Oracle课堂笔记

1 登陆

C:\Users\格子>sqlplus sys/admin as sysdba

SQL> conn / as sysdba

SQL> conn 用户名/密码

2 对用户解锁

SQL> alter user hr account unlock;（分号！）

3 修改密码

SQL> password hr

SQL> alter user hr identified by hr;

4 设置

set linesize 120 pagesize 25

set sqlprompt \_user"@"\_connect\_identifier>

SQL> col username for a20 设置某一列的宽度（username 的宽度为20个字符）

5 查看数据库所有的用户

SQL> desc dba\_users

1 界面 sqldeveloper

D:\app\oracle\product\18.3.0\sqldeveloper

sqldeveloper.exe

启动监听

lsnrctl start

(lsnrctl status查看是否启动监听)

增加连接（连接名，用户，口令，主机，端口号，SID）

sys角色是sysdba

2 EM Express

SQL> desc dbms\_xdb\_config

select dbms\_xdb\_config.GETHTTPPORT from dual;

execute dbms\_xdb\_config.SETHTTPPORT(端口号)；

http://主机名（XXX-pc:端口号）/em

http://LAPTOP-3C27SH8C:8989/em

3 查询SID

SYS@db18c>select instance\_name from v$instance;

startup 启动数据库

shutdown immedia 关闭数据库

oracle体系架构

oracle服务器（数据处理+数据存储 功能强大）

有两个部分：

一、 instance

实例：例程

1 内存结构

1）SGA（system global area系统全局区）所有会话共享 instance创建时申请

2）PGA（process global area程序全局区）每一个会话私有 当创建会话时申请

https://blog.csdn.net/ArnoBM/article/details/82387479

二、 database 文件集合

集群

用户：

1）.sys

连接有两种身份：sysdba（管理instance和database，即整个oracle服务器）

sysoper（只管理insatance）

2）.system（只管理database）不能shutdown

代码

1）.数据（用户数据+系统数据）

共享池

1 库缓存（SQL区，PL/SQL[过程性SQL语言]区）

2 数据字典缓存

数据字典（重点 使用）

1 动态性能视图（指向内存，控制文件）

v$开头

2 数据字典视图（指向数据文件）

dictionary 这是一个总表

SYS@db18c>desc dict

https://www.cnblogs.com/pengineer/p/4419275.html

user\_

all\_

dba\_

● USER\_：用户所拥有的相关对象信息；

● ALL\_：用于有权限访问的所有对象的信息；

● DBA\_：数据库所有相关对象的信息，用于需要有 select any table 的权限才能访问。

RBO 基于规则的优化器

CBO 基于开销的优化器

cache 提高读的效率 buffer 提高写的效率

1 redo log

do undo 撤销，对冲，回滚 redo 重做，恢复，前滚 日志：记录事务

ACID准则

原子性 一致性 isolation隔离性 durable持久性，永久性

2

restore 备份文件还原 recover 恢复

表 表空间 数据文件

创建表空间

create tablespace demo datafile 'D:\APP\ORACLE\ORADATA\DB18C\demo01.DBF' size 10m;

创建表

create table scott.st(id int,name varchar(20)) tablespace demo;

插入数据

insert into scott.st values(XXX);

EG:insert into st values(1,'Tom');

alter system checkpoint;//系统已更改

关掉表

alter tablespace demo offline;

删除表

\*.dbf----\*.bak **没有备份文件进行恢复**

alter database create datafile XX;利用日志

EG:

日志文件：

ORA-00376: 此时无法读取文件 5

ORA-01110: 数据文件 5: 'D:\APP\ORACLE\ORADATA\DB18C\DEMO01.DBF'

alter database create datafile 5;

recover datafile xx;//介质恢复

recover datafile 5;

完成介质恢复

alter tablespace demo online; //启动表

3 进程结构

1） 用户进程

UI：提供访问oracle的一个界面

名词解析 端口 协议 TCP 监听器 寻找服务器进程

2） 服务器进程

用户进程的代理 专用服务器进程 当用户连接数<200 共享服务器进程 200~3000

3） 后台进程 v$bgprocess

PMON：进程监视器进程 处理用户进程的异常终止

process monitor

SMON：系统监视器进程 system monitor

DBWR：数据库书写器进程 将数据从数据库高速缓存中写到数据文件

LGWR：日志书写器进程 从日志缓存中写道日志文件中

CKPT：检查点进程

select current\_scn,checkpoint\_change# from v$database;

scn:system change number 系统改变号：随时间变化的序列

checkpoint\_change# 检查点号：相对静止点

手工做检查点：alter system checkpoint；

检查点checkpoint作用：

1）同步所有的数据文件

select name,checkpoint\_change# from v$datafile;

2）同步所有的控制文件

select name,checkpoint\_change# from v$database;

3）发送信号通知DBWR写盘

检查点长：性能好，出现问题时恢复时间长

短：性能差，恢复时间短

时间根据业务需求

show parameter mttr;

fast\_start\_mttr\_target

0 为0自动检查点禁用，手动做检查点

4 SQL D80190GC11

1) QL (select)

2) DML (insert,uodate,delete,merge)

3) DDL (create,alter,drop,truncate,rename,comment)

系统语言

4) DCL (grant赋予权限,revoke取消权限)

grant select \* from scott.emp to hr;

5) TCL (commit,rollback,savepoint)

SELECT 返回表中的所有的数据

select \* from tablename;

投影 select .... from ...

选择 where...

连接 排序 实现top n（前多少个）考！

**select ... from .. order by 1(第一列) desc(降序)；**

**返回前n个**

**select ... from .. order by 1(第一列) desc(降序)；**

**fetch first n rows only；**

**选取第n个到第m个，偏移多少个取前几个**

**select ... from .. order by 1(第一列) desc(降序)；**

**offset n-1 rows fetch first m-n(n到m有多少行) rows only;**

**和第n个一样的连带显示**

**fetch first n rows with ties；**

sql命令不允许缩写 但sqlplus允许

显示所有sqlplus命令：help index

NULL 三值现象 where XX is null；

单引号 oracle中无论字符还是字符串都使用单引号 双引号

起别名

select 'a' a from XXX; 别名为大写A

select 'a' "a" from XXX; 别名为a

单引号中有单引号：里面的用两个单引号

select 'I''m a student' from XXX;

escape 查找实际的%和\_符号

1 和用户交互：提交变量

1）& 2）define 预先给变量定义一个值 3）&&

select empno,ename,sal from emp where empno=7369;

select empno,ename,sal from emp where empno=工号;

select empno,ename,sal from emp where ename=upper('&xm');姓名 大小写

sql 配置： 查看配置show 设置：set（set只对本次登陆有效）

2 函数 查找手册

sysdate 日期 current\_date systimestamp

systimestamp(9)毫秒为9位 最多9位

months\_between trunc 取整 mod 求模

1）nls参数

nls\_date\_format

用两位数表示年份时：yy格式：永远和系统日期处于同一个世纪；rr格式：

alter session set nls\_date\_format = 'yyyy-mm-dd hh:mi:ss';(hh24:mi...24小时制；hh:mi:ss am 表示上午)

select sysdate from dual;

eg:SCOTT@db18c>**alter session set nls\_date\_format = 'yyyy-mm-dd hh:mi:ss';**

2）单行函数——针对每一行进行处理

upper lower

concat 字串合并,参数为2个

eg: select concat('hello','world') from dual; 输出：helloworld

substr 查找子串 负数为从右边开始数

instr 定位某个字符在字串中的位置

eg: SCOTT@db18c>select INSTR('helloworld','w')from dual;

INSTR('HELLOWORLD','W')

-----------------------

6

SCOTT@db18c>select INSTR('helloworld','l',1,2)from dual;从第一个位置开始向左数开始第二个l的位置

INSTR('HELLOWORLD','L',1,2)

---------------------------

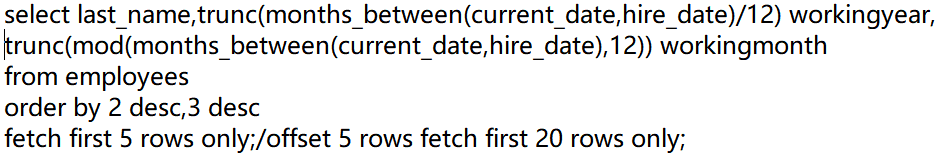
4

lpad 左填充 从前面开始补空格 rpad 右填充 round 正数保留小数位

例：

在hr用户下，有一个employees表 马尔代夫 5个名额 昆明 20个 计算到现在的工作时间 前5个 5个后的前20个

last\_name years month



3）转换函数 TO\_XX

to\_char(,' ')示例图381

修改本地货币

eg:SCOTT@db18c>alter session set NLS\_TERRITORY='United Kingdom';//会话已更改。

SCOTT@db18c>select empno,ename,to\_char(sal,'L99,999.99') sal from emp;

4）处理空值函数

NVL NVL2 nullif coalesce

1 条件表达式 处理if----then----else 逻辑

1) case语句 2) decode语句

练习

hr.employees 输出工作年份对应的人数

select count(\*) total,

sum(decode(to\_char(hire\_date,'yyyy'),'2001',1,0)) "2001",

sum(decode(to\_char(hire\_date,'yyyy'),'2002',1,0)) "2002",

sum(decode(to\_char(hire\_date,'yyyy'),'2003',1,0)) "2003"

from employees;

TOTAL 2001 2002 2003

---------- ---------- ---------- ----------

107 1 7 6

查找最大Max 最小 Min 参数只有一个

查找几个数中的最大的 greatest 最小的 least

HR@db18c>select greatest(1,2,6,7) from dual;

分组group by where:记录筛选 having:分组筛选

查询每个部门的平均工资

select department\_id,avg(salary) from employees;//报错 这是整个企业的工资

select department\_id,avg(salary) from employees

group by department\_id;

2

创建一个表

create table hr.st(name varchar(20),subject varchar(20),score int);

利用sqlldr工具进行加载

控制文件如下:

load

infile 'D:\大三下作业\实用数据库\第三周\st.csv'

into table hr.st

fields terminated by ','

( name char,

subject char,

socre integer external)

命令行执行

sqlldr hr/hr control = 'D:\大三下作业\实用数据库\第三周\st.ctl'(该句命令是在dos下运行，不用登陆数据库)

行转列

select name,sum(decode(subject,'Chinese',score,0)) Chinese,

sum(decode(subject,'Math',score,0)) Math,

sum(decode(subject,'English',score,0)) English

from st

group by name;

create table t(id int);

向表里插入10条记录

begin

for i in 1..10 loop

insert into t values(i);

end loop;

end;

/

查询数据字典

desc user\_procedures

select object\_name,object\_type from user\_procedures

where object\_type='PROCEDURE';

desc user\_source

select text from user\_source where name='P2';

execute p2;

call p2();

PL/SQL

块类型

1 匿名块

declear

begin

exception

end

2 命名块

create procedure XXX

as

begin

exception

end

eg:

create procedure p2

as

begin

for i in 1..10 loop

insert into t values(i);

end loop;

end;

/

块结构

1 声明部分：定义。。。（可选）

2 可执行部分（必须要有的）

3 异常处理部分 （可选）

循环

1 无条件循环

2 条件循环

3 固定次数循环

select a.id,b.id from t a,t b

where a.id+5=b.id;

子查询

1 嵌套子查询

2 关联子查询

多行操作符 in any all

hr.employees last\_name

条件：普通群众

1) select last\_name from employees

where employee\_id in (select manager\_id from employees);//领导

select last\_name from employees

where employee\_id not in (select nvl(manager\_id,0) from employees);

2) select last\_name from employees e where not exists (select 'X' from employees where e.employee\_id=manager\_id);

last\_name,salary,department\_id

条件：高于其部门的平均工资

利用关联子查询

之前的方法：

select a.last\_name,a.salary,a.department\_id

from employees a,

(select department\_id,avg(salary) salavg from employees

group by department\_id) b

where a.department\_id = b.department\_id and a.salary>b.salavg;

关联子查询：

select last\_name,salary,department\_id

from employees e

where salary>(select avg(salary) from employees

where e.department\_id = department\_id);

处理多个结果集

4个操作符：

1 union 重复的取一次

2 union all 重复保留

3 intersect 保留重复的

4 minus 去掉重复的

select employee\_id,job\_id

from employees

union

select employee\_id,job\_id

from job\_history

/

column XXX列名 nopint 隐藏这一列

insert (子查询)

scott:

create table emp\_hz as select empno,ename,sal

from emp where deptno = 30;

create table emp\_gz as select \* from emp\_hz where 1=2;

insert into emp\_gz values(1234,'Tom',2500) ;

insert into emp\_gz values(7900,'James',4500) ;

SCOTT@db18c>select \* from emp\_gz;

EMPNO ENAME SAL

---------- -------------------- ----------

1234 Tom 2500

7900 James 4500

merge into emp\_hz h

using emp\_gz g

on (h.empno=g.empno)

when matched then

update set

h.ename=g.ename,

h.sal=g.sal

when not matched then

insert values

(g.empno,g.ename,g.sal);//不匹配就插进去

SCOTT@db18c>select \* from emp\_hz;//这是汇总表

EMPNO ENAME SAL

---------- -------------------- ----------

7499 ALLEN 1600

7521 WARD 1250

7654 MARTIN 1250

7698 BLAKE 2850

7844 TURNER 1500

7900 James 4500

1234 Tom 2500

alter 语句做完之后会自动提交

提交：显式commit 隐式。。。

默认值

锁

检测死锁，将产生死锁的语句回滚。

v$lock

select sid,block from v$lock where block=1;//检查死锁位置,找到会话 得到一个sid

v$session

select sid,serial#,username.from v$session

where sid = XXX;//会话的信息 XXX为上面得到的sid 得到一个sid,serial#

alter system kill session 'xxx,xxx'; //杀掉进程（上面得到的sid和serial#)

表

表名需以字母开头

序列

索引

同义词

在创建数据库的时候要选择字符集

DDL语句

**修改表**

1 修改表名

rename buyers to customers;

2 增加字段

alter table add birthdate date;

3 修改字段

1）改名

alter table c rename column birthdate to birth\_date;

2）修改类型

alter table c modify birth\_date char(8);

要更改数据类型时，则要修改的列必须为空。

3）修改宽度

alter table c modify buyer\_name varchar(30);

4 删除字段

1）直接删除列

alter table c drop column birth\_date;

2）首先把删除的列标记为未使用，然后再删除标记为未使用的列

alter table c set unused column birth\_date;

alter table c drop unused column;

5 截断表

truncate table c;清空表

6 修改表为只读

alter table c read only;

7 修改表为读写

alter table c read write;

8 删除表

drop table c；

drop table c purge;

9 给表加注释

comment on table product is 'This is a test table';

select comments from user\_table\_comments

where table\_name = 'PRODUCT';//查看表的注释

给列加注释

comment on column XX is ' ';

查找列的注释

select comments from user\_column\_comments

where table\_name = 'PRODUCT'

and column\_name = 'PROD\_ID';

SQL

1 QL(select)

2 DML(insert,update,delete,merge)

3 TCL(commit,rollback,savepoint)

4 DCL(grant,revoke)

5 DDL(create,alter,drop,rename,truncate,comment)

数据完整性

数据+规则------>保证数据有效合理

1

create table test(id int,name varchar(20));

insert into test values(1,'Tom');

insert into test values(3,'Jerry');

为表添加一个unique约束

alter table test add constraint uni\_name unique(name);

insert into test values(2,'Tom');//报错 图3221

去掉约束

alter table test disable constraint uni\_name;

insert into test values(2,'Tom');//成功

使用exceptions表

select count(\*) from exceptions;//报错：表不存在 图3222

@?/rdbms/admin/utlexcpt.sql;//?=oracle+

ALTER TABLE test

ENABLE CONSTRAINT uni\_name

EXCEPTIONS INTO exceptions;//报错：图3223

查询哪个违反了约束

SELECT rowid,id,name

FROM test

WHERE rowid IN(SELECT row\_id FROM exceptions);

更新该行的数据

UPDATE hr.test set name='Sehun'

WHERE rowid = ' ';//rowid由上条命令得到；

ALTER TABLE test

ENABLE CONSTRAINT uni\_name;

TRUNCATE TABLE exceptions;

在数据字典中查询约束信息

(user\_constraints,user\_cons\_columns)

2 视图

虚表——不是存储结构，只是语句定义

表——存储结构

视图作用

1）

2）简化查询

3）

4）

视图分类 1）简单视图 2）复杂视图

创建视图

create view emp\_info as select empno,ename,sal from emp;

（授权：在sys用户下:grant create view to scott;）

修改视图

create or replace view emp\_info as

select empno,ename,sal,deptno from emp;

create or replace view emp\_info as

select empno,ename,sal,deptno from abc;

(create table abc as select\* from emp;)

create or replace view emp\_info as

select empno,ename,sal,deptno from abc

where deptno =30;(都可以改！)

create or replace view emp\_info as

select empno,ename,sal,deptno from abc

where deptno =30 with check option;(with check option就是deptno字段不允许更改)

create or replace view emp\_info as

select empno,ename,sal,deptno from abc

where deptno =30 with read only;(只读视图啥也不能改)

改视图就是改表

rownum 伪列 由结果排列

select rownum,ename from (select rownum aa,emp.\* from emp)

where aa>=10;

select rownum,ename from emp where rownum>=10;//报错：第一行小于10，不会向下继续寻找

3

管理例程

1）关闭例程（shutdown四种模式）

正常关闭 normal （shutdown就是shutdown normal）

事务性关闭 transactional

立即关闭 immediate

终止退出 abort

2）启动 3个阶段

例程启动（nomount）

startup nomount

条件：需要正确初始化参数文件

只可以访问一部分的动态性能视图（访问内容来自内存的动态性能视图，来自数据文件的不能访问）

加载数据库（mount）

条件：需要访问控制文件.CTL

alter database mount;

可以访问所有的动态性能视图

打开数据库（open）

条件：需要访问联机重做日志文件和数据问价

alter database open;

可以访问所有数据

以只读方式打开数据库

select name,open\_mode from v$database;

以受限模式打开数据库

startup restrict

select INSTANCE\_NAME,LOFINS from v$instance;

grant RESTRICTED SESSION to scott;

。。。

4

配置例程

初始化参数文件

一 文本文件 pfile init<sid>.ora ----initdb18c.ora

二 二进制文件 spfile spfilr<sid>.ora ----spfiledb18c.ora

查看初始化参数

show parameter

show parameter shared\_pool\_size

show parameter share

修改初始化参数

alter system set （shared\_pool\_size=128m）

手工创建数据库（demo)

1 创建oracle服务oracleservicedemo

oradim -new -sid demo

2 将当前例程设为demo

set oracle\_sid=demo

set oracle\_sid//查询会话id

3 创建/编辑初始化参数文件（initdemo.ora）

4 创建相应的目录结构

5 启动例程（demo）

startup nomount

6 执行创建数据库的语句

create database demo

datafile 'D:\APP\ORACLE\ORADATA\DEMO\SYSTEM01.DBF'size 400m

sysaux datafile 'D:\APP\ORACLE\ORADATA\DEMO\SYSAUX01.DBF'size 400m

undo tablespace UNDOTBS1 datafile 'D:\APP\ORACLE\ORADATA\DEMO\undotbs01.DBF'size 50m

default temporary tablespace temp tempfile 'D:\APP\ORACLE\ORADATA\DEMO\temp01.DBF'size 20m

logfile

group 1 ('D:\APP\ORACLE\ORADATA\DEMO\redo01.log') size 10m,

group 2 ('D:\APP\ORACLE\ORADATA\DEMO\redo02.log') size 10m,

group 3 ('D:\APP\ORACLE\ORADATA\DEMO\redo03.log') size 10m;

控制文件（创建数据库时会自动创建控制文件）

联机重做日志文件 一个数据库至少需要两个日志文件文件

数据文件 创建数据库时，至少创建3个表空间sysytem,sysaux,undo,tablespace)

7 创建数据字典视图

SYS@demo>@?/rdbms/admin/catalog

8 创建spfile

create spfile from pfile;

9 创建口令验证文件

C:\>orapwd file=D:\app\oracle\product\18.3.0\database\PWDdemo.ora password=admin1#3

10 创建oracle内部包

SYS@demo>@?/rdbms/admin/catproc

11 创建scott方案

SYS@demo>@?/rdbms/admin/scott

SCOTT@demo>alter user scott identified by tiger;

12 加载产品用户概要文件信息

SYSTEM@demo>@?/sqlplus/admin/PUPBLD.SQL(system的密码为19981113)

13 配置监听器和服务名

14 加载em expres

管理控制文件 记录整个数据库的状态

1 查看控制文件的名称和位置

show parameter control\_files select name from v$controlfile;

2 控制文件的内容 二进制转成文本文件

oradebug （改语言：alter session set nls\_language=english;）

oradebug setmypid

oradebug dump controlf 3//报错

oradebug tracefile\_name

3 控制文件多路复用

1) 修改初始化参数

alter system set control\_files='D:\APP\ORACLE\ORADATA\DEMO

\CONTROL01.CTL','D:\APP\ORACLE\ORADATA\DEMO

\CONTROL02.CTL','D:\NIT\CONTROL03.CTL' scope=spfile;

2) shutdown immediate

3) 复制出第三个控制文件

4) startup

5) 检查控制文件

select name from v$controlfile;

4 重建控制文件

1）alter database backup controlfile to trace as 'd:\nit\create\_controf.sql';

2）shutdown immediate

3）删除控制文件

4）startup

5）执行创建控制文件的语句

startup nomount

6）alter database open;

管理日志文件

1 日志文件的作用 恢复

2 日志工作原理

3 数据库日志模式

1) 非存档模式

2) 存档模式（归档模式）

alter system switch logfile;//切换当前日志文件

3 添加日志文件组

日志文件组：一个数据库至少两个组，每个组至少一个文件称为成员

alter database ass logfile group 4

('D:\app\oracle\oradata\DEMO\redo04a.log') size 10m;

4 查看日志组信息（v$log）

desc v$log

select group#,sequence#,status from v$log;

5 查看日志成员信息(v$logfile)

select group#,member from v$logfile;

6 添加日志成员信息

alter database add logfile member 'D:\app\oracle\oradata\DEMO\redo04b.log' to group 4;

7 删除日志文件组

组的状态（current,active inactive，unused）

alter database drop logfile group n;n为数字

8 删除日志文件成员

alter database drop logfile member 'D:\APP\ORACLE\ORADATA\DEMO\REDO01.LOG';

报错：不可以删除组中的最后一个成员

alter system switch logfile;当前使用的日志文件不能删也要使用该句

9 OMF（oracle manage file：oracle管理文件）

配置相应的初始化参数来实现OMF

show parameter db\_create

show parameter db\_create\_file\_dest

create tablespace users;//报错 需要指定路径与大小

drop tablespace users;//需要手动删除文件

alter system set db\_create\_file\_dest='D:\APP\OMF';//指定路径

create tablespace users;//不再报错

drop tablespace users;//同时会删除文件

日志文件自动OMF：

alter database add logfile;//默认在闪回区

show parameter db\_create

配置相应的初始化参数实现日志文件的OMF

alter system set db\_create\_file\_dest\_1='D:\APP\OMF\log1';

10 清除日志文件内容

alter database clear logfile group n; alter database clear logfile ' ';具体的成员

11 修改数据库的日志模式

在mount阶段做：

调整为mount阶段：

shutdown immediate

startup mount(最后的错误解决)

alter database archivelog;//改为非存档模式

archive log list//查询当前信息 图3292

alter database noarchivelog;//改为存档模式

12 设置归档日志目的地

show parameter log\_archive dest;

alter system archive log current;

默认归档日志存储在闪回区

13 查看归档日志文件信息（v$archived\_log）

select name from v$archived\_log

alter system set log\_archive\_dest\_!='D:\app\arch1';//错误：需要指定时本地location还是远程service

alter system set log\_archive\_dest\_!='location=D:\app\arch1';

alter system archive log current;//做一次归档

14 日志文件移动/重命名

select member from v$logfile;

1) 允许移动（不正在被使用，不在Open）

shut immediate

2) 利用操作系统移动并重命名

将文件手动移动到d:\app\oracle\oradata\demo

手动重命名

startup//报错：找不到日志文件

3) 更新控制文件

alter database rename file ' AAA' to 'd:\app\oracle\oradata\demo\XXX';

//XXX改后的名字，AAA为之前select member from v$logfile;的路径

alter database open;

select group#,sequence#,status from v$log;

select group#,member from v$logfile;

15 处理日志文件丢失

1) 非当前的日志文件丢失 shutdown immediate

手动删除文件 startup//报错 startup mount;

alter database clear logfile group 4; alter database open;

2) 当前日志文件丢失

做一个更改操作 update.... 提交 commit shutdown immediate

手动删除文件 startup mount;

alter database clear logfile group 1;//假设当前文件为1组的//报错

recover database until cancel;

alter database open resetlogs;

startup mount

alter database clear logfile group 1;

管理表空间和数据文件

1 数据库存储的结构层次

表空间------数据文件

段（存储结构） 区（oracle最小的空间分配单位）

块（oracle最小的IO单位） 表

查询表空间

select name from v$tablespace

2 创建users表空间，并设为数据库默认的表空间

create tablespace users datafile 'D:\APP\ORACLE\ORADATA\

DEMO\USERS01.DBF' size 30m;

alter database default tablespace users;

3 创建一个由2K的块组成的表空间smalltbs

4 表空间的空间管理（区的管理）

1）本地管理（有关区可用或不可用的信息存储在数据文件）

2）数据字典管理（有关区可用或不可用的信息存储在数据字典）

5 表空间类型

1）永久（常规）表空间

2）撤销表空间（存放回滚数据）

3）临时表空间（用来排序）

select tablespace\_name,contents from dba\_tablespaces;

**创建撤销表空间**undotbs2,并设为数据库当前的撤销表空间

show parameter undo\_tablespace//查询当前的撤销表空间

create undo tablespace undotbs2 datafile D:\APP\ORACLE\ORADATA\

DEMO\UNDO02.DBF' size 20m;

alter system set undo\_tablespace=undotbs2;

**创建永久表空间**

create tempprary tablespace temp2 tempfile D:\APP\ORACLE\ORADATA\

DEMO\temp02.DBF' size 20m;

alter database

6 表空间状态

1）联机可读写 2）只读 3）脱机（表空间在，但是不可以打开）

select tablespace\_name,status from dba\_tablespace;

alter tavlespace smalltbs read only;

不能脱机的表空间: 1）system 2）当前撤销表空间 3）临时表空间

alter tablespace xxx outline;//使表xxx脱机

alter tablespace xxx online;//

7 删除表空间

drop tablespace xxxx; select name from v$datafile;

8 ONF(oracle管理文件)

show parameter db\_create; 初始化参数dn\_create\_file\_dest

9 扩展表空间

1）扩展原数据文件大小 a.自动扩展 b.手动扩展 2）添加新的数据文件

10 移动或重命名数据文件

select name from v$datafile;

Alter database move datafile '\demo\Userdata01.dbf' to '\omf\data01.dbf';//系统文件也被移走了

Select name from v$datafile;

Alter database move datafile '\demo\Userdata01.dbf' to '\omf\data01.dbf' keep；//源文件不动，相当于copy

oracle安全

安全3A

**1 验证**

a.sys

1）操作系统验证

2）口令文件验证

（1）sqlplus / as sysdba

（2）检查初始化参数remote\_login\_passwordfile

a）NONE（禁用口令文件验证）

b）EXCLUSIVE（单例程多用户）v$pwfile\_data

c）SHARED（多例程多用户）

（3）创建口令验证文件

orapwd file=D:\app\oracle\product\18.3.0\database\PWDdemo.ora password=admin1#3 force=y（强制覆盖）

（4）检查D:\app\oracle\product\18.3.0\network\admin\sqlnet.ora文件

默认：SQLNET.AUTHENTICATION\_SERVICES = (NTS)

修改为：SQLNET.AUTHENTICATION\_SERVICES = (NONE)//操作系统不能再直接登陆，需要口令验证

（5）sqlplus / as sysdba 连不上

（6）sqlplus sys/admin1#3 as sysdba 才可以

（7）sqlplus scott/tiger as sysdba

（8）select username,sysdba from v$pwfile users;

（9）grant sysdba to scott;

（10）conn scott/tiger as sysdba;

b.non-sys sqlplus /nolog

1）数据库验证

desc user$ create user demo1 identified by admin;

2）操作系统验证

（1）检查初始化参数os\_authent\_prefix

show parameter os show parameter os\_authent\_prefix//默认值为OPS$

（2）创建操作系统用户os1

（3）创建对应的数据库用户

create user OPS$os1 identified externally;//externally表示外部用户验证

（4）赋予用户登陆权限

grant create session OPS$os1;

（5）修改注册表 （6）以操作系统用户os1登陆 runas /user:os1 cmd (7）sqlplus /

**2 授权**

1 系统权限 2 对象权限

3 权限传递规则 1）对象权限是连带的 2）系统权限不连带

4 角色（权限的集合）

1）角色的左右 （1）简化权限管理 （2）动态权限管理（角色处于激活状态时有效）

默认角色在登陆时激活,非默认角色使用set role 命令激活

带口令激活（set role r2 identified by r2;）

2）用户自定义角色 3）预定义角色

4）应用程序角色（不需要指派给用户）

create role app\_r1 identified using scott.setrole;

grant select on scott.emp to app\_r1;

create or replace procedure scott.setrole

authid current\_user

as

begin

dbms\_session.set\_role('app\_r1');

end;

/

grant execute on scott.setrole to a,b;

execute scott.setrole;

create role r1;

grant create table,select on scott.emp to r1;

SYS:

grant r1 to a;

A:

grant r1 to b;

sys:

revoke r1 from a;

B:

create role r1;

create role r2;

grant r1 to a ;

grant r2 to b;

alter user a default role none;

set role r1;

select any table //访问任何一张表：系统权限

select on scott.emp //局部

权力（全局，中心点是用户） 权限（局部，资源）

create user a identified by a;

create user b identified by b;

grant create session to a,b;

grant create table to a with frant option;//报错 应with admin option

**3 审核**

1 默认审核（强制审核）

2 标准数据库审核

1）启用审核（初始化参数）

show parameter audit\_trail

alter system set audit\_trail=bzd scope=spfile;//报错 图491 none：禁止审核

2）指定审核选项

（1）审核用户（审核权限） audit select any table by scott;

（2）审核对象 audit delete on scott,emp;

（3）审核语句 audit create tigger;

审核成功的删除操作

noaudit delete on scott,emp;//取消审核

audit delete on scott,emp whenever successful;

审核失败的登陆

audit session whenever not successful;

一次会话中相同操作审核一次

audit update on scott.emp by session;//产生一次审核记录

操作一次审核一次

audit update on scott.emp by access;

完整操作：

show parameter audit\_trail

select count(\*) from aud$;

audit select,insert,update,delete on scott.emp by access;

select count(\*) from aud$;

查询审核记录：

desc dba\_audit\_trail;

alter session set nls\_date\_format='yyyy-mm-dd hh24:mi:ss';

select timestamp,username,action\_name from dba\_audit\_trail;

col username for a20

col action\_name for a 20

select timestamp,username,action\_name from dba\_audit\_trail;

3 基于值的审核（通过触发器实现）

create table sal\_change(ENAME VARCHAR2(10),OLD\_SAL NUMBER(7,2),NEW\_SAL NUMBER(7,2));

通过触发器记录员工的工资变化情况

create or replace trigger scott.tr2

after update on scott.emp referencing old as o new as n for each row

begin

insert into sal\_change values(:o.ename,:o.sal,:n.sal);

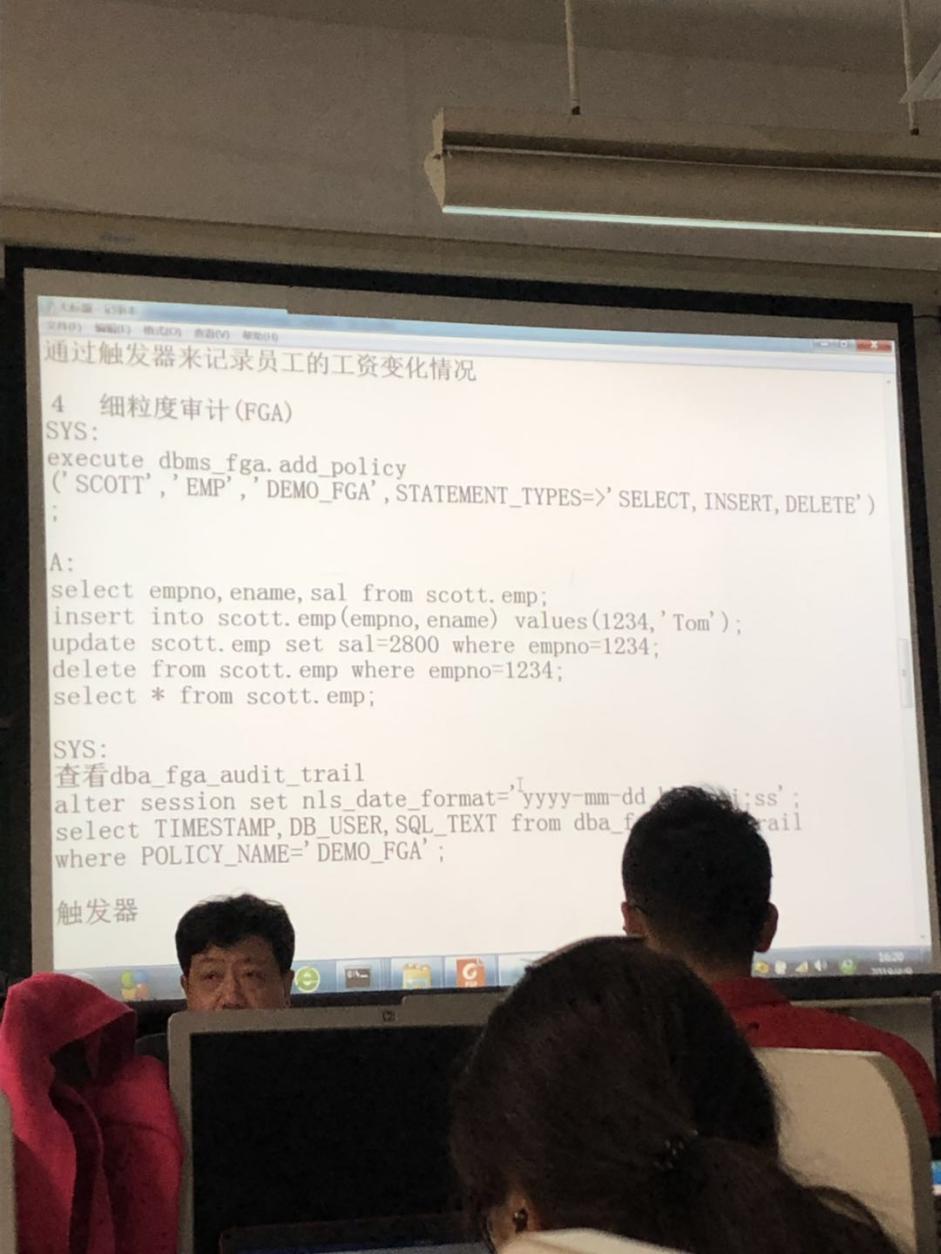
end;

/

图492

删除触发器 drop trigger scott.tr2

4 细粒度审计 FGA



**触发器** 一段可以自动调用的代码

1 DML触发器

scott:

create table tr\_test1(a varchar(20));

select \* from tr\_test1;

create or replace trigger scott.tr1

after update on scott.emp

begin

insert into scott.tr\_test1 values('Be Changed!');

end;

/

update emp set sal=3700 where empno=7300;

select \* from tr\_test1;

update emp set sal=3700 where mgr=7698;

select \* from tr\_test1;//更新5行，此时算触发一次

更改：

create or replace trigger scott.tr1

after update on scott.emp for each row

begin

insert into scott.tr\_test1 values('Be Changed!');

end;

/

for each row:逐行触发

update emp set sal=3700 where mgr=7698;

select \* from tr\_test1;//更新5行，增加5条记录

更改：

create or replace trigger scott.tr1

after update on scott.emp referencing old as o new as n for each

row when (n.sal>5000)

begin

insert into scott.tr\_test1 values('Be Changed!');

end;

/

old as o new as n for each row when (n.sal>5000)：更改的新值应该大于5000

update emp set sal=2800 where mgr=7698;//不会增加记录

update emp set sal=5400 where mgr=7698;

select \* from tr\_test1;//增加5条 12条

2 系统触发器

create table scott.logon\_rec(username varchar(20),logon\_time date);

SYS:

grant administer database trigger to scott;

create or replace trigger scott.tr3

after logon on database

begin

insert into scott.logon\_rec values(user,sysdate)

end;

/

alter session set nls\_date\_format='yyyy-mm-dd hh24:mi:ss';

select \* from scott.logon\_rec;

销钉存储过程（scott.p1)

execute dbms\_shared\_pool.keep('SCOTT.P1');

v$db\_object\_cache

查询被销钉的存储过程

select owner,name,type from v$db\_object\_cache

where owner<>'SYS' and type='PROCEDURE' and kept='YES';

查看当前用户下的存储过程

select object\_name,object\_type from user\_procedures

where object\_type='PROCEDURE';

4 细粒触发器

FGA：细粒度审计

导入导出

exp/imp

1 交互模式

2 命令模式

exp scott/tiger file=emp2.dmp tables=emp

3 获取帮助

exp help=y

4 可以导出的对象

1）数据库 full=y

2）表空间 TABLESPACES=users

3）方案 OWNER=scott

4）表 TABLES=emp.dept

5 导出表的子集

6 利用参数文件导出

userid=scott/tiger

file=d:\exp\para.dmp

tables=emp

query='where sal>2500'

保存为文件p1.par

exp parfile=p1.par

imp

1 IGNORE=y

2 fromuser,touser

练习：将scott.emp中工资大于2500的记录导出，不导出约束，然后导入到hr.emp

exp scott/tiger file=emp\_2500.dmp tables=emp query='where

"sal>2500"' constraints=n

imp hr/hr file=emp\_2500.dmp tables=emp fromuser=scott touser=hr

################################################################

数据泵

expdp/impdp

1 expdp help=y

2 创建目录对象

create directory expdir as 'd:\exp';

3 赋予相应的权限

grant read,write on directory expdir

REMAP\_TABLE=emp:emp1

REMAP\_SCHEMA=scott:hr

REMAP\_TABLESPACE=tbs1:tbs2

练习：

利用expdp工具导出scott.emp,不导出约束，然后利用impdp工具导入到hr.emp1，

并且更换表空间

expdp scott/tiger directory=expdir dumplile=empdp.dmp tables=emp

exclude=constraint

impdp hr/hr directory=expdir dumplile=empdp.dmp

REMAP\_SCHEMA=scott:hr

REMAP\_TABLE=emp:emp1

REMAP\_TABLESPACE=users:sales

################################################################

备份和恢复

1 利用LogMiner（日志挖掘机）重做日志文件

（1）启用数据库补充日志

select SUPPLEMENTAL\_LOG\_DATA\_MINA FROM v$database;

alter database add SUPPLLEMENTAL LOG DATA;//启用

（2）创建数据字典文件

create directory dict as 'd:\dict';//创建目录

execute dbms\_lomnr\_d.build('v186dict.ora','DICT');

DICT:逻辑文件目录

（3）开始一个事务

scott: update emp set sal=3800 where emp=7369; commit;

（4）添加需要分析的日志文件

select group#,status from v$log;

select group#,member from v$logfile;//得到当前日志文件的路径

execute dbms\_logmnr.add\_logfile

(LogFileName=>'上面得到的当前日志文件的路径',Option=>dbms\_logmnr.new)

（5）启动分析

execute dbms\_logmnr.start\_logmnr(DictFileName=>'d:\dict\v186dict.ora')

（6）查询

v$logmnr\_contents

alter session set nls\_data\_format='yy-mm-ss hh24:mi:ss';

select timestamp,sql\_redo from v$logmnr\_contents

where seg\_name='EMP' and operation='UODATE'；

2 RMAN（恢复管理器）

冷备份（MOUNT）

热备份（OPEN）——归档模式下