Implementing Initial Load

Practice Overview

In this practice you will perform the following Oracle GoldenGate initial load methods:

- **File to Replicat**: in this method, the Extract process captures the source data into trails on the target system. The initial load Replicat task then reads the data from the intermediate files and apply it into the target database.
- **Direct Load**: in this method, the Extract process reads the records from the source database and send them directly to a Replicat task on the target database system.

You will perform the initial load methods mentioned above to instantiate the HR schema from the source database to the target database.

Implementing Initial Load Procedure

A. Create the target database schema

When you created the target database, you did not select the option to create the examples schema. Therefore, the HR schema and the EXAMPLES tablespace do not exist in the target database. You will create a schema HRTRG in the target database. It will own the replicated tables from the source HR schema.

1. In the **target** database (`db2`), create the following schema

```
sqlplus / as sysdba
CREATE USER HRTRG IDENTIFIED BY oracle DEFAULT TABLESPACE USERS QUOTA UNLIMITED ON
USERS;
GRANT CONNECT, RESOURCE TO HRTRG;
```

B. Buildup and prepare the target dataset metadata

In this phase of the practice, you create the data structure of the target dataset and get it ready for change synchronization. Preparing involve actions like disabling the constraints, disabling the triggers, and removing the indexes.

In the scenario of this practice, we assume the data structure of the target dataset is the same as the structure of the source dataset. You can use any tool or method to get the source metadata into the target database. This practice demonstrates using the Data Pump Import utility to export the metadata of the source dataset (specific tables in HR schema) from `db1` to the HRTRG schema in the `db2` database online through a network database link.

2. In the target database (`db2`), create a database link that will be used by the Data Pump utility

```
sqlplus / as sysdba

CREATE PUBLIC DATABASE LINK db1link connect to system identified by oracle using
'(DESCRIPTION=(ADDRESS = (PROTOCOL = TCP)(HOST = ggsrv1) (PORT=1521))
(connect_data=(service_name= db1)))';

SELECT sysdate FROM DUAL@db1link;
```

3. Create a directory object that will be used by the Data Pump utility

```
SQL> CREATE DIRECTORY gg_dir AS '/home/oracle/gg_dir';

SQL> GRANT READ,WRITE ON DIRECTORY gg_dir to SYSTEM;

$ cd

$ mkdir gg_dir
```

4. Run the Data Pump Import utility to get the metadata from the source database

```
cd $ORACLE_HOME/bin
```

```
$ impdp system/oracle@db2 directory=gg_dir schemas=hr remap_schema=hr:hrtrg
remap_tablespace=example:users logfile=hr_db1.log content=METADATA_ONLY
network_link=db1link
```

5. Verify the objects have been successfully imported.

```
sqlplus system/oracle@db1
SELECT COUNT(*) FROM DBA_TABLES WHERE OWNER IN ('HR');
SELECT COUNT(*) FROM HR.EMPLOYEES;

conn system/oracle@db2
SELECT COUNT(*) FROM DBA_TABLES WHERE OWNER IN ('HRTRG');
SELECT COUNT(*) FROM HRTRG.EMPLOYEES;
```

6. Disable the HRTRG constraints in the replicated tables.

When initial load Replicat inserts the data, the target database may reject some insert operations because they violate the constraints. That is why you disable the constraints during the initial load and enable them after the initial load is finished.

```
SET SERVEROUTPUT ON
exec DBMS_OUTPUT.ENABLE(1000)
DECLARE
 V_SCHEMA VARCHAR2(10) := 'HRTRG';
 V_SQL VARCHAR2(32000);
 CURSOR c_DisableConstraints IS
 SELECT 'alter table '|| OWNER|| '.' ||TABLE_NAME||' disable constraint
'||CONSTRAINT_NAME||' cascade' sqlstatement, TABLE_NAME, CONSTRAINT_NAME
 FROM DBA_CONSTRAINTS WHERE OWNER= V_SCHEMA
 AND TABLE_NAME IN (
'JOB_HISTORY','EMPLOYEES','JOBS','DEPARTMENTS','LOCATIONS','REGIONS')
 ORDER BY TABLE_NAME, CONSTRAINT_NAME;
BEGIN
 FOR r IN c_DisableConstraints LOOP
 BEGIN
  V_SQL := r.sqlstatement;
  EXECUTE IMMEDIATE (V_SQL);
  --DBMS_OUTPUT.PUT_LINE(R.TABLE_NAME ||'-' || R.CONSTRAINT_NAME);
 EXCEPTION
  WHEN OTHERS THEN
   DBMS_OUTPUT.PUT_LINE('FAILED: ' || V_SQL);
 END ;
 END LOOP;
END;
/
```

7. Disable the HRTRG triggers in the replicated tables.

```
SET SERVEROUTPUT ON
DECLARE
 V_SCHEMA VARCHAR2(10) := 'HRTRG';
 V_SQL VARCHAR2(32000);
 CURSOR c_DisableTriggers IS
 SELECT 'alter trigger '|| OWNER ||'.'|| TRIGGER_NAME|| ' disable' sqlstatement,
TABLE_NAME, TRIGGER_NAME
 FROM DBA_TRIGGERS WHERE OWNER= V_SCHEMA
 AND TABLE_NAME IN (
'JOB_HISTORY','EMPLOYEES','JOBS','DEPARTMENTS','LOCATIONS','REGIONS')
 ORDER BY TABLE_NAME;
BEGIN
 FOR r IN c_DisableTriggers LOOP
 BEGIN
  V_SQL := r.sqlstatement;
  EXECUTE IMMEDIATE (V_SQL);
  --DBMS_OUTPUT.PUT_LINE(R.TABLE_NAME ||'-' || R.TRIGGER_NAME);
 EXCEPTION
  WHEN OTHERS THEN
   DBMS_OUTPUT.PUT_LINE('FAILED: ' || V_SQL);
 END ;
 END LOOP;
END;
/
```

C. Obtain the code to re-create the indexes then drop them

It is a common practice to drop the indexes on the target database before the initial load is kicked off. Indexes slows down the initial load. Unique indexes may cause it to fail because even if the primary key constraints are disabled, their associated unique indexes will force the uniqueness.

Therefore, you obtain the code to re-create the indexes, drop the indexes, and after the initial load is finished create the indexes again.

8. In the target system, create a directory to save the script files in it.

```
# mkdir /home/oracle/scripts
```

9. Generate the code to re-create the indexes of the target dataset.

```
SET PAGESIZE 0
SET LONG 90000
SET FEEDBACK OFF
SET ECHO OFF
SET HEADING OFF
SET LINES 100
SET LINESIZE 200
SET MARKUP HTML OFF SPOOL OFF
SET TRIMSPOOL ON
Col STATEMENT format a10000
SPOOL /home/oracle/scripts/hrtrg_cindexs.sql REPLACE

SELECT
   DBMS_METADATA.GET_DDL('INDEX',U.INDEX_NAME,'HRTRG') || ';' AS STATEMENT
FROM DBA_INDEXES U
WHERE OWNER = 'HRTRG'
   AND TABLE_NAME IN (
'JOB_HISTORY','EMPLOYEES','JOBS','DEPARTMENTS','LOCATIONS','REGIONS');

SPOOL OFF
```

10. Generate the code to drop the indexes of the target dataset.

```
SET PAGESIZE 0
SET LONG 90000
SET FEEDBACK OFF
SET ECHO OFF
SET HEADING OFF
SET LINES 100
SET MARKUP HTML OFF SPOOL OFF
SET TRIMSPOOL ON
SPOOL /home/oracle/scripts/hrtrg_dindexs.sql REPLACE

SELECT
  'DROP INDEX HRTRG.' || INDEX_NAME || ';' AS STATEMENT
FROM DBA_INDEXES U
WHERE OWNER = 'HRTRG'
   AND TABLE_NAME IN (
'JOB_HISTORY','EMPLOYEES','JOBS','DEPARTMENTS','LOCATIONS','REGIONS');

SPOOL OFF
```

11. Open the script files and remove the first and the last lines in the file which should not be part of the script.

```
vi ~/scripts/hrtrg_cindexs.sql
vi ~/scripts/hrtrg_dindexs.sql
```

12. Execute the drop indexes script

```
sqlplus / as sysdba
@/home/oracle/scripts/hrtrg_dindexs.sql
```

D. Configure the File-to-Replicat initial load Extract and Replicat parameter files

13. In both the source and target systems, make sure the manager is started. Start it, if it is down.

```
ggsci> info mgr
ggsci> start mgr
```

14. In the **source** system, configure the initial load Extract parameter file. Add its runtime parameters as shown in the following code:

Note: Unfortunately, Oracle GoldenGate treats the views as tables and, therefore, they must be explicitly excluded.

```
cd $GG_HOME
ggsci
ggsci> edit param eftrinit
```

```
-- this parameter must be used with File to Replicat method
SourceIsTable
USERID ogg, PASSWORD oracle
RmtHost ggsrv2, MgrPort 7810
RmtFile /u01/app/oracle/product/ogg/dirdat/initload.dat, Purge
Table HR.JOB_HISTORY;
Table HR.EMPLOYEES;
Table HR.JOBS;
Table HR.DEPARTMENTS;
Table HR.LOCATIONS;
Table HR.REGIONS;
```

15. Save and close the file.

16. In the **target** system, configure the initial load delivery parameter file by adding the parameters as shown below to it:

```
cd $GG_HOME
ggsci
ggsci> edit param rftrinit
```

```
SpecialRun
End Runtime
USERID ogg, PASSWORD oracle
-- AssumeTargetDefs in GG 12c this is not needed
ExtFile ./dirdat/initload.dat
Map HR.*, Target HRTRG.*;
```

17. Save and close the file.

E. Start the File to Replicat initial load, and monitor its operation

In the following steps, you start the initial load Extract and monitor its operation. Then, you stat the initial load Replicat and monitor its operation.

18. In `ggsrv1`, start the Extract process from the OS shell prompt (not from ggsci command prompt).

The progress of the Extract will be saved in the report file, not the command prompt window.
If you need to re-run the command, you must delete or rename the report file first.

```
./extract paramfile dirprm/eftrinit.prm reportfile dirrpt/eftrinit.rpt
```

19. View the report of the initial load Extract. Observer the number of inserts in the end of the report.

```
more ./dirrpt/eftrinit.rpt
```

20. In the **target** system, start the initial load Replicat task from the OS shell prompt.

```
./replicat paramfile dirprm/rftrinit.prm reportfile dirrpt/rftrinit.rpt
```

21. When the previous task is finished, view the task report. Observe the total number of inserts in the end of the report.

```
more dirrpt/rftrinit.rpt
```

22. Verify that the data has been initialized (or instantiated) in the target database.
This is a basic verification process. It does not guarantee that the target tables are 100% the same as the source tables.

```
sqlplus system/oracle@db1
SELECT COUNT(*) FROM HR.EMPLOYEES;
conn system/oracle@db2
SELECT COUNT(*) FROM HRTRG.EMPLOYEES;
```

F. Truncate the target schema

23. Truncate the tables in the HRTRG schema in the target database. This is to prepare the database for next initial load task.

```
TRUNCATE TABLE HRTRG.JOB_HISTORY;
TRUNCATE TABLE HRTRG.EMPLOYEES;
TRUNCATE TABLE HRTRG.JOBS;
TRUNCATE TABLE HRTRG.DEPARTMENTS;
TRUNCATE TABLE HRTRG.LOCATIONS;
TRUNCATE TABLE HRTRG.REGIONS;
```

G. Configure and Execute the Direct-Load initial load

In the following steps, you will implement the direct-load initial load method to instantiate the source data. In this method, the Extract *task* reads the data row by row and sends it directly to the Replicat task which inserts the data online row by row. The tasks automatically shut down after the operation is complete.

24. In the source system, add the initial load Extract task

```
ggsci> Add Extract edlinit, SourceIsTable
```

25. Create the initial load Extract task parameter file as follows

```
ggsci> edit param edlinit
```

```
Extract edlinit
USERID ogg, PASSWORD oracle
RmtHost ggsrv2, MgrPort 7810
RmtTask Replicat, Group rdlinit
Table HR.JOB_HISTORY;
Table HR.EMPLOYEES;
Table HR.JOBS;
Table HR.DEPARTMENTS;
Table HR.LOCATIONS;
Table HR.REGIONS;
```

26. Save and close the file.

27. In the **target** system, add an access rule to the Manager parameter file to allow communication from the source system

```
edit param mgr
```

```
Port 7810
PurgeOldExtracts ./dirdat/*, UseCheckpoints, MinKeepDays 5
ACCESSRULE, PROG *, IPADDR 192.168.1.77, ALLOW
```

28. Restart the Manager process:

```
stop mgr
start mgr
info mgr
```

29. In the **target** system, add the initial load Replicat task

```
ggsci> Add Replicat rdlinit, SpecialRun
```

30. Create the initial load Replicat task parameter file as follows

```
ggsci> edit param rdlinit
```

```
Replicat rdlinit
USERID ogg, PASSWORD oracle
DiscardFile ./dirrpt/rdlinit.dsc, Purge
Map HR.*, Target HRTRG.*;
```

31. Save and close the file.

32. In the source system, start the initial load Extract

```
ggsci> start extract edlinit
```

33. To check the status of the task. Status STOPPED means it is finished.

```
ggsci> info extract edlinit
ggsci> view report edlinit
ggsci> sh tail ggserr.log
```

34. On the target system, check the report of the Replicat task and `ggserr.log` for any error

```
ggsci> view report rdlinit
ggsci> sh tail ggserr.log
```

Note: After the initial load is finished, you should create the indexes that you had dropped before starting the initial load. You may also consider enabling the constraints that you had disabled. You will start the initial load again in the next practice and therefore you will create the indexes over there.

Summary

In this practice you have implemented the initial load using two methods: file to Replicate and direct load.