Full Load Prod -> erpapp01 10.10.130.31

Contents

[1. Introduction 2](#_Toc399757501)

[2. Got minus one from a read call 2](#_Toc399757502)

[3. Full load Execution Fails 4](#_Toc399757503)

[3.1 Solution 1 4](#_Toc399757504)

[3.2 Solution 2 4](#_Toc399757505)

[3.2.1 Duplicates in W\_DOMAIN\_MEMBER\_MAP\_G 5](#_Toc399757506)

[3.2.2 Duplicates in W\_EMPLOYEE\_DS 8](#_Toc399757507)

[3.2.3 Duplicates in W\_PRODUCT\_ATTRIBS\_D 11](#_Toc399757508)

[3.2.4 Duplicates in W\_STATUS\_D 14](#_Toc399757509)

[3.2.5 Duplicates in W\_PRODUCT\_D 18](#_Toc399757510)

[3.2.6 Duplicates in W\_PROC\_PROD\_D 22](#_Toc399757511)

[4. TEMP Table Space full 27](#_Toc399757512)

[5. End of Full Load 28](#_Toc399757513)

[6. Checking the execution of the customized tasks 28](#_Toc399757514)

[6.1 SDE PROCPO Fact Group 28](#_Toc399757515)

[6.2 SDE PURAGG Fact Group 29](#_Toc399757516)

[6.3 SDE OP Fact Group 29](#_Toc399757517)

[6.4 SDE PROCPURRCPT Fact Group 30](#_Toc399757518)

[6.5 SDE PROCPURREQ Fact Group 30](#_Toc399757519)

[6.6 SIL PROCPO Fact Group 31](#_Toc399757520)

[6.7 SIL PROCPURREQ Fact Group 31](#_Toc399757521)

[6.8 SIL PROCPURCY Fact Group 32](#_Toc399757522)

[7 Purchase Cost Fact Obtain the Original Promised Date 33](#_Toc399757523)

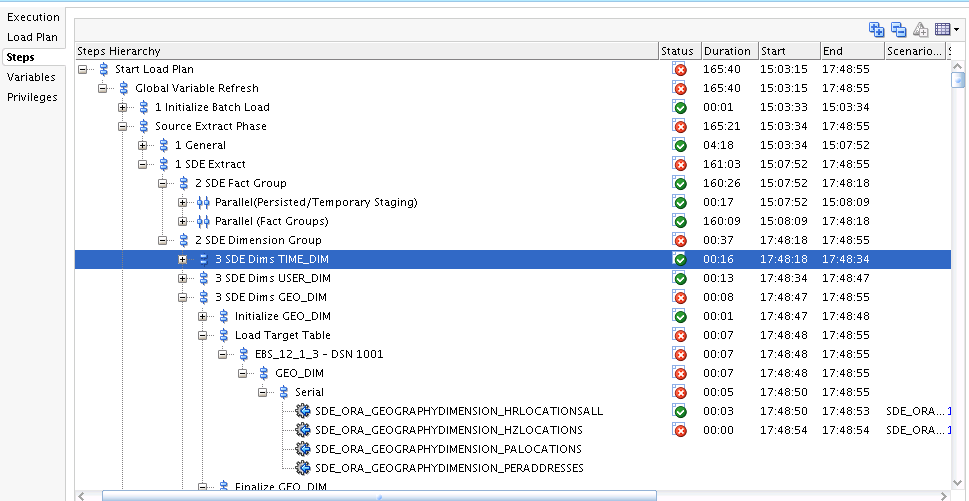
# Introduction

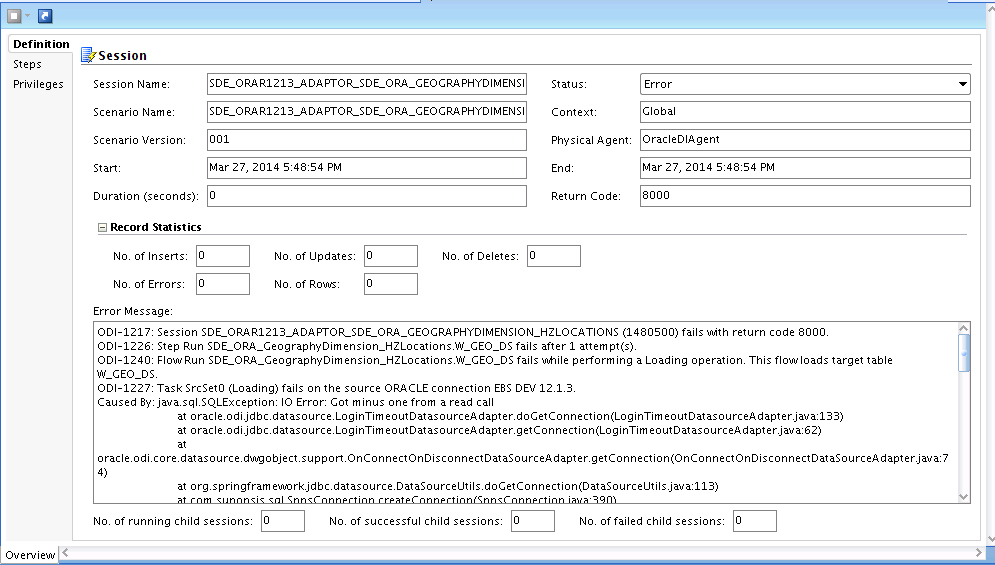
During a full load session of ODI using Oracle EBS as a source, following are the errors occurred, after looking to different articles and BLOGs, we were unable to find a proper solution.

One of the work around is to delete the duplicate rows cause failure and re-start the Execution Plan. We are doing this workaround since we install OBI in MBC.

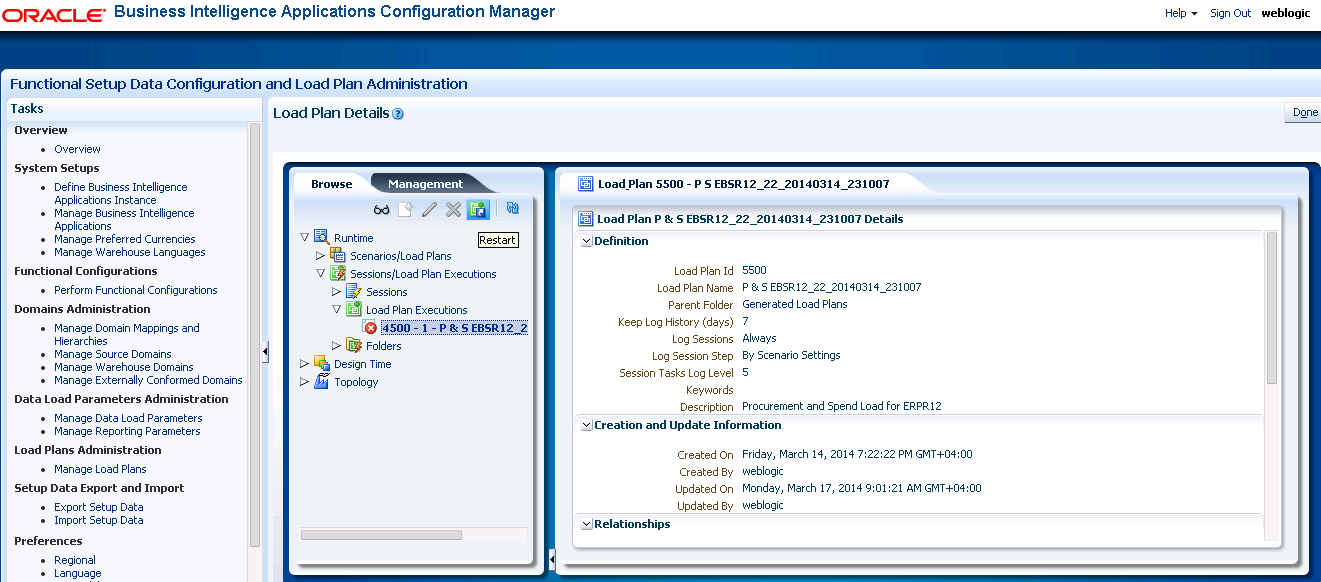
The other work around is to enable the split table configuration. It will create another table with the same structure but will store only duplicate rows whenever it occur and throw no error during full load cycle. This solution is good as far as we know that as first solution we are deleting these duplicate records, in this solution instead of deleting those duplicate records it will be store as separate split table.

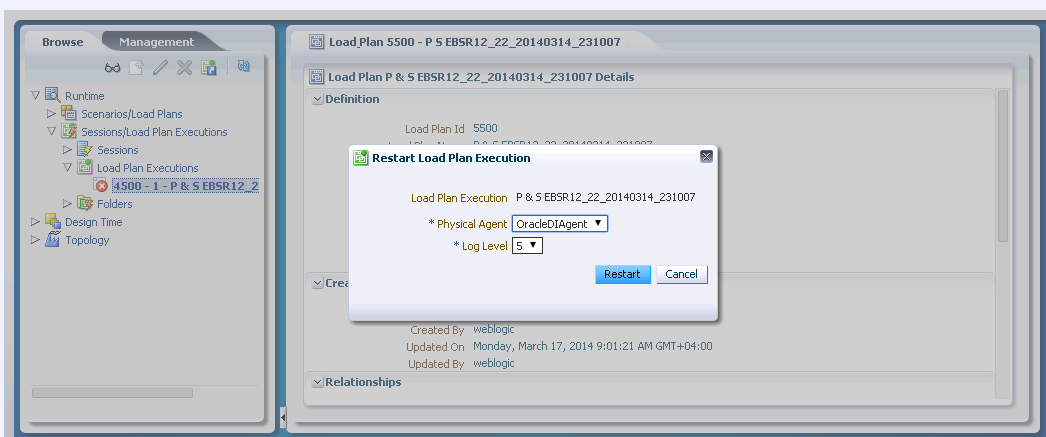
# Got minus one from a read call





Launching the Load Again from BIACM



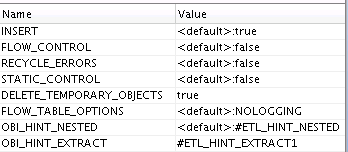


# Full load Execution Fails

When a full load runs , it fails on different task with duplicate index keys insert in the tables exception, after searching of articles and blogs , there are two work around found.

## 3.1 Solution 1

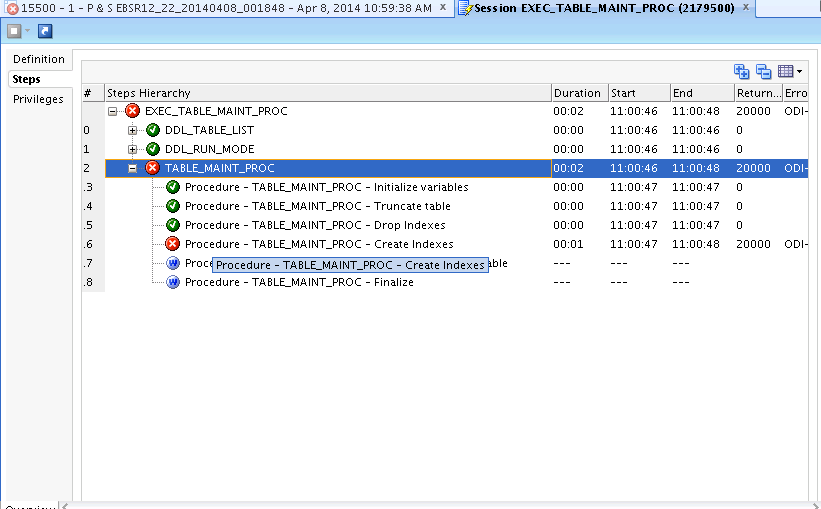
In ODI Studio, open the interface causing the issue, go to the flow tab, select the target data store, and in the KM options that will appear in the property inspector set FLOW\_CONTROL to “true”. Save and generate the scenario. Next run full load execution again, any records that violate constraints will be set aside in an E$\_<tablename> error table, and an overview of the errors can be found in SNP\_CHECK\_TABLE. As in the below snapshot.

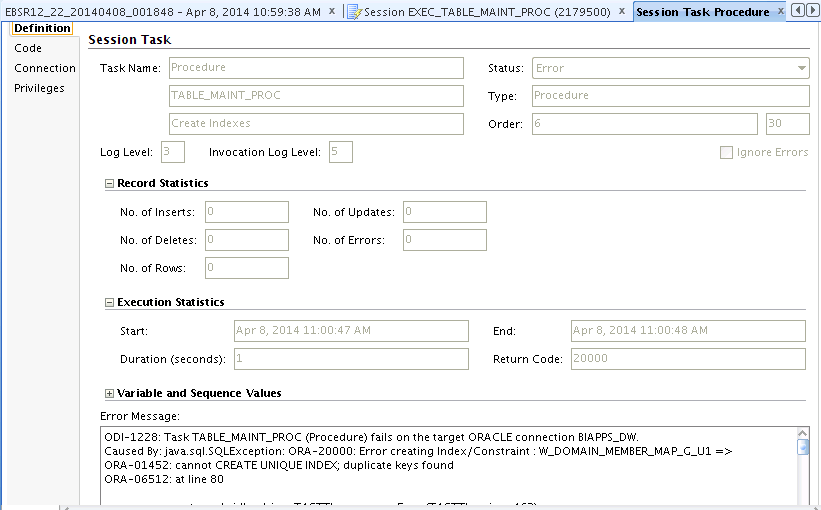


## 3.2 Solution 2

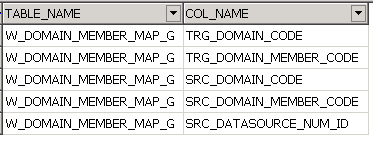
Delete the duplicate records from table and re-start the execution plan which define as below.

### 3.2.1 Duplicates in W\_DOMAIN\_MEMBER\_MAP\_G





Unique indexing defined as below table;



CREATE UNIQUE INDEX PROD\_DW.W\_DOMAIN\_MEMBER\_MAP\_G\_U1 ON PROD\_DW.W\_DOMAIN\_MEMBER\_MAP\_G (SRC\_DOMAIN\_CODE,SRC\_DOMAIN\_MEMBER\_CODE,SRC\_DATASOURCE\_NUM\_ID,TRG\_DOMAIN\_CODE,TRG\_DOMAIN\_MEMBER\_CODE) NOLOGGING;

select count('X'), SRC\_DOMAIN\_CODE,SRC\_DOMAIN\_MEMBER\_CODE,SRC\_DATASOURCE\_NUM\_ID,TRG\_DOMAIN\_CODE,TRG\_DOMAIN\_MEMBER\_CODE

from PROD\_DW. W\_DOMAIN\_MEMBER\_MAP\_G

group by SRC\_DOMAIN\_CODE,SRC\_DOMAIN\_MEMBER\_CODE,SRC\_DATASOURCE\_NUM\_ID,TRG\_DOMAIN\_CODE,TRG\_DOMAIN\_MEMBER\_CODE

having count('X') > 1;

**--+500 registers!!**

CREATE TABLE XX\_W\_DOM\_MEMB\_MAP\_G\_20140327 as select \* from W\_DOMAIN\_MEMBER\_MAP\_G;

DECLARE

l\_count NUMBER;

BEGIN

***DBMS\_OUTPUT.put\_line*** ('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

for jj in (select count('X'), SRC\_DOMAIN\_CODE,SRC\_DOMAIN\_MEMBER\_CODE,SRC\_DATASOURCE\_NUM\_ID,TRG\_DOMAIN\_CODE,TRG\_DOMAIN\_MEMBER\_CODE

from PROD\_DW.W\_DOMAIN\_MEMBER\_MAP\_G

group by SRC\_DOMAIN\_CODE,SRC\_DOMAIN\_MEMBER\_CODE,SRC\_DATASOURCE\_NUM\_ID,TRG\_DOMAIN\_CODE,TRG\_DOMAIN\_MEMBER\_CODE

having count('X') > 1) loop

FOR i IN (SELECT emp.ROWID ROW\_ID,emp.\*

FROM PROD\_DW.W\_DOMAIN\_MEMBER\_MAP\_G emp

where SRC\_DOMAIN\_CODE = jj. SRC\_DOMAIN\_CODE

AND SRC\_DOMAIN\_MEMBER\_CODE = jj. SRC\_DOMAIN\_MEMBER\_CODE

AND SRC\_DATASOURCE\_NUM\_ID = jj. SRC\_DATASOURCE\_NUM\_ID

AND TRG\_DOMAIN\_CODE = jj. TRG\_DOMAIN\_CODE

AND TRG\_DOMAIN\_MEMBER\_CODE = jj.TRG\_DOMAIN\_MEMBER\_CODE

)

LOOP

l\_count := 0;

SELECT COUNT ('X')

INTO l\_count

FROM PROD\_DW. W\_DOMAIN\_MEMBER\_MAP\_G

WHERE SRC\_DOMAIN\_CODE = i. SRC\_DOMAIN\_CODE

AND SRC\_DOMAIN\_MEMBER\_CODE = i. SRC\_DOMAIN\_MEMBER\_CODE

AND SRC\_DATASOURCE\_NUM\_ID = i. SRC\_DATASOURCE\_NUM\_ID

AND TRG\_DOMAIN\_CODE = i. TRG\_DOMAIN\_CODE

AND TRG\_DOMAIN\_MEMBER\_CODE = i.TRG\_DOMAIN\_MEMBER\_CODE

group by SRC\_DOMAIN\_CODE,SRC\_DOMAIN\_MEMBER\_CODE,SRC\_DATASOURCE\_NUM\_ID,TRG\_DOMAIN\_CODE,TRG\_DOMAIN\_MEMBER\_CODE ;

IF l\_count >1

THEN

DELETE FROM PROD\_DW. W\_DOMAIN\_MEMBER\_MAP\_G WHERE ROWID = i.ROW\_ID;

l\_count := 0;

***DBMS\_OUTPUT.put\_line*** (' SRC\_DOMAIN\_CODE ->'||i. SRC\_DOMAIN\_CODE || ' SRC\_DOMAIN\_MEMBER\_CODE ->' ||i. SRC\_DOMAIN\_MEMBER\_CODE || ' SRC\_DATASOURCE\_NUM\_ID ->'

||i. SRC\_DATASOURCE\_NUM\_ID || ' TRG\_DOMAIN\_CODE -> '||i. TRG\_DOMAIN\_CODE

|| ' TRG\_DOMAIN\_MEMBER\_CODE -> '||i.TRG\_DOMAIN\_MEMBER\_CODE);

*-- COMMIT;*

END IF;

END LOOP;

end loop;

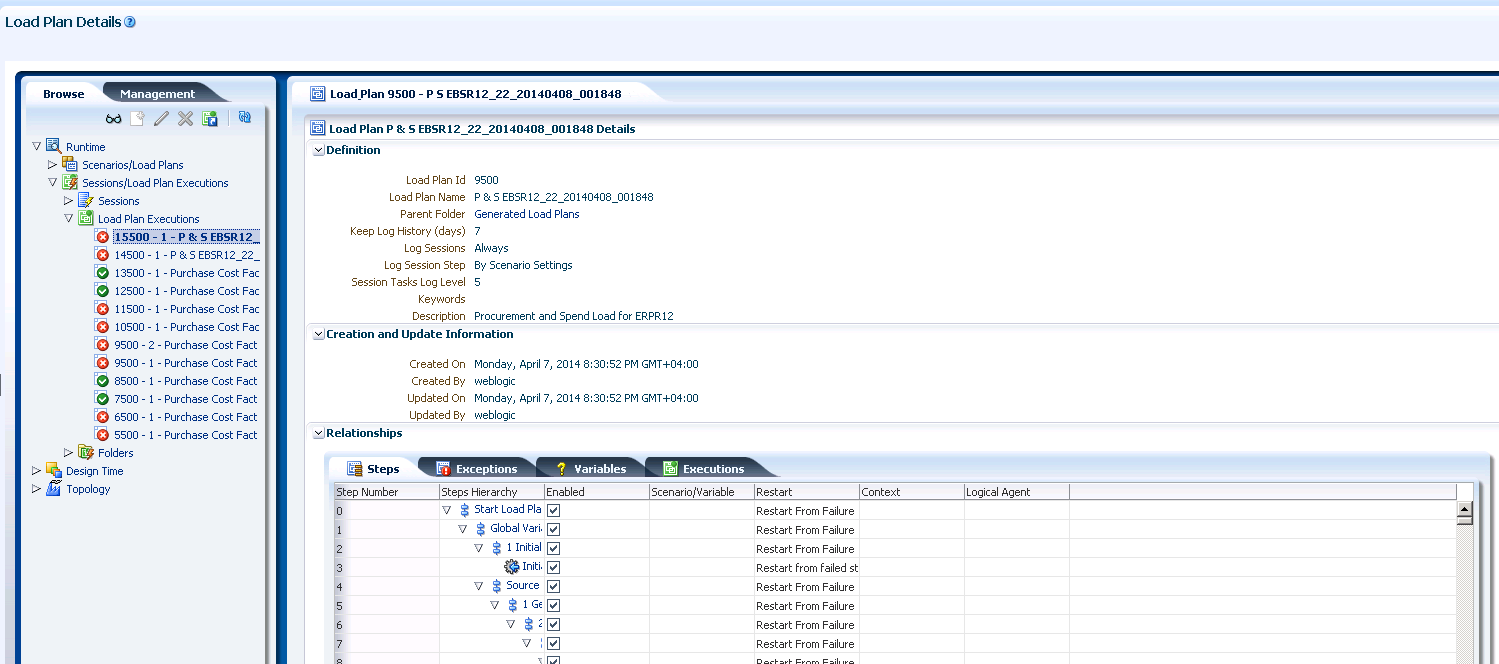
***DBMS\_OUTPUT.put\_line*** ('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

END;

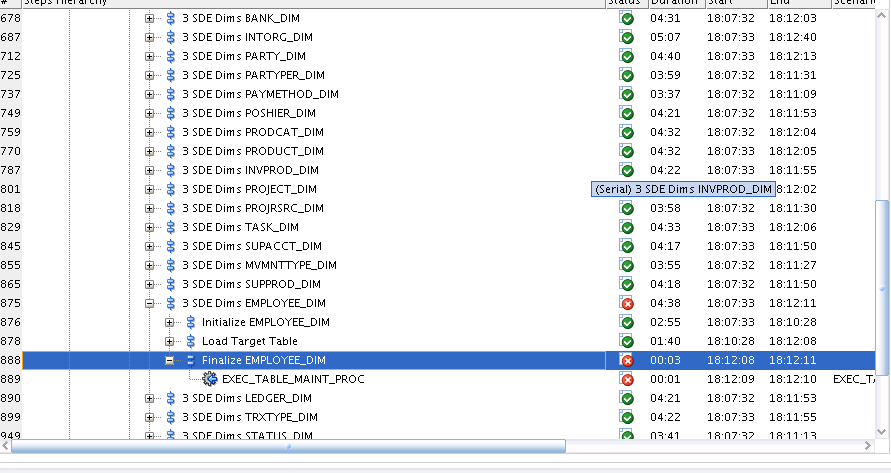
The DBMS output contains more than 3000 records

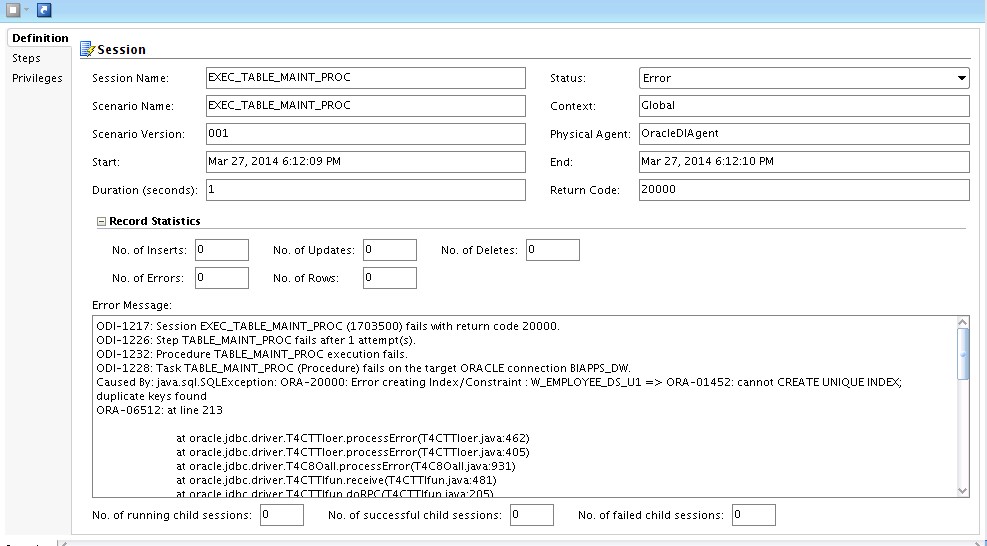


COMMIT;

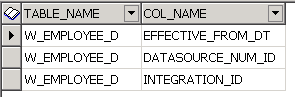


### 3.2.2 Duplicates in W\_EMPLOYEE\_DS





Unique index defined as following table.



select count('X'),INTEGRATION\_ID,DATASOURCE\_NUM\_ID,SRC\_EFF\_FROM\_DT f

from PROD\_DW.W\_EMPLOYEE\_DS group by INTEGRATION\_ID,DATASOURCE\_NUM\_ID,SRC\_EFF\_FROM\_DT having count('X') > 1;

COUNT('X') INTEGRATION\_ID DATASOURCE\_NUM\_ID F

2 PER~1478 1001 10/11/2009

2 PER~1204 1001 1/1/2012

2 PER~29752 1001 3/23/2012

2 PER~35143 1001 3/1/2014

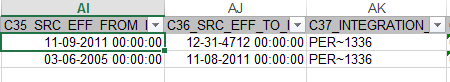
2 PER~768 1001 12/26/2011

2 PER~27411 1001 10/13/2011

2 PER~1336 1001 11/9/2011

2 PER~29752 1001 11/17/2011

One of the case from ODI generated query, return following results for PER~1336

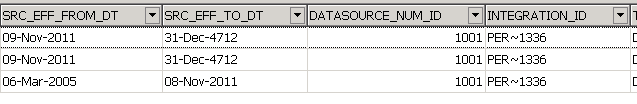


Which is only 2 records with 2 different SRC\_EFF\_FROM\_DT and SRC\_EFF\_TO\_DT,

While the unique Indexing key is set for only

Integration\_ID, DataSource\_Num\_Id and SRC\_EFF\_FROM\_DATE

Also it is found that the staging W\_EMPLOYEE\_DS table contains more than 2 records, as



CREATE TABLE XXW\_EMPLOYEE\_DS\_BAK20140327 as select \* from W\_EMPLOYEE\_DS;

---Script----------

DECLARE

l\_count NUMBER;

BEGIN

***DBMS\_OUTPUT.put\_line*** ('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

for jj in (select count('X'),INTEGRATION\_ID,DATASOURCE\_NUM\_ID,SRC\_EFF\_FROM\_DT from PROD\_DW.W\_EMPLOYEE\_DS group by INTEGRATION\_ID,DATASOURCE\_NUM\_ID,SRC\_EFF\_FROM\_DT having count('X') > 1) loop

FOR i IN (SELECT emp.ROWID ROW\_ID,emp.\* FROM PROD\_DW.W\_EMPLOYEE\_DS emp where INTEGRATION\_ID = jj.INTEGRATION\_ID

AND DATASOURCE\_NUM\_ID = jj.DATASOURCE\_NUM\_ID

AND SRC\_EFF\_FROM\_DT = jj.SRC\_EFF\_FROM\_DT )

LOOP

l\_count := 0;

SELECT COUNT ('X')

INTO l\_count

FROM PROD\_DW.W\_EMPLOYEE\_DS

WHERE INTEGRATION\_ID = i.INTEGRATION\_ID

AND DATASOURCE\_NUM\_ID = i.DATASOURCE\_NUM\_ID

AND SRC\_EFF\_FROM\_DT = i.SRC\_EFF\_FROM\_DT

group by INTEGRATION\_ID,DATASOURCE\_NUM\_ID,SRC\_EFF\_FROM\_DT;

IF l\_count >1

THEN

DELETE FROM PROD\_DW.W\_EMPLOYEE\_DS WHERE ROWID = i.ROW\_ID;

l\_count := 0;

***DBMS\_OUTPUT.put\_line*** ('FULL\_NAME->'||i.FULL\_NAME || ' INTEGRATION\_ID ->' ||i.INTEGRATION\_ID || ' DATASOURCE\_NUM\_ID->'||i.DATASOURCE\_NUM\_ID || ' SRC\_EFF\_FROM\_DT-> '||i.SRC\_EFF\_FROM\_DT);

*-- COMMIT;*

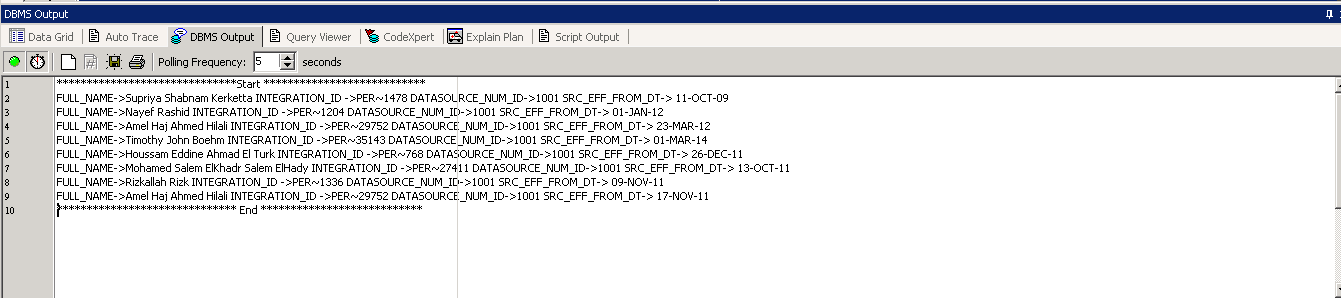
END IF;

END LOOP;

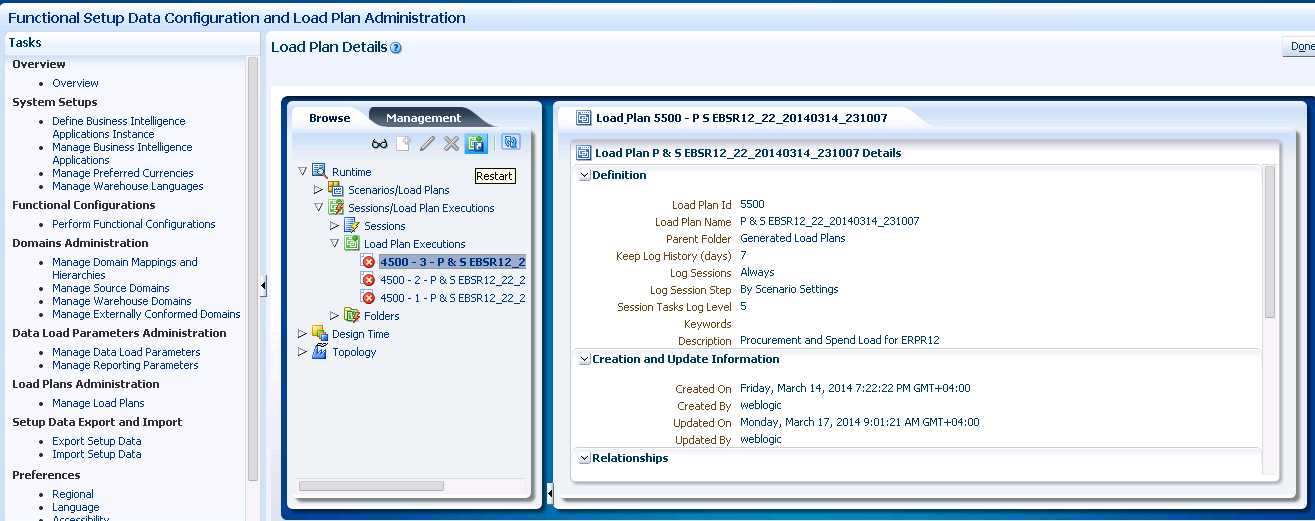
end loop;

***DBMS\_OUTPUT.put\_line*** ('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

END;

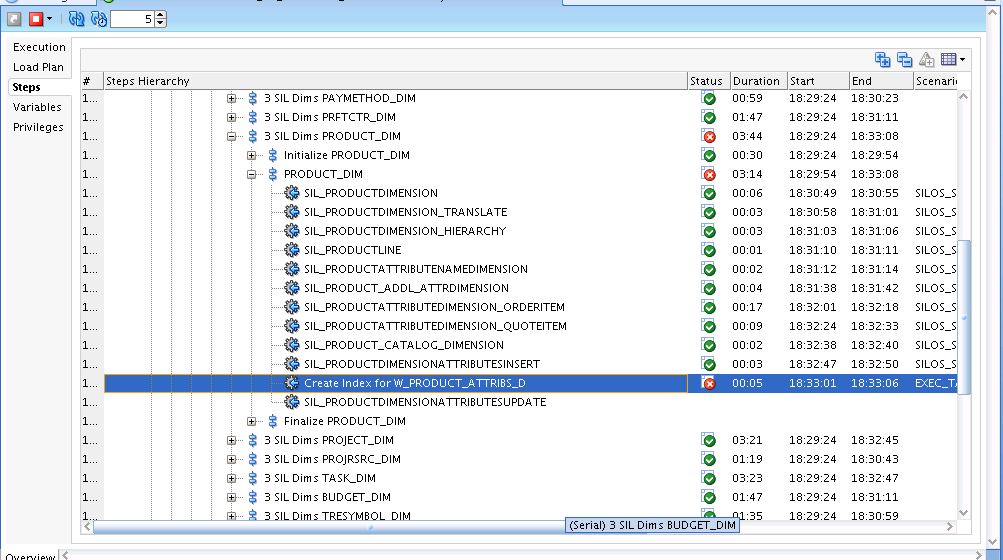


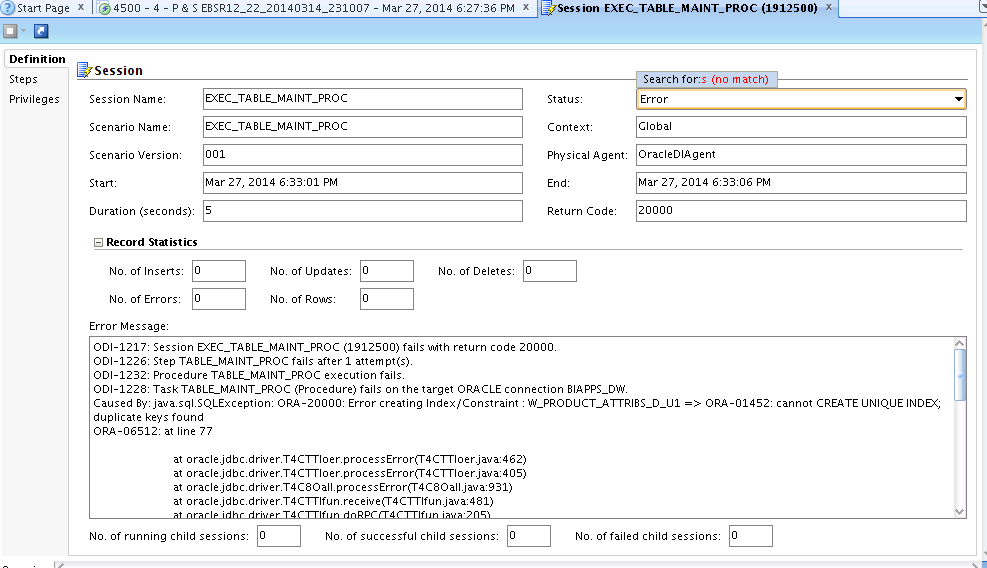
COMMIT;



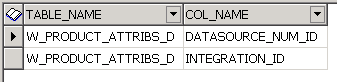
### 3.2.3 Duplicates in W\_PRODUCT\_ATTRIBS\_D

After 7 minutes





Unique index defined as following table



EXECUTE IMMEDIATE 'CREATE UNIQUE INDEX PROD\_DW.W\_PRODUCT\_ATTRIBS\_D\_U1 ON PROD\_DW.W\_PRODUCT\_ATTRIBS\_D (INTEGRATION\_ID,DATASOURCE\_NUM\_ID) NOLOGGING ';

select count('X'),INTEGRATION\_ID,DATASOURCE\_NUM\_ID from PROD\_DW.W\_PRODUCT\_ATTRIBS\_D group by INTEGRATION\_ID,DATASOURCE\_NUM\_ID having count('X') > 1;

COUNT('X') INTEGRATION\_ID DATASOURCE\_NUM\_ID

2 4 1001

2 4042 1001

2 5 1001

2 4045 1001

2 7 1001

2 4043 1001

2 3 1001

2 4044 1001

2 2 1001

2 6 1001

2 4046 1001

CREATE TABLE PROD\_DW.XXW\_PRODUCT\_ATTRIBS\_D\_BAK0327 as select \* from PROD\_DW.W\_PRODUCT\_ATTRIBS\_D;

DECLARE

l\_count NUMBER;

BEGIN

DBMS\_OUTPUT.put\_line ('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

for jj in (select count('X'),INTEGRATION\_ID,DATASOURCE\_NUM\_ID from PROD\_DW.W\_PRODUCT\_ATTRIBS\_D group by INTEGRATION\_ID,DATASOURCE\_NUM\_ID having count('X') > 1) loop

FOR i IN (SELECT emp.ROWID ROW\_ID,emp.\* from PROD\_DW.W\_PRODUCT\_ATTRIBS\_D emp where INTEGRATION\_ID = jj.INTEGRATION\_ID

AND DATASOURCE\_NUM\_ID = jj.DATASOURCE\_NUM\_ID)

LOOP

l\_count := 0;

SELECT COUNT ('X')

INTO l\_count

FROM PROD\_DW.W\_PRODUCT\_ATTRIBS\_D

WHERE INTEGRATION\_ID = i.INTEGRATION\_ID

AND DATASOURCE\_NUM\_ID = i.DATASOURCE\_NUM\_ID

group by INTEGRATION\_ID,DATASOURCE\_NUM\_ID;

IF l\_count > 1

THEN

DELETE FROM PROD\_DW.W\_PRODUCT\_ATTRIBS\_D

WHERE ROWID = i.ROW\_ID;

l\_count := 0;

***DBMS\_OUTPUT.put\_line*** ('DATASOURCE\_NUM\_ID : '||i.DATASOURCE\_NUM\_ID ||' INTEGRATION\_ID : '||i.INTEGRATION\_ID);

END IF;

END LOOP;

end loop;

***DBMS\_OUTPUT.put\_line*** ('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

END;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : 4

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : 4042

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : 5

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : 4045

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : 7

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : 4043

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : 3

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : 4044

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : 2

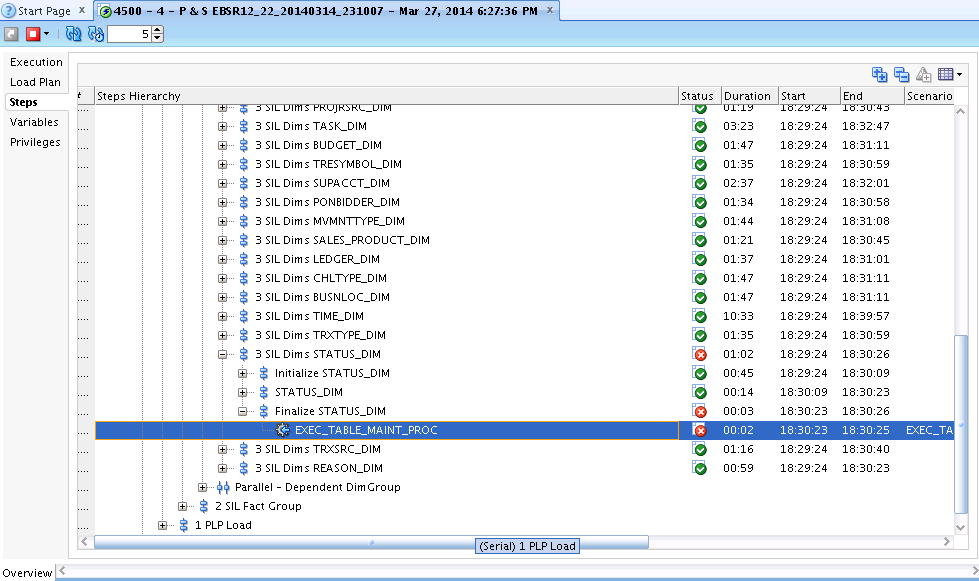
DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : 6

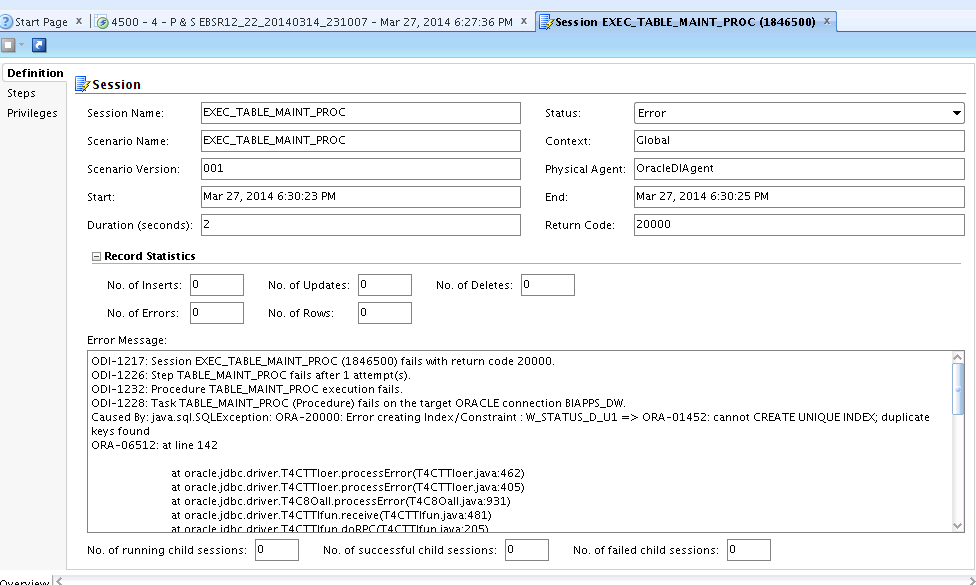
DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : 4046

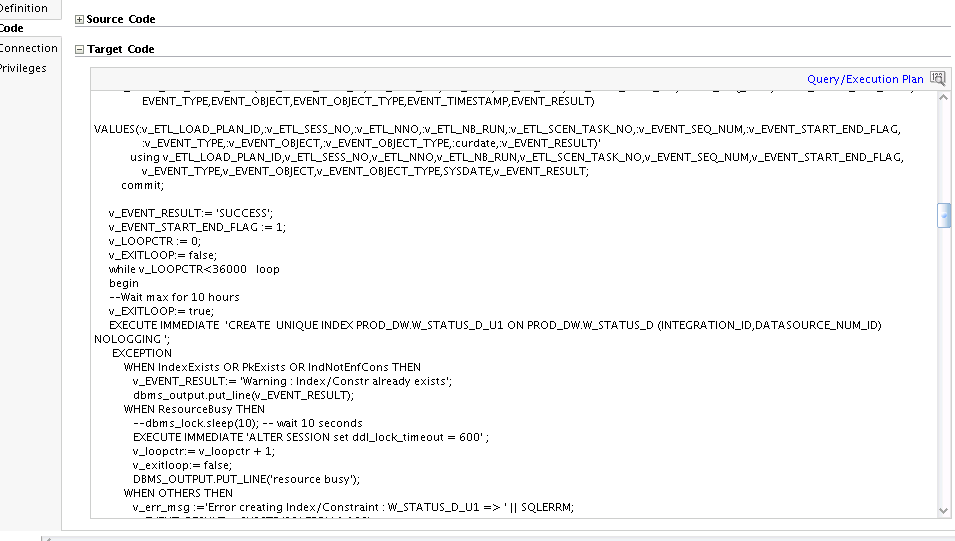
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 3.2.4 Duplicates in W\_STATUS\_D

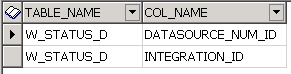
1 minute







Unique index defined as following table



select count('X'),INTEGRATION\_ID,DATASOURCE\_NUM\_ID from PROD\_DW.W\_STATUS\_D group by INTEGRATION\_ID,DATASOURCE\_NUM\_ID having count('X') > 1;

COUNT('X') INTEGRATION\_ID DATASOURCE\_NUM\_ID

2 EXPENSE\_PENDING\_STATUS~MGRAPPR 1001

2 EXPENSE\_PENDING\_STATUS~SUBMITTED 1001

2 EXPENSE\_PENDING\_STATUS~SAVED 1001

2 EXPENSE\_PENDING\_STATUS~PARPAID 1001

2 EXPENSE\_REPORT~PARPAID 1001

2 EXPENSE\_PENDING\_STATUS~MGRPAYAPPR 1001

2 EXPENSE\_REPORT~PAID 1001

2 EXPENSE\_REPORT~REJECTED 1001

2 EXPENSE\_PENDING\_STATUS~INVOICED 1001

2 EXPENSE\_PENDING\_STATUS~PAYAPPR 1001

2 EXPENSE\_REPORT~MGRAPPR 1001

2 EXPENSE\_PENDING\_STATUS~PAID 1001

2 EXPENSE\_PENDING\_STATUS~REJECTED 1001

2 EXPENSE\_REPORT~SUBMITTED 1001

2 EXPENSE\_REPORT~PAYAPPR 1001

2 EXPENSE\_REPORT~SAVED 1001

2 EXPENSE\_REPORT~INVOICED 1001

2 EXPENSE\_REPORT~MGRPAYAPPR 1001

CREATE TABLE XXW\_STATUS\_D\_BAK20140327 as select \* from W\_STATUS\_D;

DECLARE

l\_count NUMBER;

BEGIN

DBMS\_OUTPUT.put\_line ('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

for jj in (select count('X'),INTEGRATION\_ID,DATASOURCE\_NUM\_ID from PROD\_DW.W\_STATUS\_D group by INTEGRATION\_ID,DATASOURCE\_NUM\_ID having count('X') > 1) loop

FOR i IN (SELECT emp.ROWID ROW\_ID,emp.\* from PROD\_DW.W\_STATUS\_D emp where INTEGRATION\_ID = jj.INTEGRATION\_ID

AND DATASOURCE\_NUM\_ID = jj.DATASOURCE\_NUM\_ID)

LOOP

l\_count := 0;

SELECT COUNT ('X')

INTO l\_count

FROM PROD\_DW.W\_STATUS\_D

WHERE INTEGRATION\_ID = i.INTEGRATION\_ID

AND DATASOURCE\_NUM\_ID = i.DATASOURCE\_NUM\_ID

group by INTEGRATION\_ID,DATASOURCE\_NUM\_ID;

IF l\_count > 1

THEN

DELETE FROM PROD\_DW.W\_STATUS\_D

WHERE ROWID = i.ROW\_ID;

l\_count := 0;

***DBMS\_OUTPUT.put\_line*** ('DATASOURCE\_NUM\_ID : '||i.DATASOURCE\_NUM\_ID ||' INTEGRATION\_ID : '||i.INTEGRATION\_ID);

END IF;

END LOOP;

END LOOP;

***DBMS\_OUTPUT.put\_line*** ('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

*-- COMMIT;*

END;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_PENDING\_STATUS~MGRAPPR

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_PENDING\_STATUS~SUBMITTED

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_PENDING\_STATUS~SAVED

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_PENDING\_STATUS~PARPAID

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_REPORT~PARPAID

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_PENDING\_STATUS~MGRPAYAPPR

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_REPORT~PAID

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_REPORT~REJECTED

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_PENDING\_STATUS~INVOICED

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_PENDING\_STATUS~PAYAPPR

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_REPORT~MGRAPPR

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_PENDING\_STATUS~PAID

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_PENDING\_STATUS~REJECTED

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_REPORT~SUBMITTED

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_REPORT~PAYAPPR

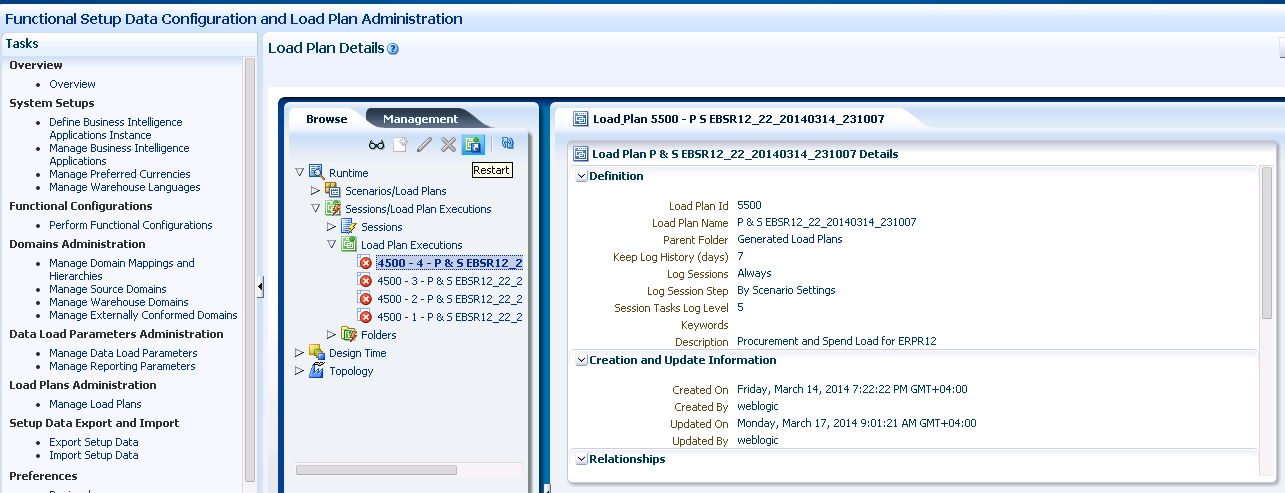
DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_REPORT~SAVED

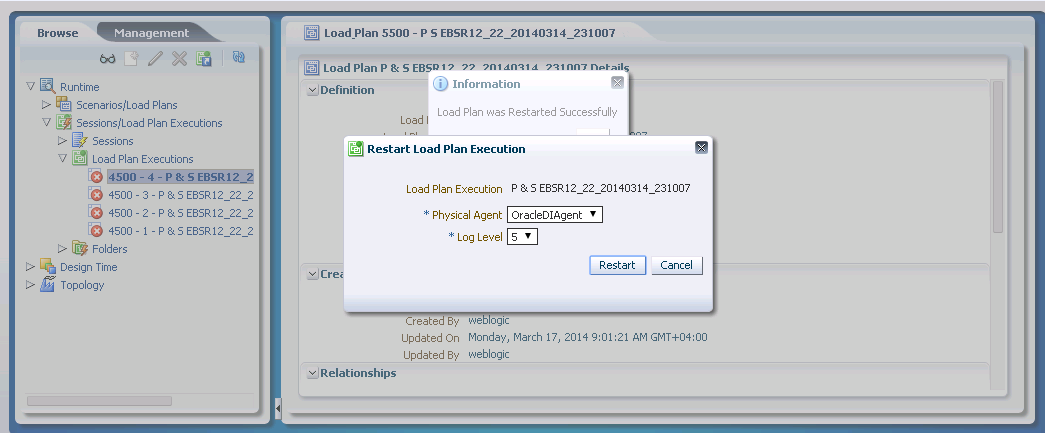
DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_REPORT~INVOICED

DATASOURCE\_NUM\_ID : 1001 INTEGRATION\_ID : EXPENSE\_REPORT~MGRPAYAPPR

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

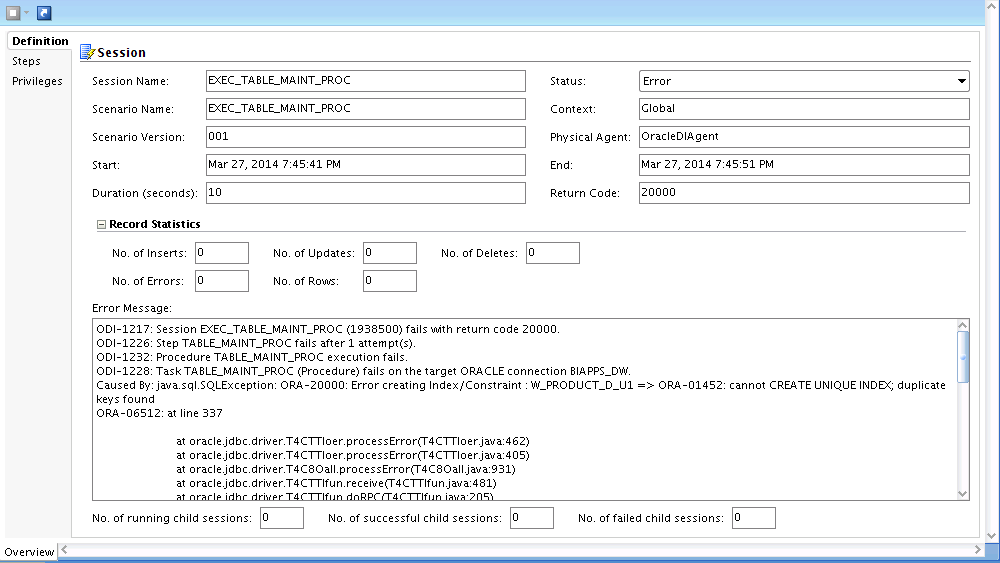
COMMIT;



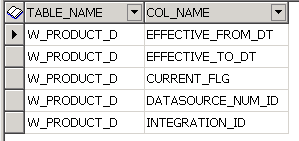


### 3.2.5 Duplicates in W\_PRODUCT\_D

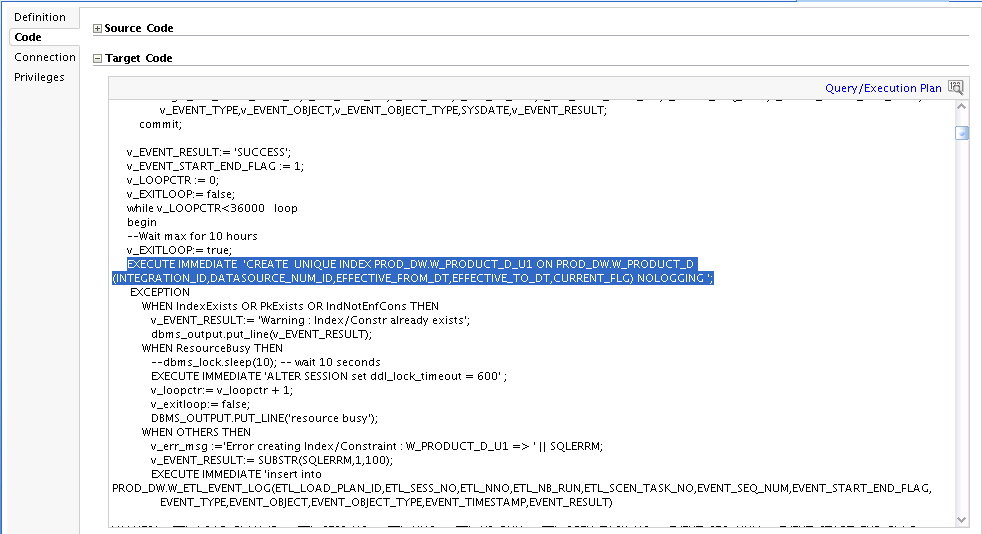
1 minute



Unique index defined as following table



EXECUTE IMMEDIATE ‘CREATE UNWIUE INDEX PROD\_DW.W\_PRODUCT\_D\_U1 ON PROD\_DW.W\_PRODUCT\_D (INTEGRATION\_ID,DATASOURCE\_NUM\_ID,EFFECTIVE\_FROM\_DT,EFFECTIVE\_TO\_DT,CURRENT\_FLG) NOLOGGING;



select count('X'),INTEGRATION\_ID,DATASOURCE\_NUM\_ID,EFFECTIVE\_FROM\_DT,EFFECTIVE\_TO\_DT,CURRENT\_FLG

from PROD\_DW.W\_PRODUCT\_D group by INTEGRATION\_ID,DATASOURCE\_NUM\_ID,EFFECTIVE\_FROM\_DT,EFFECTIVE\_TO\_DT,CURRENT\_FLG having count('X') > 1;

COUNT('X') INTEGRATION\_ID DATASOURCE\_NUM\_ID EFFECTIVE\_FROM\_DT EFFECTIVE\_TO\_DT CURRENT\_FLG

4 4046 1001 1/1/1899 1/1/3714 Y

4 4043 1001 1/1/1899 1/1/3714 Y

4 2 1001 1/1/1899 1/1/3714 Y

4 4044 1001 1/1/1899 1/1/3714 Y

4 4 1001 1/1/1899 1/1/3714 Y

4 4042 1001 1/1/1899 1/1/3714 Y

4 7 1001 1/1/1899 1/1/3714 Y

4 6 1001 1/1/1899 1/1/3714 Y

4 5 1001 1/1/1899 1/1/3714 Y

4 3 1001 1/1/1899 1/1/3714 Y

4 4045 1001 1/1/1899 1/1/3714 Y

CREATE TABLE XXW\_PRODUCT\_D\_BAK20140327 as select \* from W\_PRODUCT\_D;

DECLARE

l\_count NUMBER;

BEGIN

***DBMS\_OUTPUT.put\_line*** ('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

for jj in (select count('X'), INTEGRATION\_ID,DATASOURCE\_NUM\_ID,EFFECTIVE\_FROM\_DT,EFFECTIVE\_TO\_DT,CURRENT\_FLG from

PROD\_DW.W\_PRODUCT\_D

group by INTEGRATION\_ID,DATASOURCE\_NUM\_ID,EFFECTIVE\_FROM\_DT,EFFECTIVE\_TO\_DT,CURRENT\_FLG having count('X') > 1) loop

FOR i IN (SELECT emp.ROWID ROW\_ID,emp.\* FROM PROD\_DW.W\_PRODUCT\_D emp where INTEGRATION\_ID = jj.INTEGRATION\_ID

AND DATASOURCE\_NUM\_ID = jj.DATASOURCE\_NUM\_ID

AND EFFECTIVE\_FROM\_DT = jj.EFFECTIVE\_FROM\_DT

AND EFFECTIVE\_TO\_DT = jj.EFFECTIVE\_TO\_DT

AND CURRENT\_FLG = jj.CURRENT\_FLG)

LOOP

l\_count := 0;

SELECT COUNT ('X')

INTO l\_count

FROM PROD\_DW.W\_PRODUCT\_D

WHERE INTEGRATION\_ID = i.INTEGRATION\_ID

AND DATASOURCE\_NUM\_ID = i.DATASOURCE\_NUM\_ID

AND EFFECTIVE\_FROM\_DT = i.EFFECTIVE\_FROM\_DT

AND EFFECTIVE\_TO\_DT = i.EFFECTIVE\_TO\_DT

AND CURRENT\_FLG = i.CURRENT\_FLG

group by INTEGRATION\_ID,DATASOURCE\_NUM\_ID,EFFECTIVE\_FROM\_DT,EFFECTIVE\_TO\_DT,CURRENT\_FLG;

IF l\_count >1

THEN

DELETE FROM PROD\_DW. W\_PRODUCT\_D WHERE ROWID = i.ROW\_ID;

l\_count := 0;

***DBMS\_OUTPUT.put\_line*** (' INTEGRATION\_ID ->' ||i.INTEGRATION\_ID || ' DATASOURCE\_NUM\_ID->'||i.DATASOURCE\_NUM\_ID || ' EFFECTIVE\_FROM\_DT-> '||i.EFFECTIVE\_FROM\_DT || ' EFFECTIVE\_TO\_DT-> '||i.EFFECTIVE\_TO\_DT|| ' CURRENT\_FLG -> '||i.CURRENT\_FLG);

*-- COMMIT;*

END IF;

END LOOP;

end loop;

***DBMS\_OUTPUT.put\_line*** ('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

END;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

INTEGRATION\_ID ->4046 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->4046 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->4046 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->4043 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->4043 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->4043 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->2 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->2 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->2 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->4044 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->4044 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->4044 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->4 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->4 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->4 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->4042 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->4042 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->4042 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->7 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->7 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->7 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->6 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->6 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->6 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->5 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->5 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->5 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->3 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->3 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->3 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

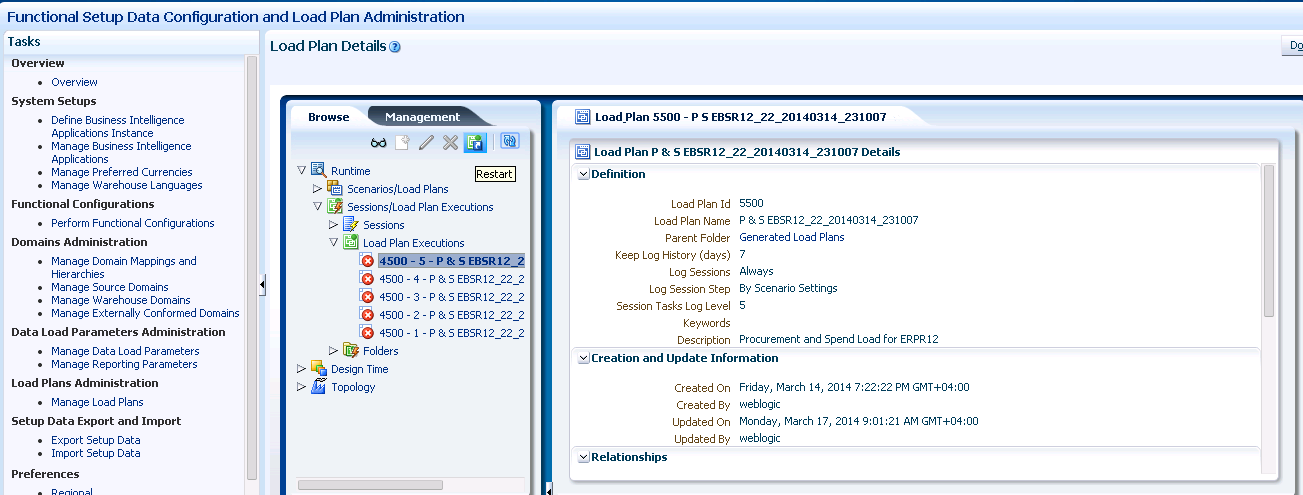
INTEGRATION\_ID ->4045 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

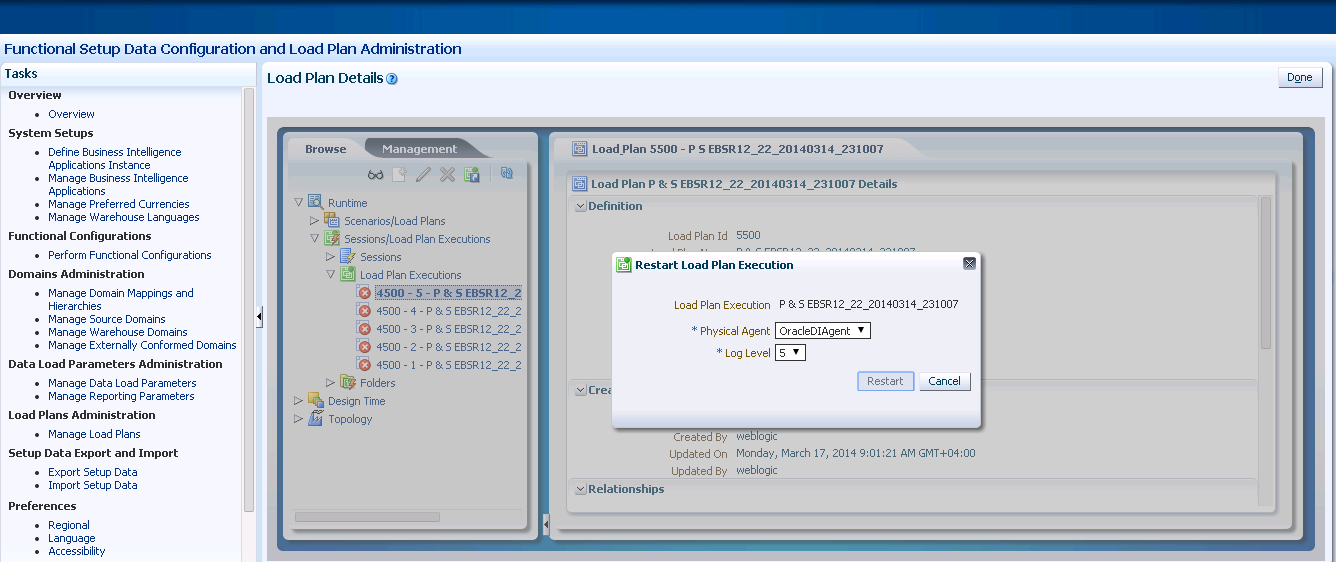
INTEGRATION\_ID ->4045 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

INTEGRATION\_ID ->4045 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14 CURRENT\_FLG -> Y

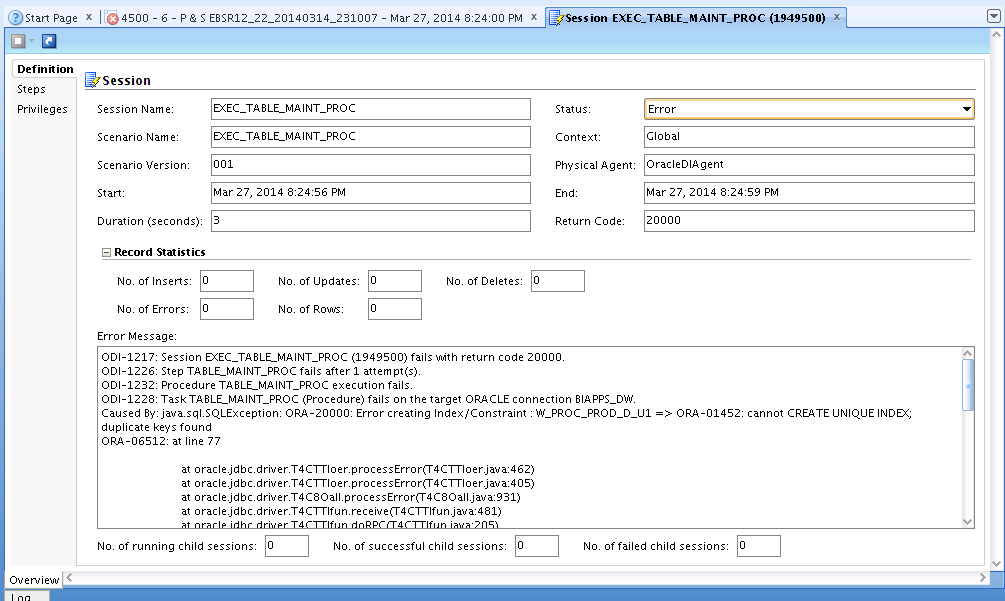
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

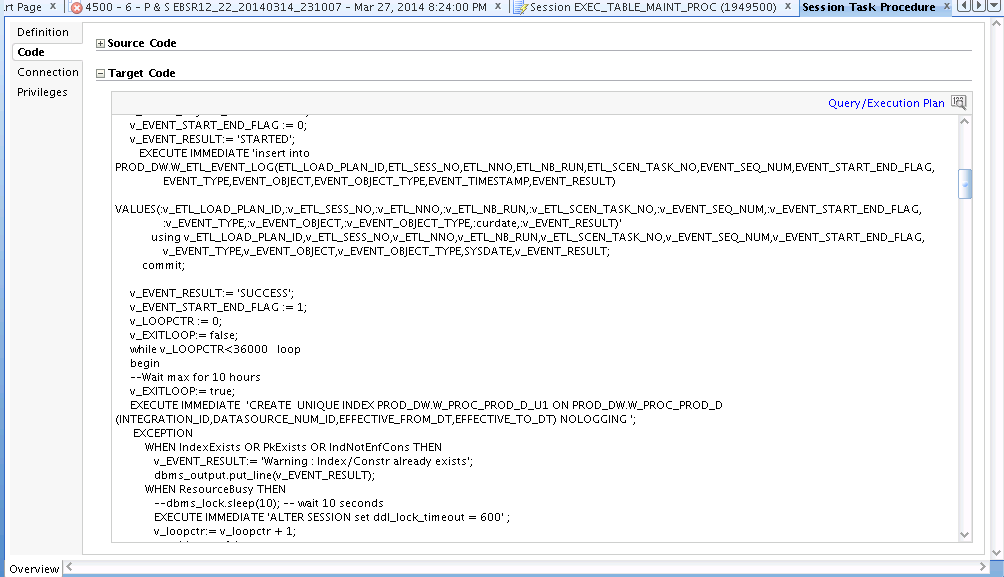
COMMIT;



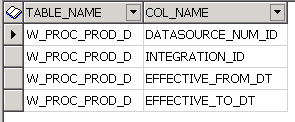


### 3.2.6 Duplicates in W\_PROC\_PROD\_D





Unique index defined as following table



EXECUTE IMMEDIATE 'CREATE UNIQUE INDEX PROD\_DW.W\_PROC\_PROD\_D\_U1 ON PROD\_DW.W\_PROC\_PROD\_D (INTEGRATION\_ID,DATASOURCE\_NUM\_ID,EFFECTIVE\_FROM\_DT,EFFECTIVE\_TO\_DT) NOLOGGING ';

select count('X'),INTEGRATION\_ID,DATASOURCE\_NUM\_ID,EFFECTIVE\_FROM\_DT,EFFECTIVE\_TO\_DT

from PROD\_DW.W\_PROC\_PROD\_D group by INTEGRATION\_ID,DATASOURCE\_NUM\_ID,EFFECTIVE\_FROM\_DT,EFFECTIVE\_TO\_DT having count('X') > 1;

COUNT('X') INTEGRATION\_ID DATASOURCE\_NUM\_ID EFFECTIVE\_FROM\_DT EFFECTIVE\_TO\_DT

2 2d70d39ca6aa7dd5910f6f59be59f9cf 1001 1/1/1899 1/1/3714

2 8b7c84e7c90d5bbeae94e040fd892c29 1001 1/1/1899 1/1/3714

2 8d213884f30a8fa4a572a5eabd75f6a4 1001 1/1/1899 1/1/3714

2 f7c3df1f377f77340e3fa2a85ecc63d0 1001 1/1/1899 1/1/3714

2 a2e4b00a55880903ea798a59f780eb61 1001 1/1/1899 1/1/3714

2 68f5e930cc87ce22e7e38cf22ef6887d 1001 1/1/1899 1/1/3714

2 6b1c8d057fa4f9b38551b0ca1f677fb8 1001 1/1/1899 1/1/3714

2 d15b84513304495da9b5f2fefdcccdcd 1001 1/1/1899 1/1/3714

2 262f3dc1526e4d8315efd65c156d4a55 1001 1/1/1899 1/1/3714

2 01b939fa9cca48f42265fecb1ae910d7 1001 1/1/1899 1/1/3714

2 498c19e9428e18306885a19305d076ab 1001 1/1/1899 1/1/3714

CREATE TABLE XXW\_PROC\_PROD\_D\_BAK20140327 as select \* from W\_PROC\_PROD\_D;

DECLARE

l\_count NUMBER;

BEGIN

***DBMS\_OUTPUT.put\_line*** ('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

for jj in (select count('X'), INTEGRATION\_ID,DATASOURCE\_NUM\_ID,EFFECTIVE\_FROM\_DT,EFFECTIVE\_TO\_DT from

PROD\_DW.W\_PROC\_PROD\_D

group by INTEGRATION\_ID,DATASOURCE\_NUM\_ID,EFFECTIVE\_FROM\_DT,EFFECTIVE\_TO\_DT having count('X') > 1) loop

FOR i IN (SELECT emp.ROWID ROW\_ID,emp.\* FROM PROD\_DW.W\_PROC\_PROD\_D emp where INTEGRATION\_ID = jj.INTEGRATION\_ID

AND DATASOURCE\_NUM\_ID = jj.DATASOURCE\_NUM\_ID

AND EFFECTIVE\_FROM\_DT = jj.EFFECTIVE\_FROM\_DT

AND EFFECTIVE\_TO\_DT = jj.EFFECTIVE\_TO\_DT)

LOOP

l\_count := 0;

SELECT COUNT ('X')

INTO l\_count

FROM PROD\_DW.W\_PROC\_PROD\_D

WHERE INTEGRATION\_ID = i.INTEGRATION\_ID

AND DATASOURCE\_NUM\_ID = i.DATASOURCE\_NUM\_ID

AND EFFECTIVE\_FROM\_DT = i.EFFECTIVE\_FROM\_DT

AND EFFECTIVE\_TO\_DT = i.EFFECTIVE\_TO\_DT

group by INTEGRATION\_ID,DATASOURCE\_NUM\_ID,EFFECTIVE\_FROM\_DT,EFFECTIVE\_TO\_DT,CURRENT\_FLG;

IF l\_count >1

THEN

DELETE FROM PROD\_DW. W\_PROC\_PROD\_D WHERE ROWID = i.ROW\_ID;

l\_count := 0;

***DBMS\_OUTPUT.put\_line*** (' INTEGRATION\_ID ->' ||i.INTEGRATION\_ID || ' DATASOURCE\_NUM\_ID->'||i.DATASOURCE\_NUM\_ID || ' EFFECTIVE\_FROM\_DT-> '||i.EFFECTIVE\_FROM\_DT || ' EFFECTIVE\_TO\_DT-> '||i.EFFECTIVE\_TO\_DT);

*-- COMMIT;*

END IF;

END LOOP;

end loop;

***DBMS\_OUTPUT.put\_line*** ('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

END;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

INTEGRATION\_ID ->2d70d39ca6aa7dd5910f6f59be59f9cf DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14

INTEGRATION\_ID ->8b7c84e7c90d5bbeae94e040fd892c29 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14

INTEGRATION\_ID ->8d213884f30a8fa4a572a5eabd75f6a4 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14

INTEGRATION\_ID ->f7c3df1f377f77340e3fa2a85ecc63d0 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14

INTEGRATION\_ID ->a2e4b00a55880903ea798a59f780eb61 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14

INTEGRATION\_ID ->68f5e930cc87ce22e7e38cf22ef6887d DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14

INTEGRATION\_ID ->6b1c8d057fa4f9b38551b0ca1f677fb8 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14

INTEGRATION\_ID ->d15b84513304495da9b5f2fefdcccdcd DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14

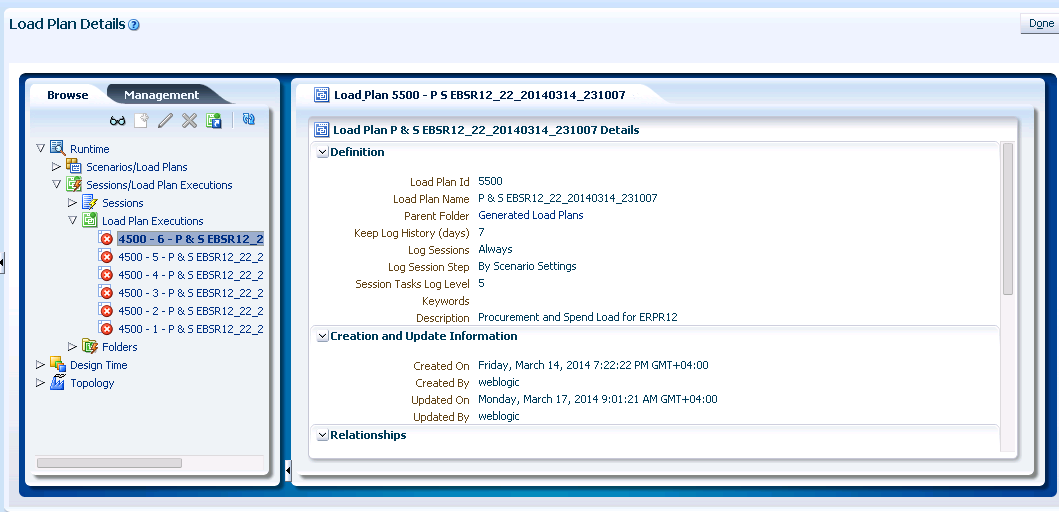
INTEGRATION\_ID ->262f3dc1526e4d8315efd65c156d4a55 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14

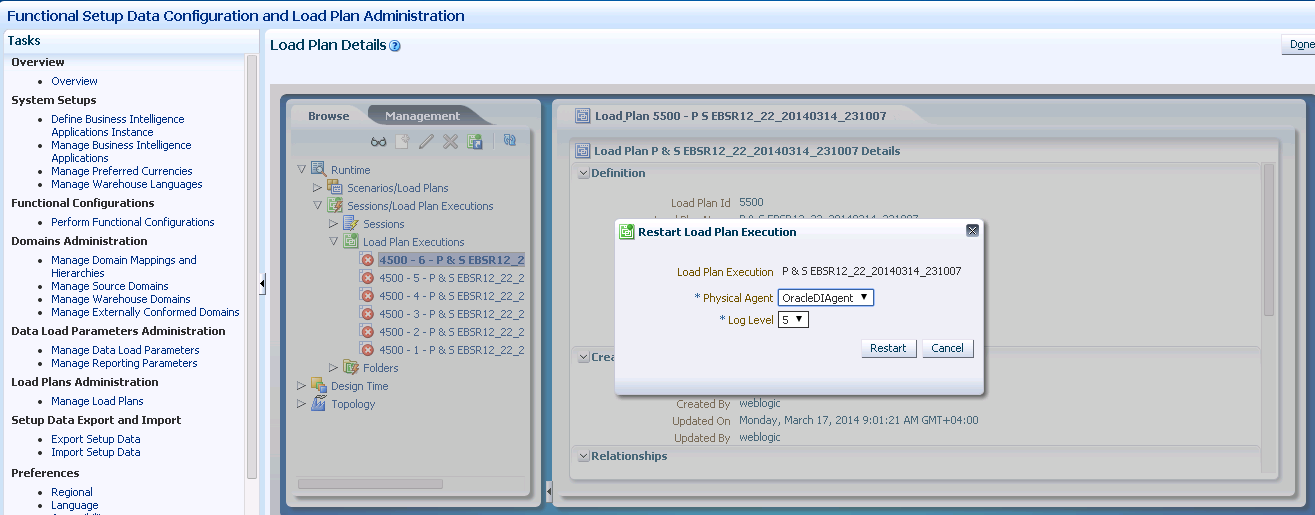
INTEGRATION\_ID ->01b939fa9cca48f42265fecb1ae910d7 DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14

INTEGRATION\_ID ->498c19e9428e18306885a19305d076ab DATASOURCE\_NUM\_ID->1001 EFFECTIVE\_FROM\_DT-> 01-JAN-99 EFFECTIVE\_TO\_DT-> 01-JAN-14

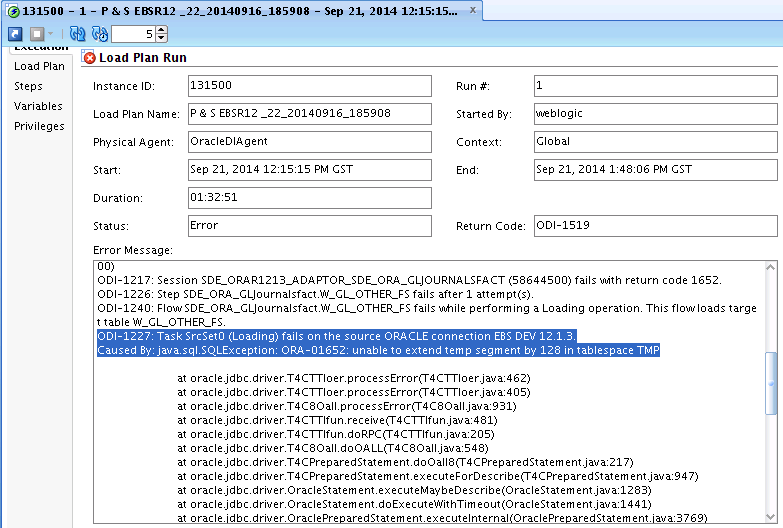
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

COMMIT;





# TEMP Table Space full



When this error/exception occurred, we need to verify the temp table space free space and used and add another temp file into DWH table space. Following are the DDL/DML are required to do this.

SELECT d.status "Status", d.tablespace\_name "Name", d.contents "Type", d.extent\_management "ExtManag",

TO\_CHAR(NVL(a.bytes / 1024 / 1024, 0),'99,999,990.900') "Size (M)", TO\_CHAR(NVL(t.bytes,0)/1024/1024,'99999,999.999') ||'/'||TO\_CHAR(NVL(a.bytes/1024/1024, 0),'99999,999.999') "Used (M)",

TO\_CHAR(NVL(t.bytes / a.bytes \* 100, 0), '990.00') "Used %"

FROM sys.dba\_tablespaces d, (select tablespace\_name, sum(bytes) bytes from dba\_temp\_files group by

tablespace\_name) a,

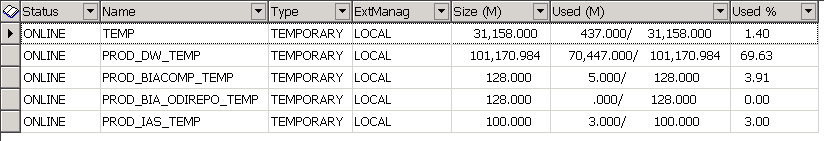
(select tablespace\_name, sum(bytes\_cached) bytes from

v$temp\_extent\_pool group by tablespace\_name) t

WHERE d.tablespace\_name = a.tablespace\_name(+) AND d.tablespace\_name = t.tablespace\_name(+)

AND d.extent\_management like 'LOCAL' AND d.contents like 'TEMPORARY';

This select statement shows the following results with different values.

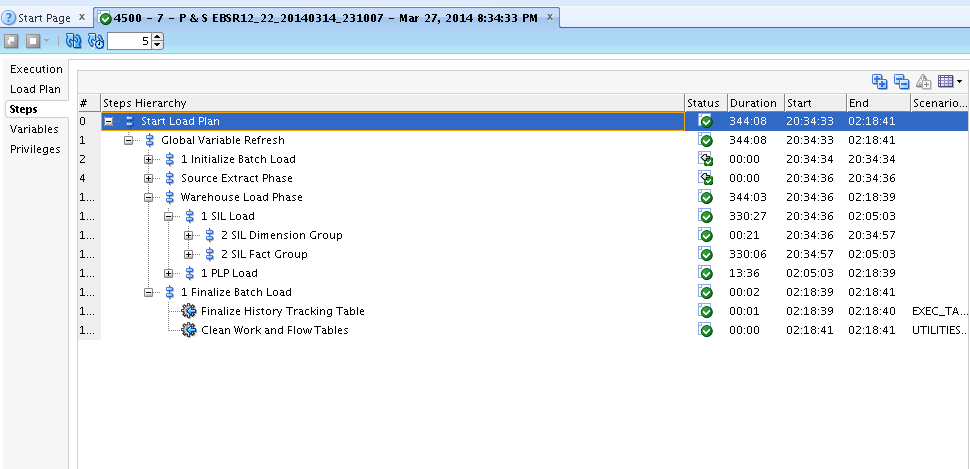


To add new add another table space temp file to database, use the following script.

alter tablespace PROD\_DW\_TEMP add tempfile '/erpsso/BI/db/oradata/bidb/BIDB/**PROD\_dwtemp05.dbf**' size 30G autoextend on;

NOTE: Remember to change the PROD\_dwtemp05.dbf filename, as it is already exist.

# End of Full Load

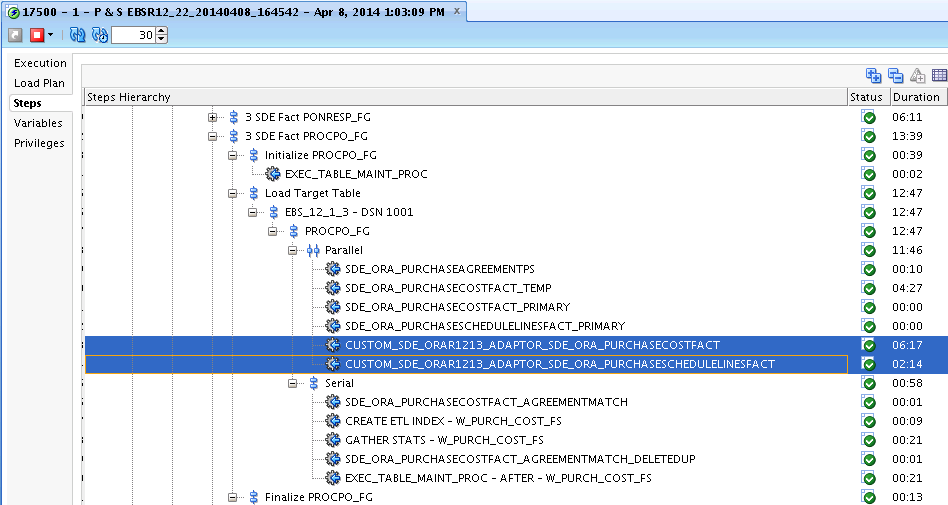


# Checking the execution of the customized tasks

## 6.1 SDE PROCPO Fact Group

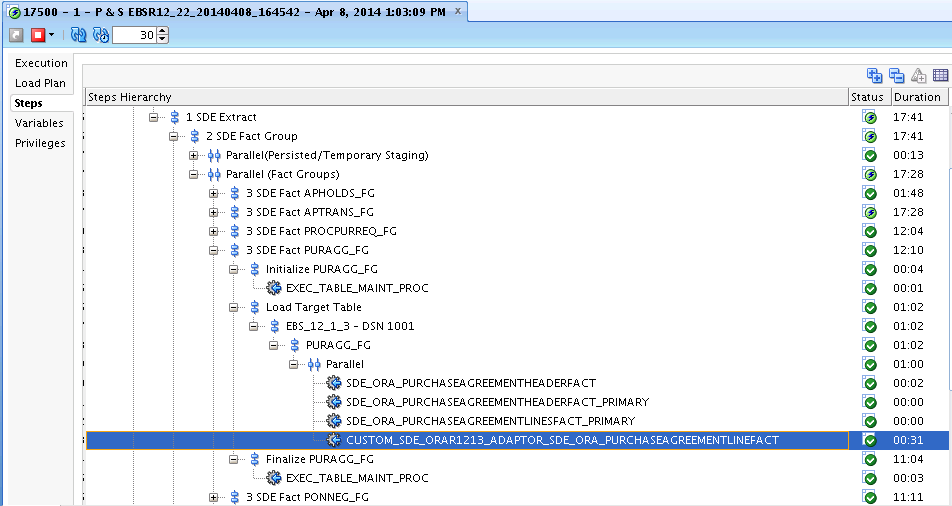
SDE\_ORA\_PurchaseCostFact

SDE\_ORA\_PurchaseScheduleLinesFact



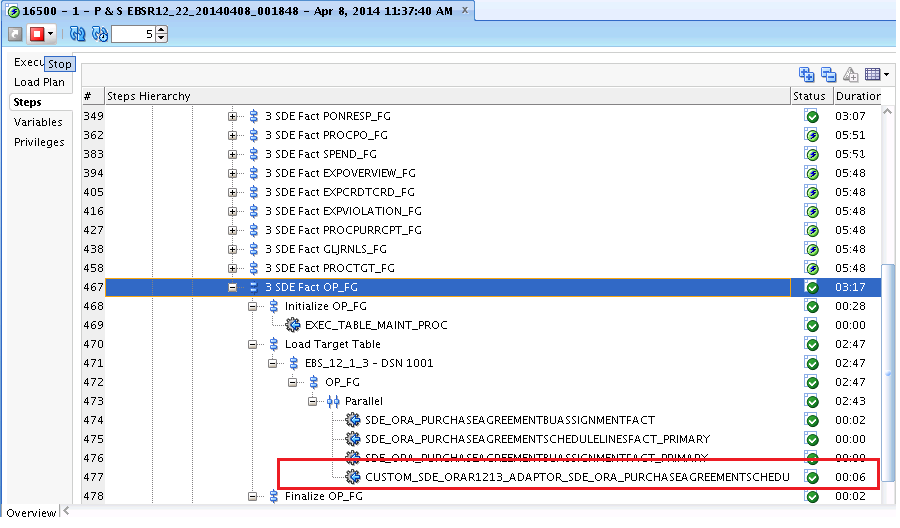
## 6.2 SDE PURAGG Fact Group

SDE\_ORA\_PurchaseAgreementLineFact



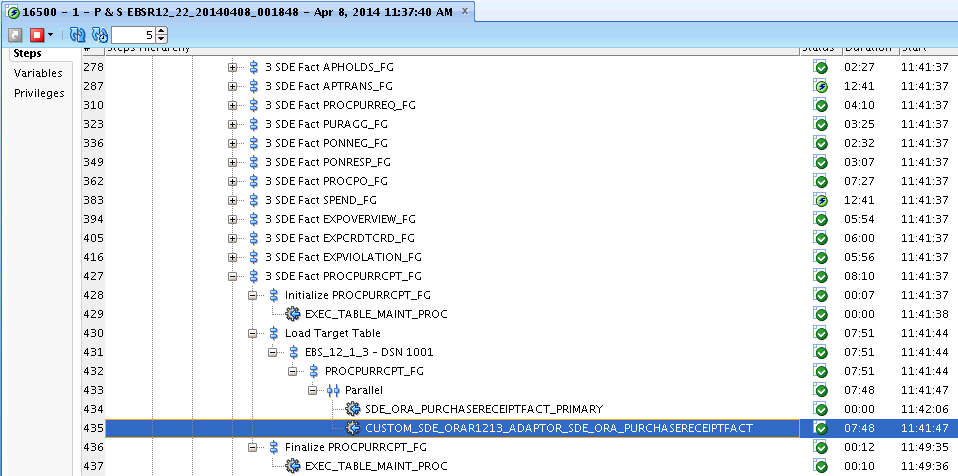
## 6.3 SDE OP Fact Group

SDE\_ORA\_PurchaseAgreementScheduleLinesFact



## 6.4 SDE PROCPURRCPT Fact Group

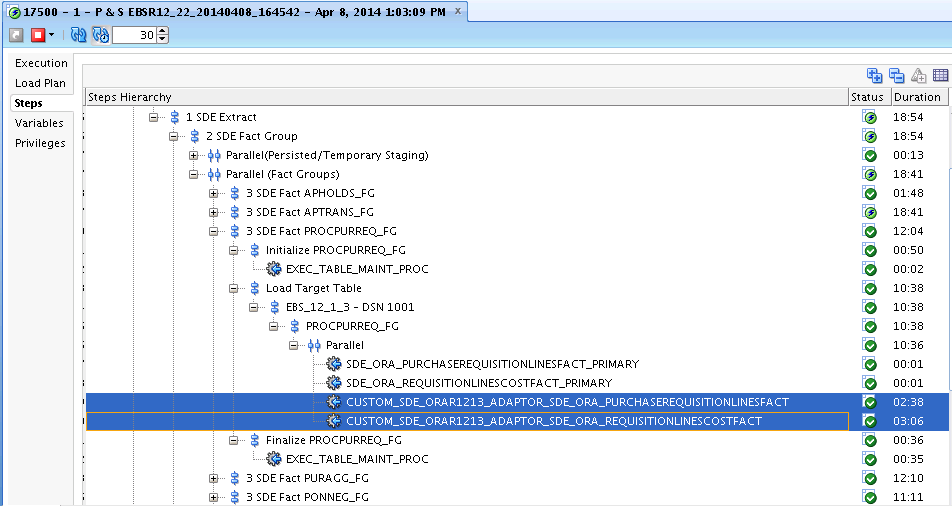
SDE\_ORA\_PurchaseReceiptFact



## 6.5 SDE PROCPURREQ Fact Group

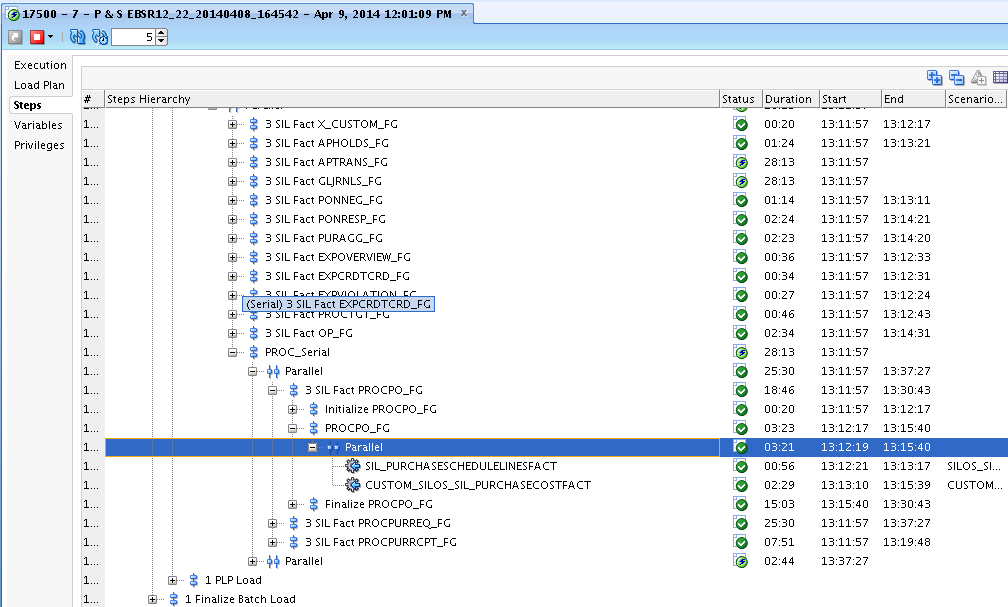
SDE\_ORA\_PurchaseRequisitionLinesFact

SDE\_ORA\_RequisitionLinesCostFact



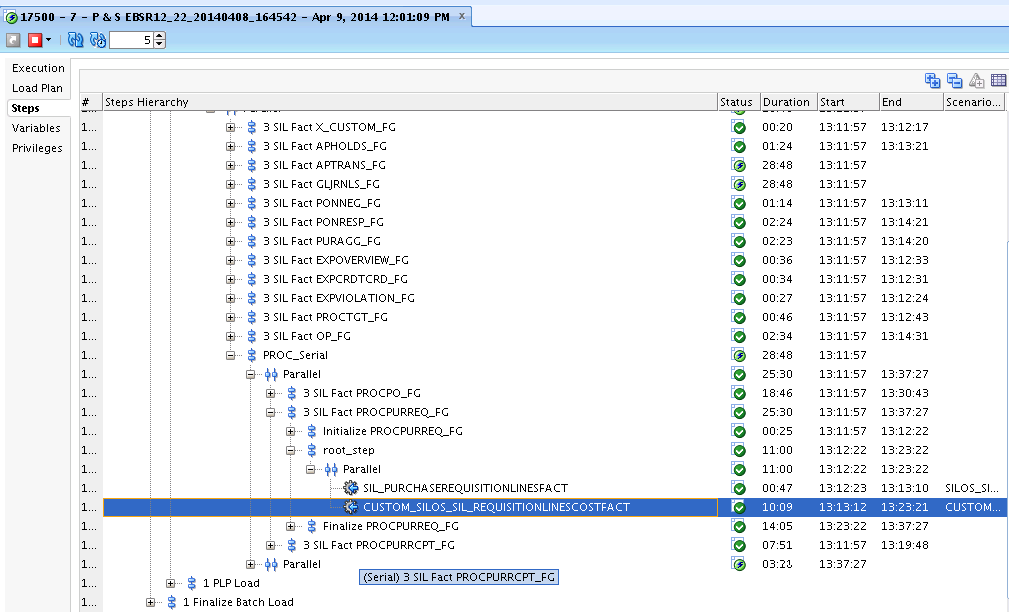
## 6.6 SIL PROCPO Fact Group

SIL\_PurchaseCostFact (Original Promised Date, Savings attributes)



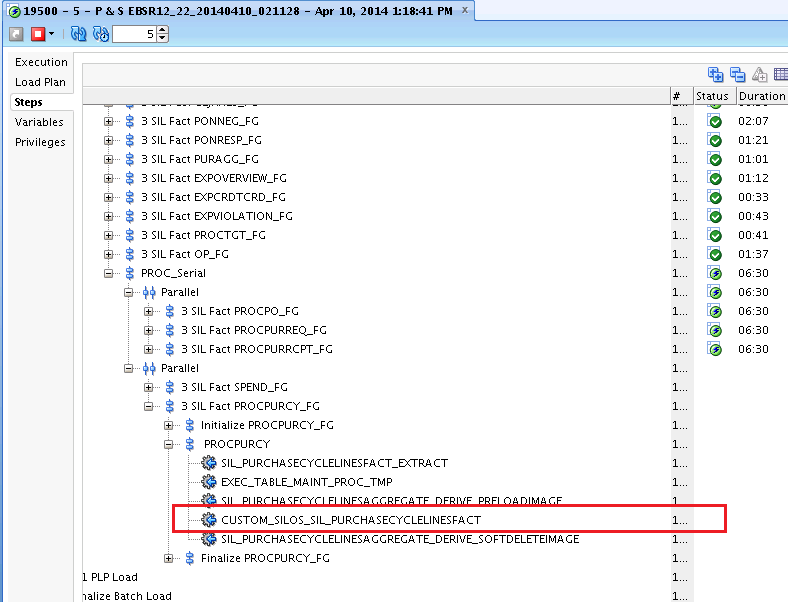
## 6.7 SIL PROCPURREQ Fact Group

SIL\_RequisitionLinesCostFact (Purchase Type DFF,



## 6.8 SIL PROCPURCY Fact Group

SIL\_PurchaseCycleLinesFact



# Purchase Cost Fact Obtain the Original Promised Date

After verifying and validation, and successful full load, execute the above task separately, this will not take more than 2 minutes.