****

**《大型数据库》**

**课内实验四**

指导老师： 盛 津 芳

专 业： 物联网工程

班 级： 物联网1802

学 号： 8208181125

姓 名： 王灏洋

2020 年 12 月

目录

[一、 实验目的和要求 2](#_Toc58918059)

[二、 实验操作环境 2](#_Toc58918060)

[三、 实验内容 2](#_Toc58918061)

[四、 实验步骤 3](#_Toc58918062)

[五、 实验中遇到的问题及解决方案 11](#_Toc58918063)

[六、 实验源代码 11](#_Toc58918064)

# 实验目的和要求

1. 本实验主要是熟悉ORACLE的备份与恢复技术。针对ORACLE表空间进行相关操作。

# 实验操作环境

1. 硬件：主频2GHz以上服务器（内存2GB以上、硬件空闲2.2GB以上），主频1GHz以上微机，内存1GB以上。
2. 软件：WINDOWS XP/2000/2003/或Win7/Win8/Win10等。如操作系统是 SERVER版， 可安装ORACLE/9i/10g/11g/12C FOR NT/WINDOWS(注意有32位与64位的区别，可选企业版)； 如果Windows非server如XP/win7等，安装时请选择个人版(PERSONAL)，注意安装时要有兼容性设置与用管理员运行。安装过程中需要关注系统预定义的账号SYS与SYSTEM的密码设置。

# 实验内容

1. 建立对应3GB大小的外部文件的tablespace tabspace\_????
2. 重建major\_????与stud\_????，指定存于建立的tabspace\_????中，
3. 重新产生样本值，包括千万级数据的stud表，看是否有性能上的提升。
4. 用EXP导出数据与IMP导入数据，请作多种尝试，直到成功！
5. 尝试对系统表空间user及自定义表空间tabspace\_????进行备份与恢复。
6. 登录DBA用户system，通过cat字典入口，找到以DBA\_开头的相关数据字典，并且每个对象显示5条记录（SQL生成SQL）。
7. 通过查找自己用户下的触发器字典，生成代码将所有触发器的状态改为disable并执行。再生成代码，将状态为disable的触发器的状态改为enable，并执行。

# 实验步骤

1. **建立对应3GB大小的外部文件的tablespace tabspace\_????**

|  |
| --- |
|  |

1. **重建major\_????与stud\_????，指定存于建立的tabspace\_????中。**

|  |
| --- |
|  |

得到的运行结果如下

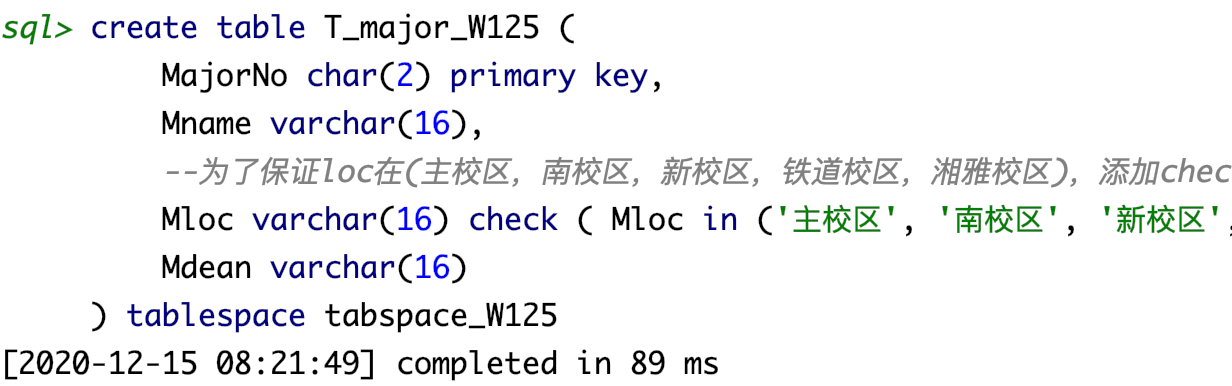


图1 重新建立T\_major\_W125表

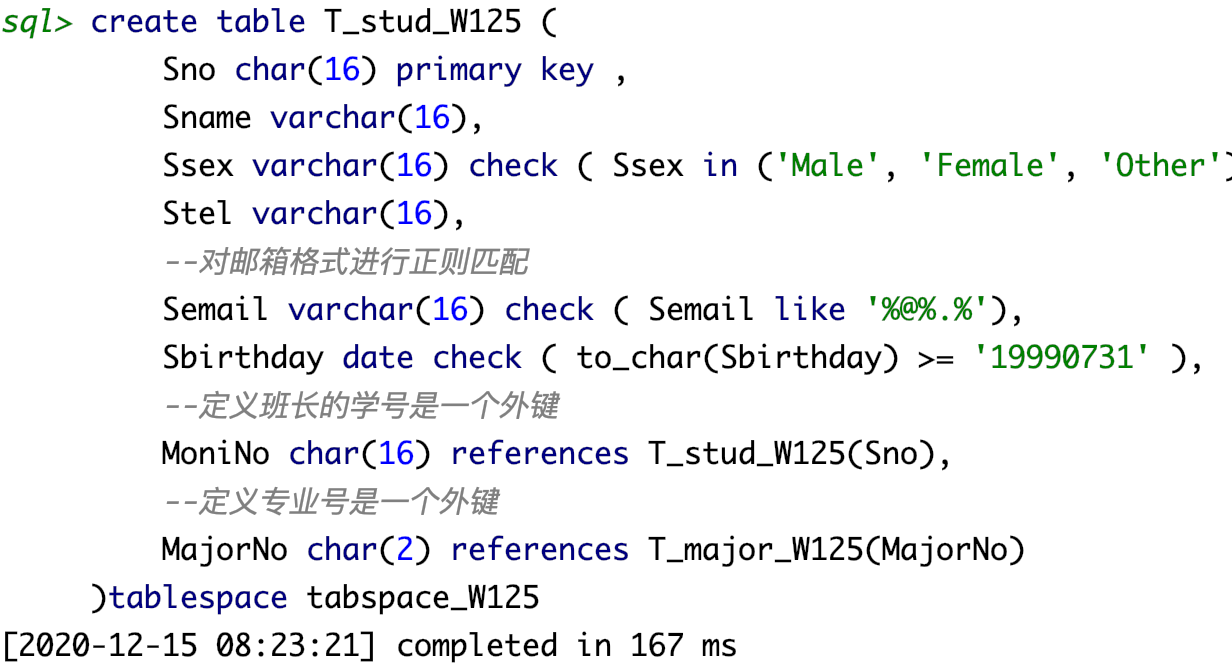


图2 重新建立T\_stud\_W125表

1. **重新产生样本值，包括千万级数据的stud表，看是否有性能上的提升。**
2. 导入姓名，使用sqlldr进行数据的导入。随后，利用笛卡尔积生成大量姓名

|  |
| --- |
|  |

1. 根据规则生成大量的学号。

|  |
| --- |
|  |

1. 根据规则产生电话号码

|  |
| --- |
|  |

1. 根据规则产生邮箱

|  |
| --- |
|  |

1. 根据规则产生生日

|  |
| --- |
|  |

1. 根据规则产生千万条数据的除姓名外的其他信息。

|  |
| --- |
|  |

1. 插入数据，并进行比较

|  |
| --- |
|  |

得到的运行结果如下：

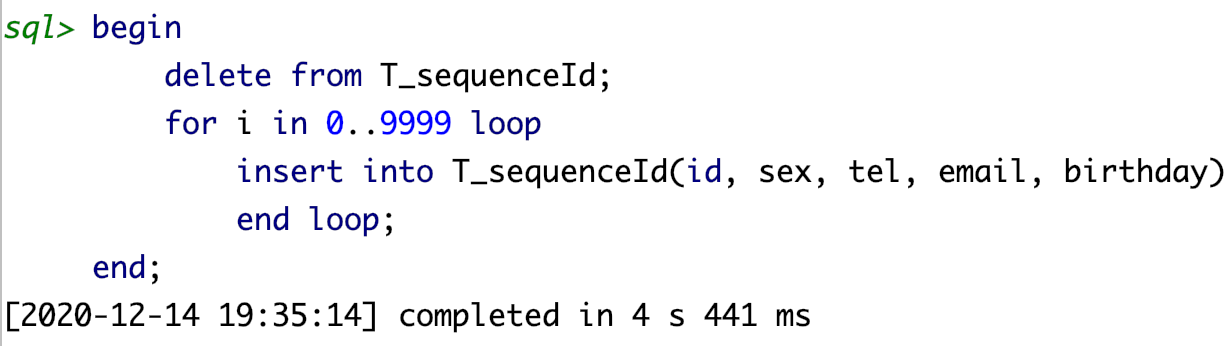


图3 重新生成1w条数据

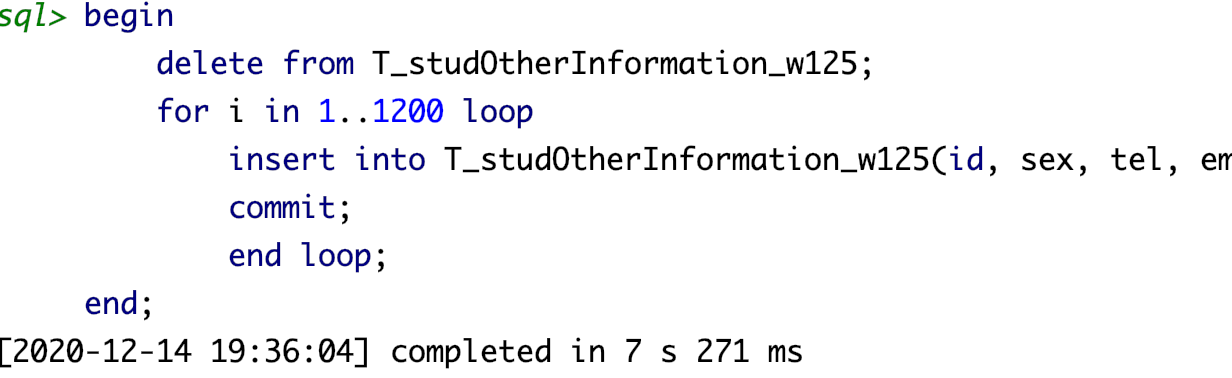


图4 得到1200W条其他数据

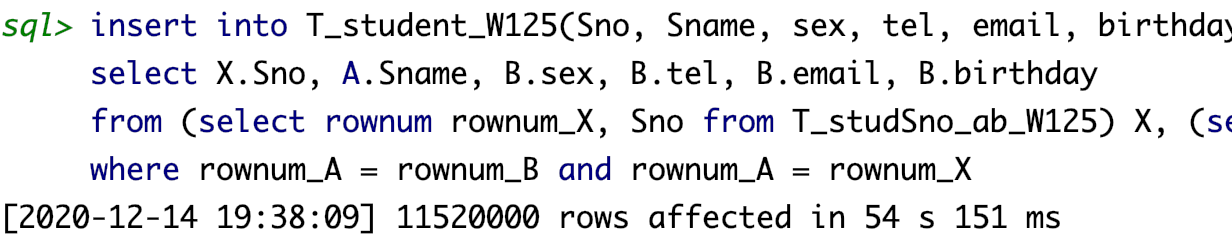


图5 得到1200W条学生数据

通过和之前实验得到的数据进行对比，有**四倍以上**的效率提升。

1. **用EXP导出数据与IMP导入数据，请作多种尝试，直到成功！**

接下来，我们使用exp和imp导入/导出用户why的所有数据。

|  |
| --- |
|  |

得到结果如下

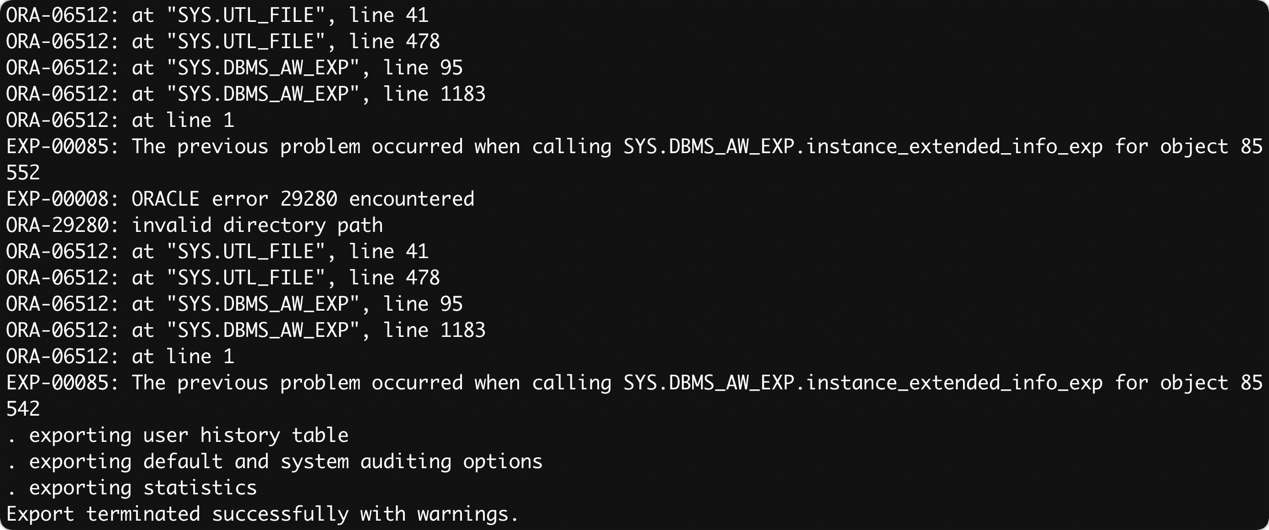


图6 成功导出数据

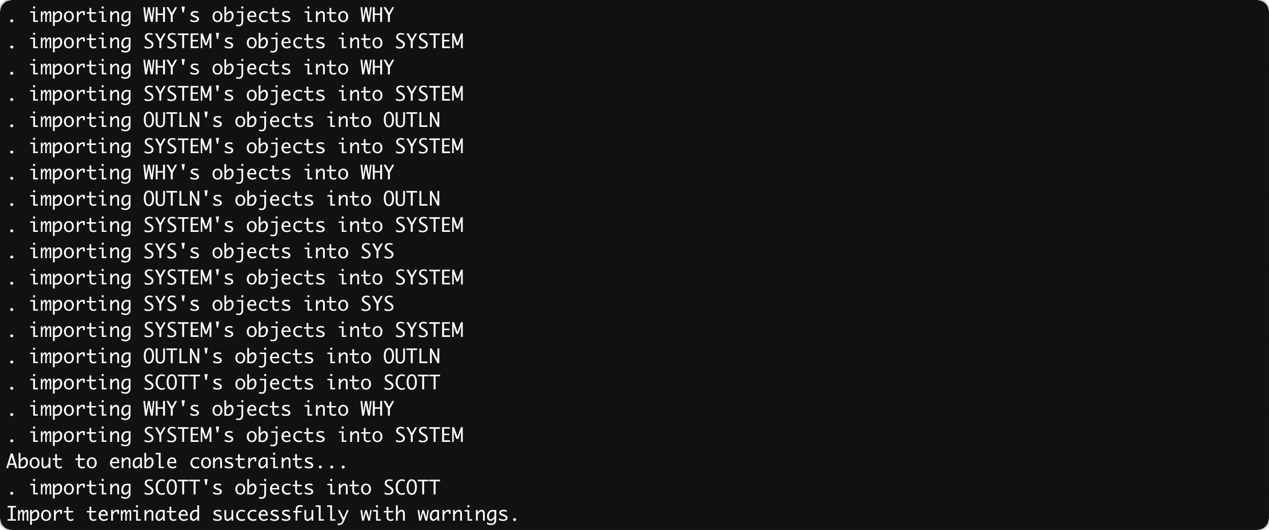


图7 成功导入数据

1. **尝试对系统表空间user及自定义表空间tabspace\_????进行备份与恢复。**

|  |
| --- |
|  |

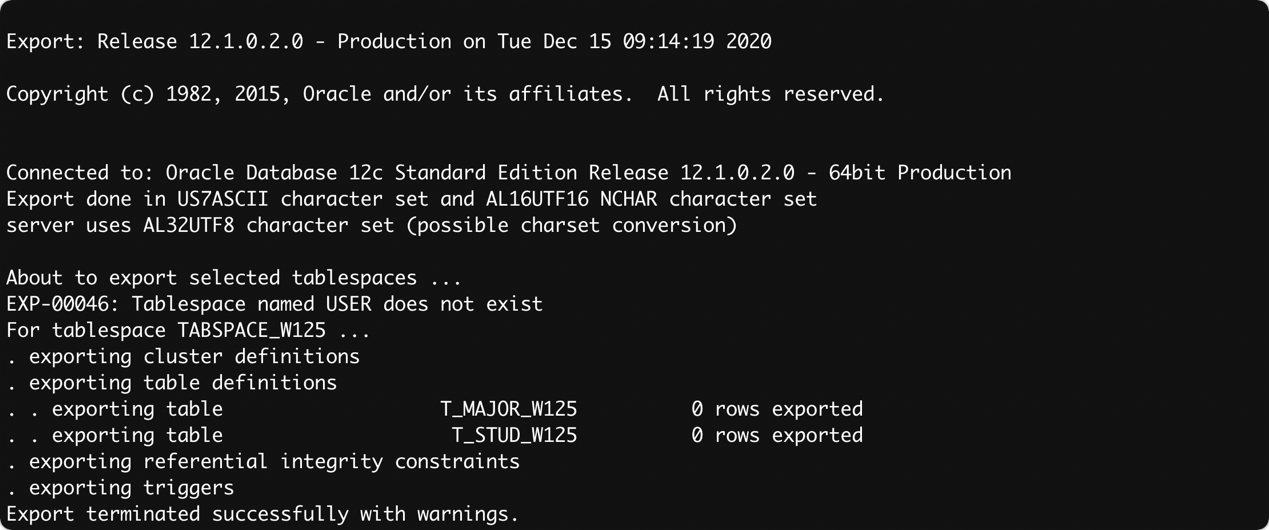
****

图8 成功导出表空间

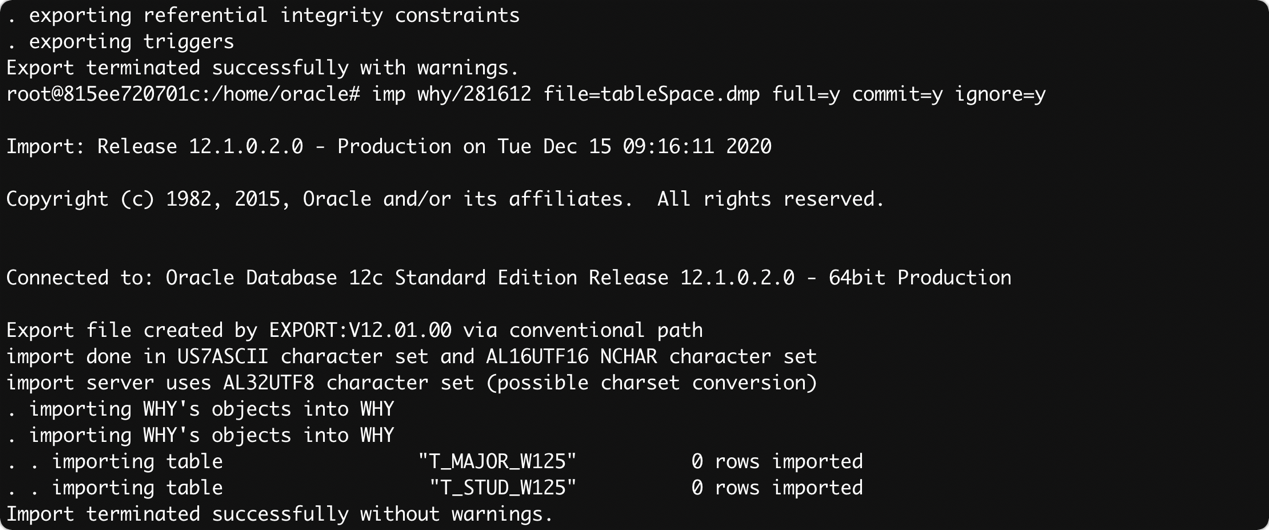
****

图9 成功导入表空间

1. **登录DBA用户system/oracle，通过cat字典入口，找到以DBA\_开头的相关数据字典，并且每个对象显示5条记录（SQL生成SQL）。**

|  |
| --- |
|  |

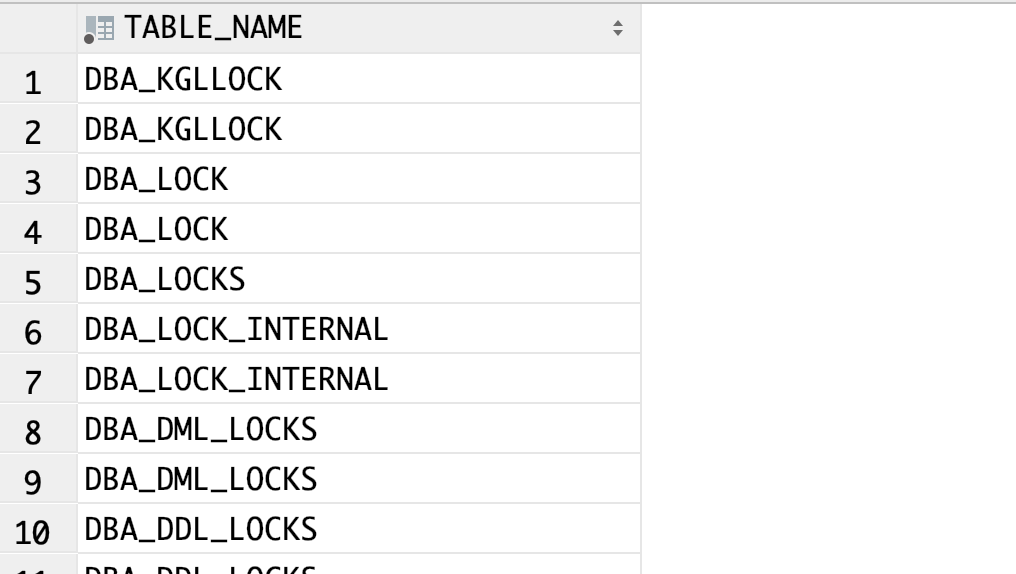
****

图10 成功查看数据字典

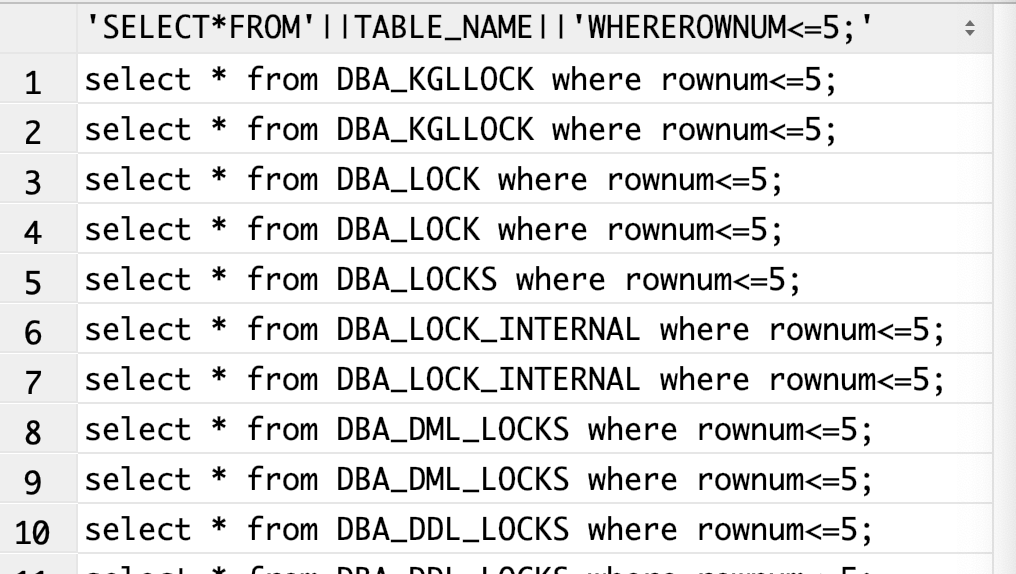
****

图11 成功查看数据字典

1. **通过查找自己用户下的触发器字典，生成代码将所有触发器的状态改为disable并执行。再生成代码，将状态为disable的触发器的状态改为enable，并执行。**

|  |
| --- |
|  |

得到的实验结果如下

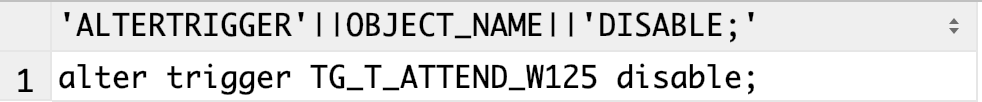
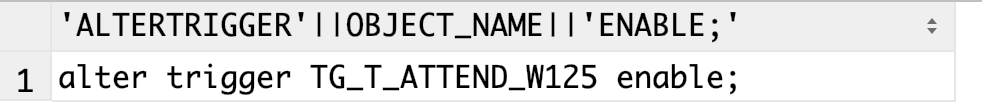
****

图12 生成SQL，完成触发器操作

# 实验中遇到的问题及解决方案

1. 在导入导出数据的时候时间有些漫长，一直让我以为是我做错什么了，但是好在是没有什么错误的，稍等一会就好了。
2. 从实验结果可以看出，建立了表空间之后，导入千万条记录所消耗的时间有了四倍以上的提升，效果非常之好
3. 通过这次实验学习，我学习到了关于表空间的相关知识，同时，实践了整个数据库或者表空间的导入导出。
4. 在不知道是dba\_catlog的时候，为了找出数据，我使用了dictionary这个数据字典，就很好奇这两个是什么关系，后来发现，dictionary是系统权限，而catlog是用户使用的。

# 实验源代码

|  |
| --- |
| *--1．建立对应3GB大小的外部文件的tablespace tabspace\_???? --2．重建major\_????与stu d\_????，指定存于建立的tabspace\_????中， --3．重新产生样本值，包括千万级数据的stud表，看是否有性能上的提升。 --4．用EXP导出数据与IMP导入数据，请作多种尝试，直到成功！ --5．尝试对系统表空间user及自定义表空间tabspace\_????进行备份与恢复。 --6．登录DBA用户system，通过cat字典入口，找到以DBA\_开头的相关数据字典，并且每个对象显示5条记录（SQL生成SQL）。 --7．通过查找自己用户下的触发器字典，生成代码将所有触发器的状态改为disable并执行。再生成代码，将状态为disable的触发器的状态改为enable，并执行。  --1．建立对应3GB大小的外部文件的tablespace tabspace\_W125* **create tablespace** tabspace\_W125 **datafile '\home\oracle\tabspace\_W125.dbf' size** 3G **autoextend on next** 100M **extent management local** ;  *--2．重建major\_????与stud\_????，指定存于建立的tabspace\_????中 --创建T\_major\_W125表* **drop table** T\_major\_W125; **create table** T\_major\_W125 (  **MajorNo char**(2) **primary key**,  **Mname varchar**(16),  *--为了保证loc在(主校区，南校区，新校区，铁道校区，湘雅校区)，添加check约束* **Mloc varchar**(16) **check** ( **Mloc in** (**'主校区'**, **'南校区'**, **'新校区'**, **'铁道校区'**, **'湘雅校区'**) ),  **Mdean varchar**(16) ) **tablespace** tabspace\_W125;  *--创建T\_stud\_W125表* **drop table** T\_stud\_W125; **create table** T\_stud\_W125 (  **Sno char**(16) **primary key** ,  **Sname varchar**(16),  **Ssex varchar**(16) **check** ( **Ssex in** (**'Male'**, **'Female'**, **'Other'**) ),  **Stel varchar**(16),  *--对邮箱格式进行正则匹配* **Semail varchar**(16) **check** ( **Semail like '%@%.%'**),  **Sbirthday date check** ( *to\_char*(**Sbirthday**) >= **'19990731'** ),  *--定义班长的学号是一个外键* **MoniNo char**(16) **references** T\_stud\_W125(**Sno**),  *--定义专业号是一个外键* **MajorNo char**(2) **references** T\_major\_W125(**MajorNo**) )**tablespace** tabspace\_W125;  *--学生信息表* **create table** T\_student\_W125; **create table** T\_student\_W125 (  **Sno varchar2**(10) **primary key** ,  **Sname varchar2**(32),  **sex varchar2**(32),  **tel varchar2**(32),  **email varchar2**(32),  **birthday date** ) **tablespace** tabspace\_W125;  *--3．重新产生样本值，包括千万级数据的stud表，看是否有性能上的提升。 --3.1创建基本姓名表* **drop table** Sname; **create table** Sname(**Sname varchar**(32)); *--3.2 插入基本姓名 --sqlldr why/281612 control = /home/oracle/persons.ctl* **select** *\** **from** Sname;  *--3.3 制造足够多的姓名，为名字的每个字创造视图* **drop view** V\_name1\_W125; **drop view** V\_name2\_W125; **drop view** V\_name3\_W125; **drop view** V\_name12\_W125; **drop view** V\_name123\_W125; **create view** V\_name1\_W125 **as select** *\** **from** (**select distinct** *substr*(**sname**,1,1) na1 **from** Sname **order by** dbms\_random.VALUE()) **where** *rownum*<=500; **create view** V\_name2\_W125 **as select** *\** **from** (**select distinct** *substr*(**sname**,2,1) na2 **from** Sname **order by** dbms\_random.VALUE()) **where** *rownum*<=1200; **create view** V\_name3\_W125 **as select** *\** **from** (**select distinct** *substr*(**sname**,3,1) na3 **from** Sname **order by** dbms\_random.VALUE()) **where** *rownum*<=20; **create view** V\_name12\_W125 **as select** *concat*(**na1**,**na2**) na12 **from** V\_name1\_W125,V\_name2\_W125; **create view** V\_name123\_W125 **as select** *concat*(**na12**,**na3**) na123 **from** V\_name12\_W125,V\_name3\_W125;  *--3.4 创建T\_studName\_W125表* **drop table** T\_studName\_W125; **create table** T\_studName\_W125(**Sname varchar**(16));  *--3.5 为姓名表T\_studName\_W125插入12600000条学生姓名* **insert into** T\_studName\_W125(**Sname**) **select na12 from** V\_name12\_W125; **insert into** T\_studName\_W125(**Sname**) **select na123 from** V\_name123\_W125; *--完成为姓名表生成12600000条记录* **select** *count*(*\**) **from** V\_name123\_W125; **select** *count*(*\**) **from** V\_name12\_W125; **select** *count*(*\**) **from** T\_studName\_W125;  *--3.6.生成规范的学号表 --每次生成两位，分四批生成* **drop table** T\_studSno\_gh\_W125; **drop table** T\_studSno\_ef\_W125; **drop table** T\_studSno\_cd\_W125; **drop table** T\_studSno\_ab\_W125; **create table** T\_studSno\_gh\_W125 (**Sno number**(10)); **create table** T\_studSno\_ef\_W125 (**Sno number**(10)); **create table** T\_studSno\_cd\_W125 (**Sno number**(10)); **create table** T\_studSno\_ab\_W125 (**Sno varchar2**(10)); *--建立过程：* **declare** i **int**; **begin for** i **in** 1..32 **loop  insert into** T\_studSno\_gh\_W125 **values**(i); **end loop**; **for** i **in** 1..50 **loop  insert into** T\_studSno\_ef\_W125 **select** 170000 + i \* 100 + T\_studSno\_gh\_W125.**sno from** T\_studSno\_gh\_W125;  **commit**; **end loop**; **for** i **in** 1..90 **loop  insert into** T\_studSno\_cd\_W125 **select** i \* 1000000 + T\_studSno\_ef\_W125.**sno from** T\_studSno\_ef\_W125;  **commit**; **end loop**; **for** i **in** 1..80 **loop  insert into** T\_studSno\_ab\_W125 **select** *substr*(10000000000 + i \* 100000000 + T\_studSno\_cd\_W125.**sno**, 2, 10) **from** T\_studSno\_cd\_W125;  **commit**; **end loop**; **end**; *--完成11520000条学号ID生成  --3.7批量处理sex, tel, email, birthday等数据 -- 产生随机性别：男、女占比为99%到99.5% -- 在性别表T\_sex\_W125中，143人，男女分别71人，一人性别为其它。随机查询一条记录时，可使男、女占比为99.3% -- 函数F\_getSex\_W125返回值为随机性别* **create or replace function** *F\_getSex\_W125* **return varchar2 is** f\_numb **number**; f\_sex **varchar2**(8); **begin** *--dual是虚拟表，主要用来借用以调用系统函数* **select** *trunc*(DBMS\_RANDOM.VALUE(1, 143)) **into** f\_numb **from** *dual*;  **if** f\_numb <= 71 **then** f\_sex := **'男'**;  **elsif** f\_numb <= 142 **then** f\_sex := **'女'**;  **else** f\_sex := **'其他'**;  **end if**;  **return** f\_sex; **end**;  *--3.8产生随机电话 -- 函数F\_getTel\_W125会返回一个随机手机号* **drop table** T\_tel\_W125; **create table** T\_tel\_W125(**tel varchar2**(4)); **insert into** T\_tel\_W125 **values**(**'132'**); **insert into** T\_tel\_W125 **values**(**'135'**); **insert into** T\_tel\_W125 **values**(**'156'**); **insert into** T\_tel\_W125 **values**(**'151'**); **insert into** T\_tel\_W125 **values**(**'138'**); **insert into** T\_tel\_W125 **values**(**'139'**); **insert into** T\_tel\_W125 **values**(**'183'**); **insert into** T\_tel\_W125 **values**(**'187'**); **insert into** T\_tel\_W125 **values**(**'153'**); **insert into** T\_tel\_W125 **values**(**'150'**); **insert into** T\_tel\_W125 **values**(**'186'**); **insert into** T\_tel\_W125 **values**(**'188'**); **create or replace function** *F\_getTel\_W125* **return varchar2 is** getTel **varchar2**(12);  numb\_head **varchar2**(4);  numb\_other **varchar2**(16); **begin  select tel into** numb\_head **from** (**select tel from** T\_tel\_W125 **order by** DBMS\_RANDOM.VALUE()) **where** *rownum* = 1 ;  **select** *substr*(*cast*(DBMS\_RANDOM.VALUE() **as varchar2**(32)), 3, 8) **into** numb\_other **from** *dual*;  getTel := numb\_head || numb\_other;  **return** getTel; **end**;  *--3.9随机产生邮箱 -- 函数F\_getEmail\_W125会返回一个随机邮箱号* **drop table** T\_email\_W125; **create table** T\_email\_W125 (**email varchar2**(16)); **insert into** T\_email\_W125 **values**(**'126'**); **insert into** T\_email\_W125 **values**(**'139'**); **insert into** T\_email\_W125 **values**(**'sohu'**); **insert into** T\_email\_W125 **values**(**'sina'**); **insert into** T\_email\_W125 **values**(**'163'**); **insert into** T\_email\_W125 **values**(**'foxmail'**); **insert into** T\_email\_W125 **values**(**'qq'**); **insert into** T\_email\_W125 **values**(**'qq'**); **insert into** T\_email\_W125 **values**(**'qq'**); **insert into** T\_email\_W125 **values**(**'qq'**); **insert into** T\_email\_W125 **values**(**'qq'**); **insert into** T\_email\_W125 **values**(**'qq'**); **insert into** T\_email\_W125 **values**(**'qq'**); **insert into** T\_email\_W125 **values**(**'qq'**); **insert into** T\_email\_W125 **values**(**'qq'**); **insert into** T\_email\_W125 **values**(**'qq'**); **insert into** T\_email\_W125 **values**(**'qq'**); **insert into** T\_email\_W125 **values**(**'qq'**);  **create or replace function** *F\_getEmail\_W125* **return varchar2 is** getEmail **varchar2**(32);  emName **varchar2**(16);  emOwn **varchar2**(16); **begin  select** *substr*(*cast*(DBMS\_RANDOM.VALUE() **as varchar2**(32)), 3, 11) **into** emName **from** *dual*;  **select email into** emOwn **from** (**select email from** T\_email\_W125 **order by** DBMS\_RANDOM.VALUE()) **where** *rownum* = 1;  getEmail := emName || **'@'** || emOwn || **'.com'**;  **return** getEmail; **end**;  *--3.10产生随机生日 -- 函数f\_getBirthday\_W125会返回一个在[19940101,19990701]时间内的日期* **create or replace function** *F\_getBirthday\_W125* **return date is** birthday **date**; **begin  select** *to\_date*(*trunc*(DBMS\_RANDOM.VALUE(2449354, 2451186)), **'J'**) **into** birthday **from** *dual*;  **return** birthday; **end**; *--性别、手机号、邮箱、出生日期等随机函数生成完毕   --3.11生成完整学生表除学号和姓名之外的所有信息 --建立基本信息模板表* **drop table** T\_sequenceId; **create table** T\_sequenceId(  **id number**(10),  **sex varchar2**(32),  **tel varchar2**(32),  **email varchar2**(32),  **birthday date** );  *--存储大量的学生其他信息，千万级以上* **drop table** T\_studOtherInformation\_w125; **create table** T\_studOtherInformation\_w125 (  **id number**(16),  **sex varchar2**(32),  **tel varchar2**(32),  **email varchar2**(32),  **birthday date** );  *--生成1万条数据模版* **begin  delete from** T\_sequenceId;  **for** i **in** 0..9999 **loop  insert into** T\_sequenceId(**id**, **sex**, **tel**, **email**, **birthday**) **values** (i, *F\_getSex\_W125*(), *F\_getTel\_W125*(), *F\_getEmail\_W125*(), *F\_getBirthday\_W125*());  **end loop**; **end**; **select** *\** **from** T\_sequenceId **where** *ROWNUM* < 100; **select** *count*(*\**) **from** T\_sequenceId;  *--每1w条数据插入一次，得到12000000数据* **begin  delete from** T\_studOtherInformation\_w125;  **for** i **in** 1..1200 **loop  insert into** T\_studOtherInformation\_w125(**id**, **sex**, **tel**, **email**, **birthday**) **select** i \* 10000 + T\_sequenceId.**id as** MSISDN, T\_sequenceId.**sex**, T\_sequenceId.**tel**, T\_sequenceId.**email**, T\_sequenceId.**birthday from** T\_sequenceId;  **commit**;  **end loop**; **end**; *--完成生成12000000条其他信息生成  --插入数据* **insert into** T\_student\_W125(**Sno**, **Sname**, **sex**, **tel**, **email**, **birthday**) **select** X.**Sno**, **A**.**Sname**, B.**sex**, B.**tel**, B.**email**, B.**birthday from** (**select** *rownum* rownum\_X, **Sno from** T\_studSno\_ab\_W125) X, (**select** *rownum* rownum\_A, **Sname from** T\_studName\_W125) **A**, (**select** *rownum* rownum\_B, **sex**, **tel**, **email**, **birthday from** T\_studOtherInformation\_w125) B **where** rownum\_A = rownum\_B **and** rownum\_A = rownum\_X;  *--4．用EXP导出数据与IMP导入数据，请作多种尝试，直到成功！ --exp why/281612 file=student.dmp full=y* **drop table** T\_student\_W125; *--imp why/281612 file=student.dmp full=y commit=y ignore=y* **select** *\** **from** T\_student\_W125; **select** *count*(*\**) **from** T\_student\_W125;  *--5．尝试对系统表空间user及自定义表空间tabspace\_????进行备份与恢复。 --exp why/281612 file=tableSpace.dmp tablespaces=(user,tabspace\_W125)  --6．登录DBA用户system/oracle，通过cat字典入口，找到以DBA\_开头的相关数据字典，并且每个对象显示5条记录（SQL生成SQL）。 /\*dba\_开头的是查全库所有的，all\_开头的是查当前用户可以看到的，user\_开头的是查当前用户的，后面可接\_users \_tables \_tablespaces \_objects\*/* **select table\_name from** USER\_CATALOG **where table\_name like 'DBA\_%'**; **select TABLE\_NAME from DICTIONARY where TABLE\_NAME like 'DBA\_%'**; **select table\_name from** USER\_CATALOG; **select TABLE\_NAME from DICTIONARY** ;  *--dba模式下执行* **select 'select \* from '**||**table\_name**||**' where rownum<=5;' from** USER\_CATALOG; **select 'select \* from '**||**table\_name**||**' where rownum<=5;' from DICTIONARY**;  *--7．通过查找自己用户下的触发器字典，生成代码将所有触发器的状态改为disable并执行。再生成代码，将状态为disable的触发器的状态改为enable，并执行。 --connect why/281612;* **select 'alter trigger '**||object\_name||**' disable;' from** user\_objects **where** OBJECT\_type=**'TRIGGER'** ; **select 'alter trigger '**||object\_name||**' enable;' from** user\_objects **where** OBJECT\_type=**'TRIGGER'**; **alter trigger** TG\_T\_ATTEND\_W125 **disable**; **alter trigger** TG\_T\_ATTEND\_W125 **enable**; |