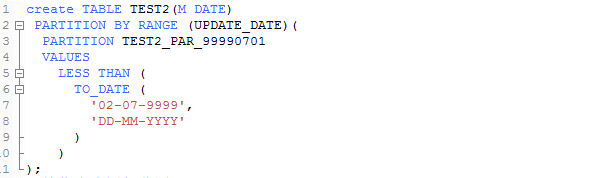
1. 创建表



注意：创建的表必须有一个足够大的初始分区，因为后续增加分区都是在当前这个分区上面分裂而来的。

1. 创建存储过程

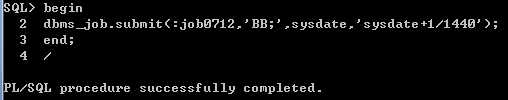


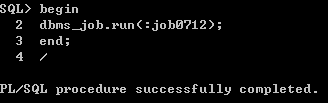
对应图中存储过程详细：

|  |
| --- |
| CREATE OR REPLACE  PROCEDURE BB(TABLE\_NAME IN VARCHAR2,BEFORE\_DAY IN NUMBER,AFTER\_DAY IN NUMBER) AS  PAR\_NAME VARCHAR2(50);--要创建的分区的名称  PAR\_NAME2 VARCHAR2(400);--要创建的分区的名称  V\_DAY DATE;--分区对应的日期  PAR\_NAME\_MAX VARCHAR2(50);--在创建表的时候一开始加入的唯一一个分区名称，对应的这个分区要设置的很大，以便在本存储过程中分裂该分区  --TABLE\_NAME VARCHAR2(50);--本存储过程要动态创建分区的对应的表名（该表必须有一个足够大的初始分区）  PAR\_EXISTS INTEGER;--记录即将创建分区是否已经存在  v\_high\_value varchar2(255);  v\_partition\_max\_date date;  CURSOR C\_CUR(BEFOREDAY NUMBER,AFTERDAY NUMBER) IS  --SELECT (SYSDATE + (ROWNUM - 1)) S\_DATE FROM DUAL CONNECT BY ROWNUM <= 60;  SELECT (SYSDATE-BEFOREDAY)+ROWNUM-1 as S\_DATE FROM DUAL CONNECT BY ROWNUM<=TRUNC((SYSDATE+AFTERDAY)-(SYSDATE-BEFOREDAY));  CURSOR C\_CUR2(TABLENAME VARCHAR2) IS  select utp.table\_name,utp.tablespace\_name,utp.partition\_name, utp.high\_value, utp.high\_value\_length, utp.partition\_position  from user\_tab\_partitions utp where utp.table\_name = TABLENAME and utp.high\_value\_length<>8 order by utp.partition\_position ASC;  BEGIN  PAR\_NAME\_MAX := concat(TABLE\_NAME,'\_PAR\_MAX');  OPEN C\_CUR(BEFORE\_DAY,AFTER\_DAY);--打开游标  loop  FETCH C\_CUR INTO V\_DAY;  EXIT WHEN C\_CUR%NOTFOUND;    --要创建的分区名称  PAR\_NAME := concat(TABLE\_NAME,concat('\_PAR\_',TO\_CHAR(V\_DAY, 'YYYYMMDD')));    for utp in C\_CUR2(TABLE\_NAME) loop  v\_high\_value := substr(utp.high\_value,11,10);  v\_partition\_max\_date := to\_date(v\_high\_value,'YYYY-MM-DD');  IF V\_DAY+1<v\_partition\_max\_date THEN  PAR\_NAME\_MAX := TABLE\_NAME||'\_PAR\_'||TO\_CHAR(v\_partition\_max\_date-1,'yyyymmdd');  EXIT;  END IF;  END loop;  PAR\_EXISTS :=0;  SELECT COUNT(\*) INTO PAR\_EXISTS FROM USER\_TAB\_PARTITIONS WHERE TABLE\_NAME = TABLE\_NAME AND PARTITION\_NAME = PAR\_NAME;  --判断要创建的分区是否已经存在了  IF PAR\_EXISTS=0 THEN  --SYS.DBMS\_OUTPUT.PUT\_LINE('alter table '||TABLE\_NAME||' split partition '||PAR\_NAME\_MAX||' AT (TO\_DATE('''||TO\_CHAR(V\_DAY+1,'yyyymmdd')||''',''yyyymmdd'')) INTO (PARTITION '||PAR\_NAME||',PARTITION '||PAR\_NAME\_MAX||')');  --EXECUTE IMMEDIATE 'alter table '||TABLE\_NAME||' split partition '||PAR\_NAME\_MAX||' AT (TO\_DATE('''||TO\_CHAR(V\_DAY,'yyyymmdd')||''',''yyyymmdd'')) INTO (PARTITION '||PAR\_NAME||',PARTITION '||PAR\_NAME\_MAX||')';  --执行：分裂分区  EXECUTE IMMEDIATE 'alter table '||TABLE\_NAME||' split partition '||PAR\_NAME\_MAX||' AT (TO\_DATE('''||TO\_CHAR(V\_DAY+1,'yyyymmdd')||''',''yyyymmdd'')) INTO (PARTITION '||PAR\_NAME||',PARTITION '||PAR\_NAME\_MAX||') UPDATE global INDEXES';  END IF;  END loop;  close C\_CUR;--关闭游标  commit;  END BB ; |

1. 在oracle的命令行界面执行下面操作，启动一个job：







说明：

一些常用命令：

--给指定表添加分区

alter table TEST3 add partition part\_aaa values less than(TO\_DATE('01-JUL-2009','DD-MON-YYYY'))

--分裂分区

alter table TEST2 split partition ORD\_ACT\_PART02 AT (TO\_DATE('01-JUL-2009','DD-MON-YYYY')) INTO (PARTITION P1,PARTITION ORD\_ACT\_PART02);

--删除指定表的指定分区

ALTER TABLE TEST3 DROP PARTITION part\_aaa;

--查看某张表所有分区信息

SELECT \* FROM USER\_TAB\_PARTITIONS WHERE TABLE\_NAME = 'TEST2'

--查看指定分区记录

SELECT \* FROM TEST2 PARTITION (P1)

--查看定时任务job相关信息

SELECT job,broken,what,interval,t.\* from user\_jobs t;

SELECT job,broken,what,interval,t.next\_sec ,SYSDATE from user\_jobs t; --对应broken为N的时候job才为运行状态

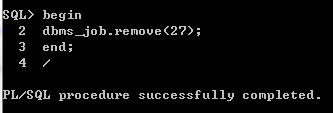
SELECT \* from dba\_jobs order by job;

--删除job

A、查找job对应的id

SELECT job,broken,what,interval,t.next\_sec ,SYSDATE from user\_jobs t;

B、根据查找到的job的id来删除指定的job



--在删除分区表的时候，用下面的语句，否则分区表的信息会放到oracle的垃圾站中去，会导致存储过程中判断将要创建的分区是否存在的时候会有问题（会把垃圾站中的也统计进去，而垃圾站中的tableName不是指定的表名如TEST2，但是在统计的时候也会统计进去），故在删除分区表的时候要彻底删除，如下面删除表的语法

DROP TABLE TEST2 PURGE

--下面的语句是清空oracle的垃圾站数据

PURGE recyclebin;