Can I partition an already existing table in oracle?

I have a table in my oradb, whitch already has a lot of rows.

I want to partition this table by month by not recreating it. So I don't want to lose the data from the table.

I want this partition type:

```
PARTITION BY RANGE (date_column)
INTERVAL (NUMTOYMINTERVAL (1,'MONTH'))
         (partition transaction_old values less than (to_date('01-01-2015','DD-MM-
YYYY')));
```

How can I set this to an already existing table? I wanted to use the **alter table** function, but it does not work:

```
alter table mytable
    PARTITION BY RANGE (date_column)
    INTERVAL (NUMTOYMINTERVAL (1,'MONTH'))
            ( partition transaction_old values less than (to_date('01-01-2015','DD-MM-
YYYY') ));
```

Is there any solution for my problem?

oracle database-partitioning partition

asked Nov 20 '15 at 12:22
victorio
1,659 6 41 70

dbms_redefinition package. Check answer here – Tatiana Nov 20 '15 at 12:30

just one warning, you might need to have dba privs to execute dbms_redefinition. You're only database user having connect,resource privs, when you can not use it (even on your own table). – ibre5041 Nov 20 '15 at 13:52

2 Answers

You cannot directly **partition** an existing non-partitioned table. You will need to create an interim table/new table depending on the following methods to partition:

- **DBMS_REDEFINITION**
  1. Create a Partitioned Interim Table
  2. Start the Redefinition Process
  3. Create Constraints and Indexes (Dependencies)
  4. Complete the Redefinition Process
- **EXCHANGE PARTITION**
  1. Create a Partitioned Destination Table
  2. EXCHANGE PARTITION
  3. SPLIT PARTITION (If required to split single large partition into smaller partitions)

answered Nov 20 '15 at 12:49
Lalit Kumar B
31.8k 8 44 78

probably the easiest, safest way will be to use dbms_redefinition - it'll copy over the grants, indexes, constraints, everything.

```
Table P is the parent table.

Table T1 is the existing non-partitioned table.
Table T2 is the new partitioned table.

At the end, we are left with P and T1 - T1 being partitioned.


ops$tkyte%ORA10GR2> create table p ( x primary key )
  2  as
  3  select user_id from all_users;

Table created.

ops$tkyte%ORA10GR2>
ops$tkyte%ORA10GR2> create table t1
  2  as
  3  select * from all_users
  4  /

Table created.

ops$tkyte%ORA10GR2> alter table t1 add constraint t1_pk primary key(user_id);

Table altered.

ops$tkyte%ORA10GR2> alter table t1 add constraint t1_fk foreign key(user_id) references
p(x);

Table altered.

ops$tkyte%ORA10GR2>
ops$tkyte%ORA10GR2> create table t2
  2  ( username varchar2(30),
  3    user_id  number,
  4    created  date
  5  )
  6  partition by hash(user_id) partitions 8;

Table created.

ops$tkyte%ORA10GR2> exec dbms_redefinition.can_redef_table( user, 'T1' );

PL/SQL procedure successfully completed.

ops$tkyte%ORA10GR2>
ops$tkyte%ORA10GR2> exec dbms_redefinition.start_redef_table( user, 'T1', 'T2' );

PL/SQL procedure successfully completed.

ops$tkyte%ORA10GR2>
ops$tkyte%ORA10GR2>
ops$tkyte%ORA10GR2> variable nerrors number
ops$tkyte%ORA10GR2> begin
  2          dbms_redefinition.copy_table_dependents
  3          ( user, 'T1', 'T2',
  4            copy_indexes => dbms_redefinition.cons_orig_params,
  5            num_errors => :nerrors );
  6  end;
  7  /

PL/SQL procedure successfully completed.

ops$tkyte%ORA10GR2> exec dbms_redefinition.finish_redef_table( user, 'T1', 'T2' );

PL/SQL procedure successfully completed.

ops$tkyte%ORA10GR2>
ops$tkyte%ORA10GR2> select dbms_metadata.get_ddl( 'TABLE', 'T1' ) from dual;

DBMS_METADATA.GET_DDL('TABLE','T1')
-------------------------------------------------------------------------------

  CREATE TABLE "OPS$TKYTE"."T1"
   (    "USERNAME" VARCHAR2(30) CONSTRAINT "SYS_C0026838" NOT NULL ENABLE NOVALIDA
TE,
        "USER_ID" NUMBER CONSTRAINT "SYS_C0026839" NOT NULL ENABLE NOVALIDATE,
        "CREATED" DATE CONSTRAINT "SYS_C0026840" NOT NULL ENABLE NOVALIDATE,
         CONSTRAINT "T1_PK" PRIMARY KEY ("USER_ID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE NOVALIDATE,
         CONSTRAINT "T1_FK" FOREIGN KEY ("USER_ID")
          REFERENCES "OPS$TKYTE"."P" ("X") ENABLE NOVALIDATE
   ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
  STORAGE(
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"
  PARTITION BY HASH ("USER_ID")
 (PARTITION "SYS_P1017"
    TABLESPACE "USERS",
  PARTITION "SYS_P1018"
    TABLESPACE "USERS",
  PARTITION "SYS_P1019"
    TABLESPACE "USERS",
  PARTITION "SYS_P1020"
    TABLESPACE "USERS",
  PARTITION "SYS_P1021"
    TABLESPACE "USERS",
  PARTITION "SYS_P1022"
    TABLESPACE "USERS",
  PARTITION "SYS_P1023"
    TABLESPACE "USERS",
  PARTITION "SYS_P1024"
    TABLESPACE "USERS")


ops$tkyte%ORA10GR2> select constraint_name, constraint_type from user_constraints where
table_name = 'T1';

CONSTRAINT_NAME                C
------------------------------ -
SYS_C0026838                   C
SYS_C0026839                   C
SYS_C0026840                   C
T1_PK                          P
T1_FK                          R
```

Source

answered Nov 20 '15 at 12:40

Tatiana
1,294 3 15