set serveroutput on;

create or replace procedure keyword\_procedure(global\_query varchar2)

AUTHID CURRENT\_USER

is

drop\_keyword\_table varchar2(4000);

create\_keyword\_table varchar2(4000);

insert\_keyword\_table varchar2(4000);

regex\_string varchar2(1000);

keyword\_found varchar2(20);

create\_runtime\_table varchar2(4000);

insert\_into\_runtime\_table varchar2(4000);

localDbVal team1\_metadata\_table%rowType;

localClmnMap team1\_metadata\_table%rowType;

finalOutput output\_table%rowType;

colVal varchar2(150);

clmnCnt smallint;

outputCol varchar2(150);

localQuery1 varchar2(2000);

localQuery2 varchar2(2000);

localQuery3 varchar2(2000);

localQuery4 varchar2(2000);

canonicalName varchar2(100);

clmnNames varchar2(100);

tblName1 varchar2(100);

tblName2 varchar2(100);

tblName3 varchar2(100);

clmnName1 varchar2(100);

clmnName2 varchar2(100);

clmnName3 varchar2(100);

rowCnt smallint;

uptilFrom varchar2(100);

afterFrom varchar2(100);

uptilWhere varchar2(100);

afterWhere varchar2(100);

FinalQuery varchar2(2000);

cursor c1 is select keyword from user\_query\_keyword;

-- cursor c1 is select keyword from user\_query\_keyword;

cursor c2 is select keyword from sql\_keyword where UPPER(keyword) = UPPER(colVal);

--cursor c3 is select \* from team1\_metadata\_table where UPPER(canonical\_rep) = UPPER(colVal);

cursor c3 is select \* from team1\_metadata\_table where UPPER(canonical\_rep) = UPPER(canonicalName);

cursor c4 is

--select \* from team1\_metadata\_table where UPPER(table\_column\_mapping) = UPPER(canonicalName) and datatype\_inventory <> 'table';

select mstr.\* from team1\_metadata\_table mstr, user\_query\_keyword chld

where UPPER(table\_column\_mapping) = UPPER(canonicalName)

-- and (mstr.datatype\_inventory <> 'table')

and canonical\_rep = replace(chld.keyword, ',', '')

order by canonical\_rep desc

;

cursor c5 is select trim(clmnNames) from dual;

cursor c6 is select \* from output\_table;

rt\_sql varchar2(200);

--clmnCnt smallint;

sysclmn ALL\_TAB\_COLUMNS%rowType;

clmnHeading varchar2(256);

clmnName varchar2(256);

row\_num INTEGER;

cmbndData varchar2(256);

cursor c7 is select \* from ALL\_TAB\_COLUMNS WHERE TABLE\_NAME = 'OUTPUT\_TABLE';

---------------------------------------------------------------------

begin

drop\_keyword\_table := 'Drop table user\_query\_keyword';

Execute Immediate drop\_keyword\_table;

create\_keyword\_table := 'create table user\_query\_keyword(keyword varchar2(50))';

Execute Immediate create\_keyword\_table;

--regex\_string := 'regexp\_substr(' || '''' || global\_query || ''',' || '''[^ ]+''' || ',1,level) ';

regex\_string := 'regexp\_substr(' || '''' || replace(global\_query, chr(39), chr(39)||chr(39)) || ''', ' || '''[^ ,]+''' || ', 1, level) ';

insert\_keyword\_table := 'insert into user\_query\_keyword(keyword)(select ' || regex\_string || 'from dual ' || 'connect by ' || regex\_string || ' is not null)';

Execute Immediate insert\_keyword\_table;

-- DBMS\_OUTPUT.put\_line('regex\_string: ' || insert\_keyword\_table );

-- DBMS\_OUTPUT.put\_line('global query: ' || global\_query );

---------------------------------------------------------------------

localquery1 := '';

localquery2 := '';

localquery3 := '';

open c1;

Loop

fetch c1 into colVal;

if c1%notfound then

exit;

else

-- DBMS\_OUTPUT.put\_line('Cursor 1 value: ' || colVal);

localquery1 := localquery1 || ' ' || colVal;

--Identifying Canonical Keyword from User Query Keyword table

if colVal = 'from' then

select conanical\_name into canonicalName from

(

select keyword, lead(keyword,1) OVER (ORDER BY rownum) as conanical\_name from

(

select rownum, keyword from user\_query\_keyword

)

)

where keyword = 'from';

-- DBMS\_OUTPUT.put\_line('canonicalName value: ' || canonicalName);

-- DBMS\_OUTPUT.put\_line('localquery1: ' || localquery1);

end if;

end if;

---------------------------------------------------------------------

open c2;

Loop

fetch c2 into keyword\_found;

if c2%notfound then

exit;

end if;

End Loop;

close c2;

---------------------------------------------------------------------

end Loop;

close c1;

localquery2 := global\_query;

localquery3 := global\_query;

localquery4 := global\_query;

open c4;

Loop

fetch c4 into localClmnMap;

if c4%notfound then

-- DBMS\_OUTPUT.put\_line('Cursor 4 localClmnMap value: ' || localClmnMap.column\_ordering);

exit;

else

if localClmnMap.Datatype\_Inventory = 'table' then

uptilFrom := substr(localquery2, 1, instr(lower(localquery2), 'from')+3 );

afterFrom := substr(localquery2, instr(lower(localquery2), 'from')+4, 50);

afterFrom := regexp\_replace(afterFrom, localClmnMap.CANONICAL\_REP, localClmnMap.column\_inventory, 1, 1);

localquery2 := uptilFrom || afterFrom;

uptilFrom := substr(localquery3, 1, instr(lower(localquery3), 'from')+3 );

afterFrom := substr(localquery3, instr(lower(localquery3), 'from')+4, 50);

afterFrom := regexp\_replace(afterFrom, localClmnMap.CANONICAL\_REP, localClmnMap.column\_ordering, 1, 1);

localquery3 := uptilFrom || afterFrom;

uptilFrom := substr(localquery4, 1, instr(lower(localquery4), 'from')+3 );

afterFrom := substr(localquery4, instr(lower(localquery4), 'from')+4, 50);

afterFrom := regexp\_replace(afterFrom, localClmnMap.CANONICAL\_REP, localClmnMap.column\_ordering2, 1, 1);

localquery4 := uptilFrom || afterFrom;

else

localquery2 := replace(localquery2, localClmnMap.CANONICAL\_REP, localClmnMap.column\_inventory);

localquery3 := replace(localquery3, localClmnMap.CANONICAL\_REP, localClmnMap.column\_ordering);

localquery4 := replace(localquery4, localClmnMap.CANONICAL\_REP, localClmnMap.column\_ordering2);

end if;

end if;

End Loop;

close c4;

FinalQuery := localquery2 || ' UNION ALL ' || localquery3 || ' UNION ALL ' || localquery4;

-- DBMS\_OUTPUT.put\_line('localquery2: ' || localquery2);

-- DBMS\_OUTPUT.put\_line('localquery3: ' || localquery3);

-- DBMS\_OUTPUT.put\_line('localquery4: ' || localquery4);

DBMS\_OUTPUT.put\_line('Final Query: ' || FinalQuery);

-----------------------------------------------------------------------

--Dynamic Table Creation

clmnNames := substr(global\_query, 8, (instr(lower(global\_query), 'from')-8) );

-- DBMS\_OUTPUT.put\_line('Column Name from select from SP: ' || clmnNames);

clmnNames := trim(clmnNames);

regex\_string := 'regexp\_substr(' || '''' || clmnNames || ''', ' || '''[^,]+''' || ',1,level) ';

drop\_keyword\_table := 'Drop table user\_query\_keyword';

Execute Immediate drop\_keyword\_table;

create\_keyword\_table := 'create table user\_query\_keyword(keyword varchar2(50))';

Execute Immediate create\_keyword\_table;

insert\_keyword\_table := 'insert into user\_query\_keyword(keyword)(select ' || regex\_string || 'from dual ' || 'connect by ' || regex\_string || ' is not null)';

Execute Immediate insert\_keyword\_table;

clmnCnt:=1;

create\_runtime\_table := 'create table output\_table(';

open c1;

Loop

fetch c1 into colVal;

if c1%notfound then

exit;

else

--create\_runtime\_table := create\_runtime\_table || trim(colVal) || ' varchar2(4000), ';

create\_runtime\_table := create\_runtime\_table || 'clmn'|| clmncnt || ' varchar2(4000), ';

clmncnt := clmncnt + 1;

end if;

end loop;

close c1;

create\_runtime\_table := substr(create\_runtime\_table,0,length(create\_runtime\_table)-2);

create\_runtime\_table := create\_runtime\_table || ')';

-- DBMS\_OUTPUT.put\_line(create\_runtime\_table);

Execute Immediate 'drop table output\_table';

Execute Immediate create\_runtime\_table;

---------------------------Data Insertion in Runtime Table--------------------------------------------

--insert\_into\_runtime\_table := 'Insert into output\_table(' || clmnNames || ') ' || finalquery ;

insert\_into\_runtime\_table := 'Insert into output\_table ' || localquery2 ;

-- DBMS\_OUTPUT.put\_line('Insert Query1: ' || insert\_into\_runtime\_table);

Execute Immediate insert\_into\_runtime\_table;

insert\_into\_runtime\_table := 'Insert into output\_table ' || localquery3 ;

-- DBMS\_OUTPUT.put\_line('Insert Query2: ' || insert\_into\_runtime\_table);

Execute Immediate insert\_into\_runtime\_table;

insert\_into\_runtime\_table := 'Insert into output\_table ' || localquery4 ;

-- DBMS\_OUTPUT.put\_line('Insert Query3: ' || insert\_into\_runtime\_table);

Execute Immediate insert\_into\_runtime\_table;

---------------------------Display final output in terminal-------------------------------------------

for i in (select \* from user\_query\_keyword)

loop

clmnHeading := clmnHeading || upper(i.keyword) || ', ';

end loop;

dbms\_output.put\_line('----------------------------------');

dbms\_output.put\_line(substr(clmnHeading, 0, length(clmnHeading)-2));

dbms\_output.put\_line('----------------------------------');

open c7;

Loop fetch c7 into sysclmn;

if c7%notfound then

exit;

end if;

clmnName := clmnName || sysclmn.COLUMN\_NAME || ' || ' || ''', '' || ';

end loop;

close c7;

-- dbms\_output.put\_line(clmnName);

for i in (select row\_number() over(order by clmn1) as row\_num, clmn1 from output\_table)

loop

rt\_sql := 'select \* from (select row\_number() over(order by clmn1) as row\_num, (' || substr(clmnName, 0, length(clmnName)-12) || ') as concat\_data from output\_table ) where row\_num = ' || i.row\_num || '';

-- dbms\_output.put\_line ( rt\_sql ) ;

execute immediate rt\_sql into row\_num, cmbndData;

dbms\_output.put\_line ( cmbndData) ;

end loop;

dbms\_output.new\_line () ;

dbms\_output.put\_line ('Processing Completed: ' || systimestamp) ;

dbms\_output.put\_line('-----------------------------------------------------------------------');

--------------------------------------------------------------

Exception

when others then

if sqlcode != -942 then

raise;

else

DBMS\_OUTPUT.put\_line('the table did not exist!');

end if;

end;