

TRUNCATE, DROP ve RESIZE ÜZERİNE

1.Giriş

Tablo içinde kayıt silsek bile, bu işlem tablonun tuttuğu alanı değiştirmez. 1 milyon kayıt varken, hepsini silseniz bile 1 milyon kaydın tutulduğu alan hâlâ bu tabloya ayrılmıştır. Yeni kayıtlar girildiğinde, tablo eski kayıtlara ait alanı kullanabilir. Ama bir başka tablo, bu alandan hak alamaz.

Bazı durumlarda da tabloya yazma silme işlemi o kadar sık yapılmıştır ki; tablo fragmente olur ve gereksiz yere alan işgal eder. Mesela 100 bin satır girdiniz. Baştaki 50 bin satır silindi. Artık elinizde 100 bin kayıtlık alan kaplayan, 50 bin kayıtlık bir tablo vardır.

Bu gibi durumlara karşı çeşitli yöntemler uygulanır. Genel olarak tablonun bir kopyası alınır ve eski tablo üzerinde truncate ya da drop işlemi gerçekleştirilir. Bu yazıda, truncate ve drop işlemlerinin tablo boyutlarına nasıl etki ettiğini görüp, datafile'ları resize etme konusuna da değineceğiz.

2. Örnek Çalışma

Aşağıdaki şekilde bir tablespace yaratıyoruz:

```
CREATE TABLESPACE DENEME_CCEBI DATAFILE
  '/datac5/oracle1/oradata/oradb1/deneme_ccebi.dbf' SIZE 1M
AUTOEXTEND ON NEXT 1M MAXSIZE UNLIMITED
LOGGING
ONLINE
PERMANENT
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
BLOCKSIZE 8K
SEGMENT SPACE MANAGEMENT AUTO
FLASHBACK ON;
```

Boyutunu ölçüyoruz, **1048576** byte çıkıyor.

```
SQL> SELECT SUM(BYTES) FROM V$DATAFILE_HEADER WHERE
TABLESPACE_NAME= 'DENEME_CCEBI' ;
```

Şimdi **deneme_ccebi** tablespace'i altında bir tablo yaratıyoruz:

```
CREATE TABLE D_CCEBI.CAGATAY
( ID NUMBER(10), VERI CHAR(1000) )
TABLESPACE DENEME_CCEBI ;
```

50 bin kayıt giriyoruz:

```
BEGIN
  FOR i IN 1..50000
  LOOP
    INSERT INTO D_CCEBI.CAGATAY VALUES(i, 'BUGUN HAVA O KADAR GUZELKI... ');
  END LOOP;
COMMIT;
END;
```

DENEME_CCEBI tablespace'ini tekrar ölçüyoruz, 60817408 byte (58 Mbyte) çıkıyor.

Tabloyu truncate edelim:

```
SQL> truncate table d_ccebi.cagatay
```

DENEME_CCEBI tablespace'ini tekrar ölçüyoruz ve yine 60817408 byte (58 Mbyte) çıkıyor.

Yukarıdaki şekilde tekrar data insert edelim. Ardından tablespace boyutunu ölçelim. Tablespace boyutu değişmiyor. Yine 58 Mbyte. Bunun nedenini şöyle açıklayabiliriz; veritabanında truncate drop işlemleri yaptığınızda kullandığı alan, bırakılır. Ama datafile'ların boyutu değişmez. Şimdi aynı işlemi tekrar yapalım ama bu sefer drop ile deneyelim:

```
SQL> drop table d_ccebi.cagatay;
```

Son olarak **DENEME_CCEBI** tablespace'ini tekrar ölçüyoruz ve yine 60817408 byte (58 Mbyte) çıkıyor. Truncate ile drop arasında bir fark olmadı. Her ikisi de, tabloyu boşaltıyor; tuttuğu alanı boşaltıyor ama datafile'i boşaltmıyor. Drop esnasında PURGE yapılırsa da değişmiyor. Ancak boşalan alan başka table'lara tekrar tahsis edilebiliyor.

Bazı durumlarda, büyük bir tablo olabilir ve bunu drop ettikten sonra, işletim sistemi üzerindeki alanı geri kazanmak isteyebiliriz. Peki datafile'ı nasıl boşaltacağız?

```
select fs.file_id, fs.block_id, fs.blocks, fs.bytes as "Bytes Free",
df.bytes "Bytes Allocated",
df.bytes-fs.BYTES "Can Be Shrunked To (Bytes)"
from dba_free_space fs, dba_data_files df
where fs.tablespace_name = 'DENEME_CCEBI'
and fs.file_id = df.FILE_ID
order by block_id desc;
```

Buna göre datafile'i 1 MB'a düşürebiliriz.

```
SQL> alter database datafile  
      '/datac5/oracle1/oradata/oradb1/deneme_ccebi.dbf' resize 1M;
```

Ancak çok önemli bir ayrıntı var. Eğer datafile üzerinde başka tablolar varsa; datafile fragmente olacaktır. Datafile fragmente olduğunda, resize işlemini bu şekilde yapmak pek mümkün olmaz. 60 MB toplamda boş alan olsa bile, 60 MB'ı bir anda boşa çıkartmak mümkün değildir. Çünkü parça parça boşlukları bir bütün olarak resize edemiyoruz.

Bu sebeple, çok yazma silme işleminin olduğu tabloları farklı bir tablespace'e taşımak; truncate etmek; zaman zaman da datafile'i resize etmek gerekir. Bunun dışında, kayıp alanın tekrar kullanımı için table shrink benzeri komutlar da bulunuyor. Ancak şimdilik onu bir başka yazıya bırakalım...