFILES 1000
CHARACTER SET ZHS16GBK
NATIONAL CHARACTER SET AL16UTF16
EXTENT MANAGEMENT LOCAL
DATAFILE '/oradata/mynewdb2/system01.dbf' SIZE 300M REUSE
SYSAUX DATAFILE '/oradata/mynewdb2/sysaux01.dbf' SIZE 300M REUSE
DEFAULT TABLESPACE users DATAFILE '/oradata/mynewdb2/users01.dbf' SIZE 500M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED
DEFAULT TEMPORARY TABLESPACE tempts1 TEMPFILE '/oradata/mynewdb2/temp01.dbf' SIZE 20M REUSE
UNDO TABLESPACE undotbs DATAFILE '/oradata/mynewdb2/undotbs01.dbf' SIZE 200M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
```

## 7. 创建字典

```
# 在sqlplus中以sysdba执行:
SQL>@?/rdbms/admin/catalog.sql 静态字典视图, 如dba_data_files
SQL>@?/rdbms/admin/catproc.sql 创建常用的PL/SQL程序包 dbms_metadata包, 并创建publd.sql。

# 在sqlplus中以system执行: 这个脚本的作用限制用户在sqlplus中执行命令
SQL>@?/sqlplus/admin/publd.sql
```

# 四. 参数文件与告警日志

## 1. Oracle实例启动的参数文件

## 概述

- 参数文件记录了数据库的配置。
- 在数据库启动时， Oracle 要根据参数文件中的参数设置， 配置数据库。如要为各个内存池分配多少内存， 允许打开的进程数和会话等等。要想让数据库启动， 必需先读取参数文件。
- 参数文件中的参数， 我们通常称其为初始化参数， 简称为参数。
- 共有两种参数文件
  - pfile -- 文本文件, 可以直接用文本编辑器打开并修改
  - spfile -- 二进制文件, 一般在数据库中用SQL语句修改
- 命名格式
  - pfile -- initSID.ora
  - spfile -- spfileSID.ora
- 默认位置
  - \$ORACLE\_HOME/dbs

## pfile

- 特性
  - 普通文本文件
  - 在数据库服务器或客户端上都可以打开
  - 数据库启动后, 无法修改pfile中的参数

- alter system更改参数, 只是在内存中修改, 实例重启及丢失
- 文件中\*号开头的参数表示在所有实例上生效(在RAC环境中, 一个数据库可能会对应多个实例)
- 文件内容

```
MEDMAN.__large_pool_size=167772160
MEDMAN.__oracle_base='/oracle/app/oracle'#ORACLE_BASE set from environment
*.compatible='11.2.0.0.0'
*.control_files='/oradata/MEDMAN/controlfile/control01.ctl','/oraarch/MEDMAN/controlfile/control02.ctl'
*.db_block_size=8192
*.db_cache_size=1073741824
*.db_files=3000
*.db_name='MEDMAN'
```

## spfile

- 特性
  - server parameter file
  - 一个spfile只能对应一个数据库
  - 二进制文件, 不能通过文本编辑器修改
  - Linux下可以通过strings查看二进制文件
  - 可以通过RMAN工具备份spfile
  - 可以通过create spfile语句从pfile或者DBCA创建
  - 参数可以在实例运行时动态修改
  - 修改参数时可以定义是在内存中还是spfile中修改

```
SQL> alter system set parameter_name=parameter_value scope=scope_value;
# scope取值
# # scope=spfile == 在spfile参数文件中修改
# # scope=memory == 在内存中修改
# # scope=both == 默认，同时修改内存和参数文件中的值
```

## 确认当前数据库使用的参数文件

```
# 如果查询结果含有TRUE则是spfile参数文件，只有FALSE则是纯文本参数文件
SQL> select distinct ISSPECIFIED from v$spparameter;

# 查看存在于spfile的参数个数
SQL> select isspecified , count(*) from v$spparameter group by isspecified ;

# 查看spfile的位置
SQL> show parameter spfile;
```

## 参数修改

Oracle 本身有大量参数，但绝大部分参数，Oracle 都定有默认值，即使不去设置，对数据库的运行不会有任何影响。只有哪些没有默认值的参数，或者是默认值不合适的参数，才需要在参数文件中设定。也只有当你更改过某参数的值后，参数文件才会包含这个参数。对于哪些从来没有更改过的，参数文件中并不包括它们。

- ALTER SYSTEM set

```
# 参数名可以对应多个参数值
parameter_name=parameter_value, parameter_value1
# 修改undo_retention注释
comment='string'
# 选择修改的地点,默认为both
scope=[spfile|memory|both]
# 指定实例,默认为*
sid=['*' | 'sid']
```

- ALTER SYSTEM reset

```
alter system reset parameter_name scope=spfile sid='*';
```

## spfile与pfile互相备份

- 从spfile生成pfile

```
SQL> CREATE PFILE='路径/pfile名字' FROM SPFILE='路径/spfile名字';
# 上面的 PFILE 和 SPFILE 后都可以省略路径和名字, Oracle会按默认的位置, 默认的名字进行操作。
SQL> create pfile from spfile;
```

- 从pfile生成spfile

```
SQL> CREATE SPFILE='路径/spfile名字' FROM PFILE='路径/pfile名字';
# 上面的 PFILE 和 SPFILE 后都可以省略路径和名字, Oracle会按默认的位置, 默认的名字进行操作。
SQL> create spfile from pfile;
```

- 注意:
  - 在数据库实例启动状态, 最好不要生成spfile, 因为又可能随时修改参数, 这样会导致报错。

- spfile和pfile的互相备份在实例关闭下都可以进行，只要以sysdba连接登录即可。 sqlplus / as sysdba

## 2. 告警日志

---

- DBA最常用的告警日志: alert\_SID.log
  - 在数据库启动时, 在NOMOUNT阶段便会打开它; 如果文件不存在, 会在NOMOUNT阶段由Oracle自动创建
  - Oracle会将一些警告, 错误信息, 或认为重要的信息写入告警日志
  - 位置由初始化参数background\_dump\_dest决定
  - 普通文本文件
  - 包含Oracle认为DBA应该知道的信息
- 包含的信息
  - 数据库启动或关闭的时间
  - 所有非缺省初始化参数的列表
  - 后台进程的启动
  - 实例使用的线程
  - 正在向其中写入信息的日志序列号
  - 有关日志切换的信息
  - 表空间的创建和回滚段的创建
  - 已发出的警报声明

- 有关ORA-600等错误消息和区错误的信息
- 查看告警文件位置

```
SQL> SELECT name,value FROM V$DIAG_INFO
```

### 3. 跟踪文件

---

#### 普通文本文件

最大大小由初始化参数 max\_dump\_file\_size 限制。如果此参数定为 100M。那么单个用户跟踪文件的最大大小不会超过 100M。多余部分将会被截断。此参数也可以设置为“UNLIMITED”，即，不限大小。另外，告警日志的大小，不受此参数控制。在告警日志里面会有跟踪文件的信息，这部分跟踪文件协助告警日志显示问题。

#### 后台进程的跟踪文件

- 位置和告警文件一样, 主要记录后台进程的一些信息

#### 用户的跟踪文件

- 位置由初始化参数 user\_dump\_dest 决定
- 由服务器进程生成, 记录一些用户运行相关的信息, 如用户所执行的SQL语句的相关信息

#### 跟踪文件的生成

- \${ORACLE\_SID}\_ora\_{OS\_PROCESS}.trc



```
# 查看当前实例的sid
SQL> select distinct sid from v$mystat;
# 查看sid对应的进程号
SQL> select s.sid,p.spid "OS_PROCESS" from v$session s,v$process p where s.paddr=p.addr and s.sid=1137;
```

## 10046事件跟踪SQL语句

```
SQL> alter session set events '10046 trace name context forever,level 4';
# 跟踪会话级别
# # level 0:关闭跟踪
# # level 1:跟踪SQL语句,等于sql_trace=true,默认
# # level 4:包括变量的详细信息
# # level 8:包括等待事件
# # level 12:包括绑定变量与等待事件, 4+8
```

## 4. 附加

从11g开始,alert文件的格式发生了变化,除了原有的文本格式之外还增加了XML格式.

告警日志文件的储存位置还受到一个新参数的影响, diagnostic\_dest

diagnostic\_dest是11g的新特性自动诊断库(Automatic Diagnostic Repository, ADR)的设置,该目录用于存放数据库诊断日志、跟踪文件等,通常称为ADR base,该参数和环境变量ORACLE\_BASE有关:

设置了ORACLE\_BASE, 则 DIAGNOSTIC\_DEST=ORACLE\_BASE

未设置ORACLE\_BASE, 则 DIAGNOSTIC\_DEST=ORACLE\_HOME/log

查询v\$diga\_info视图可以获得跟踪诊断的相关信息,其中"Default Trace File"指出跟踪文件的可能名称:

```
SQL> select name,value from v$diag_info;
```

# 五. 网络配置

---

## 1. Oracle Net Service

---

### Oracle Net Services

- 建立从客户机或中间层应用程序到 Oracle 服务器的网络连接。
- 建立网络会话之后，Oracle Net 充当客户机应用程序与数据库服务器的中间代理人。它负责建立并维护客户机应用程序和数据库服务器之间的连接，以及在二者之间交换消息。
- Oracle Net 或 Oracle Net 的模拟程序，如 Java 数据库连接 (JDBC)，位于需要与数据库服务器建立会话的每台计算机上。
- 在客户机计算机上，Oracle Net 是一个用于连接到数据库的后台组件。
- 在数据库服务器上，Oracle Net 包括一个称为监听程序的活动进程

### Oracle Net Listener

- 负责协调数据库与外部应用程序之间的连接。
- 是所有非本地用户连接到 Oracle 实例的网关。
- 单个监听程序可用于多个数据库实例以及成千上万个客户机连接。

## 2. 创建Oracle Net Listener

---

Enterprise Manager:

提供了配置和管理 Oracle Net Services 的集成环境。使用 Enterprise Manager 可在跨多个文件系统的任何 Oracle 主目录中配置 Oracle Net Services,

## Oracle Net Manager:

启动命令 netmgr 提供一个图形用户界面 (GUI), 通过这个界面可在本地客户机或 服务器主机的 Oracle 主目录中配置 Oracle Net Services。net manager有三个部分组成:

- profile: 设置可以在oracle net客户端和服务端使用的oracle选项, 并且能够影响所有oracle net连接的行为。例如, 配置oracle net跟踪。此分支配置完后在\$ORACLE\_HOME/network/admin下生成sqlnet.ora文件。
- service naming: 配置客户端名称解析。
- listeners: 配置监听。

## Oracle Net Configuration Assistant:

启动命令 netca 安装 Oracle 软件时通过 Oracle Universal Installer 来启动。使用 Oracle Net Configuration Assistant 可配置 Oracle 数据库的监听协议地址和服务信息。

## 命令行:

用于启动、停止监听程序进程或查看监听程序进程的状态。由操作系统用户 (在本程中为 Oracle) 启动或停止监听程序。如果未启动监听程序, 则不能使用 Enterprise Manager。

## netca与netmgr

一个监听器可以监听多个地址

netca只能配置监听一个地址: 只有配完一个监听后才能对另一个监听进行向导配置

netmgr允许为一个监听器配置多个监听地址

## 监听应用程序lsnrctl

- 作用
  - 启动与停止监听程序
  - 检查监听程序的状态
  - 根据配置文件参数重新初始化监听程序
  - 动态配置多个监听程序
  - 更改监听程序口令
- 语法

```
# 启动监听
lsnrctl start [listen_name]
# 停止监听
lsnrctl stop [listen_name]
# 查看监听
lsnrctl status [listen_name]
# 查看连接
lsnrctl service [listen_name]
```

- 服务的状态
  - ready : 动态监听
  - unknown : 静态监听

## 3. 动态监听与静态监听

---

## 动态监听

- 可以随时监听到实例是否开启, 只有在实例开启时监听才会注册
- 缺省情况

```
LISTENER =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = node1)(PORT = 1521))  
  )  
  
ADR_BASE_LISTENER = /oracle
```

- 修改端口

```
# 停掉监听
lsnrctl stop
# 修改监听配置文件中的端口
vim listener.ora

LISTENER =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = node1)(PORT = 1523))
  )

ADR_BASE_LISTENER = /oracle
# sys登录数据库,修改local_listener参数值
SQL> alter system set local_listener="(address=(protocol=tcp)(host=192.168.56.102)(port=1523))";
# 查看参数是否修改
SQL> show parameter local_listener;
# 退出数据库启动监听
lsnrctl start
# 查看监听服务(启动有延时)
lsnrctl service
# 服务摘要显示后登陆数据库
sqlplus scott/tiger@192.168.56.102:1523/zzdb1
```

SQL\*Plus: Release 11.2.0.1.0 Production on 星期五 12月 3 14:47:00 2021

Copyright (c) 1982, 2009, Oracle. All rights reserved.

连接到:

Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> show parameter local\_listener

ORA-00942: 表或视图不存在

SQL> exit

从 Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options 断开  
[oracle@node1 ~]\$ sqlplus sys/oracle@192.168.56.102:1523/zzdb1 as sysdba

SQL\*Plus: Release 11.2.0.1.0 Production on 星期五 12月 3 14:48:47 2021

Copyright (c) 1982, 2009, Oracle. All rights reserved.

连接到:

Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> show parameter local\_listener

NAME	TYPE	VALUE
local_listener	string	(address=(protocol=tcp)(host=192.168.56.102)(port=1523))

SQL> █

连接到:

Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> exit

从 Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options 断开  
[oracle@node1 ~]\$ sqlplus scott/tiger@zzdb1

SQL\*Plus: Release 11.2.0.1.0 Production on 星期五 12月 3 14:45:09 2021

Copyright (c) 1982, 2009, Oracle. All rights reserved.

连接到:

Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> exit

从 Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options 断开  
[oracle@node1 ~]\$ sqlplus scott/tiger@192.168.56.102:1523/zzdb1

SQL\*Plus: Release 11.2.0.1.0 Production on 星期五 12月 3 14:47:00 2021

Copyright (c) 1982, 2009, Oracle. All rights reserved.

连接到:

Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> █



服务摘要..

服务 "zzdb1" 包含 1 个实例。

实例 "zzdb1", 状态 READY, 包含此服务的 1 个处理程序...

处理程序:

"DEDICATED" 已建立:2 已拒绝:0 状态:ready

LOCAL SERVER

服务 "zzdb1XDB" 包含 1 个实例。

实例 "zzdb1", 状态 READY, 包含此服务的 1 个处理程序...

处理程序:

"D000" 已建立:0 已被拒绝:0 当前: 0 最大: 1022 状态: ready

DISPATCHER <machine: node1, pid: 21979>

(ADDRESS=(PROTOCOL=tcp)(HOST=node1)(PORT=44347))

命令执行成功

[oracle@node1 admin]\$ vim tnsnames.ora

[oracle@node1 admin]\$ lsnrctl service listener1

LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 03-12月-2021 14:47:28

Copyright (c) 1991, 2009, Oracle. All rights reserved.

正在连接到 (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=node1)(PORT=1523)))

服务摘要..

服务 "zzdb1" 包含 1 个实例。

实例 "zzdb1", 状态 READY, 包含此服务的 1 个处理程序...

处理程序:

"DEDICATED" 已建立:4 已拒绝:0 状态:ready

LOCAL SERVER

服务 "zzdb1XDB" 包含 1 个实例。

实例 "zzdb1", 状态 READY, 包含此服务的 1 个处理程序...

处理程序:

"D000" 已建立:0 已被拒绝:0 当前: 0 最大: 1022 状态: ready

DISPATCHER <machine: node1, pid: 21979>

(ADDRESS=(PROTOCOL=tcp)(HOST=node1)(PORT=44347))

命令执行成功

[oracle@node1 admin]\$

## 静态监听

- 不能像动态监听一样随时知道实例是否打开
- 可以在实例未打开的时候通过静态监听远程登录管理员打开实例(方便用在服务器端无人看顾的情况下)
- local\_listener参数缺省情况下, 静态监听端口设置为1521会同时存在静态监听和动态监听
- 设置完全的静态监听:

```
# 配置文件中增加数据库相关信息以及修改端口
vim listener.ora

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = zzdb1)
      (ORACLE_HOME = /oracle/product/11.2.0/db_1)
      (SID_NAME = zzdb1)
    )
  )
LISTENER =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = node1)(PORT = 1522))
  )
ADR_BASE_LISTENER = /oracle
# 启动监听
lsnrctl start
# 连接数据库
sqlplus scott/tiger@192.168.56.102:1522/zzdb1
```

服务摘要..

服务 "zzdb1" 包含 1 个实例。

实例 "zzdb1", 状态 UNKNOWN, 包含此服务的 1 个处理程序...

命令执行成功

[oracle@node1 admin]\$ lsnrctl service

LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 03-12月-2021 15:01:28

Copyright (c) 1991, 2009, Oracle. All rights reserved.

正在连接到 (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=node1)(PORT=1522)))

服务摘要..

服务 "zzdb1" 包含 1 个实例。

实例 "zzdb1", 状态 UNKNOWN, 包含此服务的 1 个处理程序...

处理程序:

"DEDICATED" 已建立:0 已被拒绝:0

LOCAL SERVER

命令执行成功

[oracle@node1 admin]\$ lsnrctl service

LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 03-12月-2021 15:01:43

Copyright (c) 1991, 2009, Oracle. All rights reserved.

正在连接到 (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=node1)(PORT=1522)))

服务摘要..

服务 "zzdb1" 包含 1 个实例。

实例 "zzdb1", 状态 UNKNOWN, 包含此服务的 1 个处理程序...

处理程序:

"DEDICATED" 已建立:1 已被拒绝:0

LOCAL SERVER

命令执行成功

[oracle@node1 admin]\$ vim listener.ora

[oracle@node1 admin]\$ vim listener.ora

[oracle@node1 admin]\$

NAME	TYPE	VALUE
local_listener	string	(address=(protocol=tcp)(host=192.168.56.102)(port=1523))

```
SQL> alter system set local_listener="";
alter system set local_listener=""
*
```

第 1 行出现错误:  
ORA-01741: 非法的零长度标识符

```
SQL> alter system set local_listener="(address=(protocol=tcp)(host=192.168.56.102)(port=1521))";
```

系统已更改。

```
SQL> alter system register;
```

系统已更改。

```
SQL> exit
```

从 Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options 断开  
[oracle@node1 ~]\$ sqlplus sys/oracle@192.168.56.102:1522/zzdb1 as sysdba

SQL\*Plus: Release 11.2.0.1.0 Production on 星期五 12月 3 15:01:40 2021

Copyright (c) 1982, 2009, Oracle. All rights reserved.

连接到:

Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options

```
SQL> █
```

## 4. 服务名

---

- 可以把服务理解为一组相似的逻辑组合, 即分组.
- 举例: 对于一个典型的销售企业的业务系统来说, 可以大致分为销售、采购、财务和人事这四大业务部门。那么从数据库的角度来说, 就可以划分成对应的四个服务名: sales (销售)、pur (采购)、fin (财务) 和hr (人事)。处理销售相关业务的人员全都连接到sales服务上进行操作而处理采购业务的人员则全部连接到pur服务上进行操作
- 通过这种方式, 我们对数据库的活动分了组, 从而可以**更好的进行跟踪和调优**的工作。

```
# 查看默认的服务名
SQL> show parameter service_names
# 该参数可以动态修改, 并且可以指定多个值, 也就是多个服务名, 每个服务名之间用逗号隔开。
# 设置服务名
SQL> alter system set service_names='orcl,zzdb';
# 根据服务名登录
# sqlplus scott/tiger@zzdb1
```