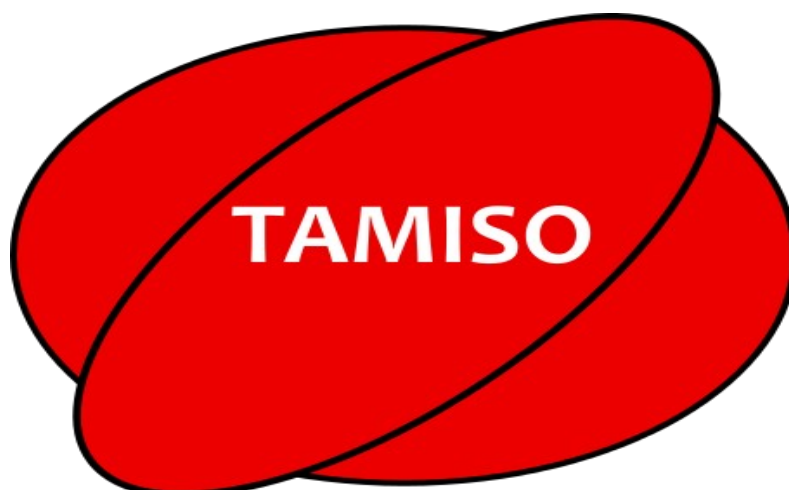


---

# “TAMISO”

## Overview



AUTHOR	Mauro A. Rossolato
First release	12.06.2014
Last revision	15.08.2020
confidentiality	internal
status	draft

## Table of Contents

1	WHAT IS.....	3
2	TAMISO/ver: WHY THIS ONE?.....	3
3	TAMISO/code: WHY THIS ONE?.....	3
4	WHO IS TAMISO USEFUL FOR?.....	3
5	ADVANTAGES.....	3
6	ACTORS.....	4
7	SCHEMAS.....	5
7.1	SCHEMA REPOSITORY.....	5
7.2	SCHEMA CLIENT.....	5
8	VERSIONING WORKFLOW.....	6
9	TMSID.....	7
10	VER: HOW IT WORKS?.....	7
10.1	MAIN TRIGGER.....	7
10.2	TRIGGER SCOPE.....	8
11	HISTORY CODE.....	9
12	ADVISOR WORKFLOW.....	10
13	CODE: HOW IT WORKS?.....	10
14	REPO INSTALLATION.....	11
14.1	SCRIPTS WORKSPACE.....	11
14.2	SCRIPTS CUSTOMIZATION.....	12
15	USE OF VERSIONING.....	13
15.1	SELLCAR'S PROJECT.....	13
15.2	NEW SCHEMA.....	13
15.3	MAGIC TRIGGER.....	13
15.4	SELLCAR'S OBJECTS.....	13
15.5	STAFF DEFINITION.....	14
15.6	PROJECT CREATION.....	14
15.7	ADD TO REPO.....	15
15.8	LOCKING OBJECTS.....	15
15.9	DV CHECKLIST.....	16
15.10	CODE IT AND COMPILE IT.....	16
16	USE OF CODE-ENHANCER.....	17
16.1	FILE SCRIPT.....	17
16.2	SQLID.....	18
17	SCHEMAS REMOVER.....	19
18	STRUCTURE.....	20
18.1	TABLES.....	20
18.2	TABLE'S TRIGGER.....	20
18.3	OBJECTS RELATIONSHIP.....	22
19	SNIPPETS.....	23
20	TAMISO/web.....	24
20.1	APEX CONFIGURATION.....	24
21	LIST OF FILES.....	25

22 ERROR CODES.....	26
23 QUESTIONS AND ANSWERS.....	27

## 1 WHAT IS

**TAMISO**, like a medal, includes two faces:

- versioning (also named **TAMISO/ver**)
- sql optimizer (also named **TAMISO/code**)

both features are designed for ORACLE® RDBMS.

And the third face? It's **TAMISO/web**.

## 2 TAMISO/ver: WHY THIS ONE?

There are several tools to maintain the script versioning: Git, SVN, BitBucket and so on.

What difference makes **TAMISO/ver** ?

Simple: **TAMISO/ver** is built-in in your database. Each DDL will be captured before deploy, automatically. No more: “...oops, forgotten, I'm sorry...”.

## 3 TAMISO/code: WHY THIS ONE?

A good application requires good performances. **TAMISO/code** helps the development team to achieve the best statements using an internal feature: **SQL Advisor**.

We can imagine **TAMISO/code** as task of **TAMISO/ver**: when your query is definitively refined you can put it into your code with a new version.

## 4 WHO IS TAMISO USEFUL FOR?

**TAMISO** is useful for large enterprise or small software factory.

Where is ORACLE®, TAMISO is near.

## 5 ADVANTAGES

Which advantage by **TAMISO** («ver» and «code» ) ?

Here is a short list:

- easy to use
- no additional command/language is required
- save the time (many task are automated)
- no losses – each DDL is captured
- scalable: suitable for other languages or scripts
- history fully detailed

## 6 ACTORS

In the software development, and maintenance, there are people (or roles) as

AD = ADministrator, project manager

AN = ANalyst

DV = DeVeloper

OP = OPerator

DA = DBA



In some circumstances a single person maintains more than a role, e.g. the AN and DV.

## 7 SCHEMAS

Both branch of **TAMISO** («ver» and «code») uses two kind of schema<sup>1</sup>:

- repository
- client

### 7.1 SCHEMA REPOSITORY

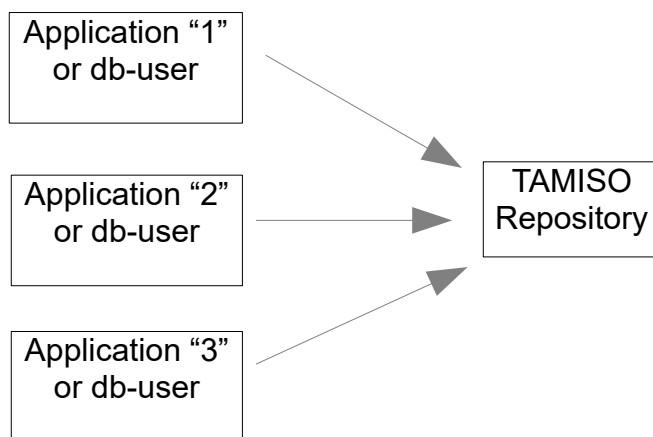
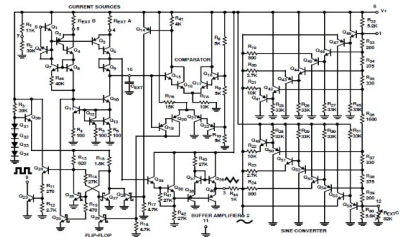
This schema (or db-user) contains the core tables of the **TAMISO** application.

The repository is used by one or more clients/application also named «schema client».

### 7.2 SCHEMA CLIENT

This schema contains the customer application.

The developer works using schema-client objects. On the others words, the DV scope is limited to the schema-client. No DV's action is required on repository.



<sup>1</sup> Also called «user», on Oracle environment.

---

## 8 VERSIONING WORKFLOW

The following steps assume that the installation already done (see chapter [REPO INSTALLATION](#)).

The entire versioning workflow may be ruled as:

ASSIGNEE	DESCRIPTION	SHORT
AD	The Administrator defines a project (PJ), resources and budget. Adds users and project into repo.	DEF
AN	The ANalyst defines the object, component. Also dispatch parts o entire PJ to DV and assign TMSID to DV. Each DV has unique TMSID for the same project	TAN
DV	The DeVeloper locks the objects (assigned) using the TMSID	LCK
OP/DV	OP (or DV) executes the script with TMSID	REL
DV	DV unlocks the objects	ULK
AD	AD closes the project (PJ)	CLO

## 9 TMSID

The versioning operation requires an relevant attribute: the **TMSID**.

The TMSID is the key attribute for **TAMISO/ver**.

An object locked for development can be modified only if TMSID is declared at *compile-time*.

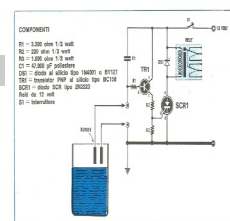
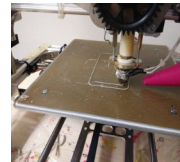
This value is passed throught the **session** to the procedure which stores the versions<sup>2</sup>.

The usage for **TMSID** is:

```
EXEC DBMS_APPLICATION_INFO.SET_CLIENT_INFO('YC001');
```

as you can see, the string “**YC001**” identifies the **TMSID**.

TMSID is the “passport” for objects release: without a correct TMSID the DDL will fails.



## 10 VER: HOW IT WORKS?

**TAMISO/ver** uses a special kind of trigger: DDL-schema.

This trigger intercept any DDL statement before the compilation task. This trigger is defined at schema level.

This trigger call a procedure named ENROLLER which executes several tasks:

- checks for valid TMSID
- checks if object locked (by TAMISO)
- the project is active
- the username of OP/DV is registered

After the validation, and before the recompilation , the previous version of the object 'll be saved into Mo8DDL SRC.

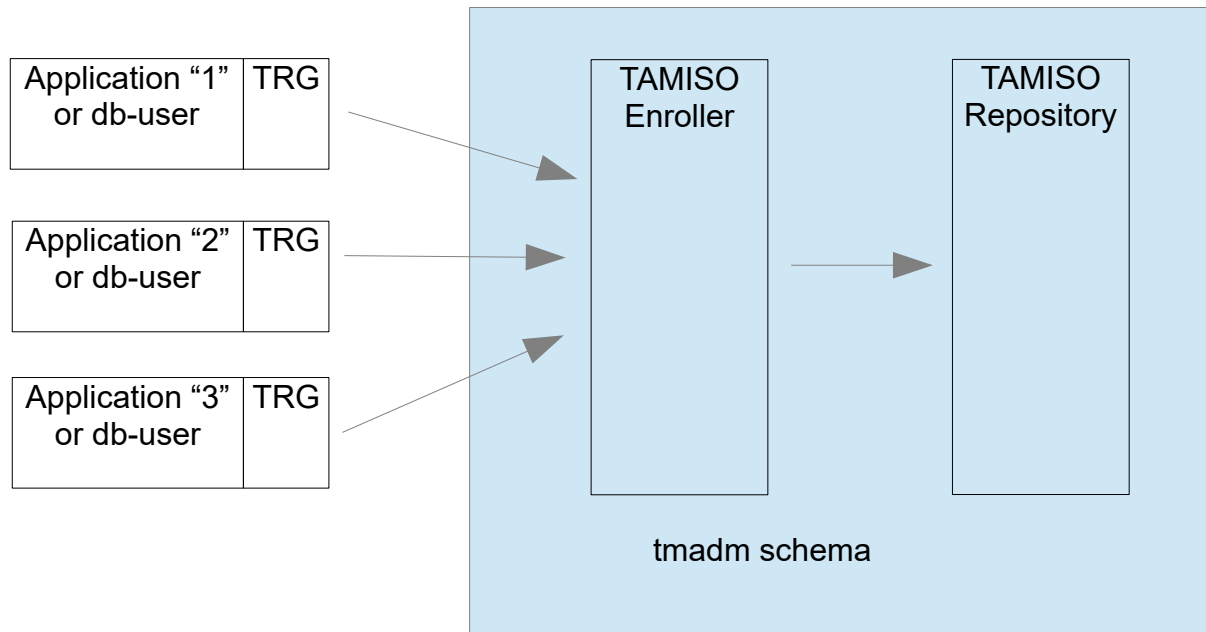
### 10.1 MAIN TRIGGER

As wrote before, the trigger Mo8BEFDDL\_TR and TMSID represents the core of **TAMISO/ver** application.

<sup>2</sup> TMSADM.ENROLLER procedure.

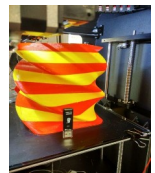
This trigger, owned by the schema-client, fires for each DDL on objects cataloged.

The automatic versioning is guaranteed by Mo8BEFDDL\_TR: **do not disable it!**



## 10.2 TRIGGER SCOPE

The Mo8BEFDDL\_TR is deployed into each schema-client and works at schema-level, however keep in mind that only categorized objects will be checked. In other words the table Mo8CATOBJ is used as “filter”. So, the Mo8BEFDDL\_TR operates using Mo8CATOBJ.





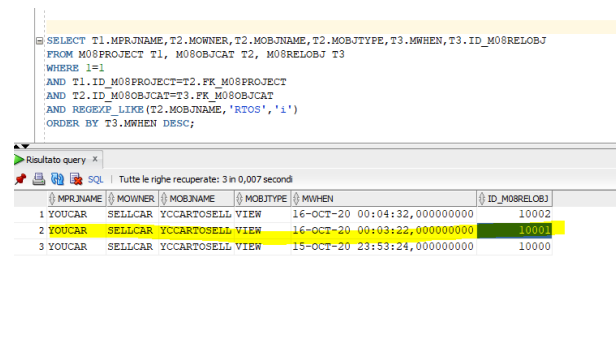
## 11 HISTORY CODE...

When you need to retrieve a previous version, the operation is very easy and quick: only two steps or query.

Before all, you have to get: OWNER, OBJECT\_NAME, OBJECT\_TYPE and VERSION.

This query may be useful:

```
SELECT T1.MPRJNAME, T2.MOWNER, T2.MOBJNAME, T2.MOBJTYPE, T3.MWHEN
FROM M08PROJECT T1, M08OBJCAT T2, M08RELOBJ T3
WHERE 1=1
AND T1.ID_M08PROJECT=T2.FK_M08PROJECT
AND T2.ID_M08OBJCAT=T3.FK_M08OBJCAT
AND REGEXP_LIKE(T2.MOBJNAME, 'RTOS', 'i')
ORDER BY T3.MWHEN DESC;
```



	MPRJNAME	MOWNER	MOBJNAME	MOBJTYPE	MWHEN	ID_M08RELOBJ
1	YOUCAR	SELLCAR	YCCARTOSELL VIEW		16-OCT-20 00:04:32,000000000	10002
2	YOUCAR	SELLCAR	YCCARTOSELL VIEW		16-OCT-20 00:03:22,000000000	10001
3	YOUCAR	SELLCAR	YCCARTOSELL VIEW		15-OCT-20 23:53:24,000000000	10000

and finally, to retrieve the version stored at 16-OCT-20 00:03:22 :

```
sqlplus /nolog
TMSADM/****
SET LINESIZE 500 PAGESIZE 9999
SELECT MTEXT
FROM M08DDL SRC
WHERE 1=1
AND FK_M08RELOBJ=10003
ORDER BY MLINE;
```

## 12 ADVISOR WORKFLOW

To get useful advice for your queries, follow the steps.

The following steps assume the installation already done (see chapter [REPO INSTALLATION](#)).

The entire versioning workflow may be ruled as:

ASSIGNEE	DESCRIPTION	SHORT
AD	The Administrator defines a project (PJ), resource and budget. Adds user and project into repo.	DEF
AN	The ANalyst defines the statements to analyze (may be from stored-code or session)	TAN
DV	The DeVeloper creates script-statement and save into directory	LCK
OP/DV	OP (or DV) executes the TAMISO/code	REL
DV	DV implements the enhancement and verifies. Possible call-back to TAMISO/ver.	ULK
AD	AD closes the project (PJ)	CLO

## 13 CODE: HOW IT WORKS?

**TAMISO/code** uses an ORACLE® package: the DBMS\_SQLTUNE.

A procedure, GETADVICETASK, reads, processes and finally saves the result (or advice) in to a clob field of Mo8COREADV table.

## 14 REPO INSTALLATION

The repository installation is very simple.

This task, named «DBA», will be performed by DA.

A new schema/user is required: TMSADM and, optionally, a specific tablespace.

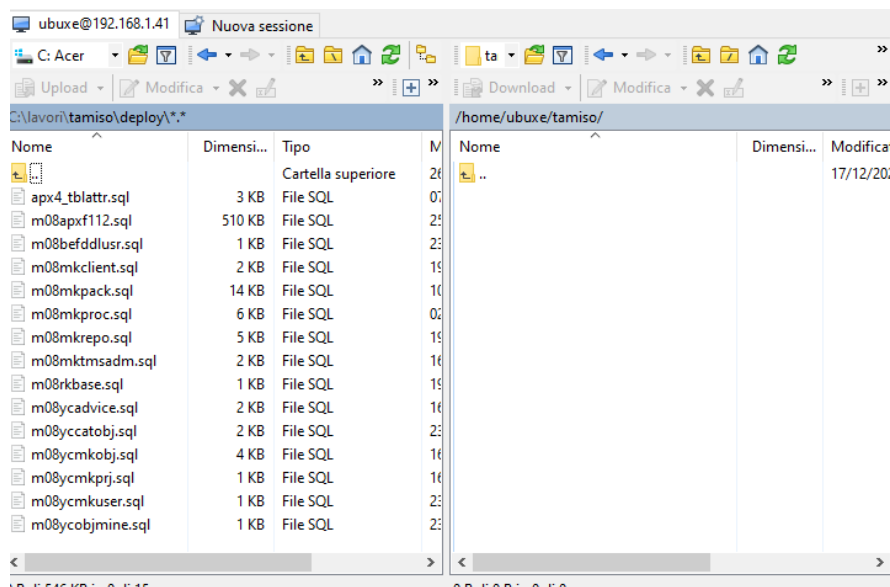


### 14.1 SCRIPTS WORKSPACE

A good practice is workspace for the scripts. Assuming a Unix-like server:

```
ubuxe@ubuxe:~$ cd
ubuxe@ubuxe:~$ mkdir tamiso
ubuxe@ubuxe:~$ cd tamiso
ubuxe@ubuxe:~/tamiso$
```

Using a sftp tool, copy all script into the just created folder:



when transferred:

```
ubuxe@ubuxe:~/tamiso$ ls -la
drwxrwxr-x  2 ubuxe ubuxe  4096 dic 17 15:01 .
drwxr-xr-x 14 ubuxe ubuxe  4096 dic 17 14:57 ..
-rw-rw-r--  1 ubuxe ubuxe  2545 ott  7 23:37 apx4_tblattr.sql
-rw-rw-r--  1 ubuxe ubuxe 521632 nov 25 14:15 m08apxf112.sql
-rw-rw-r--  1 ubuxe ubuxe   756 ott 23 19:02 m08befddlusr.sql
-rw-rw-r--  1 ubuxe ubuxe  1046 nov 19 08:07 m08mkclient.sql
-rw-rw-r--  1 ubuxe ubuxe 14160 nov 10 23:57 m08mkpack.sql
-rw-rw-r--  1 ubuxe ubuxe  5389 nov  2 17:22 m08mkproc.sql
-rw-rw-r--  1 ubuxe ubuxe  4315 nov 19 00:03 m08mkrepo.sql
-rw-rw-r--  1 ubuxe ubuxe  1147 nov 16 00:28 m08mktmsadm.sql
```

```

-rw-rw-r-- 1 ubuxe ubuxe      912 nov 19 15:03 m08rkbase.sql
-rw-rw-r-- 1 ubuxe ubuxe    1233 nov 16 00:29 m08ycadvice.sql
-rw-rw-r-- 1 ubuxe ubuxe    1543 ott 23 14:05 m08yccatobj.sql
-rw-rw-r-- 1 ubuxe ubuxe    3166 ott 16 21:59 m08ycmkobj.sql
-rw-rw-r-- 1 ubuxe ubuxe     650 ott 16 21:52 m08ycmkprj.sql
-rw-rw-r-- 1 ubuxe ubuxe     913 ott 23 13:59 m08ycmkuser.sql
-rw-rw-r-- 1 ubuxe ubuxe     575 ott 23 21:27 m08ycobjmine.sql
ubuxe@ubuxe:~/tamiso$

```

## 14.2 SCRIPTS CUSTOMIZATION

Before to launch the script, open, for editing the file **mo8mktmsadm.sql**, then change:

- the password and tablespace for the TMSADM user
- create or assign a new tablespace
- Oracle-Directory «KSTDIR»

Once modified, save, exit and run:

```

sqlplus /nolog
conn /as sysdba
@m08mktmsadm

```

With user TMSADM created, is now possible add his objects:

```

sqlplus /nolog
conn TMSADM/TMSADM123
@m08mkrepo
@m08mkpack
@m08mkproc

```

The TMSADM schema, also called «repo», will be used by all application's users. On other words the TMSADM schema supports all users on the same database.

## 15 USE OF VERSIONING

The best way to describe the **TAMISO/ver** functionality is by example: SELLCAR.

Obviously the complete repository installation was previously made.



### 15.1 SELLCAR'S PROJECT

The SELLCAR application implements some activities of cars reseller: the customer (reseller) want to keep trace of the requests for test-drive.

The main objects are:

- cars to sell (after test-drive)
- prospects
- logbook of test-drive
- a procedure to add a test-drive request

### 15.2 NEW SCHEMA

A new schema, or «schema-client», called SELLCAR 'll be created:

```
sqlplus /nolog
conn /as sysdba
@m08mkclient
```

now we are ready to deploy the schema's objects.

### 15.3 MAGIC TRIGGER

As first, we need to install the «magic trigger»:

```
sqlplus /nolog
conn SELLCAR/SELLCAR123
@m08befddlsr
```

### 15.4 SELLCAR'S OBJECTS

The application requires **few objects**: two tables, a view and a procedure.

```
sqlplus /nolog
```

```
SELLCAR/SELLCAR123
@m08ycmkobj
```

which executes<sup>3</sup>:

```
DROP TABLE YCCARS;

CREATE TABLE YCCARS (
    MPLATE      VARCHAR2(16),
    MPRICE      NUMBER,
    MKMDONE     NUMBER,
    MDTONSALE   TIMESTAMP,
    MDT SOLD    TIMESTAMP,
    MMODEL      VARCHAR2(30)
);

@apx4_tblattr.sql YCCARS APKTS
. . .
. . .
```

## 15.5 STAFF DEFINITION

The SELLCAR's staff includes few members:

```
sqlplus /nolog
SELLCAR/SELLCAR123
@m08ycmkuser
```

which executes:

```
SET SERVEROUTPUT ON SIZE 1000000
DECLARE
WRETCODE NUMBER;
WRETMMSG VARCHAR2(100);
BEGIN
TMSADM.TMSMAIN.MAKEUSER('mauro','cosaresta','main@mail.com',
'mauro rossolato','DV',WRETCODE,WRETMMSG);
DBMS_OUTPUT.PUT_LINE(TO_CHAR(WRETCODE)||' - '||WRETMMSG);

TMSADM.TMSMAIN.MAKEUSER('michael','derwolf','mich@mail.com',
'michael schenker','AD',WRETCODE,WRETMMSG);
DBMS_OUTPUT.PUT_LINE(TO_CHAR(WRETCODE)||' - '||WRETMMSG);

TMSADM.TMSMAIN.MAKEUSER('ulrich','willburnthesky','uli@mail.com',
'ulriche roth','DV',WRETCODE,WRETMMSG);
DBMS_OUTPUT.PUT_LINE(TO_CHAR(WRETCODE)||' - '||WRETMMSG);

END;
/
```

## 15.6 PROJECT CREATION

A project is a «container» for several activities: development, testing and so on. A project may

- 3 This script m08ycmkobj.sql includes several objects. The tool-file apx4\_tblattr.sql (developed for APEX) adds the audit few table's attributes and the related objects.

be associate to a «change»<sup>4</sup>.

```
sqlplus /nolog
SELLCAR/SELLCAR123
@m08ycmkprj
```

which executes:

```
SET SERVEROUTPUT ON SIZE 1000000
DECLARE
WRETCODE NUMBER;
WRETMMSG VARCHAR2(100);
BEGIN
TMSADM.TMSMAIN.MAKEPRJ('YOUCAR','ONLINE CARS SELLER',
'www.fablabway.com',TO_DATE('19.10.2020','DD.MM.YYYY'),
TO_DATE('30.11.2020','DD.MM.YYYY'),
'mauro', WRETCODE , WRETMMSG );
DBMS_OUTPUT.PUT_LINE(TO_CHAR(WRETCODE) || ' - ' || WRETMMSG);
END;
/
```

## 15.7 ADD TO REPO

This task perform the temporary association between db-object and project.

```
sqlplus /nolog
SELLCAR/SELLCAR123
@m08yccatobj
```

which executes<sup>5</sup>:

```
DECLARE
WRETCODE NUMBER;
WRETMMSG VARCHAR2(100);
BEGIN
TMSADM.TMSMAIN.CATOBJ('YCCARS','SELLCAR','TABLE' , 'YOUCAR' ,
'CARS MAIN LIST','YC001', WRETCODE , WRETMMSG );
DBMS_OUTPUT.PUT_LINE(TO_CHAR(WRETCODE) || ' - ' || WRETMMSG);
END;
/
```

The string «YC001» represent the «safety» code for the recompilation or TMSID (see [TMSID](#)).

## 15.8 LOCKING OBJECTS

To avoid conflict during the deploy, these object needs to be locked:

```
sqlplus /nolog
SELLCAR/SELLCAR123
@m08ycobjmine
```

which executes:

```
SET SERVEROUTPUT ON SIZE 1000000
```

<sup>4</sup> See ITIL documentation.

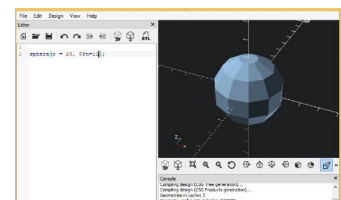
<sup>5</sup> The script file m08yccatobj.sql includes other statements.

```
DECLARE
WRETCODE NUMBER;
WRETMMSG VARCHAR2(100);
BEGIN
  TMSADM.TMSMAIN.OBJLOCK ( 'YCCARTOSELL','SELLCAR','VIEW' , 'mauro',
    'YOUCAR',WRETCODE, WRETMMSG);
  DBMS_OUTPUT.PUT_LINE (TO_CHAR(WRETCODE) || ' - ' || WRETMMSG);
END;
/
```

## 15.9 DV CHECKLIST

Before to start with code, the DV needs to verify some tasks:

- valid TMSID
- valid TAMISO's user
- unlocked objects



## 15.10 CODE IT AND COMPILE IT

The development activity starts here: DV can modify and recompile: each recompilation adds a «before image» to the repository.

A small sample of versioning:

```
sqlplus /nolog
SELLCAR/SELLCAR123
@m08ysomever
```



## 16 USE OF CODE-ENHANCER

**TAMISO/code**, which uses **SQL Advisor** of ORACLE® RDBMS, can receive the statements by two ways:

- file script
- session ID (SQLID)

### 16.1 FILE SCRIPT

The statement under investigation, must be saved as single file into an Oracle directory.

The next step requires SQLPlus\* to create (if doesn't exists yet) a new TAMISO's user:

```
DECLARE
  WRETCODE NUMBER;
  WRETMMSG VARCHAR2(100);
BEGIN
  TMSADM.TMSMAIN.MAKEUSER('MAURO','SECRET','mrossola@netscape.net',
    'MAURO ROX','ADMIN',WRETCODE,WRETMMSG);
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(WRETCODE) || ' - ' || WRETMMSG);
END;
/
```

and a new project

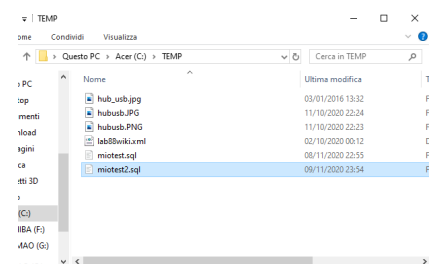
```
DECLARE
  WRETCODE NUMBER;
  WRETMMSG VARCHAR2(100);
BEGIN
  TMSMAIN.MAKEPRJ('prj2','prj2 descri','prj2 uri',
    sysdate-10,sysdate+30,'MAURO',WRETCODE,WRETMMSG);
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(WRETCODE) || ' - ' || WRETMMSG);
END;
/
```

A following procedure call, 'll load the script into a table and process it. Few attributes are required:

- target schema
- environment

Ask to DA for the path of Oracle-directory

Copy and paste the text and save it into Oracle-directory:

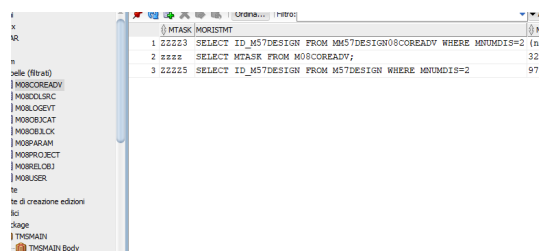


and load the script:

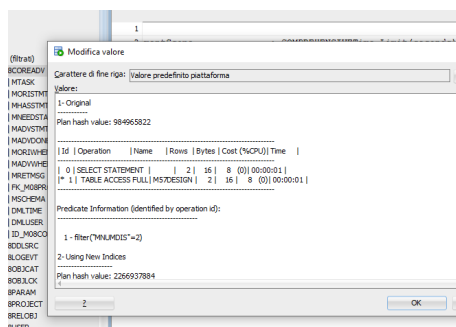
```
DECLARE
WRETCODE NUMBER;
WRETMMSG VARCHAR2(100);
BEGIN
    TMSMAIN.ADDSCRIPT('MAURO','aaa','prj2','zzzz',
        'miotest.sql','PRD',WRETCODE,WRETMMSG);
    DBMS_OUTPUT.PUT_LINE(TO_CHAR(WRETCODE) || ' - ' || WRETMMSG);
END;
/
```

and finally:

```
DECLARE
    WRETCODE NUMBER;
    WRETMMSG VARCHAR2(100);
BEGIN
    TMSMAIN.GETADVICETASK('MAURO','prj2','zzzz',
        'note...',WRETCODE,WRETMMSG);
    DBMS_OUTPUT.PUT_LINE(TO_CHAR(WRETCODE) || ' - ' || WRETMMSG);
END;
/
```



the result 'll be stored into Mo8COREADV.MADVSTMT:



## 16.2 SQLID

This alternative don't requires the Oracle directory access, because retrieve the statements

directly from session's catalog.

This way may be useful on production environment and represents the fastest way to gather the precious advice.

## 17 SCHEMAS REMOVER

To remove all TAMISO's objects and related owners, run these scripts:

- as SYSDBA – mo8mkclient\_rm.sql (for SELLCAR user, the sample)
- as SYSDBA – mo8mktmsadm\_rm.sql (for TMSADM user)

Keep in mind the Oracle Directory «KSTDIR», run as SYSDBA:

```
DROP DIRECTORY KSTDIR;
```

This is all.

## 18 STRUCTURE

The **TAMISO/ver** uses five tables, a package, a procedure and some trigger and sequence.

### 18.1 TABLES

TABLE	DESCRIPTION
Mo8PARAM	Contains user's parameters.
Mo8PROJECT	List of projects
Mo8OBJCAT	List of objects that belong to project
Mo8OBJLCK	Object locked due development activity. Any objects belongs to
Mo8LOGEVT	Application log
Mo8DDL SRC	Contains the objects's code.
Mo8RELOBJ	parent of Mo8DDL SRC
Mo8USER	Username for DV or OP.
Mo8COREADV	SQL Advisor support. Contains the statements and advices
Mo8STATUS	Status lookup

Each table includes three standard fields:

- DMLUSER for auditing
- DMLTIME for auditing
- ID\_Mo8[object] as Primary Key

The PK is defined by sequence.

### 18.2 TABLE'S TRIGGER

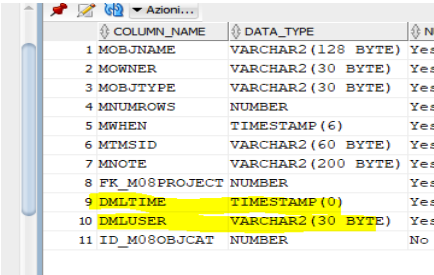
Each tables has a couple of triggers,

- sequence for Primary Key
- audit (updater and timestamp)

Here is the complete list:

M08RELOBJ_PK	BEFORE EACH ROW	M08RELOBJ
M08RELOBJ_AUD	BEFORE EACH ROW	M08RELOBJ
M08PROJECT_PK	BEFORE EACH ROW	M08PROJECT
M08PROJECT_AUD	BEFORE EACH ROW	M08PROJECT

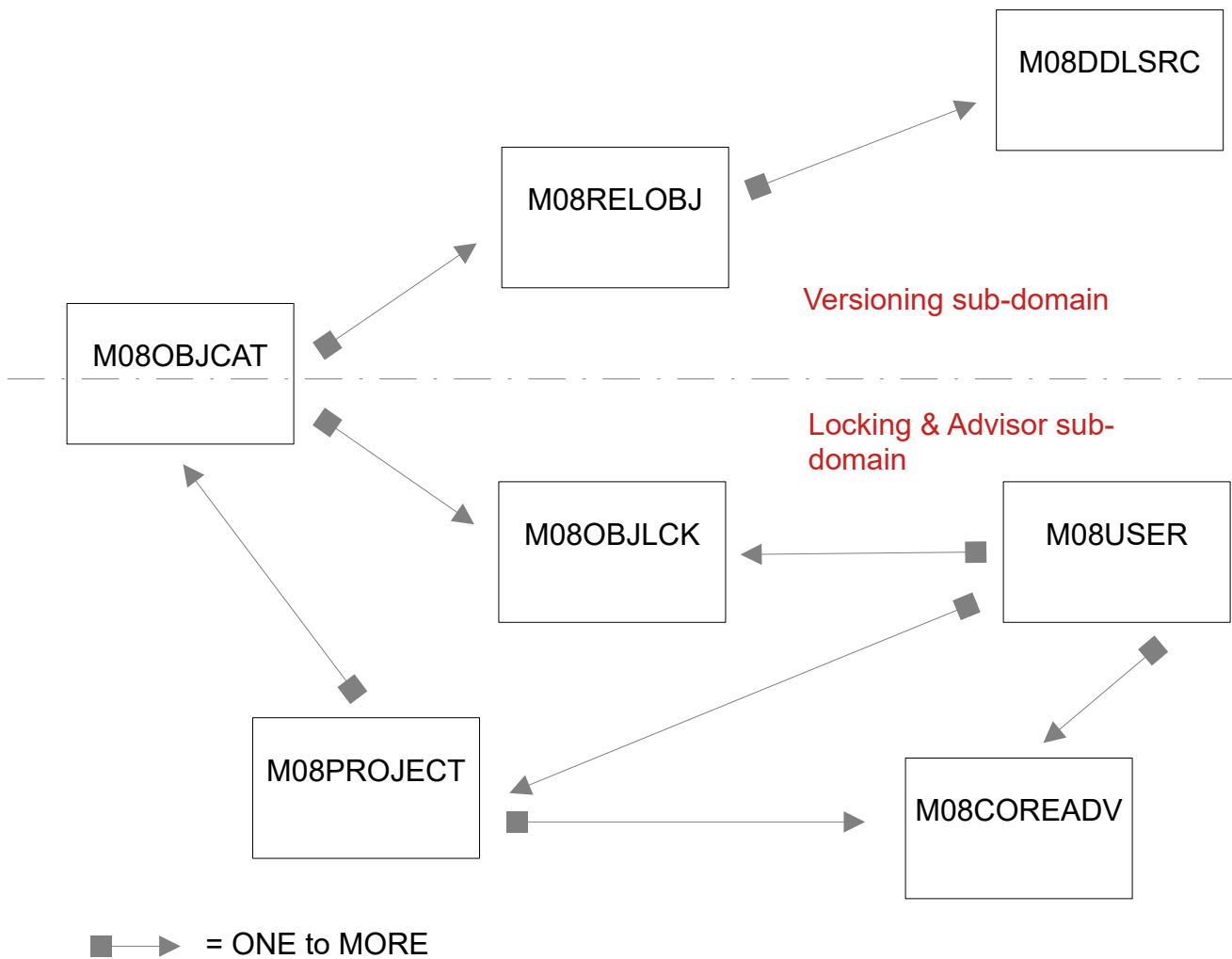
M08OBJLCK_PK	BEFORE	EACH	ROW	M08OBJLCK
M08OBJLCK_AUD	BEFORE	EACH	ROW	M08OBJLCK
M08USER_PK	BEFORE	EACH	ROW	M08USER
M08USER_AUD	BEFORE	EACH	ROW	M08USER
M08DDL SRC_PK	BEFORE	EACH	ROW	M08DDL SRC
M08DDL SRC_AUD	BEFORE	EACH	ROW	M08DDL SRC
M08OBJCAT_PK	BEFORE	EACH	ROW	M08OBJCAT
M08OBJCAT_AUD	BEFORE	EACH	ROW	M08OBJCAT
M08LOGEVT_AUD	BEFORE	EACH	ROW	M08LOGEVT
M08PARAM_PK	BEFORE	EACH	ROW	M08PARAM
M08PARAM_AUD	BEFORE	EACH	ROW	M08PARAM
M08COREADV_PK	BEFORE	EACH	ROW	M08COREADV
M08COREADV_AUD	BEFORE	EACH	ROW	M08COREADV



COLUMN_NAME	DATA_TYPE	NUM
1 M0BJNAME	VARCHAR2 (128 BYTE)	Yes
2 M0WNER	VARCHAR2 (30 BYTE)	Yes
3 M0BJTYPE	VARCHAR2 (30 BYTE)	Yes
4 MNUMROWS	NUMBER	Yes
5 MWHEN	TIMESTAMP (6)	Yes
6 MTMSID	VARCHAR2 (60 BYTE)	Yes
7 MNOTE	VARCHAR2 (200 BYTE)	Yes
8 FK_M08PROJECT	NUMBER	Yes
9 DMLTIME	TIMESTAMP (0)	Yes
10 DMLUSER	VARCHAR2 (30 BYTE)	Yes
11 ID_M08OBJCAT	NUMBER	No

## 18.3 OBJECTS RELATIONSHIP

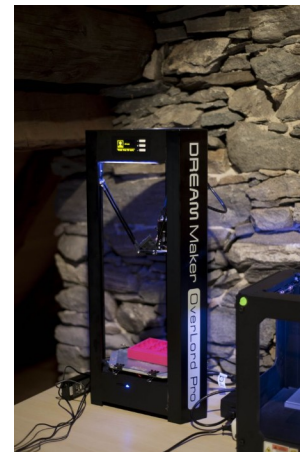
The **TAMISO** entities are related as below:



## 19 SNIPPETS


For each objects of the given (like) project, how many versions?

```
SELECT T1.MPRJNAME, T2.MOWNER, T2.MOBJNAME, T2.MOBJTYPE, T3.WHOW  
FROM M08PROJECT T1, M08OBJCAT T2,  
(SELECT COUNT(*) WHOW, FK_M08OBJCAT FROM M08RELOBJ GROUP BY FK_M08OBJCAT) T3  
WHERE 1=1  
AND T1.ID_M08PROJECT=T2.FK_M08PROJECT  
AND T2.ID_M08OBJCAT=T3.FK_M08OBJCAT  
AND REGEXP_LIKE (T1.MPRJNAME, 'YOUUC', 'i');
```



## 20 TAMISO/web

To simplify the data access, a light tool is available for ORACLE® APEX:



SQL\_ADV

SOURCE

PROJECTS

OBJECTS

LOCKED

VERSIONS

USERS

APPL.LOG

M08objck

M08OBJLCK

Search

Display

15

Go

Fk M08objcat	Fk M08user	Fk M08project	Mdateunlock	Mdateunlock	Midchgdoc	Dmltime	Dmluser
10000	10000	10000	18-NOV-20 12.04.08.000000 AM			18-NOV-20 12.04.08.000000 AM	MYAPEX
10000	10000	10000	17-NOV-20 11.31.00.000000 PM			17-NOV-20 11.55.41.000000 PM	TMSADM
10020	10000	10040	19-NOV-20 12.46.05.000000 AM			19-NOV-20 12.46.05.000000 AM	SELLCAR
10021	10000	10040	19-NOV-20 02.41.57.000000 PM			19-NOV-20 02.41.57.000000 PM	SELLCAR
			18-NOV-20 12.37.11.000000 AM			18-NOV-20 12.37.11.000000 AM	SELLCAR
			18-NOV-20 12.54.51.000000 AM			18-NOV-20 12.54.51.000000 AM	TMSADM

[Spread Sheet](#)

1 - 6

release 1.0

### 20.1 APEX CONFIGURATION

[es könnte das nächste Mal sein]



## 21 LIST OF FILES

The kit **TAMISO** («**ver**» and «**code**») contains the following files script:

FILE NAME	DESCRIPTION	SCHEMA
mo8befddlusr.sql	trigger	client
mo8yccatobj.sql	Add to catalog	client
mo8ycmkobj.sql	Add some object under development	client
mo8ycmkprj.sql	Add new project	client
mo8ycmkuser.sql	Add user or developer	client
mo8ycobjmine.sql	Lock user objects	client
mo8ysomever.sql	Versioning simulation for view	client
mo8mkclient.sql	Created schema client SELLCAR	sys
mo8mktmsadm.sql	Creates schema repo	sys
mo8mkclient_rm.sql	Removes schema client SELLCAR	sys
mo8mktmsadm_rm.sql	Removes schema repo	sys
mo8mkrepo.sql	Creates tables, sequences, ecc	tmsadm
mo8mkproc.sql	Creates procedure	tmsadm
mo8mkpack.sql	Creates package	tmsadm
mo8rkbase.sql	Adds some records	tmsadm
mo8ycadvice.sql	Advisor samples	client
mo8apxf112.sql	TAMISO/web - APEX® export	APEX®

## 22 ERROR CODES

Sometimes goes wrong...

CODE	DESCRIPTION	ACTION
30000	«TMSERR: project already defined».	Resubmit with another and new prjname.
30001	«TMSERR: end-time lower than start time»	Check for forecast end-time of project.
30002	N.D.	
30003	N.D.	
30004	N.D.	
30005	N.D.	
30006	«TMSERR: user already defined»	Resubmit with another and new username
30007	N.D.	
30008	«TMSERR: project NOT active or not found!»	The project's status isn't «active»
30009	N.D.	
30010	«TMSERR: object already assigned to prj!»	Close the old project
30011	«TMSERR: object already UNlocked!»	None
30012	«TMSERR: user not found»	Check or add a new user using MAKEUSER procedure.
30013	«TMSERR: object already locked!»	Skip over, the lock task is already done.
30014	«TMSERR: user NOT found!»	
30015	Directory not defined	Check parameter in Mo8PARAM
30016	N.D.	
30017	N.D.	
30018	«TMSERR: INVALID tmsid!»	The TMSID may be incorrect or expired
30019	N.D.	
30020	N.D.	
30021	N.D.	
30022	«TMSERR: unable to LOCK obj!»	Nothing (object) to release
39999	TMSERR: NOT defined error!	

## 23 QUESTIONS AND ANSWERS

Q.1) How long time take the installation?

A.1) 10 minutes

Q.2) The automatic versioning requires the object «locked»?

A.2) Yes, before any modification, the object needs to be locked.

Q.3) Is TMSID mandatory?

A.3) Yes, TMSID works as «key» to Oracle's catalog update (as compiling).

## Indice alfabetico

association.....	15
conn SELLCAR.....	13
DBMS_SQLTUNE.....	10
DMLUSER.....	20
DV scope.....	5
installation.....	11, 27
M08BEFDDL_TR.....	7 e seg.
magic trigger.....	13
MAGIC TRIGGER.....	13
project manager.....	4
repository.....	5, 11
SELLCAR.....	13 e segg., 25
SQL Advisor.....	3, 17
TMSADM.....	11 e seg.
tmsid.....	26
TMSID.....	6 e seg., 15 e seg., 26