Create table DEMO as select \* from dba\_objects;

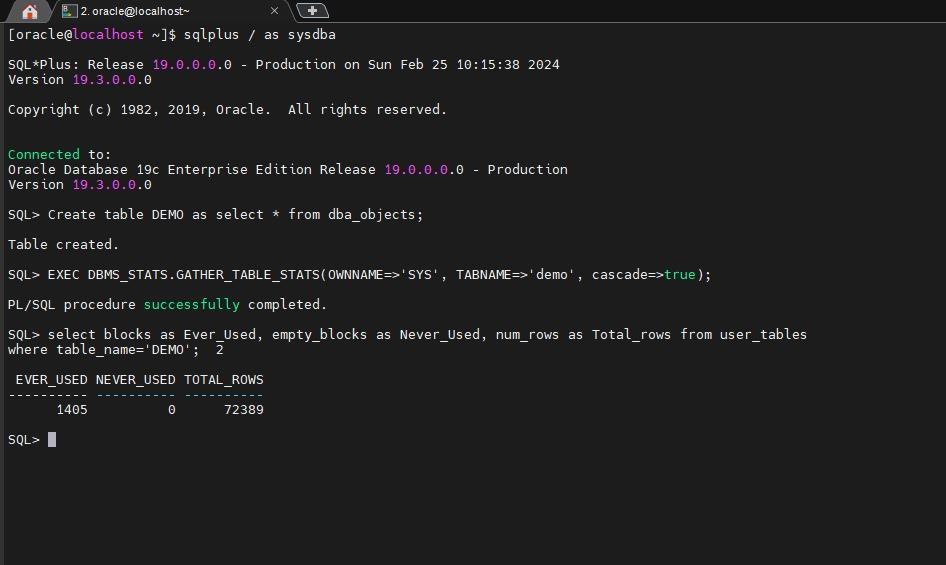
(Primeiro Efetuei a criação da Table para Efetuar os testes)

EXEC DBMS\_STATS.GATHER\_TABLE\_STATS(OWNNAME=>'SYS', TABNAME=>'demo', cascade=>true);

(Efetuei a coleta de estastisicas)

select blocks as Ever\_Used, empty\_blocks as Never\_Used, num\_rows as Total\_rows from user\_tables where table\_name='DEMO';

(Esse comando é para visualizar a quantidade de blocos que esta sendo utilizados e os que está vazio)



Delete from DEMO where OBJECT\_TYPE='SEQUENCE';

Delete from DEMO where OBJECT\_TYPE='TABLE';

Delete from DEMO where OBJECT\_TYPE='LOB';

Delete from DEMO where OBJECT\_TYPE='INDEX';

Delete from DEMO where OBJECT\_TYPE='TYPE';

Commit;

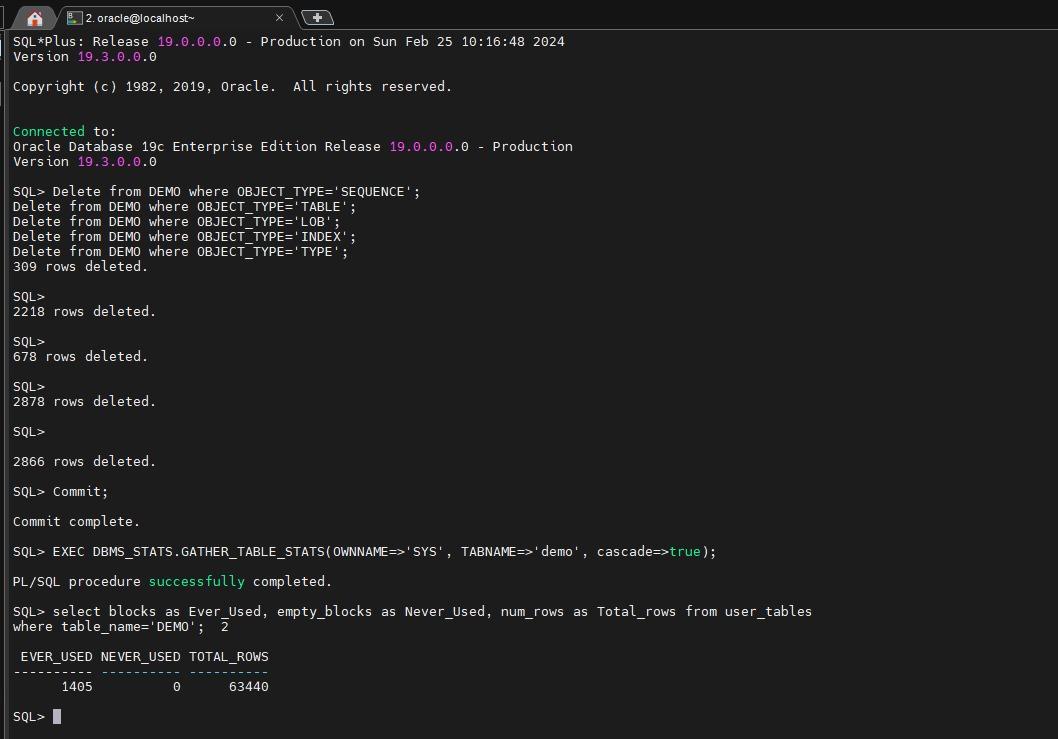
(Efetuei o DELETE de alguns arquivos para estar efetuando os testes)

EXEC DBMS\_STATS.GATHER\_TABLE\_STATS(OWNNAME=>'SYS', TABNAME=>'demo', cascade=>true);

(Coletei estatísticas novamente para atualiza o dicionário de dados após o DELETE)

elect blocks as Ever\_Used, empty\_blocks as Never\_Used, num\_rows as Total\_rows from user\_tables where table\_name='DEMO';

(Rodei o script para valida a quantidade de linhas da tabela novamente )

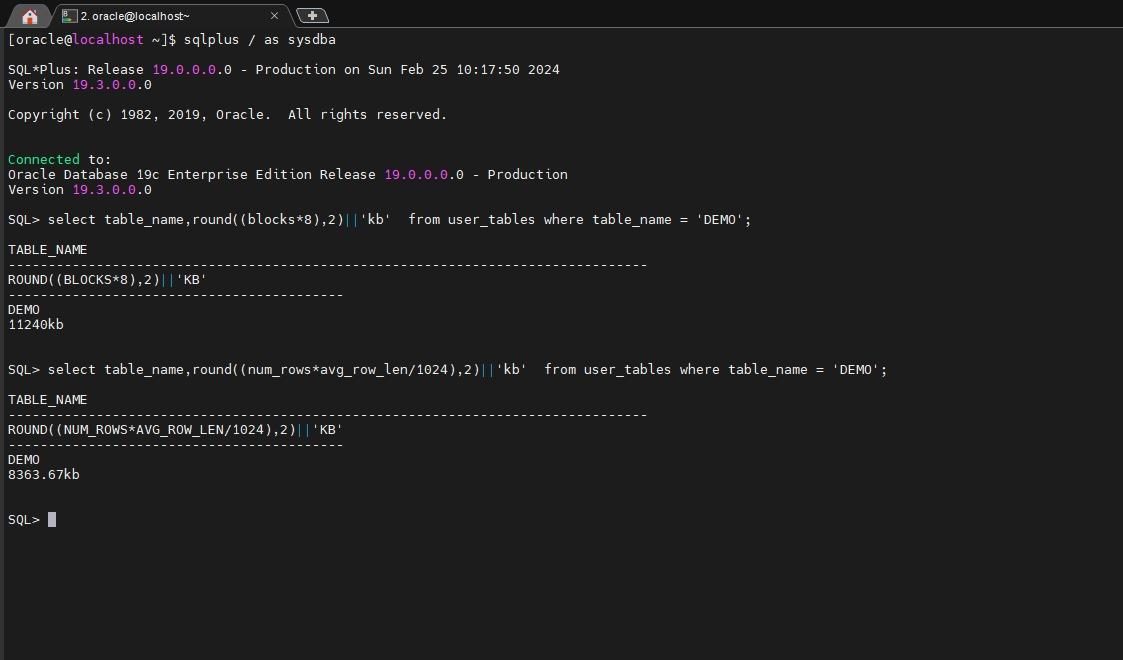


select table\_name,round((blocks\*8),2)||'kb' from user\_tables where table\_name = 'DEMO';

(Esse script mostra a quantidade de Dados Reais da Tabela)

select table\_name,round((num\_rows\*avg\_row\_len/1024),2)||'kb' from user\_tables where table\_name = 'DEMO';

(Esse script mostra a quantidade de Dados Fragmentados da Tabela)

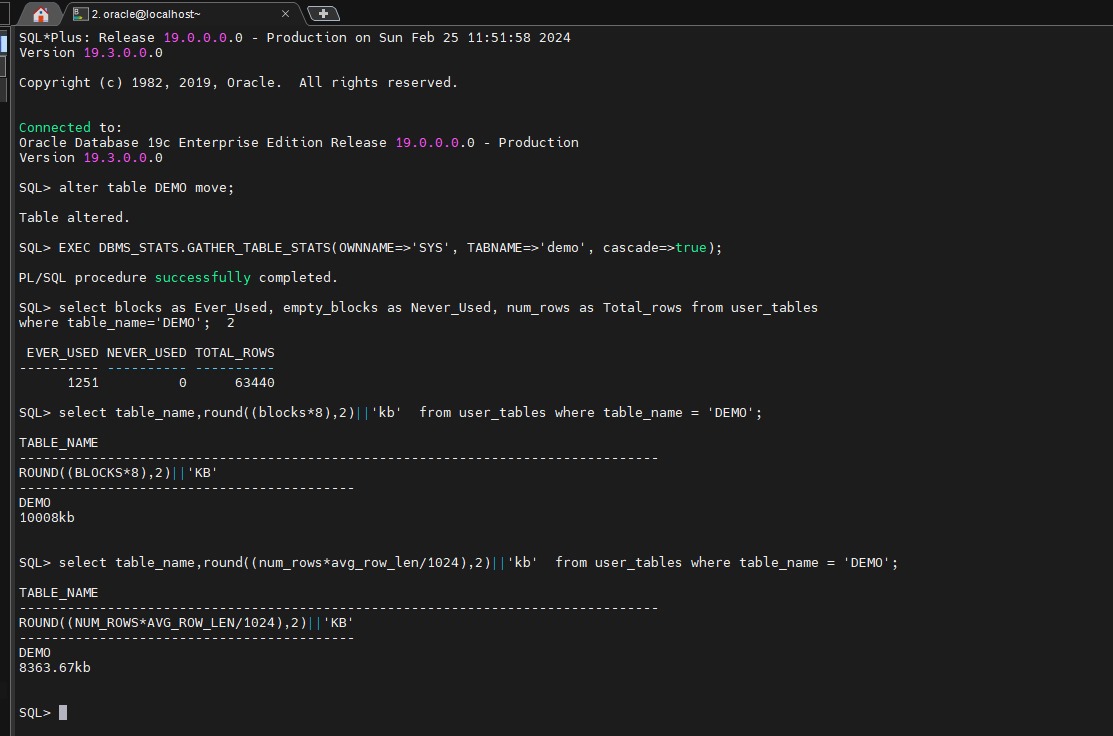


ALTER TABLE DEMO MOVE;

(Efetuei o MOVE para começar a desfragmentação da Tabela)

EXEC DBMS\_STATS.GATHER\_TABLE\_STATS(OWNNAME=>'SYS', TABNAME=>'demo', cascade=>true);

(Coletei a estatísticas novamente para atualizar o dicionário de dados após corrigir a fragmentação)



Após o MOVE tivemos uma diferença nos blocos Usados de 1450 para 1251 nesse caso o REORG surtiu efeito

Nota-se que a quantidade de Dados reais também diminuiu um pouco considerado

Uma observação importante é que não houve diferença nos blocos fragmentados por que excluir poucos dados vou estar efetuando um delete mais preciso para que possa trazer uma diferença significativa!!

select status,index\_name from user\_indexes Where table\_name = 'DEMO';

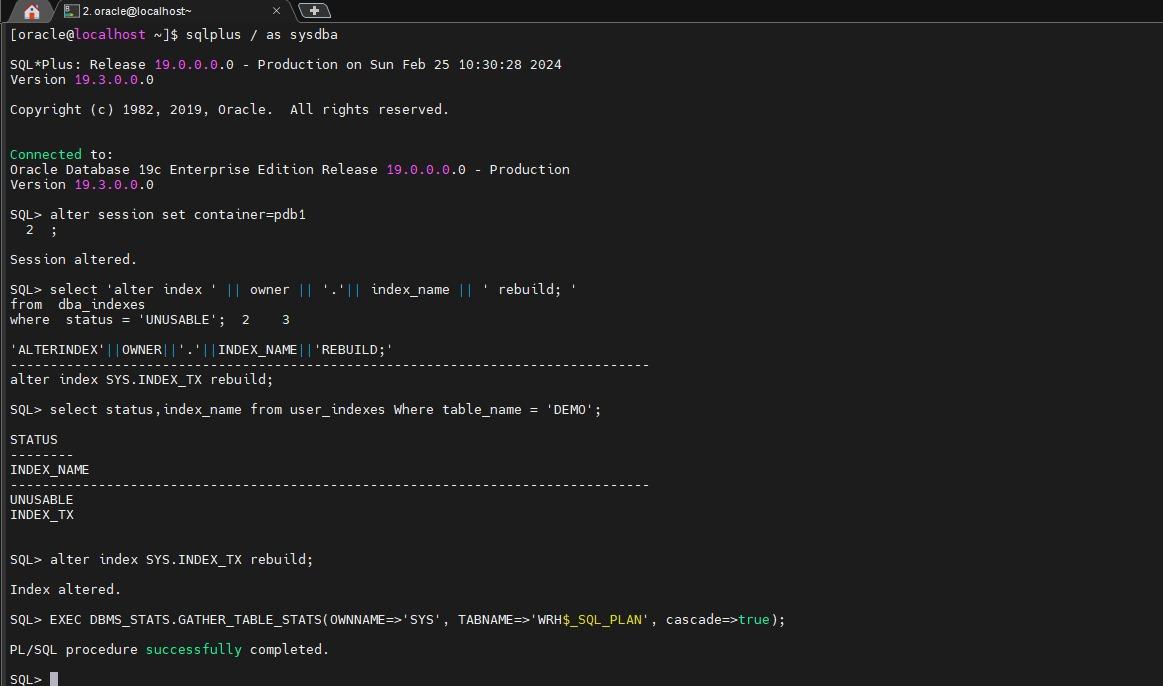
(Esse comando é para avaliarmos os INDEX da tabela que fizemos o MOVE)

select 'alter index ' || owner || '.'|| index\_name || ' rebuild; ' from dba\_indexes where status = 'UNUSABLE';

(Após efetuarmos o MOVE alguns index da Tabela Fica indisponível nesse caso é preciso efetuar o REBUILD do index para que ele fique disponível novamente)

ALTER SYS.INDEX.TX REBUILD

(Após efetuarmos o MOVE alguns index da Tabela Fica indisponível nesse caso é preciso efetuar o REBUILD do index para que ele fique disponível novamente)



delete from demo where object\_id BETWEEN 10 AND 40000;

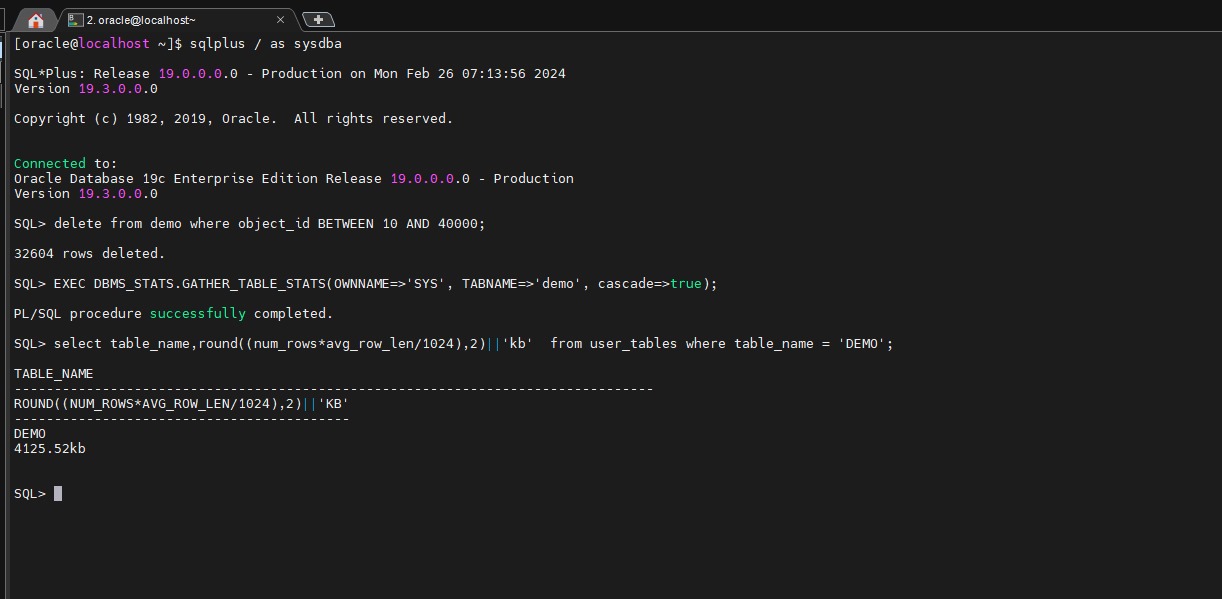
( Usei o delete com Between para efetuar um delete maior )

EXEC DBMS\_STATS.GATHER\_TABLE\_STATS(OWNNAME=>'SYS', TABNAME=>'demo', cascade=>true);

(Coletei a estatísticas novamente para atualizar o dicionário de dados após corrigir a fragmentação)

select table\_name,round((num\_rows\*avg\_row\_len/1024),2)||'kb' from user\_tables where table\_name = 'DEMO';

(Esse script mostra a quantidade de Dados Fragmentados da Tabela)



Aqui vou estar trazendo a diferença entre não efetuar o MOVE e efetuar o MOVE

Nota-se que sem o MOVE os blocos utilizados continua o mesmo valor mesmo após efetuar o DELETE e coletar estatísticas

Delete from DEMO where OBJECT\_TYPE='SEQUENCE';

Delete from DEMO where OBJECT\_TYPE='TABLE';

Delete from DEMO where OBJECT\_TYPE='LOB';

Delete from DEMO where OBJECT\_TYPE='INDEX';

Delete from DEMO where OBJECT\_TYPE='TYPE';

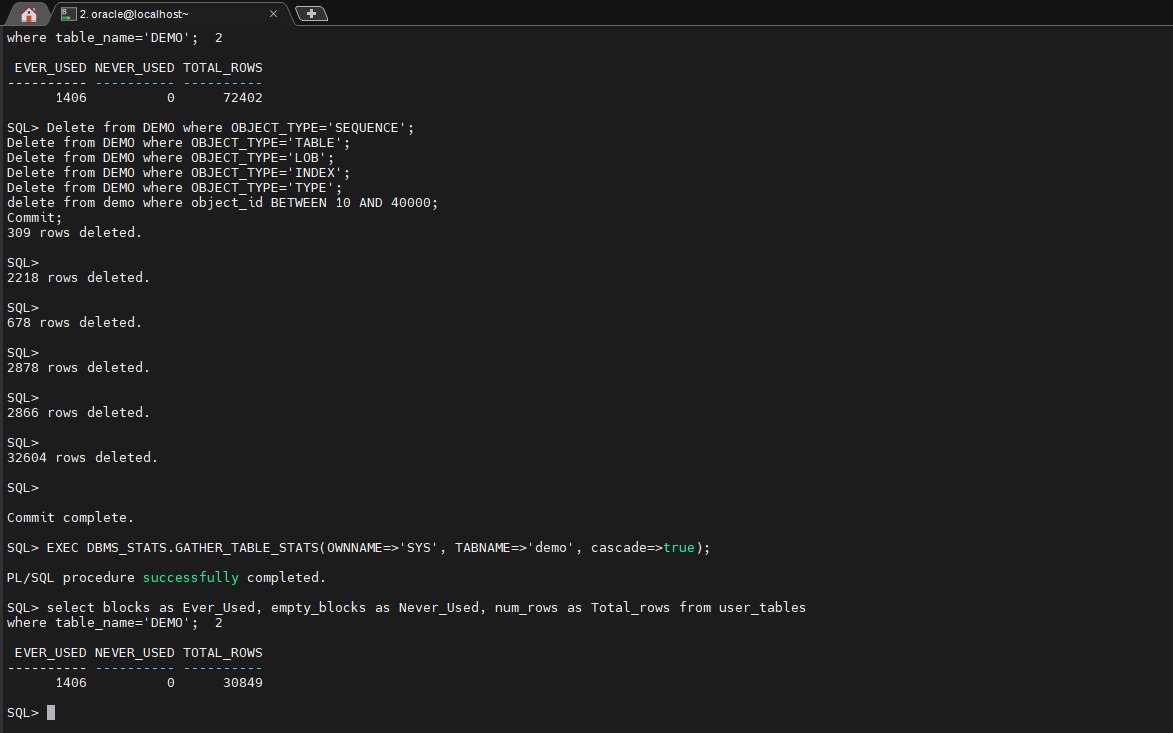
delete from demo where object\_id BETWEEN 10 AND 40000;

Commit;

EXEC DBMS\_STATS.GATHER\_TABLE\_STATS(OWNNAME=>'SYS', TABNAME=>'demo', cascade=>true);

select blocks as Ever\_Used, empty\_blocks as Never\_Used, num\_rows as Total\_rows from user\_tables where table\_name='DEMO';

SEM O MOVE



COM O MOVE

