

如何将一个普通的非分区表进行分区 (Doc ID 1985005.1)

文档内容

- [用途](#)
- [适用范围](#)
- [详细信息](#)
 - [A. 通过 Export/import 方法](#)
 - [B. 通过 Insert with a subquery 方法](#)
 - [C. 通过 Partition Exchange 方法](#)
 - [D. 通过 DBMS_REDEFINITION 方法](#)
 - [E. MODIFY clause added to the ALTER TABLE SQL Statement. \(From 12.2\)](#)
- [参考](#)

适用于:

Oracle Database - Enterprise Edition
Oracle Database Cloud Schema Service - 版本 N/A 和更高版本
Oracle Database Exadata Cloud Machine - 版本 N/A 和更高版本
Oracle Cloud Infrastructure - Database Service - 版本 N/A 和更高版本
Oracle Database Exadata Express Cloud Service - 版本 N/A 和更高版本
本文档所含信息适用于所有平台

用途

本文描述了将非分区表进行分区的四种可能的方法。

这些步骤也可以用来改变其他分区特征，比如在分区表中添加 subpartition。

虽然从应用程序的角度来看，分区是透明的，分区表与非分区表是等价的；当访问分区表时，不需要修改应用程序的 SQL，但是注意如下 Note 提到的问题

[Note 1518567.1](#) Row Movement Common Questions and Problems.

本文不讨论分区策略的选择和分区带来的好处，因为这些信息可以在如下文档中找到：

[Note 1390871.1](#) Purpose and Benefits/Advantages/Uses of Table Partitioning

[Note 1563143.1](#) How Should I Partition My Tables?

咨询问题，获得帮助，分享您对本文的经验

您想和其他 Oracle 客户，Oracle 雇员和行业专家对本话题进行更深入的探讨么？

[点击这里来加入讨论, 您可以咨询问题, 从其他人那里获得帮助, 分享您对此文章的经验。](#)

探索其他文章和主题, 请点击[这里](#)来访问 My Oracle Support 社区 Database Datawarehousing 主页。

适用范围

DBA 以及需要将非分区表进行分区的开发人员。

详细信息

您可以任选如下五种方法的一种来对非分区表进行分区：

- A) 通过 Export/import 方法
- B) 通过 Insert with a subquery 方法
- C) 通过 Partition Exchange 方法
- D) 通过 DBMS_REDEFINITION 方法
- E) ALTER TABLE 命令的MODIFY语法 （从 12.2 开始）

以上四种方法都会从一个已经存在的非分区表创建一个分区表。尽管我们已经对这四种方法的维护时间预期做了倒序排列，但实际维护时间针对不同情况有所不同。

A. 通过 Export/import 方法

这种方法的实现是先 export 一个非分区表，创建一个新的分区表，然后 import 数据到新创建的分区表中。

1) Export 您的非分区表：

```
$ exp usr/pswd tables=numbers file=exp.dmp
```

2) Drop 掉该非分区表：

```
SQL> drop table numbers;
```

3) 重新创建该表成为一个分区表：

```
SQL> create table numbers (qty number(3), name varchar2(15)) partition by range (qty) (partition p1 values less than (501), partition p2 values less than (maxvalue));
```

4) 通过 import 的 ignore=y 方式来还原备份的数据：

```
$ imp usr/pswd file=exp.dmp ignore=y
```

ignore=y 语句会让 import 忽略掉表的创建，直接加载所有数据。

如果使用 Data Pump export/import (expdp/impdp) 您可以采用 impdp 的 table_exists_action 选项，例如 table_exists_action = APPEND 或者 table_exists_action = REPLACE。

您也可以参考 [Note 552424.1](#) Export/Import Data Pump Parameter ACCESS_METHOD - How to Enforce a Method of Loading and Unloading Data?

B. 通过 Insert with a subquery 方法

1) 创建一个分区表：

```
SQL> create table partbl (qty number(3), name varchar2(15)) partition by range (qty) (partition p1 values less than (501), partition p2 values less than (maxvalue));
```

2) 将原来非分区表中的数据通过子查询 insert 到新创建的分区表中：

```
SQL> insert into partbl (qty, name) select * from origtbl;
```

3) 如果您想让新建的分区表与原表名相同，那么 drop 掉原来的非分区表然后重命名新表：

```
SQL> drop table origtbl;  
SQL> alter table partbl rename to origtbl;
```

您可以通过 direct path insert 和利用并行来改善 insert 的性能。如下的例子演示了如何实现并且如何从执行计划中来验证。

传统的 insert

```
SQL> insert into partbl (qty, name) select * from origtbl;
```

```
-----  
| Id | Operation                      | Name |  
-----  
|  0 | INSERT STATEMENT                |      |  
|  1 |  LOAD TABLE CONVENTIONAL       |      |  
|  2 |    TABLE ACCESS FULL           | ORIGTBL |  
-----
```

Direct load insert 方式

```
SQL> insert /*+APPEND*/ into partbl (qty, name) select * from origtbl;
```

```
-----  
| Id | Operation                      | Name |  
-----  
|  0 | INSERT STATEMENT                |      |  
|  1 |  LOAD AS SELECT                 |      |  
|  2 |    TABLE ACCESS FULL           | ORIGTBL |  
-----
```

Direct load insert 并且在查询部分开启并行

```
SQL> insert /*+APPEND PARALLEL*/ into partbl (qty, name) select * from origtbl;
```

```
-----  
| Id | Operation                      | Name |  
-----  
|  0 | INSERT STATEMENT                |      |  
|  1 |  LOAD AS SELECT                 |      |  
|  2 |    PX COORDINATOR               |      |  
-----
```

```

| 3 | PX SEND QC (RANDOM) | :TQ10000 |
| 4 | PX BLOCK ITERATOR | |
|* 5 | TABLE ACCESS FULL| ORIGTBL |
-----

```

注意以上执行计划中 LOAD AS SELECT 在 PX COORDINATOR 的上面。

Direct load insert 并且在查询部分和 insert 部分都开启并行

```
SQL>alter session enable parallel dml;
```

```
SQL> insert /*+APPEND PARALLEL*/ into partbl (qty, name) select * from origtbl;
```

```

-----
| Id | Operation | Name |
-----
| 0 | INSERT STATEMENT | |
| 1 | PX COORDINATOR | |
| 2 | PX SEND QC (RANDOM) | :TQ10000 |
| 3 | LOAD AS SELECT | |
| 4 | PX BLOCK ITERATOR | |
|* 5 | TABLE ACCESS FULL| ORIGTBL |
-----

```

注意在以上执行计划中 LOAD AS SELECT 在 PX COORDINATOR 的下面。

另外一种可选的方式是直接通过 select 来创建新的分区表：一次性创建新的分区表并且加载数据。执行计划同时显示 direct path load 并且 dml 以及 select 部分全部并行。

```
SQL>alter session enable parallel dml;
```

```
SQL> create table partbl (qty, name) partition by range (qty) (partition p1 values less than (501),partition p2
values less than (maxvalue))
2 as select /*+PARALLEL*/ * from origtbl;
```

```

-----
| Id | Operation | Name |
-----
| 0 | CREATE TABLE STATEMENT | |
| 1 | PX COORDINATOR | |
| 2 | PX SEND QC (RANDOM) | :TQ10000 |
| 3 | LOAD AS SELECT | |
| 4 | PX BLOCK ITERATOR | |
|* 5 | TABLE ACCESS FULL| ORIGTBL |
-----

```

C. 通过 Partition Exchange 方法

ALTER TABLE EXCHANGE PARTITION 可以通过交换数据和索引 segment 来将一个分区（或子分区）转换成一个非分区表，也可以将一个非分区表转换成一个分区表的分区（或子分区）。除了需要更新索引以外，ALTER TABLE ... EXCHANGE PARTITION 命令是一个字典操作不需要数据移动。更多关于此方法的信息参见 Oracle 联机文档（比如 11.2）和 [Note 198120.1](#)。

此方法简要步骤如下：

- 1) 根据所需的分区来创建新的分区表
- 2) 保持需要交换的非分区表与分区表的分区有相同的结构，并且确保您需要交换的非分区表具有您想要交换的内容
- 3) 执行：Alter table exchange partition partition_name with table exchange table

注意在交换过程中，所有交换的数据必须满足分区表的分区定义，否则如下错误将抛出：ORA-14099: all rows in table do not qualify for specified partition.

这是因为默认情况下分区交换是有校验的。

例子（基于 SCOTT 示例 schema）

本例创建了与分区表 p_emp 的分区相同结构的交换表。

```
SQL> CREATE TABLE p_emp
2 (sal NUMBER(7,2))
3 PARTITION BY RANGE(sal)
4 (partition emp_p1 VALUES LESS THAN (2000),
5 partition emp_p2 VALUES LESS THAN (4000));
```

Table created.

```
SQL> SELECT * FROM emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30

7782	CLARK	MANAGER	7839	09-JUN-81	2450	10
7788	SCOTT	ANALYST	7566	19-APR-87	3000	20
7839	KING	PRESIDENT		17-NOV-81	5000	10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100	20
7900	JAMES	CLERK	7698	03-DEC-81	950	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	20
7934	MILLER	CLERK	7782	23-JAN-82	1300	10

14 rows selected.

```
SQL> CREATE TABLE exchtab1 as SELECT sal FROM emp WHERE sal<2000;
```

Table created.

```
SQL> CREATE TABLE exchtab2 as SELECT sal FROM emp WHERE sal BETWEEN 2000 AND 3999;
```

Table created.

```
SQL> alter table p_emp exchange partition emp_p1 with table exchtab1;
```

Table altered.

```
SQL> alter table p_emp exchange partition emp_p2 with table exchtab2;
```

Table altered.

D. 通过 DBMS_REDEFINITION 方法

详情参见：

[Note 472449.1](#) How To Partition Existing Table Using DBMS_Redefinition

[Note 1481558.1](#) DBMS_REDEFINITION: Case Study for a Large Non-Partition Table to a Partition Table with Online Transactions occurring

[Note 177407.1](#) How to Re-Organize a Table Online

E. MODIFY clause added to the ALTER TABLE SQL Statement. (From 12.2)

从 12.2 开始，可以使用 Alter Table <table_name> MODIFY 把非分区表转换成分区表。

更多详情请参考 <https://docs.oracle.com/en/database/oracle/oracle-database/12.2/vldbg/evolve-nopartition-table.html#GUID-5FDB7D59-DD05-40E4-8AB4-AF82FA0D0FE5>

参考

[NOTE:1481558.1](#) - DBMS_REDEFINITION: Case Study for a Large Non-Partition Table to a Partition Table with Online Transactions Occuring

[NOTE:177407.1](#) - How to Re-Organize a Table Online

[NOTE:72332.1](#) - Diagnosing ORA-14097 On Alter Table Exchange Partition

[NOTE:1518567.1](#) - FAQ: Row Movement Common Questions and Problems on Partitioned Tables

[NOTE:472449.1](#) - How To Partition Existing Table Using DBMS_REDEFINITION

[NOTE:1390871.1](#) - Purpose and Benefits/Advantages/Uses of Table Partitioning

Didn't find what you are looking for?