Preparing the Environment for GoldenGate

Practice Overview

In this practice will perform several actions that aim at preparing the databases for GoldenGate. The following list is an overview of those actions:

- Verify primary key values uniqueness
- Verify the archiving mode is enabled
- Set up a GoldenGate database user
- Enable supplemental logging
- Enable GoldenGate Replication
- Configure and start the Manager process
- Configure the GLOBALS parameter file

Replication Target

In the practices of this course, you will replicate tables owned by the HR schema in the `db1` database to HRTRG schema in the `db2` database.

Preparing the Environment for GoldenGate Procedure

A. Verify primary key constraints are implemented

For each table included in your replication, GoldenGate needs a way to distinguish its records. The best method is to maintain either primary key constraint or unique indexes on all the tables.

1. In source database (`db1`), verify that all the tables included in your replication have primary key constraints

```
su - oracle
sqlplus system/oracle@db1

-- all tables in HR schema will be replicated
select t.OWNER, t.TABLE_NAME
from DBA_TABLES t
where not exists (select 1
                  from DBA_CONSTRAINTS c
                  where c.OWNER = t.OWNER and c.TABLE_NAME = t.TABLE_NAME
                    and c.CONSTRAINT_TYPE = 'P')
and t.OWNER='HR';
```

B. Verify the archiving mode is enabled

Make sure the archiving mode is enabled in the databases involved in your GoldenGate configuration. If not, enable it.

2. In **both** databases, login as `sys` user and enable the archiving mode:

```
conn / as sysdba
SELECT LOG_MODE FROM V$DATABASE ;

ALTER SYSTEM SET LOG_ARCHIVE_DEST_1 = 'LOCATION=USE_DB_RECOVERY_FILE_DEST'
scope=BOTH;
SHUTDOWN IMMEDIATE
STARTUP MOUNT
ALTER DATABASE ARCHIVELOG;
ALTER DATABASE OPEN;
ALTER SYSTEM SWITCH LOGFILE;
SELECT NAME FROM V$ARCHIVED_LOG;
```

C. Set up a GoldenGate database user

In both the source and target databases, you need to create a database user that will be used by GoldenGate. In source database, the GoldenGate user reads the data and transactions made on the source tables. In the target database, GoldenGate user will apply the transactions data. Grant the required privileges to the GoldenGate user.

3. In each database, create a database user that will be used by Oracle GoldenGate

```
-- in DB1
-- create tablespace dedicated to OGG
CREATE TABLESPACE OGG_TBS
DATAFILE '/u01/app/oracle/oradata/db1/oggtbs01.dbf' SIZE 100M
AUTOEXTEND ON NEXT 10M MAXSIZE 1G
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;

-- create OGG user
CREATE USER OGG IDENTIFIED BY oracle DEFAULT TABLESPACE OGG_TBS;

-- in DB2
-- create tablespace dedicated to OGG
CREATE TABLESPACE OGG_TBS
DATAFILE '/u01/app/oracle/oradata/db2/oggtbs01.dbf' SIZE 100M
AUTOEXTEND ON NEXT 10M MAXSIZE 1G
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;

-- create OGG user
CREATE USER OGG IDENTIFIED BY oracle DEFAULT TABLESPACE OGG_TBS;
```

4. Grant the required privileges to OGG user

```
-- (on both databases) grant privileges to OGG user
EXEC dbms_goldengate_auth.grant_admin_privilege('OGG')

-- check out the privileges granted to OGG:
SELECT PRIVILEGE FROM dba_sys_privs WHERE grantee='OGG'
union all
SELECT GRANTED_ROLE FROM dba_role_privs WHERE grantee='OGG'
union all
SELECT PRIVILEGE||' on '|| table_name FROM dba_tab_privs WHERE grantee='OGG';
```

```
-- (on both databases) grant further privileges to OGG:
GRANT CREATE SESSION TO OGG;
GRANT ALTER SYSTEM TO OGG;
GRANT RESOURCE TO OGG;
GRANT CONNECT TO OGG;
GRANT DBA to OGG;
GRANT SELECT ANY DICTIONARY TO OGG;
GRANT EXECUTE on DBMS_FLASHBACK TO OGG;
GRANT LOCK ANY TABLE TO OGG;

# (in db1):
GRANT SELECT ANY TRANSACTION TO OGG;

# (in db2) only required by Replicat:
GRANT INSERT ANY TABLE TO OGG;
GRANT UPDATE ANY TABLE TO OGG;
GRANT DELETE ANY TABLE TO OGG;
GRANT CREATE TABLE TO OGG;
```

D. Enable supplemental logging

5. You must enable the minimal supplemental logging in the database. You also must enable the table-level or schema-level supplemental logging for the data to replicate.

```
# in both databases
SELECT SUPPLEMENTAL_LOG_DATA_MIN, FORCE_LOGGING FROM V$DATABASE;

ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
ALTER DATABASE FORCE LOGGING;
ALTER SYSTEM SWITCH LOGFILE;
```

6. (Optional) Open a monitoring session to monitor the `ggserr.log` file. Use this monitoring session to view how to GoldenGate responds to your commands in the background and for troubleshooting.

```
tail -f $GG_HOME/ggserr.log
```

7. In **both** databases, enable GoldenGate replication

```
ALTER SYSTEM SET ENABLE_GOLDENGATE_REPLICATION=TRUE scope=both;
```

8. In the source database (`db1`), enable schema-level supplemental logging for HR schema

```
cd $GG_HOME
ggsci
ggsci> DBLogin UserID ogg, Password oracle
ggsci> ADD SCHEMATRANDATA hr
```

E. Configure and start the Manager process

We have not created the Manager process during in the installer wizard and it is time to create it now. As with all GoldenGate processes, you first configure the parameter file then you start the process.

9. At the GGSCI prompt, enter the command as shown below.

This command creates a new `mgr.prm` file and starts the vi editor to allow you to edit the parameter file. If the file already exists, it will start the editor and opens the file for you.

```
ggsci> edit Param mgr
```

10. Add the following runtime parameter to the file:

```
Port 7809
DynamicPortList 9500-9520
PurgeOldExtracts ./dirdat/*, UseCheckpoints, MinKeepDays 5
```

11. Save and close the file.

12. Start the Manager process. The following command is equivalent to the command “`start manager`”

```
ggsci> start mgr
```

13. View the status of the Manager process.

```
ggsci> Info mgr
```

14. Repeat the same steps to create and start the Manager process in the target database system (`ggsrv2`). Put the following parameters in its parameter file:

```
Port 7810
PurgeOldExtracts ./dirdat/*, UseCheckpoints, MinKeepDays 5
```

F. Configure the `GLOBALS` parameter file

The parameters that you add to the `GLOBALS` file affect all the processes in an Oracle GoldenGate instance. In the following steps, you will create the file and add a parameter to it.

15. In the source system at the `ggsci` prompt, issue the following command to create the `GLOBALS` file (uppercase, without a file extension).

```
ggsci> EDIT PARAMS ./GLOBALS
```

16. Add the parameter GGSCHEMA to it.

This parameter tells the GoldenGate which database schema it should use to store its objects.

```
GGSCHEMA OGG
```

17. Save and close the file.

18. Repeat the same steps in the target system (`ggsrv2`).

Summary

In this practice you have implemented the steps that prepare the databases for configuring Oracle GoldenGate. You are ready now to go on with configuring GoldenGate in the source and target databases.