

## Database Security – master, 2nd year

### Laboratory 6.

## Data masking in Oracle

Keywords: <ul style="list-style-type: none"><li>• Data masking</li></ul>	<ul style="list-style-type: none"><li>• Oracle Data Pump</li><li>• Data export &amp; import</li></ul>
--	---

### 1. Introduction

- When installed on production systems, applications run on real data. However, during the development and testing of the system, programmers need a test database.
- Test data is often extracted from production databases to simulate a test framework as close as possible to the actual operating framework.
- There are data which are sensitive to privacy policies, so they cannot simply be given for testing. These data include: personal identification numbers (“CNP”s), card PINs, GPS locations, employee identifiers etc.

**There are cases in which confidential data should be hidden.**

- There are several tools provided by Oracle (*Oracle Data Masking Pack*, *Oracle Data Pump*) and there is also the possibility to implement program units tailored to the needs of the application.
- Regardless of the chosen method, there are some important aspects that require special attention:
  - the degree to which the chosen method preserves the distribution of data values. Experts point out that application performance can change dramatically if masking changes the histogram of data in a column and / or the length of values in a column;
  - preservation of referential integrity: if a primary key needs to be hidden, it is necessary to ensure that the corresponding external key keys are modified in cascade;
  - the effort required for the data masking operation: in case of differentiated permutation of values on different columns, several full scan operations equal to the number of columns in the table must be performed.

### 2. Oracle Data Pump

- Next, the method of masking data in Oracle using **Oracle Data Pump** will be presented, with an example on the e-learning application discussed in the previous labs.
- In Windows, we find the *expdp.exe* and *impdp.exe* executable files in the BIN folder in the Oracle installation path (for example, C:\app2\<user>\product\11.2.0\dbhome\_1\bin or D:\app\<user>\virtual\product\12.2.0\dbhome\_1\bin). Alternatively, jobs can be created using the specific API (<https://oracle-base.com/articles/10g/oracle-data-pump-10g#GettingStarted>).

- In order to mask the data using the two executables mentioned above, we will go through the following steps:

1) Creating a PL/SQL package in which we include data masking functions

2) Export the concerned tables by applying the masking functions in the package:

```
expdp user/password tables=table1,table2,...
remap_data=table1.col_x:package.masking_function
remap_data=table2.col_y:package.masking_function 2 ...
directory=DIR_NAME dumpfile=FILE_NAME.dmp
```

3) Import tables with hidden data:

```
impdp user/password directory= DIR_NAME dumpfile= FILE_NAME.dmp
TABLES= table1,table2,... remap_table= table1:alias1
remap_table=table2:alias2 ...
```

### Exercise:

Let the STUDENT and RESOLVE tables have the following structure:

```
SQL> desc student;
Name                Null?    Type
-----
ID                   NOT NULL NUMBER
NUME                 VARCHAR2(30)
PRENUME              VARCHAR2(30)
ANUL                 NUMBER
SPECIALIZARE         VARCHAR2(3)
GRUPA                NUMBER

SQL> desc rezolva
Name                Null?    Type
-----
COD_STUDENT          NOT NULL NUMBER
COD_TEMA              NOT NULL NUMBER
DATA_UPLOAD          DATE
NOTA                  NUMBER(4,2)
DATA_CORECTARE       DATE
```

Use data masking for the columns *id*, *first name*, *last name*, *group* in the STUDENT table. The following is known: the column *student\_id* in the SOLVES table is a foreign key and refers to the primary key (*id*) of the STUDENT table.

**Remark:** The tables are populated with the following initial data:

```
insert into student values(135,'Avramescu','Anton',5,'Inf',531);
insert into student values(212,'Antim','Tudor',5,'Inf',532);
insert into student values(314,'Tinca','Ana',5,'Inf',531);
insert into student values(411,'Caludescu','Aristida',5,'Inf',532);
```

```
insert into subject values(1,'DB Security');
insert into homework values(1,1,SYSDATE-45,1);
insert into homework values(2,1,SYSDATE-30,1);
insert into homework values(3,1,SYSDATE+7,2);
insert into homework values(4,1,SYSDATE+28,1);
```

```
insert into solves values(135,1,sysdate-50,null,null);
insert into solves values(212,1,sysdate-45,null,null);
insert into solves values(135,2,sysdate-35,null,null);
insert into solves values(212,2,sysdate-35,null,null);
```

```
insert into solves values(314,2,sysdate-30,null,null);
insert into solves values(135,3,sysdate,null,null);
```

```
COMMIT;
```

```
SQL> select * from student;
```

ID	NUME	PRENUME	ANUL	SPE	GRUPA
135	Avramescu	Anton	5	Inf	531
212	Antim	Tudor	5	Inf	532
314	Iinca	Ana	5	Inf	531
411	Caludescu	Aristida	5	Inf	532

```
SQL> select * from rezolva;
```

COD_STUDENT	COD_TEMA	DATA_UPLO	NOTA	DATA_CORE
135	1	04-NOV-12		
212	1	09-NOV-12		
135	2	19-NOV-12		
212	2	19-NOV-12		
314	2	24-NOV-12		
135	3	24-DEC-12		

```
6 rows selected.
```

### Solution:

- **Step 1**) Creating a PL/SQL package that includes data masking functions
  - We want to keep the first character in the first name and last name, and for the id and the group we want to keep the first digit.
  - We will use an overloaded function in the package for strings, respectively for the primary key *id*, and another function for masking the group number.
  - Since the column *id* in the STUDENT table is a primary key which is referred by a foreign key, we will use an index-by table that holds the hidden value of each *id* in the position *id*. The same element in the table will be used to mask the value of the column *student\_id* in the table SOLVES.

```
create or replace package pack_masking is
```

```
    function f_masking(str varchar2) return varchar2;
```

```
    function f_masking(nb number) return number;
```

```
    function f_masking_group(nb number) return number;
```

```
end;
```

```
/
```

```
create or replace package body pack_masking is
```

```
    type t_tabind is table of number index by pls_integer;
    v_tabind t_tabind;
```

```
    function f_masking (str varchar2) return varchar2 is
```

```
        v_str varchar2(100);
```

```
        v_len number;
```

```
    begin
```

```
        v_str := substr(str,1,1);
```

```
        select length(str) into v_len from dual;
```

```
        v_str := rpad(v_str,v_len,'*'); -- we keep only the first
                                         --character and we fill with "*" up to the
                                         --original string's length
```

```

        return v_str;
    end f_masking;

function f_masking(nb number) return number is
    v_len number;
    v_min number;
    v_max number;
    v_seed VARCHAR2(100);
    v_new_nb number;
begin
    if v_tabind.exists(nb) then
        return v_tabind(nb); --it will be used when we mask the
                               --foreign key in the table SOLVES
    else
        --we generate a random number which should start with the
        -- same digit and should have the same length as nb
        v_len:=length(to_char(nb));
        dbms_output.put_line('length='||v_len);
        v_min:=to_number(rpad(substr(to_char(nb),1,1),v_len,'0'));
        v_max:=to_number(rpad(substr(to_char(nb),1,1),v_len,'9'));
        dbms_output.put_line('min='||v_min||' max=' || v_max);

        v_seed:=TO_CHAR(SYSTIMESTAMP,'YYYYDDMMHH24MISSFFFF');
        DBMS_RANDOM.seed (val => v_seed);

        v_new_nb:=round(DBMS_RANDOM.value(
                                low => v_min, high => v_max),0);
        v_tabind(nb):=v_new_nb;
        return v_new_nb;
    end if;
end f_masking;

function f_masking_group (nr number) return number is
    v_new_nb number;
    v_len number;
begin
    v_len:=length(to_char(nr));
    v_new_nb:=to_number(rpad(substr(to_char(nr),1,1),v_len,'0'));
    return v_new_nb;
end;

end;
/

```

- **Step 2)** Export the concerned tables by applying the masking function in the package
  - First, we will create a directory object:

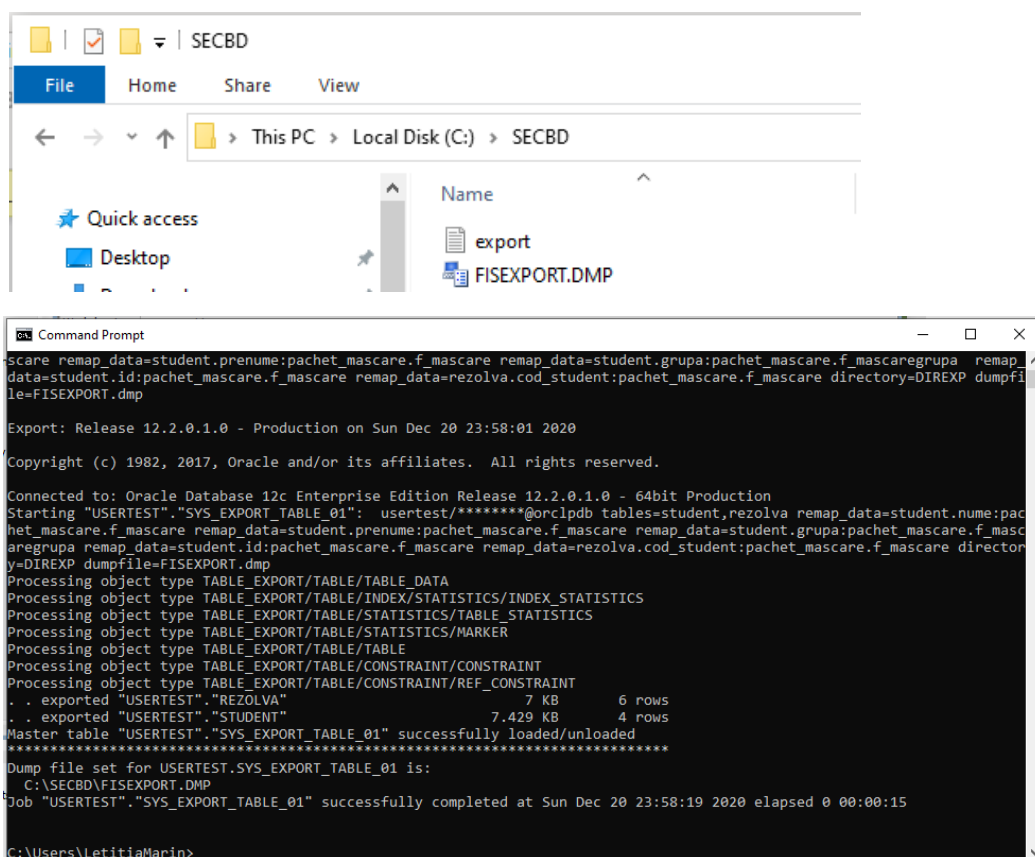
```
CREATE OR REPLACE DIRECTORY DIREXP AS 'C:\SECB'D';
```

- Secondly, we will execute the command of data export with masking (in Windows, from Command Prompt):

```
expdp usertest/usertest@orclpdb tables=student,solves
remap_data=student.last_name:pack_masking.f_masking
remap_data=student.first_name:pack_masking.f_masking
remap_data=student.grupa:pack_masking.f_masking_group
remap_data=student.id:pack_masking.f_masking
remap_data=solves.student_id:pack_masking.f_masking
directory=DIREXP dumpfile=EXPORT_FILE.dmp
```

*Note that there will be NO semicolons at the end of the export command.*

- The result will be found in the directory C:\SECB'D:



- Step 3)** Import of the tables with masked data (the command does NOT end by „;“!)  

```
impdp usertest/usertest@orclpdb directory=DIREXP dumpfile=
EXPORT_FILE.DMP TABLES=student,solves remap_table=student:stud1
remap_table=solves:solv1
```

```

C:\Users\LetitiaMarin>impdp userestest/userestest@orclpdb directory=DIREXP dumpfile=FIEXPORT.DMP TABLES=student,rezolva remap_table=student:stud1 remap_table=rezolva:rez1

Import: Release 12.2.0.1.0 - Production on Mon Dec 21 00:02:15 2020

Copyright (c) 1982, 2017, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
Master table "USERESTEST"."SYS_IMPORT_TABLE_01" successfully loaded/unloaded
Starting "USERESTEST"."SYS_IMPORT_TABLE_01": userestest/*****@orclpdb directory=DIREXP dumpfile=FIEXPORT.DMP TABLES=student,rezolva remap_table=student:stud1 remap_table=rezolva:rez1
Processing object type TABLE_EXPORT/TABLE/TABLE
Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
.. imported "USERESTEST"."REZ1" 7 KB 6 rows
.. imported "USERESTEST"."STUD1" 7.429 KB 4 rows
Processing object type TABLE_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
Processing object type TABLE_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type TABLE_EXPORT/TABLE/CONSTRAINT/REF_CONSTRAINT
Processing object type TABLE_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type TABLE_EXPORT/TABLE/STATISTICS/MARKER
Job "USERESTEST"."SYS_IMPORT_TABLE_01" successfully completed at Mon Dec 21 00:02:25 2020 elapsed 0 00:00:09

C:\Users\LetitiaMarin>

```

A SELECT query on the tables STUD1 and SOLV1 will display the imported masked data:

SQL> select \* from stud1;

ID	NUME	PRENUME	ANUL	SPE	GRUPA
164	A*****	A*****	5	Inf	500
264	A*****	T*****	5	Inf	500
364	T*****	A*****	5	Inf	500
460	C*****	A*****	5	Inf	500

SQL> select \* from rez1;

COD_STUDENT	COD_TEMA	DATA_UPLO	NOTA	DATA_CORE
164	1	04-NOV-12		
264	1	09-NOV-12		
164	2	19-NOV-12		
264	2	19-NOV-12		
364	2	24-NOV-12		
164	3	24-DEC-12		

6 rows selected.

## Bibliography:

<https://docs.oracle.com/en/database/oracle/oracle-database/12.2/sutil/oracle-data-pump.html#GUID-501A9908-BCC5-434C-8853-9A6096766B5A>

<http://www.oracle-base.com/articles/10g/oracle-data-pump-10g.php#GettingStarted>