Oracle Job定时任务的使用详解

oracle中的job能为你做的就是在你规定的时间格式里执行存储过程，定时执行一个任务 。下面是一个小案例，定时每15分钟向一张表插入一条数据

一

1.创建一张测试表

```
-- Create table
create table A8
(
  a1 VARCHAR2(500)
)
tablespace DSP_DATA
  pctfree 10
  initrans 1
  maxtrans 255
  storage
  (
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
  );
```

2.创建存储过程 实现向测试表插入数据

```
create or replace procedure proc_add_test as
begin
  insert into a8 values (to_char(sysdate, 'yyyy-mm-dd hh:mi')));/*向测试表插入数据*/
  commit;
end;
```

3.创建job定时任务 实现自动调用存储过程(当前时间 17:03)

```
declare
  job number;
BEGIN
  DBMS_JOB.SUBMIT(
        JOB => job,  /*自动生成JOB_ID*/
        WHAT => 'proc_add_test;',  /*需要执行的存储过程名称或sql语句*/
        NEXT_DATE => sysdate+3/(24*60),  /*初次执行时间-下一个3分钟*/
        INTERVAL => 'trunc(sysdate,''mi'')+1/(24*60)' /*每隔3分钟执行一次*/
        );
  commit;
end;
```

4.也就是应该从17:06开始 每隔1分钟执行一次存储过程 下面是截止17:12分的测试表的数据

	A1
2	2017-06-22 05:06
3	2017-06-22 05:07
4	2017-06-22 05:08
5	2017-06-22 05:09
7	2017-06-22 05:10
1	2017-06-22 05:11
6	2017-06-22 05:12

二

1.可以通过查询系统表查看该job信息

```
select * from user_jobs;
```

	JOB	LOG_USER	PRIV_USER	SCHEMA_USER	LAST_DATE	LAST_SEC	THIS_DATE	THIS_SEC	NEXT_DATE	NEXT_SEC	TOTAL_T
1	740	DSP0612	DSP0612	DSP0612	2017/6/22 17:24:04	17:24:04			2017/6/22 17:25:00	17:25:00	

可以通过job的id手动执行或删除job

2.手动sql调用job （直接调用job可以忽略开始时间）

```
 begin
   DBMS_JOB.RUN(40); /*40 job的id*/
 end;
```

3.删除任务

```
begin
  /*删除自动执行的job*/
  dbms_job.remove(40);
end;
```

4.停止job

```
dbms.broken(job, broken, nextdate);
dbms_job.broken(v_job,true,next_date);        /*停止一个job,里面参数true也可是false, next_date（某一时刻停止）也可是sysdate
```

5.修改间隔时间

```
dbms_job.interval(job, interval);
```

6.修改下次执行时间

```
dbms_job.next_date(job, next_date);
```

7.修改要执行的操作

```
dbms_job.what (jobno, 'sp_fact_charge_code;') ;   --修改某个job名
```

三 其他知识

1.存job信息的表user_jobs主要字段说明

列名	数据类型	解释
JOB	NUMBER	任务的唯一标示号
LOG_USER	VARCHAR2(30)	提交任务的用户
PRIV_USER	VARCHAR2(30)	赋予任务权限的用户
SCHEMA_USER	VARCHAR2(30)	对任务作语法分析的用户模式
LAST_DATE	DATE	最后一次成功运行任务的时间
LAST_SEC	VARCHAR2(8)	如HH24:MM:SS格式的last_date日期的小时，分钟和秒
THIS_DATE	DATE	正在运行任务的开始时间，如果没有运行任务则为null
THIS_SEC	VARCHAR2(8)	如HH24:MM:SS格式的this_date日期的小时，分钟和秒
NEXT_DATE	DATE	下一次定时运行任务的时间
NEXT_SEC	VARCHAR2(8)	如HH24:MM:SS格式的next_date日期的小时，分钟和秒
TOTAL_TIME	NUMBER	该任务运行所需要的总时间，单位为秒
BROKEN	VARCHAR2(1)	标志参数，Y标示任务中断，以后不会运行
INTERVAL	VARCHAR2(200)	用于计算下一运行时间的表达式
FAILURES	NUMBER	任务运行连续没有成功的次数
WHAT	VARCHAR2(2000)	执行任务的PL/SQL块

2.**INTERVAL参数常用值示例**

1. 每天午夜12点　　''TRUNC(SYSDATE + 1)''
2. 每天早上8点30分　　''TRUNC(SYSDATE + 1) + （8*60+30）/(24*60)''
3. 每星期二中午12点　　''NEXT_DAY(TRUNC(SYSDATE), ''''TUESDAY'''') + 12/24''
4. 每个月第一天的午夜12点　''TRUNC(LAST_DAY(SYSDATE) + 1)''
5. 每个季度最后一天的晚上11点 ''TRUNC(ADD_MONTHS(SYSDATE + 2/24, 3), ''Q'') -1/24''
6. 每星期六和日早上6点10分　''TRUNC(LEAST(NEXT_DAY(SYSDATE, ''''SATURDAY''''), NEXT_DAY(SYSDATE, ''SUNDAY'')) + （6×60+10）/（24×60）''
7. 每3秒钟执行一次　　'sysdate+3/(24*60*60)'
8. 每2分钟执行一次　　'sysdate+2/(24*60)'
9.
10. 1:每分钟执行
11. Interval => TRUNC(sysdate,'mi') + 1/ (24*60) --每分钟执行
12. interval => 'sysdate+1/ (24*60) ' --每分钟执行
13. interval => 'sysdate+1' --每天
14. interval => 'sysdate+1/24' --每小时
15. interval => 'sysdate+2/24*60' --每2分钟
16. interval => 'sysdate+30/24*60*60' --每30秒
17. 2:每天定时执行
18. Interval => TRUNC(sysdate+1) --每天凌晨0点执行
19. Interval => TRUNC(sysdate+1)+1/24 --每天凌晨1点执行
20. Interval => TRUNC(SYSDATE+1)+(8*60+30)/(24*60) --每天早上8点30分执行
21. 3:每固定时执行
22. Interval => TRUNC(next_day(sysdate,'星期一'))+1/24 --每周一凌晨1点执行
23. Interval => TRUNC(next_day(sysdate,1))+2/24 --每周一凌晨2点执行
24. 4:每月定时执行
25. Interval =>TTRUNC(LAST_DAY(SYSDATE)+1) --每月1日凌晨0点执行
26. Interval =>TRUNC(LAST_DAY(SYSDATE))+1+1/24 --每月1日凌晨1点执行
27. 5:每季度定时执行
28. Interval => TRUNC(ADD_MONTHS(SYSDATE,3),'q') --每季度的第一天凌晨0点执行
29. Interval => TRUNC(ADD_MONTHS(SYSDATE,3),'q') + 1/24 --每季度的第一天凌晨1点执行
30. Interval => TRUNC(ADD_MONTHS(SYSDATE+ 2/24,3),'q')-1/24 --每季度的最后一天的晚上11点执行
31. 6:每半年定时执行
32. Interval => ADD_MONTHS(trunc(sysdate,'yyyy'),6)+1/24 --每年7月1日和1月1日凌晨1点
33. 7:每年定时执行
34. Interval =>ADD_MONTHS(trunc(sysdate,'yyyy'),12)+1/24 --每年1月1日凌晨1点执行

分类： oracle

懂子
关注 - 1
粉丝 - 4
+加关注

» 下一篇： Spring整合CXF发布及调用WebService

posted @ 2017-06-23 09:53 懂子 阅读(43892) 评论(1) 编辑 收藏