

## oracle truncate 表分区 空间释放问题

参考：[https://blog.csdn.net/dba\\_waterbin/article/details/8457350](https://blog.csdn.net/dba_waterbin/article/details/8457350)  
[https://docs.oracle.com/cd/B19306\\_01/server.102/b14200/statements\\_3001.htm](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_3001.htm)  
<https://blog.csdn.net/enenwenwen/article/details/29564369>

### ② 空间释放问题

其实空间等都已经释放了，但数据字典没有被更新，  
例如你查dba\_segments视图，发现这个分区的bytes其实还为原来的大小  
我们可执行alter table \*\*\*\* allocate extent即可更新数据字典为正常状态  
例如针对范围分区如下操作：  
alter table \*\*\* modify partition PART\_\*\*\* allocate extent;

```
ALTER TABLE print_media_demo  
TRUNCATE PARTITION p1 DROP STORAGE;
```

#### ① 马上回收空间：

```
alter table table_name truncate partition partition_name drop storage;
```

#### ② 同时维护全局索引：

```
alter table table_name drop partition partition_name update global indexes;
```

#### ② 对全局索引的作用

大分区表truncate partition后，需要对全局索引进行维护，否则，global index会变成unusable  
问题介绍：

① 在drop partition时，为了维护global索引，要加update indexes或是update global indexes条件  
请问，大家研究过这两个条件的区别吗？

答：UPDATE GLOBAL INDEXES只维护全局索引

UPDATE INDEXES同时维护全局和本地索引

对于DROP/TRUNCATE PARTITION而言，二者没有太大的区别

对于MERGE和SPLIT PARTITION, 你就可以看到二者的区别了

虽然index是变得valid了，但是index的空间没有释放

因为该操作不等于REBUILD, 只是在进行DDL的时候，同步维护索引信息而已

工业环境的案例介绍：

② 我今天对分区表的一个分区做了TRUNCATE, 这个分区有一个普通唯一索引和本地索引，

TRUNCATE的时候没有用UPDATE GLOBAL INDEXES 那个命令，结果导致报：

ORA-01502: index BILL.IDX\_CHARGE\_D\_591\_0712\_SID or partition of such index is in unusable state

这是因为，truncate partition不加update global indexes, 会导致全局索引失效(unusable)。

然后，我只好：

```
alter index bill.IDX_CHARGE_D_591_0712_SID parallel 10 nologging rebuild 来整个的重建，13亿记录的大表
```

后来接着晚上有人继续插入这个表的时候，告诉我慢的要命，本来一个小时至少可以跑完400万条记录，现在3个小时了才跑130万

我马上想到会不会是本地索引问题，因为我听说虽然分区交换或者TRUNCATE对局部索引没影响，

但是实际上是有问题的，还是重建的好：

```
alter index bill.UNQ_RATING_CHARGE_D_591_0712 rebuild partition PART_20
```

把这个刚才我TRUNCATE的分区的涉及到的局部索引重新建了一下

结果立马见效果了，10分钟跑了200万条记录，600万条记录在20分钟全部跑好!比以前同期跑的还快一点

半夜被叫起来干活了

奇怪，如下写法怎么半天都执行不好

```
alter table bill.recur_rating_charge_d_591_0712 truncate partition PART_21 update global indexes ;
```

```
select count(*) from bill.recur_rating_charge_d_591_0712 partition(PART_21)
```

数据始终不变

但是我看看v\$session\_longops看到这个SID很快就做好事了，

而我看表分区记录始终在

我晕，只好采用老办法，杀掉会话后，

```
alter table bill.RECUR_RATING_CHARGE_d_591_0712 truncate partition PART_20不加update global indexes
```

然后分别维护了普通索引和局部索引，这次加NOLOGGING和PARALLEL 8，也很快，3亿的大表，维护普通索引只花了200秒

```
alter index bill.IDX_CHARGE_D_591_0712_SID rebuild parallel 8 nologging ;
```

```
alter index bill.UNQ_RATING_CHARGE_D_591_0712 rebuild partition PART_21 parallel 8 nologging;
```

猜测原因：

truncate partition PART\_20后，这个分区的和这个分区上的本地索引的统计信息是不会更新也不会丢失

当我往这个分区插入数据的时候，执行计划是根据错误的统计信息生成的，所以会很慢

当我rebuild index partition PART\_20 后，会导致这个索引的统计信息丢失，

而我的执行计划就有可能改变了，所以我的插入变快了

总结：

当你truncate后应该立即对这个分区做分析cascade => true(增加对索引的统计信息)，

同时rebuild global index 并分析global index才对

(二) 空间释放问题

其实空间等都已经释放了，但数据字典没有被更新，

例如你查dba\_segments视图，发现这个分区的bytes其实还为原来的大小

我们可执行alter table \*\*\*\* allocate extent即可更新数据字典为正常状态

例如针对范围分区如下操作：

```
alter table *** modify partition PART_*** allocate extent;
```

#### ① 马上回收空间：

```
alter table table_name truncate partition partition_name drop storage;
```

② 同时维护全局索引：

```
alter table table_name drop partition partition_name update global indexes;
```

(二) 对全局索引的作用

大分区表truncate partition后，需要对全局索引进行维护，否则，global index会变成unusable

问题介绍：

① 在drop partition时，为了维护global索引，要加update indexes或是update global indexes条件

请问，大家研究过这两个条件的区别吗？

答： UPDATE GLOBAL INDEXES只维护全局索引

UPDATE INDEXES同时维护全局和本地索引

对于DROP/TRUNCATE PARTITION而言，二者没有太大的区别

对于MERGE和SPLIT PARTITION,你就可以看到二者的区别了

虽然index是变得valid了，但是index的空间没有释放

因为该操作不等于REBUILD,只是在进行DDL的时候，同步维护索引信息而已

工业环境的案例介绍：

② 我今天对分区表的一个分区做了TRUNCATE,这个分区有一个普通唯一索引和本地索引，

TRUNCATE的时候没有用UPDATE GLOBAL INDEXES 那个命令，结果导致报：

ORA-01502: index BILL.IDX\_CHARGE\_D\_591\_0712\_SID or partition of such index is in unusable state

这是因为，truncate partition不加update global indexes,会导致全局索引失效(unusable)。

然后，我只好：

```
alter index bill.IDX_CHARGE_D_591_0712_SID parallel 10 nologging rebuild 来整个的重建，13亿记录的大表
```

后来接着晚上有人继续插入这个表的时候，告诉我慢的要命，本来一个小时至少可以跑完400万条记录，现在3个小时了才跑130万

我马上想到会不会是本地索引问题，因为我听说虽然分区交换或者TRUNCATE对局部索引没影响，

但是实际上是有点问题的，还是重建的好：

```
alter index bill.UNQ_RATING_CHARGE_D_591_0712 rebuild partition PART_20
```

把这个刚才我TRUNCATE的分区的涉及到的局部索引重新建了一下

结果立马见效果了，10分钟跑了200万条记录，600万条记录在20分钟全部跑好!比以前同期跑的还快一点

半夜被叫起来干活了

奇怪，如下写法怎么半天都执行不好

```
alter table bill.recur_rating_charge_d_591_0712 truncate partition PART_21 update global indexes ;
```

```
select count(*) from bill.recur_rating_charge_d_591_0712 partition(PART_21)
```

数据始终不变

但是我看看v\$session\_longops看到这个SID很快就做好事了，

而我看表分区记录始终在

我晕，只好采用老办法，杀掉会话后，

```
alter table bill.RECUR_RATING_CHARGE_d_591_0712 truncate partition PART_20不加update global indexes
```

然后分别维护了普通索引和局部索引，这次加NOLOGGING和PARALLEL 8 ,也很快，3亿的大表，维护普通索引只花了200秒

```
alter index bill.IDX_CHARGE_D_591_0712_SID rebuild parallel 8 nologging ;
```

```
alter index bill.UNQ_RRATING_CHARGE_D_591_0712 rebuild partition PART_21 parallel 8 nologging;
```

猜测原因：

truncate partition PART\_20后，这个分区的和这个分区上的本地索引的统计信息是不会更新也不会丢失

当我往这个分区插入数据的时候，执行计划是根据错误的统计信息生成的，所以会很慢

当我rebuild index partition PART\_20 后，会导致这个索引的统计信息丢失，

而我的执行计划就有可能改变了，所以我的插入变快了

总结：

当你truncate后应该立即对这个分区做分析cascade => true(增加对索引的统计信息)，

同时rebuild global index 并分析global index才对

(三) 空间释放问题

其实空间等都已经释放了，但数据字典没有被更新，

例如你查dba\_segments视图，发现这个分区的bytes其实还为原来的大小

我们可执行alter table \*\*\*\* allocate extent即可更新数据字典为正常状态

例如针对范围分区如下操作：

```
alter table *** modify partition PART_*** allocate extent;
```