

1 事务与并发控制

事务是数据操作的逻辑单位，是用户定义的一组操作序列，由一条或多条相关 SQL 语句组成，是数据库应用程序的基本逻辑单位。

事务就是一组不可分割的 SQL 语句，在执行时要么全部成功，要么全部撤销，Oracle 就是通过事务来确保数据库中数据的一致性的。Oracle 提供的事务控制语句包括提交事务、回退事务、设置保存点、回退到保存点、设置事务的属性等等。

(1) 提交事务

用在前面实验中自己的用户名登录，将表 table1 的所有操作权限赋予 test 用户，然后通过以下实例实现事务的提交。

```
C:\Users\JiangXue>sqlplus/nolog
SQL*Plus: Release 11.1.0.7.0 - Production on 星期一 2月 27 15:14:13 2012
Copyright (c) 1982, 2008, Oracle. All rights reserved.

SQL> conn sys/ora505 as sysdba
已连接。
SQL> select * from table1;

      ID NAME
-----
1 张三
1 张三
2 李四
3 王五
5 Lincon

SQL> grant all on table1 to test;
授权成功。

SQL> insert into table1 values(6,'Abraham');
已创建 1 行。
```

sys 用户向表 table1 中插入一行值，其中 id 为 6，未提交。然后打开另一个窗口，通过具有 table1 操作权限的 test 用户连接数据库，查询表 table1 的数据。可以看到，sys 插入的新值并没有在结果中显示出来。

```

C:\Users\JiangXue>sqlplus/nolog

SQL*Plus: Release 11.1.0.7.0 - Production on 星期一 2月 27 15:36:25 2012

Copyright (c) 1982, 2008, Oracle. All rights reserved.

SQL> conn test/test
已连接。
SQL> select * from sys.table1;

      ID NAME
-----
1 张三
1 张三
2 李四
3 王五
5 Lincon

```

在 sys 用户连接窗口中输入 commit，具体如下所示。

```

SQL> commit;

提交完成。

```

当 sys 提交完成后，test 用户查看 table1 中的 id 为 6 的行，可以查询到 sys 向表新插入的值，具体如下所示。

```

SQL> select * from sys.table1;

      ID NAME
-----
1 张三
1 张三
2 李四
3 王五
5 Lincon
6 Abraham

已选择6行。

```

(2) 回退全部事务

```
SQL> insert into table1 values(7,'Beck');
```

已创建 1 行。

```
SQL> delete from table1 where id=6;
```

已删除 1 行。

```
SQL> select * from table1;
```

ID	NAME
1	张三
1	张三
2	李四
3	王五
5	Lincon
7	Beck

已选择6行。

```
SQL> rollback;
```

回退已完成。

```
SQL> select * from table1;
```

ID	NAME
1	张三
1	张三
2	李四
3	王五
5	Lincon
6	Abraham

已选择6行。

上述操作首先向表 table1 中插入一条记录，然后删除 id 为 6 的记录，最后回退事务。通过查询可以看到，这两条操作都被回退了。回退到 commit 位置。

(3) 回退部分操作

当一个事务包含多条语句时，可以只回退几条语句，具体如下所示。

```

SQL> delete from table1 where id=6;

已删除 1 行。

SQL> savepoint pt1;

保存点已创建。

SQL> update table1 set name = 'Zesx' where id=5;

已更新 1 行。

SQL> select * from table1;

      ID NAME
-----
      1 张三
      1 张三
      2 李四
      3 王五
      5 Zesx

SQL> rollback to pt1;

回退已完成。

```

通过设置保存点，可以选择回退到具体位置。上述设置保存点为 **pt1**，且回退到该保存点。这时回退操作只回退该保存点之后的操作，而不会回退保存点之前的操作。因此查询表 **table1** 时，**id** 为 6 的记录肯定不存在表中了。具体如下所示。

```

SQL> select * from table1;

      ID NAME
-----
      1 张三
      1 张三
      2 李四
      3 王五
      5 Lincon

```

(4) 设置事务的属性

事务有是否可读写、只读、隔离级别等属性，下面举例说明如何设置事务的属性。

在 **test** 用户连接数据库的窗口执行以上操作，发现当设置事务为只读后，向表 **empinf** 插入数据会引发错误，因为事务是只读的。

```
SQL> commit;
提交完成。

SQL> set transaction read only;
事务处理集。

SQL> desc empinf;
 名称                                     是否为空? 类型
-----
ID                                     NOT NULL NUMBER(38)
NAME                                     VARCHAR2(20)
SAL                                     NUMBER(8,2)
JOB                                     VARCHAR2(10)
DEPT                                    VARCHAR2(20)

SQL> insert into empinf values(8,'Holon',3000,'clerk','财政部');
insert into empinf values(8,'Holon',3000,'clerk','财政部')
      *
第 1 行出现错误:
ORA-01456: 不能在 READ ONLY 事务处理中执行插入/删除/更新操作
```

2 进程检测

任何一个连接都会用到数据库的后台进程，一般常用的后台进程有 DBWR、LGWR、SMON、PMON、CKPT、ARCH、RECO 等。DBWR 为数据库写入进程，CKPT 为检查点进程，ARCH 用于将联机重做日志归档，LGWR 为日志写入进程。下面通过查看后台进程、连接等来实现进程的监测。

(1) 监测后台进程

后台进程可以通过动态性能视图 v\$bgsprocess 来查看，运行情况如下所示。

```
SQL> conn sys/ora505 as sysdba
已连接。
SQL> desc v$bgsprocess;
 名称                                     是否为空? 类型
-----
PADDR                                     RAW(4)
PSERIAL#                                 NUMBER
NAME                                     VARCHAR2(5)
DESCRIPTION                             VARCHAR2(64)
ERROR                                    NUMBER
```

在查看动态视图时，首先可以通过上述命令查看动态视图中包含的列，以及各列的属性。DESCRIPTION 列太长，因此查询结果表示后不方便读者浏览。首先这是该列的格式，然后再查看。运行情况如下：

```
SQL> column description format a20
SQL> select * from v$bgprocess where paddr <>'00';
```

PADDR	PSERIAL#	NAME	DESCRIPTION	ERROR
296DBDBC	1	PMON	process cleanup	#####
296DC8A4	1	UKTM	Virtual Keeper of Ti Me process	#####
296DD38C	1	DIAG	diagnosibility proce ss	#####
296DDE74	1	DBRM	Resource Manager pro cess	#####
296DE95C	1	PSP0	process spawner 0	#####
296DF444	1	DIA0	diagnosibility proce ss 0	#####
296DFF2C	1	MMAN	Memory Manager	#####
296E0A14	1	DBW0	db writer process 0	#####
296E14FC	1	LGWR	Redo etc.	#####
296E1FE4	1	CKPT	checkpoint	#####
296E97DC	3	FBDA	Flashback Data Archi ver Process	#####
296E2ACC	1	SMON	System Monitor Proce	#####
296F308C	2	SMCO	Space Manager Proces s	#####
296E35B4	1	RECO	distributed recovery	#####
296F25A4	4	CJQ0	Job Queue Coordinato	#####

从上述操作可以看到，paddr 不为'00'的各个后台进程。

(2) 检测会话

```
SQL> column username format a20
SQL> column program format a20
SQL> select sid,serial#,username,program,status from v$session;
```

SID	SERIAL#	USERNAME	PROGRAM	STATUS
108	189	DBSNMP	emagent.exe	INACTIVE
131	172		ORACLE.EXE (CJQ0)	ACTIVE
132	31	DBSNMP	emagent.exe	ACTIVE
133	179	DBSNMP	emagent.exe	INACTIVE
135	12		ORACLE.EXE (SMCO)	ACTIVE
136	12	SYSMAN	OMS	INACTIVE
137	2		ORACLE.EXE (q000)	ACTIVE
139	2274		ORACLE.EXE (W000)	ACTIVE
140	4	SYSMAN	OMS	INACTIVE
142	7	SYSMAN	OMS	INACTIVE
145	14	SYSMAN	OMS	INACTIVE

SID	SERIAL#	USERNAME	PROGRAM	STATUS
146	13		ORACLE.EXE (q001)	ACTIVE
147	212	SYS	sqlplus.exe	ACTIVE
148	1	SYSMAN	OMS	INACTIVE
152	4		ORACLE.EXE (FBDA)	ACTIVE
153	18		ORACLE.EXE (QMNC)	ACTIVE
155	9	SYSMAN	OMS	ACTIVE
156	1		ORACLE.EXE (MMNL)	ACTIVE
157	1		ORACLE.EXE (MMON)	ACTIVE
158	1		ORACLE.EXE (RECO)	ACTIVE
159	1		ORACLE.EXE (SMON)	ACTIVE
160	1		ORACLE.EXE (CKPT)	ACTIVE

SID	SERIAL#	USERNAME	PROGRAM	STATUS
161	1		ORACLE.EXE (LGWR)	ACTIVE
162	1		ORACLE.EXE (DBW0)	ACTIVE
163	1		ORACLE.EXE (MMAN)	ACTIVE
164	1		ORACLE.EXE (DIA0)	ACTIVE
165	1		ORACLE.EXE (PSP0)	ACTIVE

会话信息在 v\$session 中保存，因此通过上述查询，得到连接到数据库的会话情况。其中 sid 表示会话的 id 号，serial# 表示会话的序列号，与 sid 一起用来标识一个会话，username 表示建立该会话的用户名，program 表示这个会话使用什么工具连接到数据库，status 表示当前这个会话的状态。

上述结果中，列 username 为空的记录是 Oracle 数据库的后台进程。

3 文件监测

(1) 监测控制文件

控制文件非常重要，一般需要进行备份。连接数据库时，当完成装载后，需要读取控制文件中记录的数据文件、日志文件的位置等信息。只有控制文件完好无损，才可以打开数据库。

```
SQL> select value from v$parameter where name='control_files';
```

VALUE
D:\ORACLE\ORADATA\ORCL\CONTROL01.CTL, D:\ORACLE\ORADATA\ORCL\CONTROL02.CTL, D:\ORACLE\ORADATA\ORCL\CONTROL03.CTL

从上述可以看到，本数据库系统有三个控制文件，其存放位置如上所示。这三个控制文件的内容是完全相同的，当其中一个被破坏时，可以通过其他的来代替该控制文件。数据库在启动时，只要能找到一个可用的控制文件，就可以成功地打开数据库。

(2) 监测数据文件

数据文件用以存放用户的数据，打开数据库后用户就可以对数据库进行操作，一般都会涉及数据文件，因此数据库管理员需要检测数据文件，具体如下：

```
SQL> desc dba_free_space
```

名称	是否为空?	类型
TABSPACE_NAME		VARCHAR2(30)
FILE_ID		NUMBER
BLOCK_ID		NUMBER
BYTES		NUMBER
BLOCKS		NUMBER
RELATIVE_FNO		NUMBER

```
SQL> desc dba_data_files
```

名称	是否为空?	类型
FILE_NAME		VARCHAR2(513)
FILE_ID		NUMBER
TABSPACE_NAME		VARCHAR2(30)
BYTES		NUMBER
BLOCKS		NUMBER
STATUS		VARCHAR2(9)
RELATIVE_FNO		NUMBER
AUTOEXTENSIBLE		VARCHAR2(3)
MAXBYTES		NUMBER
MAXBLOCKS		NUMBER
INCREMENT_BY		NUMBER
USER_BYTES		NUMBER
USER_BLOCKS		NUMBER
ONLINE_STATUS		VARCHAR2(7)

```
SQL> select b.file_name 物理文件名, b.tablespace_name 表空间, b.bytes/1024/1024
大小M,
2 (b.bytes-sum(a.bytes))/1024/1024 已使用M,
3 substr((b.bytes-sum(a.bytes))/b.bytes*100,1,5) 使用率
4 from dba_free_space a, dba_data_files b
5 where a.file_id = b.file_id
6 group by b.tablespace_name, b.file_name, b.bytes
7 order by b.tablespace_name;
```



```

select b.file_name 物理文件名, b.tablespace_name 表空间,
b.bytes/1024/1024 大小 M,
(b.bytes-sum(a.bytes))/1024/1024 已使用 M,
substr((b.bytes-sum(a.bytes))/b.bytes*100,1,5) 使用率
from dba_free_space a, dba_data_files b
where a.file_id = b.file_id
group by b.tablespace_name, b.file_name, b.bytes
order by b.tablespace_name;

```

物理文件名	表空间	大小M	已使用M	使用率
-----	-----	-----	-----	-----
D:\ORACLE\JIANGXUE\O RADATA\ORCL\SYSAUX01 .DBF	SYSAUX	660.9375	628.375	95.07
D:\ORACLE\JIANGXUE\O RADATA\ORCL\SYSTEM01 .DBF	SYSTEM	720	711.9375	98.88
F:\T1.DBF	T1	2	1.25	62.5
D:\UNDOTBS01_2.DBF	UNDOTBS01	4	2.25	56.25
F:\UNDOTBS01_2.DBF	UNDOTBS01	2	1	50
D:\ORACLE\JIANGXUE\O RADATA\ORCL\UNDOTBS0 1.DBF	UNDOTBS1	75	55.375	73.83
D:\ORACLE\JIANGXUE\O RADATA\ORCL\USERS01. DBF	USERS	5	1.5625	31.25

上述命令实现了对数据库中所有数据文件信息的监控，可以看到其存储位置，大小、已使用、利用率信息。根据这些内容，数据库管理员可以判定是否需要扩大数据文件。

(3) 监测在线重做日志文件

数据库中的数据被修改并提交后，其信息会被保存在重做日志文件中。在重做日志文件正常的情况下，数据库才可以被打开，数据库管理员可以通过以下命令来监测日志文件。

```

SQL> desc v$logfile;
 名称                                是否为空? 类型
-----
GROUP#                               NUMBER
STATUS                               VARCHAR2(7)
TYPE                                 VARCHAR2(7)
MEMBER                              VARCHAR2(513)
IS_RECOVERY_DEST_FILE                VARCHAR2(3)

```

```
SQL> column member format a20
SQL> select group#,type,member from v$logfile;
```

GROUP#	TYPE	MEMBER
3	ONLINE	D:\ORACLE\ORADATA\ORCL\RED003.LOG
2	ONLINE	D:\ORACLE\ORADATA\ORCL\RED002.LOG
1	ONLINE	D:\ORACLE\ORADATA\ORCL\RED001.LOG

4 对象检测

(1) 检测表

```
SQL> select table_name from user_tables
2 where table_name not like '%$%';
```

TABLE_NAME
DUAL
OBJECT_USAGE
SMON_SCN_TIME
RESOURCE_MAP
USER_ASTATUS_MAP
_default_auditing_options_
SYSTEM_PRIVILEGE_MAP
TABLE_PRIVILEGE_MAP
STMT_AUDIT_OPTION_MAP
INCEXP
INCUID

上述查询表时，只查询不包含'\$'符的表，图中只包含一部分结果。

(2) 检测序列

```
SQL> select sequence_name,max_value from seq;
```

SEQUENCE_NAME	MAX_VALUE
APPLY\$_DEST_OBJ_ID	1.0000E+27
APPLY\$_ERROR_HANDLER_SEQUENCE	1.0000E+27
APPLY\$_SOURCE_OBJ_ID	1.0000E+27
AQ\$_ALERT_QT_N	1.0000E+27
AQ\$_AQ\$_MEM_MC_N	1.0000E+27
AQ\$_AQ\$_PROP_TABLE_N	1.0000E+27
AQ\$_CHAINSEQ	1.0000E+27
AQ\$_IOTENQTXID	1.0000E+27
AQ\$_KUPC\$DATAPUMP_QUETAB_N	1.0000E+27
AQ\$_PROPAGATION_SEQUENCE	1.0000E+27
AQ\$_PUBLISHER_SEQUENCE	1.0000E+27

上述通过查 seq 获取序列的名称以及最大值，截图只是一部分结果。

(3) 检测同义词

同义词就是为了方便表示某一名称而取的别名，通过以下操作可以查看同义词。

```
SQL> select synonym_name,table_owner,table_name from syn;
```

SYNONYM_NAME	TABLE_OWNER
DEF\$_AQCALL	SYSTEM
DEF\$_AQCALL	
DEF\$_CALLDEST	SYSTEM
DEF\$_CALLDEST	
DEF\$_SCHEDULE	SYSTEM
DEF\$_SCHEDULE	
DEF\$_ERROR	SYSTEM
DEF\$_ERROR	
DEF\$_DEFAULTDEST	SYSTEM
DEF\$_DEFAULTDEST	
DEF\$_LOB	SYSTEM
DEF\$_LOB	

上述截图只包含了部分查询结果，可以看到在 Oracle 中，为了表示方便，它为很多表起了别名。

(4) 监测触发器

触发器是为了保证数据的一致性、安全性而建立的，通过以下语句，数据库管理员可以查询到库中的触发器名称、基表的所有者，具体如下所示。

```
SQL> select trigger_name,table_owner from user_triggers;
```

TRIGGER_NAME	TABLE_OWNER
OLAPISHUTDOWNTRIGGER	SYS
OLAPISTARTUPTRIGGER	SYS
CDC_DROP_CTABLE_BEFORE	SYS
CDC_CREATE_CTABLE_BEFORE	SYS
CDC_CREATE_CTABLE_AFTER	SYS
CDC_ALTER_CTABLE_BEFORE	SYS
AW_TRUNC_TRG	SYS
AW_DROP_TRG	SYS

已选择8行。

用 test 用户登录，查看 test 用户的触发器使用情况。

```
SQL> conn test/test
```

已连接。

```
SQL> select trigger_name,table_owner from user_triggers;
```

TRIGGER_NAME	TABLE_OWNER
TR_INSERT_EMPINF	TEST
TR_EMPINF_ID	TEST

(5) 检测用户

数据库中的对象也包括用户，通过一下语句，可以查询到数据库中用户的信息。

```
SQL> conn sys/ora505 as sysdba;
已连接。
SQL> select * from all_users order by user_id;
```

USERNAME	USER_ID	CREATED
SYS	0	25-2月 -09
SYSTEM	5	25-2月 -09
OUTLN	9	25-2月 -09
DIP	13	25-2月 -09
TSMSYS	14	25-2月 -09
ORACLE_OCM	20	25-2月 -09
DBSNMP	27	25-2月 -09
WMSYS	28	25-2月 -09
EXFSYS	38	25-2月 -09
CTXSYS	39	25-2月 -09
XDB	41	25-2月 -09

USERNAME	USER_ID	CREATED
SCOTT	81	25-2月 -09
U2	83	12-2月 -12
U1	84	12-2月 -12
U3	86	12-2月 -12
U4	87	12-2月 -12
TEST	88	14-2月 -12
XS\$NULL	2147483638	25-2月 -09