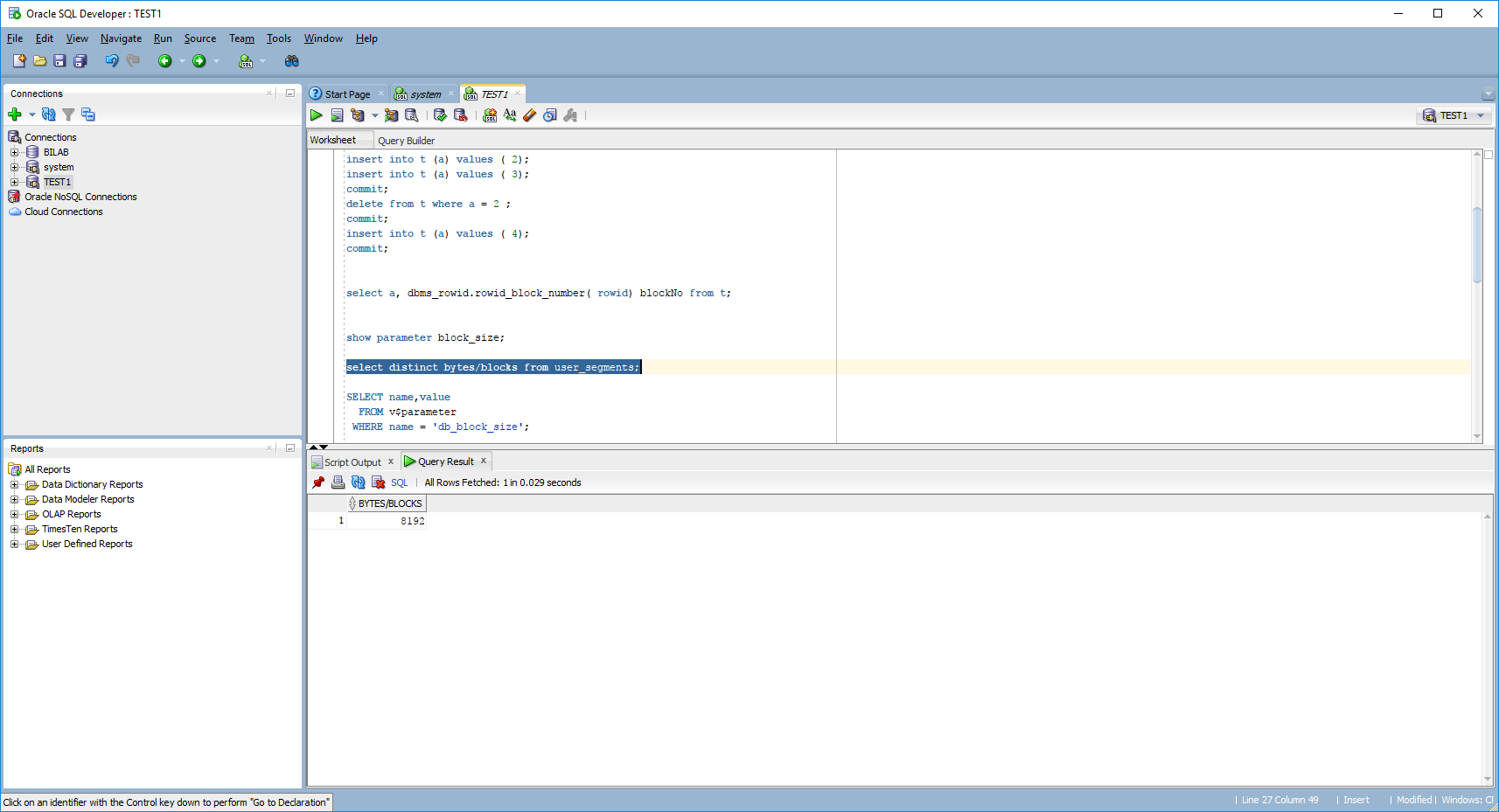
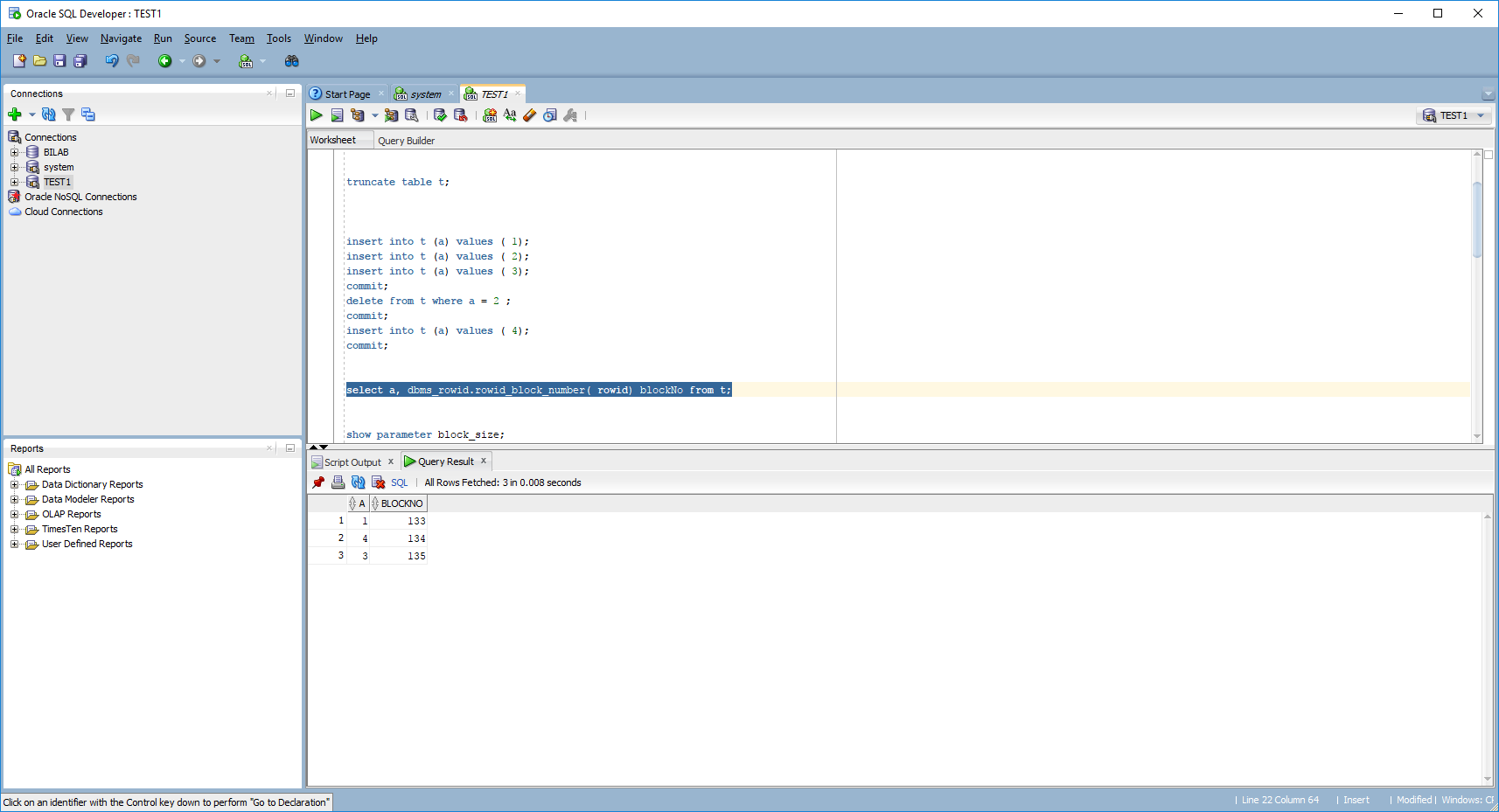
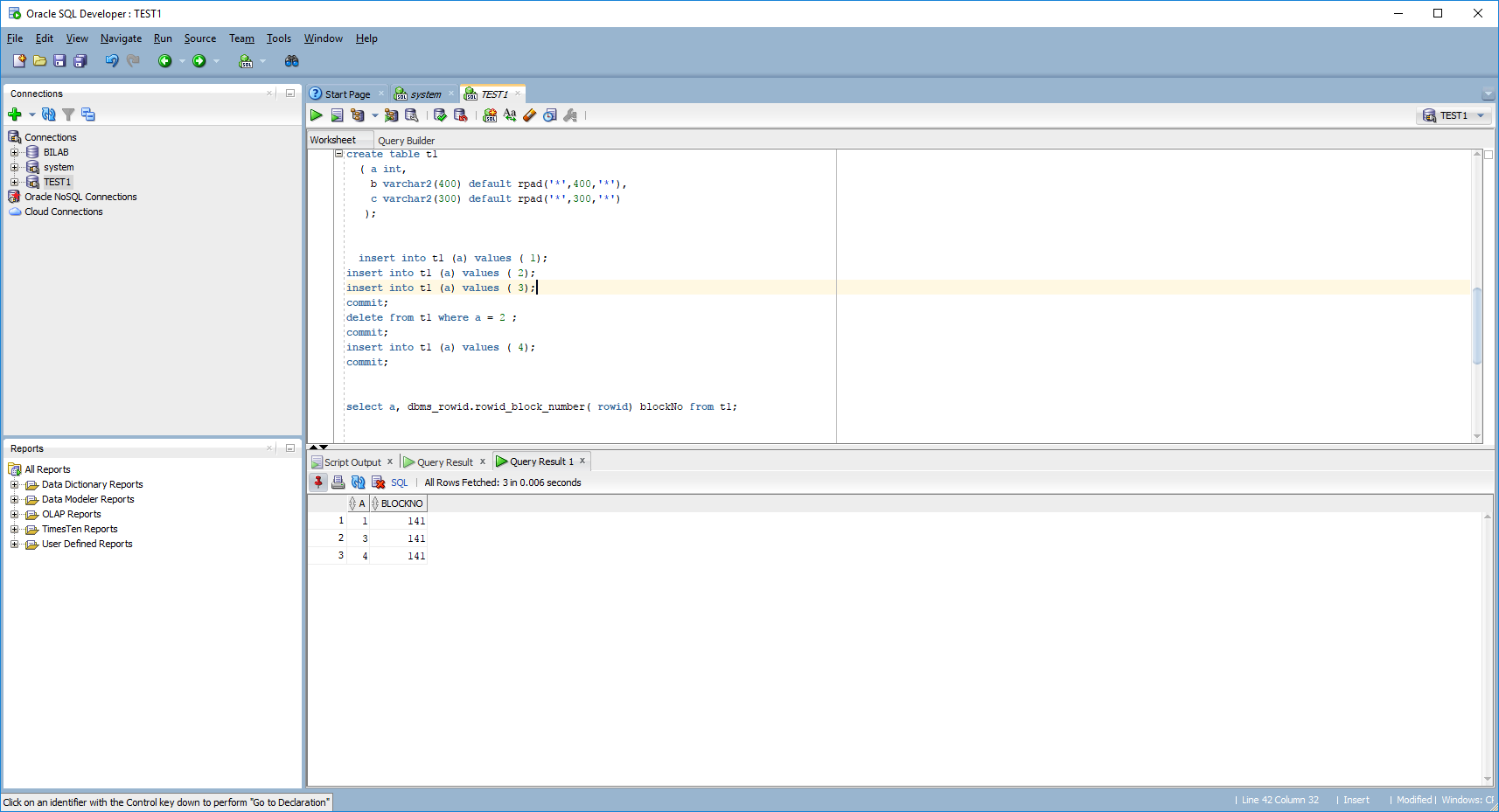
2.2

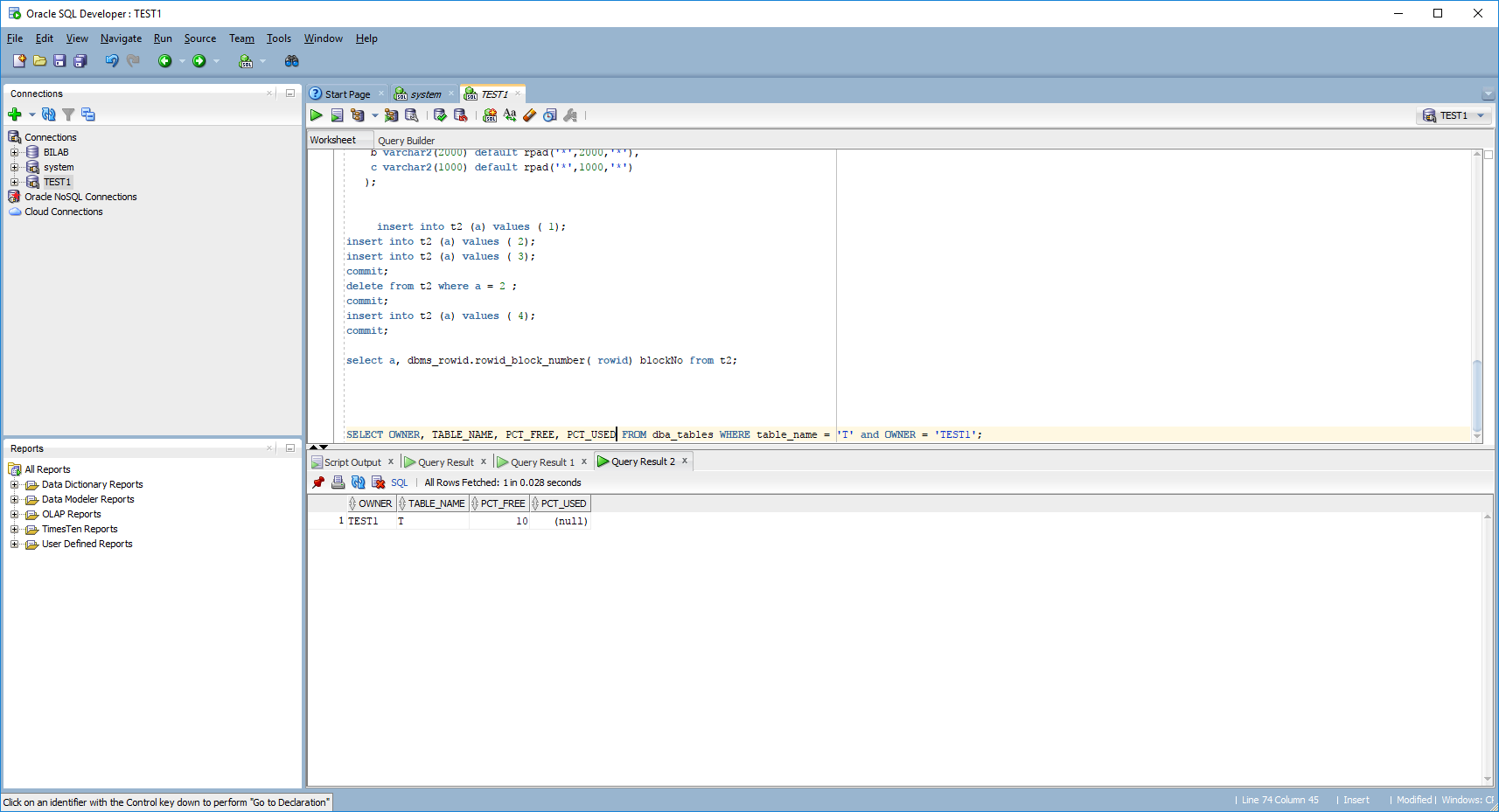


3 строчки находяться в разных блоках, т.к. размер одной строки занимает 4000+3000+2, а размер одного блока 8192.



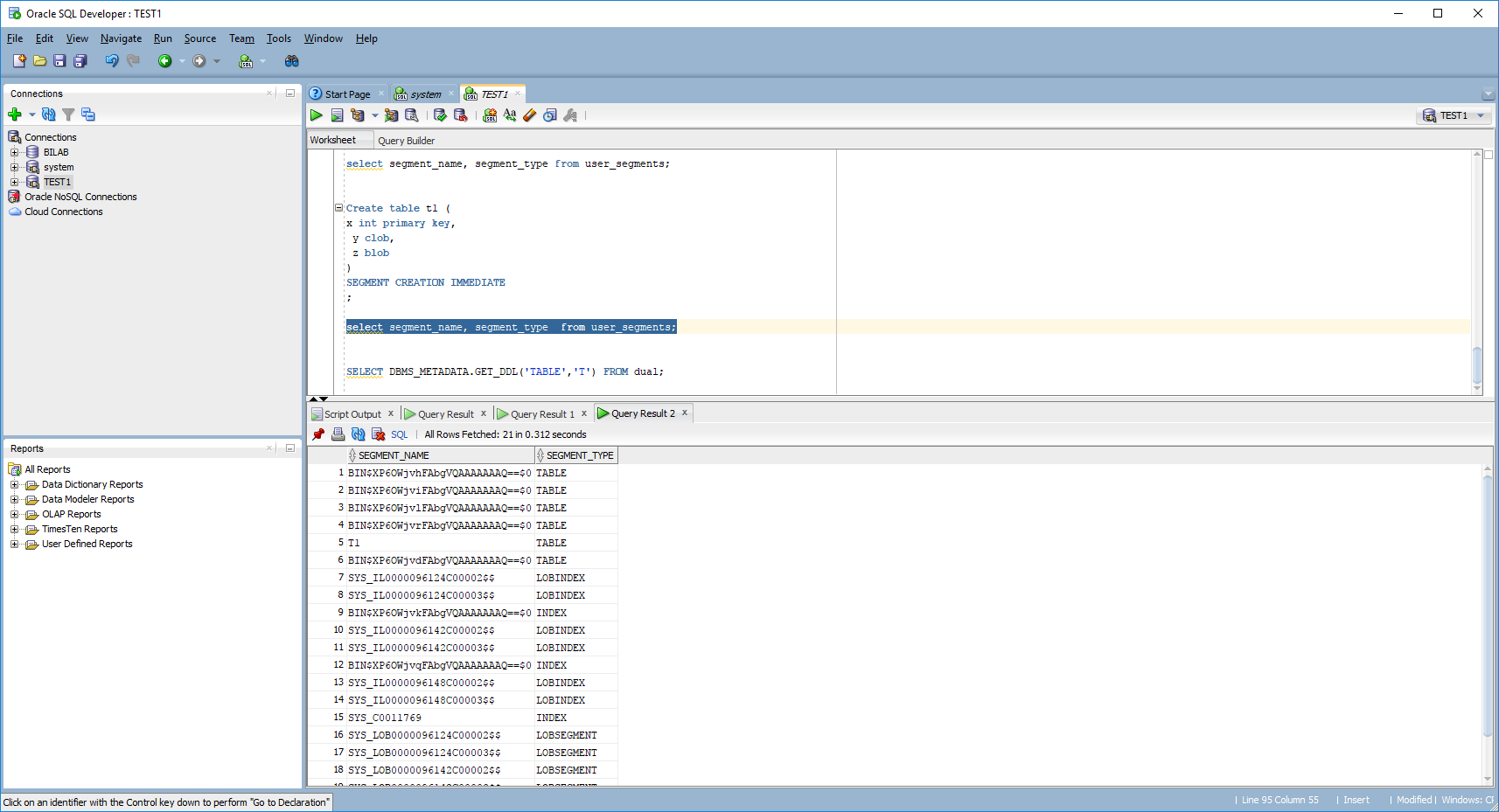
Новая таблица с размером строк 400 и 300. Все три строки помещаются в один блок.





2.4

Для второй таблицы был созданы сегменты, а для первой нет, т.к. она пустая.



Для первой таблицы

CREATE TABLE "TEST1"."T"

( "X" NUMBER(\*,0),

"Y" CLOB,

"Z" BLOB,

PRIMARY KEY ("X")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 NOLOGGING

TABLESPACE "TBS\_PDB\_TEST" ENABLE

) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS NOLOGGING

TABLESPACE "TBS\_PDB\_TEST"

LOB ("Y") STORE AS SECUREFILE (

TABLESPACE "TBS\_PDB\_TEST" ENABLE STORAGE IN ROW CHUNK 8192

NOCACHE NOLOGGING NOCOMPRESS KEEP\_DUPLICATES )

LOB ("Z") STORE AS SECUREFILE (

TABLESPACE "TBS\_PDB\_TEST" ENABLE STORAGE IN ROW CHUNK 8192

NOCACHE NOLOGGING NOCOMPRESS KEEP\_DUPLICATES ) "

Для второй таблицы

CREATE TABLE "TEST1"."T1"

( "X" NUMBER(\*,0),

"Y" CLOB,

"Z" BLOB,

PRIMARY KEY ("X")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 NOLOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TBS\_PDB\_TEST" ENABLE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255

NOCOMPRESS NOLOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "TBS\_PDB\_TEST"

LOB ("Y") STORE AS SECUREFILE (

TABLESPACE "TBS\_PDB\_TEST" ENABLE STORAGE IN ROW CHUNK 8192

NOCACHE NOLOGGING NOCOMPRESS KEEP\_DUPLICATES

STORAGE(INITIAL 106496 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT))

LOB ("Z") STORE AS SECUREFILE (

TABLESPACE "TBS\_PDB\_TEST" ENABLE STORAGE IN ROW CHUNK 8192

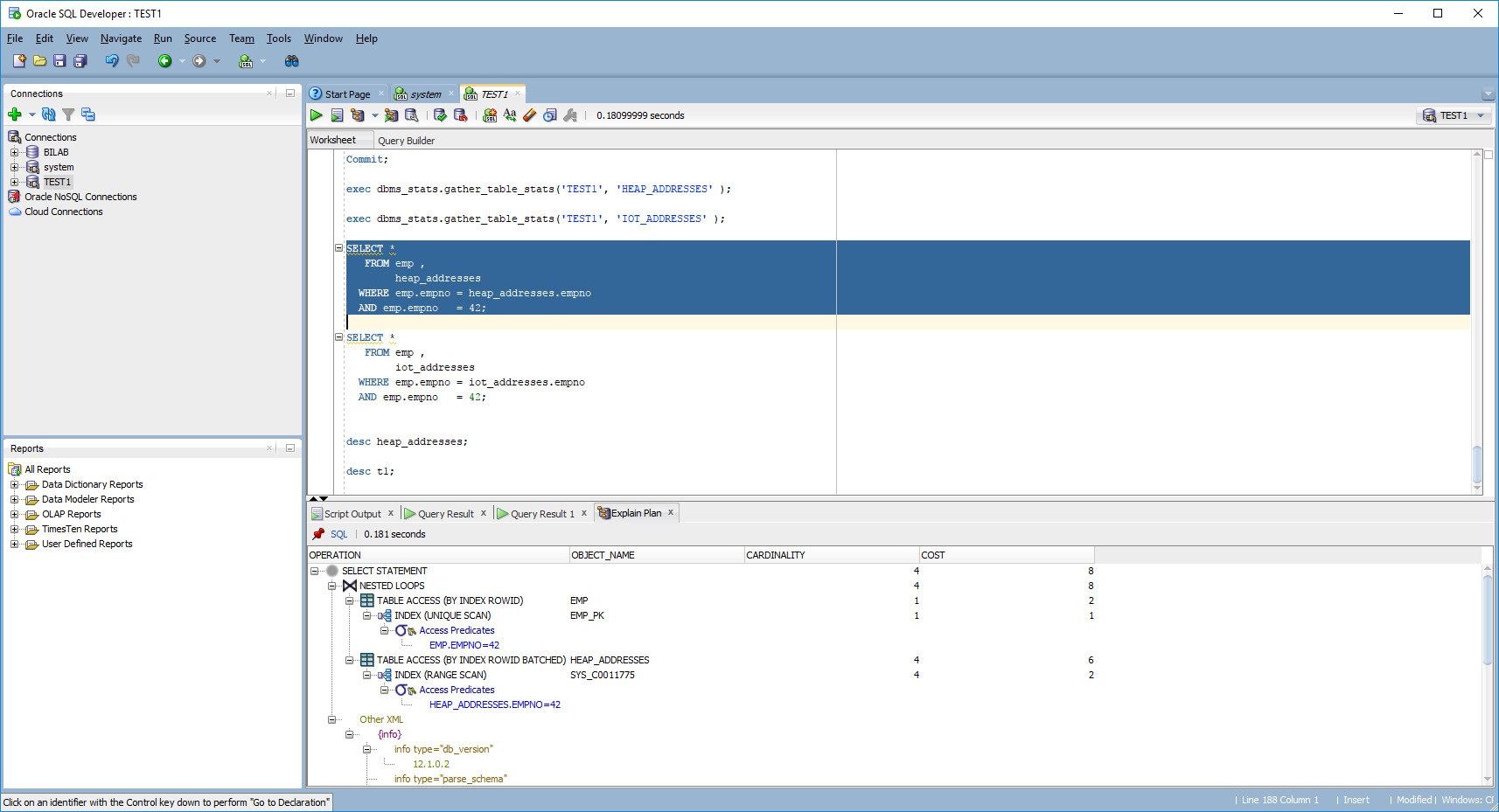
NOCACHE NOLOGGING NOCOMPRESS KEEP\_DUPLICATES

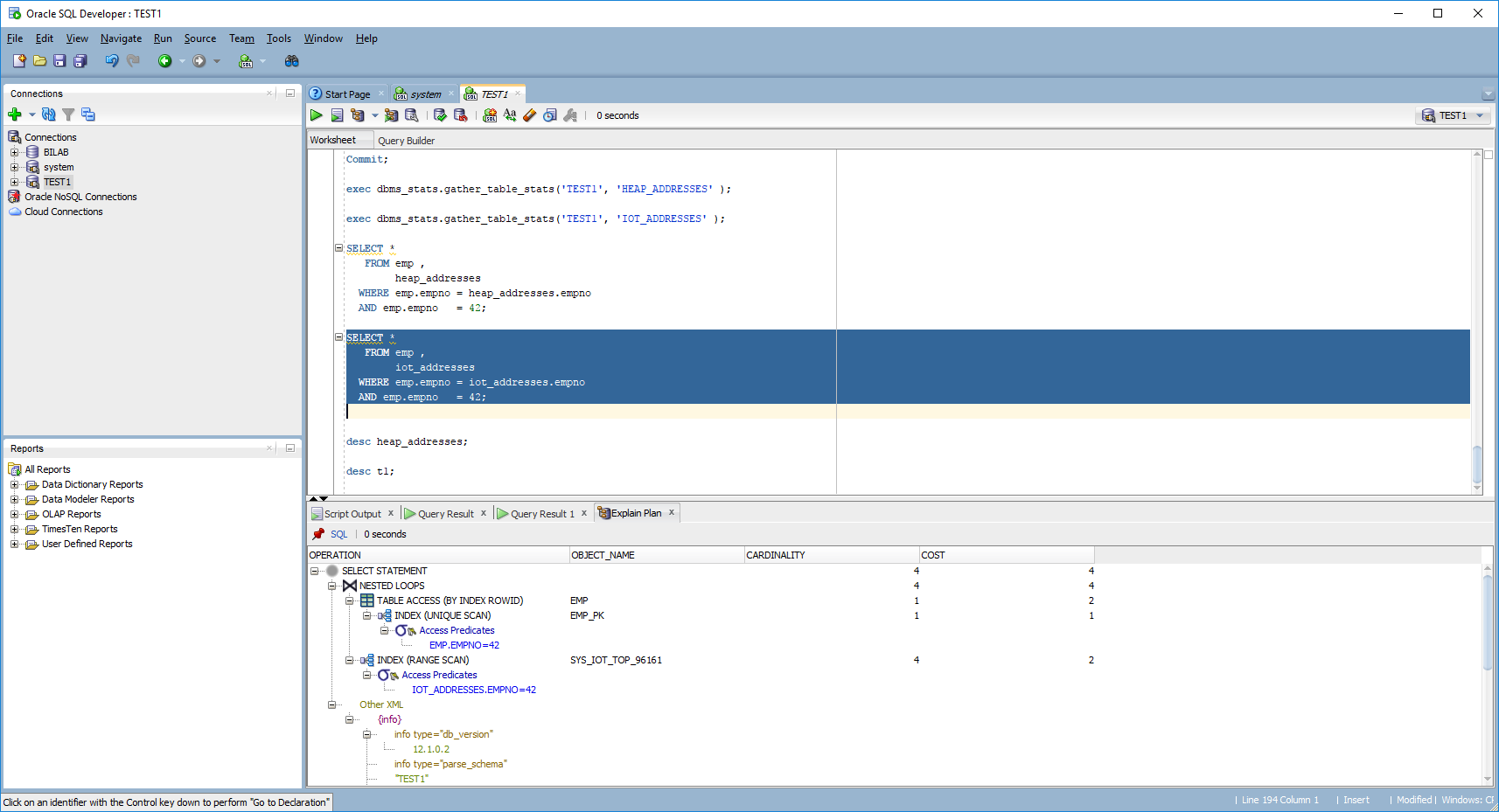
STORAGE(INITIAL 106496 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0

BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)) "

3.1

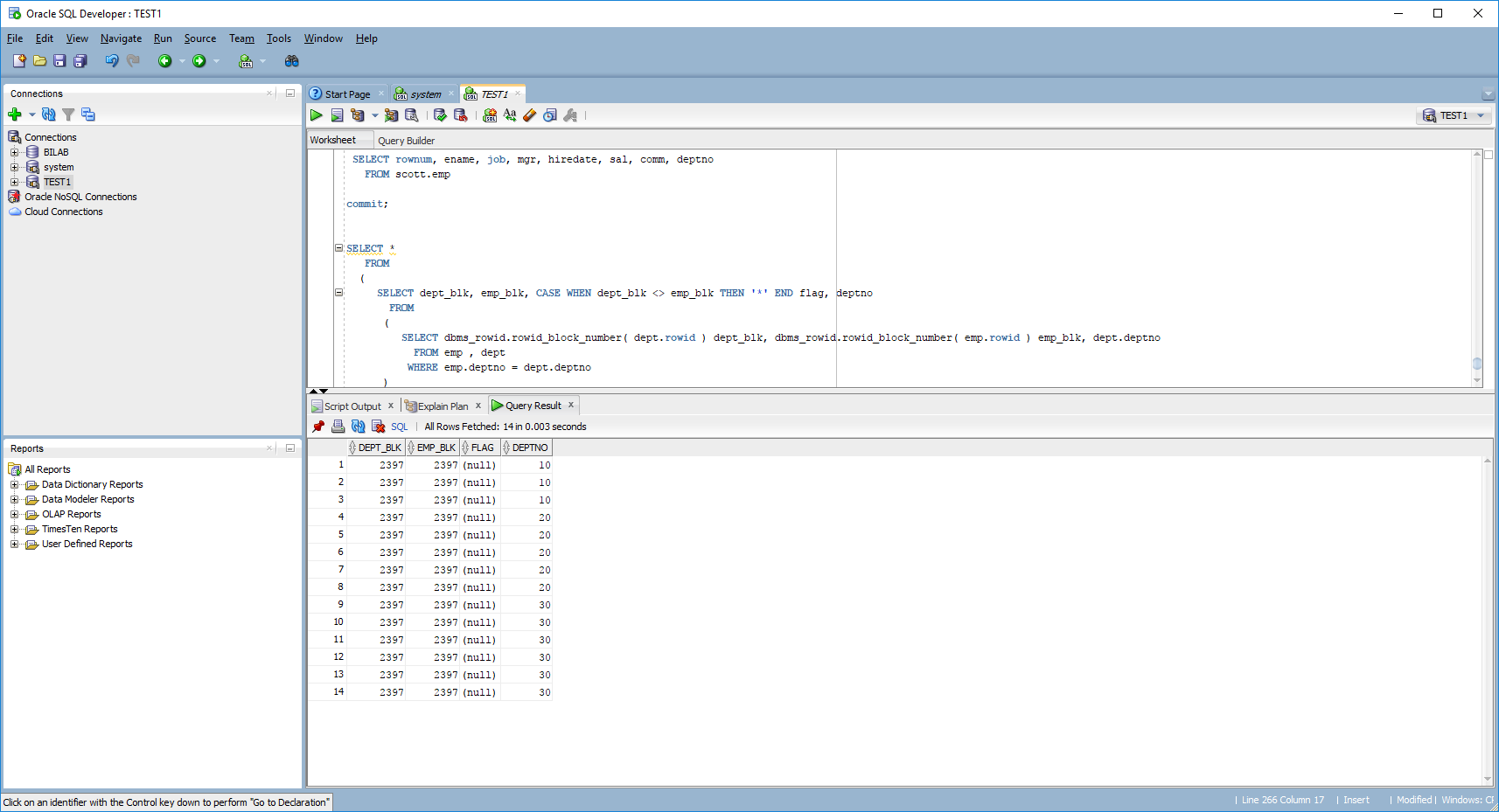




Разница в том, что все данные строки, в втором случае, храниться вместе с индексом и не требуется дополнительное время на их извлечение из таблицы как в первом случае.

4

Связанные данные храняться в одних и тех же блоках.



5

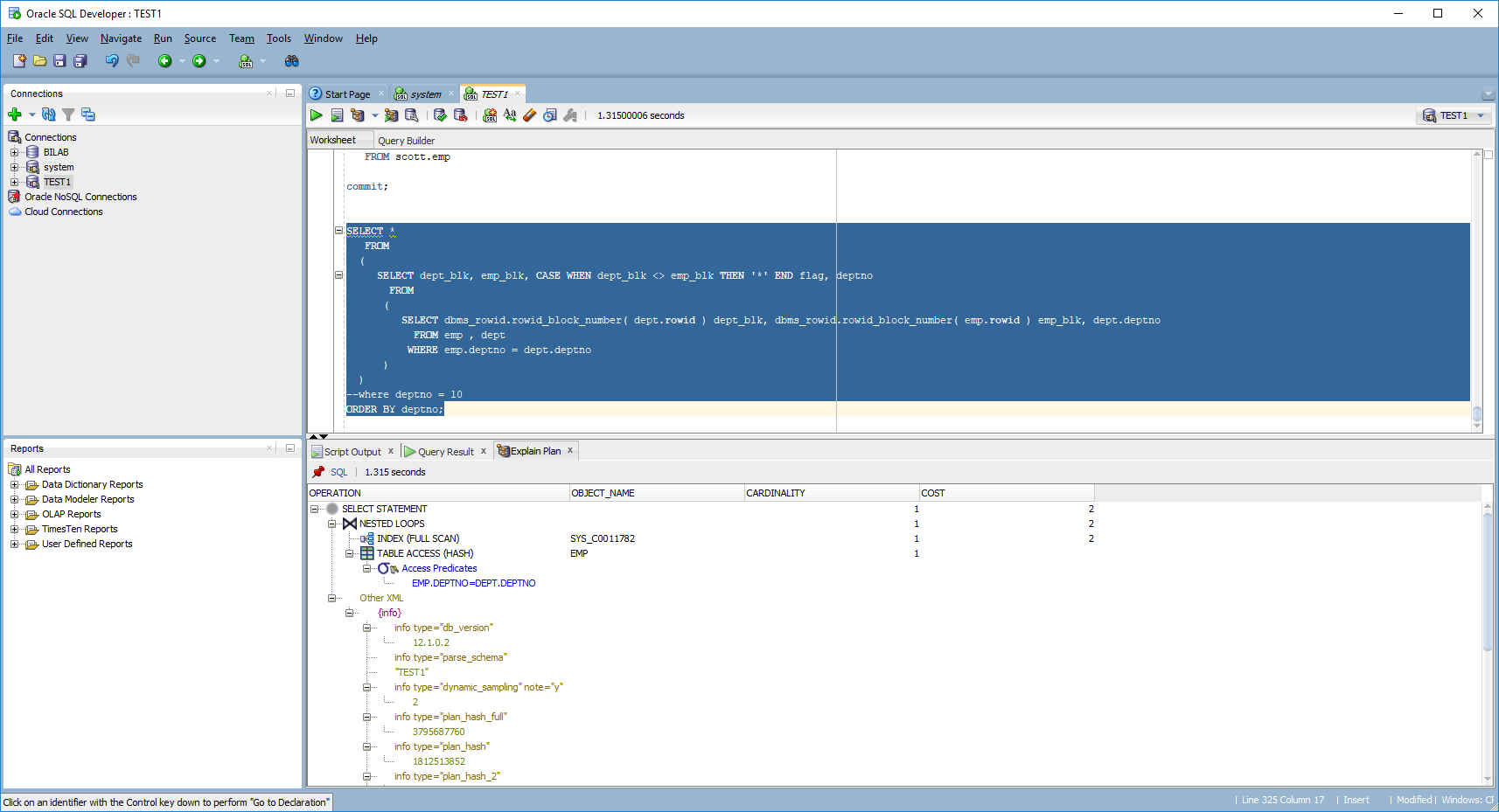
Создание HASH Custer:

CREATE cluster emp\_dept\_cluster2 (deptno NUMBER( 2 ))

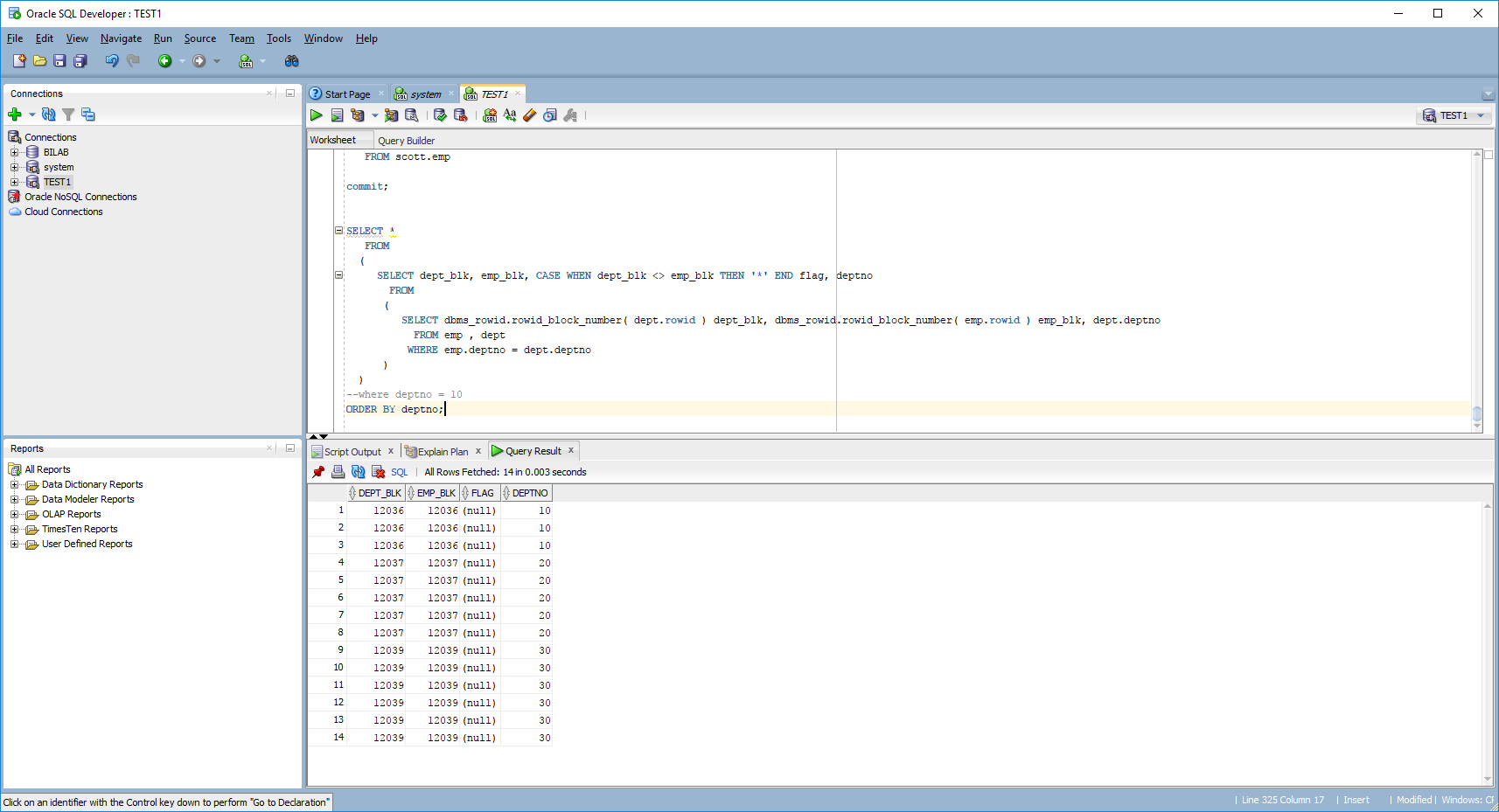
SIZE 1024

STORAGE( INITIAL 100K NEXT 50K )

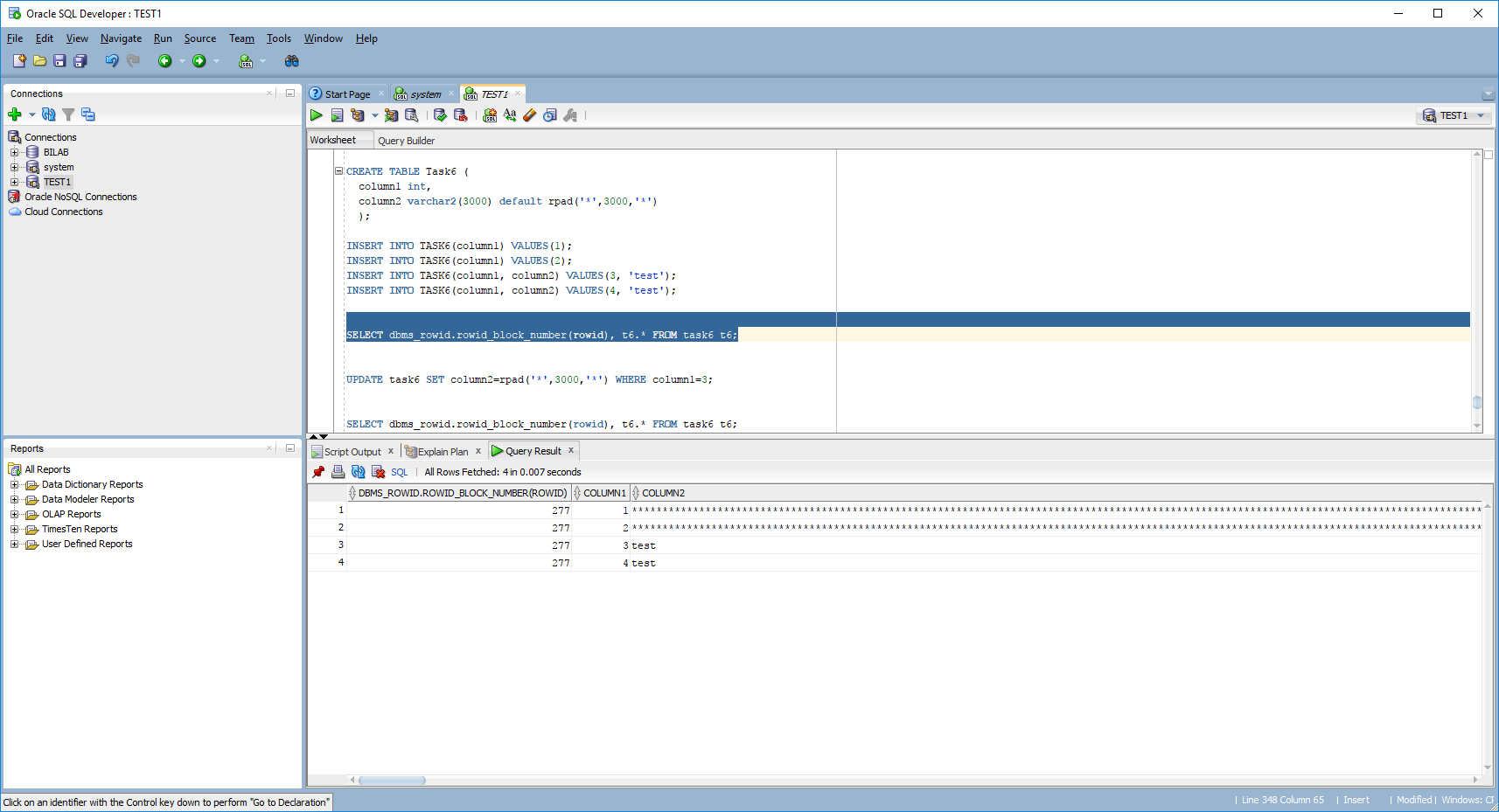
HASHKEYS 10000 HASH IS deptno;



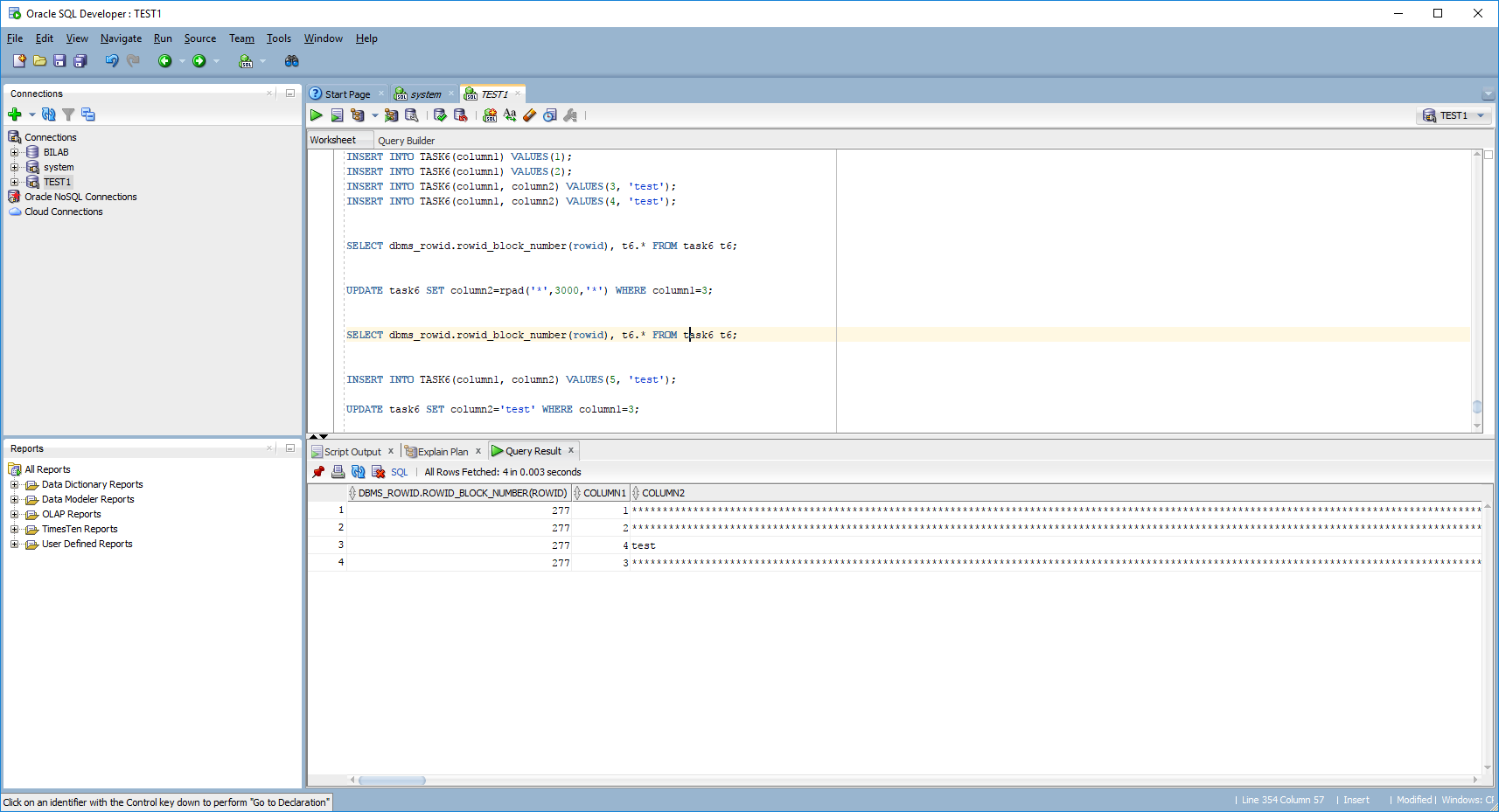
Для строк с одинаковыми deptno один блок. Скорость работы увеличилась по сравненю с Index cluster table.



6.

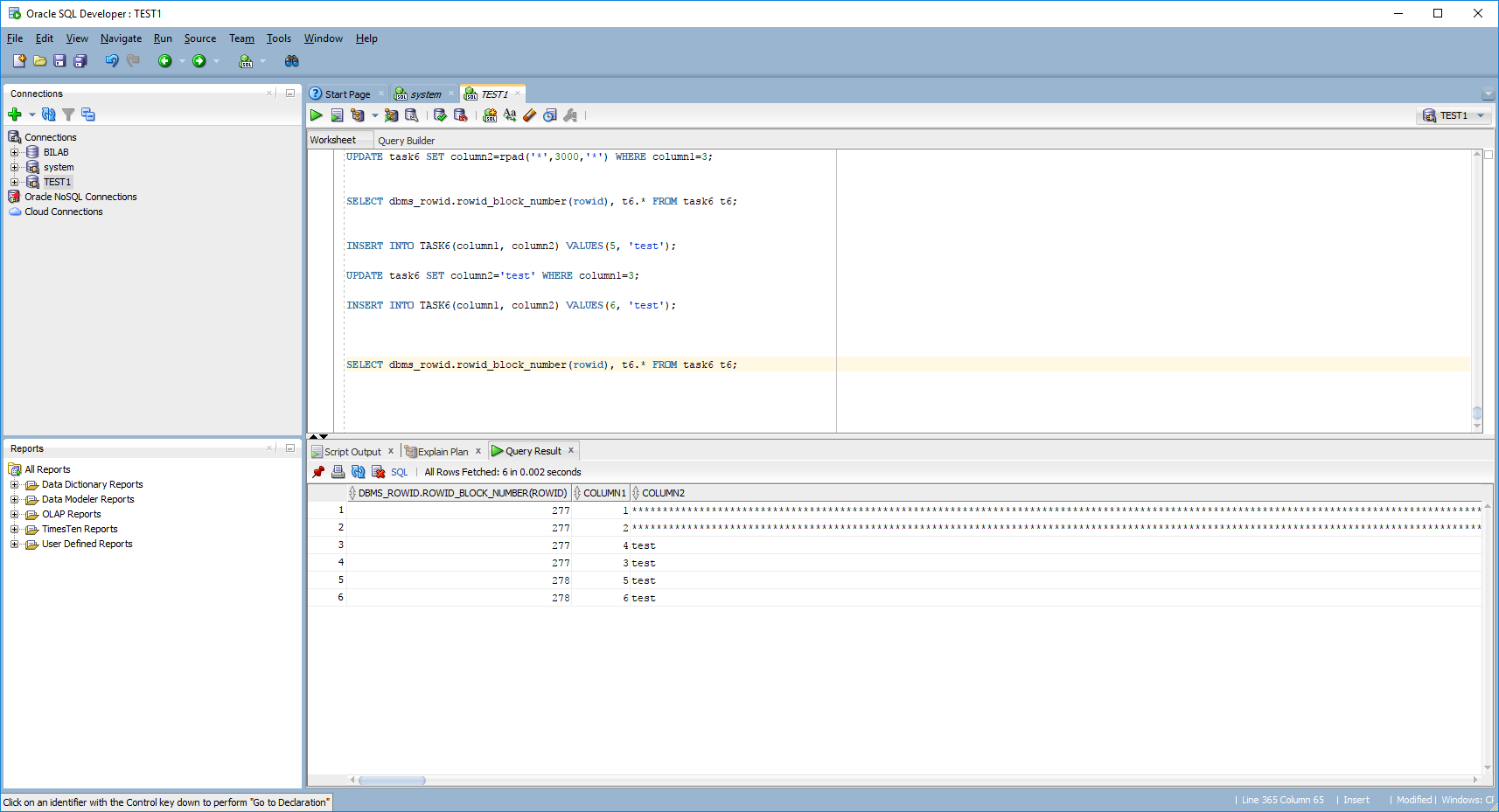


Обновление третей строки для того что бы она превысила объём данных хранящихся в блоке(первые две строки занимаю 6000 из 8192). Номер блока для строки с column1 = 3 остался тем же.



Вставляем строку небольшим размером, но она уже вставляется в другой блок. Далее обновляем строку = 3

до небольшого её размера и вставлям ещё одну новую строку, которая так же вставилась в новый блок.



select distinct bytes/blocks from user\_segments;

CREATE TABLE Task6 (

column1 int,

column2 varchar2(3000) default rpad('\*',3000,'\*')

);

INSERT INTO TASK6(column1) VALUES(1);

INSERT INTO TASK6(column1) VALUES(2);

INSERT INTO TASK6(column1, column2) VALUES(3, 'test');

INSERT INTO TASK6(column1, column2) VALUES(4, 'test');

SELECT dbms\_rowid.rowid\_block\_number(rowid), t6.\* FROM task6 t6;

UPDATE task6 SET column2=rpad('\*',3000,'\*') WHERE column1=3;

SELECT dbms\_rowid.rowid\_block\_number(rowid), t6.\* FROM task6 t6;

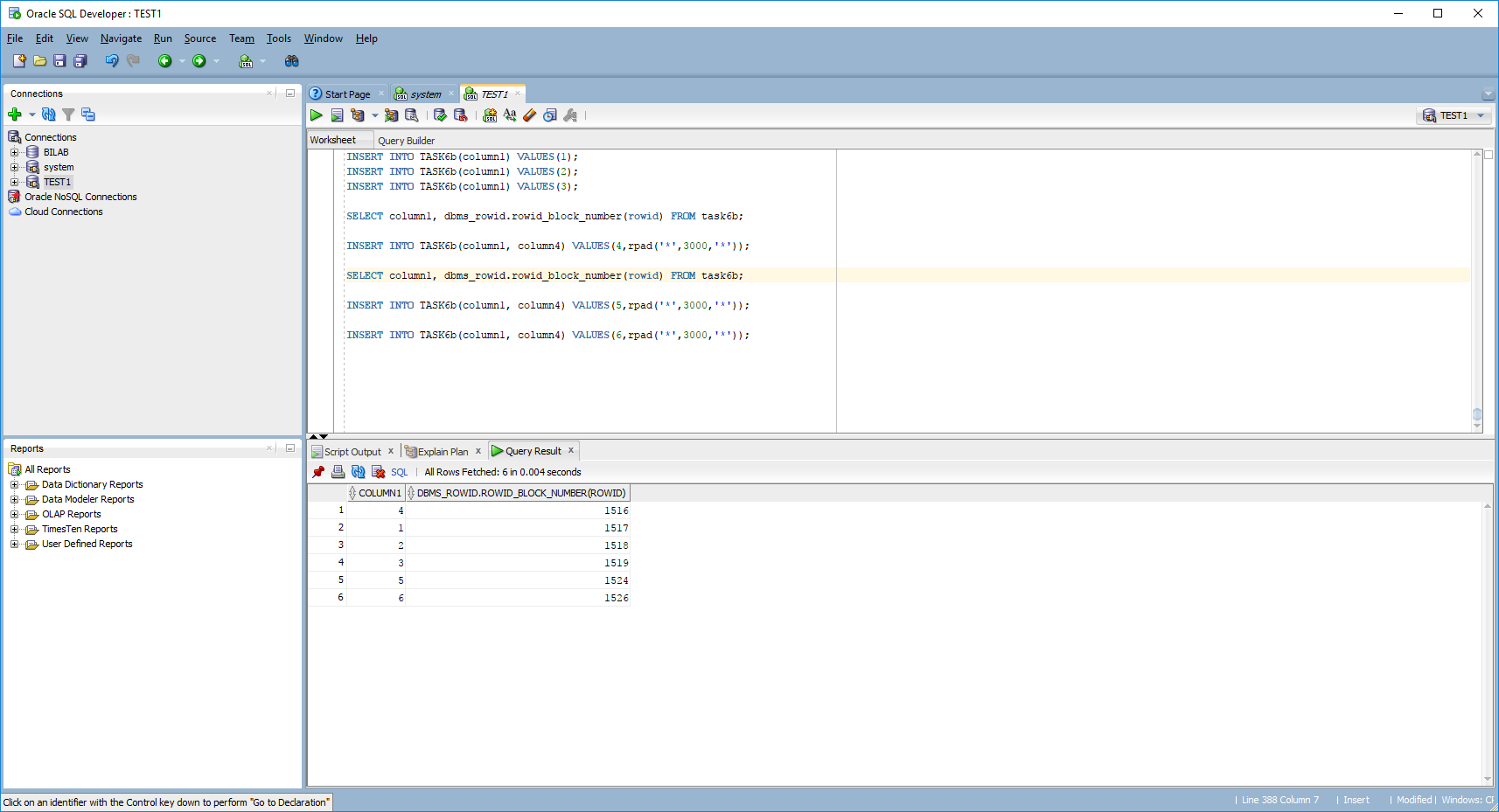
INSERT INTO TASK6(column1, column2) VALUES(5, 'test');

UPDATE task6 SET column2='test' WHERE column1=3;

INSERT INTO TASK6(column1, column2) VALUES(6, 'test');

SELECT dbms\_rowid.rowid\_block\_number(rowid), t6.\* FROM task6 t6;

Если предположить, что номера новых блоков идут по порядку, то новые строки (5 и 6) занеимают два блока.



CREATE TABLE task6b (

column1 int,

column2 varchar2(3000) default rpad('\*',3000,'\*'),

column3 varchar2(3000) default rpad('\*',3000,'\*'),

column4 varchar2(3000) default rpad('\*',1000,'\*')

);

INSERT INTO TASK6b(column1) VALUES(1);

INSERT INTO TASK6b(column1) VALUES(2);

INSERT INTO TASK6b(column1) VALUES(3);

SELECT column1, dbms\_rowid.rowid\_block\_number(rowid) FROM task6b;

INSERT INTO TASK6b(column1, column4) VALUES(4,rpad('\*',3000,'\*'));

SELECT column1, dbms\_rowid.rowid\_block\_number(rowid) FROM task6b;

INSERT INTO TASK6b(column1, column4) VALUES(5,rpad('\*',3000,'\*'));

INSERT INTO TASK6b(column1, column4) VALUES(6,rpad('\*',3000,'\*'));