Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Отчёт к лабораторной работе №3

Выполнил:

студент гр. 953501

Кондрашов И.Д.

Проверил:

Чащин C. В.

Минск 2022

# 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;

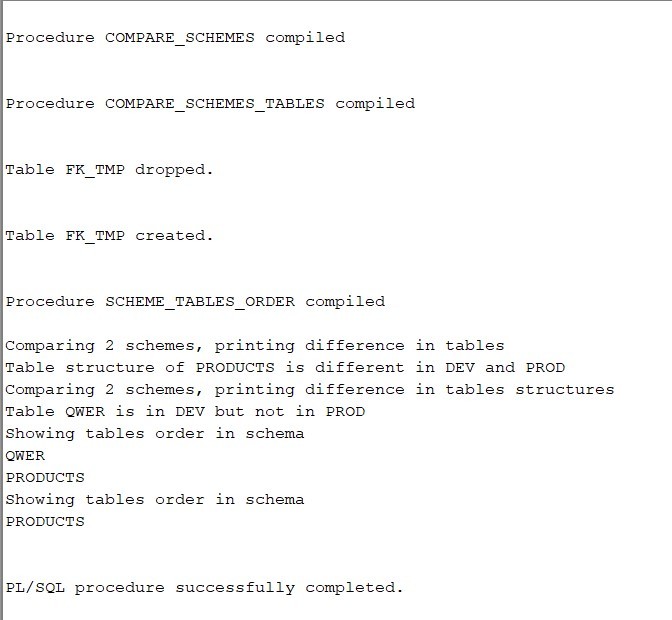
## END LOOP;

end SCHEME\_TABLES\_ORDER;

begin

## COMPARE\_SCHEMES('DEV', 'PROD'); COMPARE\_SCHEMES\_TABLES('DEV', 'PROD'); SCHEME\_TABLES\_ORDER('DEV'); SCHEME\_TABLES\_ORDER('PROD');

end;



# 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);

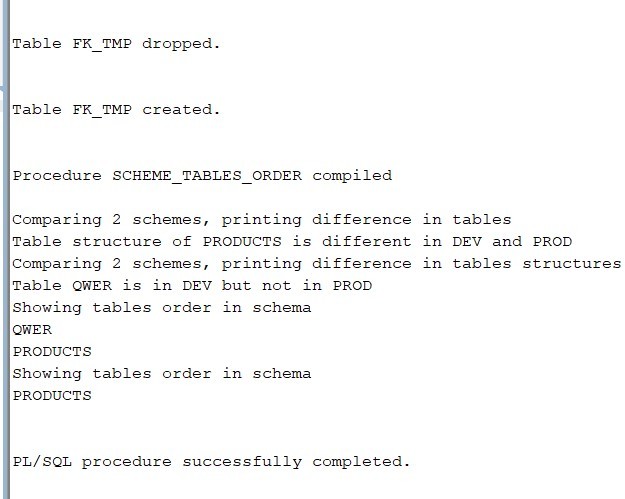
## END IF; END LOOP;

end SCHEME\_TABLES\_ORDER;

begin

## COMPARE\_SCHEMES('DEV', 'PROD'); COMPARE\_SCHEMES\_TABLES('DEV', 'PROD'); SCHEME\_TABLES\_ORDER('DEV'); SCHEME\_TABLES\_ORDER('PROD');

end;



# 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')