

Учреждение образования
«Белорусский государственный университет информатики и
радиоэлектроники»
Кафедра информатики

Отчёт

Лабораторная работа №3

Выполнил:
студент группы №853504
Кузьма В.В.

Проверил:
Чащин С.В.

Минск 2021

ИНИЦИАЛИЗАЦИЯ

DROP TABLESPACE tbs_1 INCLUDING CONTENTS AND DATAFILES;

DROP USER prod CASCADE;

CREATE TABLESPACE tbs_1

DATAFILE 'tbs_1.dat'

SIZE 10M

REUSE

AUTOEXTEND ON NEXT 10M MAXSIZE 200M;

CREATE USER prod

IDENTIFIED BY 321

DEFAULT TABLESPACE tbs_1

QUOTA 200M on tbs_1;

GRANT create session TO prod;

GRANT create table TO prod;

GRANT create view TO prod;

GRANT create any trigger TO prod;

GRANT create any procedure TO prod;

GRANT create sequence TO prod;

GRANT create synonym TO prod;

--conn Prod;

CREATE TABLE prod.products

(product_id number(10) not null,

product_name varchar2(50) not null,

category varchar2(50),

CONSTRAINT products_pk PRIMARY KEY (product_id)

);

DROP TABLESPACE tbs_2 INCLUDING CONTENTS AND DATAFILES;

DROP USER dev CASCADE;

CREATE TABLESPACE tbs_2

DATAFILE 'tbs_2.dat'

SIZE 10M

REUSE

```

        AUTOEXTEND ON NEXT 10M MAXSIZE 200M;

CREATE USER dev
    IDENTIFIED BY 321
    DEFAULT TABLESPACE tbs_2
    QUOTA 200M on tbs_2;

GRANT create session TO dev;
GRANT create table TO dev;
GRANT create view TO dev;
GRANT create any trigger TO dev;
GRANT create any procedure TO dev;
GRANT create sequence TO dev;
GRANT create synonym TO dev;

CREATE TABLE dev.products
    ( product_id number(10) not null,
      product_name varchar2(50) not null,
--    category varchar2(50),
      CONSTRAINT products_pk PRIMARY KEY (product_id)
    );

CREATE TABLE dev.qwer
    ( qwer_id number(10) not null,
      qwer_name varchar2(50) not null,
      CONSTRAINT qwer_pk PRIMARY KEY (qwer_id)
    );

SET SERVEROUTPUT ON;

```

ЗАДАНИЕ 1.

Написать процедуру/функцию на вход которой подаются два текстовых параметра (dev_schema_name, prod_schema_name), которые являются названиями схем баз данных (условно схема для разработки(Dev) и промышленная схема(Prod)), на выход процедура должна предоставить перечень таблиц, которые есть в схеме Dev, но нет в Prod, либо в которых различается структура таблиц.

```

create or replace procedure COMPARE_SCHEMES(schema1 in varchar2, schema2 in varchar2)
as
    diff NUMBER := 0;
begin
    -- DIFFERENCE IN COLUMNS
    dbms_output.put_line('Comparing 2 schemes, printing difference in tables');
    FOR same_table IN (SELECT table_name FROM all_tables tables1 WHERE OWNER =
schema1
                        INTERSECT
                        SELECT tables2.table_name FROM all_tables tables2 WHERE OWNER =
schema2) LOOP
        SELECT COUNT(*) INTO diff FROM
            (SELECT table1.COLUMN_NAME name,table1.DATA_TYPE FROM all_tab_columns
table1 WHERE OWNER=schema1 AND TABLE_NAME= same_table.table_name) cols1
            FULL JOIN
            (SELECT table2.COLUMN_NAME name,table2.DATA_TYPE FROM all_tab_columns
table2 WHERE OWNER=schema2 AND TABLE_NAME = same_table.table_name) cols2
            ON cols1.name = cols2.name
        WHERE cols1.name IS NULL OR cols2.name IS NULL;

        IF diff > 0 THEN
            dbms_output.put_line('Table structure of ' || same_table.table_name || ' is different in ' ||
schema1 || ' and ' || schema2);
        ELSE
            dbms_output.put_line('Table structure of ' || same_table.table_name || ' the same');
        END IF;
    END LOOP;
end COMPARE_SCHEMES;

/

create or replace procedure COMPARE_SCHEMES_TABLES
(
    schema1 in varchar2
    , schema2 in varchar2

```

```

) as
begin
-- DIFFERENCE IN TABLES

    dbms_output.put_line('Comparing 2 schemes, printing difference in tables structures');

    FOR other_table IN (SELECT tables1.table_name name FROM all_tables tables1 WHERE
tables1.OWNER = schema1

                        MINUS

                        SELECT tables2.table_name FROM all_tables tables2 WHERE
tables2.OWNER=schema2) LOOP

        dbms_output.put_line('Table ' || other_table.name || ' is in ' || schema1 || ' but not in ' ||
schema2);

    END LOOP;

    FOR other_table IN (SELECT tables2.table_name name FROM all_tables tables2 WHERE
tables2.OWNER=schema2

                        MINUS

                        SELECT tables1.table_name FROM all_tables tables1 WHERE tables1.OWNER
= schema1) LOOP

        dbms_output.put_line('Table ' || other_table.name || ' is in ' || schema2 || ' but not in ' ||
schema1);

    END LOOP;
end COMPARE_SCHEMES_TABLES;

```

```

DROP TABLE fk_tmp;
CREATE TABLE fk_tmp(
    id INT,
    CHILD_OBJ VARCHAR2(100),
    PARENT_OBJ VARCHAR2(100)
);

```

```

create or replace procedure SCHEME_TABLES_ORDER(schema_name in varchar2) as
begin
-- DIFFERENCE IN TABLES

    EXECUTE IMMEDIATE 'TRUNCATE TABLE fk_tmp';

    dbms_output.put_line('Showing tables order in schema');

```

```

FOR schema_table IN (SELECT tables1.table_name name FROM
                        all_tables tables1 WHERE OWNER = schema_name) LOOP

    INSERT INTO fk_tmp (CHILD_OBJ, PARENT_OBJ)
        SELECT DISTINCT a.table_name, c_pk.table_name r_table_name
        FROM all_cons_columns a
            JOIN all_constraints c ON a.owner = c.owner AND a.constraint_name =
c.constraint_name
            JOIN all_constraints c_pk ON c.r_owner = c_pk.owner AND c.r_constraint_name =
c_pk.constraint_name
        WHERE c.constraint_type = 'R' AND a.table_name = schema_table.name;

    IF SQL%ROWCOUNT = 0 THEN
        dbms_output.put_line( schema_table.name);
    END IF;

END LOOP;

FOR fk_cur IN (
    SELECT CHILD_OBJ,PARENT_obj,CONNECT_BY_ISCYCLE
    FROM fk_tmp
    CONNECT BY NOCYCLE PRIOR PARENT_OBJ = child_obj
    ORDER BY LEVEL
) LOOP
    IF fk_cur.CONNECT_BY_ISCYCLE = 0 THEN
        dbms_output.put_line(fk_cur.CHILD_OBJ);
    ELSE
        dbms_output.put_line('CYCLE IN TABLE' || fk_cur.CHILD_OBJ);
    END IF;
END LOOP;

end SCHEME_TABLES_ORDER;

```

```
begin
    COMPARE_SCHEMES('DEV', 'PROD');
    COMPARE_SCHEMES_TABLES('DEV', 'PROD');
    SCHEME_TABLES_ORDER('DEV');
    SCHEME_TABLES_ORDER('PROD');
end;
```

```
Procedure COMPARE_SCHEMES compiled

Procedure COMPARE_SCHEMES_TABLES compiled

Table FK_TMP dropped.

Table FK_TMP created.

Procedure SCHEME_TABLES_ORDER compiled

Comparing 2 schemes, printing difference in tables
Table structure of PRODUCTS is different in DEV and PROD
Comparing 2 schemes, printing difference in tables structures
Table QWER is in DEV but not in PROD
Showing tables order in schema
QWER
PRODUCTS
Showing tables order in schema
PRODUCTS

PL/SQL procedure successfully completed.
```

ЗАДАНИЕ 2.

Доработать предыдущий скрипт с учетом возможности сравнения не только таблиц, но и процедур, функций, индексов пакетов.

```
SELECT * FROM ALL_OBJECTS WHERE OBJECT_TYPE = 'TABLE' AND OWNER = 'DEV';
```

```

create or replace procedure COMPARE_SCHEMES_BY_OBJECT(schema1 in varchar2,
schema2 in varchar2, object_type in varchar2)

as

    diff NUMBER := 0;

begin

-- DIFFERENCE IN COLUMNS

    dbms_output.put_line('Comparing 2 schemes, printing difference in tables');

    FOR obj_type IN ('PROCEDURE', 'PACKAGE', 'INDEX', 'TABLE') LOOP

        FOR same_object IN (

            SELECT objects1.object_name FROM ALL_OBJECTS objects1 WHERE
OWNER = schema1 AND OBJECT_TYPE = obj_type

            INTERSECT

            SELECT objects2.table_name FROM ALL_OBJECTS objects2 WHERE
OWNER = schema2) AND OBJECT_TYPE = obj_type LOOP

                SELECT COUNT(*) INTO diff FROM

                    (SELECT object1.OBJECT_NAME name, table1.DATA_TYPE FROM
all_tab_columns table1 WHERE OWNER=schema1 AND TABLE_NAME=
same_table.table_name) cols1

                    FULL JOIN

                    (SELECT object2.OBJECT_NAME name, table2.DATA_TYPE FROM
all_tab_columns table2 WHERE OWNER=schema2 AND TABLE_NAME =
same_table.table_name) cols2

                    ON cols1.name = cols2.name

                WHERE cols1.name IS NULL OR cols2.name IS NULL;

            IF diff > 0 THEN

                dbms_output.put_line('Table structure of ' || same_table.table_name || ' is different in ' ||
schema1 || ' and ' || schema2);

            ELSE

                dbms_output.put_line('Table structure of ' || same_table.table_name || ' the same');

            END IF;

        END LOOP;

    END LOOP;

end COMPARE_SCHEMES;

```


create or replace procedure COMPARE_SCHEMES_TABLES

(

 schema1 in varchar2

, schema2 in varchar2

) as

begin

-- DIFFERENCE IN TABLES

 dbms_output.put_line('Comparing 2 schemes, printing difference in tables structures');

 FOR other_table IN (SELECT tables1.table_name name FROM all_tables tables1 WHERE
tables1.OWNER = schema1

 MINUS

 SELECT tables2.table_name FROM all_tables tables2 WHERE
tables2.OWNER=schema2) LOOP

 dbms_output.put_line('Table ' || other_table.name || ' is in ' || schema1 || ' but not in ' ||
schema2);

 END LOOP;

 FOR other_table IN (SELECT tables2.table_name name FROM all_tables tables2 WHERE
tables2.OWNER=schema2

 MINUS

 SELECT tables1.table_name FROM all_tables tables1 WHERE tables1.OWNER
= schema1) LOOP

 dbms_output.put_line('Table ' || other_table.name || ' is in ' || schema2 || ' but not in ' ||
schema1);

 END LOOP;

end COMPARE_SCHEMES_TABLES;

/

DROP TABLE fk_tmp;

CREATE TABLE fk_tmp(

 id INT,

 CHILD_OBJ VARCHAR2(100),

 PARENT_OBJ VARCHAR2(100)

);

create or replace procedure SCHEME_TABLES_ORDER(schema_name in varchar2) as

```

begin
-- DIFFERENCE IN TABLES
EXECUTE IMMEDIATE 'TRUNCATE TABLE fk_tmp';
dbms_output.put_line('Showing tables order in schema');
FOR schema_table IN (SELECT tables1.table_name name FROM
                      all_tables tables1 WHERE OWNER = schema_name) LOOP

    INSERT INTO fk_tmp (CHILD_OBJ, PARENT_OBJ)
        SELECT DISTINCT a.table_name, c_pk.table_name r_table_name
        FROM all_cons_columns a
            JOIN all_constraints c ON a.owner = c.owner AND a.constraint_name =
c.constraint_name
            JOIN all_constraints c_pk ON c.r_owner = c_pk.owner AND c.r_constraint_name =
c_pk.constraint_name
        WHERE c.constraint_type = 'R' AND a.table_name = schema_table.name;

    IF SQL%ROWCOUNT = 0 THEN
        dbms_output.put_line( schema_table.name);
    END IF;

END LOOP;

FOR fk_cur IN (
    SELECT CHILD_OBJ,PARENT_obj,CONNECT_BY_ISCYCLE
    FROM fk_tmp
    CONNECT BY NOCYCLE PRIOR PARENT_OBJ = child_obj
    ORDER BY LEVEL
) LOOP
    IF fk_cur.CONNECT_BY_ISCYCLE = 0 THEN
        dbms_output.put_line(fk_cur.CHILD_OBJ);
    ELSE
        dbms_output.put_line('CYCLE IN TABLE' || fk_cur.CHILD_OBJ);
    END IF;
END LOOP;

```

```
        END IF;
    END LOOP;
end SCHEME_TABLES_ORDER;
```

```
begin
    COMPARE_SCHEMES('DEV', 'PROD');
    COMPARE_SCHEMES_TABLES('DEV', 'PROD');
    SCHEME_TABLES_ORDER('DEV');
    SCHEME_TABLES_ORDER('PROD');
end;
```

```
Table FK_TMP dropped.
```

```
Table FK_TMP created.
```

```
Procedure SCHEME_TABLES_ORDER compiled
```

```
Comparing 2 schemes, printing difference in tables
```

```
Table structure of PRODUCTS is different in DEV and PROD
```

```
Comparing 2 schemes, printing difference in tables structures
```

```
Table QWER is in DEV but not in PROD
```

```
Showing tables order in schema
```

```
QWER
```

```
PRODUCTS
```

```
Showing tables order in schema
```

```
PRODUCTS
```

```
PL/SQL procedure successfully completed.
```

ЗАДАНИЕ 3.

Доработать предыдущий скрипт с генерацией ddl-скрипта на обновление объектов, а также с учетом необходимости удаления в схеме prod объектов, отсутствующих в схеме dev.

CREATE OR REPLACE PROCEDURE REPLACE_OBJECT

(

 schema1 IN VARCHAR2,

 schema2 IN VARCHAR2,

 object_type IN VARCHAR2,

 object_name IN VARCHAR2

)

AS

 query_string VARCHAR2(100);

BEGIN

 FOR src IN (SELECT line, text FROM ALL_SOURCE WHERE OWNER = schema1 AND
NAME = object_name)

 LOOP

 IF src.line =1 THEN

 query_string := 'CREATE OR REPLACE ' || REPLACE(src.text, LOWER(object_name),
schema2 || '.' || object_name);

 ELSE

 query_string := query_string || src.text;

 END IF;

 END LOOP;

 EXECUTE IMMEDIATE query_string;

END REPLACE_OBJECT;

/

CREATE OR REPLACE PROCEDURE CREATE_OBJECT

(

 schema1 IN VARCHAR2,

```

    schema2 IN VARCHAR2,

    object_type IN VARCHAR2,

    object_name IN VARCHAR2
)
AS

    query_string VARCHAR2(100);

BEGIN

    FOR src IN (SELECT line, text FROM ALL_SOURCE WHERE OWNER = schema1 AND
NAME = object_name)

        LOOP

            IF src.line =1 THEN

                query_string := 'CREATE ' || REPLACE(src.text, LOWER(object_name), schema2 || '.' ||
object_name);

            ELSE

                query_string := query_string || src.text;

            END IF;

        END LOOP;

        EXECUTE IMMEDIATE query_string;

    END CREATE_OBJECT;

/

CREATE OR REPLACE PROCEDURE DELETE_OBJECT

(

    schema1 IN VARCHAR2,

    object_type IN VARCHAR2,

    object_name IN VARCHAR2

)

```

AS

delete_query VARCHAR(100);

BEGIN

delete_query := 'DROP ' || object_type || ' ' || schema1 || '.' || object_name;

EXECUTE IMMEDIATE delete_query;

END DELETE_OBJECT;

/

CREATE OR REPLACE PROCEDURE COMPARE_OBJECTS

(

schema1 IN VARCHAR2,

schema2 IN VARCHAR2,

object_type IN VARCHAR2

)

AS

diff NUMBER :=0;

BEGIN

FOR pair IN (SELECT obj1.NAME AS name1, obj2.NAME AS name2

FROM

(SELECT OBJECT_NAME name FROM ALL_OBJECTS

WHERE OBJECT_TYPE = object_type AND OWNER = schema1) obj1

FULL JOIN

(SELECT OBJECT_NAME name FROM ALL_OBJECTS

WHERE OBJECT_TYPE = object_type AND OWNER = schema2) obj2

ON obj1.name = obj2.name) LOOP

IF pair.name1 IS NULL THEN

```

DELETE_OBJECT(schema2,object_type, pair.name2);

dbms_output.put_line('D');

ELSIF pair.name2 IS NULL THEN

CREATE_OBJECT(schema1, schema2, object_type, pair.name1);

dbms_output.put_line('C');

ELSE

SELECT COUNT(*) INTO diff

FROM

all_source src1 FULL JOIN all_source src2

ON src1.name = src2.name

WHERE src1.name= pair.name1 AND src1.line = src2.line

AND src1.text != src2.text

AND src1.OWNER = schema1 AND src2.OWNER = schema2;

IF diff > 0 THEN

REPLACE_OBJECT(schema1,schema2,object_type,pair.name1);

dbms_output.put_line('R');

END IF;

END IF;

END LOOP;

END COMPARE_OBJECTS;

/

EXEC COMPARE_OBJECTS('DEV', 'PROD', 'PROCEDURE')

```