

配置和管理Oracle数据库

1. Oracle体系结构与基本操作

1.Oracle管理的基本操作

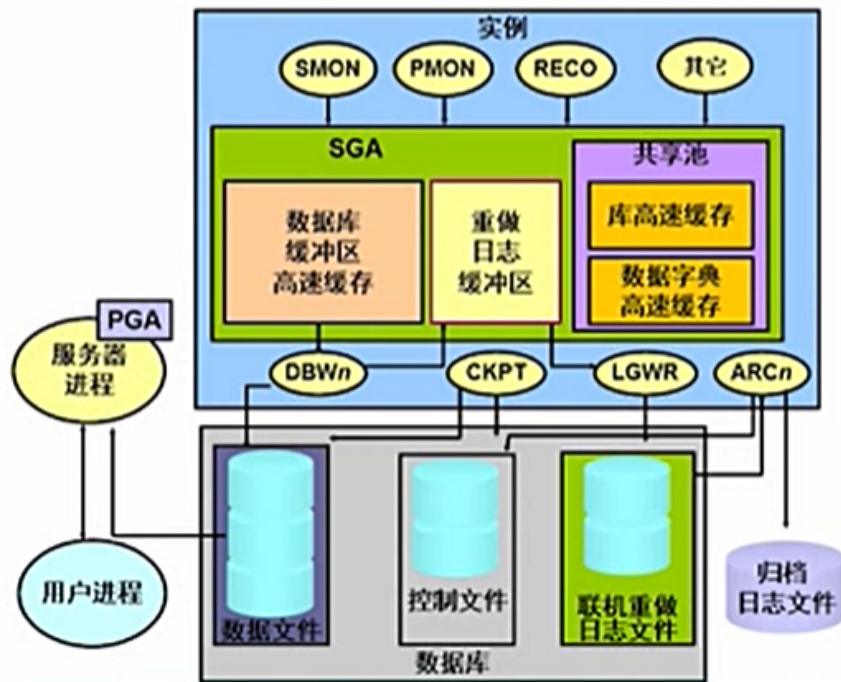
(Linux系统)切换到Oracle用户登陆数据库

```
su - oracle          # 切换用户
sqlplus / as sysdba  # 登陆Oracle
startup              # 启动实例
select instance_name, status from v$instance; # 查看实例状态，查看到状态是open 代表状态正常。
shutdown immediate; # 关闭数据库实例
lsnrctl start        # 为了让数据库能对外提供服务，需要启动监听，在Linux命令行执行
lsnrctl status        # 查看监听状态，在Linux命令行执行
```

2.列出Oracle DB的主要体系结构组件

- 问题1：使用数据文件直接存放、读取数据存在的问题。
答：磁盘的读取速度远远低于CPU运转速度，限制了性能，通过缓存解决。
- 问题2：使用内存，作为数据文件的缓存，是做只读缓存，还是读写缓存？
答：为了性能，使用读写缓存。
- 读写缓存方案，数据易丢失的问题如何解决？
答：通过日志文件记录每一次操作。日志是直接写在磁盘上的。
- 问题4：Oracle数据库在发生进程崩溃，操作系统崩溃，掉电等。能不能保障数据的可靠性？
答：可以，只要文件未损坏。

Oracle DB 体系结构：概览



3.说明内存结构

4.描述后台进程

进程体系结构

数据库结构
- 内存
→ 进程
- 存储

- 用户进程
 - 当数据库用户或批处理进程连接到 Oracle DB 时启动
- 数据库进程
 - 服务器进程：可以连接到 Oracle 实例，它在用户建立会话时启动
 - 后台进程：在启动 Oracle 实例时启动



- 数据库写进程（DBWn）

将数据库缓冲区高速缓存中经过修改的缓冲区（灰数据缓冲区）写入磁盘的两种方式：

- (1) 在执行其它处理时异步执行
- (2) 定期执行以推进检查点



- 日志写进程 (LGWR)

将重做日志缓冲区写入磁盘上的重做日志文件中，在以下情况下执行写操作：

- (1) 用户进程提交事务处理时
- (2) 重做日志缓冲区的三分之一满时
- (3) 在DBWn进程将经过修改的缓冲区写入磁盘之前



- 检查点进程 (CKPT) (向DBWn发出执行检查点的命令，同时监控DBWn写入文件的进度)

将检查点信息记录在以下位置

- (1) 控制文件
- (2) 每个数据文件头

- 系统监视器进程 (SMON)

- (1) 在实例启动时执行恢复
- (2) 清除不使用的临时段

- 进程监视器进程 (PMON)

- (1) 在用户进程失败是执行进程恢复
清除数据库缓冲区的高速缓存
释放该用户进程使用的资源
- (2) 监视会话是否发生空闲会话超时
- (3) 将数据库服务动态注册到监听程序

数据库存储体系结构

数据库存储体系结构

数据库结构

- 内存

- 进程

→ 存储



控制文件



数据文件



联机重做日志文件



参数文件



备份文件



归档重做日志文件



口令文件



预警日志和跟踪文件

实验环节

使用SQL Developer工具进行

连接不成功可能原因：

- (1) 网络问题
- (2) Oracle数据库没有启动
- (3) Oracle数据库没有启动监听

- 内存结构相关命令

```
select * from v$sga;  
select * from v$sgainfo;  
select * from v$sgastat;  
select * from v$pgastat;
```

- 进程信息相关命令

```
select * from v$process;
```

- 会话信息相关命令

```
select * from v$session;
```

- 表空间

```
select * from dba_tablespaces;  
select * from v$tablespace;
```

- 数据文件查看

```
select * from dba_data_files;
select * from v$datafile;
```

- 日志操作

```
select * from v$log;
select * from v$logfile;
alter system switch logfile;
```

2.Oracle实例管理与配置文件设置

1.Oracle 启动过程

nomount阶段---->mount阶段----->open阶段

命令:

```
startup      # 启动数据库
-----分阶段启动-----
startup nomount;
alter database mount;
alter database open;
```

四种关闭模式:

关闭模式

关闭模式	A	I	T	N
允许新连接	否	否	否	否
等待当前会话结束	否	否	否	是
等待当前事务处理结束	否	否	是	是
强制选择检查点并关闭文件	否	是	是	是

关闭模式:

- A = ABORT
- I = IMMEDIATE
- T = TRANSACTIONAL
- N = NORMAL

数据库运行过程中后台日志的路径查看

```
show parameter dump
```

2.Oracle的配置从哪里来

Linux下执行以下命令查看数据库配置

```
set | grep ORA
```

```
[oracle@myhost ~]$ set | grep ORA
ORACLE_BASE=/ora01/app/oracle
ORACLE_HOME=/ora01/app/oracle/product/11.2.0/db_1
ORACLE_SID=orcl
ORACLE_UNQNAME=orcl
[oracle@myhost ~]$
```

ORACLE_SID 确定连接的是哪个数据库实例。

Oracle用户下的 `.bash_profile` 文件在每次Oracle用户登陆时使用，如果修改了里面的内容，不会马上生效。需要重新登陆才能生效。

参数文件的选用

- 参数文件路径 \$ORACLE_HOME/dbs
- oracle在启动时回一次寻找下面三个文件，并使用第一个有效文件。

spfile.ora

spfile.ora

init.ora

- 查看当前使用的参数文件

方法一：

```
show parameter spfile
show parameter control
```

方法二：

```
select * from v$parameter      #数据库运行时的参数
select * from v$spparameter where name like 'sga'; #数据库启动时的一些参数
```

```
show parameter db_cache_size; # 查到值为0，它有Oracle动态管理
show sga;
alter system set db_cache_size = 300 scope=memory #修改db_cache_size 的大小,scope=memory , scope=spfile,scope=both
```

3.如何查看和修改参数

4.Oracle的常用参数

5.两种参数文件的使用方法

- spfile:
 - 二进制
 - 有校验位
 - 不能直接编辑
 - 在数据库中进行修改 (alter system set ... scope=spfile;)
- pfile:
 - 纯文本
 - 直接用文本编辑器编辑
 - 不在数据库中进行修改

将spfile改成pfile

```
create pfile='/home/oracle/pfile.ora' from spfile
```

3.Oracle网络管理

- 客户端是如何连接到服务器的
 - 启动监听

```
lsnrctl start #启动监听
sqlplus / as sysdba
```

通过IP 端口号服务名链接数据库,通过以下命令获取或通过配置文件查看。配置文件所在位置:

```
$ORACLE_HOME/network/admin/listener.ora
```

```
lsnrctl status
```

简便连接方法登陆数据库:

```
sqlplus system/password@IP:Port/服务名
```

如:

```
sqlplus oracle/oracle@192.168.1.104:1521/orcl
```

查看信息:

```
show user;
select count(*) from user_tables;
```

简便连接方法不能提供高可用和负载均衡

本地命名方式登陆: (可以使用图形化界面, 需要Xming工具)

可以提供高可用、故障转移和负载均衡等。管理也比较方便

- **Oracle数据库是如何对我提供服务的**

启动默认监听

```
lsnrctl start
```

查看监听状态

```
lsnrctl status
```

关闭默认监听

```
lsnrctl stop
```

新监听创建方法

方法一：netca

方法二：netmgr工具

方法三：编辑listener.ora 文件

非默认监听管理

```
lsnrctl start 监听名  
lsnrctl status 监听名  
lsnrctl stop 关闭默认监听
```

tnsping查看监听状态

```
tnsping 192.168.1.104:1521
```

注意： 监听成功不一定能被访问，因为监听上面还有不同的实例不同的实例名。

从操作系统层面观察监听

```
netstat -anltp
```

查看相关进程

```
ps -ef | grep tns
```

服务注册过程：

- 动态注册

实例启动之后，会注册到默认监听（每一分钟执行一次），即1521端口。由PMON进程完成。

快速注册方法：

```
alter system register;    # 在sql中执行
```

动态注册的服务状态是ready，静态注册的服务总是UNKNOWN的。

- 静态注册

用图形化界面注册，反应到 listener.ora 文件即修改如下内容。或者直接修改也行。

```
LISTENER1 =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = db11)(PORT = 1522))  
  )  
  
SID_LIST_LISTENER1 =  
  (SID_LIST =  
    (SID_DESC =  
      (GLOBAL_DBNAME = OASVC)  
      (ORACLE_HOME = /u01/app/oracle/product/11.2.0/db_1)  
      (SID_NAME = ORCL)  
    )  
  )  
)
```

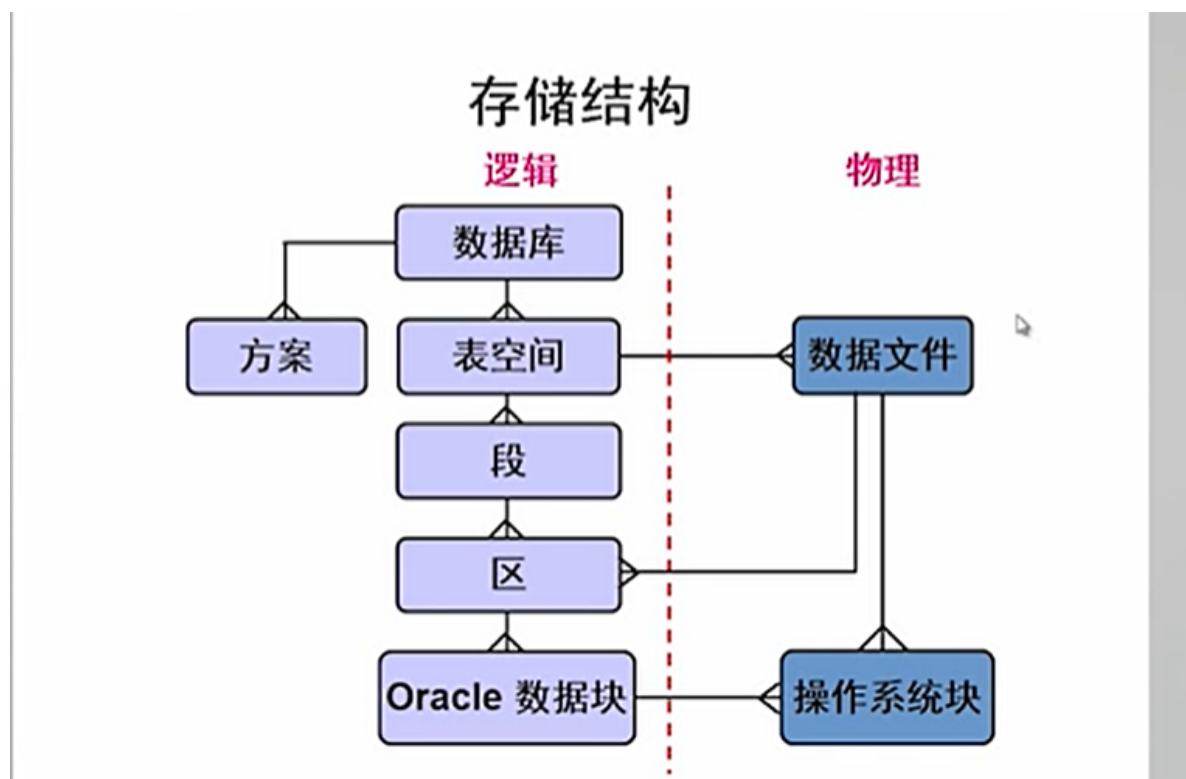
静态注册和动态注册在数据库关闭状态下的区别

关闭数据库

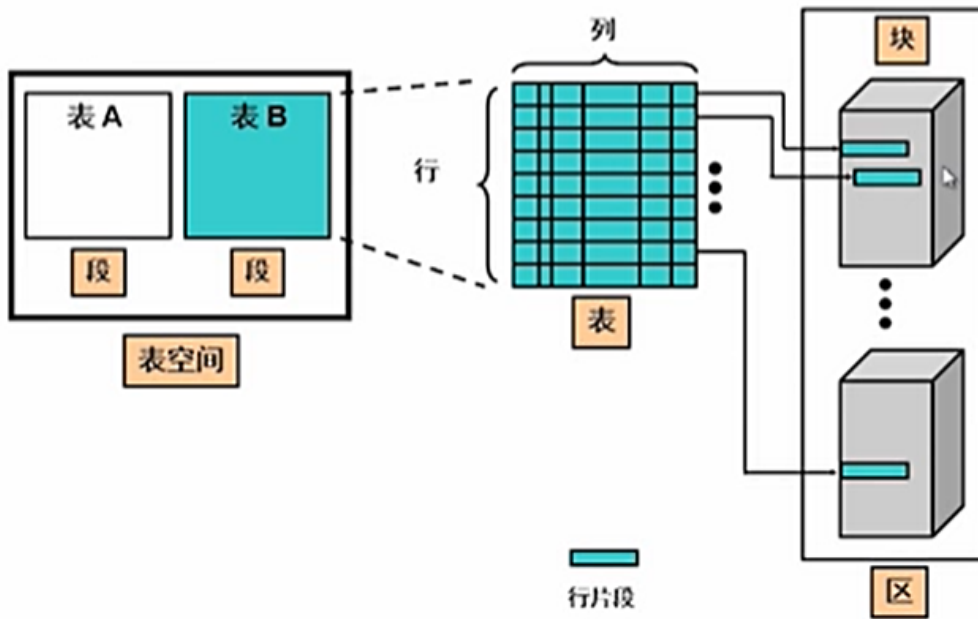
```
shutdown immediate; # sql 中执行  
lsnrctl status       # 查看服务状态
```

结论：动态注册（默认注册）提示无服务，静态注册服务依然存在。

4.Oracle逻辑存储与表空间管理



表数据的存储过程



数据库当中查看到表空间、段、区、块操作：

=====查看系统逻辑存储情况=====

查看数据库的数据库文件的相关状态

```
SELECT * FROM dba_data_files;
```

查看 表空间

```
SELECT * FROM dba_tablespace;
```

有六个默认表空间：SYSTEM 表空间存放数据字典，非常重要；SYSAUS 存放历史性能信息，一般保留最近八天的性能快照；UNDOTBS1保障回滚用的，一般自动还原管理；TEMP临时数据存放空间；USERS，用户数据，一般不使用；EXAMPLE示例方案表空间，一般用于学习。

查看 所有的段

```
SELECT * FROM dba_segments;
```

查看 所有的区

```
SELECT * FROM dba_extents;
```

查看一般用户的表/段/区

```
select * from dba_tables where owner = '用户名';
select * from dba_segments where owner = '用户名';
select * from dba_extents where owner = '用户名';
```

创建表空间

```
create tablespace testtbs
datefile '/ora01/app/oradata/orcl/testtbs01.dbf' size 10m
autoextend on next 10m maxsize 100m
extent management local autoallocate
segment space management auto;
```

解释:

size: 大小

autoextend on : 启用数据文件自动扩展, 自动扩展的扩展量是10m 最大100m

extent management DICTIONARY : 区管理: 字典管理 (不建议)

extent management local : 区管理: 本地管理 (默认)

autoallocate: 新分配的区大小自动调整 (默认)

uniform size 10m: 新分配的区大小固定为10m

segment space management auto: 段空间管理: 自动 (默认)

auto --> manual : 段空间管理: 手动 (不建议)

在指定表空间创建表

```
create table newtab(id number) tablespace testtbs; #创建表空间
select * from dba_segments where segment_name='NEWTAB'; # 表明自动转换为大写
select * from dba_extents where segment_name='NEWTAB';
alter table newtab allocate extent;
```

查看用户的默认表空间

```
select * from dba_users;
```

更改用户的默认表空间

```
alter user hr default tablespace testtbs;
```

=====表空间维护=====

调整数据文件大小

```
alter database datafile '/ora01/app/oradata/orcl/testtbs01.dbf' resize 5m
```

调整数据文件, 启用自动扩展

```
alter database datafile '/ora01/app/oradata/orcl/testtbs01.dbf' autoextend on
next 10m maxsize 200m;
```

为表空间增加数据文件

```
alter tablespace testtbs add datafile '/ora01/app/oradata/orcl/testtbs02.dbf'  
size 10m
```

删除表

```
drop table newtab;
```

删除表空间

```
drop tablespace testtbs including contents and datafiles; #删除表空间和表
```

表空间使用率查询

```
select total.tablespace_name, total_bytes, free_bytes, total_bytes-free_bytes  
used_bytes,  
round(((total_bytes-free_bytes)/total_bytes*100,2) used_rate from  
(select tablespace_name, sum(bytes) total_bytes from dba_data_files group by  
tablespace_name) total,  
(select tablespace_name, sum(bytes) free_bytes from dba_free_space group by  
tablespace_name) free  
where total.tablespace_name=free.tablespace_name;
```

5.Oracle高可用技术

IT系统的可用性指标

- 平均故障间隔时间 (MTBF)
 - 英文全称: Mean Time Between Failure
 - 指相邻两次故障之间的平均工作时间, 是衡量一个产品的可靠性指标
- 平均修复时间 (MTTR)
 - 英文全称: Mean Time To Repair
 - 描述产品由故障状态转为工作状态时修理时间的平均值
- 可用性 (Availability)
 - $Availability = MTBF / (MTBF + MTTR)$

- 共享存储——单活
使用共享存储、虚拟IP、冷备，
MTTR较长
 - RHCS
 - AIX HACMP
 - MSCS
- 共享存储——双活/多活
 - Oracle RAC
 - DB2 pureScale
- 数据同步
数据变化传输
 - Oracle DataGuard / Golden Gate
 - MySQL Replication
 - Linux DRBD
- 数据的分布式冗余
 - Oracle ASM
 - Greenplum
 - Hadoop HDFS

6.Oracle RMan备份恢复

逻辑备份与物理备份

- 逻辑备份
 - 表、行
 - 数据泵expdp，导出工具exp
 - 类似全表扫描，会有SQL的分析、数据块的分析、insert语句的生成。
 - 较慢，开销较大

逻辑备份与物理备份

• 逻辑备份

- 一致性读，使用UNDO，导出数据为同时刻同版本的一致数据，如果数据量很大，导出会很慢，对undo的保留时间要求很长。
- 可以按表导入
- 只能恢复导出时刻的数据。

逻辑备份与物理备份

• 物理备份

- RMAN允许在数据文件做修改的时候，对文件进行复制。复制后，数据文件仍然是错乱的，但是可以使用 archive log 对文件进行恢复，使其成为一致的。
- 只能全库恢复，不能按表恢复。
- 可以恢复到任意时刻（备份完成之后，当前时刻之前）
- 备份时一定要同时备份归档日志文件。

登陆RMAN

```
set | grep ORA # 查看Oracle_home 和 oracle_sid  
rman target / # 登陆RMAN，登陆的是本Oracle_home 下的 本SID实例
```

远程登陆RMAN

```
rman target oracle/oracle@192.168.1.104:1521/orcl
```

查看RMAN配置信息

```
show all; #查看所有配置信息,RMAN中查询,所有configure 策略.
select name,value from v$rman_configuration; #SQL 中查询方法
delete obsolete; # 删除过期或历史备份.
```

=====常用configure选项=====

保存策略(retention policy)

```
configure retention policy to recovery window of 3 days;
configure retention policy to redundancy 3; #保留3个有效副本
configure retention policy clear;
```

=====使用RMAN进行在线备份=====

将数据库设置成归档模式:

在Oracle中查看是否为归档模式

```
sqlplus sys/oracle@orcl as sysdba #登陆数据库,普通用户登陆会权限不足.
archive log list; #查看automatic archival 状态 ,disable为非归档模式
```

启用为归档模式

```
shutdown immediate; # 关闭数据库
startup mount; #启动mount 阶段
alter database archivelog; # 在mount 阶段切换为归档模式
alter database open; # 切换为open状态.
archive log list; # 再次查看状态,enable 为成功,此时可以实现RMAN在线热备.
```

备份:

```
backup database; # 在RMAN中执行该命令,备份所有的数据库.
backup archivelog all; # 备份归档日志
```

备份数据到指定路径

```
backup database format '/u01/backup/%d_db_%s_%T';
```

备份为压缩备份

```
backup as compressed backupset database;
```

备份数据文件及归档日志

```
backup database plus archivelog;
backup database plus archivelog delete all input;
```

0级增量备份

```
backup incremental level=0 database;
backup as compressed backupset incremental level=0 database;
```


差异增量备份

```
backup as compressed backupset incremental level=1 database;
```

累计增量备份

```
backup as compressed backupset cumulative incremental level=1 database;
```

备份表空间

```
backup tablespace users;  
backup tablespace users format '/u01/backup/%d_db_%s_%T';  
backup as compressed backupset tablespace users format  
'/u01/backup/%d_db_%s_%T';
```

备份数据文件

```
backup datafile '/ora01/app/oradata/orcl/testtbs01.dbf';  
backup datafile 4;
```

=====数据的恢复=====

使用 RMAN RESTORE 和 RECOVER 命令

- RESTORE 命令：从备份中还原数据库文件
- RECOVER 命令：通过应用重做日志文件中记录的更改来恢复已还原文件

```
RMAN> SQL 'ALTER TABLESPACE inv_tbs  
OFFLINE IMMEDIATE';  
RMAN> RESTORE TABLESPACE inv_tbs;  
RMAN> RECOVER TABLESPACE inv_tbs;  
RMAN> SQL 'ALTER TABLESPACE inv_tbs  
ONLINE';
```

restore恢复的是控制文件,在nomount下完成; 若恢复的是数据文件,在mount下完成.

=====创建一个表空间以及该表空间的一个表,并且进行备份.删除数据文件,进行恢复=====

1.使用SQL*PLUS,以system用户连接实例,创建一个表空间

```
create tablespace test_tbs datafile '/ora01/app/oradata/orcl/testtbs01.dbf' size  
10m;
```

2.在新表空间内创建一个表,并插入两条记录


```
create table test01(a number) tablespace test_tbs;  
insert into test01 values(10);  
insert into test01 values(20);  
commit;  
select * from test01;
```

3.在rman中备份

```
backup tablespace "TWST_TBS";  
list backup;    # 查看所有的备份
```

4.删除文件

```
rm /ora01/app/oradata/orcl/testtbs01.dbf    #shell中执行
```

5.查看文件

```
select * from test01;    # 查看文件,此时可能能够查看到,因为在dbbuffer_cache中存在  
alter system flush;      # 清除db_buffer_cache中缓存
```

6.在RMAN中恢复

```
recover datafile '/ora01/app/oradata/orcl/testtbs01.dbf';
```

Oracle RMAN 异机恢复

- 安装相关软件
- 创建相关路径
- 创建参数文件
- 启动到nomount阶段
- 恢复控制文件
- 启动到mount阶段
- 注册备份信息
- 设置数据文件新路径
- 恢复数据文件到新路径
- 设置恢复日志的目标时间点
- 恢复日志
- 以只读模式打开，查看数据有效性
- 以resetlog模式打开

配置和管理Oracle数据库



负责人： 华北

总学时: 303分钟 已学学时: 303分钟

总学分: 30.30分 已获学分: 30.30分

进度排名： 第 1名



要求完成日期：2021-04-09

配置和管理Oracle数据库		学时：303分钟
任务 1	配置和管理Oracle数据库	进度：100%