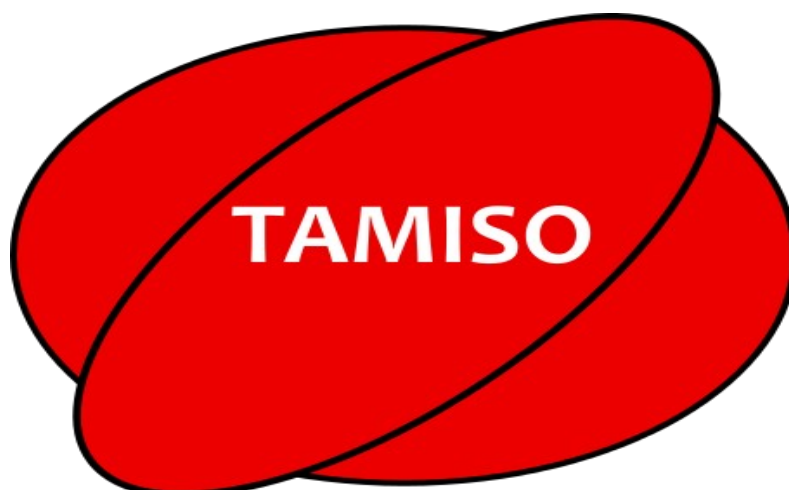


“TAMISO” Overview



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

Indice generale

1	OBJECTIVE.....	4
2	WHAT IS.....	4
3	WHY THIS ONE?.....	4
4	WHO IS TAMISO USEFUL FOR?.....	4
5	ADVANTAGES.....	4
6	ACTORS.....	5
7	HOW IT WORKS?.....	5
8	SCHEMAS.....	6
8.1	SCHEMA REPOSITORY.....	6
8.2	SCHEMA CLIENT.....	6
9	MAIN TRIGGER.....	7
9.1	TRIGGER SCOPE.....	7
10	WORKFLOW.....	8
10.1	DV CHECKLIST.....	8
11	REPO INSTALLATION.....	9
12	AN EXAMPLE.....	10
13	CLIENT INSTALLATION.....	11
13.1	WORKFLOW PATHWAY.....	11
14	SCHEMA CLIENT CONFIGURATION.....	12
14.1	NAMING CONVENTION.....	12
14.2	CLIENT OBJECTS.....	12
14.3	POST DEPLOY.....	13
14.4	ADD TO REPO.....	13
14.5	WORKFLOW PATHWAY.....	13
15	TMSID.....	14
15.1	WORKFLOW PATHWAY.....	14
16	STRUCTURE.....	15
16.1	TABLES.....	15
17	OBJECTS RELATIONSHIP.....	16
17.1	TABLE'S TRIGGER.....	17
18	HOW TO RECOVER.....	18
19	SNIPPET.....	19

Changelog

17.02.2018 – mr – added “workflow”

1 OBJECTIVE

All about **TAMISO**: description and technical reference.

2 WHAT IS

TAMISO is a versioning platform for Oracle.



3 WHY THIS ONE?

There are several tools for script versioning: Git, SVN and so on.

What difference makes **TAMISO** ?

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

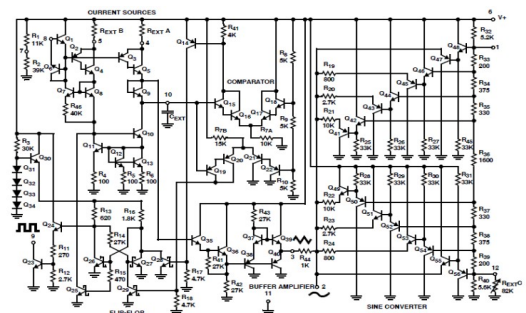
4 WHO IS TAMISO USEFUL FOR?

TAMISO is useful for database developers.

5 ADVANTAGES

TAMISO is:

- easy to use
- no additional command/language is required
- save the time (automatic)
- no losses – each DDL is captured
- scalable: suitable for other languages or scripts



6 ACTORS

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

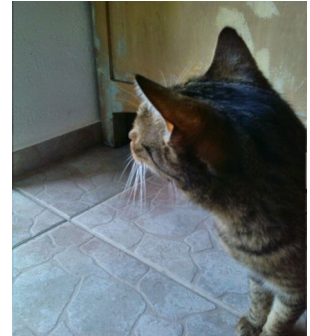
AD = ADministrator, project manager

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 HOW IT WORKS?

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

This trigger intercept any DDL statement before the compilation task.

If the involved object is registered into the «catalog» then will be saved before.

The catalog is stored into a separate schema.

8 SCHEMAS

TAMISO uses two kind of schema¹:

- repository
- client

8.1 SCHEMA REPOSITORY

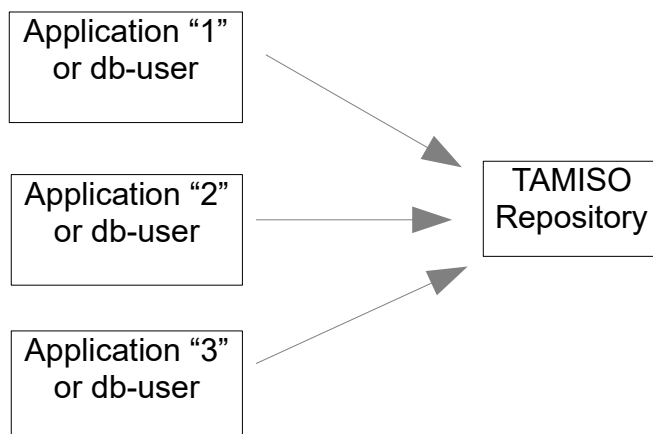
This schema (or db-user) contains the core tables of the **TAMISO** application.

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

8.2 SCHEMA CLIENT

Is the schema managed, which contains the customer application.

The developer works with schema-client objects. On the others words, the DV scope is limited to the schema-client.



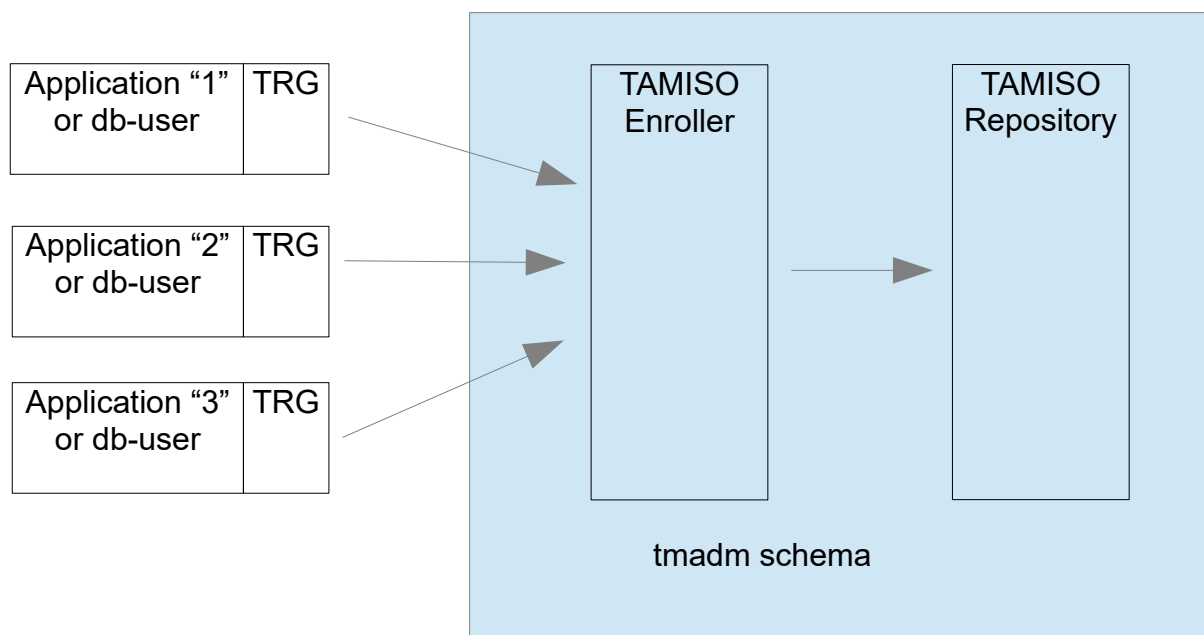
¹ Also called "user" , on Oracle environment.

9 MAIN TRIGGER

As wrote febore, the trigger Mo8BEFDDL_TR and TMSID represent the core of **TAMISO** application.

This trigger, owned by the schema-client, fires for each DDL (defined) on schema-client.

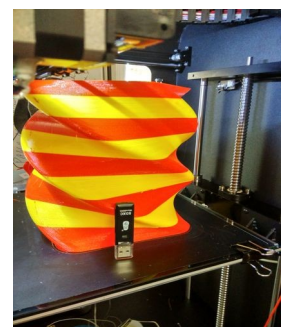
The automatic versioning is guaranteed by Mo8BEFDDL_TR: do not disable it!



9.1 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. On the others words the table Mo8CATOBJ is used as “filter”.

So, the Mo8BEFDDL_TR can “see” using Mo8CATOBJ.



10 WORKFLOW

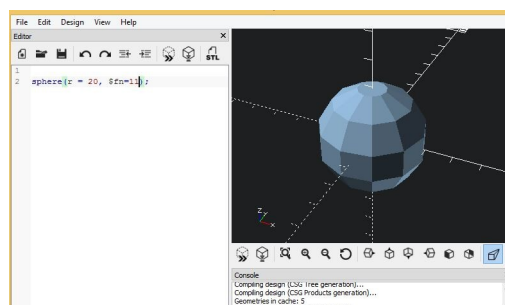
The “workflow” contains the following tasks:

ASSIGNEE	DESCRIPTION	SHORT
AD	The Administrator defines a project (PJ). Resource, budget. Adds user 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
DA	Creates (if new installation) the TSMADM and the application schema	DBA
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

10.1 DV CHECKLIST

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

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



11 REPO INSTALLATION

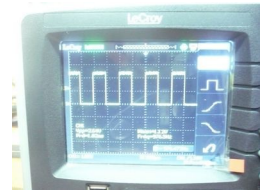
The repository installation is very simple.

This task will be performed by DA.

A new schema/user is required: TMSADM.

Before to launch the script, open, for editing, mo8mktmsadm.sql.

Change the password and tablespaces for the TMSADM user. Then save and exit.



```
sqlplus /nolog
conn /as sysdba
@m08mktmsadm
```

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

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

The TMSADM schema, also called «repo», will be used by all application user that belongs the same database.

12 AN EXAMPLE

We can start with a new application.

The AD asks to DA for a new schema called “SELLCAR”

In this show-case we assume that the repository has been already installed.



13 CLIENT INSTALLATION

The client schema requires

- user (or owner)
- grant on TMSADM (some objects)
- trigger

To create the new application user SELLCAR:

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

as last step, DA creates the schema-DDL_trigger:

```
sqlplus /nolog
SELFCAR/SELFCAR123
@m08befddlusr
```

The DA activities ends here.

Starting from here the developer will create the required objects.



13.1 WORKFLOW PATHWAY

This unit work is defined as 3th step:

DEF-TAN-DBA

14 SCHEMA CLIENT CONFIGURATION

Before all, the AD have to create the “peoples of the project”:

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

with these users (DV and AN) the AD can create the new project named “YOUCAR”:

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

From here starts the DeVeloper's scenario.

First of all, DV defines the objects:

- a table for cars to sell
- a table with prospects
- a table with association cars and prospects²
- a view with cars not yet sold
- a procedure to add a new test-drive



14.1 NAMING CONVENTION

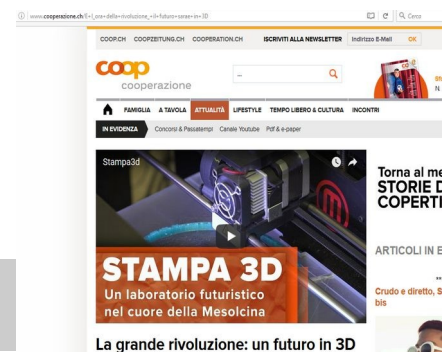
A good practice, should be to define the object name before to create them. To make so, the naming structure become:

mo8 = as domain

yc = short name for YOUCAR

14.2 CLIENT OBJECTS

```
sqlplus /nolog
conn SELLCAR/SELLCAR123
@m08ycmkobj
```



² E.g. When a customer asks for a test drive.

14.3 POST DEPLOY

Obviously a simple test:

```
DECLARE
WRETCODE NUMBER;
WRETMMSG VARCHAR2(100);
BEGIN
YCADDTEST ('CT215FZ',SYSDATE+3,'MAURO',
'377-2135141',WRETCODE, WRETMMSG);
DBMS_OUTPUT.PUT_LINE(TO_CHAR(WRETCODE) || ' - ' || WRETMMSG);
END;

SELECT * FROM YCCARTOSELL;
```

14.4 ADD TO REPO

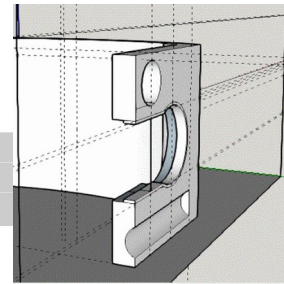
At this point **TAMISO** is ready to add the objects into the catalog³: in this manner **TAMISO** 'll be able to intercept any DDL on these objects.

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

The schema-client-setup is over. From now each DDL on schema⁴ will fire the trigger to preserve the previous statement.

And finally, lock the objects:

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



Starting from now, any DDL on registered (catalog) and locked objects will be recorded into Mo8DDL SRC and Mo8RELOBJ.

14.5 WORKFLOW PATHWAY

This unit work is defined as 4th step:

DEF-TAN-DBA-LCK

³ The catalog resides into schema-server.

⁴ Limited to objects added into repository.

15 TMSID

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

An object registered as *work-in-progress*⁵ can be modified only if TMSID is declared at *compile-time*.

This value is passed through the session to the procedure which stores the versions⁶.

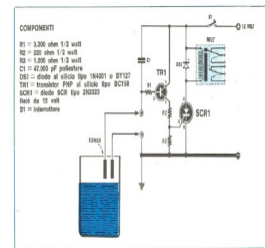
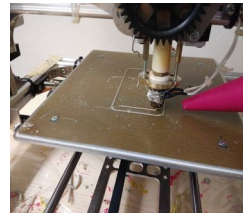
The usage for TMSID is:

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

as you can see, the string “YC001” is issued before invoking the procedure

TMSADM.TMSMAIN.CATOBJ. Is an arbitrary string long 60 chars.

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



For example, when we update the object:

```
CREATE OR REPLACE VIEW YCCARTOSELL AS
SELECT T1.MMODEL,T1.MPRICE*1.184,T1.MDTONSALE,T3.WDRIVE
FROM YCCARS T1,
(SELECT FK_YCCARS,COUNT(*) WDRIVE FROM YCCARPRO WHERE 1=1 GROUP BY FK_YCCARS) T3
WHERE 1=1
AND T1.ID_YCCARS=T3.FK_YCCARS
AND T1.MDTSOLD IS NULL;
```

the previous statement will be added into Mo8DDL SRC.

15.1 WORKFLOW PATHWAY

This unit work is defined as 5th step:

DEF-TAN-DBA-LCK-REL



⁵ Or locked.

⁶ TMSADM.ENROLLER procedure.

16 STRUCTURE

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

16.1 TABLES

Mo8PARAM

Contains user's parameters.

At 1.0 release this feature isn't yet available.

Mo8PROJECT

List of projects

Mo8OBJCAT

List of objects that belong to project

Mo8OBJLCK

Object locked due development activity. Any objects belongs to Mo8OBJCAT.

Mo8DDL SRC

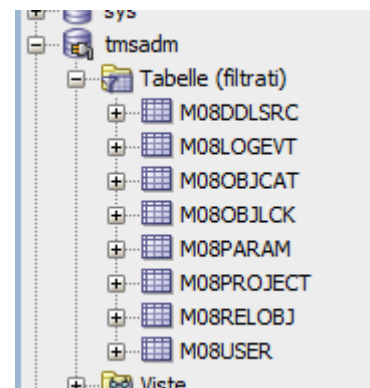
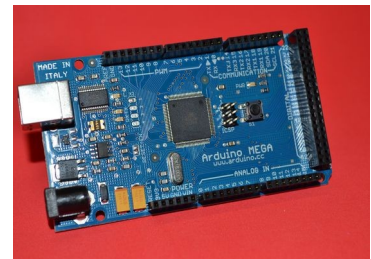
Contains the objects's code.

Mo8RELOBJ

parent of Mo8DDL SRC

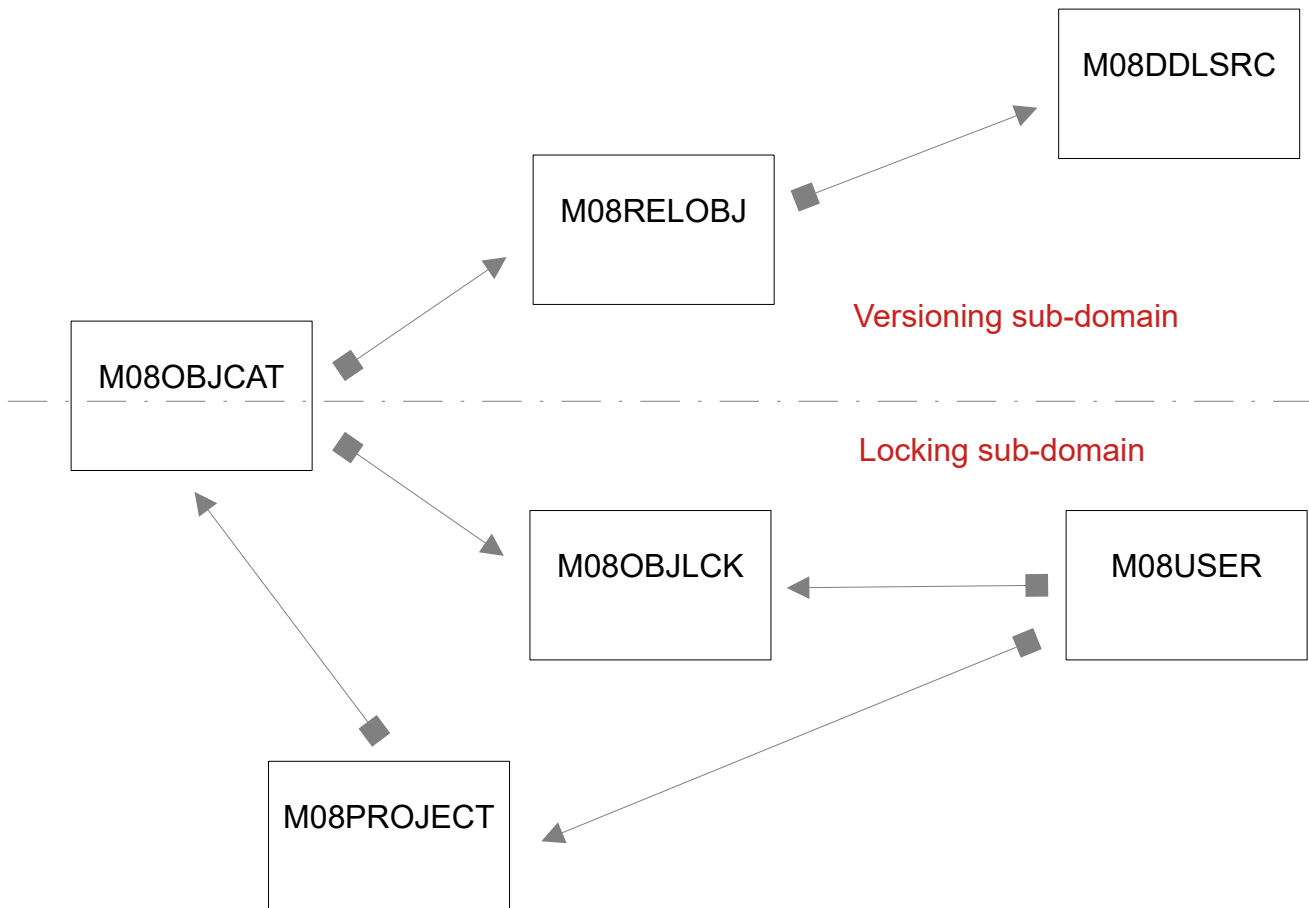
Mo8USER

Username for DV or OP.



17 OBJECTS RELATIONSHIP

At the first glance, the **TAMISO** objects are related:



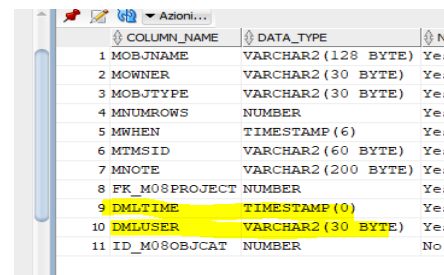
17.1 TABLE'S TRIGGER

Each tables has a couple of triggers,

- sequence
- 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



	COLUMN_NAME	DATA_TYPE	NI
1	MOBJNAME	VARCHAR2 (128 BYTE)	Yes
2	MOOWNER	VARCHAR2 (30 BYTE)	Yes
3	MOBJTYPE	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

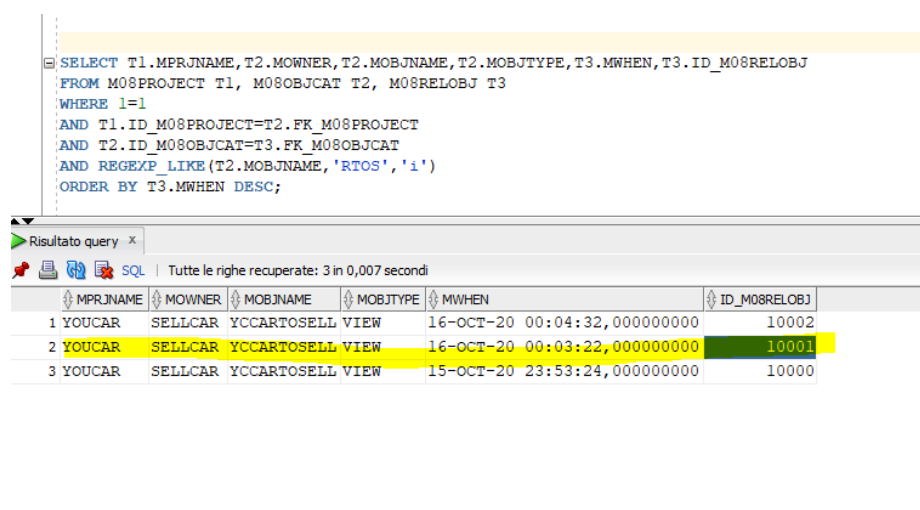
18 HOW TO RECOVER

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 know: OWNER, OBJECT_NAME and OBJECT_TYPE.

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, the second step, 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;
```

19 SNIPPET

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, 'YOUCL', 'i');
```

