

D 公司的员工需要删除某个无用的表空间即 DROP TABLESPACE INCLUDING CONTENTS 操作，但是在操作 DROP TABLESPACE 后，开发部门反映该被 DROP 掉的 TABLESPACE 上其实有一个 SCHEMA 的数据是有用且重要的，但现在表空间被 DROP 了，且无任何备份。

此时可以利用 PRM 的 No-Dict 模式去抽取被 DROP TABLESPACE 的对应的所有数据文件中的数据。通过这种方式可以恢复大部分数据，但由于是非字典模式所以需要将恢复出来的表与应用数据表一一对应起来，此时一般需要应用开发维护人员介入，通过人工识别来分辨哪些数据属于哪张表。由于 DROP TABLESPACE 操作修改了数据字典，并在 OBJ\$ 中删除了对应表空间上的对象，所以无法从 OBJ\$ 上获得 DATA_OBJECT_ID 与 OBJECT_NAME 之间的对应关系。此时我们可以利用如下的方法，尽可能多得获取 DATA_OBJECT_ID 与 OBJECT_NAME 之间的对应关系。

```
select tablespace_name,segment_type,count(*) from dba_segments where
owner='PARNASSUSDATA' group by tablespace_name,segment_type;
```

TABLESPACE	SEGMENT_TYPE	COUNT(*)
USERS	TABLE	126
USERS	INDEX	136

```
SQL> select count(*) from obj$;
```

COUNT(*)
75698

```
SQL> select current_scn, systimestamp from v$database;
```

CURRENT_SCN	SYSTIMESTAMP
1895940	25-4 月 -14 09.18.00.628000 下午 +08:00

```
SQL> select file_name from dba_data_files where tablespace_name='USERS';
```

```
FILE_NAME
```

```
-----  
H:\APP\MACLEAN\ORADATA\PARNASSUS\DATAFILE\O1_MF_USERS_9MNBMJYJ_.DBF
```

```
SQL> drop tablespace users including contents;
```

表空间已删除。

```
C:\Users\maclean>dir
```

```
H:\APP\MACLEAN\ORADATA\PARNASSUS\DATAFILE\O1_MF_USERS_9MNBMJYJ_.DBF
```

驱动器 H 中的卷是 entertainment

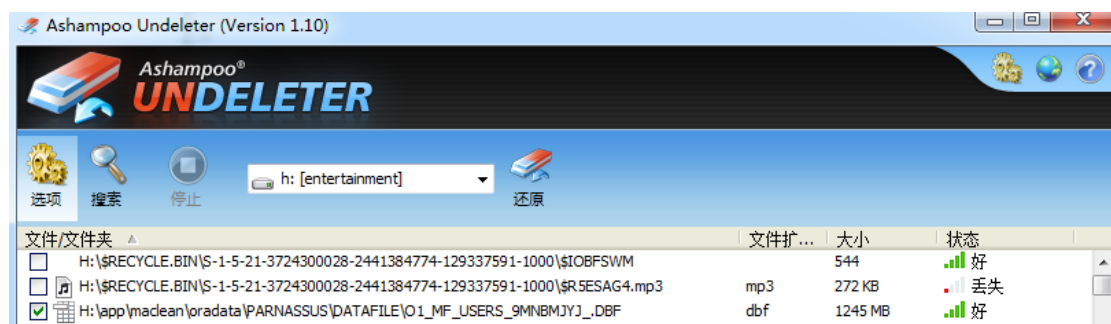
卷的序列号是 A87E-B792

```
H:\APP\MACLEAN\ORADATA\PARNASSUS\DATAFILE 的目录
```

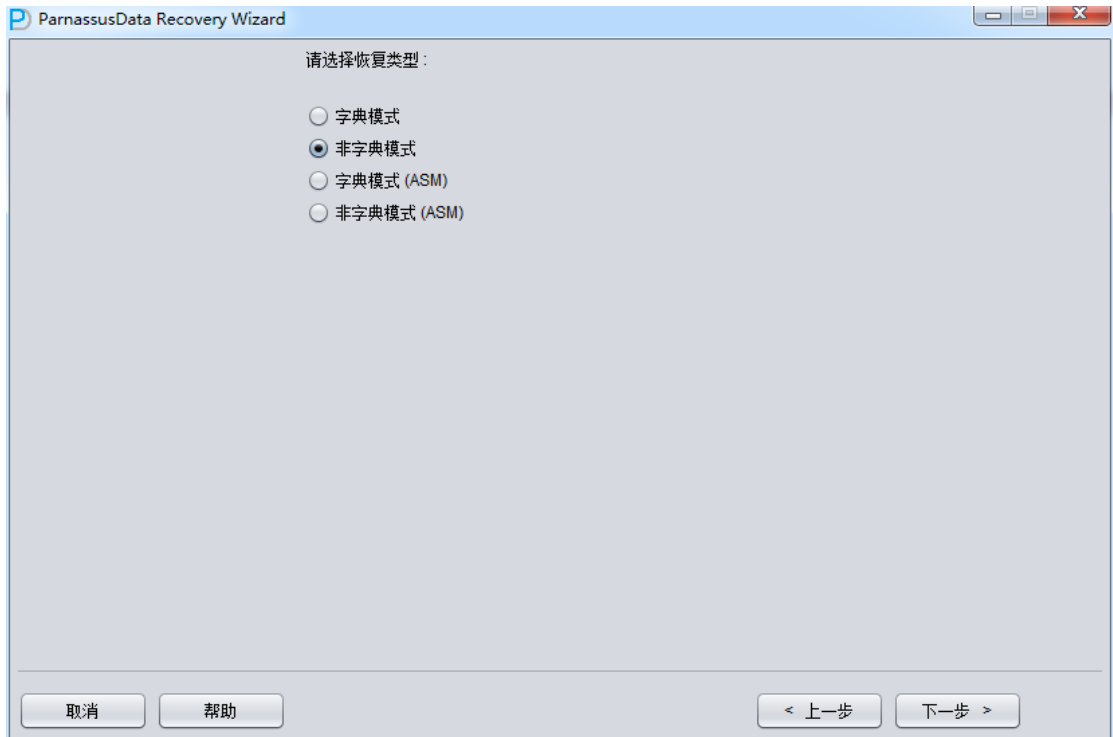
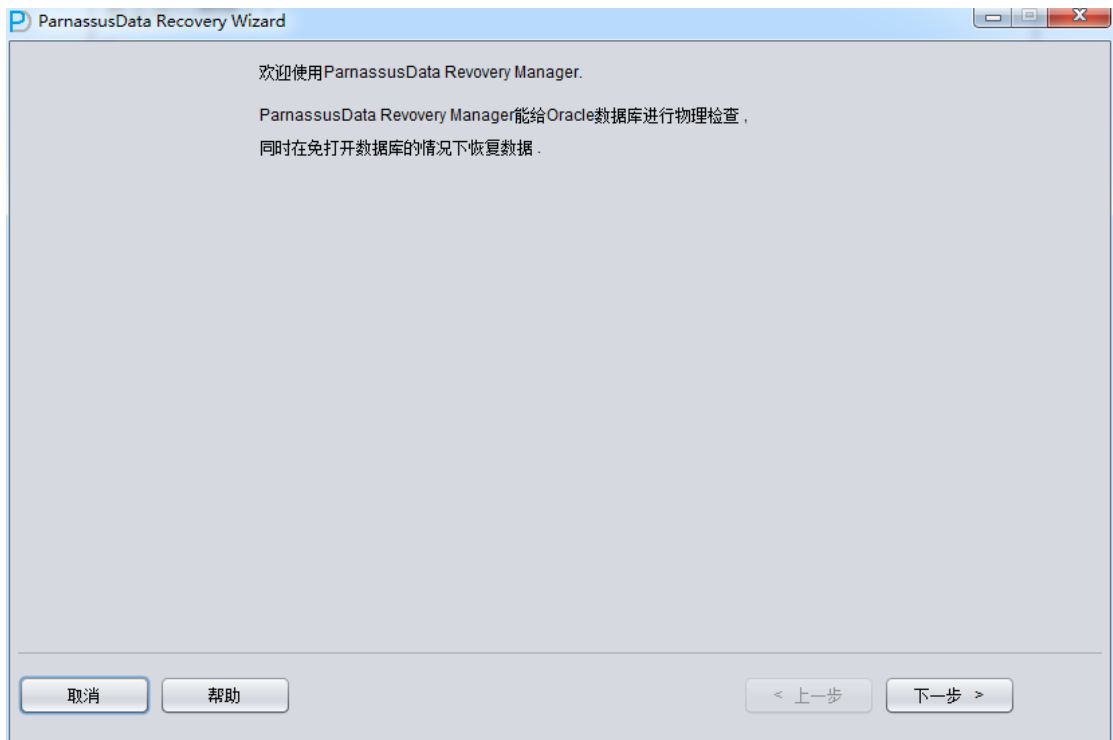
找不到文件

因为 drop tablespace 后该 TABLESPACE 对应的数据文件在 OS 上被删除。

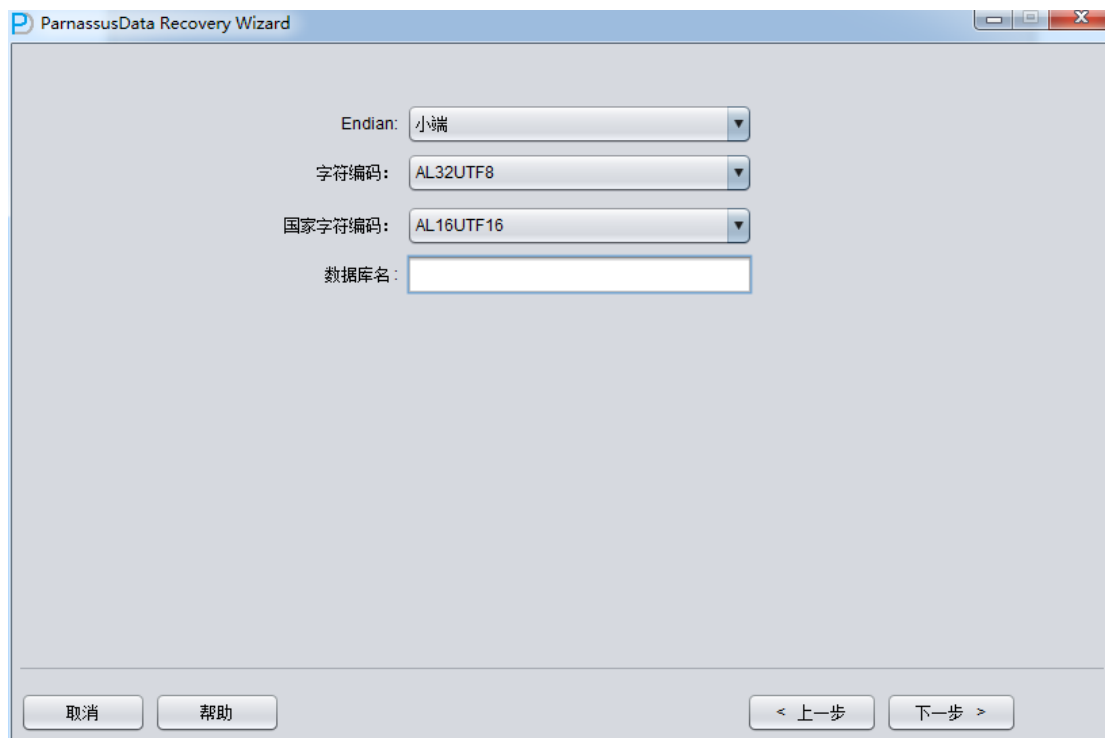
此时通过文件恢复工具例如 Windows 平台上可以使用 UNDELETER 将被误删除的数据文件还原出来



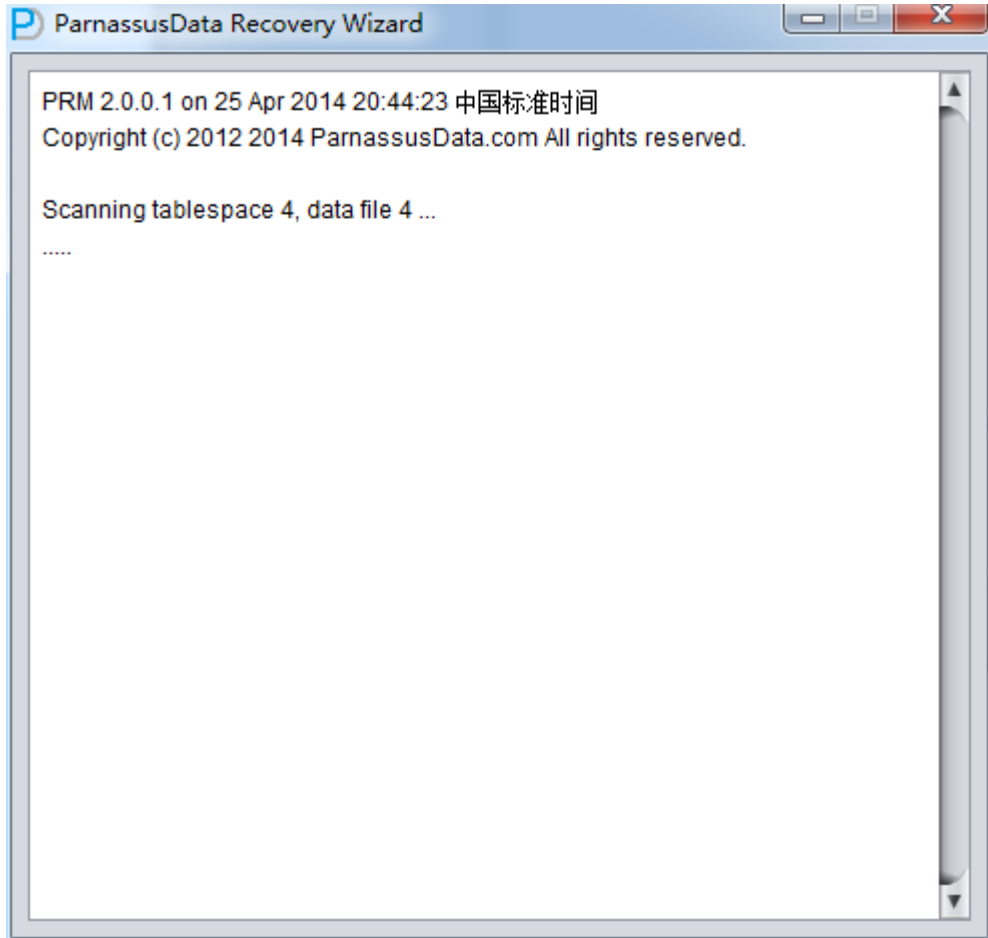
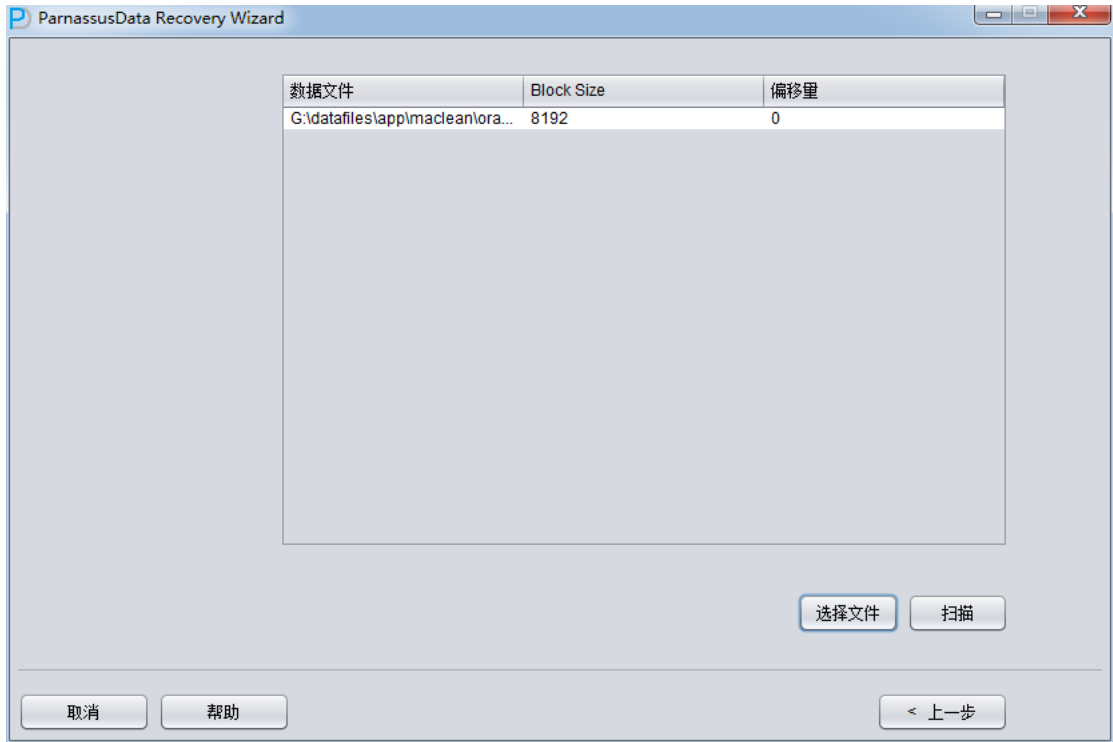
启动 PRM => recovery Wizard => 非字典模式



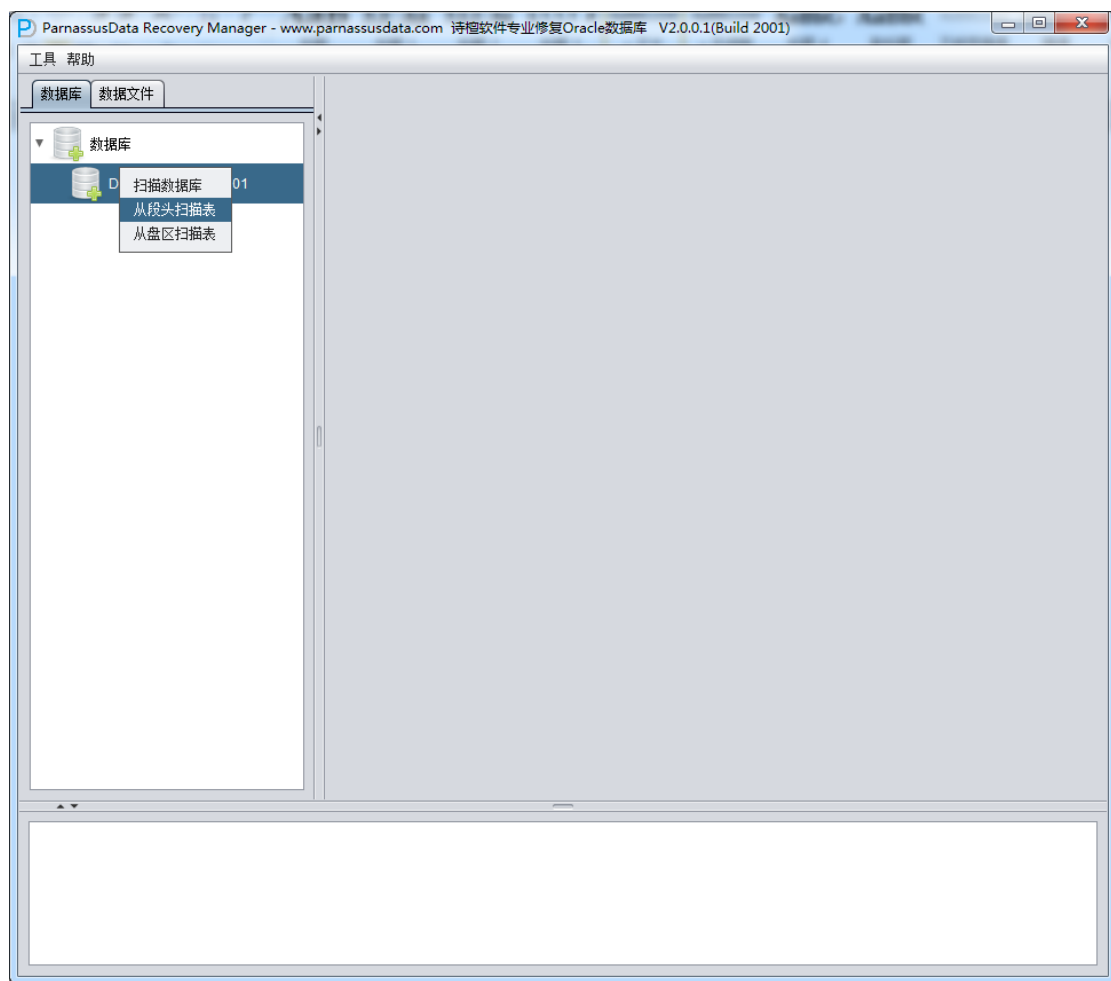
由于是非字典模式，所以需要自己选择合理的字符集！



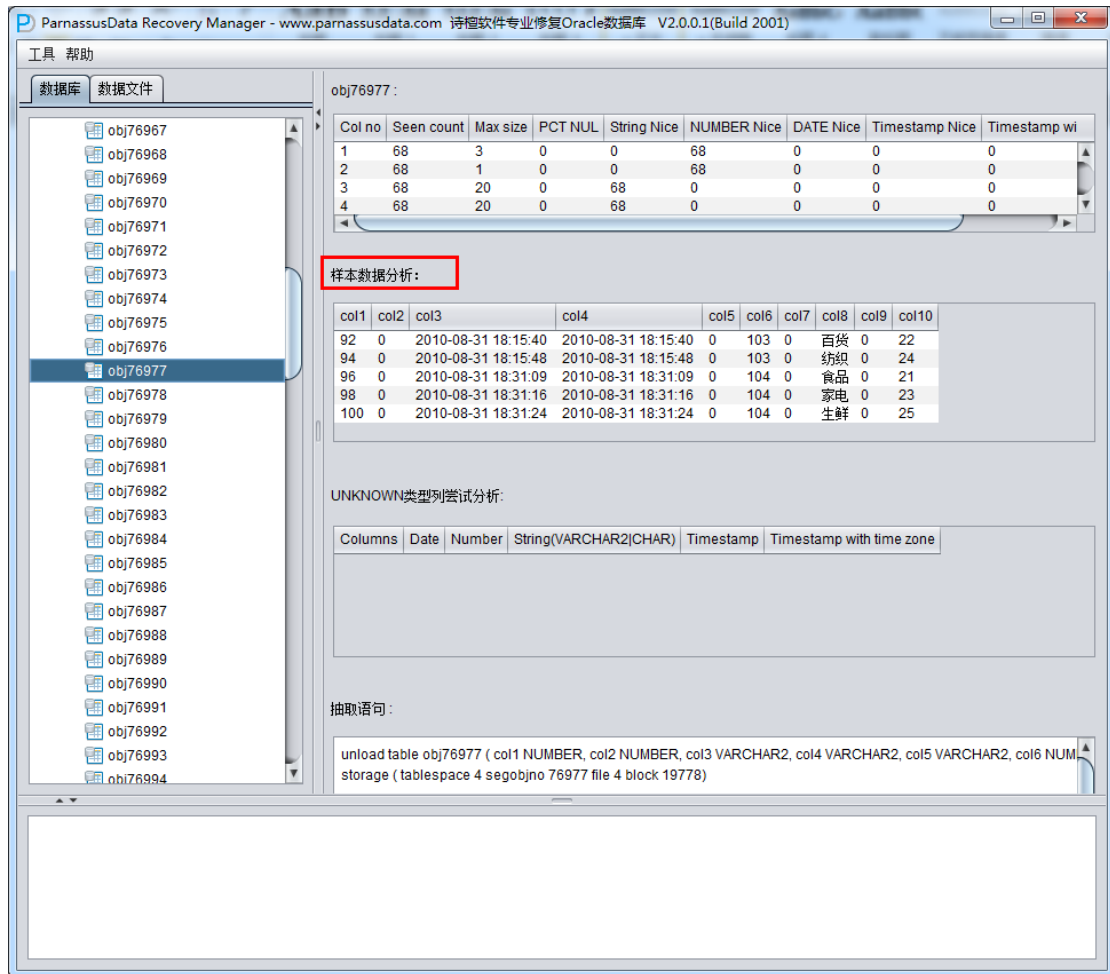
加入刚才恢复出来的数据文件并点击扫描



之后选择从段头/盘区扫描表，如果从段头扫描表未能找到所有表，则考虑用从盘区扫描：



此时可以看到主界面树形图出现大量 OBJXXXXX 的表，这里的 OBJXXXXX 实际就是表的 DATA_OBJECT_ID，一般如果有熟悉该套系统应用模式开发的技术人员可以通过浏览样本数据分析将该表与应用表对应起来：



如果没有人可以帮忙对应数据与表之间的关系，则可以考虑使用如下的手段：

由于此例子中仅仅是 DROP 了 TABLESPACE 表空间，而数据库本身完全是可用的，则此时可以利用 FLASHBACK QUERY 来获得 DATA_OBJECT_ID 与表名之间的映射关系。

```
SQL> select count(*) from sys.obj$;
```

```
COUNT(*)
```

```
-----
```

```
75436
```

```
SQL> select count(*) from sys.obj$ as of scn 1895940;
select count(*) from sys.obj$ as of scn 1895940
```

*

第 1 行出现错误:

ORA-01555: 快照过旧: 回退段号 0 (名称为 "SYSTEM") 过小

一开始想利用 FLASHBACK QUERY 来找出 OBJ\$上之前的记录, 但是发现由于使用 SYSTEM ROLLBACK SEGMENT 所以会出现 ORA-01555 错误

此时可以考虑使用 AWR 视图 DBA_HIST_SQL_PLAN, 只要在最近 7 天中访问过该表一般可以从执行计划中获得 OBJECT#和 OBJECT_NAME 的映射关系:

```
SQL> desc DBA_HIST_SQL_PLAN
```

名称	是否为空? 类型
DBID	NOT NULL NUMBER
SQL_ID	NOT NULL VARCHAR2(13)
PLAN_HASH_VALUE	NOT NULL NUMBER
ID	NOT NULL NUMBER
OPERATION	VARCHAR2(30)
OPTIONS	VARCHAR2(30)
OBJECT_NODE	VARCHAR2(128)
OBJECT#	NUMBER
OBJECT_OWNER	VARCHAR2(30)
OBJECT_NAME	VARCHAR2(31)
OBJECT_ALIAS	VARCHAR2(65)
OBJECT_TYPE	VARCHAR2(20)
OPTIMIZER	VARCHAR2(20)
PARENT_ID	NUMBER
DEPTH	NUMBER
POSITION	NUMBER
SEARCH_COLUMNS	NUMBER
COST	NUMBER
CARDINALITY	NUMBER
BYTES	NUMBER
OTHER_TAG	VARCHAR2(35)
PARTITION_START	VARCHAR2(64)
PARTITION_STOP	VARCHAR2(64)
PARTITION_ID	NUMBER
OTHER	VARCHAR2(4000)
DISTRIBUTION	VARCHAR2(20)

CPU_COST	NUMBER
IO_COST	NUMBER
TEMP_SPACE	NUMBER
ACCESS_PREDICATES	VARCHAR2(4000)
FILTER_PREDICATES	VARCHAR2(4000)
PROJECTION	VARCHAR2(4000)
TIME	NUMBER
QBLOCK_NAME	VARCHAR2(31)
REMARKS	VARCHAR2(4000)
TIMESTAMP	DATE
OTHER_XML	CLOB

例如:

```
select object_owner,object_name,object# from DBA_HIST_SQL_PLAN where
sql_id='avwjc02vb10j4'
```

OBJECT_OWNER	OBJECT_NAME	OBJECT#
PARNASSUSDATA	TORDERDETAIL_HIS	78688

可以利用如下脚本获得较多 OBJECT_ID 与 OBJECT_NAME 的映射关系

```
Select * from
(select object_name,object# from DBA_HIST_SQL_PLAN
UNION select object_name,object# from GV$SQL_PLAN) V1 where V1.OBJECT# IS NOT NULL
minus select name,obj# from sys.obj$;
```

```
select obj#,dataobj#, object_name from WRH$_SEG_STAT_OBJ where object_name not in (select
name from sys.obj$) order by object_name desc;
```

另一个查询:

```
SELECT tab1.SQL_ID,
current_obj#,
tab2.sql_text
FROM DBA_HIST_ACTIVE_SESS_HISTORY tab1,
dba_hist_sqltext tab2
WHERE tab1.current_obj# NOT IN
```

```
(SELECT obj# FROM sys.obj$  
)  
AND current_obj#!=-1  
AND tab1.sql_id =tab2.sql_id(+);
```

注意以上方法仅仅在用户确实找不到所要恢复的数据表的任何定义信息时使用(即用户找任何对该应用模式设计有了解的人、脚本和文档),且由于依赖于 **AWR** 数据,所以并不准确。