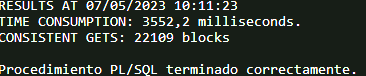
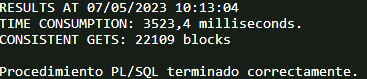
**5 iteraciones**

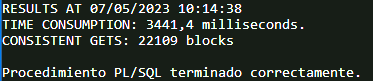
**1 vez**

****

**2 vez**

****

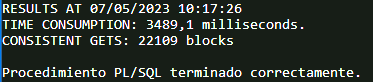
**3 vez**

****

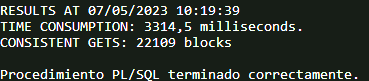
**10 iteraciones**

**10 Iteraciones**

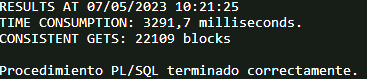
**1 vez**

****

**2 vez**

****

**3 vez**

****

**modificación de una fila**

**SET AUTOTRACE ON**

**FOR i IN 1 .. changes.COUNT LOOP**

**UPDATE tracks SET lyrics = dbms\_random.string('a', dbms\_random.value(150,2000))**

**WHERE searchk = changes(i);**

**END LOOP;**

**SELECT \* FROM TABLE(DBMS\_XPLAN.DISPLAY());**

**SELECT \* FROM ALL\_TABLES WHERE TABLE\_NAME = 'PLAN\_TABLE';**

**SAMU:**

**EXPLAIN PLAN FOR**

**DECLARE**

**TYPE changes\_type IS TABLE OF VARCHAR2(100);**

**changes changes\_type := changes\_type();**

**BEGIN**

**FOR i IN 1 .. changes.COUNT LOOP**

**UPDATE tracks SET lyrics = dbms\_random.string('a', dbms\_random.value(150, 2000)) WHERE searchk = changes(i);**

**END LOOP;**

**END;**

**/**

**SELECT \* FROM TABLE(DBMS\_XPLAN.DISPLAY());**

**DECLARE**

**TYPE tipotab IS TABLE OF VARCHAR2(20) INDEX BY BINARY\_INTEGER;**

**changes tipotab;**

**BEGIN**

**SELECT searchk**

**BULK COLLECT INTO changes**

**FROM (SELECT searchk FROM tracks ORDER BY dbms\_random.value)**

**WHERE rownum <= 1000;**

**FOR i IN 1 .. changes.COUNT LOOP**

**UPDATE tracks SET lyrics = dbms\_random.string('a', dbms\_random.value(150, 2000)) WHERE searchk = changes(i);**

**END LOOP;**

**END;**

**/**

**SELECT \* FROM TABLE(DBMS\_XPLAN.DISPLAY());**

**SET SERVEROUTPUT ON**

**EXECUTE DBMS\_SESSION.SET\_IDENTIFIER('bulk\_collect\_test')**

**CONSULTA 1 → da bien los datos**

**EXPLAIN PLAN FOR WITH authors as (select title,writer, writer musician from songs UNION select title,writer,cowriter musician from songs), authorship as (select distinct band performer, title, writer, 1 flag FROM involvement join authors using(musician) ), recordings as (select performer,tracks.title,writer from albums join tracks using(pair)), recs\_match as (select performer, round(sum(flag)\*100/count('c'),2) pct\_recs from recordings left join authorship using(performer,title,writer) group by performer), pers\_match as (select performer, round(sum(flag)\*100/count('c'),2) pct\_pers from (select performer, songtitle title, songwriter writer from performances) P left join authorship using(performer,title,writer) group by performer)**

**SELECT performer, pct\_recs, pct\_pers from recs\_match full join pers\_match using(performer);**

**SELECT \* FROM TABLE(DBMS\_XPLAN.DISPLAY());**

**CONSULTA 2 → da bien los datos**

**EXPLAIN PLAN FOR WITH recordings as (select performer,tracks.title, writer, min(rec\_date) rec, 1 token from albums join tracks using(pair) group by performer,tracks.title,writer), playbacks as (select P.performer, sum(token)\*100/count('x') percentage, avg(nvl2(rec,when-rec,rec)) as age FROM performances P left join recordings R on(P.performer=R.performer AND R.title=P.songtitle AND R.writer=P.songwriter AND P.when>R.rec) GROUP BY P.performer ORDER BY percentage desc) SELECT performer, percentage, floor(age/365.2422) years, floor(mod(age,365.2422)/30.43685) months, floor(mod(age,365.2422)-(floor(mod(age,365.2422)/30.43685)\*30.43685)) days FROM playbacks WHERE rownum<=10;**

**SELECT \* FROM TABLE(DBMS\_XPLAN.DISPLAY());**