Practice

Using Deferred Segment Creation

Practice Target

In this practice, you will examine the deferred segment creation behavior in Oracle database.

Practice Assumptions

- You have the `srv1` and its **CDB** database up and running.

Ahmed Baraka
Oracle Database Administrator

Using Deferred Segment Creation

In the following steps, you will create a series of tables, first when the deferred segment creation is disabled and second when it is enabled. Then, you will materialize the deferred segments.

1. Start a Putty session to `srv1` as `oracle`.

2. Run the following code to create a script file named as `create_tables.sql`

The script creates 10 tables with names of the format `TABLE<n>`, where n is an integer from 1 to 10.

```
cat > create_tables.sql <<EOL
-- this script creates tables TABLE1 to TABLE10
--  if a table exists, it drops it
DECLARE
 T VARCHAR2(10);
 N INTEGER;
BEGIN
 FOR I IN 1..10 LOOP
  -- table name
  T:= 'TABLE' || TO_CHAR(I) ;
  -- if the table exists, drop it
  SELECT COUNT(*) INTO N FROM USER_TABLES WHERE TABLE_NAME=T;
  IF N >0 THEN
   EXECUTE IMMEDIATE 'DROP TABLE ' || T || ' PURGE';
  END IF;
  EXECUTE IMMEDIATE 'CREATE TABLE ' || T || ' ( PERSON_ID NUMBER, PERSON_NAME
VARCHAR2(20))';
 END LOOP;
END;
/

EOL
```

3. Invoke SQL*Plus and connect to `pdb1` as `SYSTEM`.

```
sqlplus SYSTEM/ABcd##1234@//srv1/pdb1.localdomain
```

4. Check the status of the deferred segment creation.

By default, deferred segment creation is enabled.

```
SHOW PARAMETER DEFERRED_SEGMENT_CREATION
```

5. Login to `PDB1` as `HR` and disable the deferred segment creation at the session level.

```
conn hr/ABcd##1234@//srv1/pdb1.localdomain
ALTER SESSION SET DEFERRED_SEGMENT_CREATION=FALSE;
```

6. Create the testing empty tables.

```
@ create_tables.sql
```

7. Verify that the tables are created.

```
SELECT COUNT(*) FROM USER_TABLES WHERE TABLE_NAME LIKE 'TABLE%';
```

8. Check if the tables consume any disk space from the tablespace.

The tables consume 640K from the tablespace. This means 64K is consumed by each table.

```
SELECT SUM(BYTES/1024) KB FROM USER_SEGMENTS WHERE SEGMENT_NAME LIKE 'TABLE%';
```

9. Check how many extents are created for one table.

As expected, each table contains one empty extent. The extent size is 64k or 8 blocks. This size is taken from the `INITIAL_EXTENT` of the tablespace that accommodates the segment.

```
SELECT EXTENT_ID, BYTES/1024 "SIZE(KB)", BLOCKS FROM USER_EXTENTS WHERE
SEGMENT_NAME='TABLE1';
```

Ahmed Baraka
Database Administrator

Now let's check the behavior of the database when the deferred segment creation is enabled.

10. Enable the deferred segment creation at the session level then re-create the testing empty tables.

```
ALTER SESSION SET DEFERRED_SEGMENT_CREATION=TRUE;
@ create_tables.sql
```

11. Verify that the tables are created and that their segments are not created.

```
SELECT COUNT(*) FROM USER_TABLES WHERE TABLE_NAME LIKE 'TABLE%';
SELECT SUM(BYTES/1024) KB FROM USER_SEGMENTS WHERE SEGMENT_NAME LIKE 'TABLE%';
```

12. Try inserting a dummy row in `TABLE1`.

```
INSERT INTO TABLE1 VALUES ( 1, 'PERSON1');
COMMIT;
```

13. Verify that a segment is created for `TABLE1`.

```
SELECT SUM(BYTES/1024) KB FROM USER_SEGMENTS WHERE SEGMENT_NAME = 'TABLE1';
```

14. Materialize the rest of the deferred segments.

Because `HR` does not have execute privilege on `DBMS_SPACE_ADMIN`, you will execute it as `SYS`.

```
conn sys/ABcd##1234@//srv1/pdb1.localdomain as sysdba
EXEC DBMS_SPACE_ADMIN.MATERIALIZE_DEFERRED_SEGMENTS ( SCHEMA_NAME=>'HR')
```

15. Verify that the tables are materialized.

```
conn HR/ABcd##1234@//srv1/pdb1.localdomain
SELECT COUNT(*) CNT, SUM(BYTES/1024) KB FROM USER_SEGMENTS WHERE SEGMENT_NAME
LIKE 'TABLE%';
```

16. As cleanup, drop the tables.

```
DECLARE
 T VARCHAR2(10);
 N INTEGER;
BEGIN
 FOR I IN 1..10 LOOP
  -- table name
  T:= 'TABLE' || TO_CHAR(I) ;
  -- if the table exists, drop it
  SELECT COUNT(*) INTO N FROM USER_TABLES WHERE TABLE_NAME=T;
  IF N >0 THEN
   EXECUTE IMMEDIATE 'DROP TABLE ' || T || ' PURGE';
  END IF;
END LOOP;
END;
/
```

17. Delete the script file.

```
host rm create_tables.sql
```

Summary

- Deferred segment creation saves disk space and installation time when creating large number of empty tables.

- Materializing deferred segments creates extents for the segments without inserting rows into the segments.

Ahmed Baraka
Oracle Database Administrator