

New York Oracle Users Group (NYOUG) Long Island SIG

Oracle Data Integrator (ODI) Best Practices...
Do You Know How Flexible ODI Is?



igorhan@gmail.com

 <http://gurcanorhan.wordpress.com>

 http://www.twitter.com/gurcan_orhan

<http://tr.linkedin.com/in/gurcanorhan>

Gürcan Orhan
Software Architect & Expert Developer



WHO AM I ?

- +16 years of IT experience.
- +9 years of DWH experience.
- +5 years of Oracle Data Integrator experience,
- +4 years of Oracle Warehouse Builder experience.

Cognos, Microstrategy, Business Objects, OBIEE
Sybase Power Designer, CA ERwin Data Modeler

Joined Turkcell October 2008, Turkcell Technology March 2010.

Oracle Excellence Awards - Technologist of the Year 2011 :
Enterprise Architect
(Oracle Magazine Editors' Choice of Awards, Enterprise Architect)

DWH & BI Chair : **TROUG** (Turkish Oracle User Group)

Published Customer Snapshot for NODI @Oracle.com

Published video about ODI @Oracle.com (Oracle Media Network)

Presenter in Oracle Open World since 2010 (hat-trick)



ORACLE
ACE Director



Gürcan Orhan **ENTERPRISE ARCHITECT**

Architect troubleshoots network alarms with customized Oracle Data Integrator solution.

For Gürcan Orhan, software architect and senior developer at Turkcell Technology—a firm that develops applications and building infrastructure for Turkcell, the leading communications and technology company in Turkey and the third-largest mobile phone operator in Europe—success comes down to quickly diagnosing critical problems across a diverse and heterogeneous network infrastructure.

"We wanted to be able to quickly detect network alarm locations, which is difficult because alarms can be originated from more than 200,000 network nodes and produced from 50 different source systems, and their location is hidden in two unstructured columns," says Orhan, winner of the Oracle Excellence Award for Technologist of the Year: Enterprise Architect. "To overcome these challenges, we needed a robust, solid

software solution that leverages an ELT [extract, load, and transform] approach." In 2008, Turkcell Technology created a team to compare different data integration products and ultimately selected Oracle Data Integrator.

"We selected Oracle Data Integrator because its architecture leveraged the database engine power and provided best-in-class performance and scalability," says Orhan. "It eliminated the use of a middle-tier transformation server and thereby avoided additional hardware investment."

Orhan's integration solutions feed into a multiterabyte Oracle Real Application Clusters data warehouse called NODI (Network Operations Data Infrastructure), which is running on Oracle Database 11g Release 2. NODI is unique because it has an unprecedented model design and is flexible, easy to maintain and implement, and modular.

Some of the data flowing into NODI is unstructured data from network monitors and alarms, so Orhan's team had to build algorithms to analyze it and successfully identify the exact location of more than 100,000 alarms each day. "We're currently correctly identifying the five unstructured dimensions, such as reason, source system, vendor, and more, for the alarms 100 percent of the time, and the location of the alarms more than 99.99 percent of the time," says Orhan.

In the end, Turkcell's bet on Oracle Infrastructure is paying off, especially when it comes to creating flexible solutions that will continue to address any future need. "Oracle Data Integrator is a tool that can talk, or learn how to talk, with any database or operating system in its own language," says Orhan. "That's the power of Oracle Data Integrator."

JANUARY/FEBRUARY 2011 ORACLE.COM/ORACLEMAGAZINE

My definition of ODI

ODI (Oracle Data Integrator) is a tool,
that can talk,
or learn how to talk,
with any database system,
or any operating system,
in its own language.

This is the power of ODI.

ATTENTION...!!!

Make sure you have;

- ❖ Backup your repository
- ❖ Backup your Knowledge Modules (export, duplicate)
- ❖ Backup your necessary ODI development

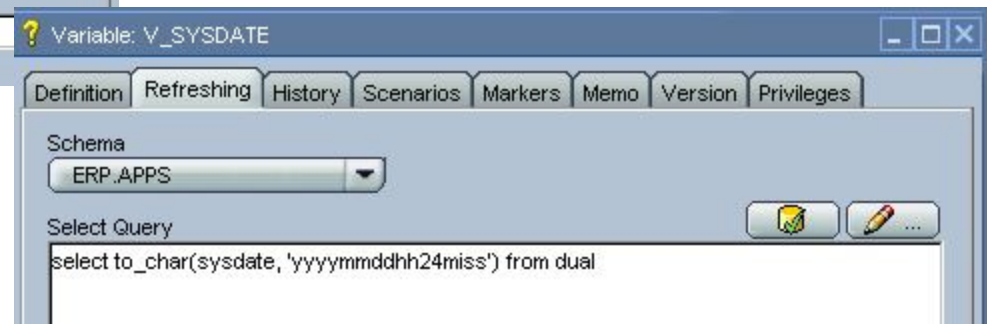
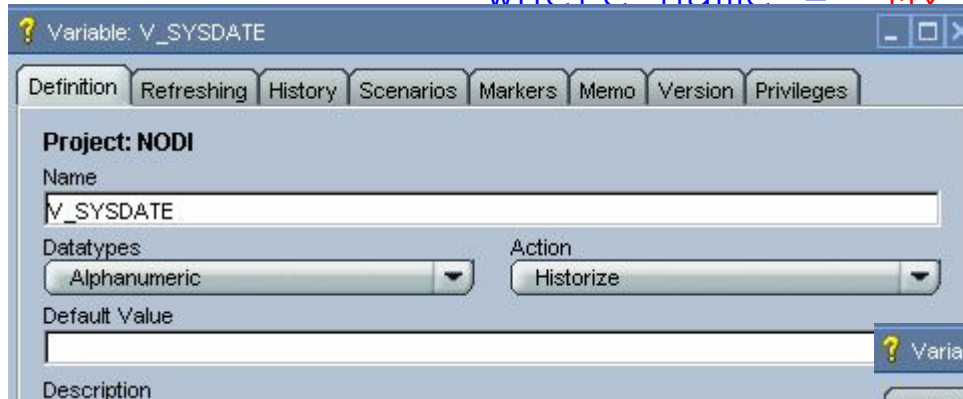
Before trying something in your environment

Remember to create a zzz_Test folder and test before apply



Variables... Use a variable within a variable

```
select count(1) from msdb.dbo.sysjobhistory
where step_id      = 0
      and run_status = 1
      and job_id = (select job_id from msdb.dbo.sysjobs
                    where name = 'Mv BI_Job')
                    #V_SYSDATE',1,8) as integer)
```



<http://gurcanorhan.wordpress.com/2010/12/14/odi-variables/>

Variables... Use a odiRef function within a variable

```
SELECT *
  FROM odiwd.snp_session sess,
       odiwd.snp_step_log step,
       odiwd.snp_sess_task sess_task,
       odiwd.snp_exp_txt exp_txt
 WHERE sess.sess_no = step.sess_no
       AND sess.sess_no = sess_task.sess_no
       AND step.nno = sess_task.nno
       AND step.i_txt_step_mess = exp_txt.i_txt
       AND step.step_status = 'E'
       AND exp_txt.txt_ord = 0
       AND sess.sess_no = <%=odiRef.getSession("SESS_NO")%>
```

```
SELECT NVL(MAX(ALARM_ID), 0) FROM
<%=odiRef.getSchemaName("MYDB.DWH", "D")%>.TABLE_NAME
```

```
SELECT NVL(MAX(ALARM_ID), 0) FROM DWH.TABLE_NAME
```

<http://gurcanorhan.wordpress.com/2010/12/14/odi-variables/>

HINTS in Oracle

Oracle's most powerful querying attribute when in right hands.
An Oracle hint is an optimizer directive that is embedded into an SQL statement to suggest to Oracle how the statement should be executed.

Most common hints in DWH system;

- ❖ APPEND
- ❖ PARALLEL
- ❖ USE_HASH
- ❖ USE_MERGE
- ❖ FULL
- ❖ INDEX
- ❖ ORDERED
- ❖ MERGE



<http://psoug.org/reference/hints.html> or just google «Oracle hints»

KM's... How to apply static HINTS.

The screenshot shows a database management tool interface with the following settings:

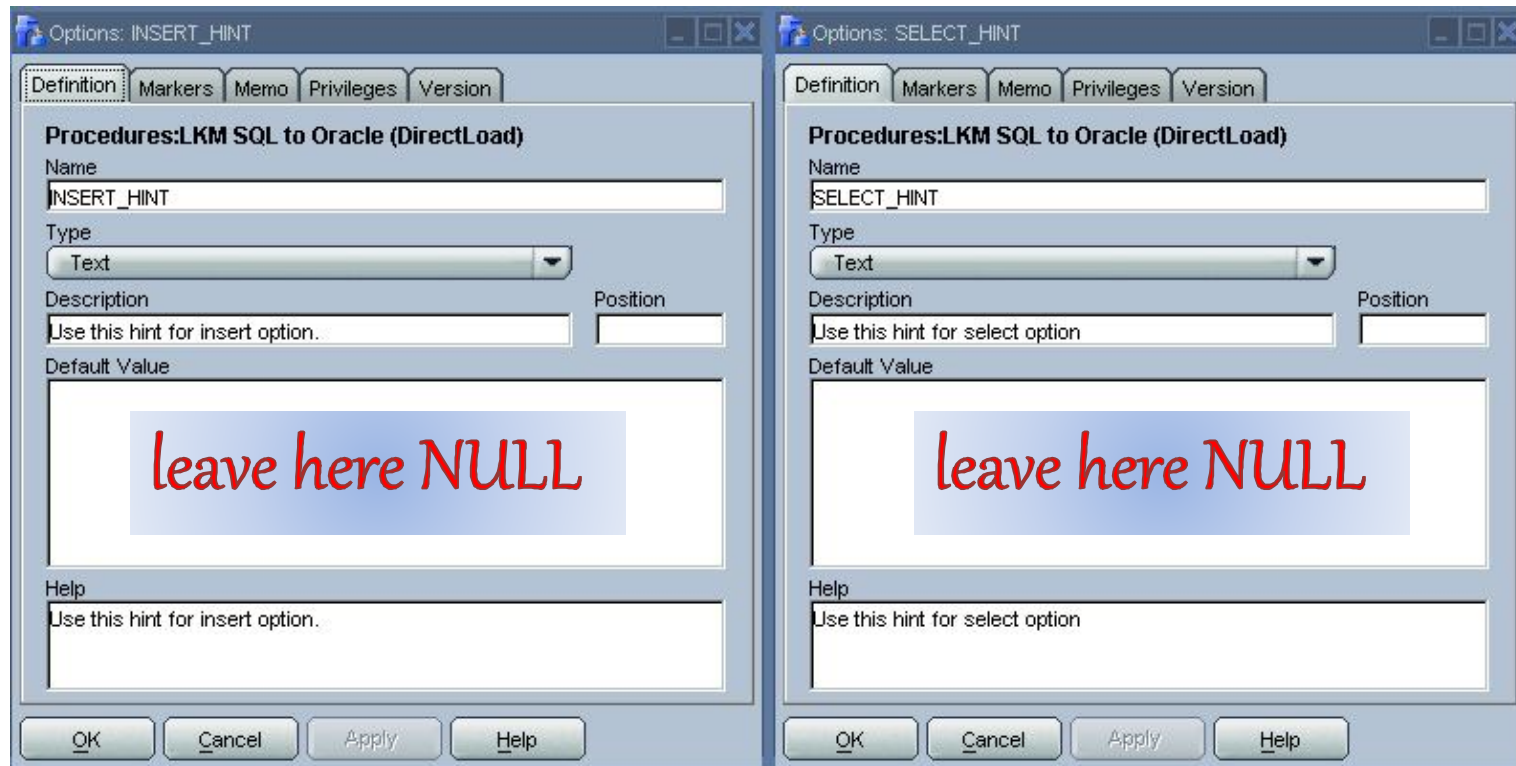
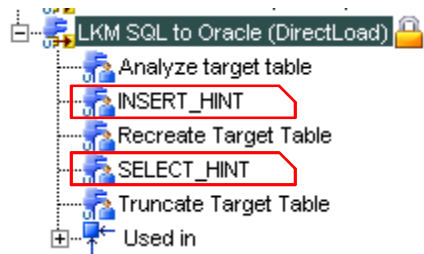
- Name: Update existing rows
- Log Counter: Update
- Log Level: 3
- Ignore Errors:
- Journalizing: Journalized Table in the Staging Area, Journalized Table in the current Interface
- Command on Target / Command on Source: (Selected)
- Technology: Oracle
- Transaction Isolation: <Undefined>
- Context: <Execution Context>
- Schema: <Undefined>
- Transaction: Transaction 1
- Commit: Commit

The Command field contains the following SQL code with static hints highlighted in red:

```
update
set
/*+ parallel(S,8) use_hash(S, T) */ <%=snpRef.getTable("L", "TARG_NAME", "A")%> T
(
  <%=snpRef.getColList("", "T.[COL_NAME]", ",\n\t", "", "(((SCD_UPD) and !TRG) and REV)")%>
  <%=snpRef.getColList("", "T.[COL_NAME]", ",\n\t", "", "(((SCD_UPD) and TRG) and REV)")%>
  <%=snpRef.getColList("", "T.[COL_NAME]", ",\n\t", "", "(UD1 and UPD)")%>
) =
(
  select
    /*+ PARALLEL(X, 4) USE_HASH(X,T) */ <%=snpRef.getPop("DISTINCT_ROWS")%>
    <%=snpRef.getColList("", "X.[COL_NAME]", ",\n\t\t", "", "(((SCD_UPD) and !TRG) and REV)")%>
    <%=snpRef.getColList("", "[EXPRESSION]", ",\n\t\t", "", "(((SCD_UPD) and TRG) and REV)")%>
    <%=snpRef.getColList("", "[EXPRESSION]", ",\n\t\t", "", "(UD1 and UPD)")%>
  from
    <%=snpRef.getTable("L", "INT_NAME", "A")%> X
  where
    <%=snpRef.getColList("", "X.[COL_NAME]=T.[COL_NAME]", " AND \n\t\t", "", "(SCD_NK)")%>
  and
    X.IND_UPDATE = 'U'
)
where (<%=snpRef.getColList("", "[COL_NAME]", ", ", "", "(SCD_NK)")%>|
in
(
  select
    /*+ PARALLEL(S, 4) USE_HASH(T,S) */ <%=snpRef.getColList("", "[COL_NAME]", ",\n\t\t", "", "(SCD_NK)")%>
  from
    <%=snpRef.getTable("L", "INT_NAME", "A")%> S
  where
    S.IND_UPDATE = 'U'
)
```


KM's... How to apply dynamic HINTS.

Step 1 : Create OPTIONS for KM's

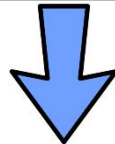


KM's... How to apply dynamic HINTS.

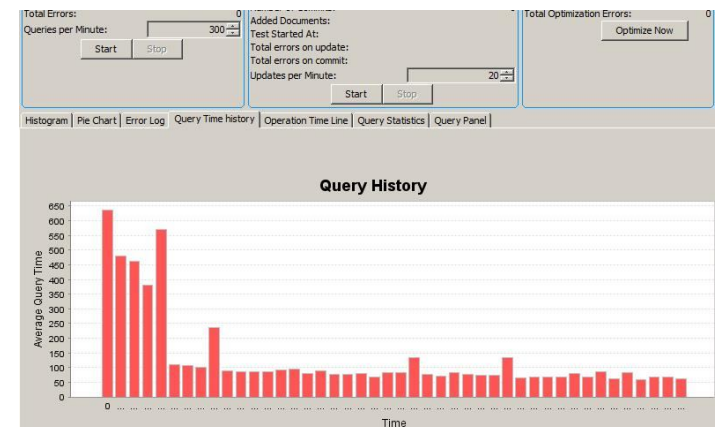
Step 2 : Insert this OPTIONS into KM's

```
insert <%=odiRef.getOption("INSERT_HINT")%>  
into <%=snpRef.getTable("L", "TARG_NAME", "A")%>
```

```
select <%=odiRef.getOption("SELECT_HINT")%>  
<%=snpRef.getPop("DISTINCT_ROWS")%>  
    <%=snpRef.getColList("", "[EXPRESSION]\t[ALIAS_SEP]  
[CX_COL_NAME]", ", \n\t", "", "")%>  
from <%=snpRef.getFrom()%>
```

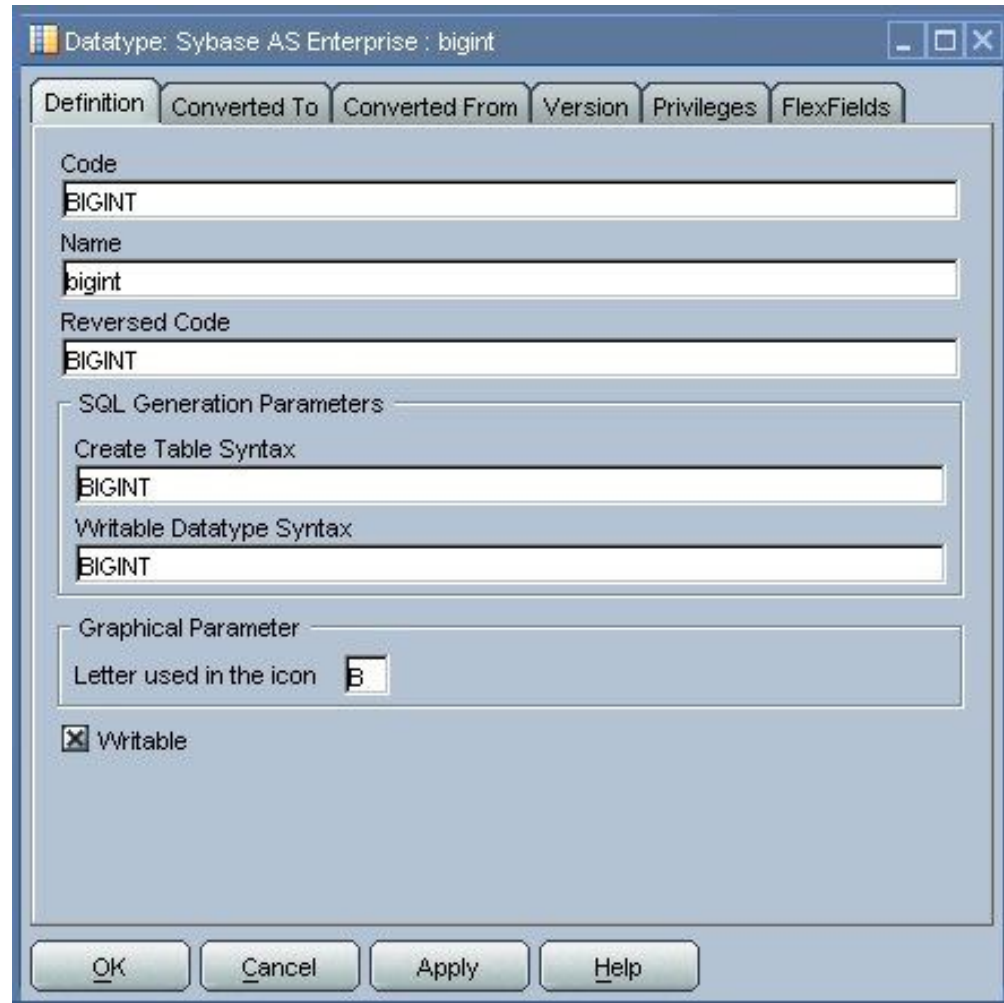
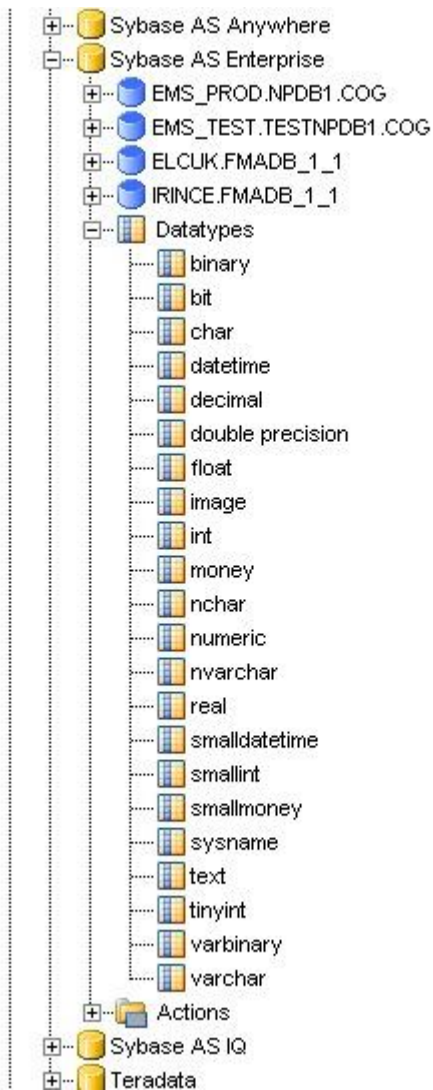


```
INSERT /*+ APPEND PARALLEL(t3, 8) */  
INTO t3  
SELECT /*+ parallel(t1) parallel(t2)  
ordered use_hash(t2) index(t1 t1_abc)  
index(t2 t2_abc) */ COUNT(*)  
FROM t1, t2  
WHERE t1.col1 = t2.col1;
```



Adding Datatypes

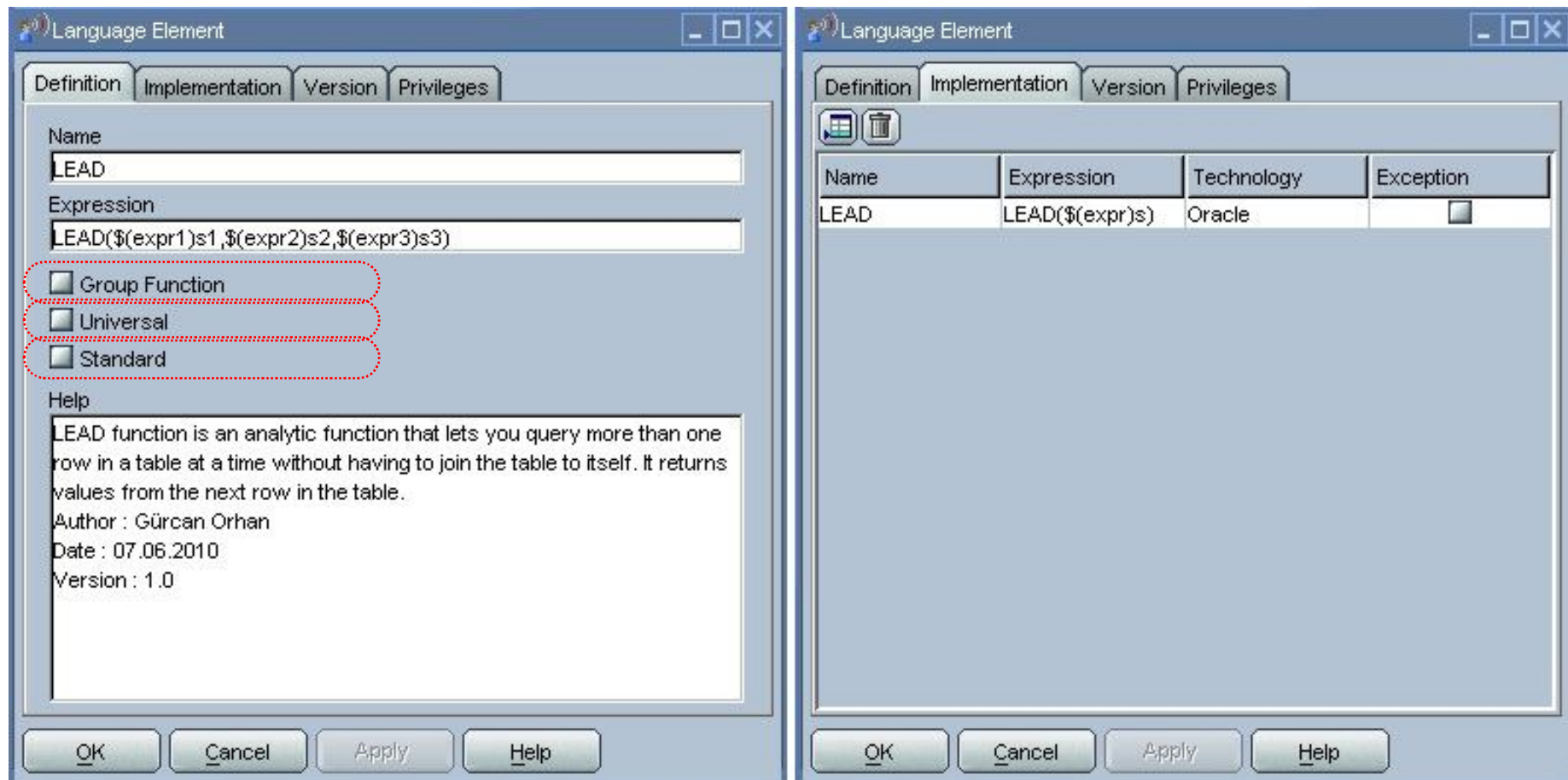
Right Click → Insert Datatype



<http://gurcanorhan.wordpress.com/2012/02/28/adding-datatypes-to-odi/>

Adding Functions

Topology Manager → Languages → SQL → Aggregate (or other)



<http://gurcanorhan.wordpress.com/2012/12/05/adding-functions-to-odi/>

Archiving ODI Logs... Required tables.

Repository table list for logging of ODI stored in work repository schema

SNP_EXP_TXT
SNP_SCEN_REPORT
SNP_SESS_STEP
SNP_SESS_TASK
SNP_SESS_TASK_LOG
SNP_SESS_TXT_LOG
SNP_SESSION
SNP_STEP_LOG
SNP_STEP_REPORT
SNP_TASK_TXT
SNP_VAR_DATA
SNP_VAR_SESS

Create these tables (**without referential integrity, constraints, indexes, etc.**) with a suffix or prefix in a different schema.

ARC_SNP_EXP_TXT
ARC_SNP_SCEN_REPORT
ARC_SNP_SESS_STEP
ARC_SNP_SESS_TASK
ARC_SNP_SESS_TASK_LOG
ARC_SNP_SESS_TXT_LOG
ARC_SNP_SESSION
ARC_SNP_STEP_LOG
ARC_SNP_STEP_REPORT
ARC_SNP_TASK_TXT
ARC_SNP_VAR_DATA
ARC_SNP_VAR_SESS

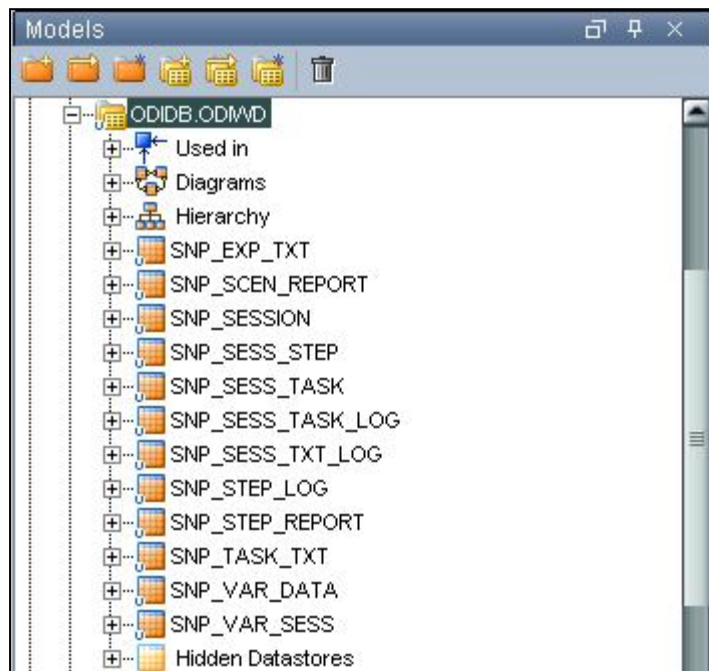


<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

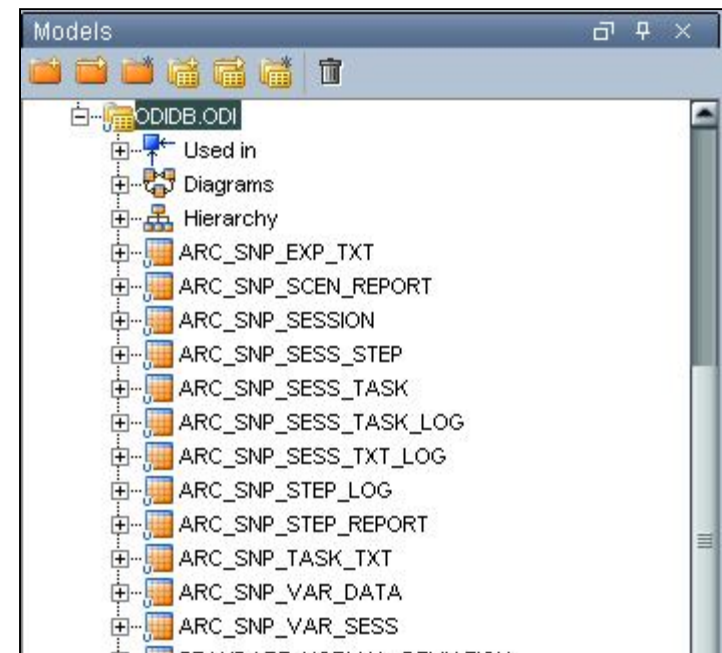
Archiving ODI Logs... Reverse.

Reverse all required tables in ODI

SOURCE



TARGET



<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

Archiving ODI Logs... Create interfaces.

I_SNP_EXP_TXT

SOURCE	TARGET
SNP_EXP_TXT	ARC_SNP_EXP_TXT

FILTERS	JOINS
TRUNC(SNP_EXP_TXT.FIRST_DATE) < TRUNC(SYSDATE) - #V_Purge_Log_Retention	(none)

<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

Archiving ODI Logs... Create interfaces.

I_SNP_SCEN_REPORT

SOURCE	TARGET
SNP_SCEN_REPORT	ARC_SNP_SCEN_REPORT

FILTERS	JOINS
SNP_SCEN_REPORT.CONTEXT_CODE = 'PRODUCTION' AND TRUNC(SNP_SCEN_REPORT.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention	(none)

<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

Archiving ODI Logs... Create interfaces.

I_SNP_SESS_STEP

SOURCE	TARGET
SNP_SESS_STEP SNP_SESSION	ARC_SNP_SESS_STEP

FILTERS	JOINS
SNP_SESSION.CONTEXT_CODE = 'PRODUCTION' AND TRUNC(SNP_SESSION.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention	SNP_SESS_STEP.SESS_NO=SNP_SESSION.SESS_NO

<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

Archiving ODI Logs... Create interfaces.

I_SNP_SESS_TASK

SOURCE	TARGET
SNP_SESS_TASK SNP_SESS_STEP SNP_SESSION	ARC_SNP_SESS_TASK
FILTERS	JOINS
SNP_SESSION.CONTEXT_CODE = 'PRODUCTION' AND TRUNC(SNP_SESSION.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention	SNP_SESS_TASK.SESS_NO=SNP_SESS_STEP.SESS_NO AND SNP_SESS_TASK.NNO=SNP_SESS_STEP.NNO SNP_SESS_STEP.SESS_NO=SNP_SESSION.SESS_NO

<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

Archiving ODI Logs... Create interfaces.

I_SNP_SESS_TASK_LOG

SOURCE	TARGET
SNP_SESS_TASK_LOG SNP_STEP_LOG SNP_SESS_STEP SNP_SESSION	ARC_SNP_SESS_TASK_LOG
FILTERS	JOINS
SNP_SESSION.CONTEXT_CODE = 'PRODUCTION' AND TRUNC(SNP_SESSION.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention	SNP_SESS_TASK_LOG.NNO=SNP_STEP_LOG.NNO AND SNP_SESS_TASK_LOG.SESS_NO=SNP_STEP_LOG.SESS_NO AND SNP_SESS_TASK_LOG.NB_RUN=SNP_STEP_LOG.NB_RUN SNP_STEP_LOG.SESS_NO=SNP_SESS_STEP.SESS_NO AND SNP_STEP_LOG.NNO=SNP_SESS_STEP.NNO SNP_SESS_STEP.SESS_NO=SNP_SESSION.SESS_NO

<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

Archiving ODI Logs... Create interfaces.

I_SNP_SESS_TXT_LOG

SOURCE	TARGET
SNP_SESS_TXT_LOG SNP_SESS_TASK_LOG SNP_STEP_LOG SNP_SESS_STEP SNP_SESSION	ARC_SNP_SESS_TXT_LOG
FILTERS	JOINS
SNP_SESSION.CONTEXT_CODE = 'PRODUCTION' AND TRUNC(SNP_SESSION.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention	SNP_SESS_TXT_LOG.SESS_NO=SNP_SESS_TASK_LOG.SESS_NO AND SNP_SESS_TXT_LOG.NNO=SNP_SESS_TASK_LOG.NNO AND SNP_SESS_TXT_LOG.NB_RUN=SNP_SESS_TASK_LOG.NB_RUN AND SNP_SESS_TXT_LOG.SCEN_TASK_NO=SNP_SESS_TASK_LOG.SCEN_TASK_NO SNP_SESS_TASK_LOG.NNO=SNP_STEP_LOG.NNO AND SNP_SESS_TASK_LOG.SESS_NO=SNP_STEP_LOG.SESS_NO AND SNP_SESS_TASK_LOG.NB_RUN=SNP_STEP_LOG.NB_RUN SNP_STEP_LOG.SESS_NO=SNP_SESS_STEP.SESS_NO AND SNP_STEP_LOG.NNO=SNP_SESS_STEP.NNO SNP_SESS_STEP.SESS_NO=SNP_SESSION.SESS_NO

<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

Archiving ODI Logs... Create interfaces.

I_SNP_SESSION

SOURCE	TARGET
SNP_SESSION	ARC_SNP_SESSION

FILTERS	JOINS
SNP_SESSION.CONTEXT_CODE = 'PRODUCTION' AND TRUNC(SNP_SESSION.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention	(none)

<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

Archiving ODI Logs... Create interfaces.

I_SNP_STEP_LOG

SOURCE	TARGET
SNP_STEP_LOG SNP_SESS_STEP SNP_SESSION	ARC_SNP_STEP_LOG
FILTERS	JOINS
SNP_SESSION.CONTEXT_CODE = 'PRODUCTION' AND TRUNC(SNP_SESSION.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention	SNP_STEP_LOG.SESS_NO=SNP_SESS_STEP.SESS_NO AND SNP_STEP_LOG.NNO=SNP_SESS_STEP.NNO SNP_SESS_STEP.SESS_NO=SNP_SESSION.SESS_NO

<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

Archiving ODI Logs... Create interfaces.

I_SNP_STEP_REPORT

SOURCE	TARGET
SNP_STEP_REPORT SNP_SCEN_REPORT	ARC_SNP_STEP_REPORT

FILTERS	JOINS
SNP_SCEN_REPORT.CONTEXT_CODE = 'PRODUCTION' AND TRUNC(SNP_SCEN_REPORT.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention	SNP_STEP_REPORT.SCEN_NO=SNP_SCEN_REPORT.SCEN_NO AND SNP_STEP_REPORT.SCEN_RUN_NO=SNP_SCEN_REPORT.SCEN_RUN_NO

<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

Archiving ODI Logs... Create interfaces.

I_SNP_TASK_TXT

SOURCE	TARGET
SNP_TASK_TXT SNP_SESS_TASK SNP_SESS_STEP SNP_SESSION	ARC_SNP_TASK_TXT
FILTERS	JOINS
SNP_SESSION.CONTEXT_CODE = 'PRODUCTION' AND TRUNC(SNP_SESSION.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention	SNP_TASK_TXT.SESS_NO=SNP_SESS_TASK.SESS_NO AND SNP_TASK_TXT.NNO=SNP_SESS_TASK.NNO AND SNP_TASK_TXT.SCEN_TASK_NO=SNP_SESS_TASK.SCEN_TASK_NO SNP_SESS_TASK.SESS_NO=SNP_SESS_STEP.SESS_NO AND SNP_SESS_TASK.NNO=SNP_SESS_STEP.NNO SNP_SESS_STEP.SESS_NO=SNP_SESSION.SESS_NO

<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

Archiving ODI Logs... Create interfaces.

I_SNP_VAR_DATA

SOURCE	TARGET
SNP_VAR_DATA	ARC_SNP_VAR_DATA

FILTERS	JOINS
SNP_VAR_DATA.CONTEXT_CODE = 'PRODUCTION' AND TRUNC(SNP_VAR_DATA.FIRST_DATE) < TRUNC(SYSDATE) - #V_Purge_Log_Retention	(none)

<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

Archiving ODI Logs... Create interfaces.

I_SNP_VAR_SESS

SOURCE	TARGET
SNP_VAR_SESS	ARC_SNP_VAR_SESS

FILTERS	JOINS
SNP_SESSION.CONTEXT_CODE = 'PRODUCTION' AND TRUNC(SNP_SESSION.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention	(none)

<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

Archiving ODI Logs... Create procedure

Create a delete procedure... Delete from parent to child

ORDER	STEP NAME	COMMAND
0	DELETE ODIWD.SNP_SESS_TXT_LOG	DELETE /*+ USE_HASH(A) PARALLEL(A) */ FROM ODIWD.SNP_SESS_TXT_LOG A WHERE SESS_NO IN (SELECT SESS_NO FROM ODIWD.SNP_SESSION SESS WHERE TRUNC(SESS.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention)
10	DELETE ODIWD.SNP_SESS_TASK_LOG	DELETE /*+ USE_HASH(A) PARALLEL(A) */ FROM ODIWD.SNP_SESS_TASK_LOG A WHERE SESS_NO IN (SELECT SESS_NO FROM ODIWD.SNP_SESSION SESS WHERE TRUNC(SESS.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention)
20	DELETE ODIWD.SNP_TASK_TXT	DELETE /*+ USE_HASH(A) PARALLEL(A) */ FROM ODIWD.SNP_TASK_TXT A WHERE SESS_NO IN (SELECT SESS_NO FROM ODIWD.SNP_SESSION SESS WHERE TRUNC(SESS.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention)
30	DELETE ODIWD.SNP_STEP_LOG	DELETE /*+ USE_HASH(A) PARALLEL(A) */ FROM ODIWD.SNP_STEP_LOG A WHERE SESS_NO IN (SELECT SESS_NO FROM ODIWD.SNP_SESSION SESS WHERE TRUNC(SESS.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention)
40	DELETE ODIWD.SNP_SESS_TASK	DELETE /*+ USE_HASH(A) PARALLEL(A) */ FROM ODIWD.SNP_SESS_TASK A WHERE SESS_NO IN (SELECT SESS_NO FROM ODIWD.SNP_SESSION SESS WHERE TRUNC(SESS.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention)

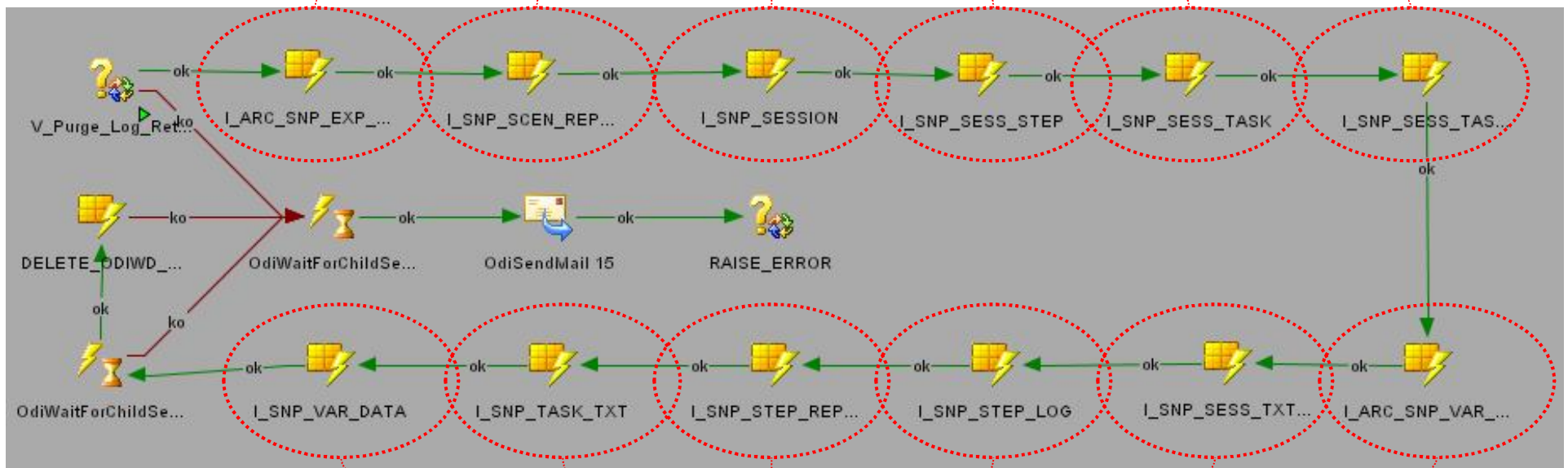
<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

Archiving ODI Logs... Create procedure

ORDER	STEP NAME	COMMAND
50	DELETE ODIWD.SNP_SESS_STEP	DELETE /*+ USE_HASH(A) PARALLEL(A) */ FROM ODIWD.SNP_SESS_STEP A WHERE SESS_NO IN (SELECT SESS_NO FROM ODIWD.SNP_SESSION SESS WHERE TRUNC(SESS.SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention)
60	DELETE ODIWD.SNP_VAR_DATA	DELETE /*+ USE_HASH(A) PARALLEL(A) */ FROM ODIWD.SNP_VAR_DATA A WHERE TRUNC(A.FIRST_DATE) < TRUNC(SYSDATE) - #V_Purge_Log_Retention
70	DELETE ODIWD.SNP_VAR_SESS	DELETE FROM ODIWD.SNP_VAR_SESS WHERE SESS_NO IN (SELECT SESS_NO FROM ODIWD.SNP_SESSION A WHERE TRUNC(SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention)
80	DELETE ODIWD.SNP_EXP_TXT	DELETE /*+ USE_HASH(A) PARALLEL(A) */ FROM ODIWD.SNP_EXP_TXT A WHERE TRUNC(A.FIRST_DATE) < TRUNC(SYSDATE) - #V_Purge_Log_Retention
90	DELETE ODIWD.SNP_SESSION	DELETE /*+ USE_HASH(A) PARALLEL(A) */ FROM ODIWD.SNP_SESSION A WHERE TRUNC(SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention
100	DELETE ODIWD.SNP_STEP_REPORT	DELETE /*+ USE_HASH(A) PARALLEL(A) */ FROM ODIWD.SNP_STEP_REPORT A WHERE TRUNC(A.STEP_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention
110	DELETE ODIWD.SNP_SCEN_REPORT	DELETE /*+ USE_HASH(A) PARALLEL(A) */ FROM ODIWD.SNP_SCEN_REPORT A WHERE TRUNC(SESS_BEG) < TRUNC(SYSDATE) - #V_Purge_Log_Retention

Archiving ODI Logs... Packaging

Running in «Asynchronous Mode»

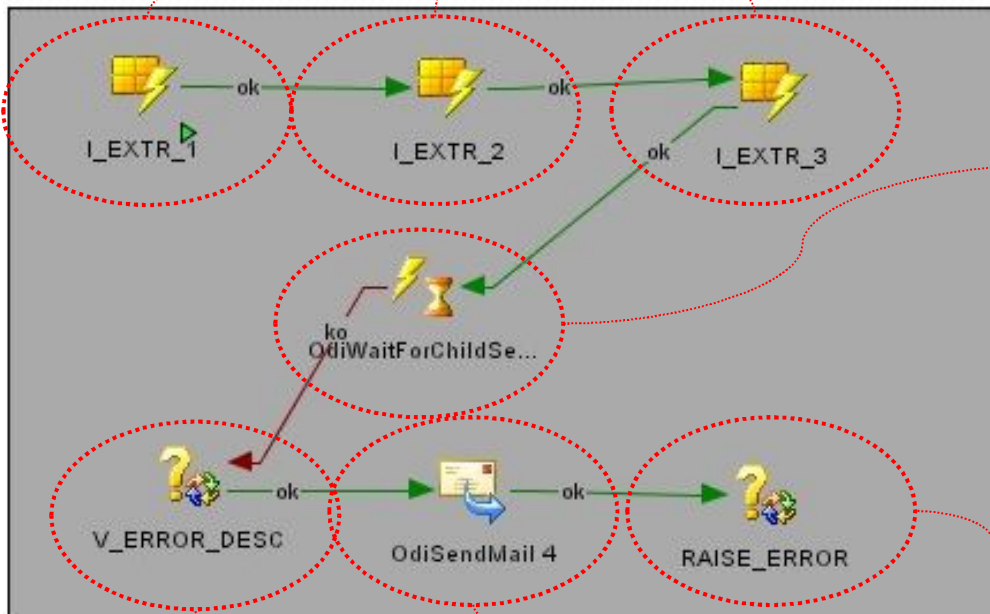


Running in «Asynchronous Mode»

<http://gurcanorhan.wordpress.com/2012/12/05/archiving-odi-logs/>

Handling Alerts

Running in «Asynchronous Mode»



Error Handler

Max. Number of Failed Child Sessions = 1

Raise Error

(error refresh variable)

Mail body
(refresh variable)

Send mail

<http://gurcanorhan.wordpress.com/2012/11/30/odi-alert-mechanism/>

Handling – {in ETL} – Data Quality

Known data quality issues that can be covered in ETL

Step 1 : Prepare your data quality scripts.

Step 2 : Put those scripts into files.

Step 3 : Read contents of those files or create a table for scripts. Execute this script into your database, insert output to an ERROR table.

Step 4 : Select count from ERROR table by ERROR_CODE and loop it from beginning to end for sysdate.

Step 5 : Send e-mail for each script, attach the appropriate file and show how many rows are generated in this ERROR_CODE.

<http://gurcanorhan.wordpress.com/2012/11/30/odi-handling-dq/>

Handling – {in ETL} – Data Quality

Step 1 : Prepare your data quality scripts.

- If you are generating a hierarchical tree, make sure your every node connects to its parent
- Check duplicates in names, addresss and other important fields
- Check primary key behaviour from your sources



<http://gurcanorhan.wordpress.com/2012/11/30/odi-handling-dq/>

Handling – {in ETL} – Data Quality

Step 2 : Put those scripts into files.

- Create as many scripts you can.
- Copy files to operating system, where agent is running. You should have read grant for this directory.



<http://www.gurcanorhan.com/ng-dq/>

Handling – {in ETL} – Data Quality

Step 3 : Create tables for scripts and output.

```
CREATE TABLE MY_ERROR_SCRIPTS
(ERROR_CODE NUMBER(2),
ERROR_DESC VARCHAR2(150 BYTE),
ERROR_SCRIPT_DWH CLOB,
ERROR_SCRIPT_OLTP CLOB
)
LOB (ERROR_SCRIPT_DWH) STORE AS (
TABLESPACE MY_TBS
ENABLE STORAGE IN ROW CHUNK 32768 RETENTION NOCACHE NOLOGGING STORAGE
(INITIAL 160K NEXT 1M MINEXTENTS 1 MAXEXTENTS UNLIMITED PCTINCREASE 0))
LOB (ERROR_SCRIPT_OLTP) STORE AS (
TABLESPACE MY_TBS
ENABLE STORAGE IN ROW CHUNK 32768 RETENTION NOCACHE NOLOGGING STORAGE
(INITIAL 160K NEXT 1M MINEXTENTS 1 MAXEXTENTS UNLIMITED PCTINCREASE 0))
TABLESPACE MY_TBS
LOGGING NOCOMPRESS NOCACHE NOPARALLEL MONITORING;
```

```
CREATE TABLE MY_ERROR_TABLE
(DATETIME DATE,
TRX_ID INTEGER,
ERROR_CODE NUMBER(2),
ERROR_DESC VARCHAR2(150 BYTE)
)
TABLESPACE MY_TBS
LOGGING NOCOMPRESS NOCACHE NOPARALLEL MONITORING;
```

<http://gurcanorhan.wordpress.com/2012/11/30/odi-handling-dq/>

Handling – {in ETL} – Data Quality

Step 3 : Read scripts. Execute this script into your database, insert output to an ERROR table.

- Delete today's records;
- Execute all Error Scripts; (since I am changing my own codes, below codes must be rewritten to your environment)

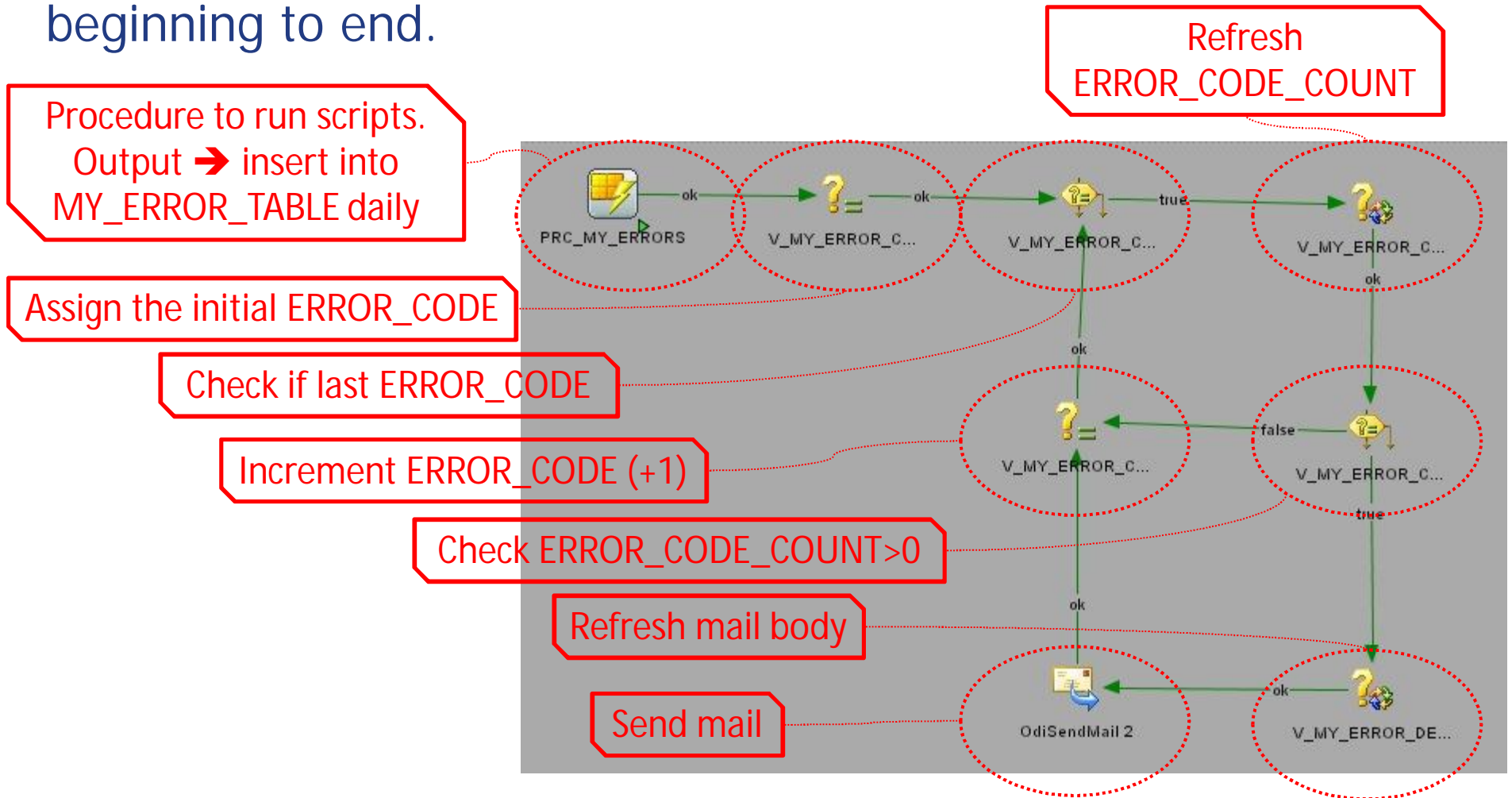
```
DECLARE
TYPE TransactionRec IS RECORD (transaction_id integer);
TYPE TransactionSet IS TABLE OF TransactionRec;
ContractSet TransactionSet;CURSOR C1 IS
select ERROR_CODE, ERROR_DESC, ERROR_SCRIPT_DWH from
<%=odiRef.getSchemaName("DB.MY_SCHEMA", "D")%>.ERROR_SCRIPTS;
BEGIN
FOR C1_REC IN C1 LOOPEXECUTE IMMEDIATE to_char(C1_REC.ERROR_SCRIPT_DWH) BULK COLLECT
INTO ContractSet;IF ContractSet.FIRST IS NOT NULL THEN
FOR i IN ContractSet.FIRST..ContractSet.LAST
LOOP
INSERT INTO <%=odiRef.getSchemaName("DB.MY_SCHEMA", "D")%>.MY_ERROR_TABLE (DATETIME,
TRANSACTION_ID, ERROR_CODE, ERROR_DESC) VALUES
(TO_DATE('SYSDATE', 'YYYYMMDD'), ContractSet(i). transaction_id, C1_REC.ERROR_CODE,
C1_REC.ERROR_DESC);
COMMIT;
END LOOP;
END IF;
END LOOP;
END;
```

```
delete from <%=odiRef.getSchemaName("DB.MY_SCHEMA",
"D")%>.MY_ERROR_TABLE where DATETIME = TO_DATE('SYSDATE', 'YYYYMMDD')
```

<http://gurcanorhan.wordpress.com/2012/11/30/odi-handling-dq/>

Handling – {in ETL} – Data Quality

Step 4 : Read ERROR table by ERROR_CODE and loop it from beginning to end.



<http://gurcanorhan.wordpress.com/2012/11/30/odi-handling-dq/>

Handling – {in ETL} – Data Quality

Step 5 : Send e-mail for each script, attach the appropriate file and show how many rows are generated in this ERROR_CODE.

- ❖ **Mail Server** : #V_MAIL_SERVER_IP
- ❖ **From** : This is static, user that you are sending mail.
- ❖ **TO** : #V_ERROR_MAIL_TO (need to be refreshed in the beginning of your ETL or current package)
- ❖ **CC** : #V_ERROR_MAIL_CC (need to be refreshed in the beginning of your ETL or current package)
- ❖ **BCC** : #V_ERROR_MAIL_BCC (need to be refreshed in the beginning of your ETL or current package)
- ❖ **Subject** : There are #V_MY_ERROR_COUNT errors exist in your system (Error Code = #V_MY_ERROR_CODE)
- ❖ **Attachment** : /data/my_errors/MY_Error_#V_MY_ERROR_CODE.txt (will represent as /data/my_errors/MY_Error_1.txt initially, then will attach regarding file in the loop, every step will attach its own script file)
- ❖ **Message Body** : #V_MY_ERROR_DESC

<http://gurcanorhan.wordpress.com/2012/11/30/odi-handling-dq/>

File2Table... Summary

Step 1 : Operating system folders

Step 2 : IKM Knowledge Module

Step 3 : ETL_FILE LOG (database table)

Step 4 : ODI Procedure to rename files for external table usage

Step 5 : ODI Procedures to finish working with files

Step 6 : ODI Procedure to *Get File List* of operating system

Step 7 : ODI Interface (loading and transforming)

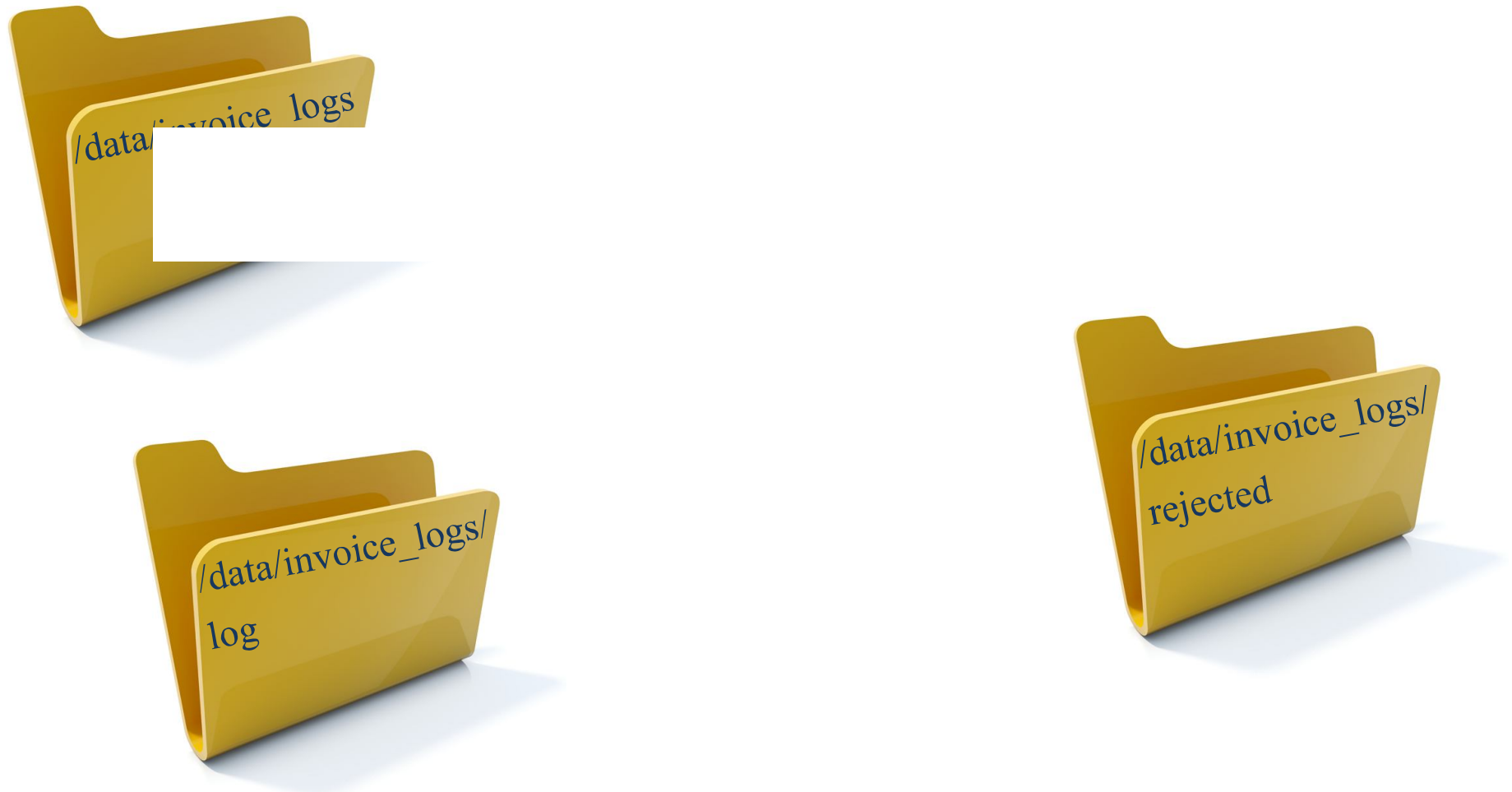
Step 8 : Some ODI Variables

Step 9 : ODI Package for running everything in right order

<http://gurcanorhan.wordpress.com/2010/11/13/loading-multiple-files-with-odi/>

File2Table... Preparing environment, folders (Step 1)

Illustration for files received from «**invoice_logs**»



<http://gurcanorhan.wordpress.com/2010/11/13/loading-multiple-files-with-odi/>

File2Table... Preparing environment, IKM (Step 2)

- ❖ Copy current KM : IKM SQL Control Append
- ❖ Rename as : IKM SQL Control Append (Direct Load,HINT)
- ❖ Add Options : «SELECT HINT», «INSERT HINT»
- ❖ Add New Row : «Parallel DML», *Transaction 0, No Commit*
`alter session enable parallel dml`
- ❖ Modify : «Insert new rows» to *Transaction 0, No Commit*
- ❖ Add New fixed Row : «Commit transaction» to *Transaction 0, Commit*
`/* commit */`

<http://gurcanorhan.wordpress.com/2010/11/13/loading-multiple-files-with-odi/>

File2Table... Preparing environment, IKM (Step 2)

```
INSERT /*+ APPEND PARALLEL(4) */
INTO ODIDB.MY_TARGET_TABLE
(
  MY_TARGET_COLUMN_1,
  MY_TARGET_COLUMN_2,
  MY_TARGET_COLUMN_3
)
SELECT
/*+ PARALLEL(MY_SOURCE_TABLE) FULL(MY_SOURCE_TABLE) */
MY_SOURCE_TABLE.MY_SOURCE_COLUMN_1,
MY_SOURCE_TABLE.MY_SOURCE_COLUMN_2,
MY_SOURCE_TABLE.MY_SOURCE_COLUMN_3
FROM ODISTG.I$MY_TARGET_TABLE MY_SOURCE_TABLE
WHERE MY_SOURCE_TABLE.MY_SOURCE_COLUMN_4 = 'USA'
      AND MY_SOURCE_TABLE.MY_SOURCE_COLUMN_5 = 'New York'
```

<http://gurcanorhan.wordpress.com/2010/11/13/loading-multiple-files-with-odi/>

File2Table... Prepare log table (Step 3)

Create table ETL_FILE_LOG

COLUMN NAME	PK	NULL?	DATA TYPE	DEF	COMMENT
FILE_ID	1	N	NUMBER (10)		The unique identification number of file.
FILE_NAME		N	VARCHAR2 (50 Byte)		The name of file to be processed.
FILE_GROUP		N	VARCHAR2 (20 Byte)		Source system name or group with the same template.
FILE_COPIED_FLAG		Y	NUMBER (1)	0	0:not copied, 1:copied successfully, 2:error in copy.
FILE_COPY_DATE		Y	DATE		Date of file copied.
FILE_READ_FLAG		Y	NUMBER (1)	0	0:not read, 1:read successfully, 2:error in read.
FILE_READ_DATE		Y	DATE		Date of file read.
FILE_PROCESSED_FLAG		Y	NUMBER (1)	0	Date of file processed.
FILE_PROCESSED_DATE		Y	DATE		0:not processed, 1:processed successfully, 2:error in process.

<http://gurcanorhan.wordpress.com/2010/11/13/loading-multiple-files-with-odi/>

File2Table... Rename File - ODI Procedure (Step 4)

❖ Create external table «STG.INVOICE_LOG»

❖ Create ODI Procedure «Rename File»

❖ Step 1 ; delete previous file

```
rm /data/invoice_logs/my_external_table.ext
```

❖ Step 2 ; rename next file

```
mv /data/invoice_logs/#FILE_NAME  
/data/invoice_logs/my_external_table.ext
```

<http://gurcanorhan.wordpress.com/2010/11/13/loading-multiple-files-with-odi/>

File2Table... Update Processed File (Step 5)

❖ Create ODI Procedure – «UPDATE PROCESSED FILE»

❖ Step 1 ; update ETL_FILE_LOG (Processed)

```
UPDATE ODI.ETL_FILE_LOG
SET FILE_PROCESSED_FLAG = 1, FILE_PROCESSED_DATE = SYSDATE
WHERE FILE_ID = #FILE_ID
```

❖ Step 2 ; delete processed data file

```
rm /data/invoice_logs/my_external_table.ext
```

❖ Step 3 ; delete processed log file

```
rm /data/invoice_logs/my_external_table.log
```

<http://gurcanorhan.wordpress.com/2010/11/13/loading-multiple-files-with-odi/>

File2Table... GetFileList from OS (Step 6)

❖ Create ODI Procedure Jyhton Technology – «GetFileList»

```
import java.lang as lang
import java.sql as sql
import snpsftp
import java.lang.String
import os
import java.io.File as File
#db connection
driverSrc = 'oracle.jdbc.driver.OracleDriver'
lang.Class.forName(driverSrc)
#Production Environment
urlSrc = 'jdbc:oracle:thin:@<host>:<port>:<sid>'
#Development Environment
#urlSrc = 'jdbc:oracle:thin:@<host>:<port>:<sid>'
userSrc = 'ODI'
passwdSrc = 'ODI'
ConSrc = sql.DriverManager.getConnection(urlSrc, userSrc, passwdSrc);
readDBLink = ConSrc.createStatement()
syslist = os.listdir('<%=odiRef.getOption("SOURCE_DIRECTORY")%>')
for sys in syslist:
str = java.lang.String(sys)
if str.length() > 8:
sqlDBLink = "select * from ODI.ETL_FILE_LOG where FILE_NAME = '" + sys + "'"
rqteDBLink = readDBLink.executeQuery(sqlDBLink)
if not rqteDBLink.next():
sqlDBLink = "insert into ODI.ETL_FILE_LOG (FILE_ID, FILE_NAME, FILE_GROUP, FILE_SUB_GROUP,
FILE_READ_FLAG, FILE_READ_DATE) values (ODI.SEQ_FILE_ID.NEXTVAL, '" + sys + "',
'<source_system_name>', '<file_type>', '1', SYSDATE)"
rqteDBLink = readDBLink.executeQuery(sqlDBLink)
ConSrc.close()
```

<http://gurcanorhan.wordpress.com/2010/11/13/loading-multiple-files-with-odi/>

File2Table... ODI Interface (Step 7)

- ❖ Create ODI Interface (external table → db table)
 - ❖ Source : STG.INVOICE_LOG
(based on external table → my_external_table.ext)
 - ❖ Target : DWH.INVOICE_LOGS
- ❖ KM : IKM SQL Control Append (Direct Load,HINT)

Truncate : No

Select Hint : */*+ PARALLEL(4) */*

Insert Hint : */*+ APPEND PARALLEL(4) NOLOGGING */*

<http://gurcanorhan.wordpress.com/2010/11/13/loading-multiple-files-with-odi/>

File2Table... ODI Variables (Step 8)

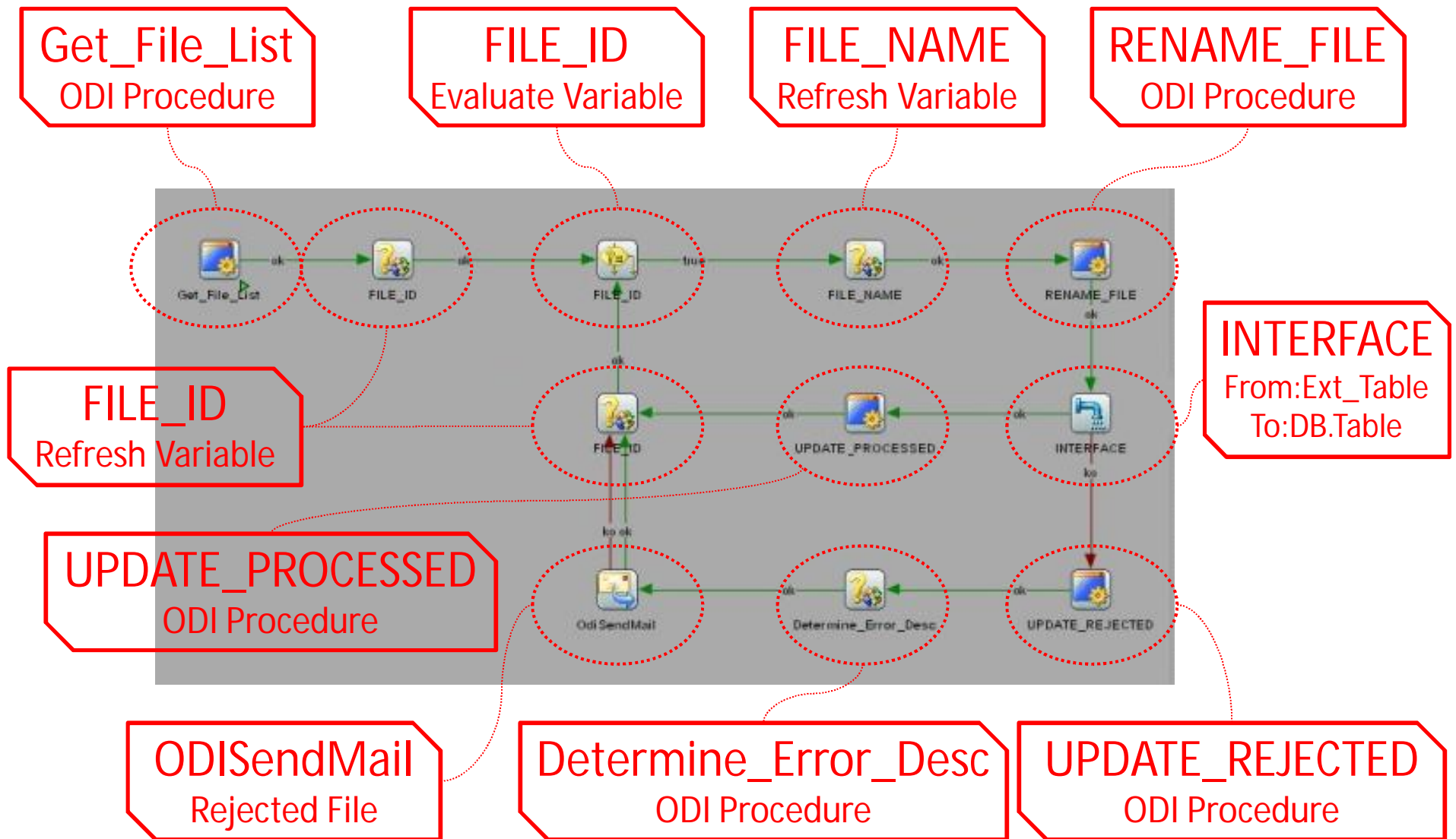
- ❖ Create ODI Variable to refresh – «File_ID»

```
SELECT NVL(MIN(FILE_ID), 0) FROM ODI.ETL_FILE_LOG
WHERE FILE_READ_FLAG = 1
      AND FILE_PROCESSED_FLAG = 0
      AND FILE_GROUP = 'INVOICE_LOGS'
```

- ❖ Create ODI Variable to refresh – «File_Name»



```
SELECT FILE_NAME FROM ODI.ETL_FILE_LOG
WHERE FILE_ID = #FILE_ID
```

File2Table... Pack-up everything (Step 9)



<http://gurcanorhan.wordpress.com/2010/11/13/loading-multiple-files-with-odi/>

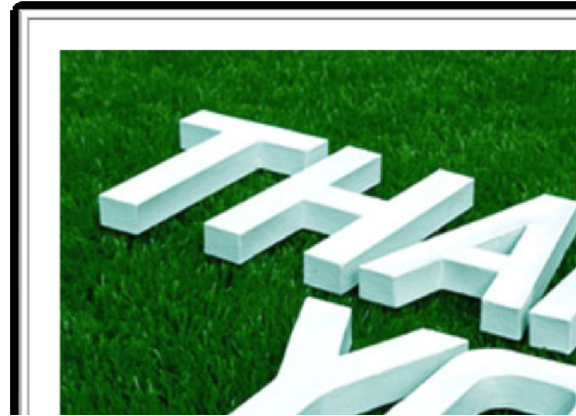
Final Words...

 <http://gurcanorhan.wordpress.com>
 http://www.twitter.com/gurcan_orhan
<http://tr.linkedin.com/in/gurcanorhan>

TROUG
TURKISH ORACLE USER GROUP



Final Words...



igorhan@gmail.com

 <http://gurcanorhan.wordpress.com>

 http://www.twitter.com/gurcan_orhan

<http://tr.linkedin.com/in/gurcanorhan>

TROUG
TURKISH ORACLE USER GROUP

